



使用者指南

Amazon Relational Database Service



Amazon Relational Database Service: 使用者指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任從何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 Amazon RDS ?	1
概觀	1
Amazon EC2 和內部部署資料庫	1
Amazon RDS 和 Amazon EC2	3
Amazon RDS Custom for Oracle 和 Microsoft SQL Server	4
Amazon RDS AWS Outposts	4
資料庫執行個體	4
資料庫引擎	5
資料庫執行個體類別	6
資料庫執行個體儲存體	6
Amazon Virtual Private Cloud (Amazon VPC)	7
AWS 區域和可用區域	7
安全	7
Amazon RDS 監控	7
如何使用 Amazon RDS	8
AWS Management Console	8
命令列界面	8
Amazon RDS API	8
Amazon RDS 收費方式	8
後續步驟 ?	9
入門	9
資料庫引擎的特定主題	9
Amazon RDS 共同的責任模型	10
資料庫執行個體	11
資料庫執行個體類別	14
資料庫執行個體類別的類型	14
支援的資料庫引擎	21
確定資料庫執行個體類別支援 AWS 區域	73
變更資料庫執行個體類別	77
設定 RDS for Oracle 的處理器	78
硬體規格	104
資料庫執行個體儲存體	131
儲存類型	131
佈建 IOPS 儲存體	132

一般用途儲存	136
比較 SSD 儲存體類型	140
磁性儲存裝置 (舊版, 不建議使用)	142
專用記錄檔磁碟區 (DLV)	143
監控儲存體效能	144
影響儲存體效能的因素	144
區域、可用區域和 Local Zones	147
AWS 地區	148
可用區域	152
本機區域	153
依區域和引擎支援的 Amazon RDS 功能	154
資料表慣例	154
功能快速參考	155
藍/綠部署	157
跨區域自動備份	158
跨區域僅供讀取複本	159
資料庫活動串流	161
雙堆疊模式	168
將快照匯出至 S3	183
IAM 資料庫身分驗證	192
Kerberos 身分驗證	196
多可用區域資料庫叢集	209
Performance Insights	215
RDS Custom	215
Amazon RDS Proxy	223
Secrets Manager 整合	234
零 ETL 整合	234
引擎原生功能	235
Amazon RDS 資料庫執行個體計費	236
隨需資料庫執行個體	238
預留資料庫執行個體	239
設定	251
註冊一個 AWS 帳戶	251
建立具有管理權限的使用者	251
授與程式設計存取權	253
判定需求	254

提供資料庫執行個體的存取權	255
入門	259
建立並連線至 MariaDB 資料庫執行個體	260
必要條件	261
步驟 1：建立 EC2 執行個體	261
步驟 2：建立 MariaDB 資料庫執行個體	267
(選擇性) 使用建立 VPC、EC2 執行個體和 MariaDB 執行個體 AWS CloudFormation	271
步驟 3：連線至 MariaDB 資料庫執行個體	273
步驟 4：刪除 EC2 執行個體和資料庫執行個體	277
(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation	278
(選用) 將資料庫執行個體連線至 Lambda 函數	278
建立並連線至 Microsoft SQL Server 資料庫執行個體	279
必要條件	280
步驟 1：建立 EC2 執行個體	280
步驟 2：建立 SQL Server 資料庫執行個體	285
(選擇性) 使用建立 VPC 雲端、EC2 執行個體和 SQL 伺服器執行個體 AWS CloudFormation	290
步驟 3：連線至 SQL Server 資料庫執行個體	292
步驟 4：探索範例資料庫執行個體	295
步驟 5：刪除 EC2 執行個體和資料庫執行個體	297
(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation	298
(選用) 將資料庫執行個體連線至 Lambda 函數	298
建立並連線至 MySQL 資料庫執行個體	299
必要條件	300
步驟 1：建立 EC2 執行個體	300
步驟 2：建立 MySQL 資料庫執行個體	305
(選擇性) 使用建立 VPC、EC2 執行個體和 MySQL 執行個體 AWS CloudFormation	310
步驟 3：連線至 MySQL 資料庫執行個體	312
步驟 4：刪除 EC2 執行個體和資料庫執行個體	316
(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation	317
(選用) 將資料庫執行個體連線至 Lambda 函數	317
建立並連線至 Oracle 資料庫執行個體	318
必要條件	319
步驟 1：建立 EC2 執行個體	319
步驟 2：建立 Oracle 資料庫執行個體	324
(選擇性) 使用建立 VPC、EC2 執行個體和 Oracle 資料庫執行個體 AWS CloudFormation	329

步驟 3：將您的 SQL 連線至 Oracle 資料庫執行個體	331
步驟 4：刪除 EC2 執行個體和資料庫執行個體	335
(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation	335
(選用) 將資料庫執行個體連線至 Lambda 函數	336
建立並連線至 PostgreSQL 資料庫執行個體	337
必要條件	338
步驟 1：建立 EC2 執行個體	338
步驟 2：建立 PostgreSQL 資料庫執行個體	344
(選擇性) 使用建立 VPC、EC2 執行個體和 PostgreSQL 執行個體 AWS CloudFormation	348
步驟 3：連線至 PostgreSQL 資料庫執行個體	350
步驟 4：刪除 EC2 執行個體和資料庫執行個體	354
(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation	355
(選用) 將資料庫執行個體連線至 Lambda 函數	355
教學：建立 Web 伺服器和 Amazon RDS 資料庫執行個體	356
啟動 EC2 執行個體	357
建立資料庫執行個體	363
安裝 Web 伺服器	381
教學課程：建立 Lambda 函數以存取您的 Amazon RDS 資料庫執行個體	392
必要條件	393
建立 Amazon RDS 資料庫執行個體	393
建立 Lambda 函數和代理	394
建立函數執行角色	395
建立 Lambda 部署套件	396
更新 Lambda 函數	399
在主控台中測試您的 Lambda 函數	400
建立 Amazon SQS 佇列	401
建立事件來源映射以調用 Lambda 函數	402
測試和監控設定	402
清除您的資源	403
教學課程和範本程式碼	405
本指南中的教學課程	405
其他 AWS 指南中的教學	406
AWSAmazon RDS Amazon PostgreSQL 討會和實驗室內容入口網站	407
AWSAmazon RDS Amazon MySQL 的研討會和實驗室內容門戶	407
教學課程和範例程式碼 GitHub	407
使用 AWS 軟體開發套件	408

Amazon RDS 的最佳實務	410
Amazon RDS 基本操作準則	410
資料庫執行個體 RAM 建議	411
AWS 資料庫驅動	411
使用 Enhanced Monitoring 來識別作業系統問題	412
使用指標來識別效能問題	412
檢視效能指標	412
評估效能指標	415
調校查詢	417
使用 MySQL 的最佳實務	417
資料表大小	417
表格數目	418
儲存引擎	418
使用 Oracle 的最佳實務	419
資料表大小	419
表格數目	420
儲存引擎	420
使用 Oracle 的最佳實務	420
使用 PostgreSQL 的最佳實務	421
將資料載入至 PostgreSQL 資料庫執行個體	421
使用 PostgreSQL 自動清空功能	421
Amazon RDS for PostgreSQL 最佳實務影片	423
使用 SQL Server 的最佳實務	423
Amazon RDS for SQL Server 最佳實務影片	424
使用資料庫參數群組	424
自動建立資料庫執行個體的最佳實務	424
Amazon RDS 新功能視頻	425
設定資料庫執行個體	426
建立資料庫執行個體	427
必要條件	427
建立資料庫執行個體	433
可用設定	439
透過 建立 資源AWS CloudFormation	466
RDS和 AWS CloudFormation 範本	466
進一步了解 AWS CloudFormation	466
連接至資料庫執行個體	467

尋找連線資訊	467
資料庫身分驗證選項	471
加密的連線	471
存取資料庫執行個體的案例	471
使用 AWS 驅動程式連接資料庫執行個體	472
連線至執行特定資料庫引擎的資料庫執行個體	473
使用 RDS Proxy 管理連線	474
使用選項群組	475
選項群組概觀	475
建立選項群組	477
刪除選項群組	479
將選項新增至選項群組	480
列出選項群組的選項和選項設定	486
修改選項設定	487
從選項群組移除選項	490
刪除選項群組	492
使用參數群組	495
參數群組概觀	495
使用資料庫參數群組	498
使用資料庫叢集參數群組	514
比較資料庫參數群組	527
指定資料庫參數	528
從 Amazon RDS 創建 ElastiCache 緩存	535
執行個體設定的 ElastiCache 快取建立概觀	535
使用 執行個體的設定建立 ElastiCache 快取	536
管理資料庫執行個體	539
停止資料庫執行個體	540
使用案例	540
支援的資料庫引擎、類別和區域	541
支援多可用區域	541
運作方式	541
限制	543
選項和參數群組	543
公用 IP 地址	543
停止資料庫執行個體	544
啟動資料庫執行個體	545

連接 AWS 運算資源	546
連線至 EC2 執行個體	546
連線 Lambda 函數	554
修改資料庫執行個體	567
排程修改設定	568
可用設定	569
維持資料庫執行個體	598
檢視待處理的維護	598
套用更新	601
異地同步備份部署的維護	603
維護時段	604
調整資料庫執行個體的維護時段	606
使用強制作業系統更新	608
升級引擎版本	612
手動升級引擎版本	613
自動升級次要引擎版本	615
重新命名資料庫執行個體	619
重新命名以取代現有的資料庫執行個體	619
將資料庫執行個體重新開機	622
在資料庫體的使用案例	622
如何重新啟動	623
異地同步備份重新啟	623
考量事項	624
必要條件	624
重新啟動資料庫體：基本步驟	624
使用資料庫執行個體僅供讀取複本	627
概觀	628
建立僅供讀取複本	635
提升僅供讀取複本	638
監控僅供讀取複本	643
跨區域僅供讀取複本	646
標記 RDS 資源	658
為什麼要使用 RDS 標籤？	658
RDS 標籤的工作原理	659
最佳實務	661
在 Amazon RDS 中管理標籤	662

將標籤複製到資料庫快照	666
教學課程：使用標籤指定要停止的資料庫執行個體	667
使用 ARN	670
建構 ARN	670
取得現有的 ARN	676
使用儲存體	680
增加資料庫執行個體的儲存容量	680
使用儲存體自動擴展自動管理容量	683
升級儲存體檔案系統	689
修改佈建 IOPS 設定	690
I/O 密集型儲存體修改	692
修改一般用途 (gp3) 設定	693
使用專用日誌磁碟區 (DLV)	695
刪除資料庫執行個體	700
刪除資料庫執行個體的先決條件	700
刪除資料庫執行個體時的考量	700
刪除資料庫執行個體	701
設定及管理多可用區部署	704
多可用區域資料庫執行個體部署	706
將資料庫執行個體修改為多可用區域資料庫執行個體部署	708
Amazon RDS 的容錯移轉程序	709
多可用區域資料庫叢集部署	714
異地同步備份資料庫叢集執行個體類別	715
多可用區域資料庫叢集概觀	715
使用管理異地同步備份資料庫叢集 AWS Management Console	716
使用多可用區域資料庫叢集的參數群組	718
升級多可用區域資料庫叢集的引擎版本	718
使用 RDS Proxy 搭配多可用區資料庫叢集	719
複本延遲和多可用區域資料庫叢集	720
多可用區域資料庫叢集的容錯移轉程序	722
建立多可用區域資料庫叢集	726
連線至多可用區域資料庫叢集	749
連線 AWS 運算資源和多可用區域資料庫叢集	754
修改多可用區域資料庫叢集	776
重新命名多可用區域資料庫叢集	792
重新啟動多可用區域資料庫叢集	795

使用多可用區域資料庫叢集僅供讀取複本	797
搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫	807
刪除多可用區域資料庫叢集	811
異地同步備份資料庫叢集限制	813
使用 RDS 擴充 Support	814
RDS 延伸 Support 概觀	814
RDS 延伸 Support 費用	815
具備 RDS 擴充 Support 的版本	816
RDS 延伸 Support 的責任	817
建立資料庫執行個體或異地同步備份資料庫叢集或 叢集	818
RDS 延伸 Support 的注意事項	818
使用 RDS 延伸 Support 建立資料庫執行個體或異	819
檢視 RDS 延伸 Support 註冊	820
將資料庫執行個體或異地同步備份資料庫叢集還原 叢集	822
RDS 延伸 Support 的注意事項	822
使用 RDS 延伸 Support 將資料庫執行個體或異	823
使用藍/綠部署進行資料庫更新	825
Amazon RDS 藍/綠部署概觀	826
區域和版本可用性	826
優勢	827
工作流程	827
授權存取	832
考量事項	833
最佳實務	836
限制	838
建立藍/綠部署	841
準備進行藍/綠部署	841
指定變更	842
處理延遲載入	844
建立藍/綠部署	844
可用設定	846
檢視藍/綠部署	849
切換藍/綠部署	853
切換逾時	853
切換防護機制	853
切換動作	854

切換最佳實務	855
切換前驗證 CloudWatch 指標	856
切換藍/綠部署	856
切換後	858
刪除藍/綠部署	860
備份、還原和匯出資料	864
備份簡介	865
備份儲存體	865
管理自動備份	866
備份時段	866
Backup retention period (備份保留期間)	868
啟用自動備份	869
保留自動備份	871
刪除保留的自動備份	873
停用自動備份	874
不支援的 MySQL 儲存引擎	876
不支援的 MariaDB 儲存引擎	877
跨區域自動備份	878
管理手動備份	893
為單一可用區資料庫執行個體建立資料庫快照	894
建立多可用區域資料庫叢集快照	897
刪除資料庫快照	899
從資料庫 快照還原	901
參數群組	902
安全群組	902
選項群組	903
標記	903
Db2	903
Microsoft SQL Server	903
Oracle 資料庫	904
從快照還原	904
Point-in-time 回收	907
將多可用區域資料庫叢集還原至指定時間	912
從快照還原至多可用區域資料庫叢集	915
從多可用區域資料庫叢集快照還原至資料庫執行個體	918
教學：從資料庫快照還原資料庫執行個體	921

複製資料庫快照	925
限制	925
快照保留	925
複製共用快照	926
處理加密	926
增量式快照複製	927
跨區域複製	928
選項群組	932
參數群組	932
複製資料庫快照	933
共享資料庫 快照	943
共用快照	944
共用公有快照	948
共用加密快照	950
停止快照共用	953
將資料庫快照資料匯出至 Amazon S3	955
區域和版本可用性	955
限制	956
匯出快照資料概觀	957
設定對 S3 儲存貯體的存取權	958
匯出資料庫快照	963
監控快照匯出	966
取消快照匯出	968
失敗訊息	970
對 PostgreSQL 許可錯誤進行故障診斷	971
檔案命名慣例	971
資料轉換	973
使用 AWS Backup	982
監控資料庫執行個體中的指標	983
監控概觀	984
監控計畫	984
效能基準	984
效能指導方針	985
監控工具	985
檢視執行處理狀	989
檢視 Aurora 叢集中 資料庫執行個體狀態	990

檢視和回應 Amazon RDS 建議	995
檢視 Amazon RDS 建議	996
回應 Amazon RDS 建議	1017
在 Amazon RDS 主控台中檢視指標	1027
在 Amazon RDS 主控台中檢視組合指標	1030
在監控索引標籤中選擇新的監控檢視	1030
在導覽窗格中選擇具有績效詳情的新監控檢視	1031
在導覽窗格中選擇具有績效詳情的舊版檢視	1033
在導覽窗格中建立具有績效詳情的自訂儀表板	1034
在導覽窗格中選擇具有績效詳情的預先設定儀表板	1036
用 CloudWatch 監控 RDS	1038
Amazon RDS 和 Amazon CloudWatch 的概觀	1039
檢視 CloudWatch 量度	1040
將 Performance Insights 指標匯出至 CloudWatch	1046
建立 CloudWatch 警示	1051
教學課程：為資料庫叢集複本延遲建立 CloudWatch 警示	1051
使用 Performance Insights 來監控資料庫負載	1059
績效詳情概觀	1059
開啟和關閉 Performance Insights	1071
為 MariaDB 或 MySQL 開啟效能結構描述	1075
績效詳情政策	1079
使用績效詳情儀表板來分析指標	1091
檢視 Performance Insights 主動建議	1138
使用績效詳情 API 來擷取指標	1140
使用 AWS CloudTrail 記錄績效詳情呼叫	1163
使用適用於 RDS 的 DevOps大師分析效能	1166
RDS 大 DevOps師的好處	1166
RDS 的 DevOps大師如何工作	1167
為 RDS 設定 DevOps大師	1168
使用增強型監控來監控作業系統	1176
增強型監視概觀	1176
設定並啟用增強型監控	1178
在 RDS 主控台中檢視作業系統指標	1183
使用 CloudWatch Logs 檢視作業系統指標	1186
RDS 指標參考	1188
CloudWatch 適用於 RDS 的指標	1188

RDS 的 CloudWatch 維度	1202
CloudWatch Performance Insights 指標	1202
Performance Insights 的計數器指標	1205
績效詳情的 SQL 統計數字	1228
增強型監控中的作業系統指標	1240
監控事件、日誌和資料庫活動串流	1251
在 Amazon RDS 主控台中檢視日誌、事件和串流	1252
監控 RDS 事件	1255
Amazon RDS 的事件概觀	1255
檢視 Amazon RDS 事件	1257
使用 Amazon RDS 事件通知	1260
建立由 Amazon RDS 事件觸發的規則	1284
Amazon RDS 事件類別和事件訊息	1289
監控 RDS 日誌	1319
檢視並列出資料庫日誌檔案	1319
下載資料庫日誌檔案	1320
查看資料庫日誌檔案	1322
發佈至 CloudWatch Logs	1323
使用 REST 讀取日誌檔案內容	1326
MariaDB 資料庫日誌檔案	1328
Microsoft SQL Server 資料庫日誌檔案	1340
MySQL 資料庫日誌檔案	1345
Oracle 資料庫日誌檔案	1357
PostgreSQL 資料庫日誌檔案	1367
在 CloudTrail 中監控 RDS API 呼叫	1379
CloudTrail 與 Amazon RDS 整合	1379
Amazon RDS 日誌檔案項目	1380
使用資料庫活動串流來監控 RDS	1384
概要	1384
設定 Oracle 統一稽核	1390
正在配置 SQL Server 稽核	1391
開始資料庫活動串流	1392
修改資料庫活動串流	1394
取得活動串流狀態	1397
停用資料庫活動串流	1398
監控活動串流	1400

管理活動串流的存取	1438
使用 Amazon RDS Custom	1442
資料庫自訂挑戰	1442
RDS Custom 管理模型和優點	1443
RDS Custom 中共同的責任模型	1444
RDS Custom 中的支援周邊和不支援的組態	1446
RDS Custom 的主要優點	1446
RDS Custom 架構	1447
VPC	1447
RDS Custom 自動化與監控	1448
Amazon S3	1451
AWS CloudTrail	1452
RDS Custom 安全性	1454
RDS Custom 如何安全地代表您管理任務	1454
SSL 憑證	1455
保護您的 Amazon S3 儲存貯體，避免混淆代理人問題	1455
輪換 RDS Custom for Oracle 憑證以符合合規計畫	1456
使用 RDS Custom for Oracle	1461
RDS Custom for Oracle 工作流程	1461
Amazon RDS Custom for Oracle 的資料庫架構	1466
適用於 Oracle 的 RDS 自訂功能的可用性和支援	1468
RDS Custom for Oracle 需求和限制	1470
設定您的 RDS Custom for Oracle 環境	1474
使用 RDS Custom for Oracle 的 CEV	1493
設定 RDS Custom for Oracle 資料庫執行個體	1523
管理 RDS Custom for Oracle 資料庫執行個體	1540
使用 RDS Custom for Oracle 的複本	1556
備份與還原 RDS Custom for Oracle DB 資料庫執行個體	1563
在適用於 Oracle 的 RDS 自訂中使用選項群組	1573
遷移至 RDS Custom for Oracle	1581
升級 RDS Custom for Oracle 資料庫執行個體	1582
針對 RDS Custom for Oracle 進行故障診斷	1593
使用 RDS Custom for SQL Server	1611
RDS Custom for SQL Server 工作流程	1611
RDS Custom for SQL Server 的需求和限制	1614
設定您的 RDS Custom for SQL Server 環境	1654

透過 RDS Custom for SQL Server 使用自有媒體	1675
使用 RDS Custom for SQL Server 的 CEV	1677
建立並連線至 RDS Custom for SQL Server 資料庫執行個體	1697
管理 RDS Custom for SQL Server 資料庫執行個體	1708
管理 RDS Custom for SQL Server 的多可用區部署	1721
備份與還原 RDS Custom for SQL Server 資料庫執行個體	1735
將內部部署資料庫移轉至 RDS Custom for SQL Server	1751
升級 RDS Custom for SQL Server 的資料庫執行個體	1754
針對 Amazon RDS Custom for SQL Server 進行故障診斷	1756
在上使用 RDS AWS Outposts	1781
必要條件	1782
Amazon RDS 功能的支援	1783
支援的資料庫執行個體類別	1788
客戶擁有的 IP 地址	1790
使用 CoIP	1790
限制	1792
異地同步備份部署	1793
使用共同的責任模型	1793
提高可用性	1793
先決條件	1794
針對 Amazon EC2 許可權使用 API 作業	1795
在 Outposts 上建立 RDS 的資料庫執行個體	1797
建立 RDS on Outposts 的僅供讀取複本	1806
還原資料庫執行個體的考量	1809
使用 RDS Proxy	1810
區域和版本可用性	1811
限制和配額	1811
RDS for MariaDB 限制	1812
RDS for SQL Server 限制	1813
MySQL 限制	1813
PostgreSQL 限制	1814
規劃在哪裡使用 RDS Proxy	1815
RDS Proxy 概念和術語	1816
RDS Proxy 概念概觀	1816
連線集區	1817
安全性	1818

容錯移轉	1819
交易	1820
RDS Proxy 入門	1821
設定網路先決條件	1821
在 Secrets Manager 中設定資料庫登入資料	1823
設定 (IAM) 政策	1827
建立 RDS Proxy	1829
檢視 RDS Proxy	1835
透過 RDS Proxy 連線	1837
管理 RDS Proxy	1840
修改 RDS Proxy	1841
新增資料庫使用者	1847
變更資料庫密碼	1847
用戶端與資料庫連線	1848
配置連線設定	1848
避免鎖定	1851
刪除 RDS Proxy	1856
使用 RDS Proxy 端點	1857
代理端點概觀	1858
多可用區域資料庫叢集的 Proxy 端點	1858
跨 VPC 存取 RDS 資料庫	1859
建立代理端點	1860
檢視代理端點	1863
修改代理端點	1864
刪除代理端點	1865
代理端點的限制	1866
監視 RDS 代理伺服器 CloudWatch	1867
使用 RDS Proxy 事件	1872
RDS Proxy 事件	1872
RDS Proxy 範例	1875
對 RDS Proxy 進行故障診斷	1877
驗證代理的連線能力	1878
常見問題與解決方案	1879
將 RDS Proxy 與 AWS CloudFormation 搭配使用	1886
使用零 ETL 整合 (預覽)	1887
優勢	1888

重要概念	1888
預覽限制	1889
一般限制	1889
RDS for MySQL 限制	1890
Amazon Redshift 限制	1890
配額	1891
支援地區	1891
開始使用零 ETL 整合	1891
步驟 1：建立自訂資料庫參數群組。	1892
步驟 2：選取或建立來源資料庫資料庫	1892
步驟 3：建立目標 Amazon Redshift 資料倉儲	1893
後續步驟	1894
建立零 ETL 整合	1894
必要條件	1895
所需的許可	1895
建立零 ETL 整合	1898
後續步驟	1901
新增和查詢資料	1901
在 Amazon Redshift 中建立目的地資料庫	1901
將資料新增至來源資料庫資料	1901
在 Amazon 紅移中查詢您的亞馬遜 RDS 數據	1902
資料類型差異	1903
檢視和監控零 ETL 整合	1906
檢視整合	1907
使用系統資料表進行監視	1908
使用監控 EventBridge	1909
刪除零 ETL 整合	1909
對零 ETL 整合進行疑難排解	1910
我無法建立零 ETL 整合	1911
我的整合停留在一種狀態 Syncing	1911
我的表沒有複製到 Amazon Redshift	1911
我的一個或多個 Amazon Redshift 資料表需要重新同步	1912
Amazon RDS 上的 DB2	1914
數據庫 2 概述	1915
Db2 功能	1915
DB2 版本	1918

DB2 授權	1922
Db2 執行個體類別	1932
Db2 參數	1934
光大局整理	1937
當地時區	1938
執行個體先決條件	1941
管理員帳戶	1941
其他考量	1941
連接至您的 Db2 資料庫執行個體	1943
尋找端點	1943
IBM Db2 CLP	1945
IBM CLPPlus	1949
DBeaver	1951
IBM Db2 Data Management Console	1955
安全群組考量	1962
保護 Db2 連線安全	1963
使用 SSL/TLS 加密	1963
使用 Kerberos 驗證	1969
管理適用於 Db2 資料庫執行個體的 RDS	1982
系統任務	1984
資料庫任務	1994
Amazon S3 整合	2007
建立 IAM 政策	2007
建立 IAM 角色並附加您的 IAM 政策	2010
將您的 IAM 角色新增至資料庫執行個體	2012
將資料移轉至 Db2	2015
使用的遷移方法 AWS	2015
原生 Db2 工具	2021
適用於 Db2 的 RDS 選項	2032
Db2 稽核記錄	2033
外部預存程序	2047
基於 Java 的外部預存程序	2047
已知問題和限制	2055
驗證限制	2055
非圍欄例程	2055
移轉期間的非自動儲存體表格空間	2055

適用於 Db2 的 RDS 預存程序	2056
授與和撤銷權限	2057
管理緩衝集區	2070
管理資料庫	2075
管理表格空間	2094
管理稽核策略	2103
適用於 Db2 的使用者定義函數	2108
检查工作狀態	2109
Amazon RDS 上的 MariaDB	2115
MariaDB 的功能支援	2117
MariaDB 主要版本	2117
支援的儲存引擎	2123
快取預備	2125
不支援的功能	2126
MariaDB 版本	2128
支援的 MariaDB 次要版本	2128
支援的 MariaDB 主要版本	2130
MariaDB 的已取代版本	2131
連接至執行 MariaDB 的資料庫執行個體	2132
尋找連線資訊	2133
從 MySQL 命令列用戶端連線 (未加密)	2136
使用 JDBC 驅動程式連線至適用於 MariaDB 的 AWS RDS	2136
使用 Python 驅動程式連接到適用於 MariaDB 的 AWS RDS	2137
故障診斷	2137
保護 MariaDB 連線安全	2138
MariaDB 安全性	2138
使用 SSL/TLS 加密	2140
使用新的 SSL/TLS 憑證	2143
使用 RDS Optimized Reads 改善查詢效能	2148
概觀	2148
使用案例	2149
最佳實務	2149
使用	2150
監控	2150
限制	2151
使用 RDS Optimized Writes 改善 MariaDB 的寫入效能	2152

概要	2152
搭配新的資料庫使用	2153
在現有的資料庫上啟用	2157
限制	2157
升級 MariaDB 資料庫引擎	2159
概觀	2160
MariaDB 版本號	2161
RDS 版本編號	2163
主要版本升級	2164
升級 MariaDB 資料庫執行個體	2164
自動次要版本升級	2164
升級時減少停機時間	2167
將資料匯入 MariaDB 資料庫執行個體	2171
從外部資料庫匯入資料	2174
減少將資料匯入資料庫執行個體時的停機時間	2176
從任何來源匯入資料	2193
使用 MariaDB 複寫	2199
使用 MariaDB 僅供讀取複本	2199
使用外部來源執行個體設定 GTID 式複寫	2213
使用外部來源執行個體設定二進位日誌檔案位置複寫	2217
MariaDB 的選項	2222
MariaDB 稽核外掛程式支援	2222
MariaDB 的參數	2227
檢視 MariaDB 參數	2227
無法使用的 MySQL 參數	2229
將資料從 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體	2231
執行遷移	2231
MariaDB 與 MySQL 之間的不相容性	2233
Amazon RDS 上的 MariaDB SQL 參考	2234
mysql.rds_replica_status	2234
mysql.rds_set_external_master_gtid	2236
mysql.rds_kill_query_id	2239
本機時區	2240
MariaDB 的已知問題和限制	2243
檔案大小限制	2243
InnoDB 保留字詞	2245

自訂連接埠	2245
Performance Insights (績效詳情)	2245
Amazon RDS 上的 Microsoft SQL Server	2246
一般管理任務	2248
限制	2250
資料庫執行個體類別支援	2252
安全性	2258
合規計劃	2259
HIPAA	2260
SSL 支援	2261
版本支援	2261
版本管理	2264
資料庫引擎修補程式和版本	2264
棄用排程	2264
功能支援	2265
SQL 伺服器功能	2266
SQL Server 2019 功能	2266
SQL Server 2017 功能	2267
SQL Server 2016 功能	2267
SQL Server 2014 功能	2268
SQL Server 2012 已終止對 Amazon RDS 的支援	2268
SQL Server 2008 R2 已終止對 Amazon RDS 的支援	2268
CDC 支援	2268
不支援的功能和具備有限支援的功能	2269
異地同步備份部署	2270
使用 TDE	2271
函數和預存程序	2271
本機時區	2277
支援的時區	2277
Amazon RDS 上的 SQL Server 授權	2287
還原授權終止的資料庫執行個體	2287
SQL Server 開發人員版本	2287
連接至執行 SQL Server 的資料庫執行個體	2289
連線之前	2289
尋找資料庫執行個體端點和連接埠號碼	2290
使用 SSMS 連接至您的資料庫執行個體	2291

使用 SQL Workbench/J 連接至您的資料庫執行個體	2294
安全群組考量	2296
故障診斷	2296
使用 Active Directory 搭配 RDS for SQL Server	2298
搭配 SQL Server 資料庫執行個體使用自我管理 Active Directory	2299
使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄	2316
將應用程式更新為使用新的 SSL/TLS 憑證	2330
判斷任何應用程式是否使用 SSL 連線至 Microsoft SQL Server 資料庫執行個體	2330
判斷用戶端是否需要驗證憑證才能連線	2331
更新應用程式信任存放區	2333
升級 SQL Server 資料庫引擎	2334
概觀	2335
主要版本升級	2335
多可用區域和記憶體內最佳化的考量	2338
僅供讀取複本考量	2338
選項群組考量	2338
參數群組考量	2338
測試升級	2339
升級 SQL Server 資料庫執行個體	2339
在支援結束之前升級已取代的資料庫執行個體	2340
匯入和匯出 SQL Server 資料庫	2341
限制與建議	2343
設定	2344
使用原生備份與還原	2349
壓縮備份檔案	2363
故障診斷	2364
使用其他方法來匯入和匯出 SQL Server 資料	2366
使用 SQL Server 僅供讀取複本	2378
設定 SQL Server 僅供讀取複本	2378
使用 SQL Server 的僅供讀取複本限制	2378
選項考量	2379
使用 SQL Server 僅供讀取複本同步資料庫使用者和物件	2381
對 SQL Server 僅供讀取複本問題進行故障診斷	2383
RDS for SQL Server 的多可用區域	2384
將多可用區域新增至 SQL Server 資料庫執行個體	2385
從 SQL Server 資料庫執行個體移除多可用區域	2386

限制、備註和建議	2386
查明次要複本的位置	2389
遷移到 Always On AG	2390
SQL Server 的其他功能	2391
對 SQL Server 資料庫執行個體使用 SSL	2392
設定安全性通訊協定和密碼	2396
Amazon S3 整合	2402
使用 Database Mail	2422
執行個體存放區支援 tempdb	2437
使用擴充事件	2440
存取交易日誌備份	2444
SQL Server 的選項	2472
列出 SQL Server 版本的可用選項	2474
連結的伺服器搭配 Oracle OLEDB	2476
原生備份與還原	2486
透明資料加密	2490
SQL Server 稽核	2502
SQL Server Analysis Services	2511
SQL Server Integration Services	2537
SQL Server Reporting Services	2559
Microsoft Distributed Transaction Coordinator	2577
SQL Server 的常用 DBA 任務	2594
存取 tempdb 資料庫	2595
使用 Database Engine Tuning Advisor 分析資料庫工作負載	2598
將資料庫的 db_owner 變更為 rdsa 帳戶	2602
定序和字元集	2603
建立資料庫使用者	2608
判斷復原模型	2609
判斷上次容錯移轉時間	2609
停用快速插入	2610
捨棄 SQL Server 資料庫	2611
重新命名多個可用區資料庫	2611
重新設定 db_owner 角色密碼	2612
還原授權終止的資料庫執行個體	2612
將資料庫從「離線」轉移至「上線」	2613
使用 CDC	2613

使用 SQL Server 代理程式	2616
使用 SQL Server 日誌	2620
使用追蹤和傾印檔案	2622
MySQL on Amazon RDS	2624
MySQL 功能支援	2627
支援的儲存引擎	2627
使用 Memcached 與其他選項	2628
InnoDB 快取預備	2628
不支援的功能	2629
MySQL 版本	2631
支援的 MySQL 次要版本	2631
支援的 MySQL 主要版本	2634
適用於 RDS 的 RDS 延伸 Support 版本	2635
資料庫預覽環境	2636
資料庫預覽環境中的 MySQL 8.3 版	2639
資料庫預覽環境中的 MySQL 版本 8.2	2639
資料庫預覽環境中的 MySQL 8.1 版	2639
已取代的 MySQL 版本	2640
連接至執行 MySQL 的資料庫執行個體	2641
尋找連線資訊	2642
安裝 MySQL 命令列用戶端	2645
從 MySQL 命令列用戶端連線 (未加密)	2645
從 MySQL Workbench 連線	2646
使用 AWS JDBC 驅動程式連線至適用於 MySQL 的 RDS	2648
使用 AWS Python 驅動程式連接到 RDS 版	2648
故障診斷	2648
保護 MySQL 連線的安全	2650
MySQL 安全性	2650
密碼驗證外掛程式	2652
使用 SSL/TLS 加密	2653
使用新的 SSL/TLS 憑證	2656
針對 MySQL 使用 Kerberos 身分驗證	2661
使用 RDS Optimized Reads 改善查詢效能	2674
概要	2674
使用案例	2675
最佳實務	2675

使用	2676
監控	2677
限制	2677
使用 RDS Optimized Writes 改善 MySQL 的寫入效能	2678
概要	2152
搭配新的資料庫使用	2679
在現有的資料庫上啟用	2683
限制	2683
升級 MySQL 資料庫引擎	2685
概要	2686
MySQL 本號碼	2687
RDS 版本編號	2689
主要版本升級	2689
測試升級	2693
升級 MySQL 資料庫執行個體	2694
自動次要版本升級	2694
升級時減少停機時間	2697
升級 MySQL 資料庫快照集引擎版本	2701
將資料匯入 MySQL 資料庫執行個體	2704
概觀	2704
資料匯入考量	2707
將備份還原至 MySQL 資料庫執行個體	2712
從外部資料庫匯入資料	2723
匯入資料並減少停機時間	2726
從任何來源匯入資料	2743
使用 MySQL 複寫	2749
使用 MySQL 僅供讀取複本	2749
使用 GTID 式複寫	2764
使用外部來源執行個體設定二進位日誌檔案位置複寫	2771
設定多重來源複製	2775
設定主動-主動叢集	2782
使用案例	2782
考量和最佳實務	2783
跨 VPC 主動-主動式叢集的先決條件	2784
必要的參數設定	2786
將資料庫執行個體轉換為主動-主動叢集	2787

使用新的資料庫執行個體設定主動-主動叢集	2793
新增資料庫執行個體	2798
監視主動-主動叢集	2801
停止資料庫執行個體上的群組複寫	2801
重新命名主動-主動叢集中的資料庫執行個體	2802
從主動-主動叢集移除資料庫執行個體	2802
主動-主動叢集的限制	2677
從 MySQL 資料庫執行個體匯出資料	2805
準備外部 MySQL 資料庫	2805
準備來源 MySQL 資料庫執行個體	2806
複製資料庫	2807
完成匯出	2809
MySQL 的選項	2811
MariaDB 稽核外掛程式	2812
memcached	2819
MySQL 的參數	2823
MySQL 的常用 DBA 任務	2825
了解預先定義的	2825
角色型權限模型	2825
結束工作階段或查詢	2829
略過目前複寫錯誤	2829
使用 InnoDB 資料表空間來縮短損毀復原時間	2831
管理全域狀態歷史記錄	2833
本機時區	2836
已知問題和限制	2839
InnoDB 保留字詞	2839
儲存已滿行為	2839
InnoDB 緩衝集區大小不一致	2840
索引合併最佳化傳回不正確的結果	2840
Amazon RDS 資料庫執行個體的 MySQL 參數例外狀況	2841
Amazon RDS 中的 MySQL 檔案大小限制	2842
不支援 MySQL Keyring 外掛程式	2844
自訂連接埠	2844
MySQL 預存程序限制	2844
使用外部來源執行個體進行 GTID 式複寫	2844
MySQL 默認身份驗證插件	2845

覆蓋緩衝池大小	2845
RDS for MySQL 預存程序	2846
設定	2847
結束工作階段或查詢	2851
日誌	2853
管理主動-主動叢集	2855
管理多重來源複製	2859
管理全域狀態歷史記錄	2879
複寫	2882
預備 InnoDB 快取	2905
Oracle on Amazon RDS	2907
Oracle 概觀	2908
Oracle 功能	2909
Oracle 版本	2913
Oracle 授權	2918
Oracle 的使用者和權限	2921
Oracle 執行個體類別	2922
Oracle 資料庫架構	2928
Oracle 參數	2929
Oracle 字元集	2930
Oracle 限制	2934
連線至 Oracle 資料庫執行個體	2936
尋找端點	2936
SQL Developer	2938
SQL*Plus	2941
安全群組考量	2942
專用和共用伺服器程序	2942
故障診斷	2943
修改 Oracle sqlnet.ora 參數	2944
保護 Oracle 連線的安全	2949
使用 SSL 加密	2949
使用新的 SSL/TLS 憑證	2950
使用 NNE 加密	2953
設定 Kerberos 身分驗證	2954
設定 UTL_HTTP 存取	2971
使用 CDB	2983

CDB 概觀	2983
設定 CDB	2988
備份與還原 CDB	2993
將非 CDB 轉換為 CDB	2994
將單一租戶組態轉換為多租戶	2996
將 RDS for Oracle 租戶資料庫新增至 CDB 執行個體	2999
修改 RDS for Oracle 租戶資料庫	3001
從 CDB 刪除 RDS for Oracle 租戶資料庫	3003
檢視租戶資料庫詳細資訊	3006
升級 CDB	3011
管理您的 Oracle 資料庫執行個體	3012
系統任務	3027
資料庫任務	3050
日誌任務	3076
RMAN 任務	3087
Oracle Scheduler 任務	3115
診斷工作	3123
其他任務	3132
設定進階 RDS for Oracle 功能	3146
設定執行個體儲存體	3146
開啟巨型分頁	3155
開啟擴充資料類型	3158
將資料匯入 Oracle	3161
使用 Oracle SQL Developer 匯入	3162
使用 Oracle 可傳輸資料表空間進行移轉	3162
使用 Oracle Data Pump 匯入	3176
使用 Oracle Export/Import 進行匯入	3192
使用 Oracle SQL*Loader 匯入	3193
使用 Oracle 具體化檢視進行遷移	3195
使用 Oracle 複本	3197
Oracle 複本概觀	3197
Oracle 複本的要求與考量	3199
準備建立 Oracle 複本	3202
建立已掛載的 Oracle 複本	3204
修改複本模式	3205
使用 Oracle 複本備份	3206

執行 Oracle Data Guard 容錯移轉	3209
對 Oracle 複本進行故障診斷	3216
Oracle 的選項	3218
Oracle 資料庫選項概觀	3218
Amazon S3 整合	3221
Application Express (APEX)	3245
Amazon EFS 整合	3266
Java 虛擬機器 (JVM)	3282
Enterprise Manager	3286
Label Security	3305
Locator	3308
原生網路加密 (NNE)	3312
OLAP。	3325
Secure Sockets Layer (SSL)	3328
Spatial	3338
SQLT	3342
Statspack	3350
時區	3354
時區檔案自動升級	3359
透明資料加密 (TDE)	3368
UTL_MAIL	3372
XML 資料庫	3375
升級 Oracle 資料庫引擎	3376
Oracle 升級概觀	3376
主要版本升級	3380
次要版本升級	3381
升級考量	3385
測試升級	3387
升級適用於 Oracle 資料庫執行個體的 RDS	3388
升級 Oracle 資料庫快照	3390
適用於 Oracle 的工具和第三方軟體	3393
使用甲骨文 GoldenGate	3394
使用 Oracle Repository Creation Utility	3411
設定 CMAN	3418
在 Amazon RDS 上的 Oracle 上安裝 Siebel 資料庫	3421
Oracle Database 引擎版本	3425

Amazon RDS 上的 PostgreSQL	3426
一般管理任務	3427
資料庫預覽環境	3431
資料庫預覽環境中不支援的功能	3431
在資料庫預覽環境中建立新的資料庫執行個體	3432
資料庫預覽環境中的第 17 版	3433
資料庫預覽環境中的 PostgreSQL 第 16 版	3434
PostgreSQL 版本	3435
棄用 PostgreSQL 第 10 版	3435
棄用 PostgreSQL 9.6 版	3436
已棄用的 PostgreSQL 版本	3437
PostgreSQL 擴充功能版本	3438
限制安裝 PostgreSQL 擴充功能	3438
PostgreSQL 可信任延伸	3439
PostgreSQL 功能	3441
自訂資料類型和列舉	3441
RDS for PostgreSQL 適用的事件觸發程序	3442
RDS for PostgreSQL 的巨型分頁	3443
邏輯複寫	3443
stats_temp_directory 的 RAM 磁碟	3446
RDS for PostgreSQL 的資料表空間	3447
適用於 EBCDIC 和其他大型機遷移的 RDS for PostgreSQL 定序	3447
連線至 PostgreSQL 執行個體	3452
安裝 psql 客戶端	3452
尋找連線資訊	3453
使用 pgAdmin 連線至 RDS for PostgreSQL 資料庫執行個體	3455
使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體	3457
使用 JDBC 驅動程式連線至 RDS 版 AWS	3458
使用 Python 驅動程式連線至 RDS 版 AWS	3459
針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解	3459
使用 SSL/TLS 保護連線	3461
將 SSL 與 PostgreSQL 資料庫執行個體搭配使用	3461
更新應用程式以使用新的 SSL/TLS 憑證	3466
使用 Kerberos 身分驗證	3470
區域和版本可用性	3470
Kerberos 身分驗證概觀	3471

設定	3472
管理網域中的資料庫執行個體	3484
使用 Kerberos 身分驗證進行連線	3485
針對傳出網路存取使用自訂 DNS 伺服器。	3488
開啟自訂 DNS 解析	3488
關閉自訂 DNS 解析	3488
設定自訂 DNS 伺服器	3488
升級 PostgreSQL 資料庫引擎	3490
升級概觀	3491
PostgreSQL 版本編號	3493
RDS 版本編號	3493
選擇主要版本升級	3494
如何執行主要版本升級	3500
自動次要版本升級	3506
升級 PostgreSQL 延伸	3509
升級 PostgreSQL 資料庫快照引擎版本	3510
使用 RDS for PostgreSQL 的僅供讀取複本	3513
僅供讀取複本上的邏輯解碼	3513
使用 PostgreSQL 的僅供讀取複本限制	3516
使用 PostgreSQL 的僅供讀取複本組態	3517
使用階層式僅供讀取複本	3519
建立跨區域串聯僅供讀取複本	3520
複寫如何用於不同的 RDS for PostgreSQL 版本	3522
監控和調校複寫程序	3525
適用於 RDS 僅供讀取複本的疑難排解	3527
使用 RDS Optimized Reads 改善查詢效能	3529
PostgreSQL 中 RDS Optimized Reads 的概觀	3529
使用案例	3530
最佳實務	3530
使用	3531
監控	3531
限制	3532
將資料匯入 PostgreSQL	3533
從 Amazon EC2 執行個體匯入 PostgreSQL 資料庫	3535
使用 \copy 命令將資料匯入 PostgreSQL 資料庫執行個體上的資料表	3537
將資料從 Amazon S3 匯入 RDS for PostgreSQL	3538

在資料庫執行個體之間傳輸 PostgreSQL 資料庫	3556
將 PostgreSQL 資料匯出至 Amazon S3	3564
安裝擴充功能	3565
匯出至 S3 的概觀	3566
指定要匯出的 Amazon S3 檔案路徑	3566
設定對 Amazon S3 儲存貯體的存取權	3568
使用 aws_s3.query_export_to_s3 函數匯出查詢資料	3572
對 Amazon S3 的存取進行故障診斷	3575
函數參考	3575
叫用來自 RDS for PostgreSQL 的 Lambda 函數	3580
步驟 1：設定傳出連線	3581
步驟 2：為執行個體和 Lambda 設定 IAM	3581
步驟 3：安裝擴充功能	3583
步驟 4：使用 Lambda helper 函數	3584
步驟 5：叫用 Lambda 函數	3584
步驟 6：授予使用者許可權限	3586
範例：叫用 Lambda 函式	3586
Lambda 函數錯誤訊息	3589
Lambda 函數和參數參考	3590
RDS for PostgreSQL 的常用 DBA 任務	3595
RDS for PostgreSQL 中支援定序	3595
了解 PostgreSQL 角色和許可	3596
使用 PostgreSQL 自動資料清理	3609
記錄機制	3623
使用 PostgreSQL 管理暫存檔案	3624
使用 pgBadger 進行 PostgreSQL 的日誌分析	3630
使用 PgSNAPPER 監控 PostgreSQL	3630
使用參數	3630
調校 RDS for PostgreSQL 的等待事件	3647
RDS for PostgreSQL 調校的基本概念	3647
RDS for PostgreSQL 等待事件	3652
客戶端：ClientRead	3653
客戶端：ClientWrite	3656
CPU	3658
IO:BufFileRead 和 IO:BufFileWrite	3664
IO：DataFileRead	3671

IO:WALWrite	3678
Lock:advisory	3681
Lock:extend	3683
Lock:Relation	3686
Lock:transactionid	3688
Lock:tuple	3691
LWLock:BufferMapping (LWLock:buffer_mapping)	3695
LWLock:BufferIO (IPC:BufferIO)	3697
LWLock:buffer_content (BufferContent)	3699
LWLock:lock_manager (LWLock:lockmanager)	3701
Timeout:PgSleep	3705
Timeout:VacuumDelay	3706
使用 Amazon DevOps Guru 主動洞察，調校 RDS for PostgreSQL	3709
資料庫在交易連線中長時間閒置	3709
使用 PostgreSQL 擴充功能	3713
使用 Orafce 中的函數	3714
使用 pg_partman 擴充功能來管理分割區	3716
使用 PgAudit 記錄資料庫活動	3722
使用 pg_cron 擴充功能排程維護	3734
使用 pglogical 跨執行個體同步資料	3743
使用 pgactive 來建立主動-主動式複寫	3756
使用 pg_repack 擴充功能減少膨脹	3768
升級和使用 PLV8	3773
使用 PL/Rust 以 Rust 語言撰寫函數	3775
使用 PostGIS 管理空間資料	3780
支援的外部資料包裝函式	3789
使用 log_fdw 擴充功能	3789
使用 postgres_fdw 存取外部資料	3791
使用 MySQL 資料庫	3792
使用 Oracle 資料庫	3796
使用 SQL Server 資料庫	3799
使用適用於 PostgreSQL 的受信任語言延伸模組	3803
術語	3804
使用受信任語言延伸模組的需求	3804
設定受信任語言延伸模組	3807
受信任語言延伸模組概觀	3811

建立 TLE 延伸模組	3812
從資料庫中捨棄您的 TLE 延伸模組	3817
解除安裝受信任語言延伸模組	3818
搭配您的延伸模組使用 PostgreSQL 掛鉤	3819
在信任的語言延伸模組中使用自訂資料類型	3824
受信任語言延伸模組的函數參考	3825
受信任語言延伸模組的掛鉤參考	3837
程式碼範例	3840
動作	3848
CreateDBInstance	3849
CreateDBParameterGroup	3864
CreateDBSnapshot	3871
DeleteDBInstance	3879
DeleteDBParameterGroup	3887
DescribeAccountAttributes	3893
DescribeDBEngineVersions	3898
DescribeDBInstances	3905
DescribeDBParameterGroups	3915
DescribeDBParameters	3923
DescribeDBSnapshots	3932
DescribeOrderableDBInstanceOptions	3939
GenerateRDSAuthToken	3947
ModifyDBInstance	3949
ModifyDBParameterGroup	3954
RebootDBInstance	3960
案例	3963
開始使用資料庫執行個體	3964
無伺服器範例	4061
在 Lambda 函數中連接到 Amazon RDS 數據庫	4061
跨服務範例	4065
建立 Aurora 無伺服器工作項目追蹤器	4065
安全性	4070
Database authentication (資料庫身分驗證)	4071
密碼身分驗證	4072
IAM 資料庫身分驗證	4073
Kerberos 身分驗證	4073

使用 RDS 和 Secrets Manager 進行密碼管理	4075
限制	4075
概觀	4076
優勢	4076
Secrets Manager 整合所需的許可	4077
強制執行 RDS 管理	4078
管理資料庫執行個體的主要使用者密碼	4078
管理多可用區域資料庫叢集的主要使用者密碼	4082
輪換資料庫執行個體的主要使用者密碼機密	4086
輪換多可用區域資料庫叢集的主要使用者密碼機密	4088
檢視資料庫執行個體機密的詳細資訊	4090
檢視多可用區域資料庫叢集之機密的詳細資訊	4093
區域和版本可用性	4096
資料保護	4096
資料加密	4097
網際網路流量隱私權	4124
身分與存取管理	4125
物件	4125
使用身分來驗證	4126
使用政策管理存取權	4129
Amazon RDS 如何搭配 IAM 運作	4130
身分型政策範例	4137
AWS 受管理政策	4154
政策更新	4159
預防跨服務混淆代理人	4172
IAM 資料庫身分驗證	4174
故障診斷	4216
記錄和監控	4218
合規驗證	4220
彈性	4221
備份和還原	4221
複寫	4221
容錯移轉	4221
基礎設施安全	4223
安全群組	4223
Public accessibility (公開存取性)	4223

VPC 端點 (AWS PrivateLink)	4225
考量	4225
可用性	4225
建立介面 VPC 端點	4226
建立 VPC 端點政策	4227
安全最佳實務	4228
使用安全群組控制存取	4229
VPC 安全群組概觀	4229
安全群組案例	4230
建立 VPC 安全群組	4231
與資料庫執行個體建立關聯	4231
主要使用者帳戶權限	4232
服務連結角色	4236
Amazon RDS 的服務連結角色許可	4236
Amazon RDS Custom 的服務連結角色許可	4239
搭配使用 Amazon RDS 與 Amazon VPC	4241
在 VPC 中使用資料庫執行個體	4241
更新資料庫執行個體的 VPC	4256
在 VPC 中存取資料庫執行個體的案例	4257
教學課程：建立要與資料庫執行個體搭配使用的 VPC (僅限 IPv4)	4263
教學課程：建立要與資料庫執行個體搭配使用的 (VPC)(雙堆疊模式)	4270
將資料庫執行個體移入 VPC。	4280
配額和條件限制	4282
Amazon RDS 中的配額	4282
Amazon RDS 中的命名限制	4286
資料庫連線數上限	4288
Amazon RDS 中的檔案大小限制	4290
故障診斷	4291
無法連線至資料庫執行個體	4291
測試資料庫執行個體連線	4293
對連線身分驗證進行故障診斷	4294
安全問題	4294
錯誤訊息「無法擷取帳戶屬性，某些主控台功能可能受損。」	4295
對不相容的網路狀態進行故障診斷	4295
原因	4295
解析度	4295

重新設定資料庫執行個體擁有者密碼	4297
資料庫執行個體停機或重新開機	4297
參數變更未生效	4298
資料庫執行個體用盡儲存體	4298
資料庫執行個體容量不足	4300
RDS 可用記憶體問題	4300
MySQL 和 MariaDB 問題	4301
MySQL 及 MariaDB 連線上限	4301
診斷和解決記憶體限制的不相容參數狀態	4302
診斷和解決僅供讀取複本之間的延遲	4303
診斷和解決 MySQL 或 MariaDB 僅供讀取複寫失敗	4305
建立將二進位日誌啟用的觸發器需要 SUPER 權限	4307
診斷和解決point-in-time 還原失敗	4308
複寫已停止錯誤	4309
僅供讀取複本建立失敗或複寫中斷，出現嚴重錯誤 1236	4309
無法將備份保留期間設定為 0	4310
Amazon RDS API 參考	4311
使用查詢 API	4311
查詢參數	4311
查詢請求身分驗證	4311
對應用程式進行故障診斷	4312
擷取錯誤	4312
對秘訣進行故障診斷	4312
文件歷史記錄	4314
舊版更新	4428
AWS 詞彙表	4453
.....	mmmmcdliv

什麼是 Amazon Relational Database Service (Amazon RDS) ?

Amazon Relational Database Service (Amazon RDS) 是一項 Web 服務，可以讓 AWS 雲端中關聯式資料庫的設定、操作和擴展更加簡單。其能為產業標準的關聯式資料庫提供具成本效益、可調整大小的容量，並管理常見的資料庫管理任務。

Note

本指南涵蓋 Amazon Aurora 以外的 Amazon RDS 資料庫引擎。如需使用 Amazon Aurora 的相關資訊，請參閱 [《Amazon Aurora 使用者指南》](#)。

如果您不熟悉 AWS 產品和服務，請利用下列資源開始學習更多資訊：

- 如需所有 AWS 產品的概觀，請參閱 [何謂雲端運算？](#)
- Amazon Web Services 提供多項資料庫服務。若要進一步了解 AWS 上可用的資料庫選項，請參閱 [選擇 AWS 資料庫服務](#) 和 [在 AWS 上執行資料庫](#)。

Amazon RDS 概觀

為什麼要在 AWS 雲端中執行關聯式資料庫？因為 AWS 接管關聯式資料庫的許多困難和繁瑣的管理工作。

主題

- [Amazon EC2 和內部部署資料庫](#)
- [Amazon RDS 和 Amazon EC2](#)
- [Amazon RDS Custom for Oracle 和 Microsoft SQL Server](#)
- [Amazon RDS AWS Outposts](#)

Amazon EC2 和內部部署資料庫

Amazon Elastic Compute Cloud (Amazon EC2) – 在 AWS 雲端中提供可擴展的運算容量。Amazon EC2 可減少前期所需的硬體投資，讓您更快速開發並部署應用程式。

購買內部部署伺服器時，CPU、記憶體、儲存空間和 IOPS 都是一起搭售。使用 Amazon EC2，這些都是分開的個別元件，方便您日後可以獨立擴展。如果您需要更多 CPU、較少 IOPS 或更多儲存空間，都可以輕鬆配置。

對於內部部署伺服器中的關聯式資料庫，您必須承擔伺服器、作業系統和軟體的全部責任。對於 Amazon EC2 執行個體上的資料庫，AWS 會管理作業系統下方的層。透過這種方式，Amazon EC2 消除了管理內部部署資料庫伺服器的部分負擔。

在下表中，您可以找到內部部署資料庫與 Amazon EC2 的管理模型比較。

功能	內部部署管理	Amazon EC2 管理
應用程式最佳化	客戶	客戶
擴展	客戶	客戶
高可用性	客戶	客戶
資料庫備份	客戶	客戶
資料庫軟體修補	客戶	客戶
資料庫軟體安裝	客戶	客戶
作業系統 (OS) 修補	客戶	客戶
作業系統安裝	客戶	客戶
伺服器維護	客戶	AWS
硬體生命週期	客戶	AWS
電源、網路和冷卻	客戶	AWS

Amazon EC2 不是全受管服務。因此，當您在 Amazon EC2 上執行資料庫時，您更容易出現使用者錯誤。例如，當您手動更新作業系統或資料庫軟體時，可能會意外造成應用程式停機。您可能會花費數小時檢查每項變更，以識別並修正問題。

Amazon RDS 和 Amazon EC2

Amazon RDS 是受管資料庫服務。它負責大部分的管理任務。透過消除繁瑣的手動任務，Amazon RDS 讓您可以專注於應用程式和使用者。對於大多數資料庫部署，我們建議您透過 Amazon EC2 使用 Amazon RDS 做為預設選擇。

在下表中，您可以找到 Amazon EC2 與 Amazon RDS 的管理模型比較。

功能	Amazon EC2 管理	Amazon RDS 管理
應用程式最佳化	客戶	客戶
擴展	客戶	AWS
高可用性	客戶	AWS
資料庫備份	客戶	AWS
資料庫軟體修補	客戶	AWS
資料庫軟體安裝	客戶	AWS
作業系統修補	客戶	AWS
作業系統安裝	客戶	AWS
伺服器維護	AWS	AWS
硬體生命週期	AWS	AWS
電源、網路和冷卻	AWS	AWS

Amazon RDS 透過未全受管的資料庫部署，提供下列特定優點：

- 您可以使用您已經熟悉的資料庫產品：Db2、MariaDB、Microsoft SQL 伺服器、MySQL、甲骨文和 PostgreSQL。
- Amazon RDS 會管理備份、軟體修補、自動故障偵測與復原作業。
- 您可以開啟自動備份，或手動建立您自己的備份快照。您可使用這些備份還原資料庫。Amazon RDS 的還原流程可靠又有效率。

- 發生問題時，您有主要執行個體與同步次要執行個體可以容錯移轉，從而享有高可用性。您也可以使用僅供讀取複本提高讀取擴展性。
- 除了您資料庫套件內含的安全機制外，您可以協助控制誰可以存取 RDS 資料庫。若要這麼做，您可以使用 AWS Identity and Access Management (IAM) 來定義使用者和許可。您也可以將資料庫置於 Virtual Private Cloud (VPC)，協助保護資料庫安全。

Amazon RDS Custom for Oracle 和 Microsoft SQL Server

Amazon RDS Custom 是 RDS 管理類型，讓您完全存取資料庫和作業系統。

您可以使用 RDS Custom 的控制功能來存取和自訂舊版和封裝商務應用程式的資料庫環境和作業系統。同時，Amazon RDS 會自動化資料庫管理任務和操作。

在此部署模型中，您可以安裝應用程式並變更組態設定，以符合您的應用程式。同時，您可以卸載資料庫管理工作，例如佈建、擴展、升級和備份至 AWS。您可以利用 Amazon RDS 的資料庫管理優勢，並具有更大的控制和彈性。

對於 Oracle 資料庫和 Microsoft SQL Server，RDS Custom 結合 Amazon RDS 的自動化與 Amazon EC2 的彈性。如需 RDS Custom 的詳細資訊，請參閱 [使用 Amazon RDS Custom](#)。

使用 RDS Custom 的共同的責任模型，您會獲得比在 Amazon RDS 中更多的控制權，也會承擔更多責任。如需詳細資訊，請參閱 [RDS Custom 中共同的責任模型](#)。

Amazon RDS AWS Outposts

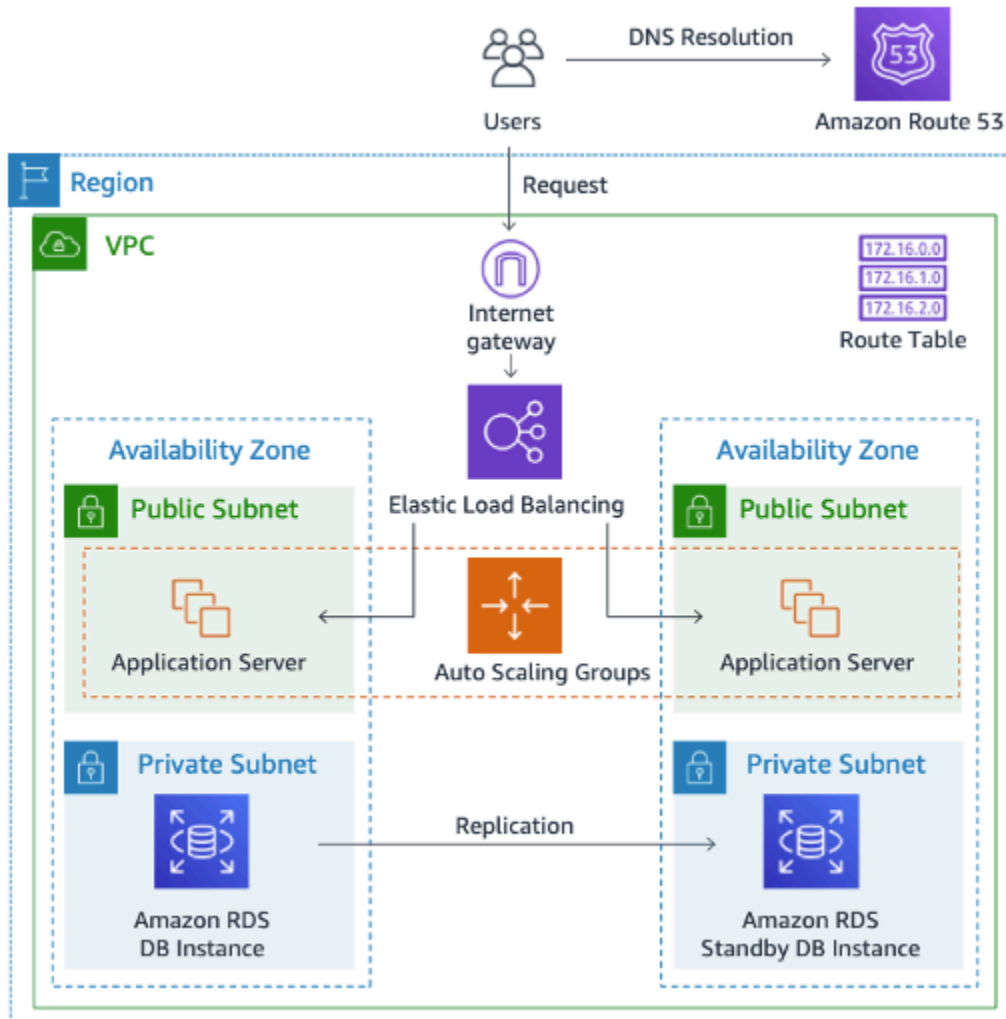
Amazon RDS 將適用於 SQL 伺服器的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 資料庫的 RDS 資料庫 AWS Outposts 延伸到 AWS Outposts 環境中。AWS Outposts 使用與公共場所相同的硬體，AWS 區域 將 AWS 服務、基礎架構和作業模型帶入內部部署。透過 Outposts 上的 RDS，您可以將受管資料庫執行個體佈建至接近必須在內部部署執行的商業應用程式之處。如需詳細資訊，請參閱 [與 Amazon RDS 上的工作 AWS Outposts](#)。

資料庫執行個體

資料庫執行個體是在 AWS 雲端執行的隔離資料庫環境。Amazon RDS 的基本建置組塊為資料庫執行個體。

您的資料庫執行個體可以包含一或多個使用者建立的資料庫。您可以使用您搭配獨立資料庫執行個體使用的同一工具和應用程式來建立資料庫執行個體。您可以使用 AWS Command Line Interface (AWS CLI)、Amazon RDS API 或 AWS Management Console。

下圖顯示使用 Amazon RDS 進行資料庫儲存的動態網站的典型使用案例。AWS 透過 Elastic Load Balancing 將要求轉送至應用程式伺服器，藉此路由使用者流量。這些應用程式伺服器與 RDS 資料庫執行個體互。應用程式伺服器和資料庫執行個體位於相同 Virtual Private Cloud (VPC) (VPC) 內的不同可用區域 (AZ) 中。主要資料庫執行個體會複製到另一個資料庫執行個體，稱為僅供讀取複本。這兩個資料庫執行個體都位於 VPC 內的私有子網路中，這表示網際網路使用者無法直接存取它們。



資料庫引擎

資料庫引擎是在資料庫執行個體上執行的特定關聯式資料庫軟體。Amazon RDS 目前支援下列引擎：

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle

- PostgreSQL

每個資料庫引擎都有各自支援的功能，資料庫引擎的各個版本可能會包含特定功能。對 Amazon RDS 功能的 Support 會因每個資料庫引擎的特定版本而有所不同。AWS 區域 若要檢查不同引擎版本和區域的功能支援，請參閱 [Amazon RDS 由 AWS 區域 和數據庫引擎支持的功能](#)。

此外，每個資料庫引擎的資料庫參數群組都具有參數集，可控制所管理之資料庫的行為。

資料庫執行個體類別

資料庫執行個體類別會決定資料庫執行個體的運算和記憶體容量。資料庫執行個體類別由資料庫執行個體類型和大小組成。每個執行個體類型皆提供不同的運算、記憶體與儲存功能。例如，db.m6g 是由重力 2 處理器提供支援的一般用途資料庫執行個體類型。AWS 在 db.m6g 執行個體類型內，db.m6g.2xlarge 是資料庫執行個體類別。

您可以選擇最符合您需求的資料庫執行個體。若您的需求隨時間而有所改變，可變更資料庫執行個體。如需相關資訊，請參閱「[資料庫執行個體類別](#)」。

Note

如需資料庫執行個體類別的定價資訊，請參閱 [Amazon RDS](#) 產品頁面的定價一節。

資料庫執行個體儲存體

Amazon EBS 提供可連接至執行中之執行個體的耐久區塊層級儲存體磁碟區。資料庫執行個體儲存有下列類型：

- 一般用途 (SSD)
- 佈建的 IOPS (PIOPS)
- 磁帶

儲存類型在效能特性和價格方面有所不同。您可以量身打造儲存效能和成本以符合您的資料庫需求。

各資料庫執行個體會最低與最高的儲存需求，需視儲存類型與支援的資料庫引擎而定。具有足夠儲存空間很重要，如此您的資料庫才有空間成長。此外，足夠的儲存空間可確保資料庫引擎的功能有空間寫入內容或日誌項目。如需詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

Amazon Virtual Private Cloud (Amazon VPC)

您可以使用 Amazon Virtual Private Cloud (Amazon VPC) 服務，在 Virtual Private Cloud (VPC) 上執行資料庫執行個體。使用 VPC 時，您可以掌控您的虛擬聯網環境。您可以選擇自己的 IP 地址範圍、建立子網路，以及設定路由和存取控制清單。無論是否在 VPC 中執行，Amazon RDS 都可提供相同的基本功能。Amazon RDS 會管理備份、軟體修補、自動故障偵測與復原作業。在 VPC 中執行資料庫執行個體無需額外成本。如需搭配 RDS 使用 Amazon VPC 的詳細資訊，請參閱[Amazon VPC](#) 和 [Amazon RDS](#)。

Amazon RDS 使用網路時間協定 (NTP) 同步資料庫執行個體的時間。

AWS 區域和可用區域

Amazon 雲端運算資源存放於全球不同區域 (例如北美、歐洲或亞洲) 的高可用性資料中心設施。每個資料中心位置稱為「AWS 區域」。

每個 AWS 區域都包含多個不同的位置，稱為可用區域或 AZ。每個可用區域旨在隔離其他可用區域的故障。每個都經過精心設計，可為相 AWS 同區域中的其他可用區域提供低成本、低延遲的網路連線能力。藉由在個別的可用區域中啟動執行個體，您就可以保護應用程式免於發生單點故障。如需更多詳細資訊，請參閱 [區域、可用區域和 Local Zones](#)。

您可以在多個可用區域中執行資料庫執行個體，這種方法稱為「異地同步備份部署」。選擇這個選項後，Amazon 會自動佈建並維護一個或多個不同可用區域的次要待命資料庫執行個體。您的主要資料庫執行個體會跨可用區域複寫至每個次要資料庫執行個體。此方法協助提供資料備援與容錯移轉支援、消除輸入/輸出凍結，以及在系統備分期間將延遲峰值降至最低。在多可用區域資料庫叢集部署中，次要資料庫執行個體也可為讀取流量提供服務。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。

安全

安全群組可控制對資料庫執行個體的存取。方法是允許使用者存取您所指定的 IP 地址範圍或 Amazon EC2 執行個體，達到控管的目的。

如需安全群組的詳細資訊，請參閱[Amazon RDS 中的安全](#)。

Amazon RDS 監控

追蹤資料庫執行個體的效能與運作狀態有多種方式。您可以使用 Amazon CloudWatch 服務監控資料庫執行個體的效能和運作狀態。CloudWatch 效能圖表會顯示在 Amazon RDS 主控台中。您也可以訂閱

Amazon RDS 事件，如此一來，只要資料庫執行個體、資料庫快照或資料庫參數群組有所變更，就能收到通知。如需詳細資訊，請參閱 [監控 Amazon RDS 執行個體中的指標](#)。

如何使用 Amazon RDS

與 Amazon RDS 互動有多種方式。

AWS Management Console

AWS Management Console 這是一個簡單的基於 Web 的用戶界面。不需編寫程式，即可從主控台管理您的資料庫執行個體。若要存取 Amazon RDS 主控台，請登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。

命令列界面

您可以使用 AWS Command Line Interface (AWS CLI) 以互動方式存取 Amazon RDS API。若要安裝 AWS CLI，請參閱 [安裝指 AWS 命令行介面](#)。若要開始使 AWS CLI 用 RDS，請 [AWS Command Line Interface 參閱 Amazon RDS 的參考資料](#)。

Amazon RDS API

如果您是開發人員，可以透過編寫程式使用 API 存取 Amazon RDS。如需詳細資訊，請參閱 [Amazon RDS API 參考](#)。

對於應用程式開發，我們建議您使用其中一個 AWS 軟體開發套件 (SDK)。AWS SDK 會處理低階詳細資料，例如驗證、重試邏輯和錯誤處理，以便您可以專注於應用程式邏輯。AWS SDK 可用於各種語言。如需詳細資訊，請參閱 [Amazon Web Services 適用工具](#)。

AWS 也提供程式庫、範例程式碼、教學課程和其他資源，協助您更輕鬆地開始使用。如需詳細資訊，請參閱 [範本程式碼與程式庫](#) 一文。

Amazon RDS 收費方式

當您使用 Amazon RDS 時，可以選擇使用隨需執行個體或預留的資料庫執行個體。如需更多詳細資訊，請參閱 [Amazon RDS 的資料庫執行個體計費](#)。

如需 Amazon RDS 定價資訊，請參閱 [Amazon RDS 產品頁面](#)。

後續步驟？

上一節為您介紹 RDS 提供的基本基礎設施元件。接下來該怎麼做？

入門

請使用[Amazon RDS 入門](#)中的指示，來建立資料庫執行個體。

資料庫引擎的特定主題

您可以在下列章節中檢閱特定資料庫引擎的專用資訊：

- [適用於 Db2 的 Amazon RDS](#)
- [Amazon RDS for MariaDB](#)
- [Amazon RDS for Microsoft SQL Server](#)
- [Amazon RDS for MySQL](#)
- [Amazon RDS for Oracle](#)
- [Amazon RDS for PostgreSQL](#)

Amazon RDS 共同的責任模型

Amazon RDS 負責託管資料庫執行個體和資料庫叢集的軟體元件和基礎設施。您則負責查詢調校，這是調整 SQL 查詢以提高效能的程序。查詢效能大幅取決於資料庫設計、資料大小、資料分佈、應用程式工作負載和查詢模式，而這些模式可能差異很大。監控和調校是您針對 RDS 資料庫的高度個人化程序。您可以使用 Amazon RDS Performance Insights 和其他工具識別出有問題的查詢。

Amazon RDS 資料庫執行個體

資料庫執行個體是在雲端執行的隔離資料庫環境。這是 Amazon RDS 的建立基礎。資料庫執行個體可能包含多個由使用者建立的資料庫，可以使用您用來存取獨立資料庫執行個體的相同用戶端工具和應用程式，來存取資料庫執行個體。您可以使用 AWS 命令列工具、Amazon RDS API 操作或 AWS Management Console，藉此輕鬆地建立和修改資料庫執行個體。

Note

Amazon RDS 可支援使用任何標準 SQL 用戶端應用程式來存取資料庫。但是，Amazon RDS 並不允許直接存取主機。

您最多可以擁有 40 個 Amazon RDS 資料庫執行個體，但有下列限制：

- 每個 SQL Server 資料庫版本 (Enterprise、Standard、Web 和 Express) 最多有 10 個「已包含授權」模式的執行個體
- "license-included" 模式下的 10 for Oracle
- 「bring-your-own-license」授權模式下的 Db2 為 40
- 40 個資料庫執行個體可以是 MySQL、MariaDB 或 PostgreSQL
- 40 適用於甲骨文在 "bring-your-own-license" (BYOL) 授權模式下

Note

如果您的應用程式需要更多的資料庫執行個體，您可以使用[此表單](#)請求額外的資料庫執行個體。

每個執行個體都有一個資料庫執行個體識別符。這是由客戶提供的唯一名稱，可在 Amazon RDS API 和 AWS CLI 命令互動時識別資料庫執行個體。在 AWS 區域內，資料庫執行個體識別符必須是該客戶的唯一標記。

資料庫執行個體識別符是組成 DNS 主機名稱的一部分，此名稱會由 RDS 配置給您的執行個體。例如，如果您指定 db1 做為資料庫執行個體識別符，則 RDS 會自動為您的執行個體配置 DNS 端點。範例端點為 `db1.abcdefghijkl.us-east-1.rds.amazonaws.com`，其中 `db1` 是您的執行個體 ID。

在範例端點 `db1.abcdefghijkl.us-east-1.rds.amazonaws.com` 中，字串 `abcdefghijkl` 是特定 AWS 區域和 AWS 帳戶組合的唯一識別符。範例中的識別符 `abcdefghijkl` 是由 RDS 在內部產生，並且對於指定的區域和帳戶組合不會變更。因此，您在此區域中的所有資料庫執行個體都會共用相同的固定識別符。請考慮固定識別符的下列功能：


- 如果您重新命名資料庫執行個體，則端點會有所不同，但固定識別符會相同。例如，如果您將 `db1` 重新命名為 `renamed-db1`，新的執行個體端點會是 `renamed-db1.abcdefghijkl.us-east-1.rds.amazonaws.com`。
- 如果您刪除並重新建立具有相同資料庫執行個體識別符的資料庫執行個體，則端點會相同。
- 如果您使用相同的帳戶在不同區域中建立資料庫執行個體，則內部產生的識別符會有所不同，因為區域不同，如 `db2.mnopqrstuvw.us-west-1.rds.amazonaws.com` 中所示。

每個資料庫執行個體皆可支援一個資料庫引擎。Amazon RDS 目前支援 Db2、MySQL、MariaDB、PostgreSQL、甲骨文、Microsoft SQL 伺服器 and Amazon Aurora 資料庫引擎。

建立資料庫執行個體時，某些資料庫引擎需要指定資料庫名稱。資料庫執行個體可以託管多個資料庫、單一 Db2 資料庫或具有多個結構描述的單一 Oracle 資料庫。資料庫名稱的值取決於資料庫引擎：

- 對於 Db2 資料庫引擎，資料庫名稱是資料庫執行個體中託管的資料庫名稱。如果要使用 Amazon RDS 存放程序 [建立](#) 或 [刪除](#) 資料庫，請勿在建立資料庫執行個體時輸入資料庫名稱。
- MySQL 和 MariaDB 資料庫引擎的資料庫名稱，是指託管在您的資料庫執行個體中的資料庫名稱。由同一資料庫執行個體託管的資料庫，在該執行個體內都必須有唯一的名稱。
- Oracle 資料庫引擎的資料庫名稱是用於設定 ORACLE_SID 的值，必須在連線至 Oracle RDS 執行個體時提供。
- Microsoft SQL Server 資料庫引擎不支援資料庫名稱參數。
- PostgreSQL 資料庫引擎的資料庫名稱是指託管在您的資料庫執行個體中的資料庫名稱。建立資料庫執行個體時不需要資料庫名稱。由同一資料庫執行個體託管的資料庫，在該執行個體內都必須有唯一的名稱。

Amazon RDS 會為您的資料庫執行個體建立主要使用者帳戶，這是建立程序的一部分。此主要使用者有許可來建立資料庫，以及對主要使用者建立的資料表執行建立、刪除、選取、更新、插入操作。在建立資料庫執行個體時，請務必設定主要使用者密碼，但您隨時都能利用 AWS CLI、Amazon RDS API 操作或 AWS Management Console 來更改該密碼。您也可以使用標準 SQL 命令來變更主要使用者密碼及管理使用者。

 Note

本指南亦涵蓋非 Aurora Amazon RDS 資料庫引擎的使用方式。如需使用 Amazon Aurora 的相關資訊，請參閱 [《Amazon Aurora 使用者指南》](#)。

資料庫執行個體類別

資料庫執行個體類別會決定 Amazon RDS 資料庫執行個體的運算和記憶體容量。您需要的資料庫執行個體類別取決於您的處理能力和記憶體需求。

資料庫執行個體類別由資料庫執行個體類別類型和大小組成。例如，db.r6g 是記憶體最佳化的資料庫執行個體類別類型，由重力 2 處理器提供支援。AWS 在 db.r6g 執行個體類別類型內，db.r6g.2xlarge 是資料庫執行個體類別。此類別的大小是 2xlarge。

如需執行個體類別定價的詳細資訊，請參閱 [Amazon RDS 定價](#)。

主題

- [資料庫執行個體類別的類型](#)
- [資料庫執行個體類別的支援資料庫引擎](#)
- [確定資料庫執行個體類別支援 AWS 區域](#)
- [變更資料庫執行個體類別](#)
- [針對 RDS for Oracle 中的資料庫執行個體類別設定處理器](#)
- [的資料庫執行個體類別的硬體規格](#)

資料庫執行個體類別的類型

Amazon RDS 支援下列使用案例的資料庫執行個體類別：

- [一般用途](#)
- [記憶體最佳化](#)
- [運算最佳化](#)
- [爆量效能](#)
- [Optimized Reads](#)

如需更多有關 Amazon EC2 執行個體類型的資訊，請參閱 Amazon EC2 文件中的 [執行個體類型](#)。

一般用途執行個體類別類型

以下是可用的一般用途資料庫執行個體類別：

- db.m7g — 由重力 on3 處理器提供支援的一般用途資料庫執行個體類別。AWS 這些執行個體類別可為廣泛的一般用途工作負載提供平衡的運算、記憶體和網路。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton3 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.m6g — 由重力 2 處理器提供支援的一般用途資料庫執行個體類別。AWS 這些執行個體可為廣泛的一般用途工作負載提供平衡的運算、記憶體和網路。db.m6gd 執行個體類別具有本機 NVMe 型 SSD 區塊層級儲存體，適用於需要高速、低延遲本機儲存的應用程式。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton2 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.m6i - 採用第三代 Intel Xeon 可擴充處理器的一般用途資料庫執行個體類別。這些執行個體通過 SAP 認證，非常適合像是支援企業應用程式的後端伺服器、遊戲伺服器、快取機群，以及應用程式開發環境等工作負載。db.m6id 和 db.m6idn 執行個體類別提供高達 7.6 TB 的本機 NVMe 型 SSD 儲存體，而 db.m6in 則提供 EBS 專用的儲存體。db.m6in 和 db.m6idn 類別提供高達 200 Gbps 的網路頻寬。
- db.m5 – 一般用途資料庫執行個體類別，可在運算、記憶體和網路資源之間取得平衡，是許多應用程式的理想選擇。db.m5d 執行個體類別提供實際連線到主機伺服器的 NVMe 型 SSD 儲存體。與先前的 db.m5 執行個體類別相比，db.m4 執行個體類別能提供更多運算容量。它們是採用 AWS Nitro 系統技術，結合了專用硬體和輕量型 Hypervisor。
- db.m4 – 一般用途資料庫執行個體類別，相較於先前的 db.m3 執行個體類別，其可提供更多運算容量。

對於 RDS for Oracle 資料庫引擎，Amazon RDS 不再支援 db.m4 資料庫執行個體類別。如果您先前建立了 RDS for Oracle db.m4 資料庫執行個體，Amazon RDS 會自動將這些資料庫執行個體類別升級至對等的 db.m5 資料庫執行個體類別。

對於適用於 MariaDB 的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 的資料庫引擎，Amazon RDS 已按照下列時間表啟動此資料庫執行個體類別的 end-of-support 程序。對於使用此執行個體類別的所有 RDS 資料庫執行個體，建議您盡快升級至新一代的資料庫執行個體類別。

動作或建議	日期
從這個日期開始，Amazon RDS 開始使用 db.m4 將執行個體自動升級到新一代 db.m5 執行個體類別。不再支援使用 db.m4 執行個體類別建立資料庫執行個體。	2024 年 6 月 1 日
Amazon RDS 終止對 db.m4 的支持。	2024 年 12 月 31 日

- db.m3 – 一般用途資料庫執行個體類別，相較於先前的 db.m1 執行個體類別，其可提供更多運算容量。

對於適用於 MariaDB 的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 的資料庫引擎，Amazon RDS 已使用下列排 end-of-life 程啟動 db.m3 資料庫執行個體類別的程序，其中包括升級建議。對於所有使用 db.m3 資料庫執行個體類別的 RDS 資料庫執行個體，建議您盡快升級至更高一代的資料庫執行個體類別。

動作或建議	日期
您再也不能建立使用 db.m3 資料庫執行個體類別的 RDS 資料庫執行個體。	現在
Amazon RDS 開始將使用 db.m3 資料庫執行個體類別的 RDS 資料庫執行個體自動升級至對等的 db.m5 DB 執行個體類別。	2023 年 2 月 1 日

記憶體優化執行個體類別類型

記憶體最佳化 Z 系列支援下列執行個體類別：

- db.z1d – 執行個體類別，已針對記憶體密集型應用程式進行最佳化處理。這些執行個體類別可同時提供大量的運算容量和記憶體。高頻 z1d 執行個體提供可持續運作且具備高達 4.0 GHz 的全核心頻率。

記憶體最佳化 X 系列支援下列執行個體類別：

- db.x2g — 針對記憶體密集型應用程式進行最佳化，並由 Graviton2 處理器提供支援的執行個體類別。AWS 這些執行個體類別為每 GiB 記憶體提供的成本低。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton2 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.x2i – 執行個體類別，已針對記憶體密集型應用程式進行最佳化處理。db.x2iedn 和 db.x2idn 執行個體類別類型採用第三代 Intel Xeon 可擴充處理器 (Ice Lake)。其中包括高達 3.8 TB 的本機 NVMe SSD 儲存體、高達 100 Gbps 的網路頻寬，以及高達 4 TiB (db.x2iden) 或 2 TiB (db.x2idn) 的記憶體。db.x2iezn 類型採用第二代 Intel Xeon 可擴充處理器 (Cascade Lake)，其具有高達 4.5 GHz 的全核心渦輪增壓頻率，以及高達 1.5 TB 的記憶體。

- db.x1 – 執行個體類別，已針對記憶體密集型應用程式進行最佳化處理。在所有資料庫執行個體類別中，這些執行個體類別能為 RAM 的每個 GiB 單位提供最低價格，且其所提供的以 DRAM 為基礎執行個體記憶體，更高達 1,952 GiB。db.x1e 執行個體類別類型提供高達 3,904 GiB 的 DRAM 型執行個體記憶體。

記憶體最佳化 R 系列支援下列執行個體類別類型：

- db.r7g — 由 AWS 重力同 3 處理器提供支援的執行個體類別。這些執行個體類別非常適合用來在 MySQL 和 PostgreSQL 等開放原始碼資料庫中執行記憶體密集型工作負載。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton3 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.r6g — 由 AWS 重力 2 處理器提供支援的執行個體類別。這些執行個體類別非常適合用來在 MySQL 和 PostgreSQL 等開放原始碼資料庫中執行記憶體密集型工作負載。db.r6gd 提供本機 NVMe 型 SSD 區塊層級儲存體，適用於需要高速、低延遲本機儲存的應用程式。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton2 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.r6i - 採用第三代 Intel Xeon 可擴充處理器的執行個體類別。這些執行個體類別通過 SAP 認證，非常適合用於像是 MySQL 和 PostgreSQL 等開放原始碼資料庫中的記憶體密集型工作負載。資料庫 .r6id、資料庫 .r6 英寸和資料庫 .r6 idn 執行個體類別的 memory-to-v CPU 比例為 8:1，以及 1 TiB 的最大記憶體。db.r6id 和 db.r6idn 類別提供高達 7.6 TB 的直接連接 NVMe 型 SSD 儲存體，而 db.r6in 則提供 EBS 專用的儲存體。db.r6idn 和 db.r6in 類別提供高達 200 Gbps 的網路頻寬。
- db.r5b – 執行個體類別，專為輸送量密集型應用程式進行記憶體最佳化。db.r5b 執行個體採用 AWS 硝基系統，可提供高達 60 Gbps 的頻寬和 26 萬 IOPS 的 EBS 效能。這是 EC2 上最快的區塊儲存效能。
- db.r5d – 執行個體類別，針對低延遲、非常高的隨機輸入/輸出效能，以及高循序讀取輸送量進行最佳化。
- db.r4 – 執行個體類別，已針對記憶體密集型應用程式進行最佳化處理。這些執行個體類別可提供網路效能的改善。它們由 AWS Nitro 系統提供動力，該系統結合了專用硬件和輕量級虛擬化管理程序。
- db.r4 - 提供聯網效能超過先前 db.r3 執行個體類別的執行個體類別。

對於適用於 Oracle 資料庫的 RDS 引擎，Amazon RDS 已使用下列排 end-of-life 程啟動 db.r4 資料庫執行個體類別的程序，其中包括升級建議。對於使用 db.r4 執行個體類別的 RDS for Oracle 資料庫執行個體，建議您盡快升級至更高一代的執行個體類別。

動作或建議	日期
您再也不能建立使用 db.r4 資料庫執行個體類別的 RDS 資料庫執行個體。	現在
Amazon RDS 開始將使用 db.r4 資料庫執行個體類別的 RDS for Oracle 資料庫執行個體自動升級至對等的 db.r5 DB 執行個體類別。	2023 年 4 月 17 日

對於適用於 MariaDB 的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 的資料庫引擎，Amazon RDS 已按照下列時間表啟動此資料庫執行個體類別的 end-of-support 程序。對於使用此執行個體類別的所有 RDS 資料庫執行個體，建議您盡快升級至新一代的資料庫執行個體類別。

動作或建議	日期
從這個日期開始，Amazon RDS 開始使用 db.r4 將執行個體自動升級到新一代 db.r5 執行個體類別。不再支援使用 db.m4 執行個體類別建立資料庫執行個體。	2024 年 6 月 1 日
Amazon RDS 終止對數據庫 .r4 的支持。	2024 年 12 月 31 日

- db.r3 – 提供記憶體最佳化的執行個體類別。

對於適用於 MariaDB 的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 的資料庫引擎，Amazon RDS 已使用下列排 end-of-life 程啟動資料庫 .r3 資料庫執行個體類別的程序，其中包括升級建議。對於所有使用 db.r3 資料庫執行個體類別的 RDS 資料庫執行個體，建議您盡快升級至更高一代的資料庫執行個體類別。

動作或建議	日期
您再也不能建立使用 db.r3 資料庫執行個體類別的 RDS 資料庫執行個體。	現在

動作或建議	日期
Amazon RDS 開始將使用 db.r3 資料庫執行個體類別的 RDS 資料庫執行個體自動升級至對等的 db.r5 DB 執行個體類別。	2023 年 2 月 1 日

運算最佳化執行個體類別類型

下列是可用的運算最佳化執行個體類別類型：

- db.c6gd — 執行個體類別，非常適合執行進階運算密集型工作負載。這些執行個體類別採用 AWS Graviton2 處理器，為需要高速、低延遲本機儲存的應用程式提供本機 NVMe 型 SSD 區塊層級儲存。

Note

只有異地同步備份資料庫叢集部署才支援 c6gd 執行個體類別。它們是唯一支援提供執行個體大小的異地同步備份資料庫叢集的 medium 執行個體類別。如需詳細資訊，請參閱 [the section called “多可用區域資料庫叢集部署”](#)。

爆量效能執行個體類別類型

下列爆量效能資料庫執行個體類別類型是可用的：

- db.t4g — 由 ARM 式重力 on2 處理器提供支援的一般用途執行個體類別。AWS 相較於適用於各種爆量一般用途工作負載的先前爆量效能資料庫執行個體類別，這些執行個體類別提供更好的價格/效能比。Amazon RDS db.t4g 執行個體設為無限制模式。這表示它們可以在 24 小時時段內大幅提升並超越基準，但需額外付費。

您可以修改資料庫執行個體，以使用其中一個由 AWS Graviton2 處理器提供支援的資料庫執行個體類別。若要這麼做，請完成與任何其他資料庫執行個體修改相同的步驟。

- db.t2 – 此執行個體類別的基準效能具有一定水準，且使用量可爆量增加，以充分利用整個 CPU。db.t3 執行個體設為無限制模式。與先前的 db.t2 執行個體類別相比，這些執行個體類別能提供更多運算容量。它們是採用 AWS Nitro 系統技術，結合了專用硬體和輕量型 Hypervisor。

- db.t2 – 此執行個體類別的基準效能具有一定水準，且使用量可爆量增加，以充分利用整個 CPU。db.t2 執行個體設定為「無限制」模式。建議您僅將該執行個體類別用在開發、測試伺服器或其他非生產伺服器上。

對於適用於 MariaDB 的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 的資料庫引擎，Amazon RDS 已按照下列時間表啟動此資料庫執行個體類別的 end-of-support 程序。對於使用此執行個體類別的所有 RDS 資料庫執行個體，建議您盡快升級至新一代的資料庫執行個體類別。

動作或建議	日期
從這個日期開始，Amazon RDS 開始使用 db.t2 自動將執行個體升級到新一代 db.t3 執行個體類別。不再支援使用 db.t2 執行個體類別建立資料庫執行個體。	2024 年 6 月 1 日
Amazon RDS 終止對 db.t2 的支持。	2024 年 12 月 31 日

Note

使用 AWS 硝基系統 (db.m5、db.r5、db.t3) 的資料庫執行個體類別會限制在合併的讀取和寫入工作負載上。

如需資料庫執行個體類別硬體規格，請參閱 [的資料庫執行個體類別的硬體規格](#)。

Optimized Reads 執行個體類別類型

下列是可用的 Optimized Reads 執行個體類別類型：

- db.r6gd — 由重力 2 處理器提供支援的執行個體類別。AWS 這些執行個體類別非常適合執行記憶體密集型工作負載，並為需要高速、低延遲本機儲存的應用程式提供本機 NVMe SSD 區塊層級儲存體。
- db.r6id - 採用第三代 Intel Xeon 可擴充處理器的執行個體類別。這些執行個體類別通過 SAP 認證，非常適合用於記憶體密集型工作負載。其提供最高 1TiB 記憶體，以及高達 7.6 TB 的直接連接 NVMe 型 SSD 儲存體。

資料庫執行個體類別的支援資料庫引擎

以下為資料庫執行個體類別的資料庫引擎專屬考量：

Db2

資料庫執行個體類別支援會根據 Db2 的版本和版本而有所不同。如需了解各版本支援的執行個體類型，請參閱 [Amazon RDS 適用於 Db2 執行個體類別](#)。

Microsoft SQL Server

資料庫執行個體類別支援會因 SQL Server 版本而異。如需了解各版本支援的執行個體類型，請參閱 [Microsoft SQL Server 的資料庫執行個體類別支援](#)。

Oracle

資料庫執行個體類別支援會因 Oracle 資料庫版本而異。RDS for Oracle 可支援其他記憶體優化執行個體類別。這些類別的名稱形式為 db.r5 *instance_size*.tpc*threads_per_core*.mem*ratio*。如需了解每個最佳化類別的 vCPU 計數和記憶體配置，請參閱 [支援的 RDS for Oracle 執行個體類別](#)。

RDS Custom

如需有關 RDS Custom 支援的資料庫執行個體類別的資訊，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#) 和 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)。

您可以在下表中找到個 Amazon RDS 資料庫引擎支援的 Amazon RDS 資料庫執行個體類別詳細資料。每個引擎的儲存格都包含下列其中一個值：

是

所有版本的資料庫引擎都支援執行個體類別。

否

資料庫引擎不支援執行個體類別。

###

僅資料庫引擎的特定資料庫版本支援執行個體類別。

Amazon RDS 會定期淘汰主要和次要資料庫引擎版本。並非所有人都 AWS 區域 可能支援舊版引擎。如需目前受支援版本的相關資訊，請參閱個別資料庫引擎的主題：[MariaDB 版本](#)、[Microsoft SQL Server 版本](#)、[MySQL 版本](#)、[Oracle 版本](#)，以及 [PostgreSQL 版本](#)。

主題

- [一般用途執行個體類別的支援資料庫引擎](#)
- [記憶體優化執行個體類別的支援資料庫引擎](#)
- [運算最佳化執行個體類別的支援資料庫引擎](#)
- [高效能執行個體類別支援的資料庫引擎](#)
- [最佳化讀取執行個體類別的支援資料庫](#)

一般用途執行個體類別的支援資料庫引擎

下表顯示一般用途執行個體類別支援的資料庫和資料庫版本。

db.m7g - 採用 AWS Graviton3 處理器的一般用途執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m7g.16xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.12xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.8xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本				更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.4xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.2xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.m7g.large	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本

db.m6g - 採用 AWS Graviton2 處理器的一般用途執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6g.10xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.m6g.12xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.m6g.8xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.m6g.4xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.m6g.2xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.m6g.xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6g.large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

db.m6gd — 由重力通 2 處理器和 SSD 儲存裝置提供支援的一般用途執行個體類別 AWS

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6gd.16xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6g.12xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6gd.8xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6gd.4xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6gd.2xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6gd.large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本
db.m6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 16、15 和 14 個版本；13.7 及更高版本；以及 13.4 版本

db.m6id — 由第三代英特爾至強可擴展處理器和 SSD 存儲提供支持的一般用途執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.3 2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.2 4xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.1 6xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.1 2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.8 xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.4 xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		以及 10.4.25 及更高版本 10.4 版本				
db.m6id.2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6id.large	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

db.m6idn - 採用第三代 Intel Xeon 可擴充處理器、SSD 儲存體和網路最佳化的一般用途執行個體類別

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.32xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.24xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.16xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.12xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.8xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.4xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.2xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.m6idn.large	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

db.m6in — 由第三代 Intel Xeon 可擴充處理器和網路最佳化提供支援的一般用途執行個體類別

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.3xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.24xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.16xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.12xlarge	否	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.8xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.4xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.2xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.large	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.m6in.large	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本

db.m6i — 由第三代 Intel Xeon 可擴充處理器提供支援的一般用途執行個體類別

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.32xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.24xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.16xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.12xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.8xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		以及 10.4.24 及更高版本 10.4 版本				
db.m6i.4xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.2xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m6i.large	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本
db.m5d.xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	Oracle 資料庫 19c	所 PostgreSQL 第 16、15 和 14 個版本；13.4、12.8 和 11.13 及更高版本

db.m5d — 由 Intel Xeon 白金級處理器和 SSD 儲存裝置提供支援的一般用途執行個體類別

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.24xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.16xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.12xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.8xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.4xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		10.4.25 及更高版本 10.4 版本				
db.m5d.2xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.m5d.large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

db.m5 — 一般用途執行個體類別 2.5 GHz 英特爾至強白金處理器

執行個體類別	Db:	Maria	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.24xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本

執行個體類別	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.16xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.12xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.8xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.4xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.2xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.m5.large	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本

db.m4 — 搭載 Intel Xeon 處理器的一般用途執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.16large	否	已棄用	是	已移除	已棄用	已棄用
db.m4.10large	否	已棄用	是	已移除	已棄用	已棄用
db.m4.4xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.m4.2xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.m4.xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.m4.large	否	已棄用	是	已移除	已棄用	已棄用

db.m3 – 一般用途執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.2xlarge	否	否	是	是	已移除	已棄用

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.xlarge	否	否	是	是	已移除	已棄用
db.m3.large	否	否	是	是	已移除	已棄用
db.m3.medium	否	否	是	是	已移除	已棄用

記憶體優化執行個體類別的支援資料庫引擎

下表顯示記憶體最佳化執行個體類別支援的資料庫和資料庫版本。

db.z1d – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.1.xlarge	否	否	是	否	是	否
db.z1d.6.large	否	否	是	否	是	否
db.z1d.3.large	否	否	是	否	是	否
db.z1d.2.large	否	否	是	否	是	否
db.z1d.xlarge	否	否	是	否	是	否
db.z1d.large	否	否	是	否	是	否

db.x2g — 由重力 2 處理器提供支援的記憶體最佳化執行個體類別 AWS

執行個體類別	Dt	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2g.1xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.1xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.8large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.4large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.2large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.x2g.large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

db.x2idn — 採用第三代 Intel Xeon 可擴充處理器的記憶體最佳化執行個體類別

執行個體類別	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2idn.32xlarge	否	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	僅限企業版	PostgreSQL 本、14.6 和 13.9 版
db.x2idn.24xlarge	否	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	僅限企業版	PostgreSQL 本、14.6 和 13.9 版
db.x2idn.16xlarge	否	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	僅限企業版	PostgreSQL 本、14.6 和 13.9 版

db.x2iedn – 具有本機 NVMe 型 SSD 的記憶體最佳化執行個體類別，採用第三代 Intel Xeon 可擴充處理器

執行個體類別	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.32xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	僅限企業版	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .24xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	僅限企業版	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.x2iedn .16xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	僅限企業版	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.x2iedn .8xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	僅限企業版	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.x2iedn .4xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	企業版和標準版 2 (SE2)	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.x2iedn .2xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	企業版和標準版 2 (SE2)	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.xlarge	是	所有 MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	僅限標準版和企業版，SQL Server 2014 12.00 及更新版本	MySQL 8.0.28 及更新版本	企業版和標準版 2 (SE2)	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

db.x2iezn – 採用第二代 Intel Xeon 可擴充處理器的記憶體最佳化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iezn.8xlarge	否	否	否	否	僅限企業版	否
db.x2iezn.6xlarge	否	否	否	否	僅限企業版	否
db.x2iezn.4xlarge	否	否	否	否	企業版和標準版 2 (SE2)	否
db.x2iezn.2xlarge	否	否	否	否	企業版和標準版 2 (SE2)	否

db.x1e – 記憶體最佳化的執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.32xlarge	否	否	是	否	是	否

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.16xlarge	否	否	是	否	是	否
db.x1e.8xlarge	否	否	是	否	是	否
db.x1e.4xlarge	否	否	是	否	是	否
db.x1e.2xlarge	否	否	是	否	是	否
db.x1e.xlarge	否	否	是	否	是	否

db.x1 – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1.32xlarge	否	否	是	否	是	否
db.x1.16xlarge	否	否	是	否	是	否

db.r7g — 由重力同 3 處理器提供支援的記憶體最佳化執行個體類別 AWS

執行個體類別	DI	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.r7g.1xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.r7g.1xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.r7g.8large	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.r7g.4large	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.r7g.2large	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本
db.r7g.xlarge	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		版本，以及 10.4.26 及更高版本 10.4 版本				
db.r7g.large	否	MariaDB 10.11 版本、10.6.10 及更高版本 10.6 版本、10.5.17 及更高版本 10.5 版本，以及 10.4.26 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本，以及 13.4 及更高版本的 13 個版本

db.r6g — 由重力 2 處理器提供支援的記憶體最佳化執行個體類別 AWS

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6g.16xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r6g.12xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r6g.8xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r6g.4xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6g.2xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r6g.xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r6g.large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.23 及更新版本	否	所有 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

db.r6gd — 由重力 2 處理器提供支援的記憶體最佳化執行個體類別 AWS

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.6xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.2xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.4xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.4xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.3xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.1xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

db.r6id - 採用第三代 Intel Xeon 可擴充處理器的記憶體優化執行個體類別

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.32xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Ora	PostgreSQL
db.r6id.24xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.16xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.12xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.8xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.4xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db:	MariaDB	MicroSQL Server	MySQL	Ora	PostgreSQL
db.r6id.2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.large	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

db.r6idn - 採用第三代 Intel Xeon 可擴充處理器的記憶體優化執行個體類別

執行個體類別	Db:	MariaDB	MicroSQL Server	MySQL	Ora	PostgreSQL
db.r6idn.32xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6idn.24xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及	否	MySQL 8.0.28 版	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		10.4.25 及更高版本 10.4 版本		及更新版本		
db.r6idn.16xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6idn.12xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6idn.8xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6idn.4xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6idn.2xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Ora	PostgreSQL
db.r6idn.xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

db.r6in - 採用第三代 Intel Xeon 可擴充處理器的記憶體優化執行個體類別

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Ora	PostgreSQL
db.r6in.3 2xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.r6in.2 4xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本
db.r6in.1 6xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個版本

執行個體類別	Db:	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.1 2xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本
db.r6in.8 xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本
db.r6in.4 xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本
db.r6in.2 xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本

執行個體類別	Db:	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.xlarge	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本
db.r6in.large	是	MariaDB 10.6.8 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所有 PostgreSQL 為 14.3 及更高版本，以及 14 個以上版本、13.7 及更高版本的 13 個版本、12.11 及更高版本 12 個版本，以及 11.16 及更高版本 11 個以上版本

db.r6i — 記憶體最佳化執行個體類別，預先設定為高記憶體、儲存和 I/O

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
數據庫 .r6i.8 倍大.	否	否	否	否	是	否
資料庫 .r6. 8 倍大.	否	否	否	否	是	否
資料庫 .r6. 6 倍大.	否	否	否	否	是	否
資料庫 .r6 .4 X 大型 .tpc2.4	否	否	否	否	是	否

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
資料庫 .r6 .4 X 大.	否	否	否	否	是	否
資料庫 .r6 .4 X 大型 .tpc2.2	否	否	否	否	是	否
資料庫 .r6i.2 大型.	否	否	否	否	是	否
資料庫 .r6i.2 大型 .tpc2.4	否	否	否	否	是	否
資料庫 .r6i.2 大型 .tpc1.2	否	否	否	否	是	否
資料庫 .r6. 大型 .tpc2.4	否	否	否	否	是	否
資料庫 .r6. 大型 .tpc2.2	否	否	否	否	是	否
大型 .tpc1.2.	否	否	否	否	是	否

db.r6i – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.3: xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個

執行個體類別	引擎	支援的引擎版本	MySQL	Oracle	PostgreSQL	
					版本，以及 10.21 及更高版本 10 個版本	
db.r6i.2xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.10xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.12xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.8xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本

執行個體類別	DB 類別	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.4large	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.2large	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.xlarge	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本
db.r6i.large	是	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.15 及更高版本 10.5 版本，以及 10.4.24 及更高版本 10.4 版本	是	MySQL 8.0.28 版及更新版本	是	所有 PostgreSQL、13.4 及更高版本、12.8 及更高版本 12 個版本、11.13 及更高版本、11.13 及更高版本 11 個版本，以及 10.21 及更高版本 10 個版本

db.r5d – 記憶體優化執行個體類別

執行個體類別	Direct Connect	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.2xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.1xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.1xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.8large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.4large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.2large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
				新版本		
db.r5d.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r5d.large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	是	MySQL 8.0.28 及更新版本	是	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

db.r5b – 記憶體最佳化執行個體類別，預先設定高記憶體、儲存和輸入/輸出

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.8xlarge.tpc2.mem3x	否	否	否	否	是	否
db.r5b.6xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5b.4xlarge.tpc2.mem4x	否	否	否	否	是	否

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.4xlarge.tpc2.mem3x	否	否	否	否	是	否
db.r5b.4xlarge.tpc2.mem2x	否	否	否	否	是	否
db.r5b.2xlarge.tpc2.mem8x	否	否	否	否	是	否
db.r5b.2xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5b.2xlarge.tpc1.mem2x	否	否	否	否	是	否
db.r5b.xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5b.xlarge.tpc2.mem2x	否	否	否	否	是	否
db.r5b.large.tpc1.mem2x	否	否	否	否	是	否

db.r5b – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.24xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.16xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.12xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.8xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	> 是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.4xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		10.4 版本，以及 10.3.34 及更高版本				
db.r5b.2xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.xlarge	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.r5b.large	否	MariaDB 10.11 版本、10.6.5 及更高版本及更高版本、10.5.12 及更高版本 10.5 版本、10.4.24 及更高版本 10.4 版本，以及 10.3.34 及更高版本	是	MySQL 8.0.25 及更新版本	是	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

db.r5 – 記憶體優化執行個體類別，預先設定高記憶體、儲存和輸入/輸出

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.12xlarge.tpc2.mem2x	否	否	否	否	是	否

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.8xlarge.tpc2.mem3x	否	否	否	否	是	否
db.r5.6xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5.4xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5.4xlarge.tpc2.mem3x	否	否	否	否	是	否
db.r5.4xlarge.tpc2.mem2x	否	否	否	否	是	否
db.r5.2xlarge.tpc2.mem8x	否	否	否	否	是	否
db.r5.2xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5.2xlarge.tpc1.mem2x	否	否	否	否	是	否
db.r5.xlarge.tpc2.mem4x	否	否	否	否	是	否
db.r5.xlarge.tpc2.mem2x	否	否	否	否	是	否
db.r5.large.tpc1.mem2x	否	否	否	否	是	否

db.r5 – 記憶體優化執行個體類別

執行個體類別	Db2	MariaD	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r5.24xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.16xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.12xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.8xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.4xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.2xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.xlarge	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本
db.r5.large	否	是	是	是	是	所 PostgreSQL、15、14、13、12 和 11 版本；10.17 及更高版本 10 版本；以及 9.6.22 及更高版本

db.r4 – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.16xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.r4.8xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.r4.4xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.r4.2xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.r4.xlarge	否	已棄用	是	已移除	已棄用	已棄用
db.r4.large	否	已棄用	是	已移除	已棄用	已棄用

db.r3 – 記憶體優化執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r3.8xlarge**	否	所有 MariaDB 10.6、10.4 和 10.3 版本	是	是	已移除	已棄用
db.r3.4xlarge	否	所有 MariaDB 10.6、10.4 和 10.3 版本	是	是	已移除	已棄用
db.r3.2xlarge	否	所有 MariaDB 10.6、10.4 和 10.3 版本	是	是	已移除	已棄用
db.r3.xlarge	否	所有 MariaDB 10.6、10.4 和 10.3 版本	是	是	已移除	已棄用

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r3.large	否	所有 MariaDB 10.6、10.4 和 10.3 版本	是	是	已移除	已棄用

運算最佳化執行個體類別的支援資料庫引擎

下表顯示運算最佳化執行個體類別所支援的資料庫和資料庫版本。

db.c6gd — 運算最佳化執行個體類別 (僅適用於異地同步備份資料庫叢集部署)

執行個體類別	Db2	Maria	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
分貝 .C6 公分. 16 大	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
分貝 .c6 公分. 12 倍大	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
分貝 .c6 公分. 8 倍大	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
分貝 .c6 公升. 4 大	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
DB.C6 公克. 2 大	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本

執行個體類別	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
分貝 .c6 公克.	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
大包 .c6 公克	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本
分貝 .c6 公克. 中	否	否	否	MySQL 8.0.28 及更新版本	否	所有版本 16 和 15 版本；14.5 及更高版本 14 個版本；13.4 和 13.7 及更高版本 13 個版本

高效能執行個體類別支援的資料庫引擎

下表顯示大量效能執行個體類別支援的資料庫和資料庫版本。

db.t4g — 由重力 2 處理器提供支援的高效能執行個體類別 AWS

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
--------	-----	---------	----------------------	-------	------	------------

db.t4g — 由重力 2 處理器提供支援的高效能執行個體類別 AWS

db.t4g.2x large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.t4g.xlarge	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
				新 版 本		
db.t4g.large	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.t4g.medium	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.t4g.small	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本
db.t4g.micro	否	所有 MariaDB 10.11、10.6、10.5 和 10.4 版本	否	MySQL 8.0.25 及更新版本	否	所 PostgreSQL 16、15、14 和 13 版本；以及 12.7 及更高版本

db.t3 – 爆量效能執行個體類別

執行個體類別	Db2	Maria	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t3.2xlarge	是	是	是	是	是	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本
db.t3.xlarge	是	是	是	是	是	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本
db.t3.large	是	是	是	是	是	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本
db.t3.medium	是	是	是	是	是	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本
db.t3.small	是	是	是	是	是	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本
db.t3.micro	否	是	否	是	僅在 Oracle 資料庫 12c 版本 1 (12.1.0.2) 上，此版本已被淘汰	所 PostgreSQL 16、15、14、13、12、11 和 10 版本；以及 9.6.22 及更高版本

db.t2 – 爆量效能執行個體類別

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t2.2xlarge	否	已棄用	否	已棄用	已棄用	已棄用
db.t2.xlarge	否	已棄用	否	已棄用	已棄用	已棄用
db.t2.large	否	已棄用	是	已移除	已棄用	已棄用
db.t2.medium	否	已棄用	是	已移除	已棄用	已棄用
db.t2.small	否	已棄用	是	已移除	已棄用	已棄用
db.t2.micro	否	已棄用	是	已移除	已棄用	已棄用

最佳化讀取執行個體類別的支援資料庫

下表顯示「最佳化讀取」執行處理類別支援的資料庫和資料庫版本。

db.r6gd — 記憶體最佳化執行個體類別，支援最佳化讀取，並由 Graviton2 處理器提供支援 AWS

執行個體類別	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.6xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

執行個體類別	Direct Connect	Database Engine	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.2xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.xlarge	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4
db.r6gd.large	否	MariaDB 10.11 版本、10.6.7 及更高版本 10.6 版本、10.5.1 6 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 及更新版本	否	所有 PostgreSQL 15 版本、14.5 及更高版本 14 個版本、13.7 及更高版本 13 個版本，以及 13.4

db.r6id — 記憶體最佳化執行個體類別，支援最佳化讀取，並由第三代 Intel Xeon 可擴充處理器提供支援

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Ora	PostgreSQL
db.r6id.3 2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.2 4xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.1 6xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.1 2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.8 xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.4 xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

執行個體類別	Db	MariaDB	Microsoft SQL Server	MySQL	Ora	PostgreSQL
		10.4.25 及更高版本 10.4 版本				
db.r6id.2xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.large	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本
db.r6id.xlarge	否	MariaDB 10.6.10 及更高版本 10.6 版本、10.5.16 及更高版本 10.5 版本，以及 10.4.25 及更高版本 10.4 版本	否	MySQL 8.0.28 版及更新版本	否	所 PostgreSQL 為 14.5 及更高版本，以及 13.7 及更高版本 13 個版本

確定資料庫執行個體類別支援 AWS 區域

若要判定特定 AWS 區域中每個資料庫引擎支援的資料庫執行個體類別，您可以採用數種方法之一。您可以針對 [AWS Management Console\(\)](#) 使用 [Amazon RDS 定價頁面](#) 或 [說明可排序的 DB 執行個體選項命令](#)。AWS Command Line Interface AWS CLI

Note

當您使用執行作業時 AWS Management Console，它會自動顯示特定資料庫引擎、資料庫引擎版本和 AWS 區域。您可以執行的操作範例包括建立和修改資料庫執行個體。

內容

- [使用 Amazon RDS 定價頁面確定資料庫執行個體類別支援 AWS 區域](#)
- [使用 AWS CLI 來判斷資料庫執行個體類別支援 AWS 區域](#)
 - [列出 AWS 區域中特定資料庫引擎版本支援的資料庫執行個體類別](#)
 - [列出 AWS 區域中支援特定資料庫執行個體類別的資料庫引擎版本](#)

使用 Amazon RDS 定價頁面確定資料庫執行個體類別支援 AWS 區域

您可以使用 [Amazon RDS 定價](#) 頁面來確定特定 AWS 區域中每個資料庫引擎支援的資料庫執行個體類別。

使用定價頁面確定區域中每個引擎支援的資料庫執行個體類別

1. 轉到 [Amazon RDS 定價](#)。
2. 在適用於 Amazon RDS 的 AWS 定價計算器區段中，選擇立即建立自訂估計值。
3. 在選擇區域中，選擇 AWS 區域。
4. 在尋找服務中，輸入 **Amazon RDS**。
5. 針對您的組態選項和資料庫引擎選擇設定。
6. 您可以使用相容執行個體區段來查看支援的資料庫執行個體類別。
7. (選擇性) 在計算器中選擇其他選項，然後選擇儲存並檢視摘要或儲存並新增服務。

使用 AWS CLI 來判斷資料庫執行個體類別支援 AWS 區域

您可以使用 AWS CLI 來判斷特定資料庫引擎和資料庫引擎版本支援的資料庫執行個體類別 AWS 區域。下表顯示了有效的資料庫引擎值。

引擎名稱	CLI 命令中的引擎值	如需版本的詳細資訊
Db2	db2-ae	Amazon RDS 版本上的 Db2
	db2-se	
MariaDB	mariadb	Amazon RDS 上的 MariaDB 版本
Microsoft SQL Server	sqlserver-ee	Amazon RDS 上的 Microsoft SQL Server 版本
	sqlserver-se	

引擎名稱	CLI 命令中的引擎值	如需版本的詳細資訊
	sqlserver-ex	
	sqlserver-web	
MySQL	mysql	MySQL on Amazon RDS 版本
Oracle	oracle-ee	Amazon RDS for Oracle 版本備註
	oracle-se2	
PostgreSQL	postgres	可用的 PostgreSQL 資料庫版本

如需有關 AWS 區域 名稱的資訊，請參閱[AWS 地區](#)。

下列範例示範如何 AWS 區域 使用可[描述可排序的 db-Instance 執行個體選項](#)命 AWS CLI 令來判斷資料庫執行個體類別支援。

Note

若要限制輸出，這些範例只會顯示一般用途 SSD (gp2) 儲存類型的結果。如有必要，您可以在命令中將儲存類型變更為一般用途 SSD (gp3)、佈建 IOPS (io1) 或磁性 (standard)。

主題

- [列出 AWS 區域中特定資料庫引擎版本支援的資料庫執行個體類別](#)
- [列出 AWS 區域中支援特定資料庫執行個體類別的資料庫引擎版本](#)

列出 AWS 區域中特定資料庫引擎版本支援的資料庫執行個體類別

若要在中列出特定資料庫引擎版本支援的資料庫執行個體類別 AWS 區域，請執行下列命令。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version \
  \
  --query "*[].[DBInstanceClass:DBInstanceClass,StorageType:StorageType]|[?
StorageType=='gp2']|[].[DBInstanceClass:DBInstanceClass]" \
```



```
--output text \  
--region region
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version  
^  
--query ".*[].[DBInstanceClass:DBInstanceClass,StorageType:StorageType]|[?  
StorageType=='gp2']|[].[DBInstanceClass:DBInstanceClass]" ^  
--output text ^  
--region region
```

例如，下列命令列出美國東部 (維吉尼亞北部) 中 RDS for PostgreSQL 資料庫引擎 13.6 版支援的資料庫執行個體類別。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4  
\  
--query ".*[].[DBInstanceClass:DBInstanceClass,StorageType:StorageType]|[?  
StorageType=='gp2']|[].[DBInstanceClass:DBInstanceClass]" \  
--output text \  
--region us-east-1
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4  
^  
--query ".*[].[DBInstanceClass:DBInstanceClass,StorageType:StorageType]|[?  
StorageType=='gp2']|[].[DBInstanceClass:DBInstanceClass]" ^  
--output text ^  
--region us-east-1
```

列出 AWS 區域中支援特定資料庫執行個體類別的資料庫引擎版本

若要列出 AWS 區域中支援特定資料庫執行個體類別的資料庫引擎版本，請執行下列命令。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-  
class DB_instance_class \  

```

```
--query "*"[].{EngineVersion:EngineVersion,StorageType:StorageType}][?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
--output text \
--region region
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class ^
--query "*"[].{EngineVersion:EngineVersion,StorageType:StorageType}][?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
--output text ^
--region region
```

例如，下列命令會列出支援 US East (N. Virginia) 中 db.r5.large 資料庫執行個體類別之 RDS for PostgreSQL 資料庫引擎的資料庫引擎版本。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
db.m7g.large \
--query "*"[].{EngineVersion:EngineVersion,StorageType:StorageType}][?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
--output text \
--region us-east-1
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
db.m7g.large ^
--query "*"[].{EngineVersion:EngineVersion,StorageType:StorageType}][?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
--output text ^
--region us-east-1
```

變更資料庫執行個體類別

您能夠變更資料庫執行個體類別，藉此改變資料庫執行個體可使用的 CPU 及記憶體。若要變更資料庫執行個體類別，請遵循[修改 Amazon RDS 資料庫執行個體](#)的相關指示，以修改資料庫執行個體。

針對 RDS for Oracle 中的資料庫執行個體類別設定處理器

Amazon RDS 資料庫執行個體支援 Intel 超執行緒技術，所以同一個 Intel Xeon CPU 核心上，能夠同時執行多重執行緒。每個執行緒皆代表資料庫執行個體上的一個虛擬 CPU (vCPU)。資料庫執行個體具有預設的 CPU 核心數量，且該數量會因資料庫執行個體類別而異。例如，在預設情況下，db.m4.xlarge 資料庫執行個體類別有兩個 CPU 核心，而每個核心有兩個執行緒，因此該類型總共有四個 vCPU。

Note

每個 vCPU 均為 Intel Xeon CPU 核心的超執行緒。

主題

- [適用於 RDS for Oracle 的處理器組態概觀](#)
- [支援處理器組態的資料庫執行個體類別](#)
- [設定資料庫執行個體類別的 CPU 核心和每 CPU 核心執行緒數量](#)

適用於 RDS for Oracle 的處理器組態概觀

使用 RDS for Oracle 時，您通常可以找到一個資料庫執行個體類別，其具有記憶體與數個 vCPU 的組合，以符合您的工作負載。不過，您也可以指定下列處理器功能，針對特定工作負載或業務需求最佳化 RDS for Oracle 資料庫執行個體：

- CPU 核心數量 – 您能夠自訂資料庫執行個體的 CPU 核心數量。執行此操作後，資料庫執行個體即可降低 CPU 核心數量，但會具備足夠的 RAM 數量，以執行記憶體密集型工作負載。如此一來，就有可能將軟體的授權成本最佳化。
- 每核心執行緒數量 – 您能夠為每個 CPU 核心指定單一執行緒，進而停用 Intel 超執行緒技術。此操作適用於特定的工作負載，例如高效能運算 (HPC) 的工作負載。

您可以分別控制 CPU 核心數量和每核心的執行緒數量。您能夠提出請求，以設定其中一個項目，或同時設定兩者。當您將該設定與資料庫執行個體建立關聯後，系統會一直保留此設定，直到進行變更為止。

資料庫執行個體的處理器設定與資料庫執行個體快照相關聯。因此，當系統還原快照時，還原資料庫執行個體所使用的處理器功能設定，將與拍攝快照時所用的設定相同。

如果您在修改資料庫執行個體的資料庫執行個體類別時，是使用非預設的處理器設定，請指定預設處理器設定，或是在修改時明確指定處理器設定。這項要求能確保您了解在修改資料庫執行個體的期間，可能會產生第三方授權成本。

當您在 RDS for Oracle 資料庫執行個體上指定處理器功能時，費用不會因而有所增減。系統會以使用預設 CPU 組態來啟動資料庫執行個體的計費方式，向您收取相同費用。

支援處理器組態的資料庫執行個體類別

您只能在符合下列條件時，設定每個核心的 CPU 核心和執行緒數目：

- 您正在設定 RDS for Oracle 資料庫執行個體。如需不同 Oracle 資料庫版本所支援的資料庫執行個體類別相關資訊，請參閱 [RDS for Oracle 執行個體類別](#)。
- 您的資料庫執行個體正在使用 RDS for Oracle 的自帶授權 (BYOL) 授權選項。如需 Oracle 授權選項的詳細資訊，請參閱 [RDS for Oracle 授權選項](#)。
- 您的資料庫執行個體不屬於具有預先定義處理器組態的 db.r5 或 db.r5b 執行個體類別之一。這些執行個體類別的名稱格式為 db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio* 或 db.r5b.*instance_size*.tpc*threads_per_core*.mem*ratio*。例如，db.r5b.xlarge.tpc2.mem4x 每個核心預先配置 2 個執行緒 (tpc2) 以及與標準 db.r5b.xlarge 執行個體類別一樣多的 4 倍記憶體。您無法設定這些最佳化執行個體類別的處理器功能。如需更多詳細資訊，請參閱 [支援的 RDS for Oracle 執行個體類別](#)。

下表會針對各個資料庫執行個體類別，列出該類別支援的 CPU 核心與每核心執行緒數量相關設定。此外，您還能夠查看各個資料庫執行個體類別，進而檢視其 CPU 核心與每核心執行緒數量的預設值和有效值。

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心數量	有效的每核心執行緒數量
db.m6i - 記憶體優化執行個體類別					
db.m6i.large	2	1	2	1	1、2
db.m6i.xlarge	4	2	2	2	1、2
db.m6i.2xlarge	8	4	2	2、4	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.m6i.4xlarge	16	8	2	2、4、6、8	1、2
db.m6i.4xlarge	16	8	2	2、4、6、8	1、2
db.m6i.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2
db.m6i.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.m6i.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.m6i.24xlarge	96	48	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32、3 4、36、38、4 0、42、44、4 6、48	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.m6i.32xlarge	128	64	2	二、四、 六、八、十 、十、十、 十四、十 六、十六、 四、四、 四、四、 四、四、四 、四、四、 四、四、 四、四、 四、四十六 、四十、 四、五十五 、五十五、 五十五、 五十六、六 十六、六十 六、64	1、2
db.m5 – 一般用途執行個體類別					
db.m5.large	2	1	2	1	1、2
db.m5.xlarge	4	2	2	2	1、2
db.m5.2xlarge	8	4	2	2、4	1、2
db.m5.4xlarge	16	8	2	2、4、6、8	1、2
db.m5.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.m5.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.m5.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.m5.24xlarge	96	48	2	4、6、8、10、 12、14、16、 18、20、22、 24、26、28、 30、32、34、 36、38、40、 42、44、46、 48	1、2

db.m5d – 一般用途執行個體類別

db.m5d.large	2	1	2	1	1、2
db.m5d.xlarge	4	2	2	2	1、2
db.m5d.2xlarge	8	4	2	2、4	1、2
db.m5d.4xlarge	16	8	2	2、4、6、8	1、2
db.m5d.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.m5d.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.m5d.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.m5d.24xlarge	96	48	2	4、6、8、10、 12、14、16、 18、20、22、 24、26、28、 30、32、34、 36、38、40、 42、44、46、 48	1、2
db.m4 – 一般用途執行個體類別					
db.m4.10xlarge	40	20	2	2、4、6、8、 10、12、14、1 6、18、20	1、2
db.m4.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r6i – 記憶體優化執行個體類別					

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r6i.large	2	1	2	1	1、2
db.r6i.xlarge	4	2	2	1、2	1、2
db.r6i.2xlarge	8	4	2	2、4	1、2
db.r6i.4xlarge	16	8	2	2、4、6、8	1、2
db.r6i.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2
db.r6i.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.r6i.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r6i.24xlarge	96	48	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32、3 4、36、38、4 0、42、44、4 6、48	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r6i.32xlarge	128	64	2	二、四、 六、八、十 、十、十、 十四、十 六、十六、 四、四、 四、四、 四、四、四 、四、四、 四、四、 四、四、 四、四十六 、四十、 四、五十五 、五十五、 五十五、 五十六、六 十六、六十 六、64	1、2

db.r5 – 記憶體優化執行個體類別

db.r5.large	2	1	2	1	1、2
db.r5.xlarge	4	2	2	2	1、2
db.r5.2xlarge	8	4	2	2、4	1、2
db.r5.4xlarge	16	8	2	2、4、6、8	1、2
db.r5.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r5.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.r5.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r5.24xlarge	96	48	2	4、6、8、10、 12、14、16、 18、20、22、 24、26、28、 30、32、34、 36、38、40、 42、44、46、 48	1、2
db.r5 – 記憶體優化執行個體類別					
db.r5b.large	2	1	2	1	1、2
db.r5b.xlarge	4	2	2	2	1、2
db.r5b.2xlarge	8	4	2	2、4	1、2
db.r5b.4xlarge	16	8	2	2、4、6、8	1、2
db.r5b.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r5b.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.r5b.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r5b.24xlarge	96	48	2	4、6、8、10、 12、14、16、 18、20、22、 24、26、28、 30、32、34、 36、38、40、 42、44、46、 48	1、2

db.r5d – 記憶體優化執行個體類別

db.r5d.large	2	1	2	1	1、2
db.r5d.xlarge	4	2	2	2	1、2
db.r5d.2xlarge	8	4	2	2、4	1、2
db.r5d.4xlarge	16	8	2	2、4、6、8	1、2
db.r5d.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r5d.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2
db.r5d.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r5d.24xlarge	96	48	2	4、6、8、10、 12、14、16、 18、20、22、 24、26、28、 30、32、34、 36、38、40、 42、44、46、 48	1、2
db.r4 – 記憶體優化執行個體類別					
db.r4.large	2	1	2	1	1、2
db.r4.xlarge	4	2	2	1、2	1、2
db.r4.2xlarge	8	4	2	1、2、3、4	1、2
db.r4.4xlarge	16	8	2	1、2、3、4、 、6、7、8	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.r4.8xlarge	32	16	2	1、2、3、4、 6、7、8、9 10、11、12、 13、14、15、 16	1、2
db.r4.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.r3 – 記憶體優化執行個體類別					
db.r3.large	2	1	2	1	1、2
db.r3.xlarge	4	2	2	1、2	1、2
db.r3.2xlarge	8	4	2	1、2、3、4	1、2
db.r3.4xlarge	16	8	2	1、2、3、4、 6、7、8	1、2
db.r3.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2
db.x2idn – 記憶體最佳化執行個體類別					
db.x2idn.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.x2idn.24xlarge	96	48	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32、3 4、36、38、4 0、42、44、4 6、48	1、2
db.x2idn.32xlarge	128	64	2	二、四、 六、八、十 、十、十、 十四、十 六、十六、 四、四、 四、四、 四、四、四 、四、四、 四、四、 四、四、 四、四十六 、四十、 四、五十五 、五十五、 五十五、 五十六、六 十六、六十 六、64	1、2
db.x2iedn – 記憶體最佳化執行個體類別					
db.x2iedn.xlarge	4	2	2	1、2	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.x2iedn.2xlarge	8	4	2	2、4	1、2
db.x2iedn.4xlarge	16	8	2	2、4、6、8	1、2
db.x2iedn.8xlarge	32	16	2	2、4、6、8、 10、12、14、1 6	1、2
db.x2iedn.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.x2iedn.24xlarge	96	48	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32、3 4、36、38、4 0、42、44、4 6、48	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.x2iedn.32xlarge	128	64	2	二、四、 六、八、十 、十、十、 十四、十 六、十六、 四、四、 四、四、 四、四、四 、四、四、 四、四、 四、四、 四、四十六 、四十、 四、五十五 、五十五、 五十五、 五十六、六 十六、六十 六、64	1、2

db.x2iezn – 記憶體最佳化執行個體類別

db.x2iezn.2xlarge	8	4	2	2、4	1、2
db.x2iezn.4xlarge	16	8	2	2、4、6、8	1、2
db.x2iezn.6xlarge	24	12	2	2、4、6、8、 0、12	1、2
db.x2iezn.8xlarge	32	16	2	2、4、6、8、 0、12、14、1 6	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.x2iezn.12xlarge	48	24	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24	1、2

db.x1 – 記憶體優化執行個體類別

db.x1.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.x1.32xlarge	128	64	2	4、8、12、16 、20、24、28 、32、36、40 、44、48、52 、56、60、64	1、2

db.x1e – 記憶體最佳化的執行個體類別

db.x1e.xlarge	4	2	2	1、2	1、2
db.x1e.2xlarge	8	4	2	1、2、3、4	1、2
db.x1e.4xlarge	16	8	2	1、2、3、4、 、6、7、8	1、2
db.x1e.8xlarge	32	16	2	1、2、3、4、 、6、7、8、9 10、11、12、 13、14、15、 16	1、2

DB instance class (資料庫執行個體類別)	預設 vCPU	預設 CPU 核心	預設每核心 執行緒	有效的 CPU 核心數量	有效的每核 心執行緒數 量
db.x1e.16xlarge	64	32	2	2、4、6、8、 10、12、14、1 6、18、20、2 2、24、26、2 8、30、32	1、2
db.x1e.32xlarge	128	64	2	4、8、12、16 、20、24、28 、32、36、40 、44、48、52 、56、60、64	1、2
db.z1d – 記憶體優化執行個體類別					
db.z1d.large	2	1	2	1	1、2
db.z1d.xlarge	4	2	2	2	1、2
db.z1d.2xlarge	8	4	2	2、4	1、2
db.z1d.3xlarge	12	6	2	2、4、6	1、2
db.z1d.6xlarge	24	12	2	2、4、6、8、 10、12	1、2
db.z1d.12xlarge	48	24	2	4、6、8、10、 12、14、16、 18、20、22、 24	1、2

Note

您可以用 AWS CloudTrail 來監控和稽核適用於 Oracle 資料庫執行個體的 Amazon RDS 程序組態的變更。如需使用的更多資訊 CloudTrail，請參閱 [在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)。

設定資料庫執行個體類別的 CPU 核心和每 CPU 核心執行緒數量

當您執行下列操作時，可針對資料庫執行個體類別設定其 CPU 核心數量，以及每核心的執行緒數量：

- [建立 Amazon RDS 資料庫執行個體](#)
- [修改 Amazon RDS 資料庫執行個體](#)
- [從資料庫 快照還原](#)
- [將資料庫執行個體還原至指定的時間](#)

Note


在您修改資料庫執行個體，以便設定 CPU 核心數量與每核心執行緒數量時，會導致資料庫執行個體短暫停機。

您可以使用 AWS Management Console、或 RDS API，針對資料庫執行個體類別設定 CPU 核心和每個 CPU 核心的執行緒。AWS CLI

主控台

當您建立、修改或還原資料庫執行個體時，即可在 AWS Management Console 中設定資料庫執行個體類別。Instance specifications (執行個體規格) 區段會顯示處理器的可用選項。另外，下圖將呈現處理器功能選項。

Instance specifications

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#) 

DB engine

Oracle Database Enterprise Edition

License model [Info](#)

bring-your-own-license ▼

DB engine version [Info](#)

Oracle 12.1.0.2.v12 ▼

DB instance class [Info](#)

db.r4.xlarge — 4 vCPU, 30.5 GiB RAM ▼

Multi-AZ deployment [Info](#)

Create replica in different zone

Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

No

Storage type [Info](#)

Provisioned IOPS (SSD) ▼

Allocated storage

100



GiB

(Minimum: 100 GiB, Maximum: 16384 GiB)

Provisioned IOPS [Info](#)

1000



▼ Additional configuration

Processor features

Override default values

You can change the number of CPU cores and threads per core on the DB instance class.

Core count [Info](#)

2 ▼

Threads per core [Info](#)

2 ▼

Estimated monthly costs

在 Processor features (處理器功能) 底下，將下列資料庫執行個體類別選項設為適當的值：

- Core count (核心計數) – 使用此選項來設定 CPU 核心數量。該值必須等於或小於資料庫執行個體類別的 CPU 核心數量上限。
- Threads per core (每核心執行緒數量) – 可將此選項指定為 2，以在每核心上啟用多重執行緒。若要在每核心上停用多重執行緒，則可指定為 1。

在修改或還原資料庫執行個體時，您也可以針對執行個體類別將 CPU 核心與每核心的執行緒數量設為預設值。

當您在主控台檢視資料庫執行個體的詳細資訊時，您可以在 Configuration (組態) 標籤上檢視其資料庫執行個體類別的處理器資訊。下圖所顯示的資料庫執行個體類別擁有一個 CPU 核心，且已在每核心上啟用多重執行緒。

Instance and IOPS	
Instance Class	db.r4.large
Core count	1
Threads per core	2
vCPU enabled	2
Storage Type	Provisioned IOPS (SSD)
IOPS	1000
Storage	100 GiB

對於 Oracle 資料庫執行個體，唯有使用自有授權 (BYOL) 的資料庫執行個體，才會顯示處理器資訊。

AWS CLI

執行下列其中一個 AWS CLI 命令時，您可以藉機設定資料庫執行個體的處理器功能：

- [create-db-instance](#)
- [modify-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

若要使用設定資料庫執行個體之資料庫執行個體類別的處理器 AWS CLI，請在命令中加入 `--processor-features` 選項。您可以使用 `coreCount` 功能名稱來指定 CPU 核心數量，並利用 `threadsPerCore` 功能名稱指定每核心是否要啟用多重執行緒。

此選項的語法如下。

```
--processor-features "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

以下是設定處理器的範例：

範例

- [設定資料庫執行個體的 CPU 核心數量](#)
- [設定資料庫執行個體的 CPU 核心數量並停用多重執行緒](#)
- [檢視資料庫執行個體類別的處理器有效值](#)
- [傳回資料庫執行個體的預設處理器設定](#)
- [傳回資料庫執行個體的預設 CPU 核心數量](#)
- [傳回資料庫執行個體的預設每核心執行緒數量](#)

設定資料庫執行個體的 CPU 核心數量

Example

以下範例會將 CPU 核心數量設定為 4 個，藉此修改 `mydbinstance`。透過 `--apply-immediately`，即可立即套用變更。如果要在下個排定的維護時段再套用變更，請省略 `--apply-immediately` 選項。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --processor-features "Name=coreCount,Value=4" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --processor-features "Name=coreCount,Value=4" ^  
  --apply-immediately
```

設定資料庫執行個體的 CPU 核心數量並停用多重執行緒

Example

以下範例會將 CPU 核心數量設定為 *mydbinstance* 個，並停用每核心上的多重執行緒，藉此修改 4。透過 `--apply-immediately`，即可立即套用變更。如果要在下個排定的維護時段再套用變更，請省略 `--apply-immediately` 選項。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" ^  
  --apply-immediately
```

檢視資料庫執行個體類別的處理器有效值

Example

您能夠執行 [describe-orderable-db-instance-options](#) 命令，並為 `--db-instance-class` 選項指定執行個體類別，以便檢視特定資料庫執行個體類別的處理器有效值。例如，以下命令的輸出結果會顯示適用於 `db.r3.large` 執行個體類別的處理器選項。


```
aws rds describe-orderable-db-instance-options --engine oracle-ee --db-instance-class db.r3.large
```

以下為命令的範例輸出 (JSON 格式)。

```
{
  "SupportsIops": true,
  "MaxIopsPerGib": 50.0,
  "LicenseModel": "bring-your-own-license",
  "DBInstanceClass": "db.r3.large",
  "SupportsIAMDatabaseAuthentication": false,
  "MinStorageSize": 100,
  "AvailabilityZones": [
    {
      "Name": "us-west-2a"
    },
    {
      "Name": "us-west-2b"
    },
    {
      "Name": "us-west-2c"
    }
  ],
  "EngineVersion": "12.1.0.2.v2",
  "MaxStorageSize": 32768,
  "MinIopsPerGib": 1.0,
  "MaxIopsPerDbInstance": 40000,
  "ReadReplicaCapable": false,
  "AvailableProcessorFeatures": [
    {
      "Name": "coreCount",
      "DefaultValue": "1",
      "AllowedValues": "1"
    },
    {
      "Name": "threadsPerCore",
      "DefaultValue": "2",
      "AllowedValues": "1,2"
    }
  ],
  "SupportsEnhancedMonitoring": true,
  "SupportsPerformanceInsights": false,
  "MinIopsPerDbInstance": 1000,
}
```

```
    "StorageType": "io1",
    "Vpc": false,
    "SupportsStorageEncryption": true,
    "Engine": "oracle-ee",
    "MultiAZCapable": true
}
```

此外，您可以執行下列命令，以取得資料庫執行個體類別的處理器資訊：

- [describe-db-instances](#) – 顯示指定的資料庫執行個體的處理器資訊。
- [describe-db-snapshots](#) – 顯示指定的資料庫快照的處理器資訊。
- [describe-valid-db-instance-modifications](#) – 顯示指定的資料庫執行個體有效的處理器修改內容。

在上述命令的輸出中，只有在符合下列條件時，處理器功能的值才不是 null：

- 您正在使用 RDS for Oracle 資料庫執行個體。
- 您的 RDS for Oracle 資料庫執行個體支援變更處理器值。
- 目前的 CPU 核心和執行緒設定設定為非預設值。

如果不符合上述條件，您可以使用 [describe-db-instances](#) 執行個體獲取執行個體類型。您可以透過執行 EC2 操作 [describe-instance-types](#) 的處理器資訊。

傳回資料庫執行個體的預設處理器設定

Example

以下範例會針對資料庫執行個體，將其資料庫執行個體類別傳回至預設處理器值，藉此修改 `mydbinstance`。透過 `--apply-immediately`，即可立即套用變更。如果要在下個排定的維護時段再套用變更，請省略 `--apply-immediately` 選項。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --use-default-processor-features \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --use-default-processor-features ^
  --apply-immediately
```

傳回資料庫執行個體的預設 CPU 核心數量

Example

以下範例會針對資料庫執行個體，將其資料庫執行個體類別傳回至預設的 CPU 核心數量，藉此修改 *mydbinstance*。然而，系統並不會變更每核心執行緒數量的設定。透過 `--apply-immediately`，即可立即套用變更。如果要在下個排定的維護時段再套用變更，請省略 `--apply-immediately` 選項。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --processor-features "Name=coreCount,Value=DEFAULT" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --processor-features "Name=coreCount,Value=DEFAULT" ^
  --apply-immediately
```

傳回資料庫執行個體的預設每核心執行緒數量

Example

以下範例會針對資料庫執行個體，將其資料庫執行個體類別傳回至預設的每核心執行緒數量，藉此修改 *mydbinstance*。然而，系統並不會變更 CPU 核心數量的設定。透過 `--apply-immediately`，即可立即套用變更。如果要在下個排定的維護時段再套用變更，請省略 `--apply-immediately` 選項。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --processor-features "Name=threadsPerCore,Value=DEFAULT" \  
  --apply-immediately
```

```
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --processor-features "Name=threadsPerCore,Value=DEFAULT" ^  
  --apply-immediately
```

RDS API

呼叫下列其中一個 Amazon RDS API 操作時，您可以藉機設定資料庫執行個體的處理器功能：

- [CreateDBInstance](#)
- [ModifyDBInstance](#)
- [恢復資料庫快照](#)
- [恢復 B S3 InstanceFrom](#)
- [恢復時InstanceToPointIn間](#)

若要使用 Amazon RDS API 為資料庫執行個體的資料庫執行個體類別設定處理器功能，請在呼叫中加入 `ProcessFeatures` 參數。

此參數的語法如下。

```
ProcessFeatures "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

您可以使用 `coreCount` 功能名稱來指定 CPU 核心數量，並利用 `threadsPerCore` 功能名稱指定每核心是否要啟用多重執行緒。

您可以執行 `InstanceOptions` 作業並指定參數的執行個體類別，以檢視特定 [DescribeOrderable資料庫執行個體類別](#) 的有效處理器 `DBInstanceClass` 值。您也可以使用下列操作：

- [DescribeDBInstances](#) – 顯示指定的資料庫執行個體的處理器資訊。
- [DescribeDBSnapshots](#) – 顯示指定的資料庫快照的處理器資訊。
- [DescribeValidDB InstanceModifications](#) — 顯示指定資料庫執行個體對處理器的有效修改。

在上述操作的輸出中，只有在符合下列條件時，處理器功能的值才不是 `null`：

- 您正在使用 RDS for Oracle 資料庫執行個體。
- 您的 RDS for Oracle 資料庫執行個體支援變更處理器值。
- 目前的 CPU 核心和執行緒設定設定為非預設值。

如果不符合上述條件，您可以取得使用 [DescribeDBInstances](#) 的執行個體類型。您可以透過執行 EC2 作業 [DescribeInstance](#) 類型取得此執行個體類型的處理器資訊。

的資料庫執行個體類別的硬體規格

下列術語用於敘述資料庫執行個體類別的硬體規格：

vCPU

虛擬中央處理單元 (CPU) 的數量。虛擬 CPU 即為容量單位，可用來比較資料庫執行個體類別。您不再購買或租用特定的處理器並使用數月或數年，而是以小時為單位租用容量。我們的目標是在實際基礎硬體的限制範圍內，提供一致且明確的 CPU 容量。

ECU

Amazon EC2 執行個體整數處理能力的相對測量單位。為了讓開發人員能輕鬆地比較不同執行個體類別的 CPU 容量，我們定義了 Amazon EC2 運算單位。分配給特定執行個體的 CPU 量以這些 EC2 運算單位來表示。目前，一個 ECU 所提供的 CPU 容量等同於 1.0–1.2 GHz 的 2007 Opteron 或 2007 Xeon 處理器。

記憶體 (GiB)

分配給資料庫執行個體的 RAM，以 GiB 為單位。記憶體和 vCPU 之間的比率通常是固定的。舉 db.r4 執行個體類別為例，其記憶體和 vCPU 的比例類似 db.r5 執行個體類別。然而，大多時候 db.r5 執行個體的效能比 db.r4 執行個體更好、更一致。

EBS 最佳化

資料庫執行個體使用最佳化組態堆疊，並為輸入/輸出提供額外專用容量。此最佳化透過減少輸入/輸出與執行個體的其他流量之間的爭用情況，來提供最佳效能。如需有關 Amazon EBS 最佳化執行個體的詳細資訊，請參閱 [Amazon EC2 使用者指南中的 Amazon EBS 優化執行個體](#)。

EBS 最佳化執行個體具有基準和最高 IOPS 速率。最高 IOPS 速率會在資料庫執行個體層級強制執行。總 IOPS 速率高於最大值的 EBS 磁碟區不能超過執行個體層級閾值。例如，如果特定資料庫執行個體類別的最大 IOPS 為 40,000，且您連線了四個 64,000 IOPS EBS 磁碟區，則最大 IOPS 為 40,000，而非 256,000。如需了解每個 EC2 執行個體類型特有的 IOPS 上限，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的 [受支援執行個體類型](#)。

最大 EBS 頻寬 (Mbps)

每秒的最高 EBS 頻寬，以百萬位元為單位。若將頻寬除以 8，即可取得預期的傳輸量 (MB/秒)。

Important

用於 Amazon RDS 執行個體的一般用途 SSD (gp2) 磁碟區，其輸送量限制在多數情況下為 250 MiB/秒。不過，輸送量限制可能因磁碟區大小而不同。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Amazon EBS 磁碟區類型](#)。

網路頻寬

相對於其他資料庫執行個體類別的網路速度。

您可以在下表中找到有關的 Amazon RDS 資料庫執行個體類別的硬體詳細資訊。

如需各資料庫執行個體類別的 Amazon RDS 資料庫引擎支援資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m7g — 具有重力 on3 處理器的一般用途執行個體類別 AWS						
db.m7g.16xlarge	64	—	256	EBS 最佳化專用	20,000	30
db.m7g.12xlarge	48	—	192	EBS 最佳化專用	15,000	22.5
db.m7g.8xlarge	32	—	128	EBS 最佳化專用	10,000	15
db.m7g.4xlarge	16	—	64	EBS 最佳化專用	最高 10,000	最高 15
db.m7g.2xlarge*	8	—	32	EBS 最佳化專用	最高 10,000	最高 15

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m7g.xlarge*	4	—	16	EBS 最佳化專用	最高 10,000	最高 12.5
db.m7g.large*	2	—	8	EBS 最佳化專用	最高 10,000	最高 12.5

db.m6g — 具有重力 on2 處理器的一般用途執行個體類別 AWS

db.m6g.16xlarge	64	—	256	EBS 最佳化專用	19,000	25
db.m6g.12xlarge	48	—	192	EBS 最佳化專用	13,500	20
db.m6g.8xlarge	32	—	128	EBS 最佳化專用	9,000	12
db.m6g.4xlarge	16	—	64	EBS 最佳化專用	4,750	最高 10
db.m6g.2xlarge*	8	—	32	EBS 最佳化專用	最高 4,750	最高 10
db.m6g.xlarge*	4	—	16	EBS 最佳化專用	最高 4,750	最高 10
db.m6g.large*	2	—	8	EBS 最佳化專用	最高 4,750	最高 10

db.m6gd — 具有重力 2 處理器和 SSD 儲存裝置的一般用途執行個體類別 AWS

db.m6gd.16xlarge	64	—	256	2 x 1900 NVMe SSD	19,000	25
db.m6gd.12xlarge	48	—	192	2 x 1425 NVMe SSD	13,500	20

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m6gd.8xlarge	32	—	128	1 x 1900 NVMe SSD	9,000	12
db.m6gd.4xlarge	16	—	64	1 x 950 NVMe SSD	4,750	最高 10
db.m6gd.2xlarge	8	—	32	1 x 474 NVMe SSD	最高 4,750	最高 10
db.m6gd.xlarge	4	—	16	1 x 237 NVMe SSD	最高 4,750	最高 10
db.m6gd.large	2	—	8	1 x 118 NVMe SSD	最高 4,750	最高 10

db.m6id - 採用第三代 Intel Xeon 可擴充處理器和 SSD 儲存體的一般用途執行個體類別

db.m6id.32xlarge	128	—	512	4 x 1900 NVMe SSD	40,000	50
db.m6id.24xlarge	96	—	384	4 x 1425 NVMe SSD	30,000	37.5
db.m6id.16xlarge	64	—	256	2 x 1900 NVMe SSD	20,000	25
db.m6id.12xlarge	48	—	192	2 x 1425 NVMe SSD	15,000	18.75
db.m6id.8xlarge	32	—	128	1 x 1900 NVMe SSD	10,000	12.5
db.m6id.4xlarge*	16	—	64	1 x 950 NVMe SSD	最高 10,000	最高 12.5
db.m6id.2xlarge*	8	—	32	1 x 474 NVMe SSD	最高 10,000	最高 12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m6id.xlarge*	4	—	16	1 x 237 NVMe SSD	最高 10,000	最高 12.5
db.m6id.large*	2	—	8	1 x 118 NVMe SSD	最高 10,000	最高 12.5

db.m6idn - 採用第三代 Intel Xeon 可擴充處理器、SSD 儲存體和網路最佳化的一般用途執行個體類別

db.m6idn.32xlarge	128	—	512	4 x 1900 NVMe SSD	80,000	200
db.m6idn.24xlarge	96	—	384	4 x 1425 NVMe SSD	60,000	150
db.m6idn.16xlarge	64	—	256	2 x 1900 NVMe SSD	40,000	100
db.m6idn.12xlarge	48	—	192	2 x 1425 NVMe SSD	30,000	75
db.m6idn.8xlarge	32	—	128	1 x 1900 NVMe SSD	20,000	50
db.m6idn.4xlarge*	16	—	64	1 x 950 NVMe SSD	最高 20,000	最高 50
db.m6idn.2xlarge*	8	—	32	1 x 474 NVMe SSD	最高 20,000	最高 40
db.m6idn.xlarge*	4	—	16	1 x 237 NVMe SSD	最高 20,000	最高 30
db.m6idn.large*	2	—	8	1 x 118 NVMe SSD	最高 20,000	最高 25

db.m6in - 採用第三代 Intel Xeon 可擴充處理器和網路最佳化的一般用途執行個體類別

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m6in.32xlarge	128	—	512	EBS 最佳化專用	80,000	200
db.m6in.24xlarge	96	—	384	EBS 最佳化專用	60,000	150
db.m6in.16xlarge	64	—	256	EBS 最佳化專用	40,000	100
db.m6in.12xlarge	48	—	192	EBS 最佳化專用	30,000	75
db.m6in.8xlarge	32	—	128	EBS 最佳化專用	20,000	50
db.m6in.4xlarge*	16	—	64	EBS 最佳化專用	最高 20,000	最高 50
db.m6in.2xlarge*	8	—	32	EBS 最佳化專用	最高 20,000	最高 40
db.m6in.xlarge*	4	—	16	EBS 最佳化專用	最高 20,000	最高 30
db.m6in.large*	2	—	8	EBS 最佳化專用	最高 20,000	最高 25

db.m6i - 採用第三代 Intel Xeon 可擴充處理器的一般用途執行個體類別

db.m6i.32xlarge	128	—	512	EBS 最佳化專用	40,000	50
db.m6i.24xlarge	96	—	384	EBS 最佳化專用	30,000	37.5
db.m6i.16xlarge	64	—	256	EBS 最佳化專用	20,000	25

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m6i.12xlarge	48	—	192	EBS 最佳化專用	15,000	18.75
db.m6i.8xlarge	32	—	128	EBS 最佳化專用	10,000	12.5
db.m6i.4xlarge*	16	—	64	EBS 最佳化專用	最高 10,000	最高 12.5
db.m6i.2xlarge*	8	—	32	EBS 最佳化專用	最高 10,000	最高 12.5
db.m6i.xlarge*	4	—	16	EBS 最佳化專用	最高 10,000	最高 12.5
db.m6i.large*	2	—	8	EBS 最佳化專用	最高 10,000	最高 12.5

db.m5d - 採用 Intel Xeon Platinum 處理器和 SSD 儲存體的一般用途執行個體類別

db.m5d.24xlarge	96	345	384	4 x 900 NVMe SSD	19,000	25
db.m5d.16xlarge	64	262	256	4 x 600 NVMe SSD	13,600	20
db.m5d.12xlarge	48	173	192	2 x 900 NVMe SSD	9,500	10
db.m5d.8xlarge	32	131	128	2 x 600 NVMe SSD	6,800	10
db.m5d.4xlarge	16	61	64	2 x 300 NVMe SSD	4,750	最高 10
db.m5d.2xlarge*	8	31	32	1 x 300 NVMe SSD	最高 4,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m5d.xlarge*	4	15	16	1 x 150 NVMe SSD	最高 4,750	最高 10
db.m5d.large*	2	10	8	1 x 75 NVMe SSD	最高 4,750	最高 10
db.m5 - 採用 Intel Xeon Platinum 處理器的一般用途執行個體類別						
db.m5.24xlarge	96	345	384	EBS 最佳化專用	19,000	25
db.m5.16xlarge	64	262	256	EBS 最佳化專用	13,600	20
db.m5.12xlarge	48	173	192	EBS 最佳化專用	9,500	10
db.m5.8xlarge	32	131	128	EBS 最佳化專用	6,800	10
db.m5.4xlarge	16	61	64	EBS 最佳化專用	4,750	最高 10
db.m5.2xlarge*	8	31	32	EBS 最佳化專用	最高 4,750	最高 10
db.m5.xlarge*	4	15	16	EBS 最佳化專用	最高 4,750	最高 10
db.m5.large*	2	10	8	EBS 最佳化專用	最高 4,750	最高 10
db.m4 - 採用 Intel Xeon 可擴充處理器的一般用途執行個體類別						
db.m4.16xlarge	64	188	256	EBS 最佳化專用	10,000	25

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m4.10xlarge	40	124.5	160	EBS 最佳化專用	4,000	10
db.m4.4xlarge	16	53.5	64	EBS 最佳化專用	2,000	高
db.m4.2xlarge	8	25.5	32	EBS 最佳化專用	1,000	高
db.m4.xlarge	4	13	16	EBS 最佳化專用	750	高
db.m4.large	2	6.5	8	EBS 最佳化專用	450	適中
db.m3 – 一般用途執行個體類別						
db.m3.2xlarge	8	26	30	EBS 最佳化專用	1,000	高
db.m3.xlarge	4	13	15	EBS 最佳化專用	500	高
db.m3.large	2	6.5	7.5	僅 EBS	—	適中
db.m3.medium	1	3	3.75	僅 EBS	—	適中
db.m1 – 一般用途執行個體類別						
db.m1.xlarge	4	4	15	EBS 最佳化專用	450	高
db.m1.large	2	2	7.5	EBS 最佳化專用	450	適中
db.m1.medium	1	1	3.75	僅 EBS	—	適中

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.m1.small	1	1	1.7	僅 EBS	—	非常低
db.x2iezn – 記憶體最佳化執行個體類別						
db.x2iezn.12xlarge	>48	—	1,536	EBS 最佳化專用	19,000	100
db.x2iezn.8xlarge	32	—	1,024	EBS 最佳化專用	12,000	75
db.x2iezn.6xlarge	24	—	768	EBS 最佳化專用	最高 9,500	50
db.x2iezn.4xlarge	16	—	512	EBS 最佳化專用	最高 4,750	最高 25
db.x2iezn.2xlarge	8	—	256	EBS 最佳化專用	最高 3,170	最高 25
db.x2iedn - 採用 SSD 儲存體和網路最佳化的記憶體優化執行個體類別						
db.x2iedn.32xlarge	128	—	4,096	2 x 1900 NVMe SSD	80,000	100
db.x2iedn.24xlarge	96	—	3,072	2 x 1425 NVMe SSD	60,000	75
db.x2iedn.16xlarge	64	—	2,048	1 x 1900 NVMe SSD	40,000	50
db.x2iedn.8xlarge	32	—	1,024	1 x 950 NVMe SSD	20,000	25
db.x2iedn.4xlarge	16	—	512	1 x 475 NVMe SSD	最高 20,000	最高 25

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.x2iedn.2xlarge	8	—	256	1 x 237 NVMe SSD	最高 20,000	最高 25
db.x2iedn.xlarge	4	—	128	1 x 118 NVMe SSD	最高 20,000	最高 25
db.x2idn - 採用 SSD 儲存體和網路最佳化的記憶體優化執行個體類別						
db.x2idn.32xlarge	128	—	2,048	2 x 1900 NVMe SSD	80,000	100
db.x2idn.24xlarge	96	—	1,536	2 x 1425 NVMe SSD	60,000	75
db.x2idn.16xlarge	64	—	1,024	1 x 1900 NVMe SSD	40,000	50
db.x2g – 記憶體優化執行個體類別						
db.x2g.16xlarge	64	—	1024	EBS 最佳化專用	19,000	25
db.x2g.12xlarge	48	—	768	EBS 最佳化專用	14,250	20
db.x2g.8xlarge	32	—	512	EBS 最佳化專用	9,500	12
db.x2g.4xlarge	16	—	256	EBS 最佳化專用	4,750	最高 10
db.x2g.2xlarge	8	—	128	EBS 最佳化專用	最高 4,750	最高 10
db.x2g.xlarge	4	—	64	EBS 最佳化專用	最高 4,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.x2g.large	2	—	32	EBS 最佳化專用	最高 4,750	最高 10
db.z1d - 採用 SSD 儲存體的記憶體優化執行個體類別						
db.z1d.12xlarge	48	271	384	2 x 900 NVMe SSD	14,000	25
db.z1d.6xlarge	24	134	192	1 x 900 NVMe SSD	7,000	10
db.z1d.3xlarge	12	75	96	1 x 450 NVMe SSD	3,500	最高 10
db.z1d.2xlarge	8	53	64	1 x 300 NVMe SSD	2,333	最高 10
db.z1d.xlarge*	4	28	32	1 x 150 NVMe SSD	最高 2,333	最高 10
db.z1d.large*	2	15	16	1 x 75 NVMe SSD	最高 2,333	最高 10
db.x1e – 記憶體最佳化的執行個體類別						
db.x1e.32xlarge	128	340	3,904	EBS 最佳化專用	14,000	25
db.x1e.16xlarge	64	179	1,952	EBS 最佳化專用	7,000	10
db.x1e.8xlarge	32	91	976	EBS 最佳化專用	3,500	最高 10
db.x1e.4xlarge	16	47	488	EBS 最佳化專用	1,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.x1e.2xlarge	8	23	244	EBS 最佳化專用	1,000	最高 10
db.x1e.xlarge	4	12	122	EBS 最佳化專用	500	最高 10
db.x1 – 記憶體優化執行個體類別						
db.x1.32xlarge	128	349	1,952	EBS 最佳化專用	14,000	25
db.x1.16xlarge	64	174.5	976	EBS 最佳化專用	7,000	10
db.r7g — 採用重力 on3 處理器的記憶體最佳化執行個體類別 AWS						
db.r7g.16xlarge	64	—	512	EBS 最佳化專用	20,000	30
db.r7g.12xlarge	48	—	384	EBS 最佳化專用	15,000	22.5
db.r7g.8xlarge	32	—	256	EBS 最佳化專用	10,000	15
db.r7g.4xlarge	16	—	128	EBS 最佳化專用	最高 10,000	最高 15
db.r7g.2xlarge*	8	—	64	EBS 最佳化專用	最高 10,000	最高 15
db.r7g.xlarge*	4	—	32	EBS 最佳化專用	最高 10,000	最高 12.5
db.r7g.large*	2	—	16	EBS 最佳化專用	最高 10,000	最高 12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
--------	------	-----	-----------	---------------	------------------	-------------

db.r6g — 具有重力 2 處理器的記憶體最佳化執行個體類別 AWS

db.r6g.16xlarge	64	—	512	EBS 最佳化專用	19,000	25
db.r6g.12xlarge	48	—	384	EBS 最佳化專用	13,500	20
db.r6g.8xlarge	32	—	256	EBS 最佳化專用	9,000	12
db.r6g.4xlarge	16	—	128	EBS 最佳化專用	4,750	最高 10
db.r6g.2xlarge*	8	—	64	EBS 最佳化專用	最高 4,750	最高 10
db.r6g.xlarge*	4	—	32	EBS 最佳化專用	最高 4,750	最高 10
db.r6g.large*	2	—	16	EBS 最佳化專用	最高 4,750	最高 10

db.r6gd — 記憶體最佳化執行個體類別，搭配重力 2 處理器和 SSD 儲存 AWS

db.r6gd.16xlarge	64	—	512	2 x 1900 NVMe SSD	19,000	25
db.r6gd.12xlarge	48	—	384	2 x 1425 NVMe SSD	13,500	20
db.r6gd.8xlarge	32	—	256	1 x 1900 NVMe SSD	9,000	12
db.r6gd.4xlarge	16	—	128	1 x 950 NVMe SSD	4,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r6gd.2xlarge	8	—	64	1 x 474 NVMe SSD	最高 4,750	最高 10
db.r6gd.xlarge	4	—	32	1 x 237 NVMe SSD	最高 4,750	最高 10
db.r6gd.large	2	—	16	1 x 118 NVMe SSD	最高 4,750	最高 10

db.r6id - 採用第三代 Intel Xeon 可擴充處理器和 SSD 儲存體的一般用途執行個體類別

db.r6id.32xlarge	128	—	1,024	4x1900 NVMe SSD	40,000	50
db.r6id.24xlarge	96	—	768	4x1425 NVMe SSD	30,000	37.5
db.r6id.16xlarge	64	—	512	2x1900 NVMe SSD	20,000	25
db.r6id.12xlarge	48	—	384	2x1425 NVMe SSD	15,000	18.75
db.r6id.8xlarge	32	—	256	1x1900 NVMe SSD	10,000	12.5
db.r6id.4xlarge*	16	—	128	1x950 NVMe SSD	最高 10,000	最高 12.5
db.r6id.2xlarge*	8	—	64	1x474 NVMe SSD	最高 10,000	最高 12.5
db.r6id.xlarge*	4	—	32	1x237 NVMe SSD	最高 10,000	最高 12.5
db.r6id.large*	2	—	16	1x118 NVMe SSD	最高 10,000	最高 12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
--------	------	-----	-----------	---------------	------------------	-------------

db.r6idn - 採用第三代 Intel Xeon 可擴充處理器、SSD 儲存體和網路最佳化的記憶體優化執行個體類別

db.r6idn.32xlarge	128	—	1,024	4x1900 NVMe SSD	80,000	200
db.r6idn.24xlarge	96	—	768	4x1425 NVMe SSD	60,000	150
db.r6idn.16xlarge	64	—	512	2x1900 NVMe SSD	40,000	100
db.r6idn.12xlarge	48	—	384	2x1425 NVMe SSD	30,000	75
db.r6idn.8xlarge	32	—	256	1x1900 NVMe SSD	20,000	50
db.r6idn.4xlarge*	16	—	128	1x950 NVMe SSD	最高 20,000	最高 50
db.r6idn.2xlarge*	8	—	64	1x474 NVMe SSD	最高 20,000	最高 40
db.r6idn.xlarge*	4	—	32	1x237 NVMe SSD	最高 20,000	最高 30
db.r6idn.large*	2	—	16	1x118 NVMe SSD	最高 20,000	最高 25

db.r6in - 採用第三代 Intel Xeon 可擴充處理器和網路最佳化的記憶體優化執行個體類別

db.r6in.32xlarge	128	—	1,024	EBS 最佳化專用	80,000	200
db.r6in.24xlarge	96	—	768	EBS 最佳化專用	60,000	150

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r6in.16xlarge	64	—	512	EBS 最佳化專用	40,000	100
db.r6in.12xlarge	48	—	384	EBS 最佳化專用	30,000	75
db.r6in.8xlarge	32	—	256	EBS 最佳化專用	20,000	50
db.r6in.4xlarge*	16	—	128	EBS 最佳化專用	最高 20,000	最高 50
db.r6in.2xlarge*	8	—	64	EBS 最佳化專用	最高 20,000	最高 40
db.r6in.xlarge*	4	—	32	EBS 最佳化專用	最高 20,000	最高 30
db.r6in.large*	2	—	16	EBS 最佳化專用	最高 20,000	最高 25

db.r6id - 採用第三代 Intel Xeon 可擴充處理器和 SSD 儲存體的一般用途執行個體類別

db.r6id.32xlarge	128	—	1,024	4x1900 NVMe SSD	40,000	50
db.r6id.24xlarge	96	—	768	4x1425 NVMe SSD	30,000	37.5
db.r6id.16xlarge	64	—	512	2x1900 NVMe SSD	20,000	25
db.r6id.12xlarge	48	—	384	2x1425 NVMe SSD	15,000	18.75
db.r6id.8xlarge	32	—	256	1x1900 NVMe SSD	10,000	12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r6id.4xlarge*	16	—	128	1x950 NVMe SSD	最高 10,000	最高 12.5
db.r6id.2xlarge*	8	—	64	1x474 NVMe SSD	最高 10,000	最高 12.5
db.r6id.xlarge*	4	—	32	1x237 NVMe SSD	最高 10,000	最高 12.5
db.r6id.large*	2	—	16	1x118 NVMe SSD	最高 10,000	最高 12.5

db.r6i — Oracle 記憶體最佳化執行個體類別，預先設定為高記憶體、儲存體和 I/O

數據庫 .r6i.8 倍大.	32	—	1024	EBS 最佳化專用	40,000	50
資料庫 .r6. 8 倍大.	32	—	768	EBS 最佳化專用	30,000	37.5
資料庫 .r6. 6 倍大.	24	—	768	EBS 最佳化專用	30,000	37.5
資料庫 .r6 .4 X 大型 .tpc2.4	16	—	512	EBS 最佳化專用	20,000	25
資料庫 .r6 .4 X 大.	16	—	384	EBS 最佳化專用	15,000	18.75
資料庫 .r6 .4 X 大型 .tpc2.2	16	—	256	EBS 最佳化專用	10,000	12.5
資料庫 .r6i.2 大型.	8	—	512	EBS 最佳化專用	20,000	12.5
資料庫 .r6i.2 大型 .tpc2.4	8	—	256	EBS 最佳化專用	10,000	12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
資料庫 .r6i.2 大型 .tpc1.2	8	—	128	EBS 最佳化專用	最高 10,000	12.5
資料庫 .r6. 大型 .tpc2.4	4	—	128	EBS 最佳化專用	最高 10,000	12.5
資料庫 .r6. 大型 .tpc2.2	4	—	64	EBS 最佳化專用	最高 10,000	12.5
大型 .tpc1.2.	2	—	32	EBS 最佳化專用	最高 10,000	12.5
db.r6i - 採用第三代 Intel Xeon 可擴充處理器的記憶體優化執行個體類別						
db.r6i.32xlarge	128	—	1,024	EBS 最佳化專用	40,000	50
db.r6i.24xlarge	96	—	768	EBS 最佳化專用	30,000	37.5
db.r6i.16xlarge	64	—	512	EBS 最佳化專用	20,000	25
db.r6i.12xlarge	48	—	384	EBS 最佳化專用	15,000	18.75
db.r6i.8xlarge	32	—	256	EBS 最佳化專用	10,000	12.5
db.r6i.4xlarge*	16	—	128	EBS 最佳化專用	最高 10,000	最高 12.5
db.r6i.2xlarge*	8	—	64	EBS 最佳化專用	最高 10,000	最高 12.5
db.r6i.xlarge*	4	—	32	EBS 最佳化專用	最高 10,000	最高 12.5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r6i.large*	2	—	16	EBS 最佳化專用	最高 10,000	最高 12.5
db.r5d - 採用 Intel Xeon Platinum 處理器和 SSD 儲存體的記憶體優化執行個體類別						
db.r5d.24xlarge	96	347	768	4 x 900 NVMe SSD	19,000	25
db.r5d.16xlarge	64	264	512	4 x 600 NVMe SSD	13,600	20
db.r5d.12xlarge	48	173	384	2 x 900 NVMe SSD	9,500	10
db.r5d.8xlarge	32	132	256	2 x 600 NVMe SSD	6,800	10
db.r5d.4xlarge	16	71	128	2 x 300 NVMe SSD	4,750	最高 10
db.r5d.2xlarge*	8	38	64	1 x 300 NVMe SSD	最高 4,750	最高 10
db.r5d.xlarge*	4	19	32	1 x 150 NVMe SSD	最高 4,750	最高 10
db.r5d.large*	2	10	16	1 x 75 NVMe SSD	最高 4,750	最高 10
db.r5b - 採用 Intel Xeon Platinum 處理器和 EBS 最佳化的記憶體優化執行個體類別						
db.r5b.24xlarge	96	347	768	EBS 最佳化專用	60,000	25
db.r5b.16xlarge	64	264	512	EBS 最佳化專用	40,000	20

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r5b.12xlarge	48	173	384	EBS 最佳化專用	30,000	10
db.r5b.8xlarge	32	132	256	EBS 最佳化專用	20,000	10
db.r5b.4xlarge	16	71	128	EBS 最佳化專用	10,000	最高 10
db.r5b.2xlarge*	8	38	64	EBS 最佳化專用	最高 10,000	最高 10
db.r5b.xlarge*	4	19	32	EBS 最佳化專用	最高 10,000	最高 10
db.r5b.large*	2	10	16	EBS 最佳化專用	最高 10,000	最高 10
db.r5 – Oracle 記憶體最佳化執行個體類別，預先設定高記憶體、儲存和輸入/輸出						
db.r5b.8xlarge.tpc2.mem3x	32	—	768	EBS 最佳化專用	60,000	25
db.r5b.6xlarge.tpc2.mem4x	24	—	768	EBS 最佳化專用	60,000	25
db.r5b.4xlarge.tpc2.mem4x	16	—	512	EBS 最佳化專用	40,000	20
db.r5b.4xlarge.tpc2.mem3x	16	—	384	EBS 最佳化專用	30,000	10
db.r5b.4xlarge.tpc2.mem2x	16	—	256	EBS 最佳化專用	20,000	10
db.r5b.2xlarge.tpc2.mem8x	8	—	512	EBS 最佳化專用	40,000	20

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r5b.2xlarge.tpc2.mem4x	8	—	256	EBS 最佳化專用	20,000	10
db.r5b.2xlarge.tpc1.mem2x	8	—	128	EBS 最佳化專用	10,000	最高 10
db.r5b.xlarge.tpc2.mem4x	4	—	128	EBS 最佳化專用	10,000	最高 10
db.r5b.xlarge.tpc2.mem2x	4	—	64	EBS 最佳化專用	最高 10,000	最高 10
db.r5b.large.tpc1.mem2x	2	—	32	EBS 最佳化專用	最高 10,000	最高 10
db.r5 - 採用 Intel Xeon Platinum 處理器的記憶體優化執行個體類別						
db.r5.24xlarge	96	347	768	EBS 最佳化專用	19,000	25
db.r5.16xlarge	64	264	512	EBS 最佳化專用	13,600	20
db.r5.12xlarge	48	173	384	EBS 最佳化專用	9,500	12
db.r5.8xlarge	32	132	256	EBS 最佳化專用	6,800	10
db.r5.4xlarge	16	71	128	EBS 最佳化專用	4,750	最高 10
db.r5.2xlarge*	8	38	64	EBS 最佳化專用	最高 4,750	最高 10
db.r5.xlarge*	4	19	32	EBS 最佳化專用	最高 4,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r5.large*	2	10	16	EBS 最佳化專用	最高 4,750	最高 10

db.r5 – Oracle 記憶體最佳化執行個體類別，預先設定高記憶體、儲存和輸入/輸出

db.r5.12xlarge.tpc2.mem2x	48	—	768	EBS 最佳化專用	19,000	25
db.r5.8xlarge.tpc2.mem3x	32	—	768	EBS 最佳化專用	19,000	25
db.r5.6xlarge.tpc2.mem4x	24	—	768	EBS 最佳化專用	19,000	25
db.r5.4xlarge.tpc2.mem4x	16	—	512	EBS 最佳化專用	13,600	20
db.r5.4xlarge.tpc2.mem3x	16	—	384	EBS 最佳化專用	9,500	10
db.r5.4xlarge.tpc2.mem2x	16	—	256	EBS 最佳化專用	6,800	10
db.r5.2xlarge.tpc2.mem8x	8	—	512	EBS 最佳化專用	13,600	20
db.r5.2xlarge.tpc2.mem4x	8	—	256	EBS 最佳化專用	6,800	10
db.r5.2xlarge.tpc1.mem2x	8	—	128	EBS 最佳化專用	4,750	最高 10
db.r5.xlarge.tpc2.mem4x	4	—	128	EBS 最佳化專用	4,750	最高 10
db.r5.xlarge.tpc2.mem2x	4	—	64	EBS 最佳化專用	最高 4,750	最高 10

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r5.large.tpc1.mem2x	2	—	32	EBS 最佳化專用	最高 4,750	最高 10
db.r4 - 採用 Intel Xeon 可擴充處理器的記憶體優化執行個體類別						
db.r4.16xlarge	64	195	488	EBS 最佳化專用	14,000	25
db.r4.8xlarge	32	99	244	EBS 最佳化專用	7,000	10
db.r4.4xlarge	16	53	122	EBS 最佳化專用	3,500	最高 10
db.r4.2xlarge	8	27	61	EBS 最佳化專用	1,700	最高 10
db.r4.xlarge	4	13.5	30.5	EBS 最佳化專用	850	最高 10
db.r4.large	2	7	15.25	EBS 最佳化專用	425	最高 10
db.r3 – 記憶體優化執行個體類別						
db.r3.8xlarge	32	104	244	僅 EBS	—	10
db.r3.4xlarge	16	52	122	EBS 最佳化專用	2,000	高
db.r3.2xlarge	8	26	61	EBS 最佳化專用	1,000	高
db.r3.xlarge	4	13	30.5	EBS 最佳化專用	500	適中

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.r3.large	2	6.5	15.25	EBS 最佳化專用	—	適中

db.c6gd — 運算最佳化執行個體類別 (僅適用於異地同步備份資料庫叢集部署)

分貝 .C6 公分. 16 大	64	—	128	2 x 1900 NVMe SSD	19,000	25
分貝 .c6 公分. 12 倍大	48	—	96	2 x 1425 NVMe SSD	13,500	20
分貝 .c6 公分. 8 倍大	32	—	64	1 x 1900 NVMe SSD	9,000	12
分貝 .c6 公升. 4 大	16	—	32	1 x 950 NVMe SSD	4,750	最高 10
DB.C6 公克. 2 大	8	—	16	1 x 474 NVMe SSD	最高 4,750	最高 10
分貝 .c6 公克.	4	—	8	1 x 237 NVMe SSD	最高 4,750	最高 10
大包 .c6 公克	2	—	4	1 x 118 NVMe SSD	最高 4,750	最高 10
分貝 .c6 公克. 中	1	—	2	1 x 59 NVMe 固態硬碟	最高 4,750	最高 10

db.t4g — 具有重力 2 處理器的高效能執行個體類別 AWS

db.t4g.2xlarge*	8	—	32	EBS 最佳化專用	最高 2,780	最高 5
db.t4g.xlarge*	4	—	16	EBS 最佳化專用	最高 2,780	最高 5

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.t4g.large*	2	—	8	EBS 最佳化專用	最高 2,780	最高 5
db.t4g.medium*	2	—	4	EBS 最佳化專用	最高 2,085	最高 5
db.t4g.small*	2	—	2	EBS 最佳化專用	最高 2,085	最高 5
db.t4g.micro*	2	—	1	EBS 最佳化專用	最高 2,085	最高 5
db.t3 – 爆量效能執行個體類別						
db.t3.2xlarge*	8	變數	32	EBS 最佳化專用	最高 2,048	最高 5
db.t3.xlarge*	4	變數	16	EBS 最佳化專用	最高 2,048	最高 5
db.t3.large*	2	變數	8	EBS 最佳化專用	最高 2,048	最高 5
db.t3.medium*	2	變數	4	EBS 最佳化專用	最高 1,536	最高 5
db.t3.small*	2	變數	2	EBS 最佳化專用	最高 1,536	最高 5
db.t3.micro*	2	變數	1	EBS 最佳化專用	最高 1,536	最高 5
db.t2 – 爆量效能執行個體類別						
db.t2.2xlarge	8	變數	32	僅 EBS	—	適中
db.t2.xlarge	4	變數	16	僅 EBS	—	適中

執行個體類別	vCPU	ECU	記憶體 (GiB)	執行個體儲存體 (GiB)	最大 EBS 頻寬 (Mbps)	網路頻寬 (Gbps)
db.t2.large	2	變數	8	僅 EBS	—	適中
db.t2.medium	2	變數	4	僅 EBS	—	適中
db.t2.small	1	變數	2	僅 EBS	—	低
db.t2.micro	1	變數	1	僅 EBS	—	低

* 這些資料庫執行個體類別支援至少每 24 小時一次維持最佳效能 30 分鐘。如需有關基礎 EC2 執行個體類型基準效能的詳細資訊，請參閱 [Amazon EC2 使用者指南中的 Amazon EBS 優化執行個體](#)。

** r3.8xlarge 資料庫執行個體類別沒有專用 EBS 頻寬，因此不提供 EBS 最佳化。對於此執行個體類別，網路流量和 Amazon EBS 流量會共用同一個 10 GB 網路界面。

Amazon RDS 資料庫執行個體儲存體

適用於 Db2、MariaDB、MySQL、PostgreSQL、甲骨文和 Microsoft SQL 伺服器的資料庫執行個體使用 Amazon 彈性區塊存放區 (Amazon EBS) 磁碟區進行資料庫和日誌儲存。

在某些情況下，您的資料庫工作負載可能無法達到佈建 IOPS 的 100%。如需詳細資訊，請參閱 [影響儲存體效能的因素](#)。

如需執行個體儲存體定價的詳細資訊，請參閱 [Amazon RDS 定價](#)。

Amazon RDS 儲存類型

Amazon RDS 提供三種儲存類型：佈建 IOPS 固態硬碟 (也稱為 io1 和 io2 區塊快速)、一般用途固態硬碟 (也稱為 gp2 和 gp3) 和磁性 (也稱為標準)。各類型的效能特性和價格不盡相同，這可讓您量身打造符合您資料庫工作負載需求的儲存效能及成本。您可以建立 Db2、MySQL、MariaDB、甲骨文、SQL 伺服器和 PostgreSQL RDS 資料庫執行個體，並具有高達 64 兆位元組 (TiB) 的儲存體。適用於 Db2 的 RDS 不支援 gp3 和磁帶儲存類型。

以下清單簡要描述這三種儲存體類型：

- 佈建 IOPS SSD – 佈建 IOPS 儲存體的設計符合輸入/輸出密集工作負載的需求，特別是低延遲要求和一致性輸入/輸出輸送量的資料庫工作負載。佈建 IOPS 儲存體最適合於生產環境。

如需佈建 IOPS 儲存體的詳細資訊 (包括儲存體大小範圍)，請參閱 [佈建 IOPS SSD 儲存體](#)。

- 一般用途 SSD – 一般用途 SSD 磁碟區提供經濟實惠的儲存體，非常適合於在中型資料庫執行個體上執行的廣泛工作負載。一般用途儲存體最適用於開發與測試環境。

如需一般用途 SSD 儲存體的詳細資訊 (包括儲存體大小範圍)，請參閱 [一般用途 SSD 儲存體](#)。

- 磁性 – Amazon RDS 也支援磁帶儲存，以提供回溯相容性。建議您針對任何新的儲存需求，使用一般用途 SSD 或佈建 IOPS SSD。磁性儲存裝置上資料庫執行個體允許的最大儲存容量為 3 TiB。如需詳細資訊，請參閱 [磁性儲存裝置 \(舊版，不建議使用\)](#)。

當您選取一般用途 SSD 或佈建 IOPS SSD 時，根據選取的引擎和請求的儲存容量，Amazon RDS 會自動跨多個磁碟區進行分割以增強效能，如下表所示。

資料庫引擎	Amazon RDS 儲存體大小	已佈建的磁碟區數量
Db2	低於 400 GiB	1

資料庫引擎	Amazon RDS 儲存體大小	已佈建的磁碟區數量
Db2	兆 GiB	4
MariaDB、MySQL 和 PostgreSQL	低於 400 GiB	1
MariaDB、MySQL 和 PostgreSQL	兆 GiB	4
Oracle	低於 200 GiB	1
Oracle	千兆 GiB	4
SQL Server	任何	1

當您修改一般用途 SSD 或佈建 IOPS SSD 磁碟區時，它會經歷一連串狀態。當磁碟區處於 `optimizing` 狀態時，您的磁碟區效能介於來源與目標組態規格之間。過渡體積的性能將不低於兩個規格中的較低。

Important

當您修改執行個體的儲存，使其從一個磁碟區變為四個磁碟區，或使用磁性儲存修改執行個體時，Amazon RDS 不會使用彈性磁碟區功能。反之，Amazon RDS 會佈建新磁碟區，並以透明方式將資料從舊磁碟區移至新磁碟區。此操作會耗用舊磁碟區和新磁碟區兩者的大量 IOPS 和輸送量。根據磁碟區的大小和修改期間出現的資料庫工作負載量而定，此作業可能會消耗大量 IOPS、大幅增加 I/O 延遲，並且需要數小時才能完成，而 RDS 執行個體仍處於 `Modifying` 狀態。

佈建 IOPS SSD 儲存體

對於需要快速且一致 I/O 效能的生產應用程式，我們建議佈建 IOPS 儲存體。佈建 IOPS 儲存體是一種提供可預測效能與一致性低延遲的儲存體類型。佈建 IOPS 儲存體已針對線上交易處理 (OLTP) 工作負載進行最佳化，這些工作負載需要一致的效能。佈建 IOPS 可協助進行這些工作負載的效能調校。

建立資料庫執行個體時，您可以指定 IOPS 速率和磁碟區的大小。Amazon RDS 會為資料庫執行個體提供該 IOPS 費率，直到您對其進行變更。

Amazon RDS 提供兩種類型的佈建 IOPS 固態硬碟儲存：[io2 塊快速存儲 \(推薦\)](#) 和 [io1 儲存裝置 \(前一代產品\)](#)。

io2 塊快速存儲 (推薦)

對於 I/O 密集型和延遲敏感的工作負載，您可以使用佈建 IOPS SSD io2 區塊快速儲存，每秒達到 256,000 個 I/O 作業 (IOPS)。io2 區塊快速磁碟區的輸送量會根據每個磁碟區佈建的 IOPS 量以及所執行的 I/O 作業大小而有所不同。

以 AWS 硝基系統為基礎的所有 RDS io2 磁碟區都是 io2 區塊快速磁碟區，並提供低於一毫秒的平均延遲。不基於 AWS 硝基系統的數據庫實例是 io2 卷。

下表顯示每個資料庫引擎和儲存體大小範圍的佈建 IOPS 範圍和最大輸送量。

資料庫引擎	儲存體大小範圍	佈建 IOPS 範圍	最大輸送量
DB2、Maria DB、MySQL 和 PostgreSQL	百分之六十 GiB	1,000–256,000 IOPS	4,000 MiB/s
Oracle	兆 GiB	一百至九萬 IOT	4,000 MiB/s
Oracle	千兆 GiB	1,000–256,000 IOPS	每秒 4 千兆
SQL Server	GiB	1,000–256,000 IOPS	4,000 MiB/s

Note

¹ 對於 Oracle，在特定情況下 (例如非常大的資料庫執行個體大小和大型讀取)，您可能會看到更高的最大輸送量。

修改 SQL 伺服器執行個體以使用 gp2、gp3 或 io1 磁碟區到 io2 磁碟區之後，您可以允許 io2 磁碟區大小增加到 64 TiB。不過，一旦 io2 磁碟區大小超過 16 TiB，就無法將儲存磁碟區變更回 gp2、gp3 或 io1。若要還原為 gp2、gp3 或 io1，請將資料大小減少為小於 16 TiB，然後繼續進行磁碟區類型變更。

IOPS 和儲存大小範圍有下列限制：

- IOPS 與已配置儲存體的比例 (以 GiB 為單位) 不得超過 1000:1。對於不以 AWS 硝基系統為基礎的資料庫執行個體，比率为 500:1。
- 可以透過大小為 256 GiB 或更大的磁碟區佈建最大 IOPS ($1,000 \text{ IOPS} \times 256 \text{ GiB} = 256,000 \text{ IOPS}$)。對於不以 AWS 硝基系統為基礎的資料庫執行個體，最大 IOPS 可達到 512 GiB ($500 \text{ IOPS} \times 512 \text{ GiB} = 256,000 \text{ IOPS}$)。
- 輸送量可按比例擴展至每個佈建 IOPS 0.256 MiB/s。具有 16 KiB I/O 大小的 256,000 IOPS，以及 16,000 個 IOPS 或更高的處理能力 (使用 256 KiB I/O 大小)，最大輸送量可達到 4,000 MiB/ 秒。對於不以 AWS 硝基系統為基礎的資料庫執行個體，在 16 KiB I/O 大小的情況下，可以在 128,000 IOPS 下達到 2,000 MB 的最大輸送量。
- 如果您使用的是儲存體自動調整規模，則 IOPS 和儲存體閾值上限之間的比率 (以 GiB 為單位) 同樣適用。如需儲存體自動調整規模的詳細資訊，請參閱 [使用 Amazon RDS 儲存體自動擴展自動管理容量](#)。

Amazon RDS io2 區塊快遞磁碟區可用於以下幾 AWS 區域種：

- 亞太區域 (香港)
- Asia Pacific (Mumbai)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太區域 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- 歐洲 (斯德哥爾摩)
- Middle East (Bahrain)
- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)

io1 儲存裝置 (前一代產品)

對於 I/O 密集型工作負載，您可以使用佈建 IOPS SSD io1 儲存體，並達成每秒高達 256,000 個 I/O 操作 (IOPS)。io1 磁碟區的輸送量會根據每個磁碟區佈建的 IOPS 量以及執行的 I/O 作業大小而有所不同。我們建議在可用的地方使用 io2 區塊快速儲存裝置。

下表顯示每個資料庫引擎和儲存體大小範圍的佈建 IOPS 範圍和最大輸送量。

資料庫引擎	儲存體大小範圍	佈建 IOPS 範圍	最大輸送量
DB2、Maria DB、MySQL 和 PostgreSQL	千兆 GiB	1,000–19,950 IOPS	500 MiB/s
DB2、Maria DB、MySQL 和 PostgreSQL	兆 GiB	1,000–256,000 IOPS	4,000 MiB/s
Oracle	兆 GiB	1,000–9,950 IOPS	500 MiB/s
Oracle	千兆 GiB	1 百至二萬五千歐羅布 ¹	4,000 MiB/s
SQL Server	20—16,384 GiB	百萬至 64,000 歐波平方	1,000 MiB/s

Note

¹ 對於甲骨文，您只能在 r5b 執行個體類型上佈建最大 256,000 IOPS。

² 對於 SQL 伺服器，只有在位於 m5*、m6i、r5*、r6i 和 z1d [執行個體類型上的基於硝基礎](#)的執行個體上才能保證最大 64,000 IOPS。其他執行個體類型的保證效能最高只到 32,000 IOPS。

IOPS 和儲存大小範圍有下列限制：

- IOPS 與配置儲存 (以 GiB 為單位) 的比率，針對 RDS for SQL Server 必須從 1 至 50，而針對其他 RDS 資料庫引擎則必須從 0.5 至 50。
- 如果您使用的是儲存體自動調整規模，則 IOPS 和儲存體閾值上限之間的比率 (以 GiB 為單位) 同樣適用。

如需儲存體自動調整規模的詳細資訊，請參閱 [使用 Amazon RDS 儲存體自動擴展自動管理容量](#)。

將佈建 IOPS 儲存體與異地同步備份部署或僅供讀取複本結合

針對生產環境 OLTP 使用案例，建議您使用異地同步備份部署，透過佈建 IOPS 儲存體強化容錯能力，提供快速且可預期的效能。

您也可以將佈建 IOPS 儲存與 MySQL、MariaDB 或 PostgreSQL 的僅供讀取複本搭配使用。僅供讀取複本的儲存體類型是獨立於主要資料庫執行個體之外的。例如，您可能將一般用途 SSD 用於主要資料庫執行個體的僅供讀取複本，且該資料庫執行個體使用了佈建 IOPS SSD 儲存體來減少成本。但是，在此情況下，僅供讀取複本的效能可能與主要資料庫執行個體和僅供讀取複本都使用佈建 IOPS 儲存的組態不同。

佈建 IOPS 儲存體成本

若使用佈建 IOPS 儲存體，不論您是否在特定月分中使用這些佈建的資源，都會被計費。

如需定價的詳細資訊，請參閱 [Amazon RDS 定價](#)。

從 Amazon RDS 佈建 IOPS 儲存體取得最佳效能

如果您的工作負載受到 I/O 限制，使用佈建 IOPS 儲存可增加系統可同時處理的 I/O 要求數目。由於佇列中的輸入/輸出請求耗費的時間較少，因此增加的並行數量可降低延遲。降低的延遲加快了資料庫遞交，這改善了回應時間也提升了資料庫傳輸量。

佈建的 IOPS 儲存可透過指定 IOPS 來保留 I/O 容量。但是，與任何其他系統容量屬性一樣，它在負載下的最大傳輸量，受限於最先消耗的資源。該資源可能是網路頻寬、CPU、記憶體或資料庫內部資源。

一般用途 SSD 儲存體

一般用途儲存體提供符合成本效益的儲存體，對於大多數不具延遲或效能敏感的資料庫工作負載，

Note

與使用佈建 IOPS 儲存的執行個體相比，使用一般用途儲存的資料庫執行個體可能會遇到更長的延遲。如果您需要在這些操作之後延遲最小的資料庫執行個體，建議您使用 [佈建 IOPS SSD 儲存體](#)。

Amazon RDS 提供兩種類型的一般用途儲存：[gp3 儲存裝置 \(建議使用\)](#)和[gp2 儲存裝置 \(前一代產品\)](#)。

gp3 儲存裝置 (建議使用)

透過使用一般用途 gp3 儲存磁碟區，您可以獨立於儲存容量自訂儲存效能。儲存體效能是每秒 I/O 操作數 (IOPS) 的組合，以及儲存磁碟區可以多快執行讀取和寫入 (儲存體輸送量)。在 gp3 儲存磁碟區上，Amazon RDS 提供 3000 IOPS 和 125 MiB/秒的基準儲存效能。

對於除了適用於 SQL Server 的 RDS 資料庫引擎之外的每個 RDS 資料庫引擎，當 gp3 磁碟區的儲存區大小達到特定臨界值時，基準儲存區效能就會增加。這是因為磁碟區分割，其中儲存體會使用四個磁碟區，而不是一個磁碟區。RDS for SQL Server 不支援磁碟區分割，因此沒有閾值。對於分段磁碟區，Amazon RDS 可提供 12,000 IOPS 和 500 MB 的基準儲存效能。

下表顯示 Amazon RDS 資料庫引擎上 gp3 磁碟區的儲存體效能 (包括閾值)。

資料庫引擎	儲存體大小	基準儲存體效能	佈建 IOPS 範圍	佈建儲存體輸送量的範圍
DB2、Maria DB、MySQL 和 PostgreSQL	GiB	3,000 IOPS/125 MiB/秒	N/A	N/A
DB2、Maria DB、MySQL 和 PostgreSQL	兆 GiB	12,000 IOPS/500 MiB/秒	12,000–64,000 IOPS	500–4,000 MiB/秒
Oracle	兆 GiB	3,000 IOPS/125 MiB/秒	N/A	N/A
Oracle	千兆 GiB	12,000 IOPS/500 MiB/秒	12,000–64,000 IOPS	500–4,000 MiB/秒
SQL Server	20—16,384 GiB	3,000 IOPS/125 MiB/秒	3,000–16,000 IOPS	125–1,000 MiB/秒

對於 RDS for SQL Server 以外的每個資料庫引擎，當儲存體大小等於或高於閾值時，您都可以佈建額外的 IOPS 和儲存體輸送量。對於 RDS for SQL Server，您可以針對任何可用的儲存體大小佈建額外的 IOPS 和儲存體輸送量。對於所有資料庫引擎，您只需支付額外佈建的儲存體效能。如需詳細資訊，請參閱 [Amazon RDS 定價](#)。

雖然新增的佈建 IOPS 和儲存體輸送量不取決於儲存體大小，但它們彼此相關。當您針對 MariaDB 和 MySQL 將 IOPS 提高到 32,000 以上時，儲存體輸送量值會自動從 500 增加。MiBps 例如，當您在適用於 MySQL 的 RDS 上將 IOPS 設定為 40,000 時，儲存區輸送量必須至少為 6 MiBps 25。Db2、甲骨文、PostgreSQL 和 SQL 伺服器資料庫執行個體不會自動增加。

對於異地同步備份資料庫叢集，Amazon RDS 會根據您佈建的 IOPS 自動設定輸送量值。您無法修改輸送量值。

RDS 上 gp3 磁碟區的儲存體效能值具有限制條件：

- 對於所有支援的資料庫引擎，儲存體輸送量與 IOPS 的最大比率為 0.25。
- 在 RDS for SQL Server 上，IOPS 與已配置儲存體 (以 GiB 為單位) 的最小比率為 0.5。其他支援的資料庫引擎沒有最小比率。
- 對於所有支援的資料庫引擎，IOPS 與已配置儲存體的最大比率為 500。
- 如果您使用的是儲存體自動調整規模，則 IOPS 和儲存體閾值上限之間的比率 (以 GiB 為單位) 同樣適用。

如需儲存體自動調整規模的詳細資訊，請參閱 [使用 Amazon RDS 儲存體自動擴展自動管理容量](#)。

gp2 儲存裝置 (前一代產品)

當您的應用程式不需要高儲存效能時，您可以使用一般用途 SSD gp2 儲存體。gp2 儲存體的基準 I/O 效能為每 GiB 有 3 IOPS，下限為 100 IOPS。此關係表示磁碟區越大，效能越好。例如，一個 100 GiB 磁碟區的基準效能為 300 IOPS。一個 1,000 TiB 磁碟區的基準效能為 3,000 IOPS。

大小低於 1,000 GiB 的個別 gp2 磁碟區也擁有在一段延伸時間內爆增至 3,000 IOPS 的能力。磁碟區 I/O 額度餘額決定了爆量效能。有關基準效能和 I/O 積分平衡如何影響效能的詳細說明，請參閱 AWS 資料庫部落格 [了解 Amazon RDS 和 gp2 的突發效能與基準效能](#) 一文。

許多工作負載絕不會耗盡爆量餘額。然而，有些工作負載可能用盡 3,000 IOPS 的爆量儲存額度餘額，因此您應該規劃符合工作負載需求的儲存容量。

對於大於 4,000 GiB 的 gp2 磁碟區，基準效能會大於突發效能。對於這類磁碟區而言，爆量不相關，因為基準效能優於 3,000 IOPS 爆量效能。不過，對於特定引擎和大小的資料庫執行個體，儲存體會跨四個磁碟區進行分割，這會提供四倍的基準輸送量，以及單一磁碟的四倍爆量 IOPS。

Amazon RDS 資料庫引擎上各種儲存大小的 gp2 磁碟區的儲存效能如下表所示。

資料庫引擎	RDS 儲存區大小	基準 IOPS 的範圍	基準輸送量範圍	爆量 IOPS
MariaDB、MySQL 和 PostgreSQL	5—399 GiB ¹	100-1197 IOPS	128-250 MiB/秒	3,000
MariaDB、MySQL 和 PostgreSQL	千兆 GiB	1,200-4,005 IOPS	500-1,000 MiB/秒	12,000
MariaDB、MySQL 和 PostgreSQL	1,336—3,999 GiB	4008-11,997 IOPS	1,000 MiB/s	12,000
MariaDB、MySQL 和 PostgreSQL	4,000—65,536 GiB	12,000-64,000 IOPS	1,000 MiB/s	N/A ²
Oracle	兆 GiB	100-597 IOPS	128-250 MiB/秒	3,000
Oracle	千兆 GiB	600-4,005 IOPS	500-1,000 MiB/秒	12,000
Oracle	1,336—3,999 GiB	4008-11,997 IOPS	1,000 MiB/s	12,000
Oracle	4,000—65,536 GiB	12,000-64,000 IOPS	1,000 MiB/s	N/A ²
SQL Server	GiB	100-999 IOPS	128-250 MiB/秒	3,000
SQL Server	334—999 GiB	1,002-2,997 IOPS	250 MiB/s	3,000
SQL Server	1,000—16,384 GiB	3,000-16,000 IOPS	250 MiB/s	N/A ²

Note

¹ 使用 AWS Management Console，您可以針對 db.t3.micro 和 db.t4g.micro 資料庫執行個體類別，在免費方案中建立儲存大小下限為 5 GiB 的資料庫執行個體。否則，最小存儲大小為 20 GiB。此限制不適用於 AWS CLI 和 RDS API。

² 音量的基準性能超過最大突發性能。

比較固態硬碟 (SSD) 儲存體類型

下表顯示 Amazon RDS 所使用之 SSD 儲存磁碟區的使用案例和效能特性。

特性	已佈建 IOPS (io2 區塊快速)	佈建 IOPS (io1)	一般用途 (gp3)	一般用途 (gp2)
描述	RDS 儲存產品組合中的最高效能 (IOPS、輸送量、延遲) 專為對延遲敏感的交易工作負載而設計	一致的儲存體效能 (IOPS、輸送量、延遲) 專為對延遲敏感的交易工作負載而設計	獨立佈建儲存體、IOPS 和輸送量的彈性 平衡各種交易工作負載的價格效能。	提供爆量 IOPS 平衡各種交易工作負載的價格效能。
使用案例	需要低於一毫秒延遲的業務關鍵交易工作負載，以及高達 256,000 IOPS 的持續 IOPS 效能	需要持續 IOPS 效能的交易工作負載，最高可達 256,000 IOPS	在開發/測試環境中於中型關聯式資料庫上執行的廣泛工作負載	在開發/測試環境中於中型關聯式資料庫上執行的廣泛工作負載
Latency (延遲)	低於一毫秒，持續提供 99.9% 的時間	個位數毫秒，在 99.9% 的時間內持續提供	個位數毫秒，在 99% 的時間內持續提供	個位數毫秒，在 99% 的時間內持續提供
磁碟區大小	百分之六十 GiB	適用於 SQL 伺服器的 RDS 上的 20—16,384 GiB)	20—65,536 GiB (適用於 SQL 伺	20—65,536 GiB (適用於 SQL 伺

特性	已佈建 IOPS (io2 區塊快速)	佈建 IOPS (io1)	一般用途 (gp3)	一般用途 (gp2)
			服务器的 RDS 上的 16,384 GiB)	服务器的 RDS 上的 16,384 GiB)
最大 IOPS	256,000	256,000 (RDS for SQL Server 上 64,000)	64,000 (RDS for SQL Server 上 16,000)	64,000 (RDS for SQL Server 上 16,000)

Note

您無法直接在 gp2 儲存體上佈建 IOPS。IOPS 會隨配置的儲存體大小而有所不同。

特性	已佈建 IOPS (io2 區塊快速)	佈建 IOPS (io1)	一般用途 (gp3)	一般用途 (gp2)
最大輸送量	<p>根據佈建 IOPS 進行擴展，最高 4,000 MB/s</p> <p>輸送量可按比例擴展至每個佈建 IOPS 0.256 MiB/s。具有 16 KiB I/O 大小的 256,000 IOPS，以及 16,000 個 IOPS 或更高的處理能力 (使用 256 KiB I/O 大小)，最大輸送量可達到 4,000 MiB/ 秒。</p> <p>對於不以 AWS 硝基系統為基礎的執行個體，在 16 KiB I/O 大小的情況下，可以在 128,000 IOPS 下達到 2,000 MB 的最大輸送量。</p>	<p>根據佈建 IOPS 進行擴展，最高 4,000 MB/s</p>	<p>佈建最高 4,000 MB/秒的額外輸送量，其中 1000 MB/秒在 RDS for SQL Server 上</p>	<p>1000 MB/s (RDS for SQL Server 上 250 MB/s)</p>
AWS CLI 和 RDS API 名稱	io2	io1	gp3	gp2

磁性儲存裝置 (舊版，不建議使用)

Amazon RDS 也支援磁帶儲存，以提供回溯相容性。建議您針對任何新的儲存需求，使用一般用途 SSD 或佈建 IOPS SSD。下列為磁性儲存體的一些限制：

- 不允許在使用 SQL Server 資料庫引擎時擴展儲存體。
- 不允許您在使用 SQL Server 資料庫引擎時轉換為不同的儲存類型。
- 不支援儲存體自動擴展。
- 不支援彈性磁碟區。
- 限制大小為 3 TiB 以下。
- 限制為 1,000 IOPS 以下。

專用記錄檔磁碟區 (DLV)

您可以使用 Amazon RDS 主控台或 Amazon RDS API，AWS CLI 針對使用佈建 IOPS (PIOPS) 儲存的資料庫執行個體使用專用日誌磁碟區 (DLV)。DLV 會將 PostgreSQL 資料庫交易記錄檔和 MySQL/MariaDB 重做記錄檔和二進位記錄移至儲存磁碟區 (與包含資料庫表格的磁碟區分開)。DLV 使交易寫入記錄更有效率且一致。DLV 非常適合具有大型配置儲存體、每秒高 I/O (IOPS) 需求或延遲敏感工作負載的資料庫。

DLV 可用於 PIOPS 儲存體 (io1 和 io2 區塊快速)，而且建立的固定大小為 1,000 GiB 和 3,000 個佈建 IOPS。

Note

一般用途儲存裝置 (gp2 和 gp3) 不支援 DLV。

Amazon RDS 支援以下版本 AWS 區域 的所有 DLV：

- MariaDB 10.6.7 和更新的 10 版本
- MySQL 8.0.28 和更新的 8.版本
- PostgreSQL 14.7 及更高版本，14 個或更高版本，15.2 及更高版本 15 個版本，以及 16.1 及更高版本 16 個版本

RDS 支援 DLV 搭配多可用區部署。修改或建立多可用區執行個體時，會同時為主要和次要執行個體建立 DLV。

RDS 支援 DLV 搭配僅供讀取複本。如果主要資料庫執行個體已啟用 DLV，則在啟用 DLV 之後建立的所有僅供讀取複本也將具有 DLV。在切換到 DLV 之前建立的任何僅供讀取複本將不會啟用它，除非明

確修改為要這樣做。建議在啟用 DLV 之前附加至主要執行個體的所有僅供讀取複本，您也須將其手動修改為具有 DLV。

在修改資料庫執行個體的 DLV 設定之後，必須重新啟動資料庫執行個體。

如需啟用 DLV 的資訊，請參閱[使用專用日誌磁碟區 \(DLV\)](#)。

監控儲存體效能

Amazon RDS 提供了您可以用來判斷資料庫執行個體執行情形的數種指標。您可以在 Amazon RDS 管理主控台中檢視執行個體摘要頁面中的指標。您也可以使用 Amazon CloudWatch 監控這些指標。如需詳細資訊，請參閱 [在 Amazon RDS 主控台中檢視指標](#)。強化的監控提供了更詳細的輸入/輸出指標，如需更多資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

以下指標在監控資料庫執行個體的儲存體時很有幫助：

- IOPS – 每秒完成的輸入/輸出操作數量。此指標會報告為指定時間間隔的平均 IOPS。Amazon RDS 每隔 1 分鐘分別報告讀取和寫入 IOPS。合計 IOPS 是讀取與寫入 IOPS 的加總。一般而言，IOPS 範圍的數值在每秒 0 至數萬之間。
- 延遲 – 在輸入/輸出請求提交和完成之間的經過時間。此指標會報告為指定時間間隔的平均延遲。Amazon RDS 每隔一分鐘分別回報讀取與寫入延遲。一般而言，延遲數值以毫秒 (ms) 計算。
- 傳輸量 – 每秒從磁碟傳出或傳入的位元組。此指標會報告為指定時間間隔的平均輸送量。Amazon RDS 使用每秒位元組單位 (B/s)，以 1 分鐘的間隔分別報告讀取和寫入輸送量。一般而言，傳輸量數值的範圍介於零到輸入/輸出通道的最大頻寬。
- 佇列深度 – 佇列中待處理的輸入/輸出請求數量。這些是已由應用程式提交的輸入/輸出請求，但尚未傳送至裝置，因為該裝置正在忙於處理其他輸入/輸出請求。佇列中耗費的等待時間為延遲和服務時間的元件 (不做為指標)。此指標會報告為指定時間間隔的平均佇列深度。Amazon RDS 每隔 1 分鐘一次報告佇列深度。一般而言，佇列深度的數值在 0 至數百之間。

測量的 IOPS 數值獨立於單獨輸入/輸出操作的大小之外。這代表當您測量輸入/輸出效能時，除了輸入/輸出操作數量之外，您也確定留意執行個體的輸送量。

影響儲存體效能的因素

系統活動、資料庫工作負載，和資料庫執行個體類別都會影響儲存體效能。

系統活動

下列系統相關的活動會消耗輸入/輸出容量，並可能會在處理時降低資料庫執行個體效能：

- 多可用區域待命建立
- 僅供讀取複本的建立
- 變更儲存類型

資料庫工作負載

在某些狀況下，您的資料庫或應用程式設計會導致並行數量問題、鎖定或其他形式的資料庫爭用。在這些情況中，您可能無法直接使用所有佈建的頻寬。此外，您有可能會遇到下列與工作負載相關的狀況：

- 達到基礎執行個體類型的傳輸量上限。
- 應用程式未驅動足夠的輸入/輸出操作，因此佇列深度一直小於 1。
- 即使部分輸入/輸出容量處於未使用的狀態，資料庫仍發生查詢爭用情形。

在某些狀況下，並無系統資源達到或接近上限，且新增執行緒數量也不會增加資料庫交易處理速率。於此狀況下，瓶頸很可能是資料庫中的爭用。最常見的問題即為資料列鎖定和索引頁鎖定爭用，但還有許多其他可能原因。若遇到這種情況，您應該向資料庫效能調校專家尋求建議。

DB instance class (資料庫執行個體類別)

為了讓 Amazon RDS 資料庫執行個體發揮最大效能，請選擇這一代執行個體類型，其應具有足夠頻寬能支援您的儲存體類型。例如，您可選擇 Amazon EBS 最佳化執行個體，以及擁有 10 GB 網路連線能力的執行個體。

Important

視您使用的執行個體類別而定，您可能會看到比 RDS 可佈建上限還要低的 IOPS 效能。如需有關資料庫執行個體類別之 IOPS 效能的特定資訊，請參閱《Amazon EC2 使用者手冊》中的 [Amazon EBS 最佳化執行個體](#)。建議您先決定執行個體類別的 IOPS 上限，然後再為資料庫執行個體設定佈建 IOPS 值。

建議您使用最新一代的執行個體，才能獲得最佳效能。上一代資料庫執行個體的最大儲存量可能較低。

某些較舊的 32 位元檔案系統的儲存容量可能較低。若要判斷資料庫執行個體的儲存容量，您可以使用 [描述驗證的 DB 執行個體修改命](#) AWS CLI 令。

下表顯示各資料庫執行個體類別的儲存上限可擴展至各資料庫引擎。

- DB2 — 64 TiB
- MariaDB – 64 TiB
- Microsoft SQL 服務器-64 次
- MySQL – 64 TiB
- Oracle – 64 TiB
- PostgreSQL – 64 TiB

下表顯示一些例外狀況的最大儲存空間 (以 TiB 為單位)。除了 io2 區塊快速儲存體之外，所有適用於 Microsoft SQL 伺服器的 RDS 資料庫執行個體都有 16 TiB 的最大儲存體，因此沒有 SQL 伺服器的項目。

執行個體類別	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.m3 – 標準執行個體類別					
db.t4g – 爆量效能執行個體類別					
db.t4g.medium	N/A	16	16	N/A	32
db.t4g.small	N/A	16	16	N/A	16
db.t4g.micro	N/A	6	6	N/A	6
db.t3 – 爆量效能執行個體類別					
db.t3.medium	32	16	16	32	32
db.t3.small	32	16	16	32	16
db.t3.micro	N/A	6	6	32	6
db.t2 – 爆量效能執行個體類別					

如需所有支援執行個體類別的詳細資訊，請參閱[上一代資料庫執行個體](#)。

區域、可用區域和 Local Zones

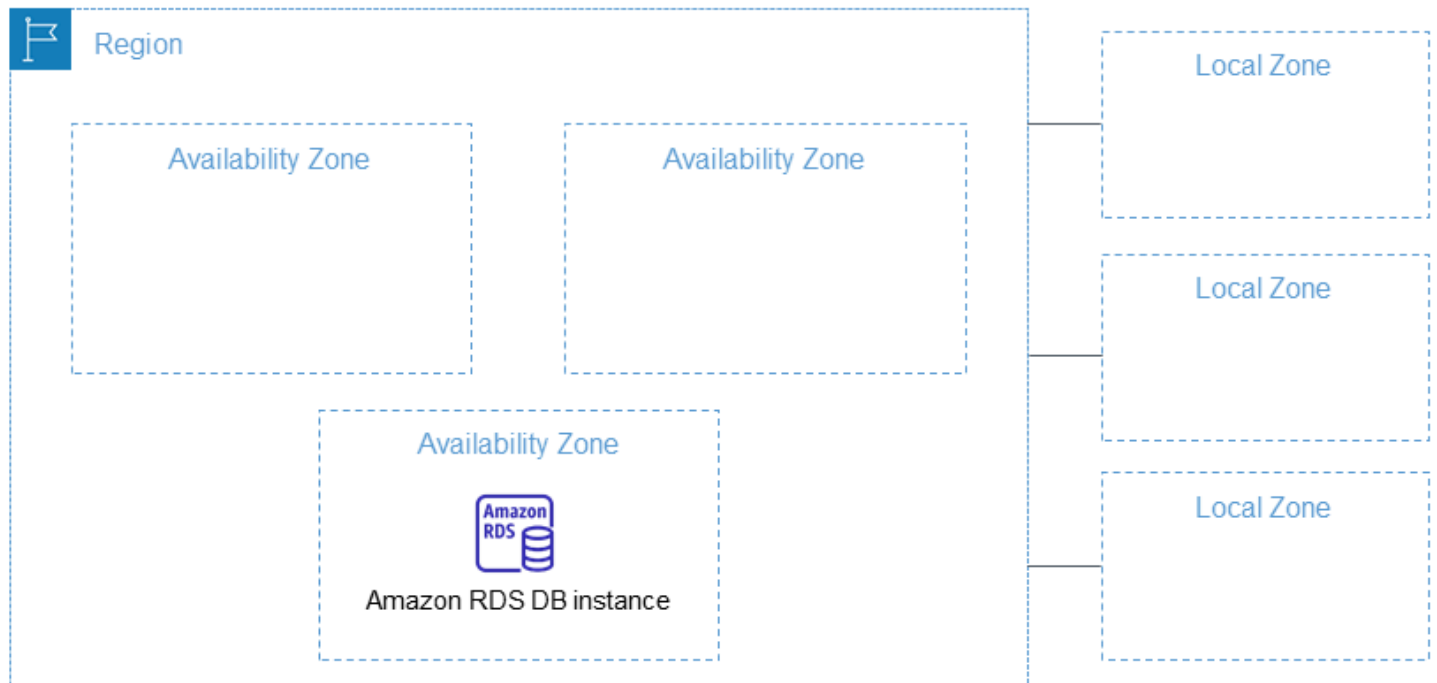
Amazon 雲端運算資源託管於全球的多個地點。這些位置由區 AWS 域、可用區域和 Local Zones 組成。各個 AWS 區域為獨立的地理區域。每個 AWS 區域都有多個隔離的位置，稱為可用區域。

Note

如需尋找區域可用區域的相關資訊，請參閱 Amazon EC2 文件中的[說明您的可用區域](#)。AWS

使用 Local Zones 可讓您將資源 (例如運算和儲存) 放在多個更靠近使用者的位置。Amazon RDS 讓您可將資源 (例如資料庫執行個體) 和資料放在多個位置。除非您特別這麼做，否則資源不會跨 AWS 區域複寫。

Amazon 運營 state-of-the-art 的高可用性數據中心。儘管故障極為少見，但仍可能影響相同位置內資料庫執行個體的可用性。若您將所有資料庫執行個體都託管於某一位置，一旦該位置受故障影響，所有資料庫執行個體都將無法使用。



重要的是要記住，每個 AWS 區域都是完全獨立的。您啟動的任何 Amazon RDS 活動 (例如，建立資料庫執行個體或列出可用的資料庫執行個體) 只會在您目前的預設 AWS 區域執行。可以在控制台中更改默認 AWS 區域，也可以通過設置 `AWS_DEFAULT_REGION` 環境變量來更改。或者，它可以通過使用 `--region` 參數與 AWS Command Line Interface (AWS CLI) 來覆蓋。如需詳細資訊，請參閱[設定 AWS Command Line Interface](#)，特別是有關環境變數和命令列選項的章節。

Amazon RDS 支持稱為特殊 AWS 區域 AWS GovCloud (US)。這些區域旨在讓美國政府機構和客戶將更敏感的工作負載移到雲端。AWS GovCloud (US) 區域處理美國政府的特定法律與法規要求。如需詳細資訊，請參閱[什麼是 AWS GovCloud \(US\) ?](#)

若要在特定區域中建立或使用 Amazon RDS 資料庫執行個體，請使用對應的 AWS 區域服務端點。

AWS 地區

每個 AWS 區域都設計為與其他區 AWS 域隔離。此設計可達到最高的容錯能力與穩定性。

當您檢視資源時，您只會看到與您指定的 AWS 區域相關聯的資源。這是因為 AWS 區域彼此隔離，而且我們不會自動跨 AWS 區域複寫資源。

區域可用性

下表顯示目前提供 Amazon RDS 的區 AWS 域以及每個區域的端點。

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
		rds-fips.us-east-2.api.aws	HTTPS
		rds.us-east-2.api.aws	HTTPS
		rds-fips.us-east-2.amazonaws.com	HTTPS
美國東部 (維吉尼亞 北部)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
美國西部 (加利佛尼 亞北部)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
美國西部 (奧勒岡)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
非洲 (開 普敦)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
亞太區域 (香港)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS
亞太區域 (海德拉 巴)	ap-south-2	rds.ap-south-2.amazonaws.com	HTTPS
		rds.ap-south-2.api.aws	HTTPS
亞太區域 (雅加達)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com	HTTPS
		rds.ap-southeast-3.api.aws	HTTPS
亞太區域 (墨爾本)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com	HTTPS
		rds.ap-southeast-4.api.aws	HTTPS
亞太區域 (孟買)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS
		rds.ap-south-1.api.aws	HTTPS
亞太區域 (大阪)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS
		rds.ap-northeast-3.api.aws	HTTPS
亞太區域 (首爾)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS
		rds.ap-northeast-2.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
亞太區域 (新加坡)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
		rds.ap-southeast-1.api.aws	HTTPS
亞太區域 (雪梨)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
		rds.ap-southeast-2.api.aws	HTTPS
亞太區域 (東京)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
加拿大 (中部)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
加拿大西部 (卡加利)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
歐洲 (法蘭克福)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
歐洲 (愛爾蘭)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
歐洲 (倫敦)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
歐洲 (米蘭)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
歐洲 (巴黎)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
歐洲 (西班牙)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
歐洲 (斯德哥爾摩)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
歐洲 (蘇黎世)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
以色列 (特拉維夫)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
中東 (巴林)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS
中東 (阿拉伯聯合大公國)	me-central-1	rds.me-central-1.amazonaws.com	HTTPS
		rds.me-central-1.api.aws	HTTPS
南美洲 (聖保羅)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
		rds.sa-east-1.api.aws	HTTPS
AWS GovCloud (美國東部)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS
		rds.us-gov-east-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
AWS GovCloud (美國西部)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS
		rds.us-gov-west-1.api.aws	HTTPS

如果您沒有明確指定端點，預設值是美國西部 (奧勒岡) 端點。

使用 AWS CLI 或 API 操作使用資料庫執行個體時，請確定已指定其區域端點。

可用區域

建立資料庫執行個體時，您可以選擇可用區域，或讓 Amazon RDS 為您隨機選擇一個可用區域。可用區域由區 AWS 域代碼後面接著字母識別碼 (例如，us-east-1a) 來表示。

使用 [describe-availability-zones](#) Amazon EC2 命令，描述為您的帳戶啟用的指定區域內的可用區域，如下所示。

```
aws ec2 describe-availability-zones --region region-name
```

例如，若要描述美國東部 (維吉尼亞北部) 區域 (us-east-1) 內針對您帳戶啟用的可用區域，請執行下列命令：

```
aws ec2 describe-availability-zones --region us-east-1
```

您無法在多可用區域資料庫部署中，為主要和次要資料庫執行個體選擇可用區域。Amazon RDS 會為您進行隨機選擇。如需異地同步備份部署的詳細資訊，請參閱[設定及管理多可用區部署](#)。

Note

RDS 隨機選擇可用區域，並不保證單一帳戶或資料庫子網路群組中可用區域之間的資料庫執行個體均勻分佈。您可以在建立或修改單一可用區域執行個體時請求特定可用區域，也可以將更多特定的資料庫子網路群組用於異地同步備份執行個體。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#) 及 [修改 Amazon RDS 資料庫執行個體](#)。

本機區域

本地區域是地理位置靠近您的用戶的 AWS 區域的擴展。您可以將來自上層 AWS 區域的 VPC 延伸至本機區域。若要這樣做，請建立新的子網路，並將其指派給 AWS 本機區域。當您在本機區域中建立子網路時，VPC 也會擴展到該本機區域。本機區域中的子網路與 VPC 中的其他子網路的運作方式相同。

建立資料庫執行個體時，可以選擇本機區域的子網路。Local Zones 有自己的網際網路連線，並支援 AWS Direct Connect。因此，在本機區域中建立的資源可以為本機使用者提供非常低延遲的通訊服務。如需詳細資訊，請參閱 [AWS Local Zones](#)。

本地區域由區 AWS 域代碼表示，後面接著指示位置的標識符，例如 us-west-2-lax-1a。

Note

本機區域無法包含在異地同步備份部署中。

使用本機區域

1. 在 Amazon EC2 主控台中啟用本機區域。

如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [啟用本地區域](#)。

2. 在本機區域中建立子網路。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [在您的 VPC 中建立子網路](#)。

3. 在本機區域中建立資料庫子網路群組。

建立資料庫子網路群組時，請選擇本機區域的可用區域群組。

如需詳細資訊，請參閱 [在 VPC 中建立資料庫執行個體](#)。

4. 在本機區域中建立使用資料庫子網路群組的資料庫執行個體。

如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

Important

目前，唯一提供 Amazon RDS 的 AWS 本地區域是美國西部 (奧勒岡) 區域的洛杉磯。

Amazon RDS 由 AWS 區域 和數據庫引擎支持的功能

對 Amazon RDS 功能和選項的 Support 會因每個資料庫引擎的特定版本而有所不同。AWS 區域 若要識別 AWS 區域指定的 RDS DB 引擎版本支援和可用性，您可以使用下列區段。

Amazon RDS 功能與引擎原生功能和選項不同。如需引擎原生功能和選項的詳細資訊，請參閱[引擎原生功能](#)。

支援的地區和資料庫引擎

- [資料表慣例](#)
- [功能快速參考](#)
- [支援 Amazon RDS 藍色/綠色部署的區域和資料庫引擎](#)
- [支援的區域和資料庫引擎，可在 Amazon RDS 中進行跨區域自動備份](#)
- [Amazon RDS 中跨區域僅供讀取複本支援的區域和資料庫引擎](#)
- [Amazon RDS 中資料庫活動串流支援的區域和資料庫引擎](#)
- [Amazon RDS 中支援雙堆疊模式的區域和資料庫引擎](#)
- [支援的區域和資料庫引擎，可將快照匯出到 Amazon RDS 中的 S3](#)
- [Amazon RDS 中的 IAM 資料庫身份驗證支援的區域和資料庫引擎](#)
- [Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎](#)
- [Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎](#)
- [支援的區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights](#)
- [RDS 自訂支援的區域和資料庫引擎](#)
- [支援 Amazon RDS 代理伺服器的區域和資料庫引擎](#)
- [Secrets Manager 與 Amazon RDS 整合的支援區域和資料庫引擎](#)
- [支援的區域和資料庫引擎，適用於 Amazon RDS 零 ETL 與 Amazon Redshift 整合](#)
- [Amazon RDS 中的引擎原生功能](#)

資料表慣例

功能區段中的資料表會使用這些模式來指定版本號碼和可用性層級：

- x.y 版 – 僅特定版本可用。
- x.y 版和更新版本 – 此特定版本及其主要版本所有更新的次要版本均受支援。例如，「10.11 版和更新版本」表示 10.11、10.11.1 和 10.12 版可用。

- - 該功能目前不適用於選取的 RDS 資料庫引擎，也無法在指定的 AWS 區域中使用。

功能快速參考

下列快速參考資料表列出每個功能和可用的 RDS DB 引擎。區域和特定版本可用性會顯示在後面的功能區段中。

功能	適用於 Db2 的 RDS	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
藍/綠部署	—	可用性	可用性	—	可用性	—
跨區域自動備份	可用性	可用性	可用性	可用性	可用性	可用性
跨區域僅供讀取複本	—	可用性	可用性	可用性	可用性	可用性
資料庫活動串流	—	—	—	可用性	—	可用性
雙堆疊模式	—	可用性	可用性	可用性	可用性	可用性
將快照匯出至	—	可用性	可用性	—	可用性	—

功能	適用於 Db2 的 RDS	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
Amazon S3						
AWS Identity and Access Management (IAM) 資料庫身份驗證	–	可用性	可用性	–	可用性	–
Kerberos 身份驗證	可用性	–	可用性	可用性	可用性	可用性
多可用區域資料庫叢集	–	–	可用性	–	可用性	–
Performance Insight	–	可用性	可用性	可用性	可用性	可用性
RDS Custom	–	–	–	可用性	–	可用性
RDS Proxy	–	可用性	可用性	–	可用性	可用性

功能	適用於 Db2 的 RDS	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
Secret Manager 整合	可用性	可用性	可用性	可用性	可用性	可用性

支援 Amazon RDS 藍色/綠色部署的區域和資料庫引擎

藍/綠部署會在個別已同步的預備環境中複製生產資料庫環境。透過使用 Amazon RDS 藍/綠部署，您可以在預備環境中對資料庫進行變更，而不會影響生產環境。例如，您可以升級主要或次要資料庫引擎版本、變更資料庫參數，或在預備環境中進行結構描述變更。備妥後，您可以將預備環境提升為新的生產資料庫環境。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

所 AWS 區域有人都支援藍/綠部署功能。

下列引擎支援藍/綠部署功能：

- RDS for MariaDB 10.2 版和更新版本
- RDS for MySQL 5.7 版和更新版本
- RDS for MySQL 8.0.15 版和更新版本
- RDS for PostgreSQL 11.21 版及更新版本
- RDS for PostgreSQL 12.16 版及更新版本
- RDS for PostgreSQL 13.12 版及更新版本
- RDS for PostgreSQL 14.9 版及更新版本
- RDS for PostgreSQL 15.4 版及更新版本
- 適用於 PostgreSQL 本 16.1 及更高版本的 RDS

不支援藍/綠部署功能與下列引擎搭配：

- 適用於 Db2 的 RDS
- RDS for SQL Server
- RDS for Oracle

支援的區域和資料庫引擎，可在 Amazon RDS 中進行跨區域自動備份

透過 Amazon RDS 中的備份複本，您可以設定 RDS 資料庫執行個體，將快照和交易記錄複寫到目的地區域。為資料庫執行個體設定備份複寫時，RDS 會在所有快照和交易日誌準備就緒時，立即啟動其跨區域複製作業。如需詳細資訊，請參閱 [將自動備份複製到另一個 AWS 區域](#)。

Backup 複製可在所有項目中使用，AWS 區域 但下列項目除外

- 非洲 (開普敦)
- 亞太區域 (香港)
- 亞太區域 (海德拉巴)
- 亞太區域 (雅加達)
- 歐洲 (米蘭)
- 歐洲 (西班牙)
- 歐洲 (蘇黎世)
- Middle East (Bahrain)
- 中東 (阿拉伯聯合大公國)

如需來源備份和目的地備份區域限制的詳細資訊，請參閱 [將自動備份複製到另一個 AWS 區域](#)。

主題

- [使用適用於 Db2 的 RDS Backup 複寫](#)
- [Backup 複寫搭配 RDS for MariaDB](#)
- [Backup 複寫搭配 RDS for MySQL](#)
- [Backup 複寫搭配 RDS for Oracle](#)
- [使用 RDS for PostgreSQL 搭配備份複本](#)
- [使用 RDS for SQL Server 搭配備份複本](#)

使用適用於 Db2 的 RDS Backup 複寫

Amazon RDS 支援所有目前可用的適用於 Db2 的 RDS 版本的備份複寫。

Backup 複寫搭配 RDS for MariaDB

Amazon RDS 支援可對所有可用版本的 RDS for MariaDB 進行備份複寫。

Backup 複寫搭配 RDS for MySQL

Amazon RDS 支援可對所有可用版本的 RDS for MySQL 進行備份複寫。

Backup 複寫搭配 RDS for Oracle

Amazon RDS 支援可對所有可用版本的 RDS for Oracle 進行備份複寫。

使用 RDS for PostgreSQL 搭配備份複本

Amazon RDS 支援可對所有可用版本的 RDS for PostgreSQL 進行備份複寫。

使用 RDS for SQL Server 搭配備份複本

Amazon RDS 支援可對所有可用版本的 RDS for SQL Server 進行備份複寫。

Amazon RDS 中跨區域僅供讀取複本支援的區域和資料庫引擎

在 Amazon RDS 使用跨區域讀寫複本時，您可以在從與來源資料庫執行個體不同的區域中建立 MariaDB、MySQL、Oracle、PostgreSQL 或 SQL Server 僅供讀取複本。如需跨區域僅供讀取複本的詳細資訊，包括來源和目的地區域考量，請參閱 [以不同的方式建立僅供讀取複本 AWS 區域](#)。

下列引擎無法使用跨區域僅供讀取複本：

- 適用於 Db2 的 RDS

主題

- [搭配 RDS for MariaDB 的跨區域僅供讀取複本](#)
- [搭配 RDS for MySQL 的跨區域僅供讀取複本](#)
- [搭配 RDS for Oracle 的跨區域僅供讀取複本](#)
- [搭配 RDS for PostgreSQL 的跨區域僅供讀取複本](#)
- [搭配 RDS for SQL Server 的跨區域僅供讀取複本](#)

搭配 RDS for MariaDB 的跨區域僅供讀取複本

若為下列版本，所有區域僅都可使用搭配 RDS for MariaDB 的跨區域僅供讀取複本：

- RDS for MariaDB 10.11 (所有可用版本)
- RDS for MariaDB 10.6 (所有可用版本)

- RDS for MariaDB 10.5 (所有可用版本)
- RDS for MariaDB 10.4 (所有可用版本)
- RDS for MariaDB 10.3 (所有可用版本)

搭配 RDS for MySQL 的跨區域僅供讀取複本

若為下列版本，所有區域僅都可使用搭配 RDS for MySQL 的跨區域僅供讀取複本：

- RDS for MySQL 8.0 (所有可用版本)
- RDS for MySQL 5.7 (所有可用版本)

搭配 RDS for Oracle 的跨區域僅供讀取複本

適用於 Oracle 的 RDS 跨區域僅供讀取複本提供給所有 AWS 區域支援的資料庫版本 (使用企業版)。僅在非 CDB 和 CDB 架構的單一租用戶組態中支援複本。CDB 架構的多租用戶組態不支援跨區域僅供讀取複本。

如需有關適 RDS for Oracle 跨區域僅供讀取複本額外需求的詳細資訊，請參閱 [RDS for Oracle 複本的要求與考量](#)。

搭配 RDS for PostgreSQL 的跨區域僅供讀取複本

若為下列版本，所有區域僅都可使用搭配 RDS for PostgreSQL 的跨區域僅供讀取複本：

- RDS 專用 PostgreSQL 有可用的版本)
- RDS for PostgreSQL 15 (所有可用版本)
- RDS for PostgreSQL 14 (所有可用版本)
- RDS for PostgreSQL 13 (所有可用版本)
- RDS for PostgreSQL 12 (所有可用版本)
- RDS for PostgreSQL 11 (所有可用版本)
- RDS for PostgreSQL 10 (所有可用版本)

搭配 RDS for SQL Server 的跨區域僅供讀取複本

若為下列版本，除了下列區域以外的所有區域僅都可使用搭配 RDS for SQL Server 的跨區域僅供讀取複本：

- 非洲 (開普敦)
- 亞太區域 (香港)
- 亞太區域 (海德拉巴)
- 亞太區域 (雅加達)
- 亞太區域 (墨爾本)
- 加拿大西部 (卡加利)
- 歐洲 (米蘭)
- 歐洲 (西班牙)
- 歐洲 (蘇黎世)
- 以色列 (特拉維夫)
- Middle East (Bahrain)
- 中東 (阿拉伯聯合大公國)

搭配 RDS for SQL Server 的跨區域僅供讀取複本適用於下列使用 Microsoft SQL Server Enterprise Edition 的版本：

- SQL 伺服器專用 RDS
- RDS for SQL Server 2019 (版本 15.00.4073.23 及更新版本)
- RDS for SQL Server 2017 (版本 14.00.3281.6 及更新版本)
- RDS for SQL Server 2016 (版本 13.00.6300.2 及更新版本)

Amazon RDS 中資料庫活動串流支援的區域和資料庫引擎

透過使用 Amazon RDS 中的資料庫活動串流，您可以監控並設定 Oracle 資料庫和 SQL Server 資料庫中稽核活動的警示。如需詳細資訊，請參閱 [資料庫活動串流概觀](#)。

資料庫活動串流不適用於下列引擎：

- 適用於 Db2 的 RDS
- RDS for MariaDB
- RDS for MySQL
- RDS for PostgreSQL

主題

- [搭配 RDS for Oracle 的資料庫活動串流](#)
- [搭配 RDS for SQL Server 的資料庫活動串流](#)

搭配 RDS for Oracle 的資料庫活動串流

下列區域和引擎版本可供搭配 RDS for Oracle 的資料庫活動串流使用。

如需使用 RDS for Oracle 使用進行資料庫活動串流額外要求的詳細資訊，請參閱 [資料庫活動串流概觀](#)。

區域	RDS for Oracle 21c	RDS for Oracle 19c
美國東部 (俄亥俄)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
美國東部 (維吉尼亞北部)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
美國西部 (加利佛尼亞北部)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
美國西部 (奧勒岡)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
非洲 (開普敦)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (香港)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)

區域	RDS for Oracle 21c	RDS for Oracle 19c
亞太區域 (海德拉巴)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (雅加達)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (墨爾本)	–	–
亞太區域 (孟買)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (大阪)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (首爾)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (新加坡)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (悉尼)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
亞太區域 (東京)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
加拿大 (中部)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)

區域	RDS for Oracle 21c	RDS for Oracle 19c
加拿大西部 (卡加利)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
中國 (北京)	–	–
中國 (寧夏)	–	–
歐洲 (法蘭克福)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (愛爾蘭)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (倫敦)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (米蘭)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
Europe (Paris)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (西班牙)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (斯德哥爾摩)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
歐洲 (蘇黎世)	–	–

區域	RDS for Oracle 21c	RDS for Oracle 19c
亞太區域 (墨爾本)	–	–
Middle East (Bahrain)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
中東 (阿拉伯聯合大公國)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
南美洲 (聖保羅)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 和更高版本，使用企業版 (EE) 或標準版 2 (SE2)
AWS GovCloud (美國東部)	–	–
AWS GovCloud (美國西部)	–	–

搭配 RDS for SQL Server 的資料庫活動串流

下列區域和引擎版本可供搭配 RDS for SQL Server 的資料庫活動串流使用。

如需使用 RDS for SQL Server 使用進行資料庫活動串流額外要求的詳細資訊，請參閱 [資料庫活動串流概觀](#)。

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	–
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	–

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	–
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	–
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (墨爾本)	–	–	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	–
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	–
加拿大西部 (卡加利)	所有可用版本	所有可用版本	所有可用版本	–
中國 (北京)	–	–	–	–

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
中國 (寧夏)	–	–	–	–
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	–
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (蘇黎世)	–	–	–	–
以色列 (特拉維夫)	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	–
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	–
AWS GovCloud (美國東部)	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–

Amazon RDS 中支援雙堆疊模式的區域和資料庫引擎

透過使用 RDS 中的雙堆疊模式，資源可透過網際網路通訊協定第 4 版 (IPv4)、網際網路通訊協定第 6 版 (IPv6) 或兩者與資料庫執行個體進行通訊。如需詳細資訊，請參閱 [雙堆疊模式](#)。

主題

- [具備適用於 Db2 之 RDS 的雙堆疊模式](#)
- [搭配 RDS for MariaDB 的雙堆疊模式](#)
- [搭配 RDS for MySQL 的雙堆疊模式](#)
- [搭配 RDS for Oracle 的雙堆疊模式](#)
- [搭配 RDS for PostgreSQL 的雙堆疊模式](#)
- [搭配 RDS for SQL Server 的雙堆疊模式](#)

具備適用於 Db2 之 RDS 的雙堆疊模式

以下區域和引擎版本適用於 RDS for Db2 的雙堆疊模式。

區域	適用於 Db2 11.5 的 RDS				
美國東部 (俄亥俄)	所有可用版本				
美國東部 (維吉尼亞北部)	所有可用版本				
美國西部 (加利佛尼亞北部)	所有可用版本				
美國西部 (奧勒岡)	所有可用版本				
非洲 (開普敦)	所有可用版本				
亞太區域 (香港)	所有可用版本				

區域	適用於 Db2 11.5 的 RDS				
亞太區域 (海德拉巴)	所有可用版本				
亞太區域 (雅加達)	所有可用版本				
亞太區域 (墨爾本)	所有可用版本				
亞太區域 (孟買)	所有可用版本				
亞太區域 (大阪)	所有可用版本				
亞太區域 (首爾)	所有可用版本				
亞太區域 (新加坡)	所有可用版本				
亞太區域 (悉尼)	所有可用版本				
亞太區域 (東京)	所有可用版本				
加拿大 (中部)	所有可用版本				
加拿大西部 (卡加利)	–				
中國 (北京)	–				
中國 (寧夏)	–				

區域	適用於 Db2 11.5 的 RDS				
歐洲 (法蘭克福)	所有可用版本				
歐洲 (愛爾蘭)	所有可用版本				
歐洲 (倫敦)	所有可用版本				
歐洲 (米蘭)	所有可用版本				
Europe (Paris)	所有可用版本				
歐洲 (西班牙)	所有可用版本				
歐洲 (斯德哥爾摩)	所有可用版本				
歐洲 (蘇黎世)	所有可用版本				
以色列 (特拉維夫)	–				
Middle East (Bahrain)	所有可用版本				
中東 (阿拉伯聯合大公國)	所有可用版本				
南美洲 (聖保羅)	所有可用版本				
AWS GovCloud (美國東部)	–				

區域	適用於 Db2 11.5 的 RDS				
AWS GovCloud (美國西部)	–				

搭配 RDS for MariaDB 的雙堆疊模式

下列區域和引擎版本可供搭配 RDS for MariaDB 的雙堆疊模式使用。

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
亞太區域 (墨爾本)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–	–	–	–
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
以色列 (特拉維夫)	–	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國西部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

搭配 RDS for MySQL 的雙堆疊模式

下列區域和引擎版本可供搭配 RDS for MySQL 的雙堆疊模式使用。

區域	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	–
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本	–
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–	–
中國 (北京)	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	–
以色列 (特拉維夫)	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國西部)	所有可用版本	所有可用版本	所有可用版本

搭配 RDS for Oracle 的雙堆疊模式

下列區域和引擎版本可供搭配 RDS for Oracle 的雙堆疊模式使用。

區域	RDS for Oracle 21c	RDS for Oracle 19c
美國東部 (俄亥俄)	所有可用版本	所有可用版本

區域	RDS for Oracle 21c	RDS for Oracle 19c
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	–	–
亞太區域 (雅加達)	所有可用版本	所有可用版本
亞太區域 (墨爾本)	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–
中國 (北京)	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本

區域	RDS for Oracle 21c	RDS for Oracle 19c
歐洲 (倫敦)	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本
歐洲 (西班牙)	–	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本
歐洲 (蘇黎世)	–	–
以色列 (特拉維夫)	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	所有可用版本	所有可用版本
AWS GovCloud (美國西部)	所有可用版本	所有可用版本

搭配 RDS for PostgreSQL 的雙堆疊模式

下列區域和引擎版本可供搭配 RDS for PostgreSQL 的雙堆疊模式使用。

區域	適用於 PostgreSQL L 16 的 RDS	RDS for PostgreSQL L 15	RDS for PostgreSQL L 14	RDS for PostgreSQL L 13	RDS for PostgreSQL L 12	RDS for PostgreSQL L 11	RDS for PostgreSQL L 10
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–	–	–	–	–	–
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
以色列 (特拉維夫)	–	–	–	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AWS GovCloud (美國西部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

搭配 RDS for SQL Server 的雙堆疊模式

下列區域和引擎版本可供搭配 RDS for SQL Server 的雙堆疊模式使用。

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	–
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	–
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	–
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	–
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (海德拉巴)	–	–	–	–
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	–

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
亞太區域 (墨爾本)	–	–	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	–
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	–
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	–
加拿大西部 (卡加利)	–	–	–	–
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	–
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	–
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (西班牙)	–	–	–	–

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	–
歐洲 (蘇黎世)	–	–	–	–
以色列 (特拉維夫)	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	–
中東 (阿拉伯聯合大公國)	–	–	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	–
AWS GovCloud (美國東部)	所有可用版本	所有可用版本	所有可用版本	–
AWS GovCloud (美國西部)	所有可用版本	所有可用版本	所有可用版本	–

支援的區域和資料庫引擎，可將快照匯出到 Amazon RDS 中的 S3

您可以將 RDS 資料庫快照資料匯出至 Amazon S3 儲存貯體。您可以匯出所有資料庫快照類型，包括手動快照、自動化系統快照，以及 AWS Backup 建立的快照。匯出資料後，您可以直接透過 Amazon Athena 或 Amazon Redshift Spectrum 等工具分析匯出後的資料。如需詳細資訊，請參閱 [將資料庫快照資料匯出至 Amazon S3](#)。

下列引擎無法將快照匯出至 S3：

- 適用於 Db2 的 RDS
- RDS for Oracle
- RDS for SQL Server

主題

- [使用 RDS for Maria DB 將快照匯出到 S3](#)
- [使用 RDS for MySQL 將快照匯出到 S3](#)
- [使用 RDS for PostgreSQL 將快照匯出到 S3](#)

使用 RDS for Maria DB 將快照匯出到 S3

下列區域和引擎版本可用於將快照匯出至搭配 RDS for MariaDB 的 S3。

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	–	–	–	–	–
亞太區域 (雅加達)	–	–	–	–	–

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
亞太區域 (墨爾本)	–	–	–	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	–	–	–	–	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	–	–	–	–	–
以色列 (特拉維夫)	–	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	–	–	–	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–	–

使用 RDS for MySQL 將快照匯出到 S3

下列區域和引擎版本可用於將快照匯出至搭配 RDS for MySQL 的 S3。

區域	RDS for MySQL 8.0	RDS for MySQL 5.7
美國東部 (俄亥俄)	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	–	–
亞太區域 (雅加達)	–	–
亞太區域 (墨爾本)	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–
中國 (北京)	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7
歐洲 (愛爾蘭)	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本
歐洲 (西班牙)	–	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本
歐洲 (蘇黎世)	–	–
以色列 (特拉維夫)	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–
AWS GovCloud (美國西部)	–	–

使用 RDS for PostgreSQL 將快照匯出到 S3

下列區域和引擎版本可用於將快照匯出至搭配 RDS for PostgreSQL 的 S3。

區域	適用於 PostgreSQL L 16 的 RDS	RDS for PostgreSQL L 15	RDS for PostgreSQL L 14	RDS for PostgreSQL L 13	RDS for PostgreSQL L 12	RDS for PostgreSQL L 11	RDS for PostgreSQL L 10
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	–	–	–	–	–	–	–
亞太區域 (雅加達)	–	–	–	–	–	–	–
亞太區域 (墨爾本)	–	–	–	–	–	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	–	–	–	–	–	–	–
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	–	–	–	–	–	–	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	–	–	–	–	–	–	–
以色列 (特拉維夫)	–	–	–	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	–	–	–	–	–	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–	–	–	–	–	–

區域	適用於 PostgreSQL L 16 的 RDS	RDS for PostgreSQL L 15	RDS for PostgreSQL L 14	RDS for PostgreSQL L 13	RDS for PostgreSQL L 12	RDS for PostgreSQL L 11	RDS for PostgreSQL L 10
AWS GovCloud (美國西 部)	–	–	–	–	–	–	–

Amazon RDS 中的 IAM 資料庫身份驗證支援的區域和資料庫引擎

透過在 Amazon RDS 中使用 IAM 資料庫身份驗證，您可以在連線至資料庫執行個體時不需要密碼進行身份驗證。而是改用身份驗證字符。如需詳細資訊，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身份驗證](#)。

下列資料庫引擎不可使用 IAM 資料庫身份驗證：

- 適用於 Db2 的 RDS
- RDS for Oracle
- RDS for SQL Server

主題

- [IAM 資料庫身份驗證搭配 RDS for MariaDB](#)
- [具有 RDS for MySQL 的 IAM 資料庫身份驗證](#)
- [適用於 RDS for PostgreSQL 的 IAM 資料庫身份驗證](#)

IAM 資料庫身份驗證搭配 RDS for MariaDB

下列區域和引擎版本可與 RDS for MariaDB 搭配使用進行 IAM 資料庫身份驗證。

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
美國東部 (俄亥俄)	所有可用版本	所有可用版本	–	–	–
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	–	–	–
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	–	–	–
美國西部 (奧勒岡)	所有可用版本	所有可用版本	–	–	–
非洲 (開普敦)	所有可用版本	所有可用版本	–	–	–
亞太區域 (香港)	所有可用版本	所有可用版本	–	–	–
亞太區域 (海德拉巴)	–	–	–	–	–
亞太區域 (雅加達)	所有可用版本	所有可用版本	–	–	–
亞太區域 (墨爾本)	–	–	–	–	–
亞太區域 (孟買)	所有可用版本	所有可用版本	–	–	–
亞太區域 (大阪)	所有可用版本	所有可用版本	–	–	–
亞太區域 (首爾)	所有可用版本	所有可用版本	–	–	–

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
亞太區域 (新加坡)	所有可用版本	所有可用版本	–	–	–
亞太區域 (悉尼)	所有可用版本	所有可用版本	–	–	–
亞太區域 (東京)	所有可用版本	所有可用版本	–	–	–
加拿大 (中部)	所有可用版本	所有可用版本	–	–	–
加拿大西部 (卡加利)	所有可用版本	所有可用版本	–	–	–
中國 (北京)	所有可用版本	所有可用版本	–	–	–
中國 (寧夏)	所有可用版本	所有可用版本	–	–	–
歐洲 (法蘭克福)	所有可用版本	所有可用版本	–	–	–
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	–	–	–
歐洲 (倫敦)	所有可用版本	所有可用版本	–	–	–
歐洲 (米蘭)	所有可用版本	所有可用版本	–	–	–
Europe (Paris)	所有可用版本	所有可用版本	–	–	–
歐洲 (西班牙)	–	–	–	–	–
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	–	–	–
歐洲 (蘇黎世)	–	–	–	–	–

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
以色列 (特拉維夫)	–	–	–	–	–
Middle East (Bahrain)	所有可用版本	所有可用版本	–	–	–
中東 (阿拉伯聯合大公國)	–	–	–	–	–
南美洲 (聖保羅)	所有可用版本	所有可用版本	–	–	–
AWS GovCloud (美國東部)	所有可用版本	所有可用版本	–	–	–
AWS GovCloud (美國西部)	所有可用版本	所有可用版本	–	–	–

具有 RDS for MySQL 的 IAM 資料庫身分驗證

若為下列版本，所有區域都可以使用 RDS for MySQL 進行 IAM 資料庫身分驗證：

- RDS for MySQL 8.0 - 所有可用版本
- RDS for MySQL 5.7 - 所有可用版本

適用於 RDS for PostgreSQL 的 IAM 資料庫身分驗證

若為下列版本，所有區域都可以使用 RDS for PostgreSQL 進行 IAM 資料庫身分驗證：

- RDS 適用 PostgreSQL — 所有可用的版本
- RDS for PostgreSQL 15 - 所有可用版本
- RDS for PostgreSQL 14 - 所有可用版本

- RDS for PostgreSQL 13 - 所有可用版本
- RDS for PostgreSQL 12 - 所有可用版本
- RDS for PostgreSQL 11 - 所有可用版本
- RDS for PostgreSQL 10 - 所有可用版本

Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎

透過在 Amazon RDS 中使用 Kerberos 身分驗證，您可以支援 Kerberos 和 Microsoft Active Directory 之資料庫使用者的外部身分驗證。使用 Kerberos 和 Active Directory，提供了資料庫使用者的單一登入和集中式身分驗證優點。

下列引擎無法使用 Kerberos 身分驗證：

- RDS for MariaDB

雖然您的 AWS 帳戶預設會啟用大部分 AWS 區域，但只有在您手動選取某些地區時，才會啟用這些區域。這些區域稱為「選擇加入區域」。相比之下，一旦您的 AWS 帳戶創建，默認情況下處於活動狀態的區域被稱為商業區域，或者簡稱為區域。對於選擇加入區域，您必須使用表單的區域化服務主體。例如，`directoryservice.rds.region_name.amazonaws.com`，對於非洲 (開普敦)，您必須將服務主體新增為 `directoryservice.rds.region-af-south-1.amazonaws.com` 至您的信任政策。如需詳細資訊，請參閱 [Kerberos 身分驗證](#)。

主題

- [使用適用於 Db2 的 RDS 的 Kerberos 驗證](#)
- [Kerberos 身分驗證搭配 RDS for MySQL](#)
- [將 Kerberos 身分驗證搭配 RDS for Oracle](#)
- [使用 RDS for PostgreSQL 的 Kerberos 身分驗證](#)
- [使用 RDS for SQL Server 的 Kerberos 身分驗證](#)

使用適用於 Db2 的 RDS 的 Kerberos 驗證

下列區域和引擎版本適用於 Db2 的 RDS 驗證可用於 Kerberos 驗證。

區域	適用於 Db2 11.5 的 RDS
美國東部 (俄亥俄)	所有版本
美國東部 (維吉尼亞北部)	所有版本
美國西部 (加利佛尼亞北部)	所有版本
美國西部 (奧勒岡)	所有版本
非洲 (開普敦)	–
亞太區域 (香港)	–
亞太區域 (海德拉巴)	–
亞太區域 (雅加達)	–
亞太區域 (墨爾本)	–
亞太區域 (孟買)	所有版本
亞太區域 (大阪)	–
亞太區域 (首爾)	所有版本
亞太區域 (新加坡)	所有版本
亞太區域 (悉尼)	所有版本
亞太區域 (東京)	所有版本
加拿大 (中部)	所有版本
加拿大西部 (卡加利)	–
中國 (北京)	所有版本
中國 (寧夏)	所有版本
歐洲 (法蘭克福)	所有版本

區域	適用於 Db2 11.5 的 RDS
歐洲 (愛爾蘭)	所有版本
歐洲 (倫敦)	所有版本
歐洲 (米蘭)	–
Europe (Paris)	–
歐洲 (西班牙)	–
歐洲 (斯德哥爾摩)	所有版本
歐洲 (蘇黎世)	–
以色列 (特拉維夫)	–
Middle East (Bahrain)	–
中東 (阿拉伯聯合大公國)	–
南美洲 (聖保羅)	所有版本
AWS GovCloud (美國東部)	–
AWS GovCloud (美國西部)	–

Kerberos 身分驗證搭配 RDS for MySQL

下列區域和引擎版本可與 RDS for MySQL 搭配使用進行 Kerberos 身分驗證。

區域	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
美國東部 (俄亥俄)	所有版本	所有版本	所有版本
美國東部 (維吉尼亞北部)	所有版本	所有版本	所有版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
美國西部 (加利佛尼亞北部)	所有版本	所有版本	所有版本
美國西部 (奧勒岡)	所有版本	所有版本	所有版本
非洲 (開普敦)	–	–	–
亞太區域 (香港)	–	–	–
亞太區域 (海德拉巴)	–	–	–
亞太區域 (雅加達)	–	–	–
亞太區域 (墨爾本)	–	–	–
亞太區域 (孟買)	所有版本	所有版本	所有版本
亞太區域 (大阪)	–	–	–
亞太區域 (首爾)	所有版本	所有版本	所有版本
亞太區域 (新加坡)	所有版本	所有版本	所有版本
亞太區域 (悉尼)	所有版本	所有版本	所有版本
亞太區域 (東京)	所有版本	所有版本	所有版本
加拿大 (中部)	所有版本	所有版本	所有版本
加拿大西部 (卡加利)	–	–	–
中國 (北京)	所有版本	所有版本	所有版本
中國 (寧夏)	所有版本	所有版本	所有版本
歐洲 (法蘭克福)	所有版本	所有版本	所有版本
歐洲 (愛爾蘭)	所有版本	所有版本	所有版本
歐洲 (倫敦)	所有版本	所有版本	所有版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
歐洲 (米蘭)	–	–	–
Europe (Paris)	–	–	–
歐洲 (西班牙)	–	–	–
歐洲 (斯德哥爾摩)	所有版本	所有版本	所有版本
歐洲 (蘇黎世)	–	–	–
以色列 (特拉維夫)	–	–	–
Middle East (Bahrain)	–	–	–
中東 (阿拉伯聯合大公國)	–	–	–
南美洲 (聖保羅)	所有版本	所有版本	所有版本
AWS GovCloud (美國東部)	–	–	–
AWS GovCloud (美國西部)	–	–	–

將 Kerberos 身分驗證搭配 RDS for Oracle

下列區域和引擎版本可與 RDS for Oracle 搭配使用進行 Kerberos 身分驗證。

區域	RDS for Oracle 21c	RDS for Oracle 19c
美國東部 (俄亥俄)	所有版本	所有版本
美國東部 (維吉尼亞北部)	所有版本	所有版本
美國西部 (加利佛尼亞北部)	所有版本	所有版本
美國西部 (奧勒岡)	所有版本	所有版本

區域	RDS for Oracle 21c	RDS for Oracle 19c
非洲 (開普敦) (選擇加入地區)	所有版本	所有版本
亞太區域 (香港) (選擇加入地區)	所有版本	所有版本
亞太區域 (海德拉巴) (選擇加入地區)	所有版本	所有版本
亞太區域 (雅加達) (選擇加入地區)	所有版本	所有版本
亞太區域 (墨爾本) (選擇加入地區)	所有版本	所有版本
亞太區域 (孟買)	所有版本	所有版本
亞太區域 (大阪)	–	–
亞太區域 (首爾)	所有版本	所有版本
亞太區域 (新加坡)	所有版本	所有版本
亞太區域 (悉尼)	所有版本	所有版本
亞太區域 (東京)	所有版本	所有版本
加拿大 (中部)	所有版本	所有版本
加拿大西部 (卡加利)	–	–
中國 (北京)	–	–
中國 (寧夏)	–	–
歐洲 (法蘭克福)	所有版本	所有版本
歐洲 (愛爾蘭)	所有版本	所有版本

區域	RDS for Oracle 21c	RDS for Oracle 19c
歐洲 (倫敦)	所有版本	所有版本
歐洲 (米蘭) (選擇加入地區)	所有版本	所有版本
Europe (Paris)	–	–
歐洲 (西班牙) (選擇加入地區)	所有版本	所有版本
歐洲 (斯德哥爾摩)	所有版本	所有版本
歐洲 (蘇黎世) (選擇加入地區)	所有版本	所有版本
以色列 (特拉維夫) (選擇加入地區)	所有版本	所有版本
中東 (巴林) (選擇加入地區)	所有版本	所有版本
中東 (阿拉伯聯合大公國) (選擇加入地區)	所有版本	所有版本
南美洲 (聖保羅)	所有版本	所有版本
AWS GovCloud (美國東部)	所有版本	所有版本
AWS GovCloud (美國西部)	所有版本	所有版本

使用 RDS for PostgreSQL 的 Kerberos 身分驗證

下列區域和引擎版本可與 RDS for PostgreSQL 搭配使用進行 Kerberos 身分驗證。

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
美國東部 (俄亥俄)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
美國東部 (維吉尼亞北部)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
美國西部 (加利佛尼亞北部)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
美國西部 (奧勒岡)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
非洲 (開普敦)	–	–	–	–	–	–	–
亞太區域 (香港)	–	–	–	–	–	–	–
亞太區域 (海德拉巴)	–	–	–	–	–	–	–
亞太區域 (雅加達)	–	–	–	–	–	–	–
亞太區域 (墨爾本)	–	–	–	–	–	–	–
亞太區域 (孟買)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
亞太區域 (大阪)	–	–	–	–	–	–	–
亞太區域 (首爾)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (新加坡)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (悉尼)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (東京)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
加拿大 (中部)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
加拿大西部 (卡加利)	–	–	–	–	–	–	–
中國 (北京)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
中國 (寧夏)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (法蘭克福)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (愛爾蘭)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
歐洲 (倫敦)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (米蘭)	–	–	–	–	–	–	–
Europe (Paris)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (西班牙)	–	–	–	–	–	–	–
歐洲 (斯德哥爾摩)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (蘇黎世)	–	–	–	–	–	–	–
以色列 (特拉維夫)	–	–	–	–	–	–	–
Middle East (Bahrain)	–	–	–	–	–	–	–
中東 (阿拉伯聯合大公國)	–	–	–	–	–	–	–
南美洲 (聖保羅)	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本	所有版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AWS GovCloud (美國東部)	–	–	–	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–	–	–	–

使用 RDS for SQL Server 的 Kerberos 身分驗證

下列區域和引擎版本可與 RDS for SQL Server 搭配使用進行 Kerberos 身分驗證。

區域	SQL 伺服器專用 RDS	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
美國東部 (俄亥俄)	所有版本	所有版本	所有版本	所有版本	所有版本
美國東部 (維吉尼亞北部)	所有版本	所有版本	所有版本	所有版本	所有版本
美國西部 (加利佛尼亞北部)	所有版本	所有版本	所有版本	所有版本	所有版本
美國西部 (奧勒岡)	所有版本	所有版本	所有版本	所有版本	所有版本
非洲 (開普敦)	所有版本	所有版本	所有版本	所有版本	所有版本

區域	SQL 伺服器 專用 RDS	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
亞太區域 (香港)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (海德拉巴)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (雅加達)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (墨爾本)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (孟買)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (大阪)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (首爾)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (新加坡)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (悉尼)	所有版本	所有版本	所有版本	所有版本	所有版本
亞太區域 (東京)	所有版本	所有版本	所有版本	所有版本	所有版本
加拿大 (中部)	所有版本	所有版本	所有版本	所有版本	所有版本
加拿大西部 (卡加利)	–	–	–	–	–
中國 (北京)	所有版本	所有版本	所有版本	所有版本	所有版本

區域	SQL 伺服器 專用 RDS	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
中國 (寧夏)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (法蘭克 福)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (愛爾蘭)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (倫敦)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (米蘭)	所有版本	所有版本	所有版本	所有版本	所有版本
Europe (Paris)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (西班牙)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (斯德哥 爾摩)	所有版本	所有版本	所有版本	所有版本	所有版本
歐洲 (蘇黎世)	所有版本	所有版本	所有版本	所有版本	所有版本
以色列 (特拉 維夫)	–	–	–	–	–
Middle East (Bahrain)	所有版本	所有版本	所有版本	所有版本	所有版本
中東 (阿拉伯 聯合大公國)	所有版本	所有版本	所有版本	所有版本	所有版本
南美洲 (聖保 羅)	所有版本	所有版本	所有版本	所有版本	所有版本
AWS GovCloud (美 國東部)	所有版本	所有版本	所有版本	所有版本	所有版本

區域	SQL 伺服器專用 RDS	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AWS GovCloud (美國西部)	所有版本	所有版本	所有版本	所有版本	所有版本

Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎

Amazon RDS 提供的多可用區域資料庫叢集部署是 Amazon RDS 的高可用性部署模式，具有兩個可讀取待命資料庫執行個體。多可用區域資料庫叢集具有寫入器資料庫執行個體和兩個讀取器資料庫執行個體，位於相同區域中的三個個別可用區域中。多可用區域資料庫叢集相較於多可用區域資料庫執行個體部署，可提供高可用性、增加讀取工作負載的容量，以及降低寫入延遲。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

多可用區域資料庫叢集不適用於下列引擎：

- 適用於 Db2 的 RDS
- RDS for MariaDB
- RDS for Oracle
- RDS for SQL Server

主題

- [使用 RDS for MySQL 的多可用區域資料庫叢集](#)
- [RDS for PostgreSQL 搭配多可用區域資料庫叢集](#)

使用 RDS for MySQL 的多可用區域資料庫叢集

下列區域和引擎版本可供搭配 RDS for MySQL 的多可用區域資料庫叢集使用。

區域	RDS for MySQL 8.0
美國東部 (俄亥俄)	8.0.28 版和更新版本
美國東部 (維吉尼亞北部)	8.0.28 版和更新版本

區域	RDS for MySQL 8.0
美國西部 (加利佛尼亞北部)	–
美國西部 (奧勒岡)	8.0.28 版和更新版本
非洲 (開普敦)	8.0.28 版和更新版本
亞太區域 (香港)	8.0.28 版和更新版本
亞太區域 (海德拉巴)	8.0.28 版和更新版本
亞太區域 (雅加達)	8.0.28 版和更新版本
亞太區域 (墨爾本)	8.0.28 版和更新版本
亞太區域 (孟買)	8.0.28 版和更新版本
亞太區域 (大阪)	8.0.28 版和更新版本
亞太區域 (首爾)	8.0.28 版和更新版本
亞太區域 (新加坡)	8.0.28 版和更新版本
亞太區域 (悉尼)	8.0.28 版和更新版本
亞太區域 (東京)	8.0.28 版和更新版本
加拿大 (中部)	8.0.28 版和更新版本
加拿大 (中部)	8.0.28 版和更新版本
加拿大西部 (卡加利)	8.0.28 版和更新版本
中國 (北京)	8.0.28 版和更新版本
中國 (寧夏)	8.0.28 版和更新版本
歐洲 (法蘭克福)	8.0.28 版和更新版本
歐洲 (愛爾蘭)	8.0.28 版和更新版本

區域	RDS for MySQL 8.0
歐洲 (倫敦)	8.0.28 版和更新版本
歐洲 (米蘭)	8.0.28 版和更新版本
Europe (Paris)	8.0.28 版和更新版本
歐洲 (西班牙)	8.0.28 版和更新版本
歐洲 (斯德哥爾摩)	8.0.28 版和更新版本
歐洲 (蘇黎世)	8.0.28 版和更新版本
以色列 (特拉維夫)	8.0.28 版和更新版本
Middle East (Bahrain)	8.0.28 版和更新版本
中東 (阿拉伯聯合大公國)	8.0.28 版和更新版本
南美洲 (聖保羅)	8.0.28 版和更新版本
AWS GovCloud (美國東部)	–
AWS GovCloud (美國西部)	–

您可以使用 AWS CLI 變更資料庫執行個體類別，以顯示其可用的引擎版本。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options \
--engine mysql \
--db-instance-class db.r5d.large \
--query '*[]|?[?SupportsClusters == `true`].[EngineVersion]' \
--output text
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options ^
--engine mysql ^
--db-instance-class db.r5d.large ^
```



```
--query "*"[]|[]?SupportsClusters == `true`].[EngineVersion]" ^
--output text
```

RDS for PostgreSQL 搭配多可用區域資料庫叢集

下列區域和引擎版本可供搭配 RDS for PostgreSQL 的多可用區域資料庫叢集使用。

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
美國東部 (俄亥俄)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
美國東部 (維吉尼亞北部)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
美國西部 (加利佛尼亞北部)	–	–	–	–
美國西部 (奧勒岡)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
非洲 (開普敦)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (香港)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (海德拉巴)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (雅加達)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (墨爾本)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (孟買)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
亞太區域 (大阪)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (首爾)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (新加坡)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (悉尼)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
亞太區域 (東京)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
加拿大 (中部)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
加拿大西部 (卡加利)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
中國 (北京)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
中國 (寧夏)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (法蘭克福)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (愛爾蘭)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (倫敦)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
歐洲 (米蘭)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
Europe (Paris)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (西班牙)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (斯德哥爾摩)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
歐洲 (蘇黎世)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
以色列 (特拉維夫)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
Middle East (Bahrain)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
中東 (阿拉伯聯合大公國)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
南美洲 (聖保羅)	PostgreSQL 本	所有 PostgreSQL 15 版本	14.5 版和更新版本	13.4 版和 13.7 版以及更新版本
AWS GovCloud (美國東部)	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–

您可以使用 AWS CLI 變更資料庫執行個體類別，以顯示其可用的引擎版本。

對於LinuxmacOS、或Unix：

```
aws rds describe-orderable-db-instance-options \  
--engine postgres \  
--db-instance-class db.r5d.large \  
--query '*[?SupportsClusters == `true`].[EngineVersion]' \  
--output text
```

在 Windows 中：

```
aws rds describe-orderable-db-instance-options ^  
--engine postgres ^  
--db-instance-class db.r5d.large ^  
--query "*[?SupportsClusters == `true`].[EngineVersion]" ^  
--output text
```

支援的區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights

Amazon RDS 中的績效詳情會擴展現有監控功能，以說明並協助您分析資料庫效能叢集效能。使用績效詳情儀表板，您可以將 Amazon RDS 資料庫執行個體上的資料庫負載視覺化。您也可以按照等待、SQL 陳述式、託管或使用者篩選負載。如需詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。

Performance Insights 適用於所有 RDS 資料庫引擎，除了適用於 Db2 的 RDS 之外。

對於可用的資料庫引擎，Performance Insights 適用於所有可用的引擎版本和所有版本 AWS 區域。

如需 Performance Insights 見功能的區域、資料庫引擎和執行個體類別支援資訊，請參閱 [Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別](#)。

RDS 自訂支援的區域和資料庫引擎

Amazon RDS Custom 會自動化資料庫管理任務和操作。透過使用 RDS Custom，您可以讓您以資料庫管理員身分存取和自訂資料庫環境和作業系統。使用 RDS Custom，您可以自訂以符合舊版、自訂和封裝應用程式的需求。如需詳細資訊，請參閱 [使用 Amazon RDS Custom](#)。

僅下列資料庫引擎支援 RDS Custom：

主題

- [適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)
- [適用於 SQL 伺服器的 RDS 自訂支援的區域和資料庫引擎](#)

適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎

下列區域和引擎版本可供 RDS Custom for Oracle 使用。

區域	Oracle 資料庫 19c	Oracle 資料庫 18c	Oracle 資料庫 12c
美國東部 (俄亥俄)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
美國東部 (維吉尼亞北部)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
美國西部 (加利佛尼亞北部)	–	–	–
美國西部 (奧勒岡)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
非洲 (開普敦)	–	–	–
Asia Pacific (Hong Kong)	–	–	–
亞太區域 (雅加達)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
亞太區域 (墨爾本)	–	–	–
亞太區域 (孟買)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2

區域	Oracle 資料庫 19c	Oracle 資料庫 18c	Oracle 資料庫 12c
亞太區域 (大阪)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
亞太區域 (首爾)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
亞太區域 (新加坡)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
亞太區域 (悉尼)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
亞太區域 (東京)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
加拿大 (中部)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
加拿大西部 (卡加利)	–	–	–
中國 (北京)	–	–	–
中國 (寧夏)	–	–	–
歐洲 (法蘭克福)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2

區域	Oracle 資料庫 19c	Oracle 資料庫 18c	Oracle 資料庫 12c
歐洲 (愛爾蘭)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
歐洲 (倫敦)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
歐洲 (米蘭)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
Europe (Paris)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
歐洲 (斯德哥爾摩)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
以色列 (特拉維夫)	–	–	–
Middle East (Bahrain)	–	–	–
中東 (阿拉伯聯合大公國)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
南美洲 (聖保羅)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2
AWS GovCloud (美國東部)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2

區域	Oracle 資料庫 19c	Oracle 資料庫 18c	Oracle 資料庫 12c
AWS GovCloud (美國西部)	2021 年 1 月或更高 RU/RUR 的 19c	2021 年 1 月或更高 RU/RUR 的 18c	2021 年 1 月或更高 RU/RUR 的 12.1 和 12.2

適用於 SQL 伺服器的 RDS 自訂支援的區域和資料庫引擎

您可以使用 RDS 提供的引擎版本 (RPEV) 或自訂引擎版本 (CEV) 來部署 RDS Custom for SQL Server：

- 如果您使用 RPEV，它會包含預設的 Amazon Machine Image (AMI) 和 SQL Server 安裝。如果您自訂或修改作業系統 (OS)，您的變更可能不會在修補程式、快照還原或自動復原期間持續存在。
- 如果您使用 CEV，則應選擇自己的 AMI，其可搭配預先安裝的 Microsoft SQL Server 或由您使用自有媒體安裝的 SQL Server。使用 AWS 提供的 CEV 時，您可以選擇最新的可用 Amazon EC2 映像 (AMI) AWS，該映像具有 SQL 伺服器的 RDS 自訂支援的累積更新 (CU)。您可以自訂作業系統和 SQL Server 的組態，以符合您的企業需求。

下列版本 AWS 區域 和資料庫引擎版本可用於 SQL 伺服器的 RDS 自訂。引擎版本的支援情形取決於是否搭配 RPEV、AWS 提供 CEV 或客戶提供 CEV 來使用 RDS Custom for SQL Server。

區域	RPEV	AWS 提供 CEV	客戶提供的 CEV
美國東部 (俄亥俄)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
美國東部 (維吉尼亞北部)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24

區域	RPEV	AWS 提供 CEV	客戶提供的 CEV
美國西部 (加利佛尼亞北部)	–	–	–
美國西部 (奧勒岡)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
非洲 (開普敦)	–	–	–
亞太區域 (香港)	–	–	–
亞太區域 (海德拉巴)	–	–	–
亞太區域 (雅加達)	–	–	–
亞太區域 (墨爾本)	–	–	–
亞太區域 (孟買)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
亞太區域 (大阪)	–	–	–
亞太區域 (首爾)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24

區域	RPEV	AWS 提供 CEV	客戶提供的 CEV
亞太區域 (新加坡)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
亞太區域 (悉尼)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
亞太區域 (東京)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
加拿大 (中部)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
加拿大西部 (卡加利)	–	–	–
中國 (北京)	–	–	–
中國 (寧夏)	–	–	–

區域	RPEV	AWS 提供 CEV	客戶提供的 CEV
歐洲 (法蘭克福)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
歐洲 (愛爾蘭)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
歐洲 (倫敦)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
歐洲 (米蘭)	–	–	–
Europe (Paris)	–	–	–
歐洲 (西班牙)	–	–	–
歐洲 (斯德哥爾摩)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 年企業版、標準或開發人員，搭配 CU17、18、CU24
歐洲 (蘇黎世)	–	–	–
以色列 (特拉維夫)	–	–	–

區域	RPEV	AWS 提供 CEV	客戶提供的 CEV
Middle East (Bahrain)	–	–	–
中東 (阿拉伯聯合大公國)	–	–	–
南美洲 (聖保羅)	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU8、CU17、	SQL 伺服器 2022 企業版、標準版或網頁版，搭配 CU9。SQL 伺服器 2019 年企業版、標準或網路，搭配 CU17、中 18、CU24	SQL 伺服器 2022 企業版、標準或開發人員，與 CU9。SQL 伺服器 2019 企業版、標準或開發人員，搭配 CU17、18、CU24
AWS GovCloud (美國東部)	–	–	–
AWS GovCloud (美國西部)	–	–	–

支援 Amazon RDS 代理伺服器的區域和資料庫引擎

Amazon RDS Proxy 是完全受管、高可用性的資料庫 Proxy，可透過共用建立的資料庫連線，讓應用程式更具可擴充性。如需詳細資訊，請參閱 [使用 Amazon RDS Proxy](#)。

RDS 代理伺服器不適用於下列引擎：

- 適用於 Db2 的 RDS
- RDS for Oracle

主題

- [RDS Proxy 具有 RDS for MariaDB](#)
- [RDS Proxy 具有 RDS for MySQL](#)
- [RDS Proxy 具有 RDS for PostgreSQL](#)
- [搭配 RDS for SQL Server 的 RDS Proxy](#)

RDS Proxy 具有 RDS for MariaDB

下列區域和引擎版本可供搭配 RDS for MariaDB 的 RDS Proxy 使用。

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
以色列 (特拉維夫)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–	–

RDS Proxy 具有 RDS for MySQL

下列區域和引擎版本可供搭配 RDS for MySQL 的 RDS Proxy 使用。

區域	RDS for MySQL 8.0	RDS for MySQL 5.7
美國東部 (俄亥俄)	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本
亞太區域 (悉尼)	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本
加拿大西部 (卡加利)	所有可用版本	所有可用版本
中國 (北京)	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本

區域	RDS for MySQL 8.0	RDS for MySQL 5.7
歐洲 (米蘭)	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本
以色列 (特拉維夫)	所有可用版本	所有可用版本
Middle East (Bahrain)	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本
南美洲 (聖保羅)	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–
AWS GovCloud (美國西部)	–	–

RDS Proxy 具有 RDS for PostgreSQL

下列區域和引擎版本可供搭配 RDS for PostgreSQL 的 RDS Proxy 使用。

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	適用於 PostgreSQL 16 的 RDS	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
以色列 (特拉維夫)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–	–	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–	–	–	–

搭配 RDS for SQL Server 的 RDS Proxy

下列區域和引擎版本可供搭配 RDS for SQL Server 的 RDS Proxy 使用。

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
美國東部 (俄亥俄)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國東部 (維吉尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (加利佛尼亞北部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
美國西部 (奧勒岡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
非洲 (開普敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (香港)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (海德拉巴)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (雅加達)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (墨爾本)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (孟買)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (大阪)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (首爾)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (新加坡)	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
亞太區域 (悉尼)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
亞太區域 (東京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大 (中部)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
加拿大西部 (卡加利)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (北京)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中國 (寧夏)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (法蘭克福)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (愛爾蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (倫敦)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (米蘭)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Europe (Paris)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (西班牙)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (斯德哥爾摩)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
歐洲 (蘇黎世)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
以色列 (特拉維夫)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
Middle East (Bahrain)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
中東 (阿拉伯聯合大公國)	所有可用版本	所有可用版本	所有可用版本	所有可用版本

區域	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
南美洲 (聖保羅)	所有可用版本	所有可用版本	所有可用版本	所有可用版本
AWS GovCloud (美國東部)	–	–	–	–
AWS GovCloud (美國西部)	–	–	–	–

Secrets Manager 與 Amazon RDS 整合的支援區域和資料庫引擎

使用 AWS Secrets Manager，您可以透過 API 呼叫 Secrets Manager 來取代程式碼中的硬式編碼認證 (包括資料庫密碼)，以程式設計方式擷取密碼。如需 Secrets Manager 的詳細資訊，請參閱 [AWS Secrets Manager 使用者指南](#)。

您可以指定 Amazon RDS 在 Secrets Manager 中管理 Amazon RDS 資料庫執行個體或多可用區域資料庫叢集的主要使用者密碼。RDS 會產生密碼、將其存放在 Secrets Manager 中，並定期對其進行輪換。如需詳細資訊，請參閱 [使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager](#)。

所有 RDS 資料庫引擎和所有版本都支援 Secrets Manager。

除了以下內容 AWS 區域 之外，所有支援 Secrets Manager 整合：

- 加拿大西部 (卡加利)
- AWS GovCloud (美國東部)
- AWS GovCloud (美國西部)

支援的區域和資料庫引擎，適用於 Amazon RDS 零 ETL 與 Amazon Redshift 整合

RDS 零 ETL 與 Amazon Redshift 整合是一種全受管解決方案，可在將交易資料寫入 Amazon RDS 資料庫執行個體之後，在 Amazon Redshift 中使用該資料。如需詳細資訊，請參閱 [使用零 ETL 整合 \(預覽\)](#)。

下列區域和引擎版本適用於與 Amazon Redshift 的零 ETL 整合。

區域	RDS for MySQL 8.0
美國東部 (維吉尼亞北部)	8.0.28 版和更新版本
美國東部 (俄亥俄)	8.0.28 版和更新版本
美國西部 (奧勒岡)	8.0.28 版和更新版本
亞太區域 (東京)	8.0.28 版和更新版本
歐洲 (愛爾蘭)	8.0.28 版和更新版本

Amazon RDS 中的引擎原生功能

Amazon RDS 資料庫引擎也支援許多最常見的引擎原生功能和功能。這些功能與此頁面上列出的 Amazon RDS 原生功能不同。部分引擎原生功能的支援或權限可能有所限制。

如需引擎原生功能的詳細資訊，請參閱：

- [適用於 Db2 的 Amazon RDS 功能](#)
- [Amazon RDS 上的 MariaDB 功能支援](#)
- [Amazon RDS 上的 MySQL 功能支援](#)
- [RDS for Oracle 功能](#)
- [使用 Amazon RDS for PostgreSQL 支援的 PostgreSQL 功能](#)
- [Amazon RDS 上的 Microsoft SQL Server 功能](#)

Amazon RDS 的資料庫執行個體計費

Amazon RDS 執行個體是根據下列要素計費：

- 資料庫執行個體小時數 (每小時) – 根據資料庫執行個體的資料庫執行個體類別 (例如 db.t2.small 或 db.m4.large)。定價以每小時為單位列出，但帳單已採用秒數為計算單位，並以十進位制顯示時間。RDS 用量以 1 秒遞增方式進行計費，最低計費標準為 10 分鐘。如需更多詳細資訊，請參閱 [資料庫執行個體類別](#)。
- 儲存 (每月每 GiB) – 您的資料庫執行個體佈建的儲存容量。如果您在月中擴展所佈建儲存容量的規模，則按比例計費。如需更多詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。
- 輸入/輸出 (I/O) 請求數 (每 100 萬個請求) – 您在計費週期中提出的儲存輸入/輸出請求總數，僅適用於 Amazon RDS 磁帶儲存。
- 佈建 IOPS (每月每 IOPS) – 佈建 IOPS 費率，不論 IOPS 的使用量如何，適用於 Amazon RDS 佈建 IOPS (SSD) 和一般用途 (SSD) gp3 儲存體。EBS 磁碟區的佈建儲存以 1 秒遞增方式進行計費，最低計費標準為 10 分鐘。
- 備份儲存 (每月每 GiB) – 備份儲存是指與自動資料庫備份，以及所有已建立之使用中資料庫快照相關聯的儲存。延長您的備份保留期或拍攝額外的資料庫快照，會增加資料庫所消耗的備份儲存。每秒計費不是適用於備份儲存 (以 GB 增量計算)。

如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

- 資料傳輸 (每 GB) - 在資料庫執行個體與網際網路和其他 AWS 區域之間的資料傳輸。

Amazon RDS 提供下列購買選項，可讓您根據需求選擇最適合的成本：

- On-Demand instances (隨需執行個體) – 依您使用的資料庫執行個體小時數付費。定價以每小時為單位列出，但帳單已採用秒數為計算單位，並以十進位制顯示時間。RDS 用量現以 1 秒遞增方式進行計費，最低計費標準為 10 分鐘。
- Reserved instances (預留執行個體) – 可選擇保留資料庫執行個體一年或三年，相較於隨需執行個體定價，可獲得極高的折扣。利用預留執行個體用量，您可以在一個小時內啟動、刪除、開始及停止等多個執行個體，並且讓所有執行個體皆獲得預留執行個體的好處。

如需 Amazon RDS 定價資訊，請參閱 [Amazon RDS 定價頁面](#)。

主題

- [Amazon RDS 的隨需資料庫執行個體](#)

- [Amazon RDS 的預留資料庫執行個體](#)

Amazon RDS 的隨需資料庫執行個體

Amazon RDS 隨需資料庫執行個體依資料庫執行個體類別計費 (例如 db.t3.small 或 db.m5.large)。如需 Amazon RDS 定價資訊，請參閱 [Amazon RDS 產品頁面](#)。

資料庫執行個體的帳單週期從該資料庫執行個體可用時開始計算。定價以每小時為單位列出，但帳單已採用秒數為計算單位，並以十進位制顯示時間。Amazon RDS 用量以 1 秒增量改進方式進行計費，最低計費標準為 10 分鐘。在應計費組態變更的情況下 (像是擴展運算或儲存容量)，系統最少會向您收取 10 分鐘的費用。計費將持續到資料庫執行個體終止為止，當您刪除資料庫執行個體，或是資料庫執行個體失敗時，執行個體就會終止。

如果您不再希望支付資料庫執行個體的費用，就必須將其停止或刪除，以免產生更多應計費的執行個體小時數。如需產生計費之資料庫執行個體狀態的詳細資訊，請參閱 [檢視 Aurora 叢集中 資料庫執行個體狀態](#)。

停止的資料庫執行個體

資料庫執行個體停止時，您需要支付佈建儲存 (包含佈建 IOPS) 的費用。您也需支付備份儲存 (含指定保留時段內的手動快照和自動備份) 的費用。您無須支付資料庫執行個體小時數的費用。

多個可用區資料庫執行個體

如果您指定資料庫執行個體為異地同步備份部署，則將根據異地同步備份定價計費 (發佈於 Amazon RDS 定價頁面)。

Amazon RDS 的預留資料庫執行個體

使用預留資料庫執行個體，即可保留資料庫執行個體一或三年。相較於隨需資料庫執行個體的定價，預留資料庫執行個體可提供您更多的折扣。預留資料庫執行個體並非實體執行個體，而是一種套用到您帳戶中特定隨需資料庫執行個體用量的計費折扣。預留資料庫執行個體的折扣依執行個體類型和 AWS 區域區域而異。

使用預留資料庫執行個體的一般流程如下：先取得關於可用的預留資料庫執行個體方案的資訊，接著購買預留資料庫執行個體方案，最後再取得關於您現有預留資料庫執行個體的資訊。

預留資料庫執行個體概觀

當您在 Amazon RDS 中購買預留資料庫執行個體時，您買到的是特定資料庫執行個體類型在預留資料庫執行個體期間內得享有折扣費率的承諾。若要使用 Amazon RDS 預留資料庫執行個體，您需要建立新的資料庫執行個體，如同建立隨需執行個體。

對於下列項目，您建立的新資料庫執行個體必須具有與預留資料庫執行個體相同的規格。

- AWS 區域
- 數據庫引擎 (DB 引擎的版本號不需要匹配。)
- 資料庫執行個體類型
- 資料庫執行個體大小 (適用於 Microsoft SQL 伺服器的 RDS 和 Amazon RDS for Oracle 文授權)
- 版本 (適用於 SQL 伺服器和 RDS 適用於甲骨文)
- 授權類型 (含授權或) bring-your-own-license

如果新資料庫執行個體規格與您帳戶現有的預留執行個體相符，您將以預留資料庫執行個體的折扣費率計費。否則，預留資料庫執行個體將按隨需費率計費。

您可以修改做為預留資料庫執行個體的資料庫執行個體。若修改符合預留資料庫執行個體的規格，則部分或全部折扣仍適用於修改後的資料庫執行個體。若修改超出規格 (例如變更執行個體類別)，則折扣將不再適用。如需詳細資訊，請參閱 [彈性大小的預留資料庫執行個體](#)。

主題

- [方案類型](#)
- [彈性大小的預留資料庫執行個體](#)
- [預留資料庫執行個體計費範例](#)

- [多可用區域資料庫叢集的預留資料庫執行個體](#)
- [刪除預留資料庫執行個體](#)

如需預留資料庫執行個體的詳細資訊，包括定價，請參閱 [Amazon RDS 預留執行個體](#)。

方案類型

預留資料庫執行個體分為三種—無預付、部分預付和全額預付—可讓您依據預期的使用量將 Amazon RDS 成本最佳化。

不預付

此選項讓您不用支付預付款便能存取預留資料庫執行個體。無預付的預留資料庫執行個體在期間中以折扣後的每小時費率計費，無論是否有使用，而且不需要預付款。這個選項只適用於為期一年的預留。

部分預付

此選項需預先支付部分的預留資料庫執行個體。期間內其餘的時數會以折扣後的每小時費率計費，無論是否有使用。此選項為先前重度使用選項的替代方案。

全額預付

期間開始時便支付完整的款項，並在期間的剩餘部分不會產生其他成本或額外的每小時費用，無論使用多少小時。

如果您使用合併帳單，組織中的所有帳戶都會視為一個帳戶處理。這表示組織中的所有帳戶可以獲得其他任何帳戶購買之預留資料庫執行個體的每小時成本利益。如需更多有關合併帳單的資訊，請參閱 AWS 帳單與成本管理使用者指南中的 [Amazon RDS 預留資料庫執行個體](#)。

彈性大小的預留資料庫執行個體

當您購買預留資料庫執行個體時，您應指定的其中一項內容為執行個體類別，例如 db.r5.large。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

如果您有資料庫執行個體，而且您需要擴展為更大的容量，則您的預留資料庫執行個體將自動套用到擴展後的資料庫執行個體。也就是說，預留資料庫執行個體會自動套用到所有的資料庫執行個體類別大小。大小靈活的預留資料庫執行個體適用於具有相同 AWS 區域 資料庫引擎的資料庫執行個體。彈性大小的預留資料庫執行個體只能在其執行個體類別類型中擴展。例如，db.r5.large 的預留資料庫執行個體可套用至 db.r5.xlarge，但不可套用至 db.r6g.large，因為 db.r5 和 db.r6g 是不同的執行個體類別類型。

預留資料庫執行個體的優點也適用於多可用區域和單一可用區的組態。彈性表示您可以在相同資料庫執行個體類別類型內的組態之間自由移動。例如，您可以從在一個大型資料庫執行個體上執行的單一可用區部署 (每小時四個標準化單位) 移至在兩個中型資料庫執行個體上執行的異地同步備份部署 (每小時 $2+2 = 4$ 個標準化單位)。

彈性大小的預留資料庫執行個體可供下列 Amazon RDS 資料庫引擎使用：

- RDS for MariaDB
- RDS for MySQL
- RDS 適用於甲骨文，攜帶您自己的許可證
- RDS for PostgreSQL

大小彈性不適用於 SQL 伺服器的 RDS 和包含 Oracle 授權的 RDS。

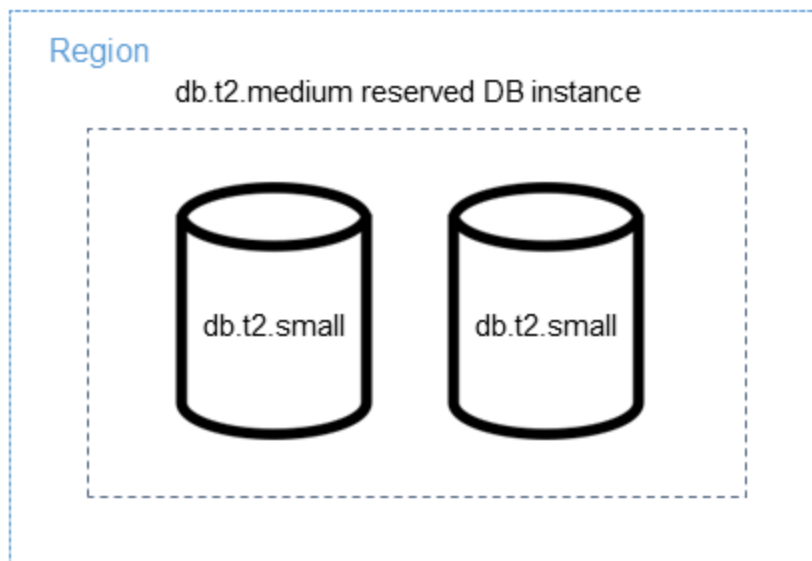
如需使用彈性大小的預留執行個體搭配 Aurora 的詳細資訊，請參閱 [Aurora 的預留資料庫執行個體](#)。

您可以使用每小時標準化單位數來比較不同預留資料庫執行個體大小的使用量。例如，兩個 db.r3.large 資料庫執行個體的一單位使用量相當於一個 db.r3.small 的每小時 8 個標準化單位使用量。下表顯示每個資料庫執行個體大小的每小時標準化單位數。

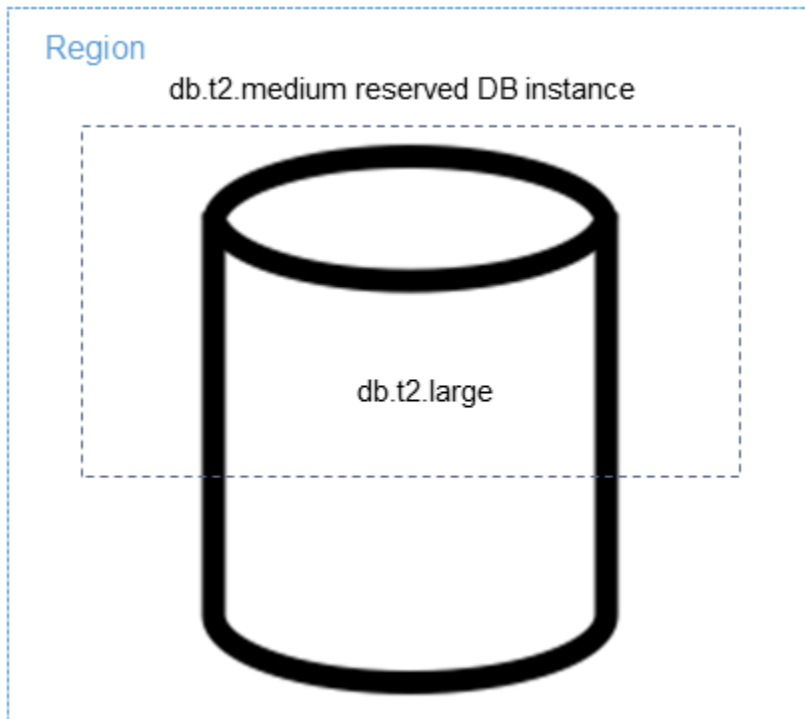
執行個體大小	每小時單一可用區標準化單位 (使用一個資料庫執行個體進行部署)	每小時多可用區資料庫執行個體標準化單位 (使用一個資料庫執行個體和一個待命部署)	每小時多可用區資料庫叢集標準化單位 (使用一個資料庫執行個體和兩個待命進行部署)
微型	0.5	1	1.5
小型	1	2	3
中型	2	4	6
大型	4	8	12
xlarge	8	16	24
2xlarge	16	32	48
4xlarge	32	64	96

執行個體大小	每小時單一可用區標準化單位 (使用一個資料庫執行個體進行部署)	每小時多可用區資料庫執行個體標準化單位 (使用一個資料庫執行個體和一個待命部署)	每小時多可用區資料庫叢集標準化單位 (使用一個資料庫執行個體和兩個待命進行部署)
6xlarge	48	96	144
8xlarge	64	128	192
10xlarge	80	160	240
12xlarge	96	192	288
16xlarge	128	256	384
24xlarge	192	384	576
32xlarge	256	512	768

例如，假設您購買 db.t2.medium 預留資料庫執行個體，且在相同 AWS 區域中您帳戶內有兩個執行中的 db.t2.small 資料庫執行個體。在此情況下，計費利益便會立即完整套用到兩個執行個體。



或者，如果您的帳戶中有一個 db.t2.large 執行個體在同一個執行個體中執行 AWS 區域，帳單優惠將套用至資料庫執行個體使用量的 50%。



預留資料庫執行個體計費範例

預留資料庫執行個體的價格不會為與儲存、備份和 I/O 相關聯的成本提供折扣。它只會對每小時隨需執行個體用量提供折扣。下列範例說明預留資料庫執行個體的每月總成本：

- 對於美國東部 (維吉尼亞北部) 無預付款選項的 RDS for MySQL 預留單一可用區域 db.r5.large 資料庫執行個體類別，執行個體的費用為 0.12 USD，或每月 90 USD
- 對於 400 GiB 的一般用途 SSD (gp2) 儲存，費用為每 GiB 每月 \$0.115，或每月 \$45.60
- 600 GiB 備份儲存為 \$0.095，每月 \$19 (400 GiB 免費)

對於預留資料庫執行個體新增所有這些費用 ($\$90 + \$45.60 + \$19$)，每月總成本為 \$154.60。

如果您選擇使用隨需資料庫執行個體，而非預留資料庫執行個體，對於美國東部 (維吉尼亞北部) 的 RDS for MySQL 預留單一可用區域 db.r5.large 資料庫執行個體類別，費用為每小時 0.1386 USD，或每月 101.18 USD。因此，對於隨需資料庫執行個體，新增所有這些選項 ($\$101.18 + \$45.60 + \$19$)，每月總成本為 \$165.78。您可以使用預留資料庫執行個體，每月節省超過 11 美元。

Note

此範例中的價格是範例價格，可能與實際價格不符。如需 Amazon RDS 定價資訊，請參閱 [Amazon RDS 定價](#)。

多可用區域資料庫叢集的預留資料庫執行個體

若要為多可用區域資料庫叢集購買同等的預留資料庫執行個體，您可以執行下列其中一項：

- 保留三個與叢集中執行個體大小相同的單一可用區資料庫執行個體。
- 保留一個多可用區域資料庫執行個體和一個單一可用區域資料庫執行個體，其與叢集中的資料庫執行個體大小相同。

例如，假設您有一個叢集，其中包含三個 db.m6gd.large 資料庫執行個體。在這種情況下，您可以購買三個 db.m6gd.large 單一可用區預留資料庫執行個體，或一個 db.m6gd.large 多可用區預留資料庫執行個體和一個 db.m6gd.large 單一可用區預留資料庫執行個體。任一選項都會保留多可用區資料庫叢集的最大預留執行個體折扣定價方案。

或者，您可以使用大小靈活的資料庫執行個體，並購買較大的資料庫執行個體，以涵蓋一或多個叢集中較小的資料庫執行個體。例如，如果您有兩個叢集，其中總共有六個 db.m6gd.large 資料庫執行個體，則可以購買三個 db.m6gd.xl 單一可用區預留資料庫執行個體。這樣做會將全部六個資料庫執行個體保留在兩個叢集中。如需詳細資訊，請參閱 [彈性大小的預留資料庫執行個體](#)。

您可以保留與叢集中資料庫執行個體大小相同的資料庫執行個體，但保留的資料庫執行個體數量少於叢集中的資料庫執行個體總數。不過，如果您這麼做，叢集只會部分保留。例如，假設您有一個叢集，其中包含三個 db.m6gd.large 資料庫執行個體，而您購買了一個 db.m6gd.large 多可用區預留資料庫執行個體。在這種情況下，叢集只會部分保留，因為預留資料庫執行個體只涵蓋叢集中三個執行個體的其中兩個。剩餘的資料庫執行個體按照隨需 db.m6gd.large 小時費率收費。

如需建立多可用區域資料庫叢集的詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

刪除預留資料庫執行個體

預留資料庫執行個體的期間一般為一年或三年承諾。您無法取消預留資料庫執行個體。但是，您可以刪除預留資料庫執行個體折扣所涵蓋的資料庫執行個體。刪除預留資料庫執行個體折扣所涵蓋之資料庫執行個體的流程，與其他任何資料庫執行個體相同。

無論您是否使用這些資源，都會向您收取預付費用。

如果刪除預留資料庫執行個體折扣所涵蓋的資料庫執行個體，您仍可以啟動其他規格相容的資料庫執行個體。在此情況下，您仍可以在保留時間（一或三年）內繼續享有折扣費率。

使用預留資料庫執行個體

您可以使用 AWS Management Console、AWS CLI、和 RDS API 來處理預留資料庫執行個體。

主控台

您可以使用 AWS Management Console 來處理預留資料庫執行個體，如下列程序所示。

取得可用的預留資料庫執行個體方案的定價與資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Reserved instances (預留執行個體)。
3. 選擇 Purchase Reserved DB Instance (購買預留資料庫執行個體)。
4. 在 Product description (產品說明) 中，選擇資料庫引擎和授權類型。
5. 在 DB instance class (資料庫執行個體類別) 中，選擇資料庫執行個體類別。
6. 針對部署選項，選擇您要單一可用區還是多可用區資料庫執行個體部署。

Note

若要為多可用區資料庫叢集部署購買同等的預留資料庫執行個體，請購買三個單一可用區預留資料庫執行個體，或者購買一個多可用區和一個單一可用區預留資料庫執行個體。如需詳細資訊，請參閱 [多可用區域資料庫叢集的預留資料庫執行個體](#)。

7. 在期限中，選擇您想要預留資料庫執行個體的時間長度。
8. 在 Offering type (方案類型) 中，選擇方案類型。

選擇方案類型後，便會顯示定價資訊。

Important

選擇 Cancel (取消)，將不會購買預留資料庫執行個體，也不會產生任何費用。

取得可用的預留資料庫執行個體方案資訊後，您便能利用這些資訊來購買方案，如以下程序所示。

購買預留資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Reserved instances (預留執行個體)。
3. 選擇 Purchase reserved DB instance (購買預留資料庫執行個體)。
4. 在 Product description (產品說明) 中，選擇資料庫引擎和授權類型。
5. 在 DB instance class (資料庫執行個體類別) 中，選擇資料庫執行個體類別。
6. 在多可用區部署中，選擇您要單一可用區還是多可用區資料庫執行個體部署。

Note

若要為多可用區資料庫叢集部署購買同等的預留資料庫執行個體，請購買三個單一可用區預留資料庫執行個體，或者購買一個多可用區和一個單一可用區預留資料庫執行個體。如需詳細資訊，請參閱 [多可用區域資料庫叢集的預留資料庫執行個體](#)。

7. 在 Term (期限) 中，選擇您想要預留資料庫執行個體的時間長度。
8. 在 Offering type (方案類型) 中，選擇方案類型。

選擇方案類型後，便會顯示定價資訊。
9. (選用) 您可將自己的識別符指派至所購買的預留執行個體，以協助您追蹤這些執行個體。針對 Reserved Id (預留 ID) 中，輸入預留資料庫執行個體的識別符。
10. 選擇提交。

您的預留資料庫執行個體已購買，然後顯示在 Reserved instances (預留執行個體) 清單中。

購買預留資料庫執行個體後，您將取得預留資料庫執行個體的資訊，如以下程序所示。

取得 AWS 帳戶預留資料庫執行個體的相關資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Navigation (導覽) 窗格中，選擇 Reserved instances (預留執行個體)。

將顯示您帳戶的預留資料庫執行個體。若要查看特定預留資料庫執行個體的詳細資訊，請選擇清單中的該執行個體。在主控台底端的詳細資訊窗格中，您即可看到該執行個體的詳細資訊。

AWS CLI

您可以使用 AWS CLI 來處理預留資料庫執行個體，如下列範例所示。

Example 取得可用的預留資料庫執行個體方案

若要取得有關可用預留資料庫執行個體產品的資訊，請呼叫 AWS CLI 命令 [describe-reserved-db-instances-offerings](#)。

```
aws rds describe-reserved-db-instances-offerings
```

此呼叫會傳回類似以下的輸出：

```
OFFERING OfferingId          Class      Multi-AZ  Duration  Fixed
Price Usage Price Description Offering Type
OFFERING 438012d3-4052-4cc7-b2e3-8d3372e0e706 db.r3.large y          1y
1820.00 USD 0.368 USD  mysql      Partial Upfront
OFFERING 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f db.r3.small n          1y
227.50 USD 0.046 USD  mysql      Partial Upfront
OFFERING 123456cd-ab1c-47a0-bfa6-12345667232f db.r3.small n          1y
162.00 USD 0.00 USD  mysql      All      Upfront
Recurring Charges: Amount Currency Frequency
Recurring Charges: 0.123 USD Hourly
OFFERING 123456cd-ab1c-37a0-bfa6-12345667232d db.r3.large y          1y
700.00 USD 0.00 USD  mysql      All      Upfront
Recurring Charges: Amount Currency Frequency
Recurring Charges: 1.25 USD Hourly
OFFERING 123456cd-ab1c-17d0-bfa6-12345667234e db.r3.xlarge n          1y
4242.00 USD 2.42 USD  mysql      No      Upfront
```

取得可用的預留資料庫執行個體方案資訊後，您便能利用這些資訊來購買方案。

若要購買預留資料庫執行個體，請[purchase-reserved-db-instances-offering](#)搭配下列參數使用 AWS CLI 命令：

- `--reserved-db-instances-offering-id` – 您想要購買之方案的 ID。請參閱上述範例，取得方案的 ID。
- `--reserved-db-instance-id` – 您可將自己的識別符指派至所購買的預留資料庫執行個體，以幫助追蹤這些執行個體。

Example 購買預留資料庫執行個體

下列範例會購買識別碼為 `649fd0c8-cf6d-47a0-bfa6-060f8e75e95f` 的預留資料庫執行個體，並指派的識別碼。*MyReservation*

對於LinuxmacOS、或Unix：

```
aws rds purchase-reserved-db-instances-offering \
  --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
  --reserved-db-instance-id MyReservation
```

在 Windows 中：

```
aws rds purchase-reserved-db-instances-offering ^
  --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
  --reserved-db-instance-id MyReservation
```

此命令會傳回類似以下的輸出：

RESERVATION	ReservationId	Class	Multi-AZ	Start Time		
Duration	Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	MyReservation	db.r3.small	y	2011-12-19T00:30:23.247Z	1y	455.00 USD 0.092 USD 1 payment-pending mysql Partial Upfront

購買預留資料庫執行個體後，您將取得預留資料庫執行個體的資訊。

若要取得 AWS 帳戶的保留資料庫執行個體相關資訊 [describe-reserved-db-instances](#)，請呼叫 AWS CLI 命令，如下列範例所示。

Example 取得您的預留資料庫執行個體

```
aws rds describe-reserved-db-instances
```

此命令會傳回類似以下的輸出：

RESERVATION	ReservationId	Class	Multi-AZ	Start Time		
Duration	Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	MyReservation	db.r3.small	y	2011-12-09T23:37:44.720Z	1y	455.00 USD 0.092 USD 1 retired mysql Partial Upfront

RDS API

您可以使用 RDS API 來處理預留資料庫執行個體：

- 若要取得可用的預留資料庫執行個體方案資訊，請呼叫 Amazon RDS API 操作 [DescribeReservedDBInstancesOfferings](#)。
- 取得可用的預留資料庫執行個體方案資訊後，您便能利用這些資訊來購買方案。請使用下列參數呼叫 [PurchaseReservedDBInstancesOffering](#) RDS API 操作：
 - `--reserved-db-instances-offering-id` – 您想要購買之方案的 ID。
 - `--reserved-db-instance-id` – 您可將自己的識別符指派至所購買的預留資料庫執行個體，以幫助追蹤這些執行個體。
- 購買預留資料庫執行個體後，您將取得預留資料庫執行個體的資訊。請呼叫 [DescribeReservedDBInstances](#) RDS API 操作。

檢視預留資料庫執行個體的帳單

您可以在 AWS Management Console 中的 Billing Dashboard (帳單儀表板) 中檢視預留資料庫執行個體的帳單。

檢視預留資料庫執行個體帳單

1. 登入 AWS Management Console。
2. 從右上角的 account menu (帳戶選單) 中，選擇 Billing Dashboard (帳單儀表板)。
3. 選擇儀表板右上角的 Bill Details (帳單詳細資訊)。
4. 在 AWS Service Charges (AWS 服務費用) 中，展開 Relational Database Service (關聯式資料庫服務)。
5. 擴展預留 AWS 區域 資料庫執行個體的位置，例如美國西部 (奧勒岡)。

當月的預留資料庫執行個體及其每小時費用會顯示在適用於#####預留執行個體的 Amazon Relational Database Service。

Amazon Relational Database Service for MySQL, Community Edition Reserved Instances		\$0.00
MySQL, db.t3.micro reserved instance applied, db.t3.micro instance used	395,000 Hrs	\$0.00
USD 0.0 hourly fee per MySQL, db.t3.micro instance	720,000 Hrs	\$0.00

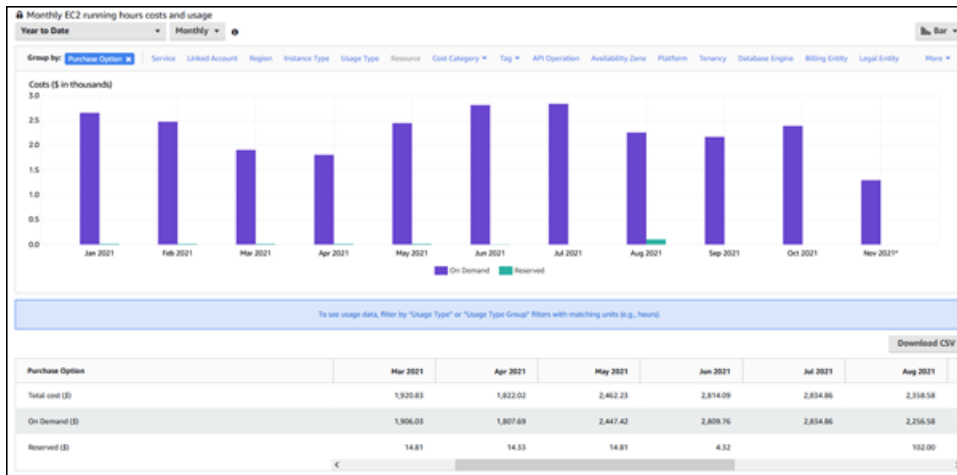
此範例中的預留資料庫執行個體是全部預付購買的，因此不會有每小時的收費。

6. 選擇 Reserved Instances (預留執行個體) 標題旁邊的 Cost Explorer (長條圖) 圖示。

Cost Explorer 會顯示 Monthly EC2 running hours costs and usage (每月 EC2 執行時數成本與用量) 圖形。

7. 清除圖表右側的 Usage Type Group (用量類型群組) 篩選條件。
8. 選擇要檢查用量成本的時間期間和時間單位。

下列範例顯示以月份為單位的今年迄今隨需和預留資料庫執行個體的用量成本。



2021 年 1 月到 6 月的預留資料庫執行個體成本是部分預付執行個體的每月費用，而 2021 年 8 月的成本則是全部預付執行個體的一次性費用。

部分預付執行個體的預留執行個體折扣在 2021 年 6 月到期，但是資料庫執行個體不會刪除。到期日之後，只會按隨需費率收費。

設定 Amazon RDS

首次使用 Amazon Relational Database Service 之前，請先完成下列任務：

主題

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)
- [授與程式設計存取權](#)
- [判定需求](#)
- [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)

如果您已有 AWS 帳戶，請瞭解您的 Amazon RDS 需求，並且偏好使用 IAM 和 VPC 安全群組的預設值，請跳到[Amazon RDS 入門](#)。

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 [root 使用者來執行需要 root 使用者存取權](#)的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理權限的使用者

註冊後，請保護您的 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。[AWS Management Console](#)在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

[若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。](#)

以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者[登入的說明](#)，請參閱[使用AWS 登入 者指南中的登入 AWS 存取入口網站](#)。

指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[新增群組](#)」。

授與程式設計存取權

如果使用者想要與 AWS 之外的 AWS Management Console 授與程式設計存取權的方式取決於正在存取的使用者類型。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> 如需詳細資訊 AWS CLI，請參閱 《使 AWS CLI 用 AWS Command Line Interface 者指南》 AWS IAM Identity Center 中的〈配置使用〉。 如需 AWS SDK、工具和 AWS API，請參閱 AWS SDK 和工具參考指南中的 IAM 身分中心身分驗證。
IAM	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	遵循 《IAM 使用者指南》 中的〈 將臨時登入資料搭配 AWS 資源 使用〉中的指示
IAM	(不建議使用) 使用長期認證簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> 如需相關資訊 AWS CLI，請參閱使用指南中的 使用 IAM 使用者登入資料進行驗證。AWS Command Line Interface 對於 AWS SDK 和工具，請參閱 AWS SDK 和工具參

哪個使用者需要程式設計存取權？	到	By
		<p>考指南中的使用長期憑據進行身份驗證。</p> <ul style="list-style-type: none">如需 AWS API，請參閱 IAM 使用者指南中的管理 IAM 使用者的存取金鑰。

判定需求

Amazon RDS 的基本建置組塊為資料庫執行個體。您可以在資料庫執行個體中建立您的資料庫。資料庫執行個體提供的網路位址稱為端點。您的應用程式使用此端點來連接至您的資料庫執行個體。建立資料庫執行個體時，您可以指定儲存體、記憶體、資料庫引擎和版本、網路組態、安全和維護期間等詳細資訊。您可以透過安全群組來控制對資料庫執行個體的網路存取。

建立資料庫執行個體和安全群組之前，您必須知道您的資料庫執行個體和網路需求。這裡是一些要考慮的注意事項：

- 資源需求 – 您的應用程式或服務的記憶體和處理器需求為何？您可以使用這些設定來幫助判定要使用的資料庫執行個體類別。如需關於資料庫執行個體類別的規格，請參閱 [資料庫執行個體類別](#)。
- VPC、子網路和安全群組 – 您的資料庫執行個體最可能處於 Virtual Private Cloud (VPC) 中。若要連接至您的資料庫執行個體，您必須設定安全群組規則。這些規則的設定方式因您使用的 VPC 的種類以及您使用它的方式而有所不同。例如，您可以使用預設 VPC 或使用者定義的 VPC。

下列清單說明每個 VPC 選項的規則：

- 預設 VPC — 如果您的 AWS 帳戶在目前 AWS 區域中具有預設 VPC，則該 VPC 會設定為支援資料庫執行個體。如果您在建立資料庫執行個體時指定預設的 VPC，請執行下列動作：
 - 確認建立 VPC 安全群組，授權從應用程式或服務到 Amazon RDS 資料庫執行個體的連線。使用 VPC 主控台上的 [安全群組] 選項，或使用建 AWS CLI 立 VPC 安全群組。如需相關資訊，請參閱「[步驟 3：建立 VPC 安全群組](#)」。
 - 指定預設的資料庫子網路群組。如果這是您在此 AWS 區域中建立的第一個資料庫執行個體，Amazon RDS 會在建立資料庫執行個體時建立預設資料庫子網路群組。
- 使用者定義的 VPC – 如果您要在建立資料庫執行個體時指定使用者定義的 VPC，請注意下列事項：

- 確認建立 VPC 安全群組，授權從應用程式或服務到 Amazon RDS 資料庫執行個體的連線。使用 VPC 主控台上的 [安全群組] 選項，或使用建 AWS CLI 立 VPC 安全群組。如需相關資訊，請參閱「[步驟 3：建立 VPC 安全群組](#)」。
- VPC 必須符合某些需求才能主控資料庫執行個體，例如具有至少兩個子網路，每個位在個別的可用區域中。如需相關資訊，請參閱「[Amazon VPC 和 Amazon RDS](#)」。
- 務必指定 DB 子網路群組，該群組定義在該 VPC 中可由資料庫叢集使用的子網路。如需詳細資訊，請參閱在 [VPC 中使用資料庫執行個體](#) 中的資料庫子網路群組小節。
- 高可用性 – 您需要容錯移轉支援嗎？在 Amazon RDS 上，異地同步備份部署會建立主要資料庫執行個體，並在另一個可用區域中建立次要待命資料庫執行個體以進行容錯移轉支援。建議對生產工作負載使用異地同步備份部署以保有高可用性。針對開發和測試目的，您可以使用異地同步備份以外的部署。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。
- IAM 政策 — 您的 AWS 帳戶是否有政策授予執行 Amazon RDS 操作所需的許可？如果您要連線到 AWS 使用 IAM 登入資料，您的 IAM 帳戶必須具有 IAM 政策，以授與執行 Amazon RDS 操作所需的許可。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。
- 開放連接埠 – 您的資料庫接聽所在的 TCP/IP 連接埠為何？某些公司的防火牆可能會封鎖與您的資料庫引擎預設連接埠的連線。如果您的公司防火牆會封鎖預設連接埠，請為新的資料庫執行個體選擇另一個連接埠。建立在您指定的連接埠上接聽的資料庫執行個體時，您可以透過修改資料庫執行個體來變更連接埠。
- AWS 地區 — 您希望數據庫位於哪個 AWS 區域？將您的資料庫放置在鄰近您的應用程式或 Web 服務的位置可以減少網路延遲。如需詳細資訊，請參閱 [區域、可用區域和 Local Zones](#)。
- 資料庫磁碟子系統 – 您的儲存體有何要求？Amazon RDS 提供了三種儲存類型：
 - 一般用途 (SSD)
 - 佈建的 IOPS (PIOPS)
 - 磁帶 (也稱為標準儲存裝置)

如需 Amazon RDS 儲存體的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

具備建立安全群組和資料庫執行個體所需的資訊時，請繼續進行下一個步驟。

建立安全群組以存取在您的 VPC 中您的資料庫執行個體

VPC 安全群組提供 VPC 中資料庫執行個體的存取權。其作用就像相關資料庫執行個體的防火牆，從資料庫執行個體層級控制傳入和傳出流量。資料庫執行個體建立時預設提供防火牆和可保護資料庫執行個體的預設安全群組。

您必須先新增規則至可讓您連線的安全群組，才可連線到資料庫執行個體。使用您的網路和組態資訊來建立規則以允許存取您的資料庫執行個體。

例如，假設您有一個應用程式會存取 VPC 中您的資料庫執行個體上的資料庫。在此情況下，您必須新增自訂 TCP 規則，指定您的應用程式用來存取資料庫的連接埠範圍和 IP 地址。如果您的應用程式在 Amazon EC2 執行個體上，則可以使用您為 Amazon EC2 執行個體設定安全群組。

您在建立資料庫執行個體時，可以將 Amazon EC2 執行個體之間的連線設定為資料庫執行個體。如需詳細資訊，請參閱 [設定與 EC2 執行個體的自動網路連線](#)。

Tip

您在建立資料庫執行個體時，可以自動設定 Amazon EC2 執行個體和資料庫執行個體之間的網路連線。如需詳細資訊，請參閱 [設定與 EC2 執行個體的自動網路連線](#)。

如需存取資料庫執行個體常見案例的相關資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

建立 VPC 安全群組

1. 登入 AWS Management Console 並開啟 Amazon VPC 主控台，網址為 <https://console.aws.amazon.com/vpc>。


Note

請確認您位於 VPC 主控台中，而不是 RDS 主控台。

2. 在的右上角 AWS Management Console，選擇您要建立 VPC 安全群組和資料庫執行個體的 AWS 區域。在 AWS 區域的 Amazon VPC 資源清單中，您應該會看到至少一個 VPC 和數個子網路。如果沒有，則該 AWS 區域中沒有預設 VPC。
3. 在導覽窗格中，選擇 Security Groups (安全群組)。
4. 選擇 Create Security Group (建立安全群組)。隨即會顯示 Create security group (建立安全群組) 頁面。
5. 在 Basic details (基本詳細資訊) 中，分別在 Security group name (安全群組名稱) 和 Description (描述) 中輸入相應內容。針對 VPC，選擇您要建立您的資料庫執行個體所在的 VPC。
6. 在 Inbound rules (入站規則) 中，選擇 Add rule (新增規則)。
 - a. 針對 Type (類型)，請選擇 Custom TCP (自訂 TCP)。

- b. 針對 Port Range (連接埠範圍)，輸入要用於資料庫執行個體的連接埠值。
 - c. 針對 Source (來源)，選擇要從中存取資料庫執行個體的安全群組名稱或輸入 IP 地址範圍 (CIDR 值)。如果您選擇 My IP (我的 IP)，此舉允許透過您的瀏覽器中偵測到的 IP 地址存取資料庫執行個體。
7. 如果需要新增更多 IP 地址或不同的連接埠範圍，請選擇 Add rule (新增規則) 並輸入規則的資訊。
 8. (選用) 在 Outbound Rules (輸出規則) 中，新增輸出流量的規則。預設會允許所有傳出流量。
 9. 選擇建立安全群組。


您可以使用剛建立的 VPC 安全群組，做為所建立之資料庫執行個體的安全群組。

 Note

如果您使用預設 VPC，系統會為您建立橫跨所有 VPC 子網路的預設子網路群組。建立資料庫執行個體時，您可以選取預設的 VPC，並在 DB Subnet Group (資料庫子網路群組) 使用 default (預設)。

在完成了設定需求之後，您可以使用您的需求和安全群組建立資料庫執行個體。若要執行此操作，請遵循 [建立 Amazon RDS 資料庫執行個體](#) 中的指示。如需建立使用特定資料庫引擎的資料庫執行個體之相關資訊，請參閱下方表單中的相關文件。

資料庫引擎	文件
MariaDB	建立並連線至 MariaDB 資料庫執行個體
Microsoft SQL Server	建立並連線至 Microsoft SQL Server 資料庫執行個體
MySQL	建立並連線至 MySQL 資料庫執行個體
Oracle	建立並連線至 Oracle 資料庫執行個體
PostgreSQL	建立並連線至 PostgreSQL 資料庫執行個體

 Note

如果您在建立資料庫執行個體之後，無法連線到該執行個體，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#) 中的疑難排解資訊。

Amazon RDS 入門

在下列範例中，您可以了解如何使用 Amazon Relational Database Service (Amazon RDS) 建立並連線至資料庫執行個體。您可以建立使用 Db2、MariaDB、MySQL、Microsoft SQL 伺服器、甲骨文或 PostgreSQL 的資料庫執行個體。

Important

您必須完成[設定 Amazon RDS](#) 中的任務，才能建立或連線至資料庫執行個體。

在建立資料庫執行個體和連線至資料庫執行個體上的資料庫時，每個資料庫引擎的作法稍有不同。在下面選擇其中一個資料庫引擎，以取得如何建立和連線至資料庫執行個體的詳細資訊。建立並連線至資料庫執行個體之後，也會有指示協助您刪除資料庫執行個體。

主題

- [建立並連線至 MariaDB 資料庫執行個體](#)
- [建立並連線至 Microsoft SQL Server 資料庫執行個體](#)
- [建立並連線至 MySQL 資料庫執行個體](#)
- [建立並連線至 Oracle 資料庫執行個體](#)
- [建立並連線至 PostgreSQL 資料庫執行個體](#)
- [教學：建立 Web 伺服器和 Amazon RDS 資料庫執行個體](#)
- [教學課程：使用 Lambda 函數來存取 Amazon RDS 資料庫](#)

建立並連線至 MariaDB 資料庫執行個體

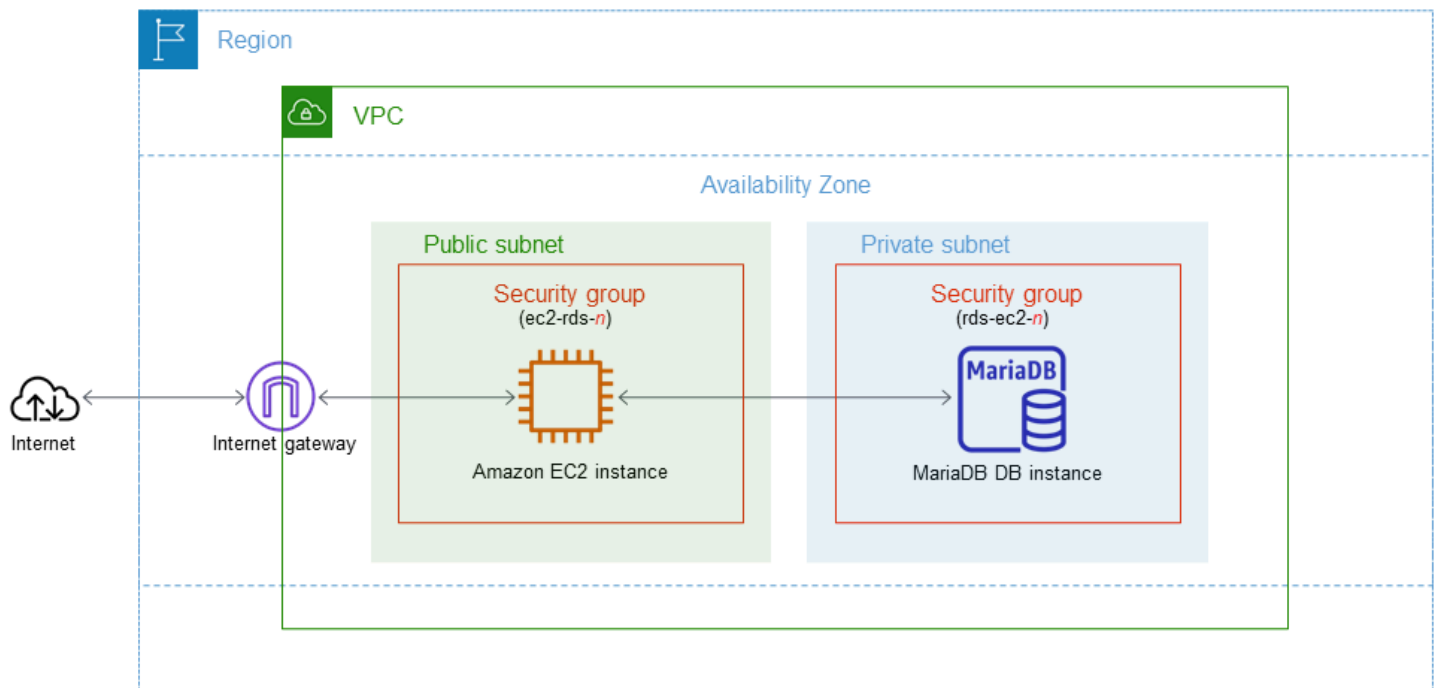
本教學課程會建立 EC2 執行個體和 RDS for MariaDB 資料庫執行個體。本教學課程說明如何使用標準 MySQL 用戶端，從 EC2 執行個體存取資料庫執行個體。本教學課程為最佳實務，會在虛擬私有雲端 (VPC) 中建立私有資料庫執行個體。在多數情況下，相同 VPC 中的其他資源 (例如 EC2 執行個體) 可以存取該資料庫執行個體，但 VPC 以外的資源便無法存取該執行個體。

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。在可用區域中，EC2 執行個體位於公有子網路中，而資料庫執行個體則可於私有子網路中。

⚠ Important

建立 AWS 帳戶。不過，若要完成本教學課程，您可能需要支付所使用的資源的費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

下圖顯示此教學課程完成時的組態。



本教學課程可讓您使用下列其中一種方法來建立資源：

1. 使用 AWS Management Console - [步驟 1：建立 EC2 執行個體](#) 和 [步驟 2：建立 MariaDB 資料庫執行個體](#)

2. 用 AWS CloudFormation 於建立資料庫執行個體和 EC2 執行個體-(選擇性) 使用建立 VPC、EC2 執行個體和 MariaDB 執行個體 [AWS CloudFormation](#)

第一種方法使用簡單的創建來創建一個私有 MariaDB 數據庫實例與 AWS Management Console 在這裡，您只指定資料庫引擎類型、資料庫執行個體大小和資料庫執行個體識別碼。Easy Create (輕鬆建立) 會使用其他組態選項的預設設定。

改為使用標準建立時，您可以在建立資料庫執行個體時指定更多組態選項。這些選項包括可用性、安全性、備份和維護的設定。若要建立公有資料庫執行個體，您必須使用標準建立。如需相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

主題

- [必要條件](#)
- [步驟 1：建立 EC2 執行個體](#)
- [步驟 2：建立 MariaDB 資料庫執行個體](#)
- [\(選擇性\) 使用建立 VPC、EC2 執行個體和 MariaDB 執行個體 AWS CloudFormation](#)
- [步驟 3：連線至 MariaDB 資料庫執行個體](#)
- [步驟 4：刪除 EC2 執行個體和資料庫執行個體](#)
- [\(選擇性\) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation](#)
- [\(選用\) 將資料庫執行個體連線至 Lambda 函數](#)

必要條件

在開始之前，請先完成下節所含步驟：

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

步驟 1：建立 EC2 執行個體

建立您會用來連線至資料庫的 Amazon EC2 執行個體。

建立 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。

2. 在的右上角 AWS Management Console，選擇您要 AWS 區域 在其中建立 EC2 執行個體的執行個體。
3. 選擇 EC2 儀表板，然後選擇啟動執行個體，如下圖所示。

Resources

You are using the following Amazon EC2 resources in the Region Region:

Instances (running)	3	Dedicated Hosts	0
Instances	3	Key pairs	5
Placement groups	0	Security groups	10
Volumes	3		

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼ **Migrate a server** ↗

Note: Your instances will launch in the US West (Oregon) Region

Service health

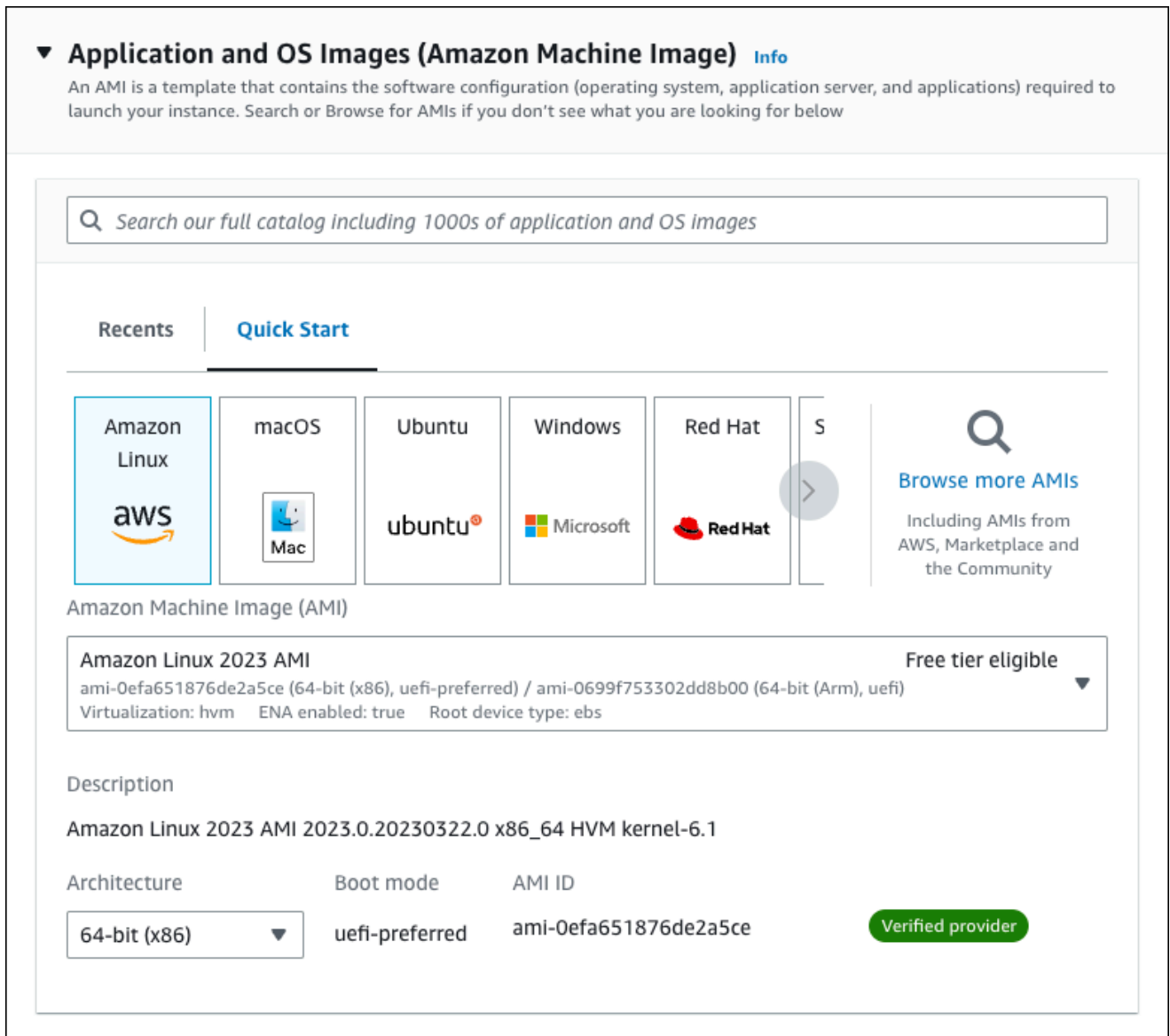
Region
Region

Zones

啟動執行個體頁面即開啟。

4. 在啟動執行個體頁面中選擇下列設定。
 - a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **ec2-database-connect**。

- b. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇 Amazon Linux，然後選擇 Amazon Linux 2023 AMI。保留其他選項的預設選擇。



▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat S

aws Mac ubuntu® Microsoft Red Hat

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0efa651876de2a5ce (64-bit (x86), uefi-preferred) / ami-0699f753302dd8b00 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0efa651876de2a5ce

Verified provider


- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需有關建立新 key pair 的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [建立 key pair](#)。

- e. 對於網路設定中的允許 SSH 流量，選擇 EC2 執行個體的 SSH 連線來源。

如果顯示的 IP 地址對 SSH 連線而言是正確的，您可以選擇 My IP (我的 IP)。否則，您可以決定用於使用 Secure Shell (SSH) 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 192.0.2.1/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

 Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

下圖顯示網路設定區段的範例。

▼ **Network settings** [Info](#) Edit

Network [Info](#)
vpc-1a2b3c4d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

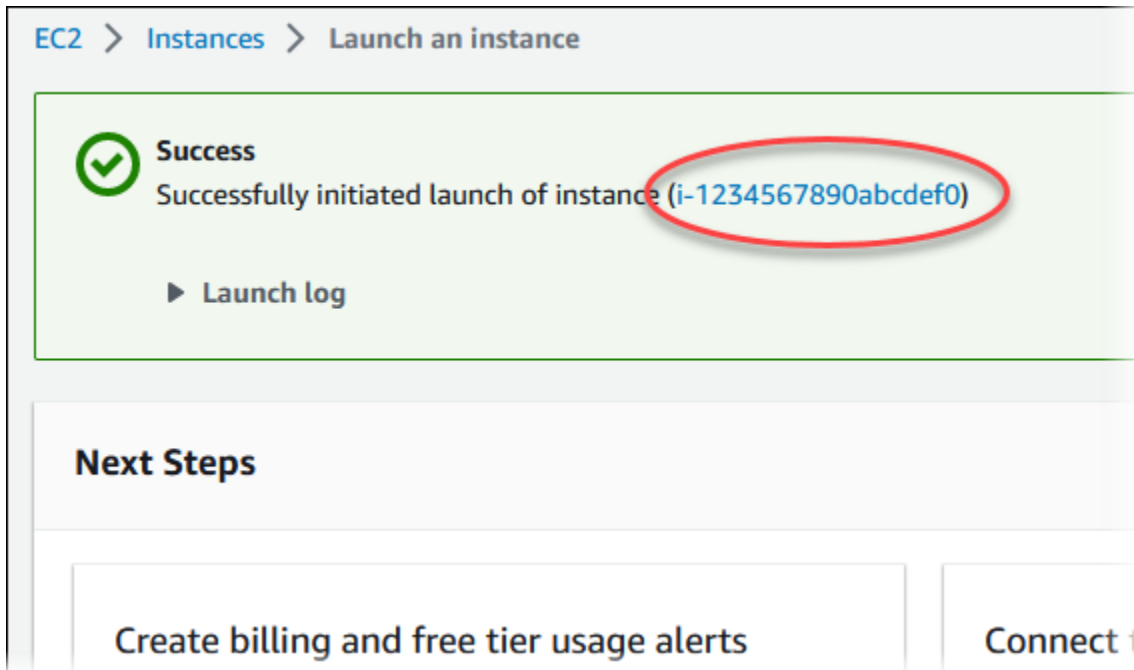
We'll create a new security group called **'launch-wizard-1'** with the following rules:

Allow SSH traffic from My IP
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

- f. 讓剩餘區段保留預設值。
 - g. 檢閱摘要面板中 EC2 執行個體組態的摘要，並在準備就緒時選擇啟動執行個體。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：i-1234567890abcdef0。



6. 選擇 EC2 執行個體識別符，以開啟 EC2 執行個體清單，然後選取您的 EC2 執行個體。
7. 在詳細資訊索引標籤中，請記下以下值，當您使用 SSH 進行連線時需要這些值：
 - a. 在執行個體摘要中，記下公用 IPv4 DNS 的值。

Details	Security	Networking	Storage	Status checks	Monitoring	Tags						
<p>▼ Instance summary Info</p> <table border="1"> <tr> <td>Instance ID i-1234567890abcdef0</td> <td>Public IPv4 address ██████████ open address</td> <td>Private IPv4 addresses ██████████</td> </tr> <tr> <td>IPv6 address -</td> <td>Instance state ⌚ Pending</td> <td>Public IPv4 DNS ec2-12-345-67-890.compute-1.amazonaws.com open address</td> </tr> </table>							Instance ID i-1234567890abcdef0	Public IPv4 address ██████████ open address	Private IPv4 addresses ██████████	IPv6 address -	Instance state ⌚ Pending	Public IPv4 DNS ec2-12-345-67-890.compute-1.amazonaws.com open address
Instance ID i-1234567890abcdef0	Public IPv4 address ██████████ open address	Private IPv4 addresses ██████████										
IPv6 address -	Instance state ⌚ Pending	Public IPv4 DNS ec2-12-345-67-890.compute-1.amazonaws.com open address										

- b. 在執行個體詳細資訊中，記下金鑰對名稱的值。

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name ec2-database-connect-key-pair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -

8. 請等待 EC2 執行個體的執行個體狀態為執行中，然後再繼續動作。

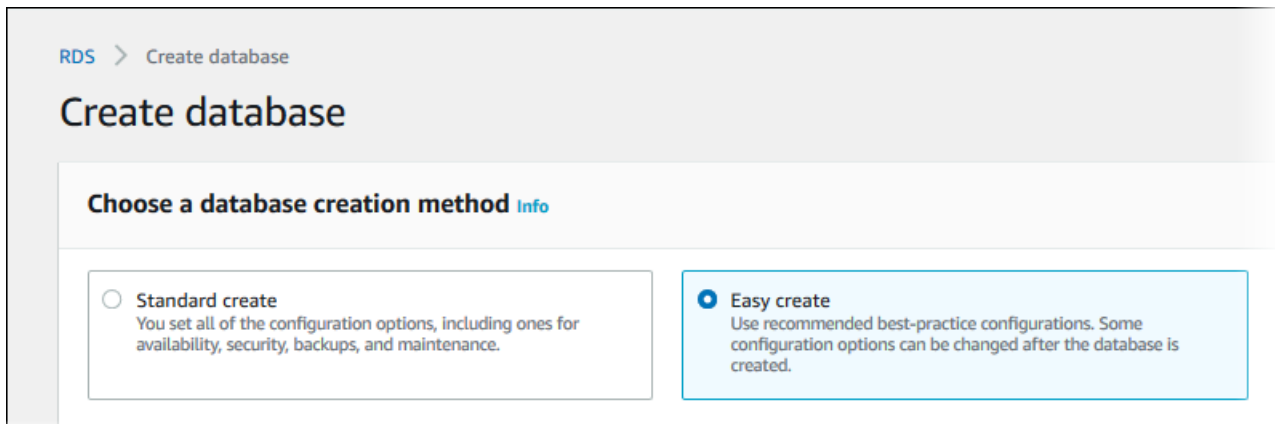
步驟 2：建立 MariaDB 資料庫執行個體

Amazon RDS 的基本建置組塊為資料庫執行個體。這是您執行 MariaDB 資料庫的環境。

於此範例中，您使用輕鬆建立來建立執行 MariaDB 資料庫引擎、資料庫執行個體類別為 db.t3.micro 的資料庫執行個體。

如要以 Easy create (輕鬆建立) 來建立 MariaDB 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇您要 AWS 區域 在其中建立資料庫執行個體的執行個體。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)，並確定選擇 Easy Create (輕鬆建立)。










5. 在 Configuration (組態)，選擇 MariaDB (MariaDB)。
6. 在 DB instance size (資料庫執行個體大小) 中，選擇 Free tier (免費方案)。
7. 對於 DB instance identifier (資料庫執行個體識別碼)，請輸入 **database-test1**。
8. 在主要使用者名稱，輸入主要使用者的名稱，或保留預設名稱。

Create database (建立資料庫) 頁面看起來應該會如下圖所示。

Configuration

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 	<input type="radio"/> MySQL 
<input checked="" type="radio"/> MariaDB 	<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 		

DB instance size

<input type="radio"/> Production db.r6g.xlarge 4 vCPUs 32 GiB RAM 500 GiB	<input type="radio"/> Dev/Test db.r6g.large 2 vCPUs 16 GiB RAM 100 GiB	<input checked="" type="radio"/> Free tier db.t3.micro 2 vCPUs 1 GiB RAM 20 GiB
---	--	---

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

9. 如要將自動產生的主要密碼用在資料庫執行個體，請選取自動產生密碼。

如要輸入您的主要密碼，請確認清除自動產生密碼方塊，然後在主要密碼和確認密碼中輸入相同的密碼。

10. 若要設定與先前建立之 EC2 執行個體的連線，請開啟設定 EC2 連線 - 選用。

選取連線至 EC2 運算資源。選擇先前建立的 EC2 執行個體。

▼ Set up EC2 connection - optional

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-

i-1234567890abcdef0

11. 開啟檢視輕鬆建立的預設設定。

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

Configuration ▼	Value	Editable after database is created ▲
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mariadb-10-6	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	10.6.10	Yes
DB parameter group	default.mariadb10.6	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

您可以檢查與 Easy Create (輕鬆建立) 一起使用的預設設定。資料庫建立後可編輯欄顯示您可以在資料庫建立後變更的選項。

- 若該欄的設定為否，而您想要其他設定，可以使用標準建立來建立資料庫執行個體。

- 若該欄的設定為是，而您想要其他設定，可以使用標準建立來建立資料庫執行個體，或在建立後修改該資料庫執行個體的設定。

12. 選擇建立資料庫。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

您可以使用出現的使用者名稱和密碼，來以主要使用者的身分連線至資料庫執行個體。

⚠ Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，您可以將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

13. 在資料庫清單中，選擇新 MariaDB 資料庫執行個體的名稱，以顯示其詳細資訊。

在其可供使用之前，資料庫執行個體會處於建立中狀態。

Summary			
DB identifier database-test1	CPU -	Status 🔄 Creating	Class db.t3.micro
Role Instance	Current activity	Engine MariaDB	Region & AZ us-east-1d

狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。

(選擇性) 使用建立 VPC、EC2 執行個體和 MariaDB 執行個體 AWS CloudFormation

您可以使用將基礎設施視為程式碼來佈建 AWS 資源，而不是使用 AWS CloudFormation 主控台建立 VPC、EC2 執行個體和 MariaDB 執行個體。為了幫助您將 AWS 資源組織成更小且更易於

管理的單元，您可以使用 AWS CloudFormation 嵌套堆棧功能。如需詳細資訊，請參閱 [在 AWS CloudFormation 主控台上建立堆疊](#) 和 [使用巢狀堆疊](#)。

Important

AWS CloudFormation 是免費的，但 CloudFormation 創建的資源是活的。您必須支付這些資源的標準使用費，直到您終止這些資源為止。總計費用會很少。如需如何將任何費用降至最低的相關資訊，請前往 [AWS 免費方案](#)。

若要使用 AWS CloudFormation 主控台建立資源，請完成以下步驟：

- 步驟 1：下載 CloudFormation 範本
- 步驟 2：使用 CloudFormation

下載 CloudFormation 範本

CloudFormation 範本是 JSON 或 YAML 文字檔案，其中包含您要在堆疊中建立之資源的相關設定資訊。此範本也會為您建立 VPC 和防禦主機，以及 RDS 執行個體。

要下載模板文件，請打開以下鏈接，[MariaDB CloudFormation](#) 的模板。

在 Github 頁面中，單擊下載原始文件按鈕以保存模板 YAML 文件。

使用 CloudFormation

Note

在開始此程序之前，請確定您的 AWS 帳戶。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

使用 AWS CloudFormation 範本時，您必須選取正確的參數，以確保資源已正確建立。請遵循下列步驟：

1. 請登入 AWS Management Console 並開啟 AWS CloudFormation 主控台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 選擇 Create Stack (建立堆疊)。

3. 在 [指定範本] 區段中，選取 [從電腦上傳範本檔案]，然後選擇 [下一步]。
4. 在「指定堆疊詳細資訊」頁面中，設定下列參數：
 - a. 將堆棧名稱設置為 MariaDB TestStack 的。
 - b. 在參數下，透過選取三個可用區域來設定可用區域。
 - c. 在 Linux 防禦主機組態下，對於金鑰名稱，選取要登入 EC2 執行個體的 key pair。
 - d. 在 Linux 防禦主機組態設定中，將允許的 IP 範圍設定為您的 IP 位址。若要使用安全殼層 (SSH) 連線至 VPC 中的 EC2 執行個體，請使用 <https://checkip.amazonaws.com> 的服務判斷您的公有 IP 位址。IP 地址的範例為 192.0.2.1/32。

Warning

如果您使用 0.0.0.0/0 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

- e. 在「資料庫一般組態」下，將「資料庫」執行個體類別設定為 db.t3.micro。
 - f. 將資料庫名稱設定為 **database-test1**。
 - g. 在資料庫主要使用者名稱中，輸入主要使用者的名稱。
 - h. 將使用密碼管理 Secrets Manager 管理資料庫主要使用者密碼設定 false 為此教學課程。
 - i. 對於資料庫密碼，請設定您選擇的密碼。請記住此密碼，以取得教學課程的進一步步驟。
 - j. 在 [資料庫儲存設定] 下，將 [資料庫儲存類型] 設定為 gp 2。
 - k. 在 [資料庫監視組態] 下，將 [啟用 RDS Performance Insights] 設定為 false。
 - l. 保留所有其他設定為預設值。按一下「下一步」繼續。
5. 在「複查堆疊」頁面中，選取檢查資料庫和 Linux 防禦主機選項後送出。

堆疊建立程序完成後，檢視包含名稱 BastionStack 和 RDSNS 的堆疊，以記下連線至資料庫所需的資訊。如需詳細資訊，請參閱 [在上檢視 AWS CloudFormation 堆疊資料和資源 AWS Management Console](#)。

步驟 3：連線至 MariaDB 資料庫執行個體

您可以使用任何標準 SQL 用戶端應用程式來連線至資料庫執行個體。在此範例中，您會使用 mysql 命令列用戶端來連線至 MariaDB 資料庫執行個體。

若要連線至 MariaDB 資料庫執行個體

1. 尋找資料庫執行個體的端點 (DNS 名稱) 和連接埠號碼。
 - a. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 - b. 在 Amazon RDS 主控台的右上角，選擇 AWS 區域 適用於資料庫執行個體的。
 - c. 在導覽窗格中，選擇 Databases (資料庫)。
 - d. 選擇 MariaDB 資料庫執行個體名稱以顯示其詳細資訊。
 - e. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

RDS > Databases > database-test1

database-test1

Summary

DB identifier database-test1	CPU 2.41%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com Port 3306	Networking Availability Zone us-east-1b VPC vpc-1a2b3c4d Subnet group default
---	--

2. 按照 Amazon EC2 使用者指南中的連線到 [Linux 執行個體中的步驟](#)，Connect 到先前建立的 EC2 執行個體。

建議您使用 SSH 連線至 EC2 執行個體。如果 SSH 用戶端公用程式已安裝在 Windows、Linux 或 Mac 上，您可以使用下列命令格式連線至執行個體：

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```


例如，假設 `ec2-database-connect-key-pair.pem` 存放在 Linux 上的 `/dir1` 中，而 EC2 執行個體的公用 IPv4 DNS 為 `ec2-12-345-678-90.compute-1.amazonaws.com`。您的 SSH 命令看起來如下所示：

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-  
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

- 更新 EC2 執行個體上的軟體，以取得最新的錯誤修正和安全性更新。若要執行此操作，請使用以下命令。

Note

`-y` 選項不要求確認就會安裝更新。若要先檢查更新再安裝，請省略此選項。

```
sudo dnf update -y
```

- 從 MariaDB 安裝 `mysql` 命令列用戶端。

若要在 Amazon Linux 2023 上安裝 MariaDB 命令列用戶端，請執行下列命令：

```
sudo dnf install mariadb105
```

- 連線到 MariaDB 資料庫執行個體。例如，輸入下列命令。此動作可讓您使用 MySQL 用戶端，連線至 MariaDB 資料庫執行個體。

取代您用於 *endpoint* 的資料庫執行個體端點 (DNS 名稱)，以及取代您用於 *admin* 的主要使用者名稱。提示您輸入密碼時，請提供您使用的主要密碼。

```
mysql -h endpoint -P 3306 -u admin -p
```

輸入使用者的密碼之後，您應該會看到類似如下的輸出。

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 156  
Server version: 10.6.10-MariaDB-log managed by https://aws.amazon.com/irds/  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]>
```

如需連線至 MariaDB 資料庫執行個體的詳細資訊，請參閱[連接至執行 MariaDB 資料庫引擎的資料庫執行個體](#)。如果您無法連線至資料庫執行個體，請參閱[無法連線至 Amazon RDS 資料庫執行個體](#)。

基於安全考量，最佳做法是使用加密連線。僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 MariaDB 連線。如需使用加密連線的詳細資訊，請參閱[使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)。

6. 執行 SQL 命令。

例如，下列 SQL 命令會顯示目前的日期和時間：

```
SELECT CURRENT_TIMESTAMP;
```

步驟 4：刪除 EC2 執行個體和資料庫執行個體

在連線至您已建立的範例 EC2 執行個體與資料庫執行個體，並探索這些執行個體之後，請將其刪除，才不會再對您收費。

如果您曾經 AWS CloudFormation 建立資源，請略過此步驟並前往下一個步驟。

刪除 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，選擇執行個體。
3. 選取 EC2 執行個體，並選取執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇 Terminate (終止)。

如需刪除 EC2 執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的[終止執行個體](#)。

刪除不含最終資料庫快照的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要刪除的資料庫執行個體。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 清除是否建立最後的快照？和保留自動備份。
6. 完成確認，然後選擇刪除。

(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation

如果您曾 AWS CloudFormation 經建立資源，請在連接並探索範例 EC2 執行個體和資料庫執行個體後刪除 CloudFormation 堆疊，這樣您就不再需要支付費用。

若要刪除資 CloudFormation 源

1. 開啟主 AWS CloudFormation 控制台。
2. 在的 [堆疊] 頁面上 CloudFormationconsole，選取根堆疊 (沒有名稱為 vPCStack 的堆疊 BastionStack 或 RDSNS)。
3. 選擇刪除。
4. 出現確認提示時，選取「刪除堆疊」。

如需有關在中刪除堆疊的詳細資訊 CloudFormation，請參閱 [《使用指南》中的〈刪除 AWS CloudFormation 主控台上的堆疊AWS CloudFormation〉](#)。

(選用) 將資料庫執行個體連線至 Lambda 函數

您也可以將 RDS for MariaDB 資料庫執行個體連線至 Lambda 無伺服器運算資源。Lambda 函數允許您在沒有佈建或管理基礎設施的情況下執行程式碼。Lambda 函數還允許您自動回應任何規模的程式碼執行請求，從每天十幾個事件到每秒數百個事件。如需更多詳細資訊，請參閱 [自動連線 Lambda 函數和資料庫執行個體](#)。

建立並連線至 Microsoft SQL Server 資料庫執行個體

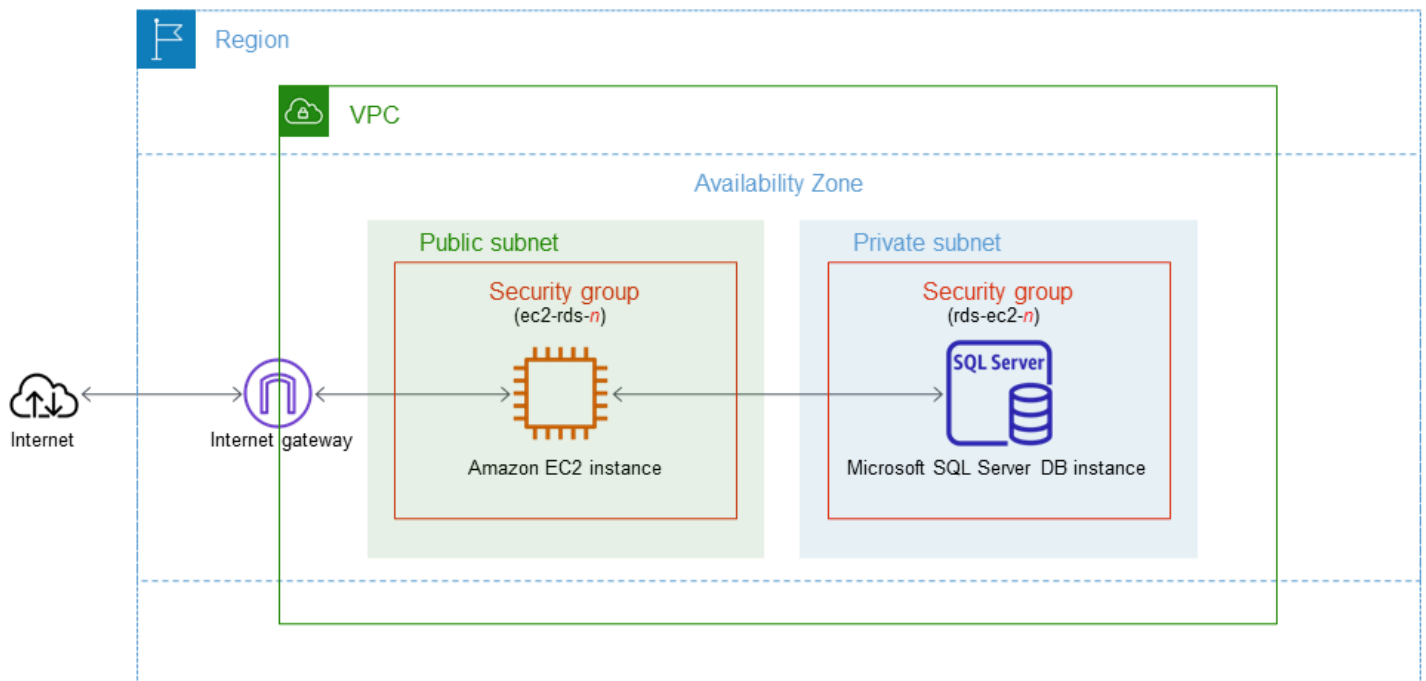
本教學課程會建立 EC2 執行個體和 RDS for Microsoft SQL Server 資料庫執行個體。本教學課程說明如何使用 Microsoft SQL Server Management Studio 用戶端，從 EC2 執行個體存取資料庫執行個體。本教學課程為最佳實務，會在虛擬私有雲端 (VPC) 中建立私有資料庫執行個體。在多數情況下，相同 VPC 中的其他資源 (例如 EC2 執行個體) 可以存取該資料庫執行個體，但 VPC 以外的資源便無法存取該執行個體。

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。在可用區域中，EC2 執行個體位於公有子網路中，而資料庫執行個體則可於私有子網路中。

⚠ Important

創建 AWS 帳戶無需支付任何費用。但是，完成此教學課程後，您可能會對使用的 AWS 資源產生費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

下圖顯示此教學課程完成時的組態。



本教學課程可讓您使用下列其中一種方法來建立資源：

1. 使用 AWS Management Console [-步驟 2：建立 SQL Server 資料庫執行個體](#) 和 [步驟 1：建立 EC2 執行個體](#)

2. 用 AWS CloudFormation 於建立資料庫執行個體和 EC2 執行個體-(選擇性) 使用建立 VPC 人雲端、EC2 執行個體和 SQL 伺服器執行個體 AWS CloudFormation

第一種方法使用輕鬆建立，透過 AWS Management Console。在這裡，您只指定資料庫引擎類型、資料庫執行個體大小和資料庫執行個體識別碼。Easy Create (輕鬆建立) 會使用其他組態選項的預設設定。

改為使用標準建立時，您可以在建立資料庫執行個體時指定更多組態選項。這些選項包括可用性、安全性、備份和維護的設定。若要建立公有資料庫執行個體，您必須使用標準建立。如需相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

主題

- [必要條件](#)
- [步驟 1：建立 EC2 執行個體](#)
- [步驟 2：建立 SQL Server 資料庫執行個體](#)
- [\(選擇性\) 使用建立 VPC 人雲端、EC2 執行個體和 SQL 伺服器執行個體 AWS CloudFormation](#)
- [步驟 3：連線至 SQL Server 資料庫執行個體](#)
- [步驟 4：探索範例 SQL Server 資料庫執行個體](#)
- [步驟 5：刪除 EC2 執行個體和資料庫執行個體](#)
- [\(選擇性\) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation](#)
- [\(選用\) 將資料庫執行個體連線至 Lambda 函數](#)

必要條件

在開始之前，請先完成下節所含步驟：

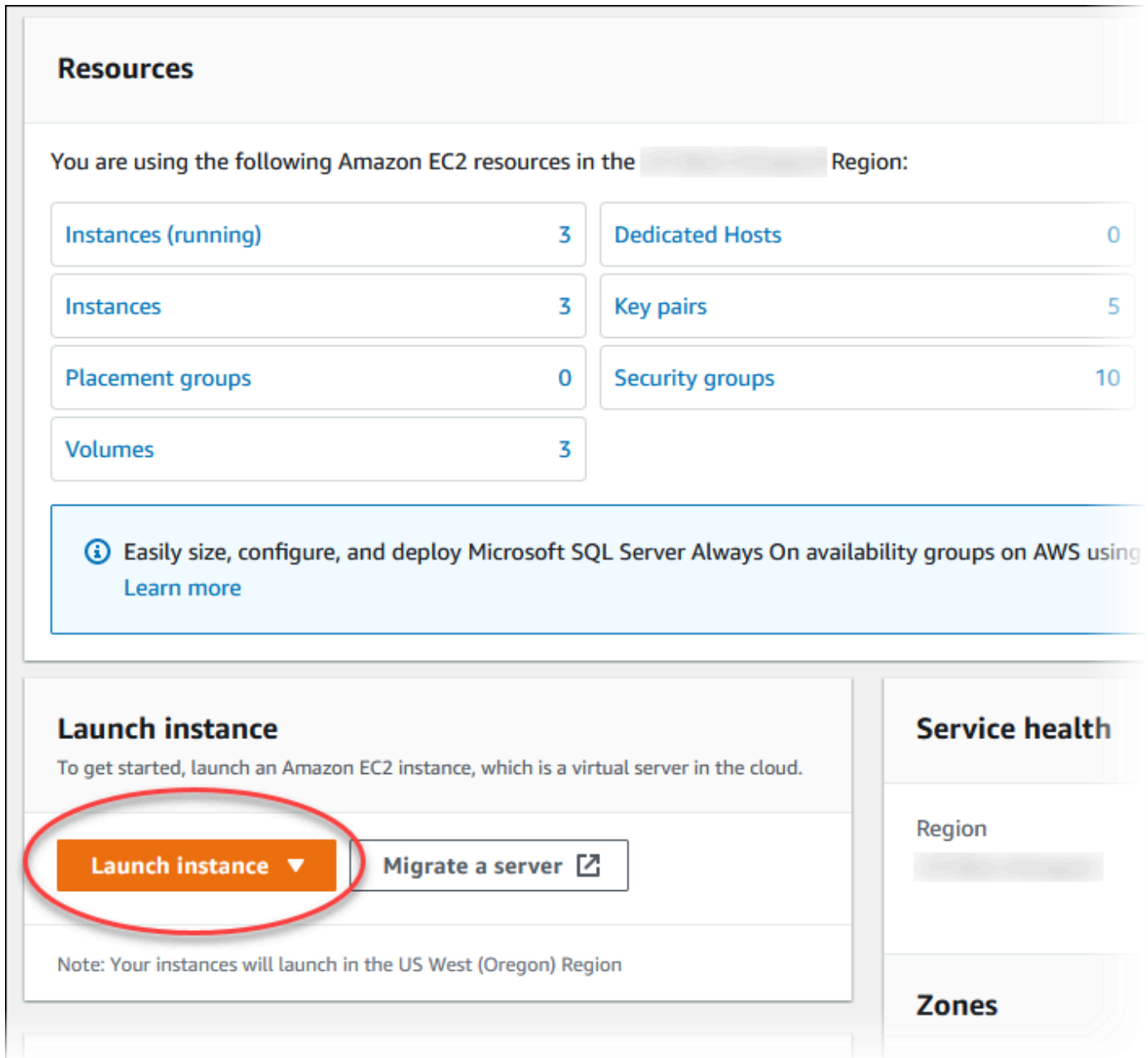
- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

步驟 1：建立 EC2 執行個體

建立您會用來連線至資料庫的 Amazon EC2 執行個體。

建立 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在的右上角 AWS Management Console，選擇 AWS 區域 您先前用於資料庫的。
3. 選擇 EC2 儀表板，然後選擇啟動執行個體，如下圖所示。



啟動執行個體頁面即開啟。

4. 在啟動執行個體頁面中選擇下列設定。

- a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **ec2-database-connect**。
- b. 在應用程式和作業系統映像 (Amazon Machine Image) 下，選擇 Windows，然後選擇 Microsoft Windows Server 2022 Base。保留其他選項的預設選擇。

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux | macOS | Ubuntu | **Windows** | Red Hat | S

aws | Mac | ubuntu® | Microsoft | Red Hat

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Microsoft Windows Server 2022 Base Free tier eligible

ami-039965e18092d85cb (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Microsoft Windows Server 2022 Full Locale English AMI provided by Amazon

Architecture	AMI ID	
64-bit (x86)	ami-039965e18092d85cb	Verified provider


- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需有關建立新的金鑰對的詳細資訊，請參閱《Windows 執行個體的 Amazon EC2 使用者指南》中的[建立金鑰對](#)。

- e. 對於網路設定中的防火牆 (安全群組)，選擇允許 RDP 流量來自以連線至 EC2 執行個體。

如果顯示的 IP 地址對 RDP 連線而言是正確的，您可以選擇我的 IP。否則，您可以決定用來使用 RDP 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 192.0.2.1/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

 Warning

如果您使用 0.0.0.0/0 進行 RDP 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 RDP 存取您的 EC2 執行個體。

下圖顯示網路設定區段的範例。

▼ **Network settings** [Info](#) Edit

Network [Info](#)
vpc-1a2b3c4d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

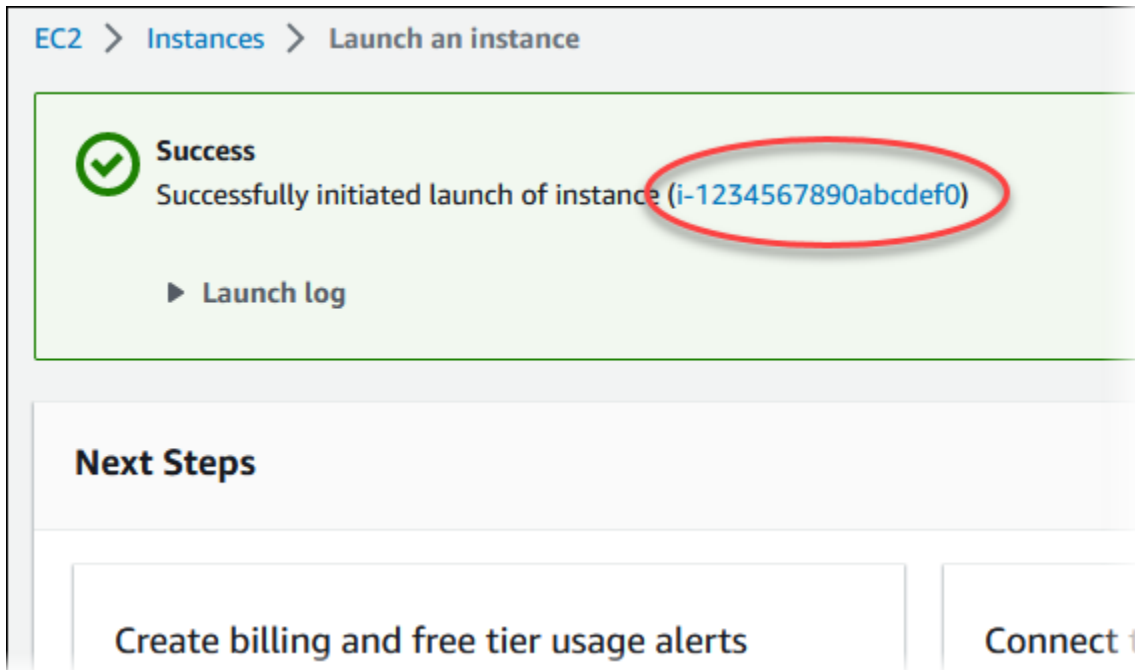
We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow RDP traffic from My IP
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

- f. 讓剩餘區段保留預設值。
 - g. 檢閱摘要面板中 EC2 執行個體組態的摘要，並在準備就緒時選擇啟動執行個體。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：i-1234567890abcdef0。



6. 選擇 EC2 執行個體識別碼，以開啟 EC2 執行個體清單。
7. 請等待 EC2 執行個體的執行個體狀態為執行中，然後再繼續動作。

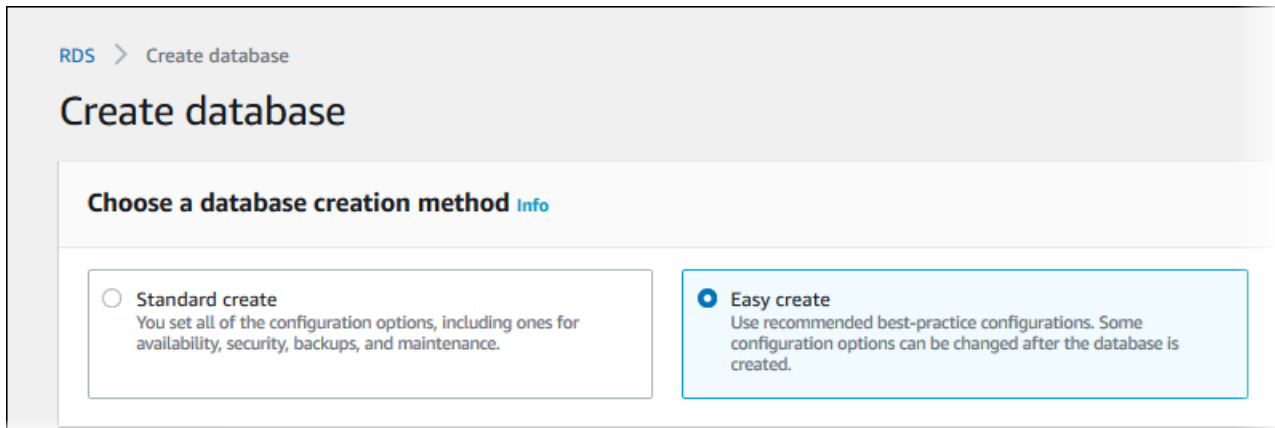
步驟 2：建立 SQL Server 資料庫執行個體

Amazon RDS 的基本建置組塊為資料庫執行個體。這是您執行 SQL Server 資料庫的環境。

在此範例中，您使用輕鬆建立來建立執行 SQL Server 資料庫引擎、資料庫執行個體類別為 db.t2.micro 的資料庫執行個體。

用 Easy Create (輕鬆建立) 建立 Microsoft SQL Server 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇您要 AWS 區域 在其中建立資料庫執行個體的執行個體。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)，並確定選擇 Easy Create (輕鬆建立)。





5. 在 Configuration (組態) 中，選擇 Microsoft SQL Server (Microsoft SQL Server)。
6. 針對版本，選擇 SQL Server Express 版本。
7. 在 DB instance size (資料庫執行個體大小) 中，選擇 Free tier (免費方案)。
8. 對於 DB instance identifier (資料庫執行個體識別碼)，請輸入 **database-test1**。


Create database (建立資料庫) 頁面看起來應該會如下圖所示。


Configuration


Engine type [Info](#)


Aurora (MySQL Compatible)


Aurora (PostgreSQL Compatible)


MySQL


MariaDB


PostgreSQL


Microsoft SQL Server


Edition

- SQL Server Express Edition**
Affordable database management system that supports database sizes up to 10 GB.
- SQL Server Web Edition**
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
- SQL Server Standard Edition**
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
- SQL Server Enterprise Edition**
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

DB instance size

Production
 db.r5.xlarge
 4 vCPUs
 32 GiB RAM
 500 GiB

Dev/Test
 db.m5.large
 2 vCPUs
 8 GiB RAM
 100 GiB

Free tier
 db.t2.micro
 1 vCPUs
 1 GiB RAM
 20 GiB

DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

9. 在主要使用者名稱，輸入主要使用者的名稱，或保留預設名稱。
10. 若要設定與先前建立之 EC2 執行個體的連線，請開啟設定 EC2 連線 – 選用。

選取連線至 EC2 運算資源。選擇先前建立的 EC2 執行個體。

▼ Set up EC2 connection - *optional*

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-
i-1234567890abcdef0



11. 如要將自動產生的主要密碼用在資料庫執行個體，請選取 Auto generate a password (自動產生密碼) 方塊。

如要輸入您的主要密碼，請清除 Auto generate a password (自動產生密碼) 方塊，然後在 Master password (主要密碼) 和 Confirm password (確認密碼) 中輸入相同的密碼。

12. 開啟檢視輕鬆建立的預設設定。

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

Configuration ▼	Value	Editable after database is created ▲
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:sqlserver-ex-14-00	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	1433	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.00.3451.2.v1	Yes
DB parameter group	default.sqlserver-ex-14.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

您可以檢查與 Easy Create (輕鬆建立) 一起使用的預設設定。資料庫建立後可編輯欄顯示您可以在資料庫建立後變更的選項。

- 若該欄的設定為否，而您想要其他設定，可以使用標準建立來建立資料庫執行個體。

- 若該欄的設定為是，而您想要其他設定，可以使用標準建立來建立資料庫執行個體，或在建立後修改該資料庫執行個體的設定。

13. 選擇建立資料庫。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。


您可以使用出現的使用者名稱和密碼，來以主要使用者的身分連線至資料庫執行個體。

Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，您可以將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

14. 在資料庫清單中，選擇新 SQL Server 資料庫執行個體的名稱，以顯示其詳細資訊。

在其可供使用之前，資料庫執行個體會處於建立中狀態。

Summary			
DB identifier database-test1	CPU -	Status  Creating	Class db.t2.micro
Role Instance	Current activity	Engine SQL Server Express Edition	Region & AZ us-east-1c

狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。

(選擇性) 使用建立 VPC 人雲端、EC2 執行個體和 SQL 伺服器執行個體 AWS CloudFormation

您可以使用將基礎設施視為程式碼來佈建 AWS 資源，而不是使用 AWS CloudFormation 主控台建立 VPC、EC2 執行個體和 SQL Server 執行個體。為了幫助您將 AWS 資源組織成更小且更易於管理的單元，您可以使用 AWS CloudFormation 嵌套堆棧功能。如需詳細資訊，請參閱[在 AWS CloudFormation 主控台上建立堆疊](#)和[使用巢狀堆疊](#)。

⚠ Important

AWS CloudFormation 是免費的，但 CloudFormation 創建的資源是活的。您必須支付這些資源的標準使用費，直到您終止這些資源為止。總計費用會很少。如需如何將任何費用降至最低的相關資訊，請前往[AWS 免費方案](#)。

若要使用 AWS CloudFormation 主控台建立資源，請完成以下步驟：

- 步驟 1：下載 CloudFormation 範本
- 步驟 2：使用 CloudFormation

下載 CloudFormation 範本

CloudFormation 範本是 JSON 或 YAML 文字檔案，其中包含您要在堆疊中建立之資源的相關設定資訊。此範本也會為您建立 VPC 和防禦主機，以及 RDS 執行個體。

若要下載範本檔案，請開啟下列連結，[SQL Server CloudFormation 範本](#)。

在 Github 頁面中，單擊下載原始文件按鈕以保存模板 YAML 文件。

使用 CloudFormation


📘 Note

在開始此程序之前，請確定您的 AWS 帳戶。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

使用 AWS CloudFormation 範本時，您必須選取正確的參數，以確保資源已正確建立。請遵循下列步驟：

1. 請登入 AWS Management Console 並開啟 AWS CloudFormation 主控台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 選擇 Create Stack (建立堆疊)。
3. 在 [指定範本] 區段中，選取 [從電腦上傳範本檔案]，然後選擇 [下一步]。
4. 在「指定堆疊詳細資訊」頁面中，設定下列參數：

- a. 將堆棧名稱設置為 SQL ServerTestStack。
- b. 在參數下，透過選取三個可用區域來設定可用區域。
- c. 在 Linux 防禦主機組態下，對於金鑰名稱，選取要登入 EC2 執行個體的 key pair。
- d. 在 Linux 防禦主機組態設定中，將允許的 IP 範圍設定為您的 IP 位址。若要使用安全殼層 (SSH) 連線至 VPC 中的 EC2 執行個體，請使用 <https://checkip.amazonaws.com> 的服務判斷您的公有 IP 位址。IP 地址的範例為 192.0.2.1/32。

 Warning

如果您使用 0.0.0.0/0 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

- e. 在「資料庫一般組態」下，將「資料庫」執行個體類別設定為 db.t3.micro。
 - f. 將資料庫名稱設定為 **database-test1**。
 - g. 在資料庫主要使用者名稱中，輸入主要使用者的名稱。
 - h. 將使用密碼管理 Secrets Manager 管理資料庫主要使用者密碼設定 false 為此教學課程。
 - i. 對於資料庫密碼，請設定您選擇的密碼。請記住此密碼，以取得教學課程的進一步步驟。
 - j. 在 [資料庫儲存設定] 下，將 [資料庫儲存類型] 設定為 gp 2。
 - k. 在 [資料庫監視組態] 下，將 [啟用 RDS Performance Insights] 設定為 false。
 - l. 保留所有其他設定為預設值。按一下「下一步」繼續。
5. 在 [設定堆疊選項] 頁面中，保留所有預設選項。按一下「下一步」繼續。
 6. 在「複查堆疊」頁面中，選取檢查資料庫和 Linux 防禦主機選項後送出。

堆疊建立程序完成後，檢視包含名稱 BastionStack 和 RDSNS 的堆疊，以記下連線至資料庫所需的資訊。如需詳細資訊，請參閱 [在上檢視 AWS CloudFormation 堆疊資料和資源 AWS Management Console](#)。

步驟 3：連線至 SQL Server 資料庫執行個體

在下列程序中，您會使用 Microsoft SQL Server Management Studio (SSMS) 連線至您的資料庫執行個體。

使用 SSMS 連線至 RDS for SQL Server 資料庫執行個體

1. 尋找資料庫執行個體的端點 (DNS 名稱) 和連接埠號碼。
 - a. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 - b. 在 Amazon RDS 主控台的右上角，選擇 AWS 區域 適用於資料庫執行個體的。
 - c. 在導覽窗格中，選擇 Databases (資料庫)。
 - d. 選擇 SQL Server 資料庫執行個體名稱以顯示其詳細資訊。
 - e. 在 Connectivity (連線) 標籤上，複製端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

RDS > Databases > database-test1

database-test1

Summary

DB identifier database-test1	CPU 2.95%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events

Connectivity & security

Endpoint & port	Networking
Endpoint database-test1.0123456789012.us-west-2.rds.amazonaws.com	Availability Zone
Port 1433	VPC vpc-
	Subnet group default-vpc-

- 請遵循《Amazon EC2 Windows 執行個體使用者指南》中的[連線至您的 Microsoft Windows 執行個體](#)，以連線至您先前建立的 EC2 執行個體。
- 安裝來自 Microsoft 的 SQL Server Management Studio (SSMS) 用戶端。

若要將 SSMS 的獨立版本下載至您的 EC2 執行個體，請參閱 Microsoft 文件中的[下載 SQL Server Management Studio \(SSMS\)](#)。

- 使用「開始」功能表開啟 Internet Explorer。

- b. 使用 Internet Explorer 下載並安裝 SSMS 的獨立版本。如果系統提示您網站不受信任，請將該網站新增至信任的網站清單。
4. 啟動 SQL Server Management Studio (SSMS)。

Connect to Server (連線至伺服器) 對話方塊隨即出現。

5. 為範例資料庫執行個體提供下列資訊：
 - a. 針對 Server type (伺服器類型)，選擇 Database Engine (資料庫引擎)。
 - b. 針對 Server name (伺服器名稱)，輸入 DNS 名稱，後接逗號和連接埠號碼 (預設連接埠為 1433)。例如，您的伺服器名稱應看起來如下所示。

`database-test1.0123456789012.us-west-2.rds.amazonaws.com,1433`
 - c. 針對 Authentication (身分驗證)，選擇 SQL Server Authentication (SQL Server 身分驗證)。
 - d. 針對登入，輸入您選擇用於範例資料庫執行個體的使用者名稱。這也稱之為主要使用者名稱。
 - e. 在 Password (密碼) 中，輸入您稍早為範例資料庫執行個體選擇的密碼。這也稱之為主要使用者密碼。
6. 選擇 Connect (連線)。

幾分鐘後，SSMS 會連線至資料庫執行個體。基於安全考量，最佳做法是使用加密連線。僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 SQL Server 連線。如需使用加密連線的詳細資訊，請參閱 [對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)

如需有關連線至 Microsoft SQL Server 資料庫執行個體的詳細資訊，請參閱 [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。

如需連線問題的詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

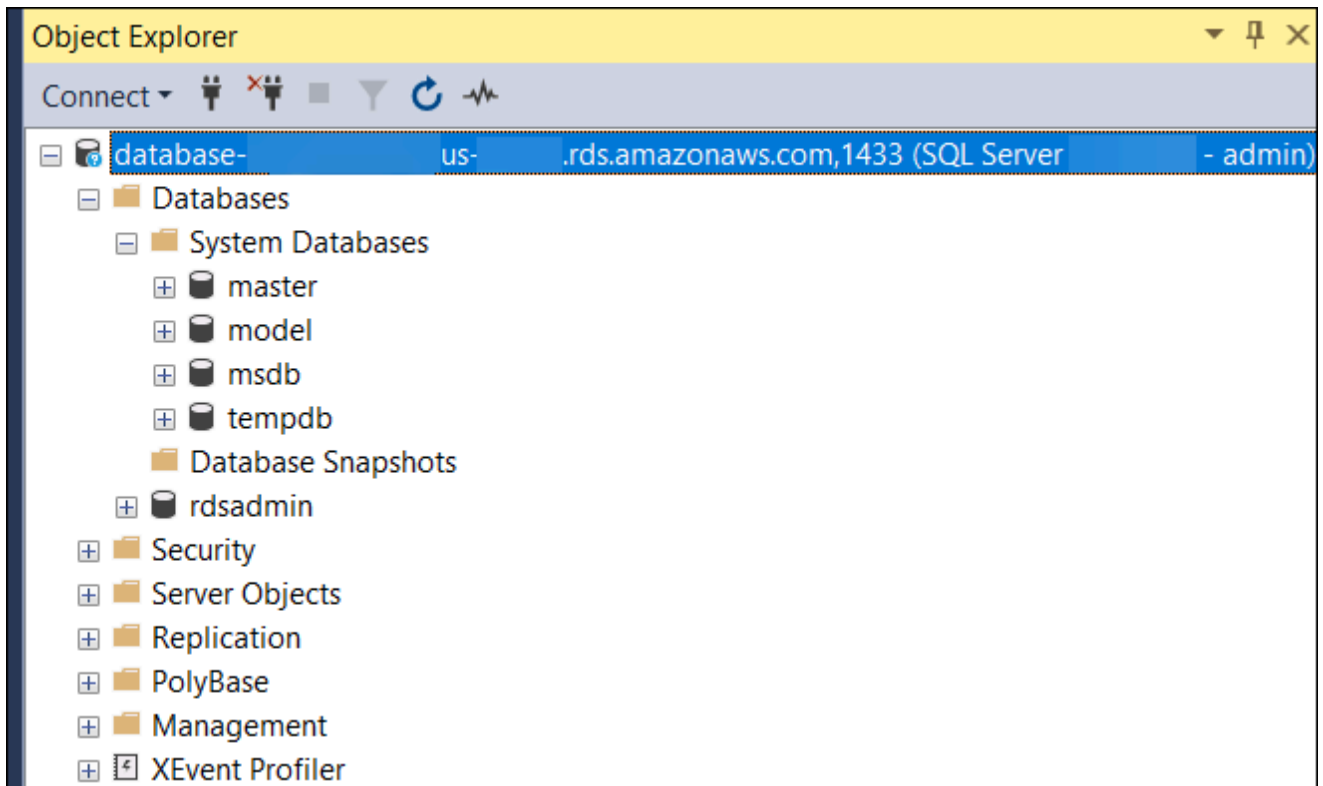
步驟 4：探索範例 SQL Server 資料庫執行個體

您可以使用 Microsoft SQL Server Management Studio (SSMS) 來探索範例資料庫執行個體。

使用 SSMS 來探索資料庫執行個體

1. 您的 SQL Server 資料庫執行個體隨附 SQL Server 的標準內建系統資料庫 (master、model、msdb 和 tempdb)。若要瀏覽系統資料庫，請執行下列動作：
 - a. 在 SSMS 的 View (檢視) 功能表中，選擇 Object Explorer (物件總管)。

- b. 展開您的資料庫執行個體、展開 Databases (資料庫)，再展開 System Databases (系統資料庫)，如下所示。



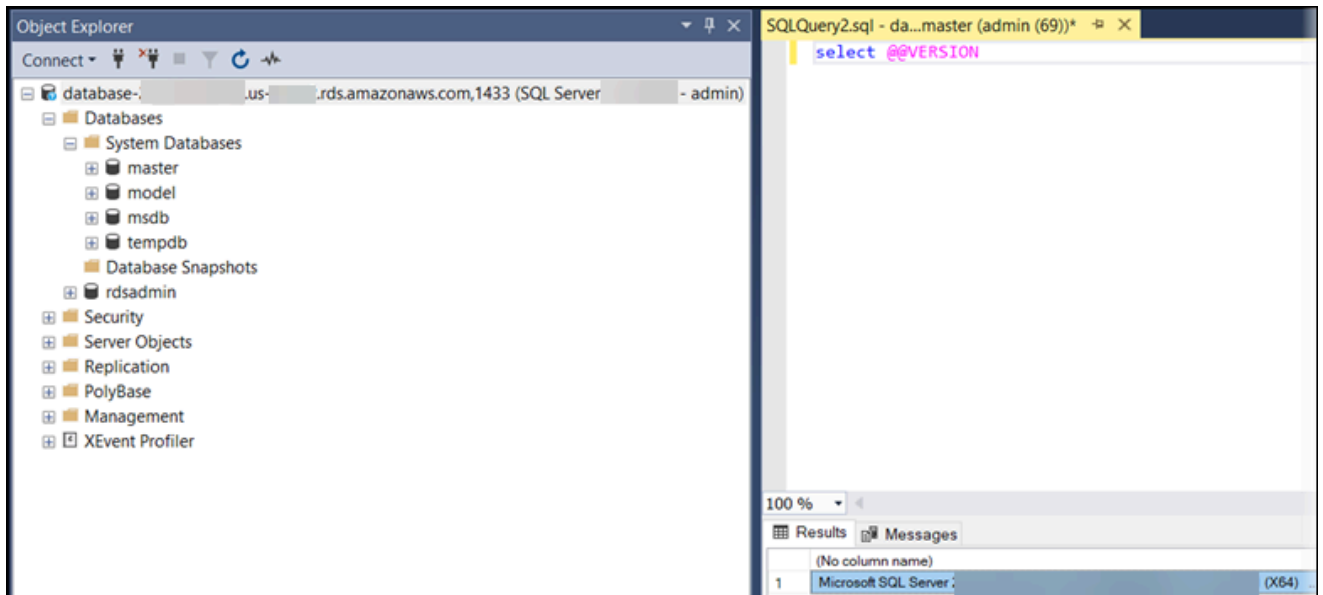
您的 SQL Server 資料庫執行個體也隨附一個名為 `rdsadmin` 的資料庫。Amazon RDS 將使用此資料庫，以便存放用於管理資料庫的物件。`rdsadmin` 資料庫也包含可供您執行的預存程序，可讓您執行進階任務。

2. 開始建立您自己的資料庫，並照常對您的資料庫執行個體和資料庫執行查詢。若要在範例資料庫執行個體上執行測試查詢，請執行下列動作：

- a. 在 SSMS 中，於檔案功能表上，指向新增，然後選擇使用目前的連線查詢。
- b. 輸入下列 SQL 查詢：

```
select @@VERSION
```

- c. 執行查詢。SSMS 會傳回 Amazon RDS 資料庫執行個體的 SQL Server 版本。



步驟 5：刪除 EC2 執行個體和資料庫執行個體

在連線至您已建立的範例 EC2 執行個體與資料庫執行個體，並探索這些執行個體之後，請將其刪除，才不會再對您收費。

如果您曾經 AWS CloudFormation 建立資源，請略過此步驟並前往下一個步驟。

刪除 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，選擇執行個體。
3. 選取 EC2 執行個體，並選取執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇 Terminate (終止)。

如需刪除 EC2 執行個體的詳細資訊，請參閱《Windows 執行個體使用者指南》中的[終止執行個體](#)。

刪除不含最終資料庫快照的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要刪除的資料庫執行個體。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 清除是否建立最後的快照？和保留自動備份。
6. 完成確認，然後選擇刪除。

(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation

如果您曾 AWS CloudFormation 經建立資源，請在連接並探索範例 EC2 執行個體和資料庫執行個體後刪除 CloudFormation 堆疊，這樣您就不再需要支付費用。

若要刪除資 CloudFormation 源

1. 開啟主 AWS CloudFormation 控制台。
2. 在的 [堆疊] 頁面上 CloudFormationconsole，選取根堆疊 (沒有名稱為 vPCStack 的堆疊 BastionStack 或 RDSNS)。
3. 選擇刪除。
4. 出現確認提示時，選取「刪除堆疊」。

如需有關在中刪除堆疊的詳細資訊 CloudFormation，請參閱《[使用指南](#)》中的〈[刪除 AWS CloudFormation 主控台上的堆疊AWS CloudFormation](#)〉。

(選用) 將資料庫執行個體連線至 Lambda 函數

您也可以將適用於 RDS for SQL Server 資料庫執行個體連線至 Lambda 無伺服器運算資源。Lambda 函數允許您在沒有佈建或管理基礎設施的情況下執行程式碼。Lambda 函數還允許您自動回應任何規模的程式碼執行請求，從每天十幾個事件到每秒數百個事件。如需更多詳細資訊，請參閱 [自動連線 Lambda 函數和資料庫執行個體](#)。

建立並連線至 MySQL 資料庫執行個體

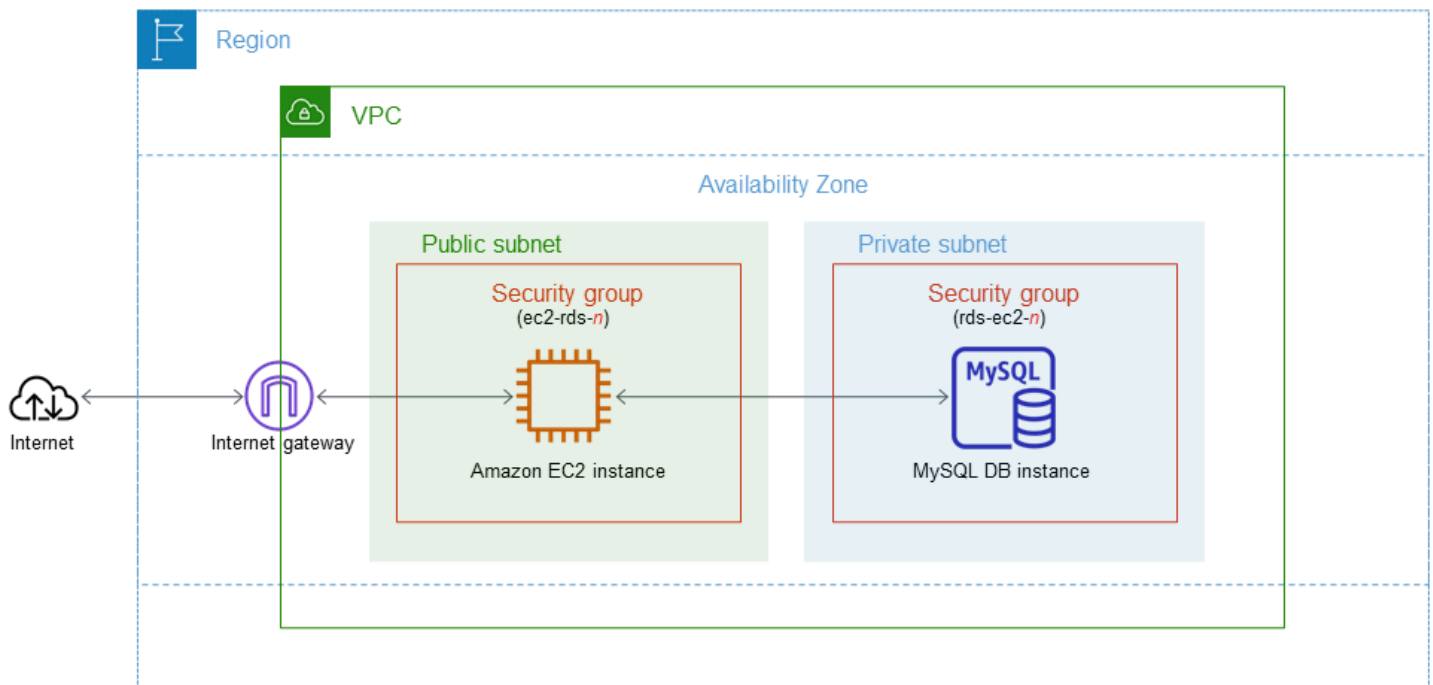
本教學課程會建立 EC2 執行個體和 RDS for MySQL 資料庫執行個體。本教學課程說明如何使用標準 MySQL 用戶端，從 EC2 執行個體存取資料庫執行個體。本教學課程為最佳實務，會在虛擬私有雲端 (VPC) 中建立私有資料庫執行個體。在多數情況下，相同 VPC 中的其他資源 (例如 EC2 執行個體) 可以存取該資料庫執行個體，但 VPC 以外的資源便無法存取該執行個體。

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。在可用區域中，EC2 執行個體位於公有子網路中，而資料庫執行個體則可於私有子網路中。

⚠ Important

創建 AWS 帳戶無需支付任何費用。但是，完成此教學課程後，您可能會對使用的 AWS 資源產生費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

下圖顯示此教學課程完成時的組態。



本教學課程可讓您使用下列其中一種方法來建立資源：

1. 使用 AWS Management Console - [步驟 2：建立 MySQL 資料庫執行個體](#) 和 [步驟 1：建立 EC2 執行個體](#)

2. 用 AWS CloudFormation 於建立資料庫執行個體和 EC2 執行個體-[\(選擇性\) 使用建立 VPC、EC2 執行個體和 MySQL 執行個體 AWS CloudFormation](#)

第一種方法使用輕鬆創建來創建一個私有 MySQL 數據庫實例 AWS Management Console。在這裡，您只指定資料庫引擎類型、資料庫執行個體大小和資料庫執行個體識別碼。Easy Create (輕鬆建立) 會使用其他組態選項的預設設定。

改為使用標準建立時，您可以在建立資料庫執行個體時指定更多組態選項。這些選項包括可用性、安全性、備份和維護的設定。若要建立公有資料庫執行個體，您必須使用標準建立。如需相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

主題

- [必要條件](#)
- [步驟 1：建立 EC2 執行個體](#)
- [步驟 2：建立 MySQL 資料庫執行個體](#)
- [\(選擇性\) 使用建立 VPC、EC2 執行個體和 MySQL 執行個體 AWS CloudFormation](#)
- [步驟 3：連線至 MySQL 資料庫執行個體](#)
- [步驟 4：刪除 EC2 執行個體和資料庫執行個體](#)
- [\(選擇性\) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation](#)
- [\(選用\) 將資料庫執行個體連線至 Lambda 函數](#)

必要條件

在開始之前，請先完成下節所含步驟：

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

步驟 1：建立 EC2 執行個體

建立您會用來連線至資料庫的 Amazon EC2 執行個體。

建立 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。

2. 在的右上角 AWS Management Console，選擇您要 AWS 區域 在其中建立 EC2 執行個體的。
3. 選擇 EC2 儀表板，然後選擇啟動執行個體，如下圖所示。

Resources

You are using the following Amazon EC2 resources in the Region:

Instances (running)	3	Dedicated Hosts	0
Instances	3	Key pairs	5
Placement groups	0	Security groups	10
Volumes	3		

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼ **Migrate a server** ↗

Note: Your instances will launch in the US West (Oregon) Region

Service health

Region

Zones

啟動執行個體頁面即開啟。

4. 在啟動執行個體頁面中選擇下列設定。
 - a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **ec2-database-connect**。
 - b. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇 Amazon Linux，然後選擇 Amazon Linux 2023 AMI。保留其他選項的預設選擇。

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0efa651876de2a5ce (64-bit (x86), uefi-preferred) / ami-0699f753302dd8b00 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0efa651876de2a5ce

Verified provider

- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需有關建立新 key pair 的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [建立 key pair](#)。

- e. 對於網路設定中的允許 SSH 流量，選擇 EC2 執行個體的 SSH 連線來源。

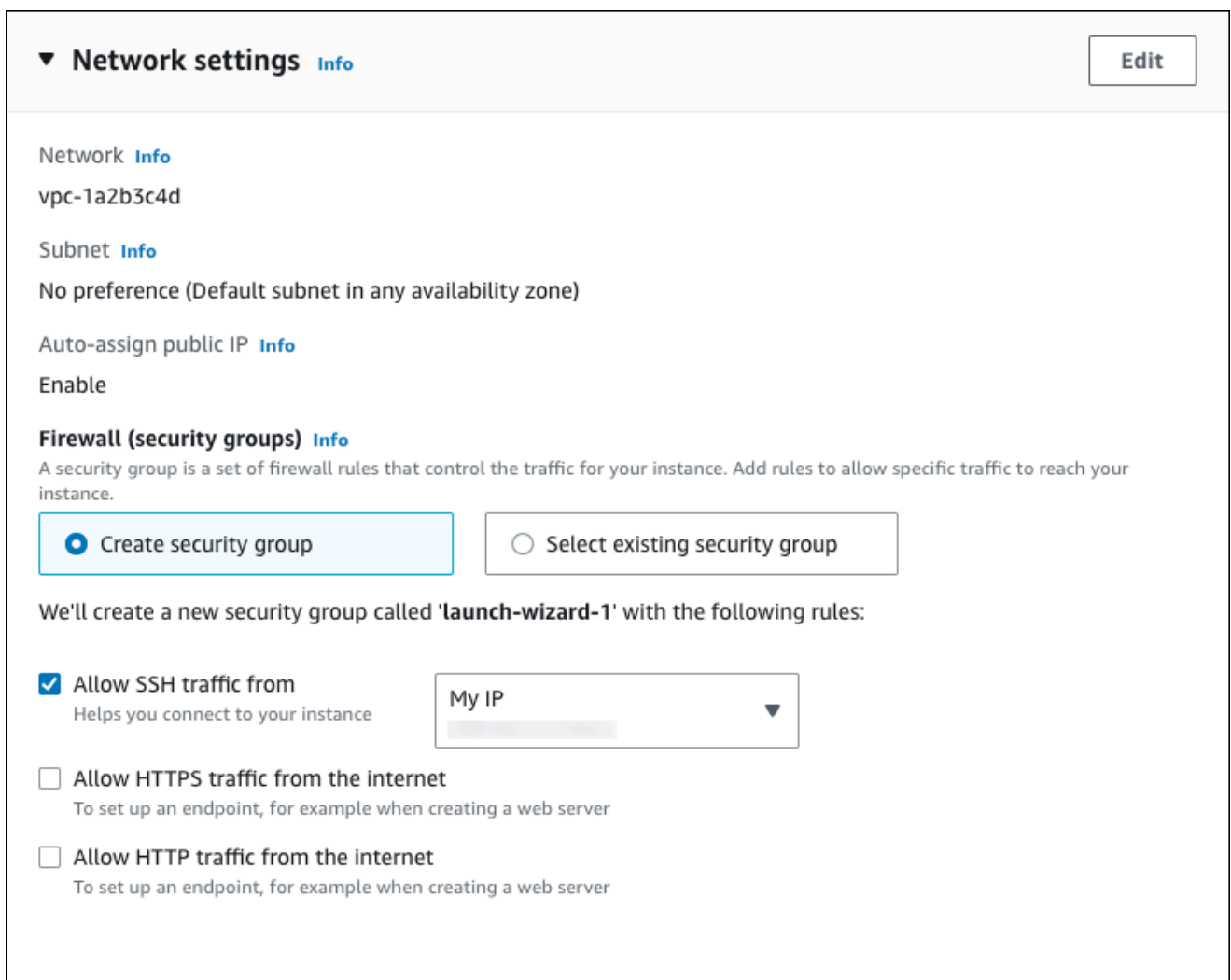
如果顯示的 IP 地址對 SSH 連線而言是正確的，您可以選擇 My IP (我的 IP)。否則，您可以決定用於使用 Secure Shell (SSH) 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 192.0.2.1/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

下圖顯示網路設定區段的範例。



▼ Network settings [Info](#) Edit

Network [Info](#)
vpc-1a2b3c4d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called **'launch-wizard-1'** with the following rules:

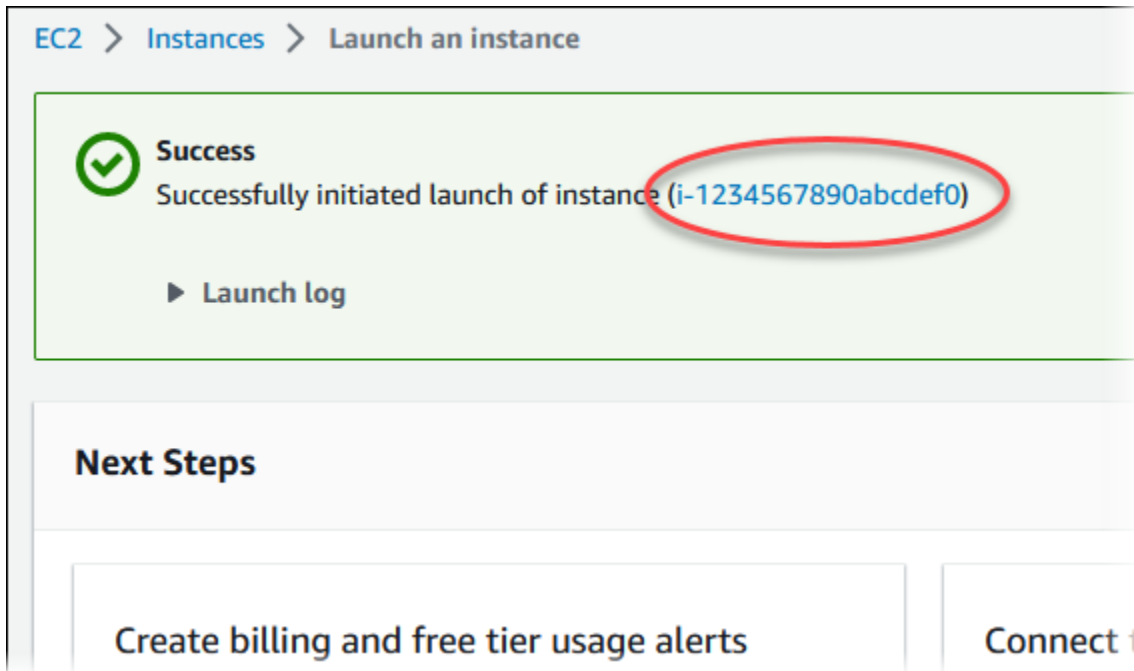
Allow SSH traffic from My IP
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

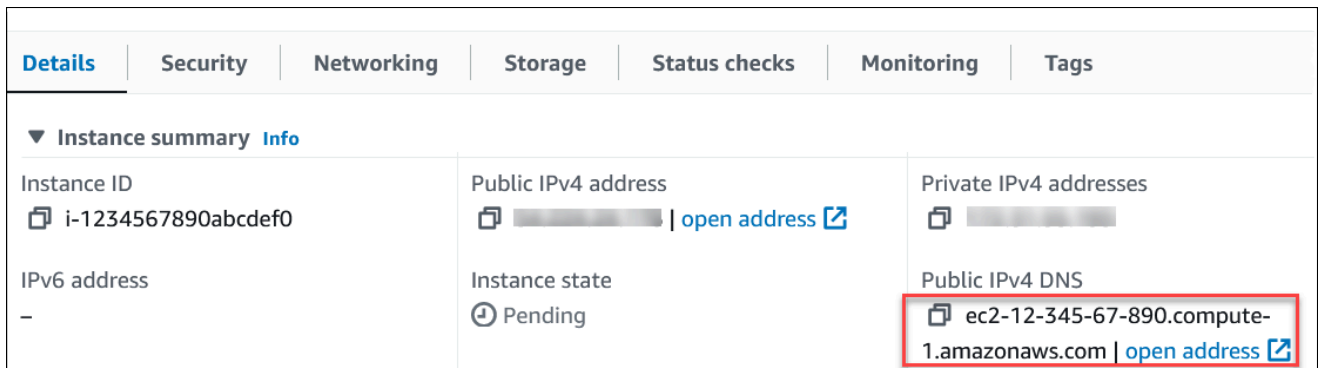
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

f. 讓剩餘區段保留預設值。


- g. 檢閱摘要面板中 EC2 執行個體組態的摘要，並在準備就緒時選擇啟動執行個體。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：`i-1234567890abcdef0`。



6. 選擇 EC2 執行個體識別符，以開啟 EC2 執行個體清單，然後選取您的 EC2 執行個體。
7. 在詳細資訊索引標籤中，請記下以下值，當您使用 SSH 進行連線時需要這些值：
 - a. 在執行個體摘要中，記下公用 IPv4 DNS 的值。



- b. 在執行個體詳細資訊中，記下金鑰對名稱的值。

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name  ec2-database-connect-key-pair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -

8. 請等待 EC2 執行個體的執行個體狀態為執行中，然後再繼續動作。

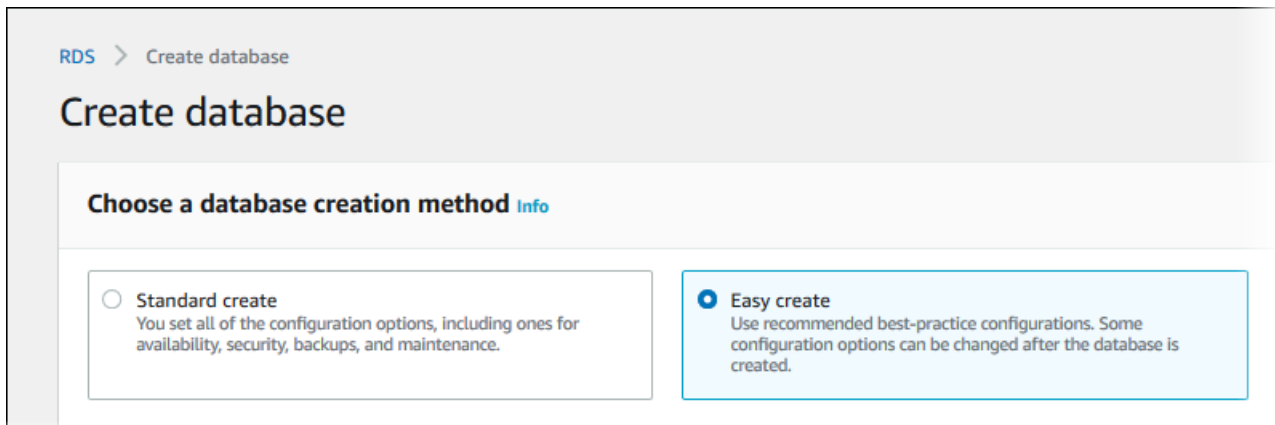
步驟 2：建立 MySQL 資料庫執行個體

Amazon RDS 的基本建置組塊為資料庫執行個體。這是您執行 MySQL 資料庫的環境。

於此範例中，您使用輕鬆建立來建立執行 MySQL 資料庫引擎、資料庫執行個體類別為 db.t3.micro 的資料庫執行個體。

如要以 Easy create (輕鬆建立) 來建立 MySQL 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇 AWS 區域 您先前用於 EC2 執行個體的執行個體。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)，並確定選擇 Easy Create (輕鬆建立)。










5. 在 Configuration (組態)，選擇 MySQL (MySQL)。
6. 在 DB instance size (資料庫執行個體大小) 中，選擇 Free tier (免費方案)。
7. 對於 DB instance identifier (資料庫執行個體識別碼)，請輸入 **database-test1**。

8. 在主要使用者名稱，輸入主要使用者的名稱，或保留預設名稱。

Create database (建立資料庫) 頁面看起來應該會如下圖所示。

Configuration

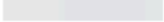
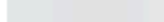
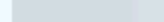
Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 	<input checked="" type="radio"/> MySQL 
<input type="radio"/> MariaDB 	<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 		

Edition

MySQL Community

DB instance size

<input type="radio"/> Production db.r6g.xlarge 4 vCPUs 32 GiB RAM 500 GiB 	<input type="radio"/> Dev/Test db.r6g.large 2 vCPUs 16 GiB RAM 100 GiB 	<input checked="" type="radio"/> Free tier db.t3.micro 2 vCPUs 1 GiB RAM 20 GiB 
--	---	---

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

9. 如要將自動產生的主要密碼用在資料庫執行個體，請選取自動產生密碼。

如要輸入您的主要密碼，請確認清除自動產生密碼方塊，然後在主要密碼和確認密碼中輸入相同的密碼。

10. 若要設定與先前建立之 EC2 執行個體的連線，請開啟設定 EC2 連線 - 選用。

選取連線至 EC2 運算資源。選擇先前建立的 EC2 執行個體。

▼ Set up EC2 connection - optional

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

Compute resource



Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-  

11. (選用) 開啟 View default settings for Easy create (檢視輕鬆建立的預設設定)。

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

Configuration ▼	Value	Editable after database is created ▲
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mysql-8-0	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0cc53de1b4d1763cf	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	8.0.28	Yes
DB parameter group	default.mysql8.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

您可以檢查與 Easy Create (輕鬆建立) 一起使用的預設設定。資料庫建立後可編輯欄顯示您可以在資料庫建立後變更的選項。

- 若該欄的設定為否，而您想要其他設定，可以使用標準建立來建立資料庫執行個體。

- 若該欄的設定為是，而您想要其他設定，可以使用標準建立來建立資料庫執行個體，或在建立後修改該資料庫執行個體的設定。

12. 選擇建立資料庫。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

您可以使用出現的使用者名稱和密碼，來以主要使用者的身分連線至資料庫執行個體。

⚠ Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，您可以將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

13. 資料庫清單中，選擇新 MySQL 資料庫執行個體的名稱，以顯示其詳細資訊。

在其可供使用之前，資料庫執行個體會處於建立中狀態。

Summary			
DB identifier database-test1	CPU -	Status 🔄 Creating	Class db.r6g.large
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-east-1c

狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。

(選擇性) 使用建立 VPC、EC2 執行個體和 MySQL 執行個體 AWS CloudFormation

您可以將基礎設施視為程式碼來佈建 AWS 資源，而不是使用 AWS CloudFormation 主控台建立 VPC、EC2 執行個體和 MySQL 執行個體。為了幫助您將 AWS 資源組織成更小且更易於管理的單元，

您可以使用 AWS CloudFormation 嵌套堆棧功能。如需詳細資訊，請參閱在 [AWS CloudFormation 主控台上建立堆疊](#)和[使用巢狀堆疊](#)。

Important

AWS CloudFormation 是免費的，但 CloudFormation 創建的資源是活的。您必須支付這些資源的標準使用費，直到您終止這些資源為止。總計費用會很少。如需如何將任何費用降至最低的相關資訊，請前往[AWS 免費方案](#)。

若要使用 AWS CloudFormation 主控台建立資源，請完成以下步驟：

- 步驟 1：下載 CloudFormation 範本
- 步驟 2：使用 CloudFormation

下載 CloudFormation 範本

CloudFormation 範本是 JSON 或 YAML 文字檔案，其中包含您要在堆疊中建立之資源的相關設定資訊。此範本也會為您建立 VPC 和防禦主機，以及 RDS 執行個體。

要下載模板文件，請打開以下鏈接，[MySQL CloudFormation 模板](#)。

在 Github 頁面中，單擊下載原始文件按鈕以保存模板 YAML 文件。

使用 CloudFormation


Note

在開始此程序之前，請確定您的 AWS 帳戶。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

使用 AWS CloudFormation 範本時，您必須選取正確的參數，以確保資源已正確建立。請遵循下列步驟：

1. 請登入 AWS Management Console 並開啟 AWS CloudFormation 主控台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 選擇 Create Stack (建立堆疊)。

3. 在 [指定範本] 區段中，選取 [從電腦上傳範本檔案]，然後選擇 [下一步]。
4. 在「指定堆疊詳細資訊」頁面中，設定下列參數：
 - a. 將堆棧名稱設置為 MySQL TestStack。
 - b. 在參數下，選取三個可用區域來設定可用區域。
 - c. 在 Linux 防禦主機組態下，對於金鑰名稱，選取要登入 EC2 執行個體的 key pair。
 - d. 在 Linux 防禦主機組態設定中，將允許的 IP 範圍設定為您的 IP 位址。若要使用安全殼層 (SSH) 連線至 VPC 中的 EC2 執行個體，請使用 <https://checkip.amazonaws.com> 的服務判斷您的公有 IP 位址。IP 地址的範例為 192.0.2.1/32。

 Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

- e. 在「資料庫一般組態」下，將「資料庫」執行個體類別設定為 db.t3.micro。
 - f. 將資料庫名稱設定為 **database-test1**。
 - g. 在資料庫主要使用者名稱中，輸入主要使用者的名稱。
 - h. 將使用密碼管理 Secrets Manager 管理資料庫主要使用者密碼設定 `false` 為此教學課程。
 - i. 對於資料庫密碼，請設定您選擇的密碼。請記住此密碼，以取得教學課程的進一步步驟。
 - j. 在 [資料庫儲存設定] 下，將 [資料庫儲存類型] 設定為 gp 2。
 - k. 在 [資料庫監視組態] 下，將 [啟用 RDS Performance Insights] 設定為 `false`。
 - l. 保留所有其他設定為預設值。按一下「下一步」繼續。
5. 在 [設定堆疊選項] 頁面中，保留所有預設選項。按一下「下一步」繼續。
 6. 在「複查堆疊」頁面中，選取檢查資料庫和 Linux 防禦主機選項後送出。

堆疊建立程序完成後，檢視包含名稱 BastionStack 和 RDSNS 的堆疊，以記下連線至資料庫所需的資訊。如需詳細資訊，請參閱 [AWS CloudFormation 在 AWS Management Console](#)。

步驟 3：連線至 MySQL 資料庫執行個體

您可以使用任何標準 SQL 用戶端應用程式來連線至資料庫執行個體。在此範例中，您會使用 `mysql` 命令列用戶端來連線至 MySQL 資料庫執行個體。

連線至 MySQL 資料庫執行個體

1. 尋找資料庫執行個體的端點 (DNS 名稱) 和連接埠號碼。
 - a. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 - b. 在 Amazon RDS 主控台的右上角，選擇 AWS 區域 適用於資料庫執行個體的。
 - c. 在導覽窗格中，選擇 Databases (資料庫)。
 - d. 選擇 MySQL 資料庫執行個體名稱以顯示其詳細資訊。
 - e. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

RDS > Databases > database-test1

database-test1

Summary

DB identifier database-test1	CPU 2.58%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com Port 3306	Networking Availability Zone us-east-1c VPC vpc- Subnet group default
---	--

2. 按照 Amazon EC2 使用者指南中的連線到 [Linux 執行個體中的步驟](#)，Connect 到先前建立的 EC2 執行個體。

建議您使用 SSH 連線至 EC2 執行個體。如果 SSH 用戶端公用程式已安裝在 Windows、Linux 或 Mac 上，您可以使用下列命令格式連線至執行個體：

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

例如，假設 `ec2-database-connect-key-pair.pem` 存放在 Linux 上的 `/dir1` 中，而 EC2 執行個體的公用 IPv4 DNS 為 `ec2-12-345-678-90.compute-1.amazonaws.com`。您的 SSH 命令看起來如下所示：

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-user@ec2-12-345-678-90.compute-1.amazonaws.com
```

- 更新 EC2 執行個體上的軟體，以取得最新的錯誤修正和安全性更新。若要執行此操作，請使用以下命令。

Note

`-y` 選項不要求確認就會安裝更新。若要先檢查更新再安裝，請省略此選項。

```
sudo dnf update -y
```

- 若要在 Amazon Linux 2023 上安裝 MariaDB 的 `mysql` 命令列用戶端，請執行下列命令：

```
sudo dnf install mariadb105
```

- 連線到 MySQL 資料庫執行個體。例如，輸入下列命令。此動作可讓您使用 MySQL 用戶端，連線至 MySQL 資料庫執行個體。

取代您用於 `endpoint` 的資料庫執行個體端點 (DNS 名稱)，以及取代您用於 `admin` 的主要使用者名稱。提示您輸入密碼時，請提供您使用的主要密碼。

```
mysql -h endpoint -P 3306 -u admin -p
```

輸入使用者的密碼之後，您應該會看到類似如下的輸出。

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3082
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MySQL [(none)]>
```

如需連線至 MySQL 資料庫執行個體的詳細資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。如果您無法連線至資料庫執行個體，請參閱[無法連線至 Amazon RDS 資料庫執行個體](#)。

基於安全考量，最佳做法是使用加密連線。僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 MySQL 連線。如需使用加密連線的詳細資訊，請參閱[使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)。

6. 執行 SQL 命令。

例如，下列 SQL 命令會顯示目前的日期和時間：

```
SELECT CURRENT_TIMESTAMP;
```

步驟 4：刪除 EC2 執行個體和資料庫執行個體

在連線至您已建立的範例 EC2 執行個體與資料庫執行個體，並探索這些執行個體之後，請將其刪除，才不會再對您收費。

如果您曾經 AWS CloudFormation 建立資源，請略過此步驟並前往下一個步驟。

刪除 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，選擇執行個體。
3. 選取 EC2 執行個體，並選取執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇 Terminate (終止)。

如需刪除 EC2 執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的[終止執行個體](#)。

刪除不含最終資料庫快照的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要刪除的資料庫執行個體。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 清除是否建立最後的快照？和保留自動備份。
6. 完成確認，然後選擇刪除。

(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation

如果您曾 AWS CloudFormation 經建立資源，請在連接並探索範例 EC2 執行個體和資料庫執行個體後刪除 CloudFormation 堆疊，這樣您就不再需要支付費用。

若要刪除資 CloudFormation 源

1. 開啟主 AWS CloudFormation 控台。
2. 在的 [堆疊] 頁面上 CloudFormationconsole，選取根堆疊 (沒有名稱為 vPCStack 的堆疊 BastionStack 或 RDSNS)。
3. 選擇刪除。
4. 出現確認提示時，選取「刪除堆疊」。

如需有關在中刪除堆疊的詳細資訊 CloudFormation，請參閱《[使用指南](#)》中的〈[刪除 AWS CloudFormation 主控台上的堆疊AWS CloudFormation](#)〉。

(選用) 將資料庫執行個體連線至 Lambda 函數

您也可以將 RDS for MySQL 資料庫執行個體連線至 Lambda 無伺服器運算資源。Lambda 函數允許您在沒有佈建或管理基礎設施的情況下執程式碼。Lambda 函數還允許您自動回應任何規模的程式碼執行請求，從每天十幾個事件到每秒數百個事件。如需更多詳細資訊，請參閱 [自動連線 Lambda 函數和資料庫執行個體](#)。

建立並連線至 Oracle 資料庫執行個體

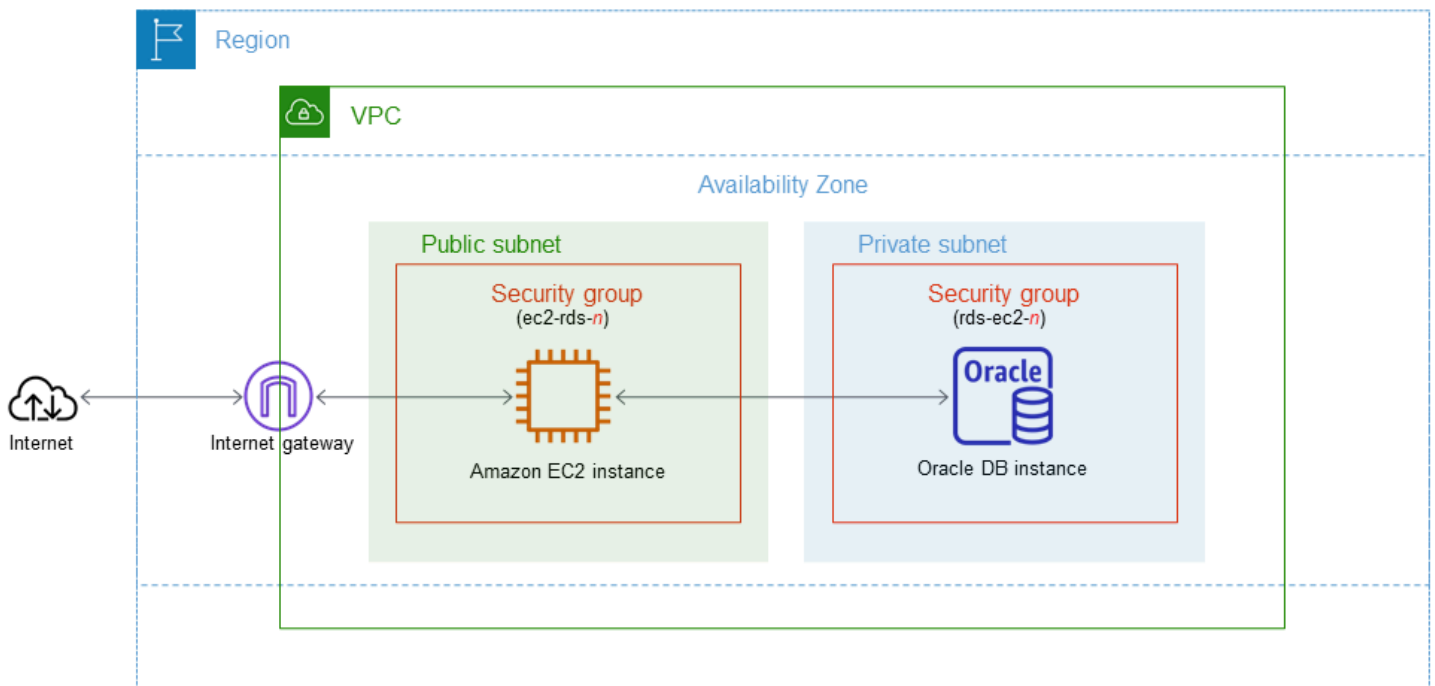
本教學課程會建立 EC2 執行個體和 RDS for Oracle 資料庫執行個體。本教學課程說明如何使用標準 Oracle 用戶端，從 EC2 執行個體存取資料庫執行個體。本教學課程為最佳實務，會在虛擬私有雲端 (VPC) 中建立私有資料庫執行個體。在多數情況下，相同 VPC 中的其他資源 (例如 EC2 執行個體) 可以存取該資料庫執行個體，但 VPC 以外的資源便無法存取該執行個體。

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。在可用區域中，EC2 執行個體位於公有子網路中，而資料庫執行個體則可於私有子網路中。

⚠ Important

建立 AWS 帳戶無須支付任何費用。但是，完成此教學課程後，您可能會對使用的 AWS 資源產生費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

下圖顯示此教學課程完成時的組態。



本教學課程可讓您使用下列其中一種方法來建立資源：

1. 使用 AWS Management Console - [步驟 2：建立 Oracle 資料庫執行個體](#) 和 [步驟 1：建立 EC2 執行個體](#)

2. 用 AWS CloudFormation 於建立資料庫執行個體和 EC2 執行個體-(選擇性) 使用建立 VPC、EC2 執行個體和 Oracle 資料庫執行個體 AWS CloudFormation

第一種方法使用「輕鬆建立」來建立私有 Oracle 資料庫執行個體 AWS Management Console。在這裡，您只指定資料庫引擎類型、資料庫執行個體大小和資料庫執行個體識別碼。Easy Create (輕鬆建立) 會使用其他組態選項的預設設定。

改為使用標準建立時，您可以在建立資料庫執行個體時指定更多組態選項。這些選項包括可用性、安全性、備份和維護的設定。若要建立公有資料庫執行個體，您必須使用標準建立。如需相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

主題

- [必要條件](#)
- [步驟 1：建立 EC2 執行個體](#)
- [步驟 2：建立 Oracle 資料庫執行個體](#)
- [\(選擇性\) 使用建立 VPC、EC2 執行個體和 Oracle 資料庫執行個體 AWS CloudFormation](#)
- [步驟 3：將您的 SQL 連線至 Oracle 資料庫執行個體](#)
- [步驟 4：刪除 EC2 執行個體和資料庫執行個體](#)
- [\(選擇性\) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation](#)
- [\(選用\) 將資料庫執行個體連線至 Lambda 函數](#)

必要條件

在開始之前，請先完成下節所含步驟：

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

步驟 1：建立 EC2 執行個體

建立您會用來連線至資料庫的 Amazon EC2 執行個體。

建立 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。

2. 在的右上角 AWS Management Console，選擇您要 AWS 區域 在其中建立 EC2 執行個體的。
3. 選擇 EC2 儀表板，然後選擇啟動執行個體，如下圖所示。

Resources

You are using the following Amazon EC2 resources in the Region:

Instances (running)	3	Dedicated Hosts	0
Instances	3	Key pairs	5
Placement groups	0	Security groups	10
Volumes	3		

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼ **Migrate a server** ↗

Note: Your instances will launch in the US West (Oregon) Region

Service health

Region

Zones

啟動執行個體頁面即開啟。

4. 在啟動執行個體頁面中選擇下列設定。
 - a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **ec2-database-connect**。
 - b. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇 Amazon Linux，然後選擇 Amazon Linux 2023 AMI。保留其他選項的預設選擇。

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0efa651876de2a5ce (64-bit (x86), uefi-preferred) / ami-0699f753302dd8b00 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0efa651876de2a5ce

Verified provider

- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需建立新 key pair 的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [建立 key pair](#)。

- e. 對於網路設定中的允許 SSH 流量，選擇 EC2 執行個體的 SSH 連線來源。

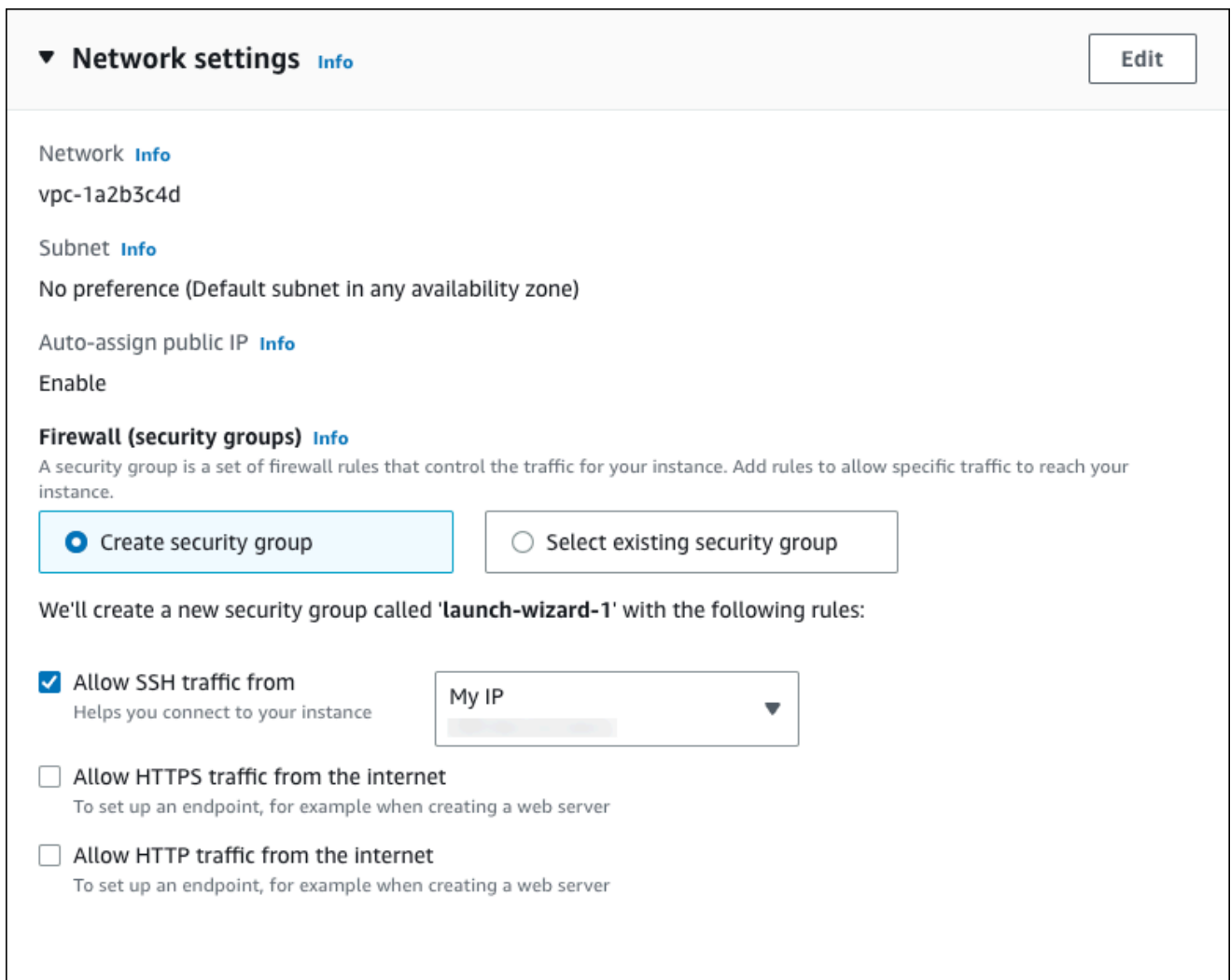
如果顯示的 IP 地址對 SSH 連線而言是正確的，您可以選擇 My IP (我的 IP)。否則，您可以決定用於使用 Secure Shell (SSH) 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 192.0.2.1/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

下圖顯示網路設定區段的範例。



▼ **Network settings** [Info](#) Edit

Network [Info](#)
vpc-1a2b3c4d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called **'launch-wizard-1'** with the following rules:

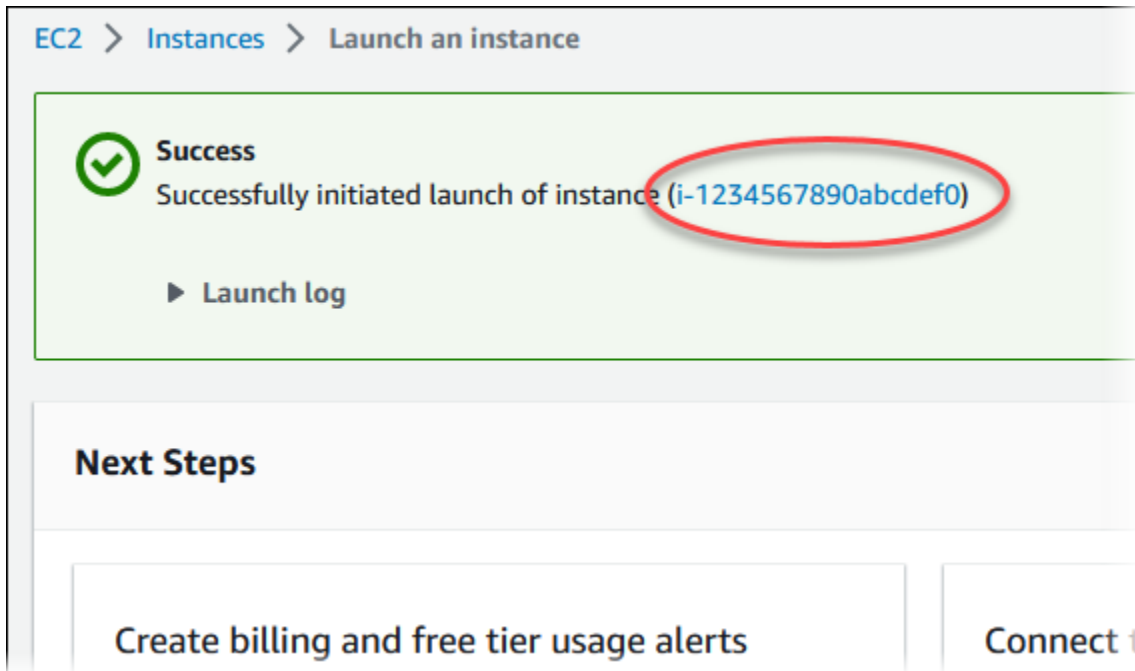
Allow SSH traffic from My IP
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

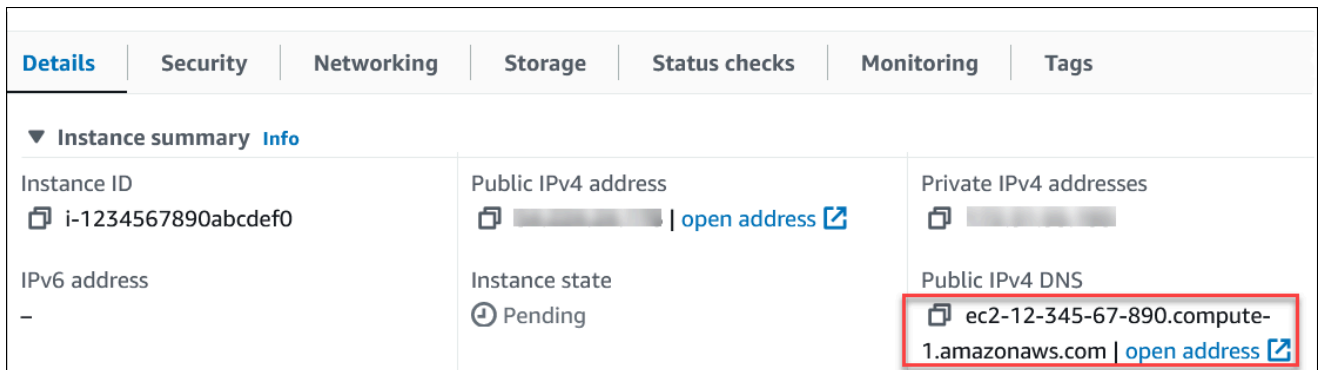
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

f. 讓剩餘區段保留預設值。


- g. 檢閱摘要面板中 EC2 執行個體組態的摘要，並在準備就緒時選擇啟動執行個體。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：i-1234567890abcdef0。



6. 選擇 EC2 執行個體識別符，以開啟 EC2 執行個體清單，然後選取您的 EC2 執行個體。
7. 在詳細資訊索引標籤中，請記下以下值，當您使用 SSH 進行連線時需要這些值：
 - a. 在執行個體摘要中，記下公用 IPv4 DNS 的值。



- b. 在執行個體詳細資訊中，記下金鑰對名稱的值。

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name  ec2-database-connect-key-pair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -

8. 請等待 EC2 執行個體的執行個體狀態為執行中，然後再繼續動作。

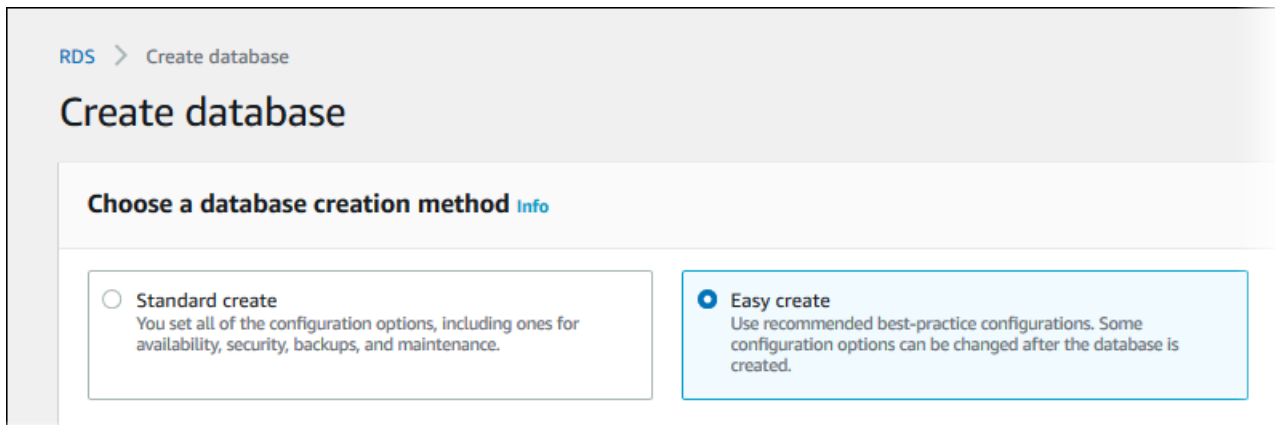
步驟 2：建立 Oracle 資料庫執行個體

Amazon RDS 的基本建置組塊為資料庫執行個體。這是您執行 Oracle 資料庫的環境。

在此範例中，您使用輕鬆建立來建立執行 Oracle 資料庫引擎、資料庫執行個體類別為 db.m5.large 的資料庫執行個體。

使用「輕鬆建立」來建立 Oracle 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要 AWS 區域 在其中建立資料庫執行個體的執行個體。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)，並確定選擇 Easy Create (輕鬆建立)。



5. 在 Configuration (組態)，選擇 Oracle (Oracle)。
6. 在 DB instance size (資料庫執行個體大小) 中，選擇 Dev/Test (開發/測試)。
7. 對於 DB instance identifier (資料庫執行個體識別碼)，請輸入 **database-test1**。

8. 在主要使用者名稱，輸入主要使用者的名稱，或保留預設名稱。

Create database (建立資料庫) 頁面看起來應該會如下圖所示。

Configuration

Engine type [Info](#)

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



Edition

Oracle Enterprise Edition

Affordable and full-featured database management system supporting up to 16 vCPUs.

Oracle Standard Edition Two

Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

DB instance size

Production

db.r5.large
2 vCPUs
16 GiB RAM
500 GiB

Dev/Test

db.m5.large
2 vCPUs
8 GiB RAM
100 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)

步驟 2: 建立 Oracle 資料庫執行個體

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter.

9. 如要將自動產生的主要密碼用在資料庫執行個體，請選取自動產生密碼。

如要輸入您的主要密碼，請確認清除自動產生密碼方塊，然後在主要密碼和確認密碼中輸入相同的密碼。

10. 若要設定與先前建立之 EC2 執行個體的連線，請開啟設定 EC2 連線 - 選用。

選取連線至 EC2 運算資源。選擇先前建立的 EC2 執行個體。

▼ Set up EC2 connection - optional

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i- ▼

i-1234567890abcdef0

11. 開啟檢視輕鬆建立的預設設定。

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

Configuration ▼	Value	Editable after database is created ▲
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:oracle-se2-19	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0a1b2c3d	Yes
Publicly accessible	No	Yes
Database port	1521	Yes
DB instance identifier	database-test1	Yes
DB engine version	19.0.0.0.ru-2023-01.rur-2023-01.r1	Yes
DB parameter group	default.oracle-se2-19	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

您可以檢查與 Easy Create (輕鬆建立) 一起使用的預設設定。資料庫建立後可編輯欄顯示您可以在資料庫建立後變更的選項。

- 若該欄的設定為否，而您想要其他設定，可以使用標準建立來建立資料庫執行個體。

- 若該欄的設定為是，而您想要其他設定，可以使用標準建立來建立資料庫執行個體，或在建立後修改該資料庫執行個體的設定。

12. 選擇建立資料庫。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

您可以使用出現的使用者名稱和密碼，來以主要使用者的身分連線至資料庫執行個體。

⚠ Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，您可以將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

13. 在資料庫清單中，選擇新 Oracle 資料庫執行個體的名稱，以顯示其詳細資訊。

在其可供使用之前，資料庫執行個體會處於建立中狀態。

Summary			
DB identifier database-test1	CPU -	Status Creating	Class db.r6g.large
Role Instance	Current activity	Engine Oracle Standard Edition Two	Region & AZ -

狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。建立資料庫執行個體時，您可以繼續下一步並建立 EC2 執行個體。

(選擇性) 使用建立 VPC、EC2 執行個體和 Oracle 資料庫執行個體 AWS CloudFormation

您可以將基礎設施視為程式碼來佈建資源，而不是使用 AWS CloudFormation 主控台建立 VPC、EC2 執行個體和 Oracle AWS 資料庫執行個體。為了幫助您將 AWS 資源組織成更小且更易於管理的單元，

您可以使用 AWS CloudFormation 嵌套堆棧功能。如需詳細資訊，請參閱在 [AWS CloudFormation 主控台上建立堆疊](#)和[使用巢狀堆疊](#)。

Important

AWS CloudFormation 是免費的，但 CloudFormation 創建的資源是活的。您必須支付這些資源的標準使用費，直到您終止這些資源為止。總計費用會很少。如需如何將任何費用降至最低的相關資訊，請前往[AWS 免費方案](#)。

若要使用 AWS CloudFormation 主控台建立資源，請完成以下步驟：

- 步驟 1：下載 CloudFormation 範本
- 步驟 2：使用以下方式設定資源 CloudFormation

下載 CloudFormation 範本

CloudFormation 範本是 JSON 或 YAML 文字檔案，其中包含您要在堆疊中建立之資源的相關設定資訊。此範本也會為您建立 VPC 和防禦主機以及 RDS 執行個體。

若要下載樣板檔案，請開啟下列連結，[Oracle CloudFormation 樣板](#)。

在 Github 頁面中，單擊下載原始文件按鈕以保存模板 YAML 文件。

使用 CloudFormation


Note

在開始此程序之前，請確定您的 AWS 帳戶。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

使用 AWS CloudFormation 範本時，您必須選取正確的參數，以確保資源已正確建立。請遵循下列步驟：

1. 請登入 AWS Management Console 並開啟 AWS CloudFormation 主控台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 選擇 Create Stack (建立堆疊)。

3. 在 [指定範本] 區段中，選取 [從電腦上傳範本檔案]，然後選擇 [下一步]。
4. 在「指定堆疊詳細資訊」頁面中，設定下列參數：
 - a. 將堆棧名稱設置為 OracleTest 堆棧。
 - b. 在參數下，選取三個可用區域來設定可用區域。
 - c. 在 Linux 防禦主機組態下，對於金鑰名稱，選取要登入 EC2 執行個體的 key pair。
 - d. 在 Linux 防禦主機組態設定中，將允許的 IP 範圍設定為您的 IP 位址。若要使用安全殼層 (SSH) 連線至 VPC 中的 EC2 執行個體，請使用 <https://checkip.amazonaws.com> 的服務判斷您的公有 IP 位址。IP 地址的範例為 192.0.2.1/32。

 Warning

如果您使用 0.0.0.0/0 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

- e. 在「資料庫一般組態」下，將「資料庫」執行個體類別設定為 db.t3.micro。
 - f. 將資料庫名稱設定為 **database-test1**。
 - g. 在資料庫主要使用者名稱中，輸入主要使用者的名稱。
 - h. 將使用密碼管理 Secrets Manager 管理資料庫主要使用者密碼設定 false 為此教學課程。
 - i. 對於資料庫密碼，請設定您選擇的密碼。請記住此密碼，以取得教學課程的進一步步驟。
 - j. 在 [資料庫儲存設定] 下，將 [資料庫儲存類型] 設定為 gp 2。
 - k. 在 [資料庫監視組態] 下，將 [啟用 RDS Performance Insights] 設定為 false。
 - l. 保留所有其他設定為預設值。按一下「下一步」繼續。
5. 在 [設定堆疊選項] 頁面中，保留所有預設選項。按一下「下一步」繼續。
 6. 在「複查堆疊」頁面中，選取檢查資料庫和 Linux 防禦主機選項後送出。

堆疊建立程序完成後，檢視包含名稱 BastionStack 和 RDSNS 的堆疊，以記下連線至資料庫所需的資訊。如需詳細資訊，請參閱 [AWS CloudFormation 在 AWS Management Console](#)。

步驟 3：將您的 SQL 連線至 Oracle 資料庫執行個體

您可以使用任何標準 SQL 用戶端應用程式來連線至您的資料庫執行個體。在此範例中，您會使用 **Oracle 命令列用戶端來連線至 Oracle 資料庫執行個體**。

連線至 Oracle 資料庫執行個體

1. 尋找資料庫執行個體的端點 (DNS 名稱) 和連接埠號碼。
 - a. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 - b. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體的 AWS 區域。
 - c. 在導覽窗格中，選擇 Databases (資料庫)。
 - d. 選擇 Oracle 資料庫執行個體名稱以顯示其詳細資訊。
 - e. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

database-test1 Modify

Summary

DB identifier database-test1	CPU 1.88%	Status Available	Class db.m5.large
Role Instance	Current activity 0.00 sessions	Engine Oracle Standard Edition Two	Region & AZ us-east-1d

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups rds-ec2-1 (sg-0a1234567b8cd9e01) Active default (sg-0a1bcd2e) Active
Port 1521	VPC vpc-1a2c3c4d	

2. 按照 Amazon EC2 使用者指南中的連線到 [Linux 執行個體中的步驟](#)，Connect 到先前建立的 EC2 執行個體。

建議您使用 SSH 連線至 EC2 執行個體。如果 SSH 用戶端公用程式已安裝在 Windows、Linux 或 Mac 上，您可以使用下列命令格式連線至執行個體：

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

例如，假設 `ec2-database-connect-key-pair.pem` 存放在 Linux 上的 `/dir1` 中，而 EC2 執行個體的公用 IPv4 DNS 為 `ec2-12-345-678-90.compute-1.amazonaws.com`。您的 SSH 命令看起來如下所示：

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-user@ec2-12-345-678-90.compute-1.amazonaws.com
```

- 更新 EC2 執行個體上的軟體，以取得最新的錯誤修正和安全性更新。為此，請使用下列命令。

Note

`-y` 選項不要求確認就會安裝更新。若要先檢查更新再安裝，請省略此選項。

```
sudo dnf update -y
```

- 在 Web 瀏覽器中，前往 <https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html>。
- 如需出現在網頁上的最新資料庫版本，請複製即時用戶端基本套件和 SQL*Plus 套件的 .rpm 連結 (不是 .zip 連結)。例如，下列是針對 Oracle Database 21.9 版的連結：

- https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
- https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm

- 在您的 SSH 工作階段中，執行 `wget` 命令，從您在上一個步驟中取得的連結下載 .rpm 檔案。下列範例會下載 Oracle Database 21.9 版的 .rpm 檔案：

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

- 執行 `dnf` 命令安裝套件，如下所示：

```
sudo dnf install oracle-instantclient-*.rpm
```

8. 啟動 SQL*Plus，然後連線至 Oracle 資料庫執行個體。例如，輸入下列命令。

以資料庫執行個體端點 (DNS 名稱) 取代 *oracle-db-instance-endpoint*，以主要使用者名稱取代 *admin*。當您針對 Oracle 使用輕鬆建立時，資料庫名為 DATABASE。提示您輸入密碼時，請提供您使用的主要密碼。

```
sqlplus admin@oracle-db-instance-endpoint:1521/DATABASE
```

輸入使用者的密碼之後，您應該會看到類似如下的輸出。

```
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Mar 1 16:41:28 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Mar 01 2023 16:30:52 +00:00

Connected to:
Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL>
```

如需連線至 RDS for Oracle 資料庫執行個體的詳細資訊，請參閱 [連線至 RDS for Oracle 資料庫執行個體](#)。如果您無法連線至資料庫執行個體，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

基於安全考量，最佳做法是使用加密連線。僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 Oracle 連線。如需使用加密連線的詳細資訊，請參閱 [保護 Oracle 資料庫執行個體連線的安全](#)。

9. 執行 SQL 命令。

例如，下列 SQL 命令會顯示目前的日期：

```
SELECT SYSDATE FROM DUAL;
```

步驟 4：刪除 EC2 執行個體和資料庫執行個體

在連線至您已建立的範例 EC2 執行個體與資料庫執行個體，並探索這些執行個體之後，請將其刪除，才不會再對您收費。

如果您曾經 AWS CloudFormation 建立資源，請略過此步驟並前往下一個步驟。

刪除 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，選擇執行個體。
3. 選取 EC2 執行個體，並選取執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇 Terminate (終止)。

如需刪除 EC2 執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的[終止執行個體](#)。

刪除不含最終資料庫快照的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要刪除的資料庫執行個體。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 清除是否建立最後的快照？和保留自動備份。
6. 完成確認，然後選擇刪除。

(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation

如果您曾 AWS CloudFormation 經建立資源，請在連接並探索範例 EC2 執行個體和資料庫執行個體後刪除 CloudFormation 堆疊，這樣您就不再需要支付費用。

刪除 CloudFormation 資源的步驟

1. 開啟主 AWS CloudFormation 控台。
2. 在的 [堆疊] 頁面上 CloudFormationconsole，選取根堆疊 (沒有名稱為 vPCStack 的堆疊 BastionStack 或 RDSNS)。

3. 選擇刪除。
4. 出現確認提示時，選取「刪除堆疊」。

如需有關在中刪除堆疊的詳細資訊 CloudFormation，請參閱《[使用指南](#)》中的〈[刪除 AWS CloudFormation 主控台上的堆疊](#)〉。

(選用) 將資料庫執行個體連線至 Lambda 函數

您也可以將 RDS for Oracle 資料庫執行個體連線至 Lambda 無伺服器運算資源。Lambda 函數允許您在沒有佈建或管理基礎設施的情況下執行程式碼。Lambda 函數還允許您自動回應任何規模的程式碼執行請求，從每天十幾個事件到每秒數百個事件。如需更多詳細資訊，請參閱 [自動連線 Lambda 函數和資料庫執行個體](#)。

建立並連線至 PostgreSQL 資料庫執行個體

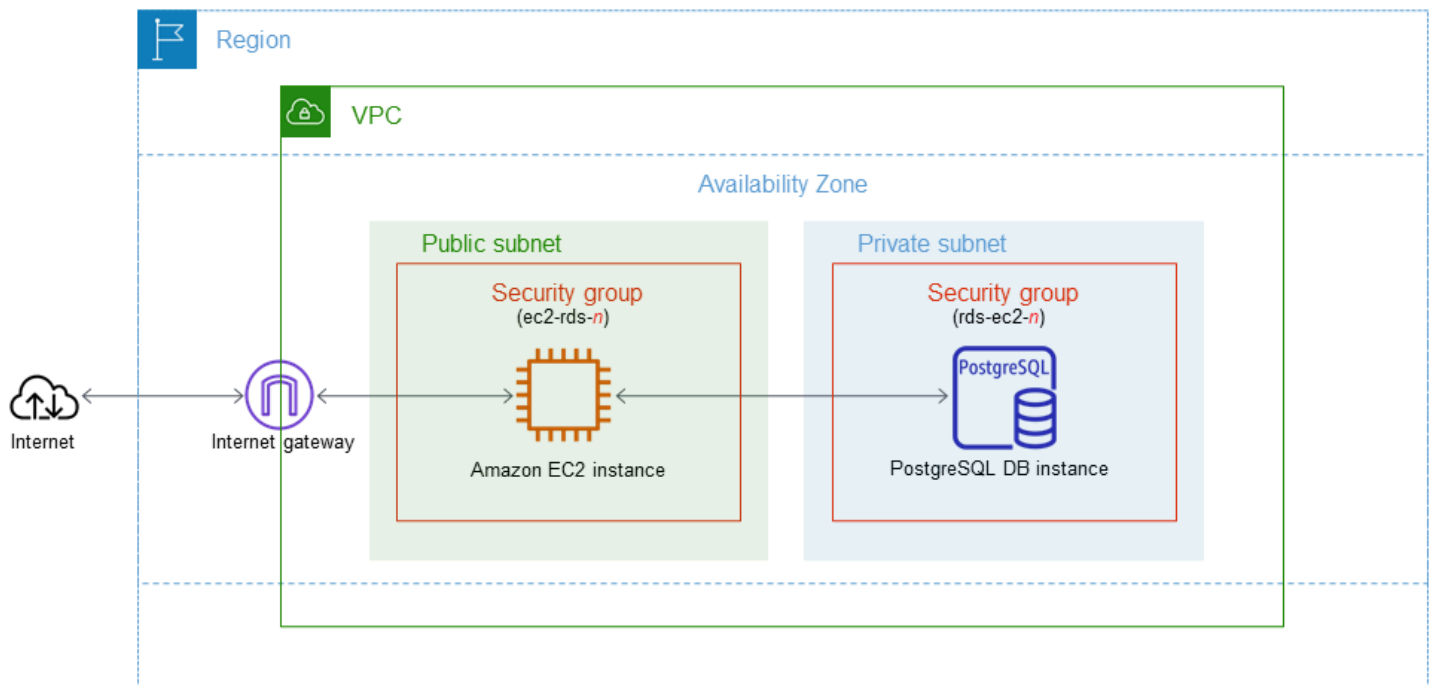
本教學課程會建立 EC2 執行個體和 RDS for PostgreSQL 資料庫執行個體。本教學課程說明如何使用標準 PostgreSQL 用戶端，從 EC2 執行個體存取資料庫執行個體。本教學課程為最佳實務，會在虛擬私有雲端 (VPC) 中建立私有資料庫執行個體。在多數情況下，相同 VPC 中的其他資源 (例如 EC2 執行個體) 可以存取該資料庫執行個體，但 VPC 以外的資源便無法存取該執行個體。

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。在可用區域中，EC2 執行個體位於公有子網路中，而資料庫執行個體則可於私有子網路中。

⚠ Important

創建 AWS 帳戶無需支付任何費用。但是，完成此教學課程後，您可能會對使用的 AWS 資源產生費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

下圖顯示此教學課程完成時的組態。



本教學課程可讓您使用下列其中一種方法來建立資源：

1. 使用 AWS Management Console - [步驟 1：建立 EC2 執行個體](#) 和 [步驟 2：建立 PostgreSQL 資料庫執行個體](#)

2. 用 AWS CloudFormation 於建立資料庫執行個體和 EC2 執行個體-[\(選擇性\) 使用建立 VPC、EC2 執行個體和 PostgreSQL 執行個體 AWS CloudFormation](#)

第一種方法使用輕鬆建立，透過 AWS Management Console 在這裡，您只指定資料庫引擎類型、資料庫執行個體大小和資料庫執行個體識別碼。Easy Create (輕鬆建立) 會使用其他組態選項的預設設定。

改為使用標準建立時，您可以在建立資料庫執行個體時指定更多組態選項。這些選項包括可用性、安全性、備份和維護的設定。若要建立公有資料庫執行個體，您必須使用標準建立。如需相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

主題

- [必要條件](#)
- [步驟 1：建立 EC2 執行個體](#)
- [步驟 2：建立 PostgreSQL 資料庫執行個體](#)
- [\(選擇性\) 使用建立 VPC、EC2 執行個體和 PostgreSQL 執行個體 AWS CloudFormation](#)
- [步驟 3：連線至 PostgreSQL 資料庫執行個體](#)
- [步驟 4：刪除 EC2 執行個體和資料庫執行個體](#)
- [\(選擇性\) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation](#)
- [\(選用\) 將資料庫執行個體連線至 Lambda 函數](#)

必要條件

在開始之前，請先完成下節所含步驟：

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

步驟 1：建立 EC2 執行個體

建立您會用來連線至資料庫的 Amazon EC2 執行個體。

建立 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。

2. 在的右上角 AWS Management Console，選擇您要 AWS 區域 在其中建立 EC2 執行個體的執行個體。
3. 選擇 EC2 儀表板，然後選擇啟動執行個體，如下圖所示。

Resources

You are using the following Amazon EC2 resources in the Region Region:

Instances (running)	3	Dedicated Hosts	0
Instances	3	Key pairs	5
Placement groups	0	Security groups	10
Volumes	3		

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼ **Migrate a server** ↗

Note: Your instances will launch in the US West (Oregon) Region

Service health

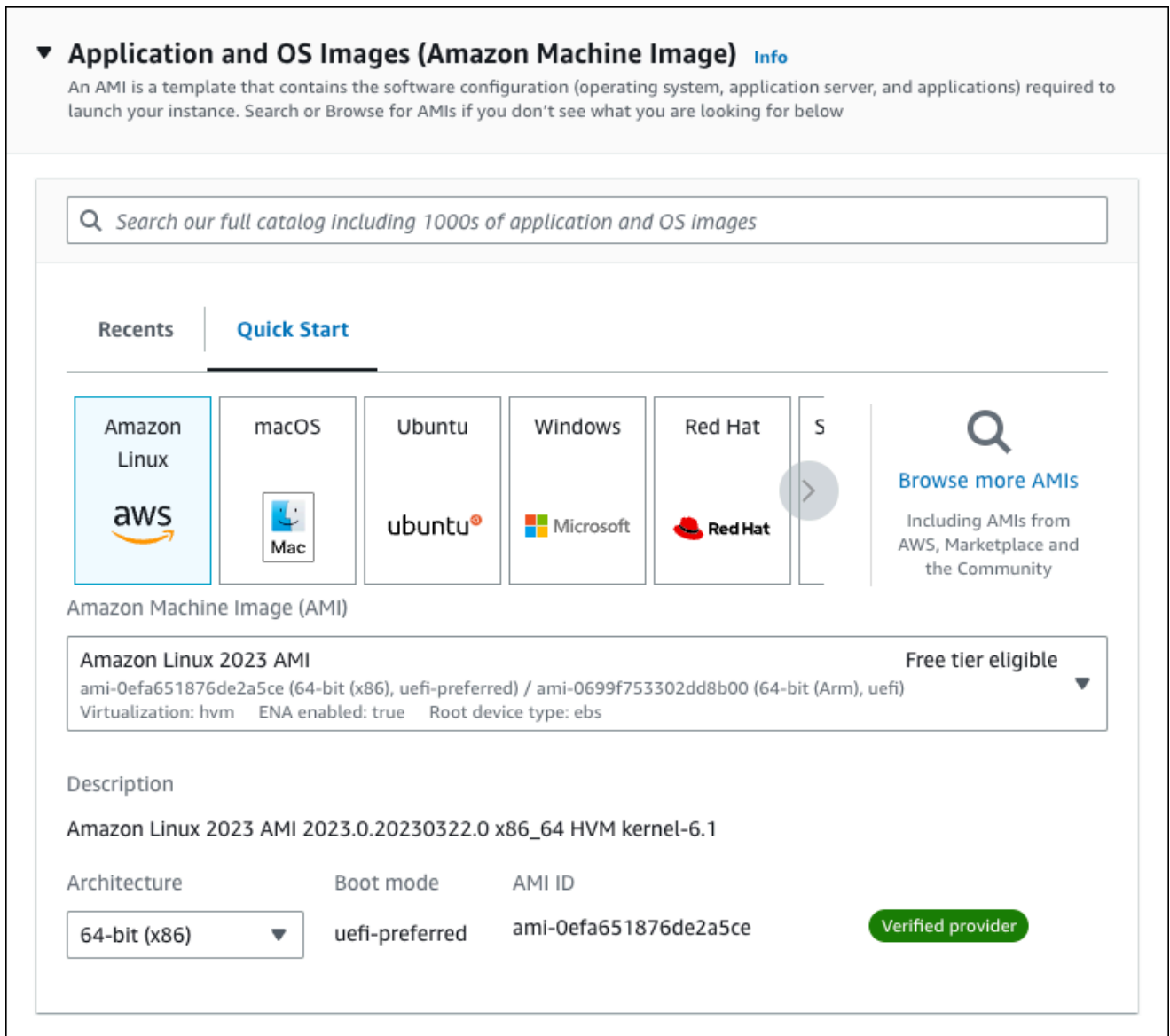
Region
Region

Zones

啟動執行個體頁面即開啟。

4. 在啟動執行個體頁面中選擇下列設定。
 - a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **ec2-database-connect**。

- b. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇 Amazon Linux，然後選擇 Amazon Linux 2023 AMI。保留其他選項的預設選擇。



▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0efa651876de2a5ce (64-bit (x86), uefi-preferred) / ami-0699f753302dd8b00 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0efa651876de2a5ce

Verified provider


- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需有關建立新 key pair 的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [建立 key pair](#)。

- e. 對於網路設定中的允許 SSH 流量，選擇 EC2 執行個體的 SSH 連線來源。

如果顯示的 IP 地址對 SSH 連線而言是正確的，您可以選擇 My IP (我的 IP)。否則，您可以決定用於使用 Secure Shell (SSH) 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 192.0.2.1/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

 Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

下圖顯示網路設定區段的範例。

▼ **Network settings** [Info](#) Edit

Network [Info](#)
vpc-1a2b3c4d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

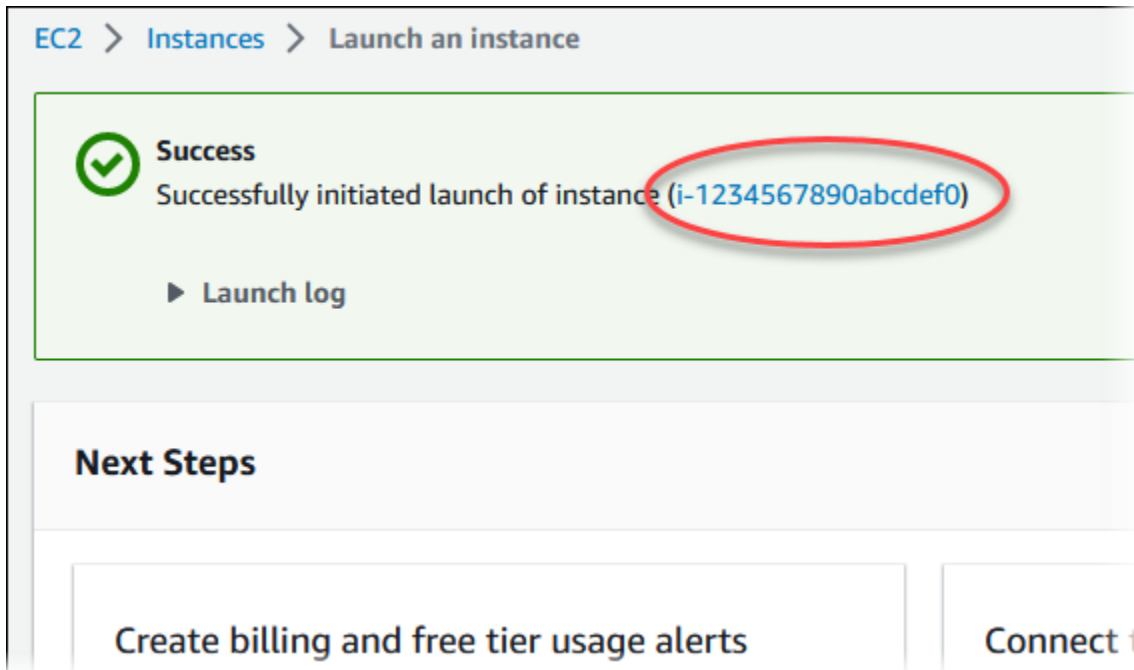
We'll create a new security group called **'launch-wizard-1'** with the following rules:

Allow SSH traffic from My IP
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

- f. 讓剩餘區段保留預設值。
 - g. 檢閱摘要面板中 EC2 執行個體組態的摘要，並在準備就緒時選擇啟動執行個體。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：i-1234567890abcdef0。



6. 選擇 EC2 執行個體識別符，以開啟 EC2 執行個體清單，然後選取您的 EC2 執行個體。
7. 在詳細資訊索引標籤中，請記下以下值，當您使用 SSH 進行連線時需要這些值：
 - a. 在執行個體摘要中，記下公用 IPv4 DNS 的值。

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
▼ Instance summary Info						
Instance ID i-1234567890abcdef0	Public IPv4 address [redacted] open address	Private IPv4 addresses [redacted]	IPv6 address -	Instance state Pending	Public IPv4 DNS ec2-12-345-67-890.compute-1.amazonaws.com open address	

- b. 在執行個體詳細資訊中，記下金鑰對名稱的值。

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name ec2-database-connect-key-pair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -

8. 請等待 EC2 執行個體的執行個體狀態為執行中，然後再繼續動作。

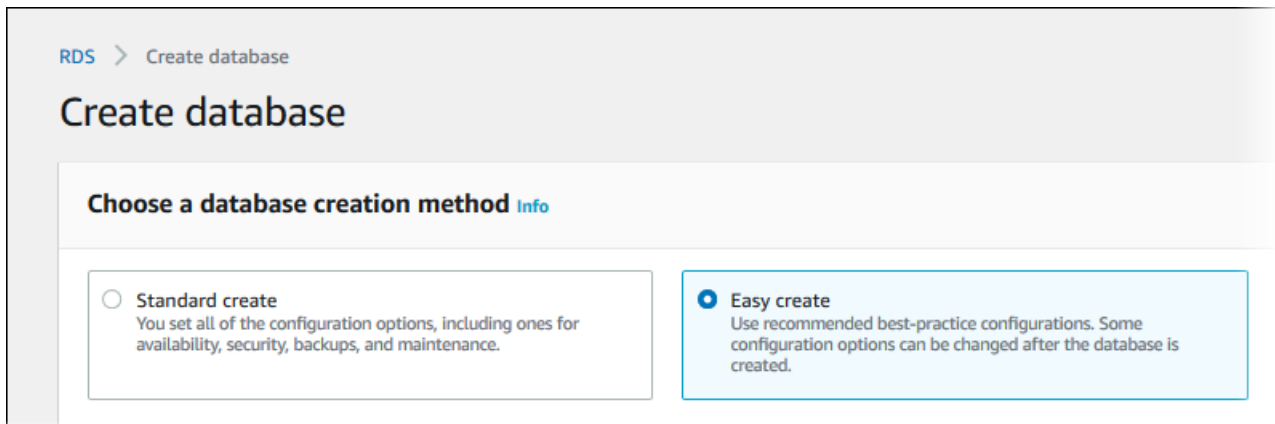
步驟 2：建立 PostgreSQL 資料庫執行個體

Amazon RDS 的基本建置組塊為資料庫執行個體。這是您執行 PostgreSQL 資料庫的環境。

在此範例中，您使用輕鬆建立來建立執行 PostgreSQL 資料庫引擎、資料庫執行個體類別為 db.t3.micro 的資料庫執行個體。

如要以 Easy create (輕鬆建立) 來建立 PostgreSQL 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇您要在其中建立資料庫執行個體的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)，並確定選擇 Easy Create (輕鬆建立)。









5. 在 Configuration (組態)，選擇 PostgreSQL (PostgreSQL)。
6. 在 DB instance size (資料庫執行個體大小) 中，選擇 Free tier (免費方案)。
7. 對於 DB instance identifier (資料庫執行個體識別碼)，請輸入 **database-test1**。
8. 針對主要使用者名稱，輸入主要使用者的名稱，或保留預設名稱 (**postgres**)。

Create database (建立資料庫) 頁面看起來應該會如下圖所示。

Configuration

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 	<input type="radio"/> MySQL 
<input type="radio"/> MariaDB 	<input checked="" type="radio"/> PostgreSQL 	<input type="radio"/> Microsoft SQL Server 

DB instance size

<input type="radio"/> Production db.r6g.xlarge 4 vCPUs 32 GiB RAM 500 GiB	<input type="radio"/> Dev/Test db.r6g.large 2 vCPUs 16 GiB RAM 100 GiB	<input checked="" type="radio"/> Free tier db.t3.micro 2 vCPUs 1 GiB RAM 20 GiB
---	--	---

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

9. 如要將自動產生的主要密碼用在資料庫執行個體，請選取自動產生密碼。

如要輸入您的主要密碼，請確認清除自動產生密碼方塊，然後在主要密碼和確認密碼中輸入相同的密碼。

10. 若要設定與先前建立之 EC2 執行個體的連線，請開啟設定 EC2 連線 - 選用。

選取連線至 EC2 運算資源。選擇先前建立的 EC2 執行個體。

▼ Set up EC2 connection - *optional*

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-
i-1234567890abcdef0



11. 開啟檢視輕鬆建立的預設設定。

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

Configuration ▼	Value	Editable after database is created ▲
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:postgres-14	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.6	Yes
DB parameter group	default.postgres14	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

您可以檢查與 Easy Create (輕鬆建立) 一起使用的預設設定。資料庫建立後可編輯欄顯示您可以在資料庫建立後變更的選項。

- 若該欄的設定為否，而您想要其他設定，可以使用標準建立來建立資料庫執行個體。

- 若該欄的設定為是，而您想要其他設定，可以使用標準建立來建立資料庫執行個體，或在建立後修改該資料庫執行個體的設定。

12. 選擇建立資料庫。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

您可以使用出現的使用者名稱和密碼，來以主要使用者的身分連線至資料庫執行個體。

⚠ Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，您可以將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

13. 在資料庫清單中，選擇新 PostgreSQL 資料庫執行個體的名稱，以顯示其詳細資訊。

在其可供使用之前，資料庫執行個體會處於建立中狀態。

Summary			
DB identifier database-test1	CPU -	Status Creating	Class db.r6g.large
Role Instance	Current activity	Engine PostgreSQL	Region & AZ -

狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。

(選擇性) 使用建立 VPC、EC2 執行個體和 PostgreSQL 執行個體 AWS CloudFormation

您可以使用將基礎設施視為程式碼來佈建 AWS 資源，而不是使用 AWS CloudFormation 主控台建立 VPC、EC2 執行個體和 PostgreSQL 執行個體。為了幫助您將 AWS 資源組織成更小且更易於管理的單元，您可以使用 AWS CloudFormation 嵌套堆棧功能。如需詳細資訊，請參閱[在 AWS CloudFormation 主控台上建立堆疊](#)和[使用巢狀堆疊](#)。

⚠ Important

AWS CloudFormation 是免費的，但 CloudFormation 創建的資源是活的。您必須支付這些資源的標準使用費，直到您終止這些資源為止。總計費用會很少。如需如何將任何費用降至最低的相關資訊，請前往[AWS 免費方案](#)。

若要使用 AWS CloudFormation 主控台建立資源，請完成以下步驟：

- 步驟 1：下載 CloudFormation 範本
- 步驟 2：使用 CloudFormation

下載 CloudFormation 範本

CloudFormation 範本是 JSON 或 YAML 文字檔案，其中包含您要在堆疊中建立之資源的相關設定資訊。此範本也會為您建立 VPC 和防禦主機，以及 RDS 執行個體。

要下載模板文件，請打開以下鏈接，[PostgreSQL CloudFormation](#) 模板。

在 Github 頁面中，單擊下載原始文件按鈕以保存模板 YAML 文件。

使用 CloudFormation


📘 Note

在開始此程序之前，請確定您的 AWS 帳戶。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

使用 AWS CloudFormation 範本時，您必須選取正確的參數，以確保資源已正確建立。請遵循下列步驟：

1. 請登入 AWS Management Console 並開啟 AWS CloudFormation 主控台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 選擇 Create Stack (建立堆疊)。
3. 在 [指定範本] 區段中，選取 [從電腦上傳範本檔案]，然後選擇 [下一步]。
4. 在「指定堆疊詳細資訊」頁面中，設定下列參數：

- a. 將堆棧名稱設置為 PostgreSQL TestStack。
- b. 在參數下，透過選取三個可用區域來設定可用區域。
- c. 在 Linux 防禦主機組態下，對於金鑰名稱，選取要登入 EC2 執行個體的 key pair。
- d. 在 Linux 防禦主機組態設定中，將允許的 IP 範圍設定為您的 IP 位址。若要使用安全殼層 (SSH) 連線至 VPC 中的 EC2 執行個體，請使用 <https://checkip.amazonaws.com> 的服務判斷您的公有 IP 位址。IP 地址的範例為 192.0.2.1/32。

 Warning

如果您使用 0.0.0.0/0 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有 EC2 執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的 EC2 執行個體。

- e. 在「資料庫一般組態」下，將「資料庫」執行個體類別設定為 db.t3.micro。
 - f. 將資料庫名稱設定為 **database-test1**。
 - g. 在資料庫主要使用者名稱中，輸入主要使用者的名稱。
 - h. 將使用密碼管理 Secrets Manager 管理資料庫主要使用者密碼設定 false 為此教學課程。
 - i. 對於資料庫密碼，請設定您選擇的密碼。請記住此密碼，以取得教學課程的進一步步驟。
 - j. 在 [資料庫儲存設定] 下，將 [資料庫儲存類型] 設定為 gp 2。
 - k. 在 [資料庫監視組態] 下，將 [啟用 RDS Performance Insights] 設定為 false。
 - l. 保留所有其他設定為預設值。按一下「下一步」繼續。
5. 在 [設定堆疊選項] 頁面中，保留所有預設選項。按一下「下一步」繼續。
 6. 在「複查堆疊」頁面中，選取檢查資料庫和 Linux 防禦主機選項後送出。

堆疊建立程序完成後，檢視包含名稱 BastionStack 和 RDSNS 的堆疊，以記下連線至資料庫所需的資訊。如需詳細資訊，請參閱 [檢視上的 AWS CloudFormation 堆疊資料和資源 AWS Management Console](#)。

步驟 3：連線至 PostgreSQL 資料庫執行個體

您可以使用 pgadmin 或 psql 連線至資料庫執行個體。此範例說明如何使用 psql 命令列用戶端來連線至 PostgreSQL 資料庫執行個體。

使用 psql 連線至 PostgreSQL 資料庫執行個體

1. 尋找資料庫執行個體的端點 (DNS 名稱) 和連接埠號碼。
 - a. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 - b. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體的 AWS 區域。
 - c. 在導覽窗格中，選擇 Databases (資料庫)。
 - d. 選擇 PostgreSQL 資料庫執行個體名稱以顯示其詳細資訊。
 - e. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

RDS > Databases > database-test1

database-test1

Summary

DB identifier database-test1	CPU 5.82%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port	Networking
Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com	Availability Zone us-east-1c
Port 5432	VPC vpc-
	Subnet group default

2. 按照 Amazon EC2 使用者指南中的連線到 [Linux 執行個體中的步驟](#)，Connect 到先前建立的 EC2 執行個體。

建議您使用 SSH 連線至 EC2 執行個體。如果 SSH 用戶端公用程式已安裝在 Windows、Linux 或 Mac 上，您可以使用下列命令格式連線至執行個體：

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

例如，假設 `ec2-database-connect-key-pair.pem` 存放在 Linux 上的 `/dir1` 中，而 EC2 執行個體的公用 IPv4 DNS 為 `ec2-12-345-678-90.compute-1.amazonaws.com`。您的 SSH 命令看起來如下所示：

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-user@ec2-12-345-678-90.compute-1.amazonaws.com
```

- 更新 EC2 執行個體上的軟體，以取得最新的錯誤修正和安全性更新。若要執行此操作，請使用以下命令。

Note

`-y` 選項不要求確認就會安裝更新。若要先檢查更新再安裝，請省略此選項。

```
sudo dnf update -y
```

- 若要在 Amazon Linux 2023 上安裝 PostgreSQL 的 `mysql` 命令列用戶端，請執行下列命令：

```
sudo dnf install postgresql15
```

- 連線至 PostgreSQL 資料庫執行個體。例如，在用戶端電腦上的命令提示字元中輸入下列命令。此動作可讓您使用 `psql` 用戶端，連線至 PostgreSQL 資料庫執行個體。

取代 *endpoint* 的資料庫執行個體端點 (DNS 名稱)，取代 *postgres* 要連線的資料庫名稱 `--dbname`，以及取代您用於 *postgres* 的主要使用者名稱。提示您輸入密碼時，請提供您使用的主要密碼。

```
psql --host=endpoint --port=5432 --dbname=postgres --username=postgres
```

輸入使用者的密碼之後，您應該會看到類似如下的輸出：

```
psql (14.3, server 14.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
compression: off)
Type "help" for help.
```

```
postgres=>
```

如需連線至 PostgreSQL 資料庫執行個體的詳細資訊，請參閱 [連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)。如果您無法連線至資料庫執行個體，請參閱 [針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解](#)。

基於安全考量，最佳做法是使用加密連線。僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 PostgreSQL 連線。如需使用加密連線的詳細資訊，請參閱 [透過 SSL 連接至 PostgreSQL 資料庫執行個體](#)。

6. 執行 SQL 命令。

例如，下列 SQL 命令會顯示目前的日期和時間：

```
SELECT CURRENT_TIMESTAMP;
```

步驟 4：刪除 EC2 執行個體和資料庫執行個體

在連線至您已建立的範例 EC2 執行個體與資料庫執行個體，並探索這些執行個體之後，請將其刪除，才不會再對您收費。

如果您曾經 AWS CloudFormation 建立資源，請略過此步驟並前往下一個步驟。

刪除 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，選擇執行個體。
3. 選取 EC2 執行個體，並選取執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇 Terminate (終止)。

如需刪除 EC2 執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [終止執行個體](#)。

刪除不含最終資料庫快照的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要刪除的資料庫執行個體。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 清除是否建立最後的快照？和保留自動備份。
6. 完成確認，然後選擇刪除。

(選擇性) 刪除使用建立的 EC2 執行個體和資料庫執行個體 CloudFormation

如果您曾 AWS CloudFormation 經建立資源，請在連接並探索範例 EC2 執行個體和資料庫執行個體後刪除 CloudFormation 堆疊，這樣您就不再需要支付費用。

若要刪除資 CloudFormation 源

1. 開啟主 AWS CloudFormation 控制台。
2. 在的 [堆疊] 頁面上 CloudFormationconsole，選取根堆疊 (沒有名稱為 vPCStack 的堆疊 BastionStack 或 RDSNS)。
3. 選擇刪除。
4. 出現確認提示時，選取「刪除堆疊」。

如需有關在中刪除堆疊的詳細資訊 CloudFormation，請參閱 [《使用指南》中的〈刪除 AWS CloudFormation 主控台上的堆疊AWS CloudFormation〉](#)。

(選用) 將資料庫執行個體連線至 Lambda 函數

您也可以將 RDS for PostgreSQL 資料庫執行個體連線至無伺服器運算資源。Lambda 函數允許您在沒有佈建或管理基礎設施的情況下執行程式碼。Lambda 函數還允許您自動回應任何規模的程式碼執行請求，從每天十幾個事件到每秒數百個事件。如需更多詳細資訊，請參閱 [自動連線 Lambda 函數和資料庫執行個體](#)。

教學：建立 Web 伺服器 and Amazon RDS 資料庫執行個體

本教學課程說明如何安裝支援 PHP 的 Apache Web 伺服器，並建立 MariaDB、MySQL 或 PostgreSQL 資料庫。Web 伺服器會在使用 Amazon Linux 2023 的 Amazon EC2 執行個體上執行，而且您可以選擇 MySQL 或 PostgreSQL 資料庫執行個體。Amazon EC2 執行個體和 資料庫執行個體都會在以 Amazon VPC 服務為基礎的 Virtual Private Cloud (VPC) 中執行。

Important

建立 AWS 帳戶是免費的。不過，若要完成本教學課程，您可能需要支付所使用的 AWS 資源的費用。如果不再需要這些資源，您可以在完成教學課程後刪除這些資源。

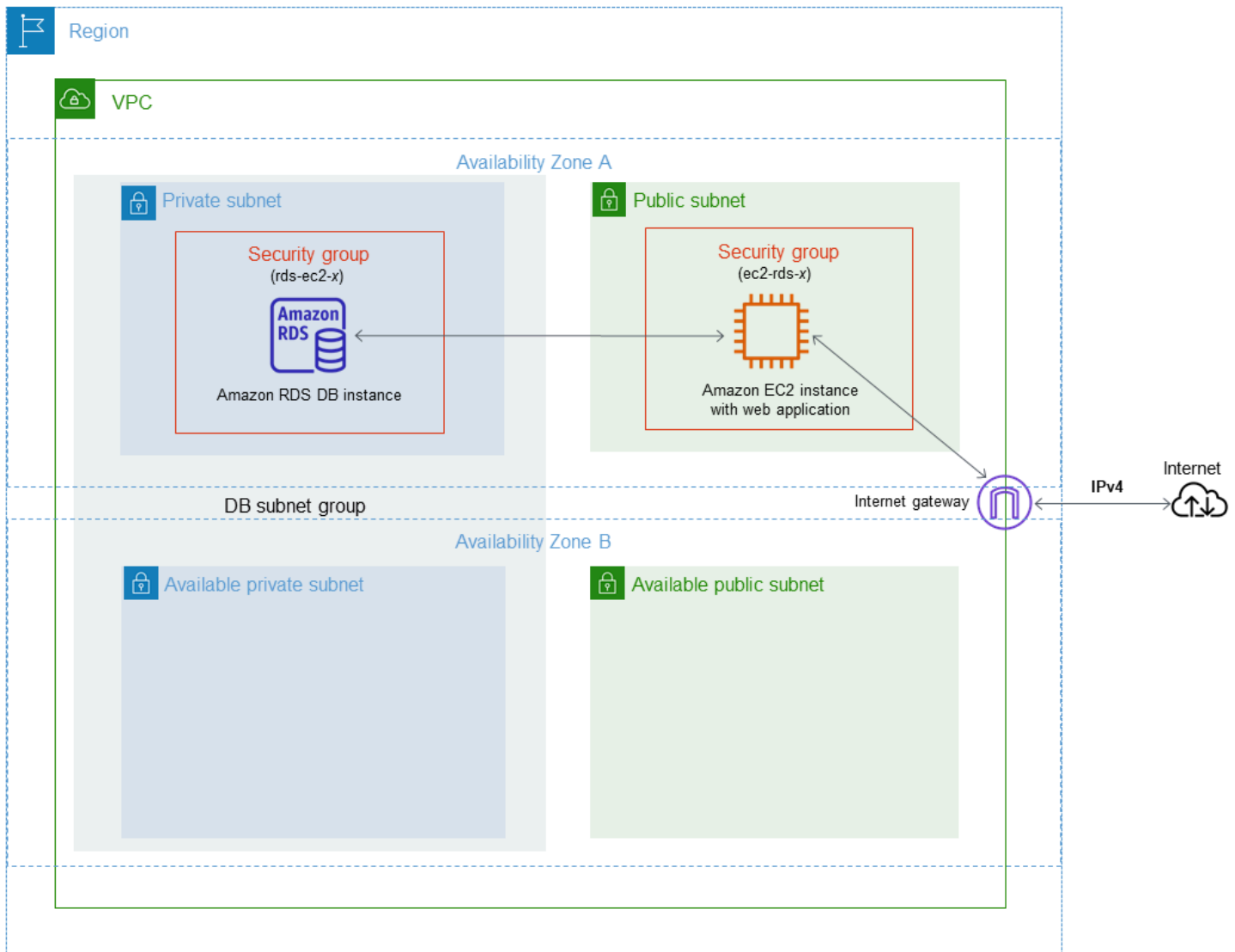
Note

本教學課程使用 Amazon Linux 2023，可能不適用於其他 Linux 版本。

在後續的教學課程中，您會為 AWS 帳戶 建立使用預設 VPC、子網路和安全群組的 EC2 執行個體。本教學課程向您介紹如何建立資料庫執行個體，並自動設定與您建立的 EC2 執行個體的連線。然後，本教學課程將向您展示如何在 EC2 執行個體上安裝 Web 伺服器。您可以使用資料庫執行個體端點，將 Web 伺服器連線到 VPC 中的資料庫執行個體。

1. [啟動 EC2 執行個體](#)
2. [建立 Amazon RDS 資料庫執行個體](#)
3. [在您的 EC2 執行個體上安裝 Web 伺服器](#)

下圖顯示此教學課程完成時的組態。



Note

完成教學課程後，在您的 VPC 中，每個可用區域都有一個公有子網路和私有子網路。本教學課程使用您 AWS 帳戶的預設 VPC，並自動設定 EC2 執行個體和資料庫執行個體之間的連線。如果您想改為針對此案例設定新的 VPC，請完成 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#) 中的任務。

啟動 EC2 執行個體

在 VPC 的公有子網路中建立 Amazon EC2 執行個體。

啟動 EC2 執行個體

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在的右上角 AWS Management Console，選擇您 AWS 區域 要建立 EC2 執行個體的位置。
3. 選擇 EC2 Dashboard (EC2 儀表板)，然後選擇 Launch Instance (啟動執行個體)，如下所示。

The screenshot displays the AWS Management Console interface for the Amazon EC2 service. At the top, under the 'Resources' section, it shows the user is using various EC2 resources in a specific region. A table lists the following resources:

Resource	Count
Instances (running)	3
Instances	3
Placement groups	0
Volumes	3
Dedicated Hosts	0
Key pairs	5
Security groups	10

Below the resource list, there is a blue banner with an information icon and the text: 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using Learn more'.

The main section is titled 'Launch instance' and includes the text: 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' Below this text, there are two buttons: 'Launch instance' (highlighted with a red circle) and 'Migrate a server'. A note below the buttons states: 'Note: Your instances will launch in the US West (Oregon) Region'.

On the right side of the dashboard, there is a 'Service health' section showing the 'Region' and a 'Zones' section.

4. 在啟動執行個體頁面中選擇下列設定。

- a. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **tutorial-ec2-instance-web-server**。
- b. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇 Amazon Linux，然後選擇 Amazon Linux 2023 AMI。保留其他選項的預設值。

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0efa651876de2a5ce (64-bit (x86), uefi-preferred) / ami-0699f753302dd8b00 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0efa651876de2a5ce

Verified provider

- c. 在 Instance type (執行個體類型) 下，選擇 t2.micro。
- d. 在 Key pair (login) (金鑰對 (登入)) 下，選擇 Key pair name (金鑰對名稱)，以使用現有金鑰對。若要為 Amazon EC2 執行個體建立新的金鑰對，請選擇 Create new key pair (建立新的金鑰對)，然後使用 Create key pair (建立金鑰對) 視窗來建立金鑰對。

如需有關建立新 key pair 的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [建立 key pair](#)。


e. 在 Network settings (網路設定) 下設定這些值，其他值則維持預設值：

- 對於 Allow SSH traffic from (允許 SSH 流量來自)，選擇 EC2 執行個體做為 SSH 的連線來源。

如果顯示的 IP 地址對 SSH 連線而言是正確的，您可以選擇 My IP (我的 IP)。

否則，您可以決定用於使用 Secure Shell (SSH) 連線至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 203.0.113.25/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請務必確定用戶端電腦所使用的 IP 地址範圍。

 Warning

如果您使用 0.0.0.0/0 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的執行個體。

- 開啟 Allow HTTPs traffic from the internet (允許來自網際網路的 HTTPS 流量)。
- 開啟 Allow HTTP traffic from the internet (允許來自網際網路的 HTTP 流量)。

▼ **Network settings** [Get guidance](#) Edit

Network [Info](#)
vpc-2aed394c

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

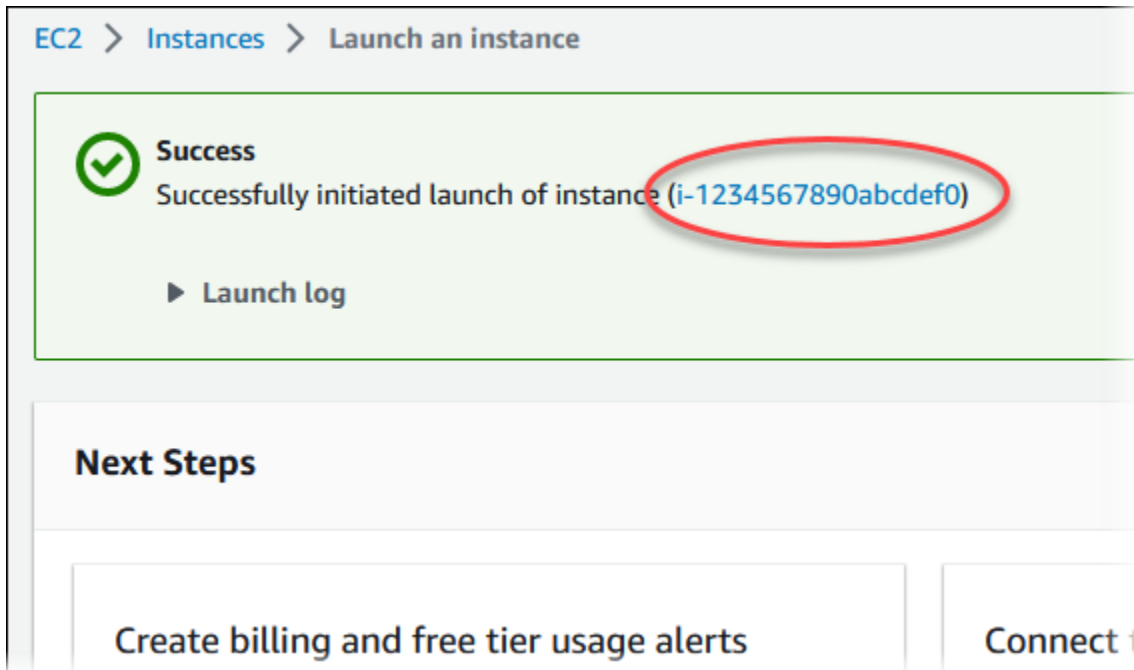
Allow SSH traffic from My IP
Helps you connect to your instance

Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ×

- f. 讓剩餘區段保留預設值。
 - g. 檢閱 Summary (摘要) 面板中執行個體組態的摘要，並在準備就緒時選擇 Launch instance (啟動執行個體)。
5. 在啟動狀態頁面上，記下新的 EC2 執行個體的識別碼，例如：i-1234567890abcdef0。



6. 選擇 EC2 執行個體識別符，以開啟 EC2 執行個體清單，然後選取您的 EC2 執行個體。
7. 在詳細資訊索引標籤中，請記下以下值，當您使用 SSH 進行連線時需要這些值：
 - a. 在執行個體摘要中，記下公用 IPv4 DNS 的值。

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
▼ Instance summary Info						
Instance ID i-1234567890abcdef0	Public IPv4 address [redacted] open address	Private IPv4 addresses [redacted]	IPv6 address -	Instance state Pending	Public IPv4 DNS ec2-12-345-67-890.compute-1.amazonaws.com open address	

- b. 在執行個體詳細資訊中，記下金鑰對名稱的值。

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name ec2-database-connect-key-pair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -

8. 繼續之前，等待執行個體的 Instance Status (執行個體狀態) 顯示為 Running (執行中)。

9. 完成 [建立 Amazon RDS 資料庫執行個體](#)。

建立 Amazon RDS 資料庫執行個體

建立 RDS for MariaDB、RDS for MySQL 或 RDS for PostgreSQL 資料庫執行個體，以維護 Web 應用程式所使用的資料。









RDS for MariaDB

若要建立 MariaDB 執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 AWS Management Console 右上角中，核取 AWS 區域。它應與您已在其中建立 EC2 執行個體的區域相同。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。
5. 在建立資料庫頁面上，選擇標準建立。
6. 在引擎選項中，選擇 MariaDB。

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input type="radio"/> MySQL 	<input checked="" type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

7. 在範本區段中，選擇免費方案。

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.	<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info
---	---	--

8. 在 Availability and durability (可用性與耐久性) 區段中，保留預設值。
9. 在 Settings (設定) 區段中，設定這些值：
 - DB instance identifier (資料庫執行個體識別符) – 輸入 **tutorial-db-instance**。
 - Master username (主要使用者名稱) – 輸入 **tutorial_user**。
 - Auto generate a password (自動產生密碼) – 保持選項關閉。
 - Master password (主要密碼) – 輸入密碼。
 - Confirm password (確認密碼) – 重新輸入密碼。

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)

10. 在 Instance configuration (執行個體組態) 區段中，設定這些值：
 - Burstable classes (includes t classes) (高載類別 (包括 t 類別))
 - db.t3.micro

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

11. 在 Storage (儲存) 區段中，保留預設值。
12. 在 Connectivity (連線能力) 區段中，設定這些值，並且其他值都維持預設值：
 - 對於 Compute resource (運算資源)，選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源)。
 - 對於 EC2 執行個體，請選擇先前建立的 EC2 執行個體，例如教程-ec2-。instance-web-server

Connectivity Info ↻

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-1234567890abcdef0
▼

i Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group `rds-ec2-X` is added to the database and another called `ec2-rds-X` to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

13. 在 Database authentication (資料庫身分驗證) 區段中，請確定已選取 Password authentication (密碼身分驗證)。
14. 在 Additional configuration (其他設定) 區段中，輸入 **sample** 做為 Initial database name (初始資料庫名稱)。其他選項都保留預設設定。
15. 若要建立 MariaDB 執行個體，請選擇建立資料庫。

您的新資料庫執行個體會出現在 Databases (資料庫) 清單中，狀態為 Creating (建立中)。

16. 等待新資料庫執行個體的 Status (狀態) 顯示為 available (可用)。接著，選擇資料庫執行個體的名稱，以顯示其詳細資訊。
17. 在 Connectivity & security (連線能力與安全性) 區段中，檢視資料庫執行個體的 Endpoint (端點) 和 Port (連接埠)。

The screenshot shows the Amazon RDS console interface for a database instance named 'tutorial-db-instance'. The breadcrumb navigation at the top reads 'RDS > Databases > tutorial-db-instance'. The main title is 'tutorial-db-instance'. Below this is a 'Summary' section with two columns of information:

DB identifier tutorial-db-instance	CPU 3.10%
Role Instance	Current activity 0 Connections

Below the summary is a navigation bar with five tabs: 'Connectivity & security' (selected), 'Monitoring', 'Logs & events', 'Configuration', and 'Maintenance'. The 'Connectivity & security' section is active, showing two sub-sections:

- Endpoint & port:** The 'Endpoint' is 'tutorial-db-instance. [redacted] west-2.rds.amazonaws.com' and the 'Port' is '3306'. Both the endpoint text and the port number are circled in red in the image.
- Networking:** The 'Availability Zone' is 'us-west-2a', the 'VPC' is 'tutorial-vpc (vpc-04badc20a546242e6)', and the 'Subnet group' is listed below.

記下資料庫執行個體的端點和連接埠。您會使用此資訊，將 Web 伺服器連接至資料庫執行個體。

18. 完成 [在您的 EC2 執行個體上安裝 Web 伺服器](#)。









RDS for MySQL

建立 MySQL 資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在 AWS Management Console 右上角中，核取 AWS 區域。它應與您已在其中建立 EC2 執行個體的區域相同。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。
5. 在建立資料庫頁面上，選擇標準建立。
6. 在引擎選項 中，選擇 MySQL。

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input checked="" type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

7. 在範本區段中，選擇免費方案。

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.	<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info
---	---	--

8. 在 Availability and durability (可用性與耐久性) 區段中，保留預設值。
9. 在 Settings (設定) 區段中，設定這些值：
 - DB instance identifier (資料庫執行個體識別符) – 輸入 **tutorial-db-instance**。
 - Master username (主要使用者名稱) – 輸入 **tutorial_user**。
 - Auto generate a password (自動產生密碼) – 保持選項關閉。
 - Master password (主要密碼) – 輸入密碼。
 - Confirm password (確認密碼) – 重新輸入密碼。

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)

10. 在 Instance configuration (執行個體組態) 區段中，設定這些值：
 - Burstable classes (includes t classes) (高載類別 (包括 t 類別))
 - db.t3.micro

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

11. 在 Storage (儲存) 區段中，保留預設值。
12. 在 Connectivity (連線能力) 區段中，設定這些值，並且其他值都維持預設值：
 - 對於 Compute resource (運算資源)，選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源)。
 - 對於 EC2 執行個體，請選擇先前建立的 EC2 執行個體，例如教程-ec2-。instance-web-server

Connectivity Info ↻

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-1234567890abcdef0
▼

Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group `rds-ec2-X` is added to the database and another called `ec2-rds-X` to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

13. 在 Database authentication (資料庫身分驗證) 區段中，請確定已選取 Password authentication (密碼身分驗證)。
14. 在 Additional configuration (其他設定) 區段中，輸入 **sample** 做為 Initial database name (初始資料庫名稱)。其他選項都保留預設設定。
15. 若要建立 MySQL 資料庫執行個體，請選擇 Create database (建立資料庫)。

您的新資料庫執行個體會出現在 Databases (資料庫) 清單中，狀態為 Creating (建立中)。

16. 等待新資料庫執行個體的 Status (狀態) 顯示為 available (可用)。接著，選擇資料庫執行個體的名稱，以顯示其詳細資訊。
17. 在 Connectivity & security (連線能力與安全性) 區段中，檢視資料庫執行個體的 Endpoint (端點) 和 Port (連接埠)。

RDS > Databases > tutorial-db-instance

tutorial-db-instance

Summary

DB identifier tutorial-db-instance	CPU 3.10%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance

Connectivity & security

Endpoint & port	Networking
Endpoint tutorial-db-instance. [redacted] west-2.rds.amazonaws.com	Availability Zone us-west-2a
Port 3306	VPC tutorial-vpc (vpc-04badc20a546242e6)
	Subnet group

記下資料庫執行個體的端點和連接埠。您會使用此資訊，將 Web 伺服器連接至資料庫執行個體。

18. 完成 [在您的 EC2 執行個體上安裝 Web 伺服器](#)。









RDS for PostgreSQL

若要建立 PostgreSQL 資料庫執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 AWS Management Console 右上角中，核取 AWS 區域。它應與您已在其中建立 EC2 執行個體的區域相同。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。
5. 在建立資料庫頁面上，選擇標準建立。
6. 針對引擎選項，選擇 PostgreSQL。

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input checked="" type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

7. 在範本區段中，選擇免費方案。

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.	<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info
---	---	--

8. 在 Availability and durability (可用性與耐久性) 區段中，保留預設值。
9. 在 Settings (設定) 區段中，設定這些值：
 - DB instance identifier (資料庫執行個體識別符) – 輸入 **tutorial-db-instance**。
 - Master username (主要使用者名稱) – 輸入 **tutorial_user**。
 - Auto generate a password (自動產生密碼) – 保持選項關閉。
 - Master password (主要密碼) – 輸入密碼。
 - Confirm password (確認密碼) – 重新輸入密碼。

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)

10. 在 Instance configuration (執行個體組態) 區段中，設定這些值：
 - Burstable classes (includes t classes) (高載類別 (包括 t 類別))
 - db.t3.micro

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

11. 在 Storage (儲存) 區段中，保留預設值。
12. 在 Connectivity (連線能力) 區段中，設定這些值，並且其他值都維持預設值：
 - 對於 Compute resource (運算資源)，選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源)。
 - 對於 EC2 執行個體，請選擇先前建立的 EC2 執行個體，例如教程-ec2-。instance-web-server

Connectivity Info ↻

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-1234567890abcdef0
▼

tutorial-ec2-instance-web-server

Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

13. 在 Database authentication (資料庫身分驗證) 區段中，請確定已選取 Password authentication (密碼身分驗證)。
14. 在 Additional configuration (其他設定) 區段中，輸入 **sample** 做為 Initial database name (初始資料庫名稱)。其他選項都保留預設設定。
15. 若要建立 PostgreSQL 資料庫執行個體，請選擇建立資料庫。


您的新資料庫執行個體會出現在 Databases (資料庫) 清單中，狀態為 Creating (建立中)。

16. 等待新資料庫執行個體的 Status (狀態) 顯示為 available (可用)。接著，選擇資料庫執行個體的名稱，以顯示其詳細資訊。
17. 在 Connectivity & security (連線能力與安全性) 區段中，檢視資料庫執行個體的 Endpoint (端點) 和 Port (連接埠)。

RDS > Databases > tutorial-db-instance

tutorial-db-instance

Summary

DB identifier tutorial-db-instance	CPU  2.21%
Role Instance	Current activity

[Connectivity & security](#) | [Monitoring](#) | [Logs & events](#) | [Configuration](#) | [Maintenance](#)

Connectivity & security

Endpoint & port Endpoint tutorial-db-instance.██████████-west-2.rds.amazonaws.com Port 5432	Networking Availability Zone us-west-2d VPC vpc-██████████ Subnet group default
--	--

記下資料庫執行個體的端點和連接埠。您會使用此資訊，將 Web 伺服器連接至資料庫執行個體。

18. 完成 [在您的 EC2 執行個體上安裝 Web 伺服器](#)。

在您的 EC2 執行個體上安裝 Web 伺服器

您在 EC2 中的 Linux 執行個體上安裝 Web 伺服器 [啟動 EC2 執行個體](#)。Web 伺服器將連接至在中建立的 Amazon RDS 資料庫執行個體 [建立 Amazon RDS 資料庫執行個體](#)。

使用 PHP 和 MariaDB 安裝 Apache Web 伺服器

連線至 EC2 執行個體並安裝 Web 伺服器。

連接至 EC2 執行個體並安裝支援 PHP 的 Apache Web 伺服器

1. 按照 Amazon EC2 使用者指南中的連線到 [Linux 執行個體中的步驟](#)，[Connect 到先前建立的 EC2 執行個體](#)。

建議您使用 SSH 連線至 EC2 執行個體。如果 SSH 用戶端公用程式已安裝在 Windows、Linux 或 Mac 上，您可以使用下列命令格式連線至執行個體：

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

例如，假設 `ec2-database-connect-key-pair.pem` 存放在 Linux 上的 `/dir1` 中，而 EC2 執行個體的公用 IPv4 DNS 為 `ec2-12-345-678-90.compute-1.amazonaws.com`。您的 SSH 命令看起來如下所示：

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-user@ec2-12-345-678-90.compute-1.amazonaws.com
```

2. 更新 EC2 執行個體上的軟體，以取得最新的錯誤修正和安全性更新。若要執行此操作，請使用以下命令。

Note

`-y` 選項不要求確認就會安裝更新。若要先檢查更新再安裝，請省略此選項。

```
sudo dnf update -y
```

3. 更新完成後，使用下列命令安裝 Apache Web 伺服器、PHP、MariaDB 和 PostgreSQL 軟體。此命令會同時安裝多個軟體套件和相關的依存項目。

MariaDB & MySQL

```
sudo dnf install -y httpd php php-mysqli mariadb105
```

PostgreSQL

```
sudo dnf install -y httpd php php-pgsql postgresql15
```

如果收到錯誤，您的執行個體可能不是以 Amazon Linux 2023 AMI 啟動。您可能正在改用 Amazon Linux 2 AMI。您可以使用下列命令來檢視您的 Amazon Linux 版本。

```
cat /etc/system-release
```

如需詳細資訊，請參閱[更新執行個體軟體](#)。

4. 使用如下所示的命令來啟動 Web 伺服器。

```
sudo systemctl start httpd
```

您可以測試 Web 伺服器是否已正確安裝並啟動。若要執行此動作，請在 Web 瀏覽器的網址列中輸入 EC2 執行個體的公有網域名稱系統 (DNS) 名稱，例如：<http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com>。如果 Web 伺服器正在執行，您會看到 Apache 測試頁面。

如果您沒有看到 Apache 測試頁面，請檢查您在 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#) 中建立的 VPC 安全性群組的輸入規則。請確定輸入規則包含允許 HTTP (連接埠 80) 存取連線至 Web 伺服器的 IP 地址的規則。

Note

只有當文件根目錄 `/var/www/html` 中沒有內容時，Apache 測試頁面才會出現。當您將內容新增至文件根目錄之後，您的內容會出現在 EC2 執行個體的公有 DNS 地址。在此之前，它會出現在 Apache 測試頁面上。

5. 使用 `systemctl` 命令，設定 Web 伺服器在每次系統開機時啟動。

```
sudo systemctl enable httpd
```

若要允許 `ec2-user` 管理 Apache Web 伺服器之預設根目錄中的檔案，請修改 `/var/www` 目錄的所有權和許可。有多種方法可以達成這件任務。在本教學中，您會將 `ec2-user` 新增至 `apache` 群組，以向 `apache` 群組授予 `/var/www` 目錄的所有權，並指派寫入許可。

設定 Apache Web 伺服器的檔案許可

1. 將 `ec2-user` 使用者新增至 `apache` 群組。

```
sudo usermod -a -G apache ec2-user
```

2. 登出以重新整理許可並包含新的 `apache` 群組。

```
exit
```

3. 重新登入並使用 `apache` 命令來確認 `groups` 群組存在。

```
groups
```

您的輸出結果類似如下：

```
ec2-user adm wheel apache systemd-journal
```

4. 將 `/var/www` 目錄及其內容的群組所有權變更至 `apache` 群組。

```
sudo chown -R ec2-user:apache /var/www
```

5. 變更 `/var/www` 及其子目錄的目錄許可，以新增群組寫入許可，並設定未來建立之子目錄的群組 ID。

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. 對 `/var/www` 目錄及其子目錄中的檔案，遞迴地變更許可，以新增群組寫入許可。

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

現在，`ec2-user` (以及 `apache` 群組的任何未來成員) 可以新增、刪除和編輯 Apache 文件根目錄中的檔案。這可讓您新增內容，例如靜態網站或 PHP 應用程式。

Note

執行 HTTP 通訊協定的 Web 伺服器不會為其傳送或接收的資料提供傳輸安全性。當您使用 Web 瀏覽器連線到 HTTP 伺服器時，網路路徑上任何一處的竊聽者都可以看到許多資訊。此資訊包括您造訪的 URL、您收到的網頁內容，以及任何 HTML 表單內容 (包括密碼)。保護您的 Web 伺服器的最佳實務是安裝對 HTTPS (HTTP Secure) 的支援。此通訊協定會使用 SSL/TLS 加密保護您的資料。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[教學課程：使用 Amazon Linux AMI 設定 SSL/TLS](#)。

將 Apache Web 伺服器連接至資料庫執行

接下來，您需要將內容新增至 Apache Web 伺服器 (連接到 Amazon RDS 資料庫執行個體)。

將內容新增至連接到資料庫執行個體的 Apache Web 伺服器

1. 在仍然連接至 EC2 執行個體的情況下，切換至 `/var/www` 目錄，建立名為 `inc` 的新子目錄。

```
cd /var/www
mkdir inc
cd inc
```

2. 在 `inc` 目錄中建立名為 `dbinfo.inc` 的新檔案，然後呼叫 `nano` (或您選擇的編輯器) 來編輯此檔案。

```
>dbinfo.inc
nano dbinfo.inc
```

3. 將下列內容新增至 `dbinfo.inc` 檔案：在此處，`db_instance_endpoint` 是您的資料庫執行個體端點，沒有連接埠，適用於您的資料庫執行個體。

Note

建議您將使用者名稱和密碼資訊放在不屬於 Web 伺服器文件根目錄的資料夾中。這樣做會降低您暴露安全性資訊的可能性。
確定在您的應用程式中將 `master password` 變更為合適的密碼。

```
<?php
```

```
define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');
?>
```

4. 儲存並關閉 dbinfo.inc 檔案。若您使用 nano，請以 Ctrl+S 加 Ctrl+X，儲存並關閉檔案。
5. 將目錄切換至 /var/www/html。

```
cd /var/www/html
```

6. 在 html 目錄中建立名為 SamplePage.php 的新檔案，然後呼叫 nano (或您選擇的編輯器) 來編輯此檔案。

```
>SamplePage.php
nano SamplePage.php
```

7. 將下列內容新增至 SamplePage.php 檔案：

MariaDB & MySQL

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to MySQL and select the database. */
$connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
mysqli_connect_error();

$databse = mysqli_select_db($connection, DB_DATABASE);

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);
```

```
    if (strlen($employee_name) || strlen($employee_address)) {
        AddEmployee($connection, $employee_name, $employee_address);
    }
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
    <table border="0">
        <tr>
            <td>NAME</td>
            <td>ADDRESS</td>
        </tr>
        <tr>
            <td>
                <input type="text" name="NAME" maxlength="45" size="30" />
            </td>
            <td>
                <input type="text" name="ADDRESS" maxlength="90" size="60" />
            </td>
            <td>
                <input type="submit" value="Add Data" />
            </td>
        </tr>
    </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
    <tr>
        <td>ID</td>
        <td>NAME</td>
        <td>ADDRESS</td>
    </tr>

<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
    echo "<tr>";
    echo "<td>",$query_data[0], "</td>";
    echo "<td>",$query_data[1], "</td>";
    echo "<td>",$query_data[2], "</td>";
    echo "</tr>";
}
```

```
}
?>

</table>

<!-- Clean up. -->
<?php

    mysqli_free_result($result);
    mysqli_close($connection);

?>

</body>
</html>

<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
    $n = mysqli_real_escape_string($connection, $name);
    $a = mysqli_real_escape_string($connection, $address);

    $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a')";

    if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
    if(!TableExists("EMPLOYEES", $connection, $dbName))
    {
        $query = "CREATE TABLE EMPLOYEES (
            ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            NAME VARCHAR(45),
            ADDRESS VARCHAR(90)
        )";

        if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</p>");
    }
}
}
```



```
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
    $t = mysqli_real_escape_string($connection, $tableName);
    $d = mysqli_real_escape_string($connection, $dbName);

    $checktable = mysqli_query($connection,
        "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
        AND TABLE_SCHEMA = '$d'");

    if(mysqli_num_rows($checktable) > 0) return true;

    return false;
}
?>
```

PostgreSQL

```
<?php include "../inc/dbinfo.inc"; ?>

<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to PostgreSQL and select the database. */
$constring = "host=" . DB_SERVER . " dbname=" . DB_DATABASE . " user=" .
    DB_USERNAME . " password=" . DB_PASSWORD ;
$connection = pg_connect($constring);

if (!$connection){
    echo "Failed to connect to PostgreSQL";
    exit;
}

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);
```

```
if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
}

?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
    <table border="0">
        <tr>
            <td>NAME</td>
            <td>ADDRESS</td>
        </tr>
        <tr>
            <td>
                <input type="text" name="NAME" maxlength="45" size="30" />
            </td>
            <td>
                <input type="text" name="ADDRESS" maxlength="90" size="60" />
            </td>
        </tr>
        <tr>
            <td colspan="2" style="text-align: center;">
                <input type="submit" value="Add Data" />
            </td>
        </tr>
    </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
    <tr>
        <td>ID</td>
        <td>NAME</td>
        <td>ADDRESS</td>
    </tr>

<?php

$result = pg_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = pg_fetch_row($result)) {
    echo "<tr>";
    echo "<td>",$query_data[0], "</td>";
    echo "<td>",$query_data[1], "</td>";
    echo "<td>",$query_data[2], "</td>";
    echo "</tr>";
}
```

```
}
?>
</table>

<!-- Clean up. -->
<?php

    pg_free_result($result);
    pg_close($connection);
?>
</body>
</html>

<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
    $n = pg_escape_string($name);
    $a = pg_escape_string($address);
    echo "Forming Query";
    $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a')";

    if(!pg_query($connection, $query)) echo("<p>Error adding employee data.</p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
    if(!TableExists("EMPLOYEES", $connection, $dbName))
    {
        $query = "CREATE TABLE EMPLOYEES (
            ID serial PRIMARY KEY,
            NAME VARCHAR(45),
            ADDRESS VARCHAR(90)
        )";

        if(!pg_query($connection, $query)) echo("<p>Error creating table.</p>");
    }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
    $t = strtolower(pg_escape_string($tableName)); //table name is case sensitive
```

```
$d = pg_escape_string($dbName); //schema is 'public' instead of 'sample' db
name so not using that

$query = "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME =
'$t'";
$checktable = pg_query($connection, $query);

if (pg_num_rows($checktable) >0) return true;
return false;

}
?>
```

8. 儲存並關閉 SamplePage.php 檔案。
9. 開啟 Web 瀏覽器並瀏覽至 `http://EC2 instance endpoint/SamplePage.php`，以驗證 Web 伺服器是否成功連接至執行個體，例如：`http://ec2-12-345-67-890.us-west-2.compute.amazonaws.com/SamplePage.php`。

您可以使用 SamplePage.php 將資料新增至資料庫執行個體。您新增的資料將會顯示在頁面上。若要驗證資料是否已插入資料表中，請在 Amazon EC2 執行個體上安裝 MySQL 用戶端。然後連線至資料庫執行個體，並查詢資料表。

如需安裝 MySQL 用戶端和連接至資料庫執行個體的相關資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。

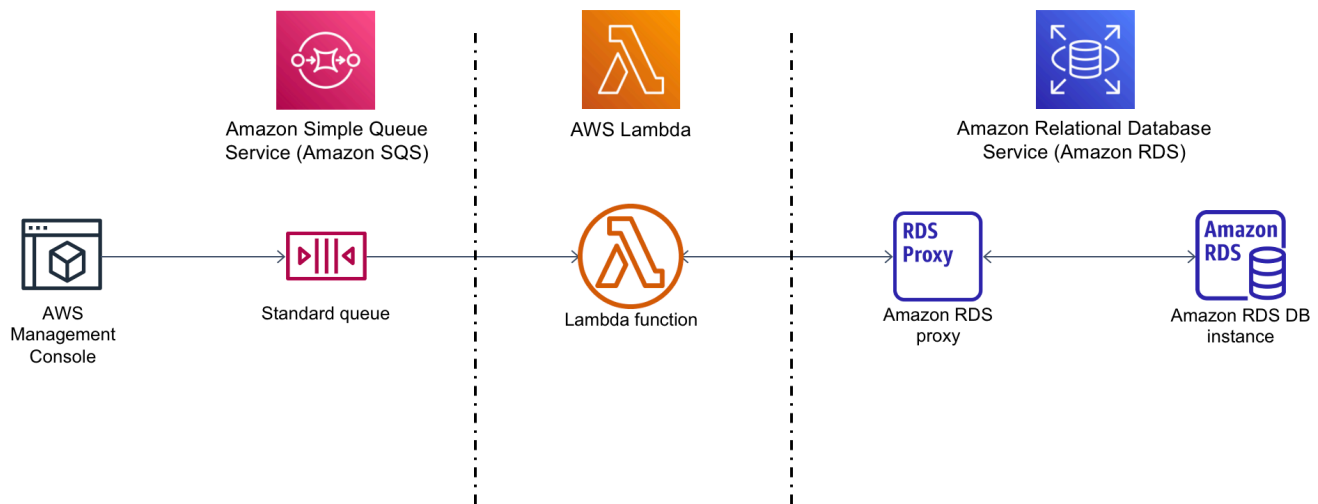
為確保資料庫執行個體盡可能安全，請確認 VPC 以外的來源無法連線到資料庫執行個體。

測試完 Web 伺服器和資料庫之後，您應該刪除資料庫執行個體和 Amazon EC2 執行個體。

- 若要刪除資料庫執行個體，請遵循 [刪除資料庫執行個體](#) 中的指示。您不需要建立最終快照。
- 若要終止 Amazon EC2 執行個體。請按照《Amazon EC2 使用指南》中[終止執行個體](#)的指示進行操作。

教學課程：使用 Lambda 函數來存取 Amazon RDS 資料庫

在本教學課程中，您會透過 RDS Proxy 使用 Lambda 函數將資料寫入 [Amazon Relational Database Service](#) (Amazon RDS) 資料庫。每當有訊息新增，您的 Lambda 函數就會從 Amazon Simple Queue Service (Amazon SQS) 佇列中讀取記錄，然後將新項目寫入資料庫中的資料表。在此範例中，您可以使用 AWS Management Console 手動將訊息新增至佇列。下圖顯示您用來完成自學課程的 AWS 資源。



透過 Amazon RDS，您可以使用常見的資料庫產品 (如 Microsoft SQL Server、MariaDB、MySQL、Oracle 資料庫和 PostgreSQL) 在雲端中執行受管關聯式資料庫。透過使用 Lambda 存取您的資料庫，您可以讀取和寫入資料以回應事件，例如在您網站上註冊的新客戶。您的函數、資料庫執行個體和代理也會自動擴展，以滿足高需求的期間。

請完成下列任務以完成本教學課程：

1. 在您 AWS 帳戶的預設虛擬私人雲端中啟動適用於 MySQL 的 RDS 資料庫執行個體和代理伺服器。
2. 建立並測試 Lambda 函數，用於在資料庫中建立新資料表並將資料寫入。
3. 建立 Amazon SQS 佇列並將其設定為在新增訊息時調用 Lambda 函數。
4. 使用 [記錄] 將訊息新增至佇列，AWS Management Console 並使用 CloudWatch 記錄監視結果，以測試完整的設定。

完成這些步驟的過程中，您將了解：

- 如何使用 Amazon RDS 建立資料庫執行個體和代理，以及將 Lambda 函數連線至代理。
- 如何使用 Lambda 在 Amazon RDS 資料庫上執行建立和讀取操作。
- 如何使用 Amazon SQS 調用 Lambda 函數。

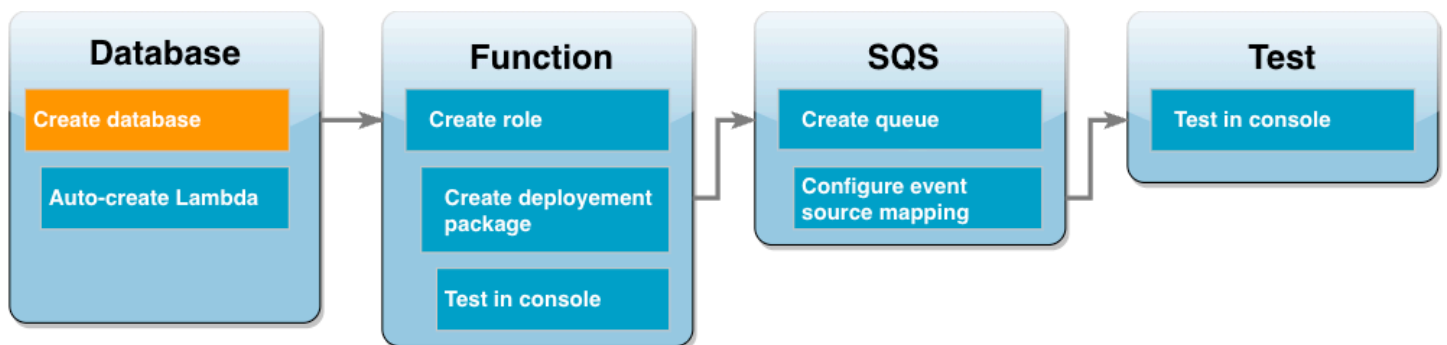
您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 來完成本自學課程。

必要條件

在開始之前，請先完成下節所含步驟：

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)

建立 Amazon RDS 資料庫執行個體



Amazon RDS 資料庫執行個體是在 AWS 雲端中執行的隔離資料庫環境。執行個體可以包含一或多個使用者建立的資料庫。除非另有指定，否則 Amazon RDS 會 VPC 您 AWS 帳戶的。如需 Amazon VPC 的詳細資訊，請參閱 [《Amazon Virtual Private Cloud 使用者指南》](#)。

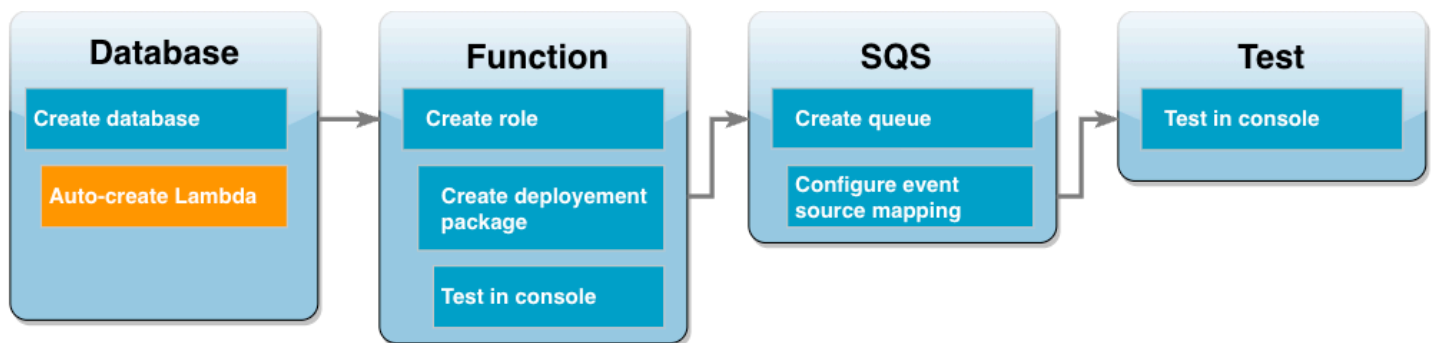
在本教學課程中，您 AWS 帳戶會在預設 VPC 中建立新執行個體，並建立 ExampleDB 在該執行個體中命名的資料庫。您可以使用 AWS Management Console 或建立資料庫執行個體和資料庫 AWS CLI。

建立資料庫執行個體

1. 開啟 Amazon RDS 主控台並選擇建立資料庫。
2. 保持選取標準建立選項，然後在引擎選項中選擇 MySQL。
3. 在範本區段中，選擇免費方案。

- 在設定中，在資料庫執行個體識別符輸入 **MySQLForLambda**。
- 執行以下操作以設定您的使用者名稱和密碼：
 - 在憑證設定中，將主要使用者名稱設為 **admin**。
 - 在主要密碼輸入並確認用於存取資料庫的密碼。
- 執行下列操作以指定資料庫名稱：
 - 保持選取所有剩餘的預設選項，並向下捲動至其他組態部分。
 - 展開此區段並在初始資料庫名稱輸入 **ExampleDB**。
- 保持選取所有剩餘的預設選項，然後選擇建立資料庫。

建立 Lambda 函數和代理



您可以使用 RDS 主控台在與資料庫相同的 VPC 中建立 Lambda 函數和代理。

Note

只有當資料庫已完成建立並處於可用狀態時，才能建立這些關聯的資源。

建立關聯的函數和代理

- 從資料庫頁面中，檢查您的資料庫是否處於可用狀態。若是，則請繼續下一個步驟。否則，請等待您的資料庫直到為可用狀態。
- 選取您的資料庫並從動作選擇設定 Lambda 連線。
- 在設定 Lambda 連線頁面上，選擇建立新函數。

將新的 Lambda 函數名稱設定為 **LambdaFunctionWithRDS**。

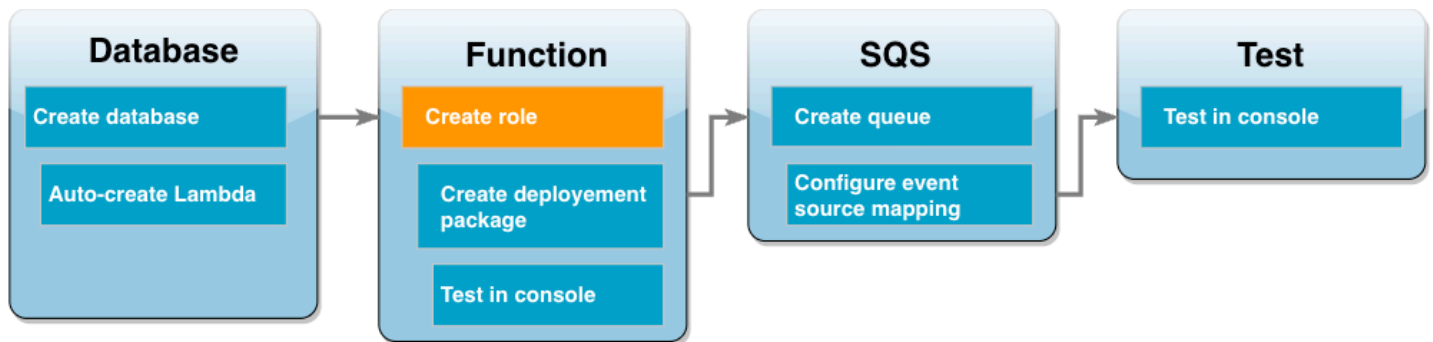
- 在 RDS Proxy 區段中，選取使用 RDS Proxy 連線選項。進一步選擇建立新的代理。

- 針對資料庫憑證，選擇資料庫使用者名稱和密碼。
- 針對使用者名稱，指定 admin。
- 針對密碼，輸入資料庫執行個體的密碼。

5. 選取設定以完成代理和 Lambda 函數的建立。

精靈會完成設定，並提供 Lambda 主控台的連結，以檢閱您的新功能。切換至 Lambda 主控台之前，請注意代理端點。

建立函數執行角色



建立 Lambda 函數之前須先建立執行角色，為函數提供必要的許可。在本教學課程中，Lambda 需要許可，才能管理與包含資料庫執行個體之 VPC 的網路連線，以及輪詢來自 Amazon SQS 佇列的訊息。

為了提供 Lambda 函數所需的許可，本教學課程使用 IAM 受管政策。這些政策會授予適用於許多常見使用案例的許可，並可在您的 AWS 帳戶中取用。如需使用受管政策的詳細資訊，請參閱：[政策最佳實務](#)。

建立 Lambda 執行角色

1. 開啟 IAM 主控台的[角色](#)頁面，然後選擇 建立角色。
2. 信任的實體類型請選擇 AWS 服務，而針對使用案例，請選擇 Lambda。
3. 選擇 下一步。
4. 執行下列操作以新增 IAM 受管政策：
 - a. 使用政策搜尋方塊，搜尋 **AWSLambdaSQSQueueExecutionRole**。
 - b. 在結果清單中，選取角色旁的核取方塊，然後選擇清除篩選條件。
 - c. 使用政策搜尋方塊，搜尋 **AWSLambdaVPCLambdaAccessExecutionRole**。

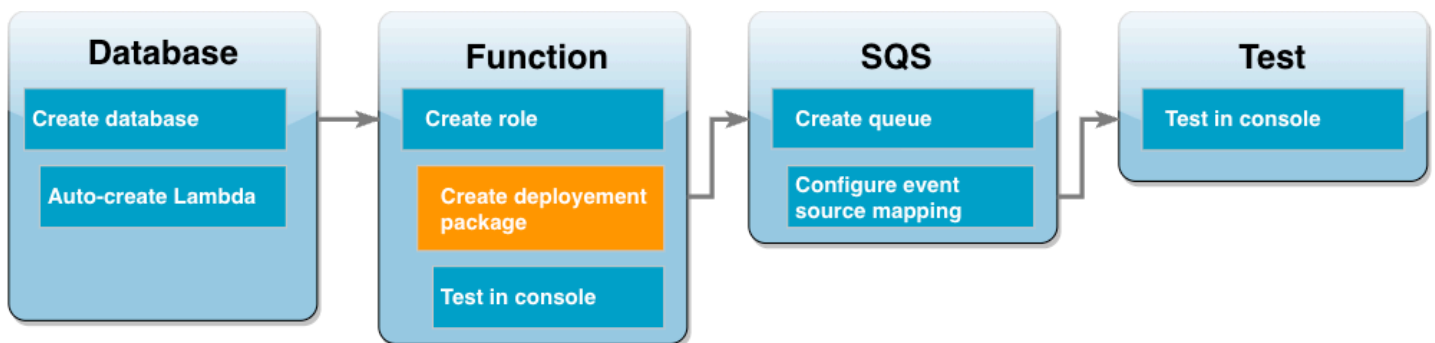
- d. 在結果清單中，選取角色旁的核取方塊，然後選擇下一步。
5. 在角色名稱輸入 **lambda-vpc-sqs-role**，然後選擇 建立角色。

在教學課程的後續階段中，需用到您剛才建立的執行角色之 Amazon Resource Name (ARN)。

尋找執行角色 ARN 的方式

1. 開啟 IAM 主控台的[角色](#)頁面，然後選擇您的角色 (lambda-vpc-sqs-role)。
2. 複製 ARN 區段中顯示的摘要。

建立 Lambda 部署套件



下面的示例 Python 代碼使用 [PyMySQL](#) 包打開到數據庫的連接。您第一次調用函數時，它還會建立一個名為 Customer 的新資料表。該資料表使用以下結構描述，其中 CustID 是主索引鍵：

```
Customer(CustID, Name)
```

該函數還使用 PyMy SQL 將記錄添加到此表中。函數會使用您要新增至 Amazon SQS 佇列的訊息中指定的客戶 ID 和名稱以新增記錄。

程式碼會在處理常式函數之外建立與資料庫的連線。在初始化程式碼中建立連線，允許後續的函數調用重複使用連線並提高效率。在生產應用程式中，您也可以使用[佈建並行](#)初始化請求的資料庫連線數目。只要函數調用後就可以使用這些連線。

```
import sys
import logging
import pymysql
import json
import os
```

```
# rds settings
user_name = os.environ['USER_NAME']
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_PROXY_HOST']
db_name = os.environ['DB_NAME']

logger = logging.getLogger()
logger.setLevel(logging.INFO)

# create the database connection outside of the handler to allow connections to be
# re-used by subsequent function invocations.
try:
    conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password,
        db=db_name, connect_timeout=5)
except pymysql.MySQLError as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
    logger.error(e)
    sys.exit(1)

logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")

def lambda_handler(event, context):
    """
    This function creates a new RDS database table and writes records to it
    """
    message = event['Records'][0]['body']
    data = json.loads(message)
    CustID = data['CustID']
    Name = data['Name']

    item_count = 0
    sql_string = f"insert into Customer (CustID, Name) values(%s, %s)"

    with conn.cursor() as cur:
        cur.execute("create table if not exists Customer ( CustID int NOT NULL, Name
varchar(255) NOT NULL, PRIMARY KEY (CustID))")
        cur.execute(sql_string, (CustID, Name))
        conn.commit()
        cur.execute("select * from Customer")
        logger.info("The following items have been added to the database:")
        for row in cur:
            item_count += 1
            logger.info(row)
    conn.commit()
```

```
return "Added %d items to RDS for MySQL table" %(item_count)
```

Note

在此範例中，您的資料庫存取憑證會儲存為環境變數。在生產應用程式中，我們建議您將 [AWS Secrets Manager](#) 用作更安全的選項。請注意，如果您的 Lambda 函數在 VPC 中，若要連線到 Secrets Manager，您需要建立 VPC 端點。請參閱 [How to connect to Secrets Manager service within a Virtual Private Cloud](#)，以了解更多內容。

若要在函數程式碼中包含 PyMy SQL 相依性，請建立 .zip 部署套件。下列指令適用於 Linux、macOS 或 Unix：

建立 .zip 部署套件的方式

1. 將程式碼範例儲存為名為 lambda_function.py 的檔案。
2. 在您建立 lambda_function.py 檔案的相同目錄中，建立名為的新目錄 package 並安裝 PyMy SQL 程式庫。

```
mkdir package
pip install --target package pymysql
```

3. 創建一個包含您的應用程序代碼和 PyMy SQL 庫的 zip 文件。在 Linux 或 MacOS 中，執行下列 CLI 命令。在 Windows 中，使用您偏好的 zip 工具建立 lambda_function.zip 檔案。您的 lambda_function.py 原始程式碼檔案和包含相依項的資料夾必須安裝在 .zip 檔案的根目錄中。

```
cd package
zip -r ../lambda_function.zip .
cd ..
zip lambda_function.zip lambda_function.py
```

您也可以使用 Python 虛擬環境建立部署套件。請參閱 [使用 .zip 封存檔部署 Python Lambda 函數](#)

更新 Lambda 函數

現在您可以使用剛才建立的 .zip 套件，使用 Lambda 主控台更新 Lambda 函數。若要讓函數存取資料庫，還需要使用存取憑證來設定環境變數。

更新 Lambda 函數

1. 開啟 Lambda 主控台的[函數](#)頁面，然後選擇您的函數 LambdaFunctionWithRDS。
2. 在 [執行階段設定] 索引標籤中，選取 [編輯]，將函數的執行階段變更為 Python 3.10。
3. 將處理器變更為 `lambda_function.lambda_handler`。
4. 在程式碼標籤中，選擇上傳來源，然後選擇 .zip 檔案。
5. 選取您在上一個階段建立的 `lambda_function.zip` 檔案，然後選擇儲存。

現在，請使用您先前建立的執行角色設定函數。這會授予函數存取資料庫執行個體和輪詢 Amazon SQS 佇列所需的許可。

函數的執行角色。

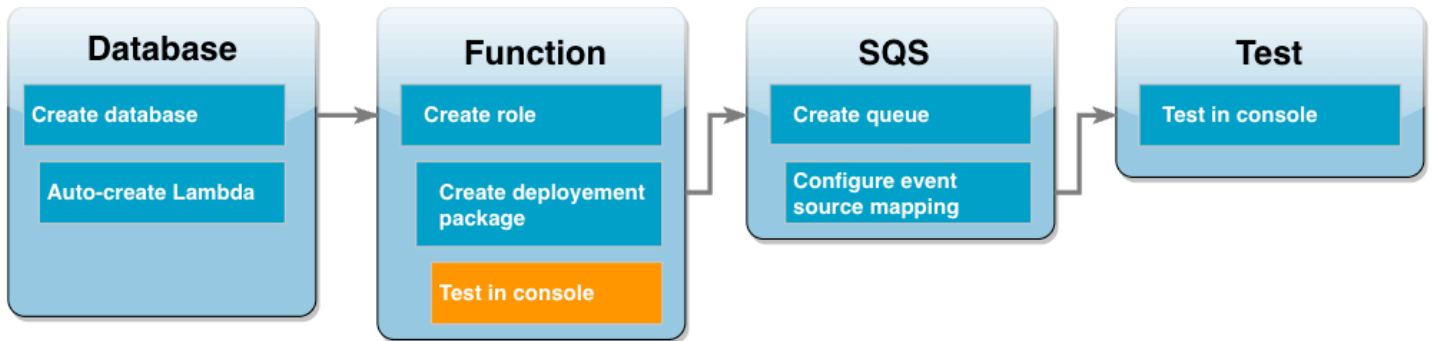
1. 在 Lambda 主控台的[函數](#)頁面中，選取組態索引標籤，然後選擇許可。
2. 在執行角色中，選擇編輯。
3. 在現有角色中，選擇您的執行角色 (`lambda-vpc-sqs-role`)。
4. 選擇儲存。

設定函數的環境變數

1. 在 Lambda 主控台的[函數](#)頁面中，選取組態索引標籤，然後選擇環境變數。
2. 選擇編輯。
3. 若要新增資料庫存取憑證，請執行下列動作：
 - a. 選擇新增環境變數，然後針對索引鍵輸入 **USER_NAME**，針對值輸入 **admin**。
 - b. 選擇新增環境變數，然後針對索引鍵輸入 **DB_NAME**，針對值輸入 **ExampleDB**。
 - c. 選擇新增環境變數，然後針對索引鍵輸入 **PASSWORD**，針對值輸入您在建立資料庫時選擇的密碼。
 - d. 選擇新增環境變數，然後針對索引鍵輸入 **RDS_PROXY_HOST**，針對值輸入您先前備註的 RDS Proxy 端點。

e. 選擇儲存。

在主控台中測試您的 Lambda 函數



您現在可使用 Lambda 主控台來測試您的函數。請建立一個測試事件，該事件會模擬您在教學課程的最後階段中，使用 Amazon SQS 調用函數時會接收到的資料。測試事件會包含一個 JSON 物件，指定要新增到函數所建立 Customer 資料表中的客戶 ID 和客戶名稱。

測試 Lambda 函數

1. 開啟 Lambda 主控台的[函數](#)頁面，然後選擇您的函數。
2. 選擇測試區段。
3. 選擇建立新活動並輸入 **myTestEvent** 對於事件的名稱。
4. 將下列程式碼複製到事件 JSON 中，然後選擇儲存。

```

{
  "Records": [
    {
      "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
      "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "body": "{\n  \"CustID\": 1021,\n  \"Name\": \"Martha Rivera\"\n}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1545082649183",
        "SenderId": "AIDAIENQZJOL023YVJ4V0",
        "ApproximateFirstReceiveTimestamp": "1545082649185"
      },
      "messageAttributes": {},
      "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-west-2:123456789012:my-queue",
    }
  ]
}
  
```

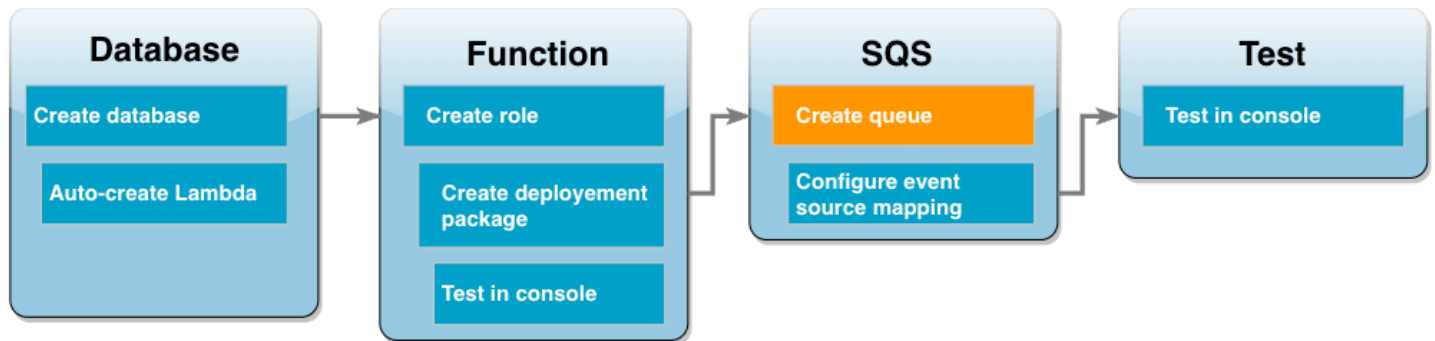
```
    "awsRegion": "us-west-2"  
  }  
]  
}
```

5. 選擇 測試。

在執行結果索引標籤中，您應該會看到類似函數日誌中顯示的結果：

```
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f The following  
items have been added to the database:  
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f (1021, 'Martha  
Rivera')
```

建立 Amazon SQS 佇列

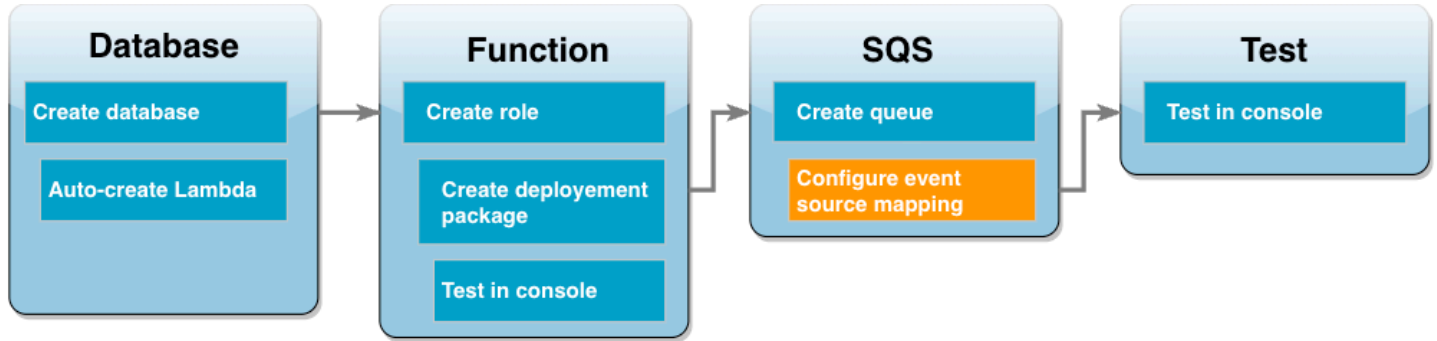


您已成功測試 Lambda 函數和 Amazon RDS 資料庫執行個體的整合。現在請建立 Amazon SQS 佇列，用於在教學課程的最後階段調用您的 Lambda 函數。

建立 Amazon SQS 佇列的方式 (主控台)

1. 開啟 Amazon SQS 主控台的 [佇列](#) 頁面，然後選取建立佇列。
2. 將類型保留為標準，然後在佇列名稱輸入 **LambdaRDSQueue**。
3. 保持選取所有預設選項，然後選擇建立佇列。

建立事件來源映射以調用 Lambda 函數



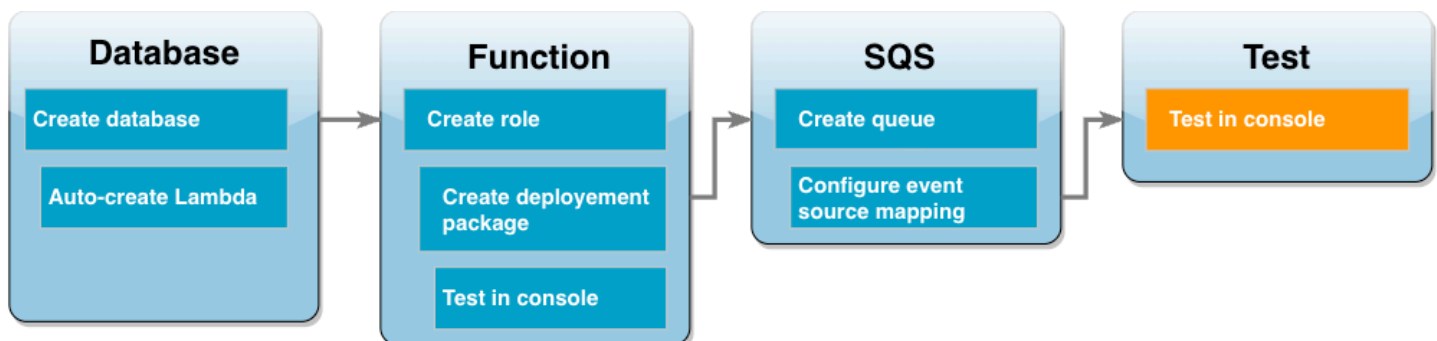
事件來源映射是一種 Lambda 資源，它可從串流或佇列中讀取項目並調用 Lambda 函數。設定事件來源映射時，您可以指定批次大小，以便將串流或佇列中的記錄批次處理成單一承載。在此範例中，您將批次大小設定為 1，以便在每次傳送訊息到佇列時都調用 Lambda 函數。您可以使用 AWS CLI 或 Lambda 主控台來設定事件來源對應。

建立事件來源映射的方式 (主控台)

1. 開啟 Lambda 主控台的 [函數](#) 頁面，然後選擇您的函數 (LambdaFunctionWithRDS)。
2. 在函數概觀區段中，選擇新增觸發條件。
3. 來源請選取 Amazon SQS，然後選取佇列的名稱 (LambdaRDSQueue)。
4. 批次大小請輸入 **1**。
5. 將所有其他選項設定保留為預設值，然後選擇新增。

您現在可以透過將訊息新增至 Amazon SQS 佇列以測試完整的設定。

測試和監控設定



若要測試完整的設定，請使用主控台將訊息新增至 Amazon SQS 佇列。然後，您可以使用記 CloudWatch 錄來確認 Lambda 函數正如預期將記錄寫入資料庫。

測試和監控設定的方式

1. 開啟 Amazon SQS 主控台的[佇列](#)頁面，然後選取您的佇列 (LambdaRDSQueue)。
2. 選擇傳送和接收訊息，然後將以下 JSON 貼到傳送訊息區段裡的訊息內文中。

```
{
  "CustID": 1054,
  "Name": "Richard Roe"
}
```

3. 選擇 傳送訊息。

將訊息傳送至佇列會導致 Lambda 透過事件來源映射調用您的函數。若要確認 Lambda 已如預期般叫用您的函數，請使用 CloudWatch 記錄來確認函數是否已將客戶名稱和 ID 寫入您的資料庫資料表。

4. 開啟主 CloudWatch 主控台的 [\[記錄群組\]](#) 頁面，然後選取函數的記錄群組 (/aws/lambda/LambdaFunctionWithRDS)。
5. 在日誌串流區段中，選擇最新的日誌串流。

您的資料表應該會包含兩個客戶記錄，分別來自您的函數的每次調用。在日誌串流中，您應該會看到類似以下的訊息：

```
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 The following
items have been added to the database:
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1021, 'Martha
Rivera')
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1054,
'Richard Roe')
```

清除您的資源

除非您想要保留為此教學課程建立的資源，否則您現在便可刪除。刪除不再使用的 AWS 資源，即可避免 AWS 帳戶產生不必要的費用。

若要刪除 Lambda 函數

1. 開啟 Lambda 主控台中的 [函數頁面](#)。
2. 選擇您建立的函數。

3. 選擇 Actions (動作)、Delete (刪除)。
4. 選擇 Delete (刪除)。

若要刪除執行角色

1. 開啟 IAM 主控台中的 [角色頁面](#)。
2. 選取您建立的執行角色。
3. 選擇 Delete role (刪除角色)。
4. 選擇 Yes, delete (是, 刪除)。

刪除 MySQL 資料庫執行個體

1. 在 Amazon RDS 主控台開啟 [Databases \(資料庫\) 頁面](#)。
2. 選取您建立的資料庫。
3. 選擇 動作、刪除。
4. 取消勾選 建立最終快照 核取方塊。
5. 在文字方塊中輸入 **delete me**。
6. 選擇 刪除。

刪除 Amazon SQS 佇列

1. 登入 AWS Management Console 並開啟 Amazon SQS 主控台，網址為 <https://console.aws.amazon.com/sqs/>。
2. 選取您建立的佇列。
3. 選擇 刪除。
4. 在文字方塊中輸入 **delete**。
5. 選擇 刪除。

Amazon RDS 教學課程和範本程式碼

本 AWS 文件包含數個教學課程，可引導您完成常見的 Amazon RDS 使用案例。其中許多教學課程會示範如何將 Amazon RDS Amazon 與其他 AWS 服務搭配使用。此外，您還可以在中存取範例程式碼 GitHub。

Note

您可以在 [AWS 資料庫部落格](#) 找到更多教學課程。如需培訓的詳細資訊，請參閱 [AWS 培訓與認證](#)。

主題

- [本指南中的教學課程](#)
- [其他 AWS 指南中的教學](#)
- [AWS Amazon RDS Amazon PostgreSQL 討會和實驗室內容入口網站](#)
- [AWS Amazon RDS Amazon MySQL 的研討會和實驗室內容門戶](#)
- [教學課程和範例程式碼 GitHub](#)
- [搭配 AWS SDK 使用此服務](#)

本指南中的教學課程

本指南中的以下教學課程會向您展示如何使用 Amazon RDS 執行常見的任務。

- [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)

了解如何在以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中包括資料庫執行個體。在此情況下，VPC 會與在相同 VPC 的 Amazon EC2 執行個體上執行的 Web 伺服器共用資料。

- [教學課程：建立要與資料庫執行個體搭配使用的 \(VPC\)\(雙堆疊模式\)](#)

了解如何在以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中包括資料庫執行個體。在此情況下，VPC 會與相同 VPC 中的 Amazon EC2 執行個體共用資料。於本教學課程中，您將為此案例建立與以雙堆疊模式執行之資料庫搭配使用的 VPC。

- [教學：建立 Web 伺服器和 Amazon RDS 資料庫執行個體](#)

了解如何安裝支援 PHP 的 Apache Web 伺服器，並建立 MySQL 資料庫。Web 伺服器在使用 Amazon Linux 的 Amazon EC2 執行個體上執行，而 MySQL 資料庫是 MySQL 資料庫執行個體，。 Amazon EC2 執行個體和資料庫執行個體皆在 Amazon VPC 中執行。

- [教學課程：從資料庫快照還原 Amazon RDS 資料庫執行個體](#)

了解如何從資料庫快照還原資料庫執行個體

- [教學課程：使用 Lambda 函數來存取 Amazon RDS 資料庫](#)

了解如何從 RDS 主控台建立 Lambda 函數來透過代理存取資料庫、建立資料表、新增一些記錄，以及從資料表擷取記錄。您也會學習如何叫用 Lambda 函數，並確認查詢結果。

- [教學課程：使用標籤指定要停止的資料庫執行個體](#)

了解如何使用標籤指定要停止哪些資料庫執行個體

- [教學：使用 Amazon 記錄資料庫執行個體狀態變更 EventBridge](#)

了解如何使用 Amazon EventBridge 和記錄資料庫執行個體狀態變更 AWS Lambda。

- [教學課程：為多可用區域資料庫叢集複本延遲建立 Amazon CloudWatch 警示](#)

了解如何在異地同步備份資料庫叢集的複本延遲超過閾值時建立傳送 Amazon SNS 訊息的 CloudWatch 警示。警示會在您指定的期間監看 ReplicaLag 指標。此動作是傳送到 Amazon SNS 主題或 Amazon EC2 Auto Scaling 政策的通知。

其他 AWS 指南中的教學

其他 AWS 指南中的下列教學課程說明如何使用 Amazon RDS Amazon 執行一般任務：

- [教學課程：旋轉AWS Secrets Manager 使用者指南中 AWS 資料庫的密碼](#)

了解如何為 AWS 資料庫建立密碼，以及如何設定密碼以按排程輪換。手動觸發一個輪換，然後確認秘密新版本持續提供存取。

- AWS Elastic Beanstalk 開發人員指南中的[教學課程和範例](#)

了解如何部署搭配使用 Amazon RDS 資料庫的應用程式 AWS Elastic Beanstalk。

- 在 Amazon Machine Learning Developer Guide 中[使用 Amazon RDS 資料庫中的資料建立 Amazon ML 資料來源](#)

了解如何從存放在 MySQL 資料庫執行個體的資料中建立 Amazon Machine Learning (Amazon ML) 資料來源物件。

- 在 [Amazon QuickSight 用戶指南中手動啟用對 VPC 中的 Amazon RDS 執行個體的存取](#)

了解如何在 VPC 中啟 QuickSight 用 Amazon RDS 資料庫執行個體的 Amazon 存取權。

AWSAmazon RDS Amazon PostgreSQL 研討會和實驗室內容入口網站

以下精選的研討會和其他實作內容可協助您了解 Amazon RDS PostgreSQL 的特性和功能：

- [建立資料庫執行個體](#)

了解如何建立資料庫執行個體。

- [使用 RDS 工具進行效能監控](#)

了解如何使用 AWS 和 SQL 工具 (Cloudwatch、增強型監控、慢速查詢記錄、Performance Insights、PostgreSQL 目錄檢視) 來了解效能問題，並找出改善資料庫效能的方法。

AWSAmazon RDS Amazon MySQL 的研討會和實驗室內容門戶

以下精選的研討會和其他實作內容可協助您了解 Amazon RDS MySQL 的特性和功能：

- [建立資料庫執行個體](#)

了解如何建立資料庫執行個體。

- [使用績效詳情](#)

了解如何使用績效詳情來監控和調整資料庫執行個體。

教學課程和範例程式碼 GitHub

中的下列教學課程和範例程式碼將說 GitHub 明如何使用 Amazon RDS Amazon 執行常見任務：

- [建立 Amazon Relational Database Service 項目追蹤器](#)


了解如何建立一個應用程式，追蹤和報告工作項目。此應用程式會使用 Amazon RDS、Amazon Simple Email Service、Elastic Beanstalk 和 SDK for Java 2.x。

搭配 AWS SDK 使用此服務

AWS 軟件開發套件 (SDK) 可用於許多流行的編程語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
AWS SDK for C++	AWS SDK for C++ 程式碼範例
AWS CLI	AWS CLI 程式碼範例
AWS SDK for Go	AWS SDK for Go 程式碼範例
AWS SDK for Java	AWS SDK for Java 程式碼範例
AWS SDK for JavaScript	AWS SDK for JavaScript 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
AWS SDK for .NET	AWS SDK for .NET 程式碼範例
AWS SDK for PHP	AWS SDK for PHP 程式碼範例
AWS Tools for PowerShell	PowerShell 程式碼範例的工具
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 程式碼範例
AWS SDK for Ruby	AWS SDK for Ruby 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

如需此服務的特定範例，請參閱 [使用 AWS 開發套件的 Amazon RDS 的程式碼範例](#)。

 可用性範例

找不到所需的內容嗎？請使用本頁面底部的提供意見回饋連結申請程式碼範例。

Amazon RDS 的最佳實務

了解使用 Amazon RDS 的最佳實務。在新最佳實務確定時，我們會不斷更新此小節。

主題

- [Amazon RDS 基本操作準則](#)
- [資料庫執行個體 RAM 建議](#)
- [AWS 資料庫驅動](#)
- [使用 Enhanced Monitoring 來識別作業系統問題](#)
- [使用指標來識別效能問題](#)
- [調校查詢](#)
- [使用 MySQL 的最佳實務](#)
- [使用 Oracle 的最佳實務](#)
- [使用 Oracle 的最佳實務](#)
- [使用 PostgreSQL 的最佳實務](#)
- [使用 SQL Server 的最佳實務](#)
- [使用資料庫參數群組](#)
- [自動建立資料庫執行個體的最佳實務](#)
- [Amazon RDS 新功能視頻](#)

Note

如需 Amazon RDS 的常用建議，請參閱 [檢視和回應 Amazon RDS 建議](#)。

Amazon RDS 基本操作準則

下列是使用 Amazon RDS 時，每個使用者應遵循的基本操作準則。請注意，Amazon RDS 服務水準協議需要您遵循這些準則：

- 使用指標監控您的記憶體、CPU、複本延遲和儲存體用量。您可以設定 Amazon CloudWatch 在使用模式變更或部署接近容量限制時通知您。這可讓您維持系統效能和可用性。

- 在您處理儲存容量限制時向上擴展資料庫執行個體。您應該在儲存體和記憶體中具有一些緩衝，以容納應用程式需求中未知的增加。
- 啟用自動備份並將備份時段設定在每日寫入 IOPS 較低的期間發生。此時備份對資料庫使用情況產生的干擾最少。
- 如果資料庫工作負載所需的 I/O 較您佈建得多，容錯移轉或資料庫失敗之後的復原將會很緩慢。若要增加資料庫執行個體的 I/O 容量，請執行下列任何或所有動作：
 - 遷移至具有高 I/O 容量的其他資料庫執行個體類別。
 - 根據您需要的增加量而定，從磁帶儲存體轉換成一般用途或佈建 IOPS 儲存體。如需可用儲存體類型的詳細資訊，請參閱[Amazon RDS 儲存類型](#)。

如果轉換為佈建 IOPS 儲存體，請確定您也使用已針對佈建 IOPS 最佳化的資料庫執行個體類別。如需佈建的 IOPS 的詳細資訊，請參閱[佈建 IOPS SSD 儲存體](#)。

- 如果您已在使用佈建 IOPS 儲存體，請佈建額外的傳輸量容量。
- 如果您的用戶端應用程式快取資料庫執行個體的網域名稱服務 (DNS) 資料，請將 time-to-live (TTL) 值設定為小於 30 秒。資料庫執行個體的基礎 IP 位址可能會在容錯移轉後變更。長時間快取 DNS 資料可能從而導致連線失敗。您的應用程式可能會嘗試連線到不再提供服務的 IP 地址。
- 測試資料庫執行個體的容錯移轉，以了解您的特定使用案例執行此程序需時多長。也會測試容錯移轉，以確保可存取您資料庫執行個體的應用程式，可以在容錯移轉發生之後，自動連線到新的資料庫執行個體。

資料庫執行個體 RAM 建議

Amazon RDS 的效能最佳實務是配置足夠的 RAM，使得您的工作集幾乎能完全在記憶體中。工作集是您的執行個體經常使用的資料和索引。您越常使用資料庫執行個體，工作集會成長越多。

若要判斷您的工作集是否幾乎全部都在記憶體中，請在資料庫執行個體負載時檢查 ReadIOPS 指標 (使用 Amazon CloudWatch)。ReadIOPS 的值應該很小並且穩定。在某些情況下，將資料庫執行個體類別縱向擴展至擁有更多 RAM 的類別，這會導致 ReadIOPS 大幅下降。在這些情況下，您的工作集幾乎完全不在記憶體中。請繼續向上擴展，直到擴展操作之後 ReadIOPS 不再大幅下降，或 ReadIOPS 減少至非常小的數量為止。如需監控資料庫執行個體之指標的詳細資訊，請參閱在[Amazon RDS 主控台中檢視指標](#)。

AWS 資料庫驅動

我們建議您使用應用程式連線的驅動程式 AWS 套件。這些驅動程式的設計旨在提供更快的切換和容錯移轉時間，以及使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合

身分進行身份驗證的支援。AWS 驅動程式仰賴監控資料庫執行個體狀態並瞭解執行個體拓撲來判斷新的寫入器。這種方法可將切換和容錯移轉時間縮短為個位數秒，而開放原始碼驅動程式則需要數十秒。

隨著新的服務功能推出，驅動程序 AWS 套件的目標是內置支持這些服務功能。

如需詳細資訊，請參閱 [使用 AWS 驅動程式連接資料庫執行個體](#)。

使用 Enhanced Monitoring 來識別作業系統問題

啟用「增強型監控」時，Amazon RDS 會即時提供執行資料庫執行個體在其上執行之作業系統 (OS) 的指標。您可以使用主控台來檢視資料庫執行個體的指標。您也可以在您選擇的監控系統中使用 Amazon CloudWatch 日誌中的增強型監控 JSON 輸出。如需增強型監控的詳細資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

使用指標來識別效能問題

若要識別由於資源不足和其他常見瓶頸造成的效能問題，您可以監控 Amazon RDS 資料庫執行個體的可用指標。

檢視效能指標

您應該定期監控效能指標以查看各種時間範圍的平均值、最大值和最小值。如果這麼做，當效能降級時您便可以得知。您也可以為特定的測量結果閾值設定 Amazon CloudWatch 警示，以便在達到警示時收到警示。

為了對效能問題進行故障診斷，請務必了解系統的基礎效能。當您設定資料庫執行個體並使用一般工作負載執行它時，請擷取所有效能指標的平均值、最大值和最小值。以數個不同的間隔（例如，一小時、24 小時、一週、兩週）進行此操作。這可讓您了解正常運作情況為何。這有助於比較尖峰與離峰時段的操作。之後您可以使用此資訊來得知效能是否下降至標準層級之下。

如果使用多可用區域資料庫叢集，請監控寫入器資料庫執行個體上的最新交易與讀取器資料庫執行個體上最新套用交易之間的時間差異。這種差異被稱為複本延遲。如需詳細資訊，請參閱 [複本延遲和多可用區域資料庫叢集](#)。

您可以在「Performance Insights」儀表中檢視合併的「Performance Insights」和 CloudWatch 指標，並監控資料庫執行個體。若要使用此監控檢視，必須為您的資料庫執行個體開啟績效詳情。如需此監控檢視的相關資訊，請參閱 [在 Amazon RDS 主控台中檢視組合指標](#)。

您可以針對特定時間區間建立效能分析報告，並檢視所識別出的洞見和解決問題的建議。若要取得更多資訊，請參閱 [建立績效分析報告](#)。

檢視效能指標

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇一個資料庫執行個體。
3. 選擇 Monitoring (監控)。

儀表板提供效能指標。這些指標預設為顯示過去三小時的資訊。

4. 使用右上角的編號按鈕，來切換至其他指標頁面，或調整設定以查看更多指標。
5. 選擇用來調整時間範圍的效能指標，以查看當日以外的資料。您可以變更 Statistic (統計資料)、Time Range (時間範圍) 和 Period (期間) 值來調整顯示的資訊。例如，您可能想要查看某個指標過去兩週每天的尖峰值。若是如此，請將 Statistic (統計資料) 設定為 Maximum (最大值)、將 Time Range (時間範圍) 設定為 Last 2 Weeks (過去 2 週)、將 Period (期間) 設定為 Day (日)。

您也可以使用 CLI 或 API 來檢視效能指標。如需詳細資訊，請參閱 [在 Amazon RDS 主控台中檢視指標](#)。

設定 CloudWatch 鬧鐘

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇一個資料庫執行個體。
3. 選擇 Logs & events (日誌和事件)。
4. 在 [鬧CloudWatch 鐘] 區段中，選擇 [建立鬧鐘]。

Create alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

Settings

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send notifications

Yes
 No

Send notifications to

ARN
 New email or SMS topic

Topic name

Name of the topic.

With these recipients

Email addresses or phone numbers of SMS enabled devices to send the notifications to

Metric

Average ▼ of CPU Utilization ▼

Threshold

>= ▼ Percent

Evaluation period

1 consecutive period(s) of 5 Minutes ▼

Figure: CPU Utilization Percent

mydbinstancecf

Name of alarm

- 針對 寄送通知，選擇 Yes (是)，針對 Send notifications to (寄送通知至)，選擇 New email or SMS topic (新的電子郵件或 SMS 主題)。

6. 針對 Topic name (主題名稱)，輸入通知的名稱，並針對 With these recipients (含有以下收件人)，輸入以逗號分隔的電子郵件地址或電話號碼清單。
7. 針對 Metric (指標)，選擇警示數統計資料和指標來設定。
8. 針對 Threshold (臨界值)，指定指標是否必須大於、小於或等於閾值，並指定閾值。
9. 針對 Evaluation Period (評估期間)，選擇警示的評估期間。針對 consecutive period(s) of (連續期間)，選擇臨界值必須符合才能觸發警示的期間。
10. 針對 Name of alarm (警示名稱)，輸入警示的名稱。
11. 選擇 Create Alarm (建立警示)。

鬧鐘會出現在 CloudWatch 鬧鐘區段中。

評估效能指標

資料庫執行個體有一些不同類別的指標，而如何判斷可接受的值取決於指標。

CPU

- CPU 使用率 – 已使用的電腦處理容量之百分比。

記憶體

- 可用記憶體 — 資料庫執行個體上有多少 RAM 可用，以位元組為單位。在監控標籤指標中會針對 CPU、記憶體和儲存體指標，將紅線標記在 75%。如果執行個體記憶體的耗用量經常超過該線，這表示您應該檢查工作負載或升級執行個體。
- 交換使用量 — 資料庫執行個體使用了多少交換空間，以位元組為單位。

磁碟空間

- 可用儲存空間 – 資料庫執行個體目前尚未使用的磁碟空間 (以 MB 為單位)。

輸入/輸出操作

- 讀取 IOPS、寫入 IOPS – 磁碟讀取或寫入操作的每秒平均次數。
- 讀取延遲、寫入延遲 – 讀取或寫入操作的平均時間 (以毫秒為單位)。
- 讀取傳輸量、寫入傳輸量 – 每秒自磁碟讀取或寫入至其中的平均數量 (以 MB 為單位)。

- 佇列深度 – 等候寫入至磁碟或從其中讀取的 I/O 操作數量。

網路流量

- 網路接收輸送量、網路傳輸輸送量 – 每秒自資料庫執行個體來回傳送的網路流量速度 (以位元組為單位)。

資料庫連線

- 資料庫連線 – 連線至資料庫執行個體的用戶端工作階段數量。

如需可用效能指標的個別詳細說明，請參閱[使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。

一般來說，效能指標的可接受值視您的基準為何，以及您的應用程式所做之事而定。調查距離基準的一致或趨勢變異。關於特定類型指標的建議如下所示：

- 高 CPU 或 RAM 耗用量 – CPU 或 RAM 耗用量的高值可能是適當的。例如，如果它們與應用程式的目標 (如輸送量或並行) 一致且是預期的，它們可能就是這樣。
- 磁碟空間消耗量 – 如果使用的空間持續保持在等於或高於總磁碟空間的 85%，請調查磁碟空間消耗量。看看從執行個體刪除資料或將資料封存至不同的系統來釋出空間是否可行。
- 網路流量 – 對於網路流量，請洽系統管理員，以了解您的網域網路和網際網路連線預期的輸送量。調查網路流量的傳輸量是否如預期一致地降低。
- 資料庫連線 – 如果您看到大量使用者連線，同時執行個體效能下降且回應時間延長，請考慮限制資料庫連線。資料庫執行個體使用者連線的最佳數量，將因執行個體類別和要執行的操作複雜性而不同。若要判定資料庫連線的數目，請將資料庫執行個體與參數群組建立關聯。在此群組中，將 User Connections (使用者連線) 參數設定為 0 (無限制) 以外的值。您可以使用現有的參數群組或建立新的參數群組。如需更多詳細資訊，請參閱[使用參數群組](#)。
- IOPS 指標 – IOPS 指標的預期值視磁碟規格和伺服器組態而定，因此請使用您的基準來了解何謂典型。調查值是否與您的基準一致地不同。為獲得最佳 IOPS 效能，請確定您的一般工作集將放入記憶體中，以將讀取和寫入操作降到最低。

對於效能指標的問題，改善效能的第一步是調整最常用和最昂貴的查詢。調整它們，以查看這樣做是否會降低系統資源的壓力。如需詳細資訊，請參閱[調校查詢](#)。

如果您的查詢經過調整且問題仍然存在，請考慮升級您的 Amazon RDS [資料庫執行個體類別](#)。您可能將其升級至具有更多資源 (CPU、RAM、磁碟空間、網路頻寬、I/O 容量)，且與問題相關的類別。

調校查詢

改善資料庫執行個體效能的一個最佳方式是調校最常使用與資源最密集的查詢。在這裡，您可以調整它們，讓它們的執行成本更便宜。如需有關改善查詢的資訊，請使用下列資源：

- MySQL – 請參閱 MySQL 文件中的[最佳化 SELECT 陳述式](#)。如需其他查詢調校資源，請參閱[MySQL 效能調校和最佳化資源](#)。
- Oracle – 請參閱 Oracle 資料庫文件中的[資料庫 SQL 調校指南](#)。
- SQL 伺服器 – 請參閱 Microsoft 文件中的[分析查詢](#)。您也可以使用 Microsoft 文件中[系統動態管理檢視](#)中所述的與執行、索引和 I/O 相關的資料管理檢視 (DMV)，對 SQL Server 查詢問題進行故障診斷。

查詢調校的一般層面為建立有效的索引。如需資料庫執行個體的潛在索引改進，請參閱 Microsoft 文件中的[Database Engine Tuning Advisor](#)。如需在 RDS for SQL Server 上使用 Tuning Advisor 的資訊，請參閱[使用 Database Engine Tuning Advisor 分析 Amazon RDS for SQL Server 資料庫執行個體上的資料庫工作負載](#)。

- PostgreSQL – 請參閱 PostgreSQL 文件中的[使用 EXPLAIN](#)，瞭解如何分析查詢計畫。您可以使用此資訊來修改查詢或基礎資料表以改善查詢效能。

有關如何在您的查詢中指定聯結以獲得最佳效能的資訊，請參閱[使用明確 JOIN 子句來控制計畫者](#)。

- MariaDB – 請參閱 MariaDB 文件中的[查詢最佳化](#)。

使用 MySQL 的最佳實務

MySQL 資料庫中的資料表大小和數量都會影響效能。

資料表大小

一般而言，作業系統對檔案大小的約束會決定 MySQL 資料庫的有效資料表大小上限。所以，限制通常不是由內部 MySQL 約束所決定的。

在 MySQL 資料庫執行個體上，請避免讓資料庫中的資料表變得太大。雖然一般儲存體限制為 64 TiB，但佈建的儲存體限制會將 MySQL 資料表檔案的大小上限限制在 16 TiB。請分割大型資料表，使得檔案大小不超過 16 TiB 的限制。此方法也可以改善效能和復原時間。如需更多詳細資訊，請參閱[Amazon RDS 中的 MySQL 檔案大小限制](#)。

非常大型的資料表 (大小大於 100 GB) 會對讀取和寫入 (包括 DML 陳述式，特別是 DDL 陳述式) 的效能造成負面影響。大型表格上的索引可以顯著改善選取效能，但它們也會降低 DML 陳述式的效

能。DDL 陳述式 (例如 ALTER TABLE) 對於大型資料表而言可能會顯著減慢，因為這些操作可能會在某些情況下完全重建資料表。這些 DDL 陳述式可能會在操作的持續時間鎖定資料表。

MySQL 讀取和寫入所需的記憶體數量取決於操作中涉及的資料表。這是一個最佳實務，至少有足夠的 RAM 來保留正在使用中的資料表的索引。若要尋找在資料庫中的十個最大資料表和索引，請使用下列查詢：

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
      ( data_length ) / 1024 / 1024 as dat,
      ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

表格數目

您的基礎檔案系統可能對代表資料表的檔案數量有所限制。不過，MySQL 對資料表的數量沒有限制。儘管這樣，MySQL InnoDB 儲存引擎中的資料表總數可能會導致性能降低 (無論這些資料表的大小如何)。若要限制作業系統的影響，您可以將資料表分割到同一個 MySQL 資料庫執行個體中的多個資料庫。這樣做可能會限制目錄中的檔案數量，但無法解決整體問題。

當因為大量的資料表 (超過 1 萬) 而導致效能下降時，它是由 MySQL 處理儲存文件引起的，包括開啟和關閉它們。若要解決這個問題，您可以增加 `table_open_cache` 和 `table_definition_cache` 參數的大小。但是，增加這些參數的值可能會顯著增加 MySQL 使用的記憶體數量，甚至可能會使用所有可用的記憶體。如需詳細資訊，請參閱 MySQL 文件中的 [MySQL 如何開啟及關閉資料表](#)。

此外，太多的資料表會顯著影響 MySQL 啟動時間。正常關機和重新啟動和損壞復原都會受到影響，尤其是在 MySQL 8.0 之前的版本中。

建議在資料庫執行個體中所有資料庫間的資料表總數少於 10,000 個資料表。對於 MySQL 資料庫中含有大量資料表的使用案例，請參閱 [MySQL 8.0 中的一百萬個資料表](#)。

儲存引擎

Amazon RDS for MySQL 的還 point-in-time 原和快照還原功能需要當機可復原的儲存引擎。僅 InnoDB 儲存引擎支援這些功能。雖然 MySQL 支援多種功能不盡相同的儲存引擎，但並非所有引擎的損毀復原能力和資料耐用性都經過最佳化設計。例如，MyISAM 儲存引擎不支援可靠的損毀復原，而

且可能會阻止 point-in-time 還原或快照還原如預期運作。這可能在損毀後，重新啟動 MySQL 時造成資料遺失或損毀。

InnoDB 為 Amazon RDS 上 MySQL 資料庫執行個體上建議和支援的儲存引擎。InnoDB 執行個體也可以遷移至 Aurora，而 MyISAM 執行個體則無法遷移。不過，如果您需要密集的全文搜尋功能，MyISAM 的效能優於 InnoDB。如果您仍選擇使用 MyISAM 搭配 Amazon RDS，則[利用不受支援的 MySQL 儲存引擎進行自動備份](#)中說明的步驟，對某些快照還原功能案例而言，相當實用。

如果要將現有的 MyISAM 資料表轉換為 InnoDB 資料表，可以使用 [MySQL 文件](#) 中說明的程序。MyISAM 和 InnoDB 各有優劣之處，以應用程式執行轉換作業之前，應先完整評估相關影響。

此外，Amazon RDS for MySQL 目前不支援聯合儲存引擎。

使用 Oracle 的最佳實務

資料表大小和 MariaDB 資料庫中的資料表數量可能會影響效能。

資料表大小

一般而言，作業系統對檔案大小的約束會決定 MariaDB 資料庫的有效資料表大小上限。所以，限制通常不是由內部 MariaDB 約束所決定的。

在 MariaDB 資料庫執行個體上，請避免讓資料庫中的資料表變得太大。雖然一般儲存體限制為 64 TiB，但佈建的儲存體限制會將 MariaDB 資料表檔案的大小上限限制在 16 TiB。請分割大型資料表，使得檔案大小不超過 16 TiB 的限制。此方法也可以改善效能和復原時間。

非常大型的資料表 (大小大於 100 GB) 會對讀取和寫入 (包括 DML 陳述式，特別是 DDL 陳述式) 的效能造成負面影響。大型表格上的索引可以顯著改善選取效能，但它們也會降低 DML 陳述式的效能。DDL 陳述式 (例如 ALTER TABLE) 對於大型資料表而言可能會顯著減慢，因為這些操作可能會在某些情況下完全重建資料表。這些 DDL 陳述式可能會在操作的持續時間鎖定資料表。

MariaDB 讀取和寫入所需的記憶體數量取決於操作中涉及的資料表。這是一個最佳實務，至少有足夠的 RAM 來保留正在使用中的資料表的索引。若要尋找在資料庫中的十個最大資料表和索引，請使用下列查詢：

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
        ( data_length ) / 1024 / 1024 as dat,
        ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
```



```
order by 3 desc
limit 10;
```

表格數目

您的基礎檔案系統可能對代表資料表的檔案數量有所限制。不過，MariaDB 對資料表的數量沒有限制。儘管這樣，MariaDB InnoDB 儲存引擎中的資料表總數可能會導致性能降低 (無論這些資料表的大小如何)。若要限制作業系統的影響，您可以將資料表分割到同一個 MariaDB 資料庫執行個體中的多個資料庫。這樣做可能會限制目錄中的檔案數量，但無法解決整體問題。

當由於大量的資料表 (超過 10,000) 而導致效能下降時，它是由使用儲存檔案的 MariaDB 所引起的。這項工作包括 MariaDB 開啟和關閉儲存檔案。若要解決這個問題，您可以增加 `table_open_cache` 和 `table_definition_cache` 參數的大小。不過，增加這些參數的值可能會顯著增加 MariaDB 使用的記憶體數量。它甚至可能使用所有可用的記憶體。如需詳細資訊，請參閱 MariaDB 文件中的[最佳化 table_open_cache](#)

此外，太多的資料表會顯著影響 MariaDB 啟動時間。正常關機和重新啟動和損壞復原都會受到影響。建議在資料庫執行個體中所有資料庫間的資料表總數少於一萬個資料表。

儲存引擎

適用於 MariaDB 的 Amazon RDS 的還 point-in-time 原和快照還原功能需要當機可復原的儲存引擎。雖然 MariaDB 支援多種功能不盡相同的儲存引擎，但並非所有引擎的損毀復原能力和資料耐用性都經過最佳化設計。例如，雖然 Aria 是 MyISAM 的當機安全替代品，但它仍然可能會阻止還 point-in-time 原或快照還原如預期運作。這可能在損毀後，重新啟動 MariaDB 時造成資料遺失或損毀。InnoDB 為 Amazon RDS 上 MariaDB 資料庫執行個體上建議和支援的儲存引擎。如果您仍選擇使用 Aria 搭配 Amazon RDS，則[利用不受支援的 MariaDB 儲存引擎進行自動備份](#)中說明的步驟，對某些快照還原功能案例而言，相當實用。

如果要將現有的 MyISAM 資料表轉換為 InnoDB 資料表，可以使用 [MariaDB 文件](#) 中說明的程序。MyISAM 和 InnoDB 各有優劣之處，以應用程式執行轉換作業之前，應先完整評估相關影響。

使用 Oracle 的最佳實務

如需 Amazon RDS for Oracle 適用的最佳實務的相關資訊，請參閱[在 Amazon Web Services 上執行 Oracle 資料庫的最佳實務](#)。

2020 年的 AWS 虛擬研討會包括在 Amazon RDS 上運行生產甲骨文數據庫的演示文稿。您可以在[此處](#)找到簡報的影片。

使用 PostgreSQL 的最佳實務

您可以改善 RDS for PostgreSQL 效能的兩個重要時機，一是將資料載入至資料庫執行個體時。另一個是使用 PostgreSQL 自動清空功能時。下列小節涵蓋我們對這些時機的一些實務建議。

如需有關 Amazon RDS 如何執行其他常見 PostgreSQL DBA 任務的資訊，請參閱[Amazon RDS for PostgreSQL 的常用 DBA 任務](#)。

將資料載入至 PostgreSQL 資料庫執行個體

將資料載入至 Amazon RDS for PostgreSQL 資料庫執行個體時，請修改資料庫執行個體設定和資料庫參數群組值。設定這些項目可讓您以最有效率的方式將資料匯入至資料庫執行個體。

將資料庫執行個體設定修改為下列：

- 停用資料庫執行個體備份 (將 `backup_retention` 設為 0)
- 停用多個可用區

修改資料庫參數群組來包含下列設定。此外，測試參數設定，為資料庫執行個體找出最有效的設定：

- 增加 `maintenance_work_mem` 參數的值。如需 PostgreSQL 資源耗用量參數的詳細資訊，請參閱[PostgreSQL 文件](#)。
- 增加 `max_wal_size` 和 `checkpoint_timeout` 參數的值，以減少對預先寫入 (WAL) 日誌的寫入次數。
- 停用 `synchronous_commit` 參數。
- 停用 PostgreSQL 自動清空參數。
- 確定已記錄任何您要匯入的資料表。存放在未記錄資料表的資料可能在容錯移轉期間遺失。如需詳細資訊，請參閱[CREATE TABLE UNLOGGED](#)。

使用 `pg_dump -Fc` (壓縮) 或 `pg_restore -j` (平行) 命令搭配這些設定。

載入作業完成後，將資料庫執行個體和資料庫參數回復為正常設定。

使用 PostgreSQL 自動清空功能

PostgreSQL 資料庫的自動清空功能為我們強烈建議您使用的功能，您可用來維護 PostgreSQL 資料庫執行個體的運作狀態。自動清空會自動執行 `VACUUM` 和 `ANALYZE` 命令；PostgreSQL 會要求使用

自動清空 (而非由 Amazon RDS 強加), 其使用對良好效能而言非常重要。依預設, 所有新的 Amazon RDS for PostgreSQL 資料庫執行個體都會啟用此功能, 而且依預設會適當設定相關的組態參數。

您的資料庫管理員需要知道並了解此維護操作。如需關於自動清空的 PostgreSQL 文件, 請參閱[自動清空協助程式](#)。

自動清空不是「不耗用資源」的操作, 但會在背景中運作並盡可能產生使用者操作。啟用時, 自動清空會檢查有大量更新或刪除元組的資料表。由於交易 ID 包圍, 它也會保護非常舊的資料, 避免遭到遺失。如需詳細資訊, 請參閱[避免交易 ID 包圍失敗](#)。

您不應將自動清空視為可以減少以獲得更好效能的高成本操作。相反地, 如果自動清空未執行, 更新和刪除速度快的資料表將隨著時間快速惡化。

Important

未執行自動清空最後可能會使得您必須停機, 以執行更為侵入式的清空操作。在某些情況下, RDS for PostgreSQL 資料庫執行個體可能會因為過度保守使用自動清空而變成無法使用。在這些情況下, PostgreSQL 資料庫會關閉以保護自己。此時, Amazon RDS 必須直接在資料庫執行個體上執行 single-user-mode 完整真空。這種完全清空可能會造成系統停機好幾個小時。因此, 強烈建議您不要關閉預設開啟的自動清空選項。

自動清空參數會判斷自動清空何時運作與運作強度。autovacuum_vacuum_threshold 和 autovacuum_vacuum_scale_factor 參數會判斷自動清空何時執行。autovacuum_max_workers、autovacuum_nap_time, autovacuum_cost_limit 和 autovacuum_cost_delay 參數會判斷自動清空的運作強度。如需有關自動清空、何時執行以及需要什麼參數的詳細資訊, 請參閱 PostgreSQL 文件中的[常式清空](#)。

以下的查詢顯示名為 table1 之資料表中「無效」元組的數量：

```
SELECT relname, n_dead_tup, last_vacuum, last_autovacuum FROM
pg_catalog.pg_stat_all_tables
WHERE n_dead_tup > 0 and relname = 'table1';
```

查詢的結果將類似下列：

```
relname | n_dead_tup | last_vacuum | last_autovacuum
-----+-----+-----+-----
tasks   |    81430522 |              |
(1 row)
```

Amazon RDS for PostgreSQL 最佳實務影片

2020 年的 AWS RE：發明會議包括一個簡報，介紹了在 Amazon RDS 上使用 PostgreSQL 的新功能和最佳實務。您可以在[此處](#)找到簡報的影片。

使用 SQL Server 的最佳實務

具有 SQL Server 資料庫執行個體之多個可用區部署的最佳實務包括下列：

- 使用 Amazon RDS 資料庫事件來監控容錯移轉。例如，在資料庫執行個體容錯移轉時，您可以透過文字訊息或電子郵件收到通知。如需 Amazon RDS 事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。
- 如果您的應用程式會快取 DNS 值，請將存留時間 (TTL) 設定為少於 30 秒。在發生容錯移轉時，以此方式設定 TTL 是很好的做法。在容錯移轉中，IP 地址可能變更且快取值可能不再提供服務。
- 建議您不要啟用下列模式，因為它們會關閉交易日誌，而這是多個可用區所需：
 - 簡單復原模式
 - 離線模式
 - 唯讀模式
- 測試以判斷您的資料庫執行個體進行容錯移轉所需的時間。容錯移轉的時間可能因資料庫的類型、執行個體類別和您使用的儲存類型而異。您也應該測試應用程式在發生容錯移轉時繼續運作的能力。
- 若要縮短容錯移轉時間，請執行下列動作：
 - 確保已為您的工作負載配置足夠的佈建 IOPS。不足的 I/O 可能加長容錯移轉時間。資料庫復原需要 I/O。
 - 使用較小的交易。資料庫復原仰賴於交易，因此，如果您可以將大型交易分成多個較小的交易，您的容錯移轉時間應該會縮短。
- 請注意，在容錯移轉期間，延遲可能延長。在容錯移轉程序中，Amazon RDS 會將您的資料自動複製至新的待命執行個體。此複製表示新資料正在遞交至兩個不同的資料庫執行個體。因此，直到備用資料庫執行個體追上新的主要資料庫執行個體之前，可能會有一些延遲。
- 在所有可用性區域中部署您的應用程式。如果可用性區域發生故障，您在其他可用性區域的應用程式仍將可用。

使用 SQL Server 的多個可用區部署時，請記得，Amazon RDS 會為您的執行個體上的所有 SQL Server 資料庫建立複本。如果您不想要讓特定資料庫有次要複本，請為那些資料庫設定不使用多個可用區的個別資料庫執行個體。

Amazon RDS for SQL Server 最佳實務影片

2019 年的 AWS RE：發明會議包括在 Amazon RDS 上使用 SQL 服務器的新功能和最佳實踐的演示文稿。您可以在[此處](#)找到簡報的影片。

使用資料庫參數群組

建議您在測試資料庫執行個體上嘗試進行資料庫參數群組變更，再將參數群組變更套用至生產資料庫執行個體。未正確設定資料庫參數群組中的資料庫引擎參數，可能產生各種意外影響，包括降低效能和系統不穩定。修改資料庫引擎參數時請務必謹慎，在修改資料庫參數群組之前，請備份您的資料庫執行個體。

如需備份資料庫執行個體的詳細資訊，請參閱[備份、還原和匯出資料](#)。

自動建立資料庫執行個體的最佳實務

使用偏好的資料庫引擎次要版本建立資料庫執行個體的 Amazon RDS 最佳實務。您可以使用 Amazon RDS API AWS CLI，或 AWS CloudFormation 將資料庫執行個體建立自動化。使用這些方法時，您只能指定主要版本，因為 Amazon RDS 會使用偏好的次要版本自動建立執行個體。例如，如果 PostgreSQL 12.5 是偏好的次要版本，而且如果您使用 `create-db-instance` 指定 12 版，則資料庫執行個體將為 12.5 版。

若要決定偏好的次要版本，您可以使用以下範例所示 `describe-db-engine-versions` 的選項來執行命令 `--default-only`。

```
aws rds describe-db-engine-versions --default-only --engine postgres

{
  "DBEngineVersions": [
    {
      "Engine": "postgres",
      "EngineVersion": "12.5",
      "DBParameterGroupFamily": "postgres12",
      "DBEngineDescription": "PostgreSQL",
      "DBEngineVersionDescription": "PostgreSQL 12.5-R1",
      ...some output truncated...
    }
  ]
}
```

如需以程式設計方式建立資料庫執行個體的資訊，請參閱以下資源：

- 使用 AWS CLI — [create-db-instance](#)
- 使用 Amazon RDS API — [CreateDBInstance](#)
- 使用 AWS CloudFormation — [AWS::RDS::資料庫執行個體](#)

Amazon RDS 新功能視頻

2023 AWS RE：發明會議包括有關 Amazon RDS 新功能的演示文稿。您可以在[此處](#)找到簡報的影片。

設定 Amazon RDS 資料庫執行個體

本節說明如何設定您的 Amazon RDS 資料庫執行個體。建立一個資料庫執行個體之前，決定將執行資料庫執行個體的資料庫執行個體類別。此外，請選擇 AWS 區域來決定資料庫執行個體的執行位置。接下來，建立資料庫執行個體。

您可以使用選項群組和資料庫參數群組來設定資料庫執行個體。

- 選項群組指定稱作選項的功能，可用於特定的 Amazon RDS 資料庫執行個體。
- 資料庫參數群組扮演引擎組態值的容器，以套用至一或多個資料庫執行個體。

選項和參數可用與否取決於資料庫引擎和資料庫引擎版本。您可以在建立資料庫執行個體時，指定選項群組和資料庫參數群組。您也可以修改資料庫執行個體來指定它們。

主題

- [建立 Amazon RDS 資料庫執行個體](#)
- [透過 AWS CloudFormation 建立 Amazon RDS 資源](#)
- [連接至 Amazon RDS 資料庫執行個體](#)
- [使用選項群組](#)
- [使用參數群組](#)
- [使用 Amazon RDS 資料庫執行個體設定建立 Amazon ElastiCache 快取](#)

建立 Amazon RDS 資料庫執行個體

基本建置區塊 Amazon RDS 是您在其中建立資料庫的資料庫執行個體。建立資料庫執行個體時，您要選擇它的引擎專屬特性。您也可以選擇執行資料庫伺服器之 AWS 執行個體的儲存容量、CPU、記憶體等。

主題

- [執行個體先決條件](#)
- [建立資料庫執行個體](#)
- [資料庫執行個體的設定](#)

執行個體先決條件

Important

在您可以建立 Amazon RDS 資料庫執行個體之前，請先完成 [設定 Amazon RDS](#) 中的任務。

下列是建立 RDS 資料庫執行個體的先決條件。

主題

- [設定資料庫執行個體的網路](#)
- [其他先決條件](#)

設定資料庫執行個體的網路

您只能在以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中建立 Amazon RDS 資料庫執行個體。此外，它必須位於至少具 AWS 區域 有兩個可用區域的。您為資料庫執行個體選擇的資料庫子網路群組必須包含至少兩個可用區域。此組態可確保您在建立資料庫執行個體時設定多可用區部署，或在未來輕鬆移至其中一個可用區域。

若要在同一 VPC 中設定新資料庫執行個體與 Amazon EC2 執行個體之間的連線，則可在建立資料庫執行個體時這麼做。若要在同一 VPC 中從 EC2 執行個體以外的資源連線到資料庫執行個體，請手動設定網路連線。

主題

- [設定與 EC2 執行個體的自動網路連線](#)
- [手動設定網路](#)

設定與 EC2 執行個體的自動網路連線

建立 RDS 資料庫執行個體時，可以使用設 AWS Management Console 定 EC2 執行個體和新資料庫執行個體之間的連線。當您這樣做時，RDS 會自動設定您的 VPC 和網路設定。資料庫執行個體建立在與 EC2 執行個體相同的 VPC 中，以便 EC2 執行個體可以存取資料庫執行個體。

以下是將 EC2 執行個體與資料庫執行個體連線的要求：

- EC2 執行個體必須存在於中，AWS 區域 才能建立資料庫執行個體。

如果中沒有 EC2 執行個體 AWS 區域，則主控台會提供建立 EC2 執行個體的連結。

- 建立資料庫執行個體的使用者必須擁有執行下列操作的許可：

- `ec2:AssociateRouteTable`
- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateRouteTable`
- `ec2:CreateSubnet`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

使用此選項可建立私有資料庫執行個體。資料庫執行個體使用只包含私有子網路的資料庫子網路群組來限制對 VPC 內資源的存取。

若要將 EC2 執行個體連線到資料庫執行個體，請選擇 Create database (建立資料庫) 頁面上 **Connectivity (連線能力)** 區段的 **Connect to an EC2 compute resource (連線至 EC2 運算資源)**。

Connectivity Info
↻

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 Instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

Choose EC2 instances
▼

當您選擇 **Connect to an EC2 compute resource (連線至 EC2 運算資源)** 時，RDS 會自動設定下列選項。除非您透過選擇 **Don't connect to an EC2 compute resource (不要連線至 EC2 運算資源)**，選擇不設定與 EC2 執行個體的連線，否則無法變更設定。

主控台選項	自動設定
Network type (網路類型)	RDS 將網路類型設定為 IPv4。目前，當您在 EC2 執行個體和資料庫執行個體之間設定連線時，不支援雙堆疊模式。
Virtual Private Cloud (VPC)	RDS 會將 VPC 設定為與 EC2 執行個體關聯的 VPC。
DB subnet group (資料庫子網路群組)	<p>RDS 需要在相同可用區域中具有私有子網路的資料庫子網路群組作為 EC2 執行個體。如果存在符合此要求的資料庫子網路群組，則 RDS 會使用現有的資料庫子網路群組。依預設，此選項會設為 Automatic setup (自動設定)。</p> <p>當您選擇 Automatic setup (自動設定)，且沒有符合此需求的資料庫子網路群組時，則會發生下列動作。RDS 在三個可用區域中使用三個可用的私有子網路，其中一個可用區域與 EC2 執行個體相同。如果可用區域中無法使用私有子網路，RDS 會在可用區域中建立私有子網路。RDS 接著建立資料庫子網路群組。</p>

主控台選項	<h3>自動設定</h3> <p>當私有子網路可用時，RDS 會使用與該子網路相關聯的路由表，並將其建立的任何子網路新增至此路由表。當沒有可用的私有子網路時，RDS 會建立沒有網際網路閘道存取權的路由表，並將其建立的子網路新增至路由表。</p> <p>RDS 也可讓您使用現有的資料庫子網路群組。如果想要使用您選擇的現有資料庫子網路群組，請選取 Choose existing (選擇現有的)。</p>
公用存取	<p>RDS 選擇 No (否)，使資料庫執行個體無法公開存取。</p> <p>為了安全起見，最佳實務是保持資料庫為私有，並確保無法從網際網路存取該資料庫。</p>
VPC security group (firewall) (VPC 安全群組 (防火牆))	<p>RDS 會建立與資料庫執行個體關聯的新安全群組。安全群組命名為 <code>rds-ec2-<i>n</i></code>，其中 <i>n</i> 為數字。此安全群組包含以 EC2 VPC 安全群組 (防火牆) 做為來源的傳入規則。與資料庫執行個體關聯的安全群組可讓 EC2 執行個體存取資料庫執行個體。</p> <p>RDS 也會建立與 EC2 執行個體關聯的新安全群組。安全群組命名為 <code>ec2-rds-<i>n</i></code>，其中 <i>n</i> 為數字。此安全群組包含傳出規則，並將資料庫執行個體的 VPC 安全群組做為來源。此安全群組允許 EC2 執行個體將流量傳送到資料庫執行個體。</p> <p>您可以選擇 Create New (建立新的)，並輸入新安全群組的名稱，以新增其他的新安全群組。</p> <p>您可以選擇 Choose existing (選擇現有)，然後選取要新增的安全群組，以新增現有的安全群組。</p>

主控台選項	自動設定
可用區域	<p>當您在 Availability & durability (可用性和耐用性) (單一 AZ 部署) 選擇 Single DB instance (單一執行個體) 時，RDS 會選擇 EC2 執行個體的可用區域。</p> <p>當您在 Availability & durability (可用性和耐用性) 中選擇執行 (多可用區域資料庫執行個體部署) 時，RDS 會為該部署中的一個資料庫執行個體選擇 EC2 執行個體的可用區域。RDS 會隨機為其他資料庫執行個體選擇不同的可用區域。主要資料庫執行個體或待命複本會在與 EC2 執行個體相同的可用區域中建立。當您選擇 Multi-AZ DB instance (多可用區域資料庫執行個體) 時，如果資料庫執行個體和 EC2 執行個體位於不同的可用區域，可能會產生跨可用區域成本。</p>

如需這些設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

如果您在建立資料庫執行個體後變更這些設定，這些變更可能會影響 EC2 執行個體與資料庫執行個體之間的連線。

手動設定網路

若要在同一 VPC 中從 EC2 執行個體以外的資源連線到資料庫執行個體，請手動設定網路連線。如果您使 AWS Management Console 用建立資料庫執行個體，您可以讓 Amazon RDS 自動為您建立 VPC。或者，您可以為資料庫執行個體使用現有的 VPC 或建立新的 VPC。使用任何方法，您的 VPC 都會在至少兩個可用區域的每一個中要求至少有一個子網路，可與 RDS 資料庫執行個體搭配使用。

Amazon RDS 預設會自動為您建立資料庫執行個體的可用區域。若要選擇特定的可用區域，您必須將 Availability & durability (可用性和耐用性) 設定變更為單一資料庫執行個體。這樣做會暴露一個可讓您從 VPC 的可用區域中選擇的 Availability Zone (可用區域) 設定。然而，如果您選擇多可用區部署，RDS 會自動選擇主要或寫入器資料庫執行個體的可用區域，且 Availability Zone (可用區域) 設定不會出現。

在某些情況下，您可能沒有預設 VPC 或尚未建立 VPC。在這些情況下，您可讓 Amazon RDS 在使用主控台建立資料庫執行個體時自動為您建立 VPC。若否，請執行下列操作：

- 在您要部署資料庫執行個體的至少兩個可用區域中，每個子網路 AWS 區域 中至少有一個子網路建立 VPC。如需詳細資訊，請參閱 [在 VPC 中使用資料庫執行個體](#) 及 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。
- 指定可授權連線到資料庫執行個體的 VPC 安全群組。如需詳細資訊，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#) 及 [使用安全群組控制存取](#)。
- 指定 RDS DB 子網路群組，該子網路群組會在資料庫執行個體使用的 VPC 中至少定義兩個子網路。如需詳細資訊，請參閱 [使用資料庫子網路群組](#)。

如果您想要連線至與資料庫執行個體不在相同 VPC 中的資源，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 中的適當案例。

其他先決條件

建立資料庫執行個體之前，請考慮下列其他先決條件：

- 如果您要連線到 AWS 使用 AWS Identity and Access Management (IAM) 登入資料，您的 AWS 帳戶必須具有特定的 IAM 政策。這些會授與執行 Amazon RDS 操作所需的許可。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。

若要使用 IAM 存取 RDS 主控台，請使用您的 IAM 使 AWS Management Console 用者登入資料登入。接著前往 Amazon RDS 主控台：<https://console.aws.amazon.com/rds/>。

- 若要為您的資料庫執行個體量身打造組態參數，請使用必要的參數設定來指定資料庫參數群組。如需建立或修改資料庫參數群組的資訊，請參閱 [使用參數群組](#)。

Important

如果您使用適用於 Db2 的 Amazon RDS BYOL 模型，則在建立資料庫執行個體之前，必須先建立包含您的和的自訂參數群組。IBM Site ID IBM Customer ID如需詳細資訊，請參閱 [為 Db2 帶來您自己的授權](#)。

- 決定要為資料庫執行個體指定的 TCP/IP 連接埠號碼。某些公司的防火牆會封鎖 RDS 資料庫執行個體預設連接埠的連線。如果您公司的防火牆會封鎖預設連接埠，請為資料庫執行個體選擇另一個連接埠。Amazon RDS 資料庫引擎的預設連接埠為：

適用於 Db2 的 RDS	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
50000	3306	3306	1521	5432	1433

對於 SQL Server 的 RDS，會保留下列連接埠，而且您無法在建立資料庫執行個體時使用它們：1234，1434，3260，3343，3389，47001，和49152-49156。

建立資料庫執行個體

您可以使用 AWS Management Console、或 RDS API 建立 Amazon RDS 資料庫執行個體。

Note

對於 RDS for Db2，我們建議您先設定授權模式所需的項目，然後再建立 RDS for Db2 資料庫執行個體。如需詳細資訊，請參閱 [Amazon 適用於 Db2 的授權選項](#)。

主控台

您可以使用啟用或未啟用「AWS Management Console 輕鬆建立」來建立資料庫執行個體。在 Easy create (輕鬆建立) 啟用的情形下，您僅指定資料庫引擎類型、資料庫執行個體大小與資料庫執行個體識別符。Easy create (輕鬆建立) 會將預設設定用於其他設定選項。在 Easy create (輕鬆建立) 未啟用的情形下，您會在建立資料庫時指定更多設定選項，包含可用性、安全性、備份和維護的選項。

Note

於下列程序中，Standard Create (標準建立) 已加以啟用，而 Easy Create (輕鬆建立) 尚未啟用。此程序使用 Microsoft SQL Server 作為範例。

如需使用 Easy Create (輕鬆建立) 引導您逐步建立並連線至每個引擎的範例資料庫執行個體範例，請參閱 [Amazon RDS 入門](#)。


建立資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要建立資料庫執行個體的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇建立資料庫，然後選擇 標準建立。
5. 針對引擎類型，請選擇 IBM Db2、MariaDB、Microsoft SQL 伺服器、MySQL、甲骨文或 PostgreSQL。

Microsoft SQL Server 顯示於此。

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input checked="" type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

Database management type [Info](#)

- Amazon RDS**
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.
- Amazon RDS Custom**
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.

Edition

- SQL Server Express Edition**
Affordable database management system that supports database sizes up to 10 GB.
- SQL Server Web Edition**
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
- SQL Server Standard Edition**
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
- SQL Server Enterprise Edition**
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

License

license-included

Engine Version

SQL Server 2022 16.00.4085.2.v1
▼


6. 針對資料庫管理類型，若您使用 Oracle 或 SQL Server，請選擇 Amazon RDS 或 Amazon RDS Custom。

Amazon RDS 顯示於此。如需 RDS Custom 的詳細資訊，請參閱 [使用 Amazon RDS Custom](#)。

7. 對於版本，如果您使用的是 Db2、甲骨文或 SQL Server，請選擇您要使用的資料庫引擎版本。
MySQL 只有一個選項的版本，MariaDB 和 PostgreSQL 則沒有。
8. 在 Version (版本) 中，選擇引擎版本。
9. 在 Templates (範本) 中，選擇符合您使用案例的範本。如果您選擇 Production (生產)，在後續步驟中會預先選取下列項目：

- 多可用區域容錯移轉選項
- Provisioned IOPS SSD (io1) (佈建 IOPS SSD (io1)) 儲存體選項
- 啟用刪除保護選項

建議針對任何生產環境使用這些功能。

 Note

範本選項會依版本而有所不同。

10. 若要輸入您的主要密碼，請執行以下動作：
 - a. 在 Settings (設定) 區段，開啟 Credential Settings (登入資料設定)。
 - b. 如果您想要指定密碼，請清除 Auto generate a password (自動產生密碼) 核取方塊 (如果已選取)。
 - c. (選用) 變更 Master username (主要使用者名稱) 值。
 - d. 在 Master password (主要密碼) 和 Confirm password (確認密碼) 中輸入相同的密碼。
11. (選擇性) 為此資料庫執行個體設定與運算資源的連線。

您可以在建立資料庫執行個體期間設定 Amazon EC2 執行個體與新資料庫執行個體之間的連線。如需詳細資訊，請參閱 [設定與 EC2 執行個體的自動網路連線](#)。

12. 在 VPC 安全群組 (防火牆) 下的連線能力區段中，如果您選取建立新的，則系統會建立 VPC 安全群組並提供傳入規則，允許本機電腦的 IP 地址存取資料庫。
13. 在其餘區段，指定資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

14. 選擇 Create database (建立資料庫)。

如果您選擇使用自動產生的密碼，View credential details (檢視登入資料詳細資訊) 按鈕會出現在 Databases (資料庫) 頁面。

若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

若要以主要使用者的身分連線至資料庫執行個體，請使用出現的使用者名稱和密碼。

⚠ Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。如果您需要在資料庫執行個體可供使用後變更主要使用者密碼，則可將資料庫執行個體修改為這麼做。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

15. 在 Databases (資料庫) 中，選擇新資料庫執行個體的名稱。

在 RDS 主控台上，新資料庫執行個體的詳細資訊即會出現。在資料庫執行個體建立完成且可供使用之前，資料庫執行個體會處於 Creating (建立中) 狀態。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和分配的儲存空間而定，新的執行個體可能需要幾分鐘才能使用。

database-1				Modify	Actions ▾
Summary					
DB identifier database-1	CPU	Info 🕒 Creating	Class db.t2.micro		
Role Instance	Current activity	Engine SQL Server Express Edition	Region & AZ -		
Connectivity & security Monitoring Logs & events Configuration Maintenance & backups Tags					

AWS CLI

Note

如果您想要透過使用 Db2 授權 AWS Marketplace，您必須先使用 AWS Management Console。AWS Marketplace 如需詳細資訊，請參閱 [訂閱 Db2 Marketplace 清單並註冊 IBM](#)。

若要使用建立資料庫執行個體 AWS CLI，請使用下列參數呼叫 [create-db-instance](#) 命令：

- `--db-instance-identifier`
- `--db-instance-class`
- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

此範例會使用 Microsoft SQL Server。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --engine sqlserver-se \  
  --db-instance-identifier mysftssqlserver \  
  --allocated-storage 250 \  
  --db-instance-class db.t3.large \  
  --vpc-security-group-ids mysecuritygroup \  
  --db-subnet-group mydbsubnetgroup \  
  --master-username masterawsuser \  
  --manage-master-user-password \  
  --backup-retention-period 3
```

在 Windows 中：

```
aws rds create-db-instance ^
  --engine sqlserver-se ^
  --db-instance-identifier mydbinstance ^
  --allocated-storage 250 ^
  --db-instance-class db.t3.large ^
  --vpc-security-group-ids mysecuritygroup ^
  --db-subnet-group mydbsubnetgroup ^
  --master-username masterawsuser ^
  --manage-master-user-password ^
  --backup-retention-period 3
```

此命令會產生類似下列的輸出。

```
DBINSTANCE mydbinstance db.t3.large sqlserver-se 250 sa creating 3 **** n
10.50.2789
SECGROUP default active
PARAMGRP default.sqlserver-se-14 in-sync
```

RDS API

Note

如果您想要透過使用 Db2 授權 AWS Marketplace，您必須先使用 AWS Management Console。AWS Marketplace 如需詳細資訊，請參閱 [訂閱 Db2 Marketplace 清單並註冊 IBM](#)。

若要使用 Amazon RDS API 來建立資料庫執行個體，請呼叫 [CreateDBInstance](#) 操作。


如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

資料庫執行個體的設定

在下表中，您可以找到在建立資料庫執行個體時選擇之設定的詳細資訊。此表格也會顯示支援每個設定的資料庫引擎。

您可以使用主控台、[create-db-instance](#) CLI 命令，或 [CreateDBInstance](#) RDS API 操作，建立新的資料庫執行個體。

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
配置儲存	<p>要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。在部分情況下，配置給資料庫執行個體的儲存空間容量若高於資料庫的大小，可改善輸入/輸出效能。</p> <p>如需更多詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體。</p>	<p>CLI 選項：</p> <pre>--allocated-storage</pre> <p>API 參數：</p> <pre>AllocatedStorage</pre>	全部
架構設定	<p>如果您選擇 Oracle 多租戶架構，則 RDS for Oracle 會建立容器資料庫 (CDB)。如果未選擇此選項，RDS for Oracle 會建立一個非 CDB。非 CDB 使用傳統的 Oracle 資料庫架構。CDB 可以包含插入式資料庫 (PDB)，而非 CDB 則不能。</p> <p>Oracle Database 21c 僅使用 CDB 架構。Oracle Database 19c 可以使用 CDB 或非 CDB 架構。低於 Oracle Database 19c 的版本僅使用非 CDB 架構。</p> <p>如需詳細資訊，請參閱 RDS for Oracle CDB 概觀。</p>	<p>CLI 選項：</p> <pre>--engine oracle-ee</pre> <pre>-cdb (Oracle 多租戶)</pre> <pre>--engine oracle-se</pre> <pre>2-cdb (Oracle 多租戶)</pre> <pre>--engine oracle-ee</pre> <pre>(傳統型)</pre> <pre>--engine oracle-se</pre> <pre>2 (傳統型)</pre> <p>API 參數：</p> <pre>Engine</pre>	Oracle
架構組態	<p>只有當您針對架構設定選擇 Oracle 多租戶架構時，這些設定才有效。請選擇下列其中一個額外設定：</p> <ul style="list-style-type: none"> 使用多租戶組態時，您的 RDS for Oracle CDB 執行個體可以包含 1—30 個租用戶資料庫，具體取決於資料庫版本和任何必要的選項授權。在 Oracle 	<p>CLI 選項：</p> <pre>--multi-tenant (多租</pre> <pre>戶組態)</pre> <pre>--no-multi-tenant</pre> <pre>(單一租戶組態)</pre> <p>API 參數：</p> <pre>MultiTenant</pre>	Oracle

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
	<p>資料庫的內容中，租戶資料庫是 PDB。不支援應用程式 PDB 和 Proxy PDB。</p> <p>您的資料庫執行個體是搭配 1 個初始租戶資料庫建立的。選擇租戶資料庫名稱、租戶資料庫主要使用者名稱、租戶資料庫主要密碼和租戶資料庫字元集的值</p> <p>多租戶組態是永久的。因此，您無法將多租戶組態轉換回單一租戶組態。多租戶組態的最低支援版本更新 (RU) 是 19.0.0.0.ru-2022-01.rur-2022.r1。</p> <div data-bbox="365 877 922 1381" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Amazon RDS 功能稱為「multi-tenant」而不是「multi-tenant」，因為它是 RDS 平台的一項功能，而不只是 Oracle 資料庫引擎而已。「Oracle 多租戶」一詞僅指與內部部署和 RDS 部署相容的 Oracle 資料庫架構。</p></div> <ul style="list-style-type: none">• 使用單一租戶組態時，您的 RDS for Oracle CDB 包含 1 個 PDB。這是您建立 CDB 時的預設組態。您無法刪除初始 PDB 或新增更多的 PDB。您可以稍後將 CDB 的單一租戶組態轉換為多租戶組態，但之後您無法轉換回單一租戶組態。		

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
	<p>無論您選擇哪種組態，CDB 都會包含單一初始 PDB。在多租戶組態中，您可以稍後使用 RDS API 建立更多的 PDB。</p> <p>如需詳細資訊，請參閱 RDS for Oracle CDB 概觀。</p>		
Auto minor version upgrade (自動次要版本升級)	<p>選擇 [啟用 auto 動次要版本升級]，讓您的資料庫執行個體在推出時自動接收偏好的次要資料庫引擎版本升級。這是預設行為。Amazon RDS 會在維護時段執行自動次要版本升級。如果您未選擇「啟用 auto 動次要版本升級」，則當新的次要版本可用時，資料庫執行個體不會自動升級。</p> <p>如需詳細資訊，請參閱 自動升級次要引擎版本。</p>	<p>CLI 選項：</p> <pre>--auto-minor-version-upgrade</pre> <pre>--no-auto-minor-version-upgrade</pre> <p>API 參數：</p> <pre>AutoMinorVersionUpgrade</pre>	全部
Availability zone (可用區域)	<p>資料庫執行個體的可用區域。除非您想要指定可用區域，否則使用預設值 No Preference (無偏好設定)。</p> <p>如需更多詳細資訊，請參閱 區域、可用區域和 Local Zones。</p>	<p>CLI 選項：</p> <pre>--availability-zone</pre> <p>API 參數：</p> <pre>AvailabilityZone</pre>	全部
AWS KMS key	<p>只有在 Encryption (加密) 設為 Enable encryption (啟用加密) 時才能使用。選擇要用於加密此資料庫執行個體的 AWS KMS key。如需詳細資訊，請參閱 加密 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <pre>--kms-key-id</pre> <p>API 參數：</p> <pre>KmsKeyId</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
備份複寫	<p>選擇 Enable replication in another AWS Region (在另一個 AWS 區域中啟用複寫)，藉此在其他區域建立備份以進行災難復原。</p> <p>然後選擇其他備份的 Destination Region (目的地區域)。</p>	無法在建立資料庫執行個體時使用。如需使用 AWS CLI 或 RDS API 啟用跨區域備份的相關資訊，請參閱 啟用跨區域自動備份 。	Oracle PostgreSQL SQL Server
Backup retention period (備份保留期間)	<p>您想要資料庫執行個體之自動備份保留的天數。針對任何重要的資料庫執行個體，請將此值設為 1 或更大。</p> <p>如需更多詳細資訊，請參閱 備份簡介。</p>	<p>CLI 選項：</p> <pre>--backup-retention-period</pre> <p>API 參數：</p> <pre>BackupRetentionPeriod</pre>	全部
備份目標	<p>選擇AWS 雲端將自動備份和手動快照儲存在父 AWS 區域中。選擇 Outposts (on-premises) (Outposts (內部部署))，將它們本機存放在您的 Outpost 上。</p> <p>此選項設定只會套用至 RDS on Outposts。如需詳細資訊，請參閱 建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--backup-target</pre> <p>API 參數：</p> <pre>BackupTarget</pre>	MySQL、PostgreSQL、SQL Server
備份時段	<p>Amazon RDS 自動備份資料庫執行個體的時段。除非要在特定時段備份資料庫，否則請使用預設值 No Preference (無偏好設定)。</p> <p>如需詳細資訊，請參閱 備份簡介。</p>	<p>CLI 選項：</p> <pre>--preferred-backup-window</pre> <p>API 參數：</p> <pre>PreferredBackupWindow</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
憑證授權單位	<p>資料庫執行個體所使用之伺服器憑證的憑證授權單位 (CA)。</p> <p>如需詳細資訊，請參閱。</p>	<p>CLI 選項：</p> <pre>--ca-certificate-identifier</pre> <p>RDS API 參數：</p> <pre>CACertificateIdentifier</pre>	全部
字元集	<p>資料庫執行個體的字元集。資料庫字元集的預設值 AL32UTF8 適用於 Unicode 5.0 UTF-8 通用字元集。建立資料庫執行個體之後，您無法變更資料庫字元集。</p> <p>在單一租用戶組態中，非預設的資料庫字元集只會影響 PDB，而不會影響 CDB。如需更多詳細資訊，請參閱 CDB 架構的單一租用戶組態。</p> <p>資料庫字元集與國家字元集不同，後者稱為 NCHAR 字元集。與資料庫字元集不同，NCHAR 字元集會指定 NCHAR 資料類型 (NCHAR、NVARCHAR2 和 NCLOB) 欄的編碼，而不會影響資料庫中繼資料。</p> <p>如需更多詳細資訊，請參閱 RDS for Oracle 字元集。</p>	<p>CLI 選項：</p> <pre>--character-set-name</pre> <p>API 參數：</p> <pre>CharacterSetName</pre>	Oracle

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
定序	<p>資料庫執行個體適用的伺服器層級定序。</p> <p>如需更多詳細資訊，請參閱 Microsoft SQL Server 的伺服器層級定序。</p>	<p>CLI 選項：</p> <p><code>--character-set-name</code></p> <p>API 參數：</p> <p><code>CharacterSetName</code></p>	SQL Server
Copy tags to snapshots (將標籤複製到快照)	<p>此選項可在您建立快照時將任何資料庫執行個體標籤複製到某個資料庫快照。</p> <p>如需更多詳細資訊，請參閱 標記 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <p><code>--copy-tags-to-snapshot</code></p> <p><code>--no-copy-tags-to-snapshot</code></p> <p>RDS API 參數：</p> <p><code>CopyTagsToSnapshot</code></p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
<p>Database authentication (資料庫身分驗證)</p>	<p>您想要使用的資料庫身分驗證選項。</p> <p>選擇 Password authentication (密碼身分驗證)，僅使用資料庫密碼驗證資料庫使用者。</p> <p>選擇 Password and IAM DB authentication (密碼和 IAM 資料庫身分驗證)，透過使用者和角色搭配資料庫密碼和使用憑證來驗證資料庫使用者。如需詳細資訊，請參閱 適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證。此選項僅支持 MySQL 及 PostgreSQL。</p> <p>選擇密碼和 Kerberos 驗證，透過建立的資料庫密碼和 Kerberos 驗證來驗證資料庫使用者。AWS Managed Microsoft AD AWS Directory Service 接下來，選擇目錄或選擇 Create a new Directory (建立新目錄)。</p> <p>如需詳細資訊，請參閱下列其中一個項目：</p> <ul style="list-style-type: none"> • 使用適用於 Db2 的 Amazon RDS 的 Kerberos 身份驗證 • 針對 MySQL 使用 Kerberos 身分驗證 • 為 Amazon RDS for Oracle 設定 Kerberos 身分驗證 • 搭配 Amazon RDS for PostgreSQL 使用 Kerberos 身分驗證 	<p>IAM :</p> <p>CLI 選項 :</p> <pre>--enable-iam-database-authentication</pre> <pre>--no-enable-iam-database-authentication</pre> <p>RDS API 參數 :</p> <pre>EnableIAMDatabaseAuthentication</pre> <p>Kerberos :</p> <p>CLI 選項 :</p> <pre>--domain</pre> <pre>--domain-iam-role-name</pre> <p>RDS API 參數 :</p> <pre>Domain</pre> <pre>DomainIAMRoleName</pre>	<p>依身分驗證類型而有所不同</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
資料庫管理類型	<p>如果您不需要自訂您的環境，則選擇 Amazon RDS。</p> <p>如果您想要自訂資料庫、作業系統和基礎設施，則選擇 Amazon RDS Custom。如需詳細資訊，請參閱 使用 Amazon RDS Custom。</p>	對於 CLI 和 API，您可以指定資料庫引擎類型。	Oracle SQL Server
Database port (資料庫連線埠)	<p>您想要透過其存取資料庫執行個體的連接埠。會顯示預設連接埠。</p> <div data-bbox="332 747 922 1163" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>某些公司的防火牆會封鎖與預設 MariaDB、MySQL 及 PostgreSQL 連接埠的連線。如果您的公司防火牆會封鎖預設連接埠，請為您的資料庫執行個體輸入另一個連接埠。</p> </div>	<p>CLI 選項：</p> <p><code>--port</code></p> <p>RDS API 參數：</p> <p>Port</p>	全部
DB engine version (資料庫引擎版本)	您要使用的資料庫引擎版本。	<p>CLI 選項：</p> <p><code>--engine-version</code></p> <p>RDS API 參數：</p> <p>EngineVersion</p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
DB instance class (資料庫執行個體類別)	<p>資料庫執行個體的組態。例如，db.t3.small 資料庫執行個體類別具有 2 GiB 記憶體、2 個 vCPU、1 個虛擬核心、一個可變 ECU，以及一個中等輸入/輸出容量。</p> <p>可能的話，選擇足夠大的資料庫執行個體類別，讓一般查詢工作集可以保留在記憶體中。當工作集保留在記憶體中時，系統可以避免寫入至磁碟，因而可改善效能。如需詳細資訊，請參閱 資料庫執行個體類別。</p> <p>在 RDS for Oracle 中，您可以選取 Include additional memory configurations (包含額外的記憶體組態)。這些組態經過最佳化，可達到高記憶體與 vCPU 的比例。例如：db.r5.6xlarge.tpc2.mem4x 是一個 db.r5.8x 執行個體，具有每核心 2 個執行緒 (tpc2) 和標準 db.r5.6xlarge 資料庫執行個體的 4 倍記憶體。如需詳細資訊，請參閱 RDS for Oracle 執行個體類別。</p>	<p>CLI 選項：</p> <pre>--db-instance-class</pre> <p>RDS API 參數：</p> <p>DBInstanceClass</p>	全部
DB instance identifier (資料庫執行個體識別符)：	<p>資料庫執行個體的名稱。以您命名現場部署伺服器的相同方式，命名您的資料庫執行個體。您的資料庫執行個體識別碼最多可包含 63 個英數字元，而且在所選 AWS 區域中的帳戶必須是唯一的。</p>	<p>CLI 選項：</p> <pre>--db-instance-identifier</pre> <p>RDS API 參數：</p> <p>DBInstanceIdentifier</p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
DB parameter group (資料庫參數群組)	<p>資料庫執行個體的參數群組。您可以選擇預設參數群組，或可以建立自訂參數群組。</p> <p>如果您使用 RDS for Db2 的 BYOL 模型，則在建立資料庫執行個體之前，必須先建立包含和的自訂參數群組。IBM Site ID IBM Customer ID如需詳細資訊，請參閱 為 Db2 帶來您自己的授權。</p> <p>如需詳細資訊，請參閱 使用參數群組。</p>	<p>CLI 選項：</p> <pre>--db-parameter-group-name</pre> <p>RDS API 參數：</p> <pre>DBParameterGroupName</pre>	全部
DB subnet group (資料庫子網路群組)	<p>您要用於資料庫叢集的資料庫子網路群組。</p> <p>選取 Choose existing (選擇現有的) 來使用現有的資料庫子網路群組。然後，從 Existing DB subnet groups (現有資料庫子網路群組) 下拉式清單中選擇所需的子網路群組。</p> <p>選擇 Automatic setup (自動設定) 以讓 RDS 選取相容的資料庫子網路群組。如果不存在，RDS 會為您的叢集建立新的子網路群組。</p> <p>如需詳細資訊，請參閱 使用資料庫子網路群組。</p>	<p>CLI 選項：</p> <pre>--db-subnet-group-name</pre> <p>RDS API 參數：</p> <pre>DBSubnetGroupName</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
專用日誌磁碟區	<p>使用專用日誌磁碟區 (DLV)，將資料庫交易日誌儲存在儲存體磁碟區上，此儲存體磁碟區與包含資料庫資料表的磁碟區不同。</p> <p>如需詳細資訊，請參閱 使用專用日誌磁碟區 (DLV)。</p>	<p>CLI 選項：</p> <pre>--dedicated-log-volume</pre> <p>RDS API 參數：</p> <pre>DedicatedLogVolume</pre>	全部
刪除保護	<p>啟用刪除保護可避免您的資料庫執行個體遭意外刪除。如果您使用建立生產資料庫執行個體 AWS Management Console，則預設會啟用刪除保護。</p> <p>如需詳細資訊，請參閱 刪除資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--deletion-protection</pre> <pre>--no-deletion-protection</pre> <p>RDS API 參數：</p> <pre>DeletionProtection</pre>	全部
加密	<p>Enable Encryption (啟用加密)，用來針對此資料庫執行個體啟用靜態加密。</p> <p>如需更多詳細資訊，請參閱 加密 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <pre>--storage-encrypted</pre> <pre>--no-storage-encrypted</pre> <p>RDS API 參數：</p> <pre>StorageEncrypted</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Enhanced Monitoring (增強型監控)	<p>Enable enhanced monitoring (啟用增強型監控)，以針對資料庫執行個體執行所在的作業系統即時收集指標。</p> <p>如需更多詳細資訊，請參閱 使用增強型監控來監控作業系統指標。</p>	<p>CLI 選項：</p> <pre>--monitoring-interval</pre> <pre>--monitoring-role-arn</pre> <p>RDS API 參數：</p> <pre>MonitoringInterval</pre> <pre>MonitoringRoleArn</pre>	全部
Engine type (引擎類型)	選擇要用於此資料庫執行個體的資料庫引擎。	<p>CLI 選項：</p> <pre>--engine</pre> <p>RDS API 參數：</p> <pre>Engine</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
初始資料庫名稱	<p>資料庫執行個體上資料庫的名稱。如果您沒有提供名稱，Amazon RDS 將不會在資料庫執行個體上建立資料庫 (除了 Oracle 和 PostgreSQL 以外)。該名稱不能是由資料庫引擎保留的單詞，而且會視資料庫引擎而有其他約束。</p> <p>二分百二：</p> <ul style="list-style-type: none"> 名稱必須包含 1–8 個英數字元。 它必須以 a-z、A-Z、@、\$ 或 # 開頭，然後是 a-z、A-Z、0-9、_、@、# 或 \$。 不可含有空格。 如需詳細資訊，請參閱 其他考量。 <p>MariaDB 及 MySQL：</p> <ul style="list-style-type: none"> 必須包含 1–64 個英數字元。 <p>Oracle：</p> <ul style="list-style-type: none"> 名稱必須包含 1–8 個英數字元。 不能是 NULL。預設值為 ORCL。 必須以字母開頭。 <p>PostgreSQL：</p>	<p>CLI 選項：</p> <p>--db-name</p> <p>RDS API 參數：</p> <p>DBName</p>	<p>除 SQL Server 以外的所有項目</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
	<ul style="list-style-type: none"> 名稱必須包含 1–63 個英數字元。 開頭必須是字母或底線。後續字元可以是字母、底線或數字 (0-9)。 初始資料庫名稱為 postgres。 		
授權	<p>授權模式的有效值：</p> <ul style="list-style-type: none"> 自攜許可證或 Db2 的市場許可證。 適用於 MariaDB 的一般公有授權模式。 適用於 Microsoft SQL Server 的已包含授權模式。 適用於 MySQL 的一般公有授權模式。 適用於 Oracle 的已包含授權或自帶授權模式。 適用於 PostgreSQL 的 postgresql 授權。 	<p>CLI 選項：</p> <p><code>--license-model</code></p> <p>RDS API 參數：</p> <p><code>LicenseModel</code></p>	全部
Log exports (日誌匯出)	<p>要發佈到 Amazon CloudWatch 日誌的資料庫日誌檔類型。</p> <p>如需詳細資訊，請參閱 將資料庫日誌發佈至 Amazon CloudWatch Logs。</p>	<p>CLI 選項：</p> <p><code>--enable-cloudwatch-logs-exports</code></p> <p>RDS API 參數：</p> <p><code>EnableCloudwatchLogsExports</code></p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Maintenance window (維護時段)	<p>將擱置中修改套用至資料庫執行個體的 30 分鐘時段。如果不在意時段，請選擇 No Preference (無偏好設定)。</p> <p>如需詳細資訊，請參閱 Amazon RDS 維護時段。</p>	<p>CLI 選項：</p> <pre>--preferred-maintenance-window</pre> <p>RDS API 參數：</p> <pre>PreferredMaintenanceWindow</pre>	全部
管理中的主要認證 AWS Secrets Manager	<p>選取 Manage master credentials in AWS Secrets Manager (管理 AWS Secrets Manager 中的主要憑證) 以秘密管理 Secrets Manager 中的主要使用者密碼。</p> <p>選擇性地選擇要用來保護機密的 KMS 金鑰。從您帳戶中的 KMS 金鑰進行選擇，或輸入來自不同帳戶的金鑰。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager。</p>	<p>CLI 選項：</p> <pre>--manage-master-user-password --no-manage-master-user-password</pre> <pre>--master-user-secret-kms-key-id</pre> <p>RDS API 參數：</p> <pre>ManageMasterUserPassword</pre> <pre>MasterUserSecretKmsKeyId</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Master password (主要密碼)	<p>主要使用者帳戶的密碼。根據資料庫引擎，密碼會有下列數量的可列印 ASCII 字元 (不含 /、"、空格和 @)：</p> <ul style="list-style-type: none"> • 二級分貝：八至五十五 • Oracle：8–30 • MariaDB 及 MySQL：8–41 • SQL Server 及 PostgreSQL：8–128 	<p>CLI 選項：</p> <pre>--master-user-password</pre> <p>RDS API 參數：</p> <pre>MasterUserPassword</pre>	全部
主要使用者名稱	<p>用作具有所有資料庫權限之主要使用者名稱來登入資料庫執行個體的名稱。請注意以下命名限制：</p> <ul style="list-style-type: none"> • 名稱可以包含 1 至 16 個英數字元和底線。 • 第一個字元必須是字母。 • 名稱不能是資料庫引擎的保留字。 <p>您無法在建立資料庫執行個體之後變更主要使用者名稱。</p> <p>對於 Db2，我們建議您使用與自我管理 Db2 執行個體名稱相同的主要使用者名稱。</p> <p>如需授予主要使用者權限的詳細資訊，請參閱 主要使用者帳戶權限。</p>	<p>CLI 選項：</p> <pre>--master-username</pre> <p>RDS API 參數：</p> <pre>MasterUsername</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Microsoft SQL Server Windows Authentication (Microsoft SQL Server Windows 驗證)	Enable Microsoft SQL Server Windows authentication (啟用 Microsoft SQL Server Windows 驗證), 然後 Browse Directory (瀏覽目錄) 以選擇您要允許授權網域使用者使用 Windows 驗證以驗證此 SQL Server 執行個體的目錄。	CLI 選項 : --domain --domain-iam-role-name RDS API 參數 : Domain DomainIAMRoleName	SQL Server
Multi-AZ deployment (異地同步備份部署)	Create a standby instance (建立待命執行個體), 在另一個可用區域中建立資料庫執行個體的被動次要複本, 以提供容錯移轉支援。我們建議針對生產工作負載使用 Multi-AZ 以維持高可用性。 針對開發和測試, 您可以選擇 Do not create a standby instance (不要建立待命執行個體)。	CLI 選項 : --multi-az --no-multi-az RDS API 參數 : MultiAZ	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
國家字元集 (NCHAR)	<p>資料庫執行個體的國家字元集，通常稱為 NCHAR 字元集。您可以將國家字元集設定為 AL16UTF16 (預設) 或 UTF-8。建立資料庫執行個體之後，您無法變更國家字元集。</p> <p>國家字元集與資料庫字元集不同。與資料庫字元集不同，國家字元集只會指定 NCHAR 資料類型 (NCHAR、NVARCHAR2 和 NCLOB) 欄的編碼，而不會影響資料庫中繼資料。</p> <p>如需更多詳細資訊，請參閱 RDS for Oracle 字元集。</p>	<p>CLI 選項：</p> <pre>--nchar-character-set-name</pre> <p>API 參數：</p> <pre>NcharCharacterSetName</pre>	Oracle
Network type (網路類型)	<p>資料庫執行個體支援的 IP 定址通訊協定。</p> <p>IPv4 (預設值)，指定資源只能透過網際網路通訊協定第 4 版 (IPv4) 定址通訊協定與資料庫執行個體進行通訊。</p> <p>Dual-stack mode (雙堆疊模式)指定資源可透過 IPv4、網際網路通訊協定第 6 版 (IPv6) 或兩者與資料庫執行個體進行通訊。如果您有任何資源必須透過 IPv6 定址通訊協定與您的資料庫執行個體進行通訊，請使用雙堆疊模式。此外，請確保將 IPv6 CIDR 區塊與您指定的資料庫子網路群組中的所有子網路相關聯。</p> <p>如需詳細資訊，請參閱 Amazon RDS IP 定址。</p>	<p>CLI 選項：</p> <pre>--network-type</pre> <p>RDS API 參數：</p> <pre>NetworkType</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Option group (選項群組)	<p>資料庫執行個體的選項群組。您可以選擇預設選項群組，或可以建立自訂選項群組。</p> <p>如需更多詳細資訊，請參閱 使用選項群組。</p>	<p>CLI 選項：</p> <pre>--option-group-name</pre> <p>RDS API 參數：</p> <pre>OptionGroupName</pre>	全部
Performance Insights (績效詳情)	<p>Enable Performance Insights (啟用績效詳情) 會監控您的資料庫執行個體負載，讓您可分析資料庫效能並對其進行故障排除。</p> <p>選擇保留期間，以判斷要保留多少 Performance Insights 資料歷史記錄。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 Performance Insights 的定價和資料保留。</p> <p>選擇一個 KMS 金鑰，用來保護用於加密此資料庫磁碟區的金鑰。從您帳戶中的 KMS 金鑰進行選擇，或輸入來自不同帳戶的金鑰。</p> <p>如需詳細資訊，請參閱 在 Amazon RDS 上使用績效詳情監控資料庫負載。</p>	<p>CLI 選項：</p> <pre>--enable-performance-insights</pre> <pre>--no-enable-performance-insights</pre> <pre>--performance-insights-retention-period</pre> <pre>--performance-insights-kms-key-id</pre> <p>RDS API 參數：</p> <pre>EnablePerformanceInsights</pre> <pre>PerformanceInsightsRetentionPeriod</pre> <pre>PerformanceInsightsKMSKeyId</pre>	除了 Db2 以外的全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
佈建 IOPS	<p>資料庫執行個體的新佈建 IOPS (每秒輸入/輸出作業數) 值。只有在您針對 Storage type (儲存體類型) 選擇下列其中一項時，才能使用此設定：</p> <ul style="list-style-type: none"> • General purpose SSD (gp3) (一般用途 SSD (gp3)) • Provisioned IOPS SSD (io1) (佈建 IOPS SSD (io1)) • 佈建的 IOPS 固態硬碟 (io2) <p>如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體。</p>	<p>CLI 選項：</p> <p><code>--iops</code></p> <p>RDS API 參數：</p> <p>Iops</p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
公用存取	<p>Yes (是), 用來給予資料庫執行個體一個公有 IP 地址, 這表示可在 VPC 外加以存取。資料庫執行個體也必須位於 VPC 的公有子網路中, 才能公開存取。</p> <p>No (否) 會讓資料庫執行個體只能從 VPC 內存取。</p> <p>如需詳細資訊, 請參閱 在 VPC 中的網際網路中隱藏資料庫執行個體。</p> <p>若要從其 VPC 外部連線到資料庫執行個體, 資料庫執行個體必須可公開存取。亦須使用資料庫執行個體安全群組的傳入規則授予存取權。此外, 必須滿足其他要求。如需詳細資訊, 請參閱 無法連線至 Amazon RDS 資料庫執行個體。</p> <p>如果您的資料庫執行個體無法公開存取, 請使用 AWS Site-to-Site VPN 連線或 AWS Direct Connect 連線從私人網路存取。如需詳細資訊, 請參閱 網際網路流量隱私權。</p>	<p>CLI 選項 :</p> <p><code>--publicly-accessible</code></p> <p><code>--no-publicly-accessible</code></p> <p>RDS API 參數 :</p> <p><code>PubliclyAccessible</code></p>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
RDS 延伸 Support	<p>選取「啟用 RDS 延伸 Support 援」以允許支援的主要引擎版本在 RDS 標準支援結束日期之後繼續執行。</p> <p>當您建立資料庫執行個體時，Amazon RDS 預設為 RDS 延伸 Support。若要防止在 RDS 標準 Support 結束日期之後建立新的資料庫執行個體，並避免 RDS 延伸支援的費用，請停用此設定。在 RDS 延伸 Support 定價開始日期之前，您現有的資料庫執行個體不會產生費用。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS 延長支援。</p>	<p>CLI 選項：</p> <pre>--engine-lifecycle-support</pre> <p>RDS API 參數：</p> <pre>EngineLifecycleSupport</pre>	<p>MySQL</p> <p>PostgreSQL</p>
RDS Proxy	<p>選擇 Create an RDS Proxy (建立 RDS Proxy)，針對您的資料庫執行個體建立一個代理。Amazon RDS 會自動為代理建立 IAM 角色和 Secrets Manager 機密。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS Proxy。</p>	無法在建立資料庫執行個體時使用。	<p>MariaDB</p> <p>MySQL</p> <p>PostgreSQL</p>
儲存體自動調整規模	<p>Enable storage autoscaling (啟用儲存體自動調整規模) 可讓 Amazon RDS 在需要避免資料庫執行個體耗盡儲存空間時，自動增加儲存體。</p> <p>使用 Maximum storage threshold (儲存體閾值上限) 可設定 Amazon RDS 的上限，以自動增加資料庫執行個體的儲存體。預設為 1,000 GiB。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS 儲存體自動擴展自動管理容量。</p>	<p>CLI 選項：</p> <pre>--max-allocated-storage</pre> <p>RDS API 參數：</p> <pre>MaxAllocatedStorage</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
Storage throughput (訊息輸送量)	<p>資料庫執行個體的儲存體輸送量值。只有在您針對 Storage type (儲存體類型) 選擇 General purpose SSD (gp3) (一般用途 SSD (gp3)) 時，才能使用此設定。</p> <p>如需詳細資訊，請參閱 gp3 儲存裝置 (建議使用)。</p>	<p>CLI 選項：</p> <pre>--storage-throughput</pre> <p>RDS API 參數：</p> <pre>StorageThroughput</pre>	全部
儲存體類型	<p>資料庫執行個體的儲存類型。</p> <p>如果您選擇 General Purpose SSD (gp3) (一般用途 SSD (gp3))，您可以在 Advanced settings (進階設定) 下佈建額外的佈建 IOPS 和儲存體輸送量。</p> <p>如果您選擇佈建 IOPS 固態硬碟 (io1) 或佈建 IOPS 固態硬碟 (io2)，請輸入佈建的 IOPS 值。</p> <p>如需詳細資訊，請參閱 Amazon RDS 儲存類型。</p>	<p>CLI 選項：</p> <pre>--storage-type</pre> <p>RDS API 參數：</p> <pre>StorageType</pre>	全部
Subnet group (子網路群組)	<p>要與此資料庫執行個體建立關聯的資料庫子網路群組。</p> <p>如需詳細資訊，請參閱 使用資料庫子網路群組。</p>	<p>CLI 選項：</p> <pre>--db-subnet-group-name</pre> <p>RDS API 參數：</p> <pre>DBSubnetGroupName</pre>	全部

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
租戶資料庫名稱	<p>Oracle 架構之多租戶組態中初始 PDB 的名稱。只有當您針對架構組態選擇多租戶組態時，才能使用此設定。</p> <p>租用戶資料庫名稱必須與 CDB 的名稱 (名為 RDSCDB) 不同。您無法變更 CDB 名稱。</p>	<p>CLI 選項：</p> <p>--db-name</p> <p>RDS API 參數：</p> <p>DBName</p>	Oracle
租戶資料庫主要使用者名稱	<p>用作具有所有資料庫權限之主要使用者名稱來登入租戶資料庫 (PDB) 的名稱。只有當您針對架構組態選擇多租戶組態時，才能使用此設定。</p> <p>請注意以下命名限制：</p> <ul style="list-style-type: none"> 名稱可以包含 1 至 16 個英數字元和底線。 第一個字元必須是字母。 名稱不能是資料庫引擎的保留字。 <p>您無法執行下列：</p> <ul style="list-style-type: none"> 在建立租戶資料庫之後，變更租戶主要使用者名稱。 使用租戶主要使用者名稱登入 CDB。 	<p>CLI 選項：</p> <p>--master-username</p> <p>RDS API 參數：</p> <p>MasterUsername</p>	Oracle
租戶資料庫主要密碼	<p>租戶資料庫 (PDB) 之主要使用者帳戶的密碼。只有當您針對架構組態選擇多租戶組態時，才能使用此設定。</p> <p>密碼必須包含 8 至 30 個可列印的 ASCII 字元 (/、"、空格和 @ 除外)。</p>	<p>CLI 選項：</p> <p>--master-password</p> <p>RDS API 參數：</p> <p>MasterPassword</p>	Oracle

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
租戶資料庫字元集	<p>初始租戶資料庫的字元集。只有當您針對架構組態選擇多租戶組態時，才能使用此設定。僅支援 RDS for Oracle CDB 資料庫執行個體。</p> <p>租戶資料庫字元集的預設值 AL32UTF8 適用於 Unicode 5.0 UTF-8 通用字元集。您可以選擇不同於 CDB 字元集的租戶資料庫字元集。</p> <p>如需詳細資訊，請參閱 RDS for Oracle 字元集。</p>	<p>CLI 選項：</p> <pre>--character-set-name</pre> <p>RDS API 參數：</p> <pre>CharacterSetName</pre>	Oracle
租戶資料庫國家字元集	<p>租戶資料庫的國家字元集，通常稱為 NCHAR 字元集。只有當您針對架構組態選擇多租戶組態時，才能使用此設定。僅支援 RDS for Oracle CDB 資料庫執行個體。</p> <p>您可以將國家字元集設定為 AL16UTF16 (預設值) 或 UTF-8。在建立租戶資料庫之後，您無法變更國家字元集。</p> <p>租戶資料庫國家字元集與租戶資料庫字元集不同。國家字元集只會針對使用 NCHAR 資料類型 (NCHAR、NVARCHAR2 和 NCLOB) 的資料行指定編碼，而且不會影響資料庫中繼資料。</p> <p>如需詳細資訊，請參閱 RDS for Oracle 字元集。</p>	<p>CLI 選項：</p> <pre>--nchar-character-set-name</pre> <p>API 參數：</p> <pre>NcharCharacterSetName</pre>	Oracle

主控台設定	設定說明	CLI 選項和 RDS API 參數	支援的資料庫引擎
時區	<p>資料庫執行個體的時區。如果您不選擇時區，資料庫執行個體會使用預設時區。在建立資料庫執行個體之後，就無法變更時區。</p> <p>如需詳細資訊，請參閱 適用於 Db2 資料庫執行個體的 Amazon RDS 的當地時區 及 Microsoft SQL Server 資料庫執行個體的本機時區。</p>	<p>CLI 選項：</p> <p>--timezone</p> <p>RDS API 參數：</p> <p>Timezone</p>	<p>Db2</p> <p>SQL Server</p> <p>RDS Custom for SQL Server</p>
Virtual Private Cloud (VPC)	<p>要與此資料庫執行個體相關聯的 VPC (以 Amazon VPC 服務為基礎)。</p> <p>如需詳細資訊，請參閱 Amazon VPC 和 Amazon RDS。</p>	對於 CLI 及 API，您指定 VPC 安全性群組 ID。	全部
VPC security group (firewall) (VPC 安全群組 (防火牆))	<p>要與資料庫執行個體建立關聯的安全群組。</p> <p>如需詳細資訊，請參閱 VPC 安全群組概觀。</p>	<p>CLI 選項：</p> <p>--vpc-security-group-ids</p> <p>RDS API 參數：</p> <p>VpcSecurityGroupIds</p>	全部

透過 AWS CloudFormation 建立 Amazon RDS 資源

Amazon RDS 已整合 AWS CloudFormation，這項服務可協助您建立 AWS 資源的模型和設定，以減少建立和管理資源和基礎設施的時間。您可以建立描述您想要的所有 AWS 資源的範本 (例如資料庫執行個體和資料庫參數群組)，並且 AWS CloudFormation 會為您佈建和設定這些資源。

當您使用 AWS CloudFormation 時，您可以重複使用範本，以便重複且一致地設定 RDS 資源。只需描述一次您的資源，即可在多個 AWS 帳戶與區域內重複佈建相同資源。

RDS和 AWS CloudFormation 範本

若要佈建和設定 RDS 與相關服務的資源，您必須了解 [AWS CloudFormation 範本](#)。範本是以 JSON 或 YAML 格式化的文本檔案。而您亦可以透過這些範本的說明，了解欲在 AWS CloudFormation 堆疊中佈建的資源。如果您不熟悉 JSON 或 YAML，您可以使用 AWS CloudFormation Designer 協助您開始使用 AWS CloudFormation 範本。如需更多詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 [什麼是 AWS CloudFormation 設計工具？](#)。

RDS 支援在 AWS CloudFormation 中建立資源。如需更多詳細資訊 (包括這些資源的 JSON 和 YAML 範本範例)，請參閱 AWS CloudFormation 使用者指南中的 [RDS 資源類型參考](#)。

進一步了解 AWS CloudFormation

若要進一步了解 AWS CloudFormation，請參閱下列資源：

- [AWS CloudFormation](#)
- 《[AWS CloudFormation 使用者指南](#)》
- [AWS CloudFormation API 參考](#)
- 《[AWS CloudFormation 命令列介面使用者指南](#)》

連接至 Amazon RDS 資料庫執行個體

您必須先建立資料庫執行個體，才能連線到資料庫執行個體。如需相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。在 Amazon RDS 佈建您的資料庫執行個體之後，請使用任何標準用戶端應用程式或資料庫引擎公用程式來連線至資料庫執行個體。在連線字串中，指定來自資料庫執行個體端點的 DNS 位址做為主機參數。此外，也會指定來自資料庫執行個體端點的連接埠號碼做為連接埠參數。

主題

- [尋找 Amazon RDS 資料庫執行個體的連線資訊](#)
- [資料庫身分驗證選項](#)
- [加密的連線](#)
- [在 VPC 中存取資料庫執行個體的案例](#)
- [使用 AWS 驅動程式連接資料庫執行個體](#)
- [連線至執行特定資料庫引擎的資料庫執行個體](#)
- [使用 RDS Proxy 管理連線](#)

尋找 Amazon RDS 資料庫執行個體的連線資訊

資料庫執行個體的連線資訊包括其端點、連接埠，以及有效的資料庫使用者，例如主要使用者。例如，對於 MySQL 資料庫執行個體，假設端點值為 `mydb.123456789012.us-east-1.rds.amazonaws.com`。在此情況下，連接埠值為 3306，資料庫使用者為 `admin`。藉由此資訊，您可以在連線字串中指定下列值：

- 對於主機或主機名稱或 DNS 名稱，請指定 `mydb.123456789012.us-east-1.rds.amazonaws.com`。
- 對於連接埠，指定 3306。
- 對於使用者，請指定 `admin`。

每個資料庫執行個體的端點都是唯一的，連接埠和使用者的值可能會有所不同。下列清單顯示每個資料庫引擎最常見的連接埠：

- DB2 — 50000
- MariaDB – 3306
- Microsoft SQL Server – 1433

- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

若要連線到資料庫執行個體，請針對資料庫引擎使用任何用戶端。例如，您可以使用 mysql 公用程式來連線至 MariaDB 或 MySQL 資料庫執行個體。您可以使用 Microsoft SQL Server Management Studio 連線至 SQL Server 資料庫執行個體。您可能會使用 Oracle SQL Developer 連線至 Oracle 資料庫執行個體。同樣地，您可能會使用 psql 命令列公用程式，連線至 PostgreSQL 資料庫執行個體。

若要尋找資料庫執行個體的連線資訊，請使用 AWS Management Console。您也可以使用 AWS Command Line Interface (AWS CLI) [describe-db-instances](#) 命令或 RDS API [DescribeDBInstances](#) 作業。

主控台

若要尋找資料庫執行個體的連線資訊 AWS Management Console

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫) 以顯示資料庫執行個體的清單。
3. 選擇資料庫執行個體的名稱以顯示其詳細資訊。
4. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連接至資料庫執行個體。

RDS > Databases > mydb

mydb

Summary

DB identifier mydb	CPU 2.33%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port

Endpoint mydb. [redacted].us-east-1.rds.amazonaws.com	Network
Port 3306	Availability Zone us-east-1
	VPC vpc-65
	Subnet default

5. 如果您需要尋找主要使用者名稱，請選擇 Configuration (組態) 索引標籤並檢視 Master username (主要使用者名稱) 值。

AWS CLI

若要使用尋找資料庫執行個體的連線資訊 AWS CLI，請呼叫命[describe-db-instances](#) 令。在呼叫中，查詢資料庫執行個體 ID、端點、連接埠和主要使用者名稱。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-instances \  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

在 Windows 中：

```
aws rds describe-db-instances ^  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

輸出內容應如下所示：

```
[  
  [  
    "mydb",  
    "mydb.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "myoracledb",  
    "myoracledb.123456789012.us-east-1.rds.amazonaws.com",  
    1521,  
    "dbadmin"  
  ],  
  [  
    "mypostgresldb",  
    "mypostgresldb.123456789012.us-east-1.rds.amazonaws.com",  
    5432,  
    "postgresadmin"  
  ]  
]
```

RDS API

若要使用 Amazon RDS API 尋找資料庫執行個體的連線資訊，請呼叫 [DescribeDBInstances](#) 操作。在輸出中，尋找端點地址、端點連接埠和主要使用者名稱的值。

資料庫身分驗證選項

Amazon RDS 支援以下方式來對資料庫使用者進行身分驗證：

- 密碼身分驗證 – 您的資料庫執行個體會執行使用者帳戶的所有管理。您可以使用 SQL 陳述式建立使用者並指定密碼。您可以使用的 SQL 陳述式取決於您的資料庫引擎。
- AWS Identity and Access Management (IAM) 資料庫身份驗證 — 連線到資料庫執行個體時，不需要使用密碼。而是改用身分驗證字符。
- Kerberos 身分驗證 – 您使用 Kerberos 和 Microsoft Active Directory 之資料庫使用者的外部身分驗證。Kerberos 是網路身分驗證通訊協定，使用票證和對稱式金鑰加密技術，免除透過網路傳輸密碼的需要。Kerberos 已內建至 Active Directory，旨在驗證網路資源 (例如資料庫) 的使用者身分。

IAM 資料庫身分驗證和 Kerberos 身分驗證僅適用於特定的資料庫引擎和版本。

如需更多詳細資訊，請參閱 [使用 Amazon RDS 進行資料庫身分驗證](#)。

加密的連線

您可以從應用程式使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來加密與資料庫執行個體的連線。每個資料庫引擎都有自己實作 SSL/TLS 的程序。如需更多詳細資訊，請參閱。

在 VPC 中存取資料庫執行個體的案例

您可以使用 Amazon Virtual Private Cloud 端 (Amazon VPC) 將 AWS 資源 (例如 Amazon RDS 資料庫執行個體) 啟動到虛擬私有雲端 (VPC)。使用 Amazon VPC 時，您可以掌控您的虛擬聯網環境。您可以選擇自己的 IP 地址範圍、建立子網路，以及設定路由和存取控制清單。

VPC 安全群組可控制對 VPC 內資料庫執行個體的存取。每個 VPC 安全群組都可讓特定來源存取 VPC 中與該 VPC 安全群組相關聯的資料庫執行個體。來源可以是地址的來源 (例如，203.0.113.0/24) 或另一個 VPC 安全群組。藉由指定 VPC 安全群組做為來源，您允許從使用來源 VPC 安全群組的所有執行個體 (通常指的是應用程式伺服器) 傳入的流量。

嘗試連線至資料庫執行個體之前，請為您的使用案例設定 VPC。以下為在 VPC 中存取資料庫執行個體的常見案例：

- 由同一 VPC 中的 Amazon EC2 執行個體存取的 VPC 資料庫執行個體 – VPC 中資料庫執行個體的常見用途是與在同一個 VPC 中的 EC2 執行個體中執行的應用程式伺服器共享資料。EC2 執行個體可能會使用與資料庫執行個體互動的應用程式來執行 Web 伺服器。

- 由不同 VPC 中的 EC2 執行個體存取之 VPC 中的資料庫執行個體 – 在某些情況下，資料庫執行個體與您用來存取它的 EC2 執行個體各自位於不同的 VPC 中。若是如此，您可以使用 VPC 對等互連來存取資料庫執行個體。
- 用戶端應用程式透過網際網路存取之 VPC 中的資料庫執行個體 – 若要透過網際網路從用戶端應用程式存取 VPC 中的資料庫執行個體，請設定具有單一公有子網路的 VPC。您也可以設定網際網路閘道，啟用透過網際網路的通訊。

若要從其 VPC 外部連線到資料庫執行個體，資料庫執行個體必須可公開存取。此外，必須使用資料庫執行個體安全群組的入站規則授予存取權，且必須符合其他需求。如需詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

- 私有網路存取之 VPC 中的資料庫執行個體 – 如果您的資料庫執行個體無法公開存取，您可以使用下列其中一個選項，從私有網路存取該執行個體。
 - AWS Site-to-Site VPN 連線
 - 一個 AWS Direct Connect 連接
 - 一個 AWS Client VPN 連接

如需詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

使用 AWS 驅動程式連接資料庫執行個體

驅動程式 AWS 套件的設計旨在提供更快的切換和容錯移轉時間，以及使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合身分進行身份驗證的支援。AWS 驅動程式仰賴監控資料庫執行個體狀態並瞭解執行個體拓撲來判斷新的主執行個體。這種方法可將切換和容錯移轉時間縮短為個位數秒，而開放原始碼驅動程式則需要數十秒。

下表列出每個驅動程式支援的功能。隨著新的服務功能推出，驅動程序 AWS 套件的目標是內置支持這些服務功能。

功能	AWS 驅動程式	AWS Python 驅動
故障轉移支	是	是
增強的容錯移轉	是	是
讀/寫分割	是	是
驅動程式元數據	是	N/A

功能	AWS 驅動程式	AWS Python 驅動
遙測	是	是
Secrets Manager	是	是
IAM 身分驗證	是	是
同盟身分識別 (AD FS)	是	是
同盟身份 (奧克塔)	是	否
多可用區域資料庫叢集	是	是

如需有關 AWS 驅動程式的詳細資訊，請參閱適用於您的 RDS、[MySQL 版 RDS MariaDB 適用於 PostgreSQL 資料庫執行個體的 RDS](#) 的對應語言驅動程式。

Note

適用於 MariaDB 的 RDS 唯一支援的功能是使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合身分進行身分驗證。

連線至執行特定資料庫引擎的資料庫執行個體

如需連線至執行特定資料庫引擎之資料庫執行個體的相關資訊，請遵循資料庫引擎的指示：

- [連接到您的 Amazon RDS 用 Db2 資料庫執行個體](#)
- [連接至執行 MariaDB 資料庫引擎的資料庫執行個體](#)
- [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)
- [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)
- [連線至 RDS for Oracle 資料庫執行個體](#)
- [連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)

使用 RDS Proxy 管理連線

您也可以使用 Amazon RDS Proxy 來管理 RDS for MariaDB、RDS for Microsoft SQL Server、RDS for MySQL 和 RDS for PostgreSQL 資料庫執行個體的連線。RDS Proxy 允許應用程式將資料庫連線置於集區和共用，以改善可擴展性。如需更多詳細資訊，請參閱 [使用 Amazon RDS Proxy](#)。

使用選項群組

某些資料庫引擎提供了簡化資料與資料庫管理的其他功能，並提供額外的資料庫安全性。Amazon RDS 使用選項群組來啟用並設定這些功能。選項群組可指定稱作選項的功能，可用於特定的 Amazon RDS 資料庫執行個體。選項擁有的設定，可以指定該選項如何運作。當您將資料庫執行個體關聯到一個選項群組後，便會啟用該資料庫執行個體的指定選項和選項設定。

Amazon RDS 支援下列資料庫引擎的選項：

資料庫引擎	相關文件
MariaDB	MariaDB 資料庫引擎的選項
Microsoft SQL Server	Microsoft SQL Server 資料庫引擎的選項
MySQL	MySQL 資料庫執行個體的選項
Oracle	將選項新增至 Oracle 資料庫執行個體
PostgreSQL	PostgreSQL 不使用選項和選項群組。PostgreSQL 使用擴展和模塊來提供附加功能。如需更多詳細資訊，請參閱 支援的 PostgreSQL 擴充功能版本 。

選項群組概觀

Amazon RDS 為每個新資料庫執行個體提供了空的預設選項群組。您不能修改或刪除此預設選項群組，但任何您新建的選項群組都會沿用預設選項群組的設定。要套用選項至資料庫執行個體，您必須執行下列項目：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 在選項群組中新增一或多個選項。
3. 將選項群組與資料庫執行個體建立關聯。

若要建立選項群組與資料庫執行個體的關聯，請修改資料庫執行個體。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

資料庫執行個體和資料庫快照都可以與選項群組關聯。在某些情況下，您可能會從資料庫快照還 point-in-time 原或為資料庫執行個體執行還原。在這類情況下，與資料庫快照或資料庫執行個體關聯的選項

群組，依據預設會與還原的資料庫執行個體產生關聯。您可以將不同的選項群組關聯至還原的資料庫執行個體。但是，新的選項群組必須包含任何在原始選項群組中所納入的持久性或永久性選項。持久性與永久性選項如下所述。

選項需要額外記憶體才能在資料庫執行個體上執行，因此您可能需要啟動更大的執行個體以使用選項，視您目前的資料庫執行個體使用狀態而定。例如，「Oracle Enterprise Manager 資料庫控制」使用大約 300 MB 的 RAM。如果您為小型資料庫執行個體啟用此選項，可能會遇到效能問題或 out-of-memory 錯誤。

持久性與永久性選項

在將持久性與永久性兩種類型選項新增至選項群組時，需要特殊考量。

資料庫執行個體與選項群組建立關聯時，無法由選項群組移除持久性選項。例如 Microsoft SQL Server 透明資料加密 (TDE) 的 TDE 選項就是持久性選項。在從選項群組中刪除持久性選項之前，必須先取消所有資料庫執行個體與選項群組的關聯。在某些情況下，您可能會從資料庫快照 point-in-time 還原或執行還原。在這類情況下，如果與該資料庫快照關聯的選項群組包含持久性選項，則只能將還原的資料庫執行個體與該選項群組建立關聯。

永久性選項 (例如 Oracle 進階安全性 TDE 的 TDE 選項) 絕不能從選項群組中移除。您可變更使用永久性選項資料庫執行個體的選項群組。不過與資料庫執行個體建立關聯的選項群組，必須包含相同的永久性選項。在某些情況下，您可能會從資料庫快照 point-in-time 還原或執行還原。在這類情況下，如果與該資料庫快照關聯的選項群組包含永久性選項，則只能將還原的資料庫執行個體與包含永久性選項的該選項群組建立關聯。

對於 Oracle 資料庫執行個體，您可以複製具有 Timezone 或 OLS 選項 (或兩者) 的共用資料庫快照。作法是在您複製資料庫快照時，指定包含這些選項的目標選項群組。只有對執行 Oracle 12.2 版或更高版本的 Oracle 資料庫執行個體，OLS 才是永久和持續的選項。如需這些選項的詳細資訊，請參閱 [Oracle 時區](#) 和 [Oracle Label Security](#)。

VPC 考量

與資料庫執行個體相關聯的選項群組會與資料庫執行個體的 VPC 連結。這表示若您嘗試將該執行個體還原至不同的 VPC，將無法使用指派給資料庫執行個體的選項群組。如果您將資料庫執行個體還原至不同的 VPC，可以進行以下其中一項作業：

- 將預設選項群組指派至資料庫執行個體。
- 指派與該 VPC 連結的選項群組。
- 建立新的選項群組，並將其指派至資料庫執行個體。

若使用持久性或永久性選項 (例如 Oracle TDE)，您必須建立新的選項群組。此選項群組必須在將資料庫執行個體還原至不同的 VPC 時包含持久性或永久性選項。

選項設定控制了選項的行為。例如，Oracle 進階安全性選項 `NATIVE_NETWORK_ENCRYPTION` 並擁有您可以用來為進出資料庫執行個體網路流量指定加密演算法的設定。某些選項設定已為搭配 Amazon RDS 的使用進行最佳化且不能變更。

互斥選項

某些選項是互斥的。您只能一次使用一個選項。以下選項是互斥的：

- [Oracle Enterprise Manager Database Express](#)、與 [適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent](#)
- [Oracle 原生網路加密](#)、與 [Oracle Secure Sockets Layer](#)

建立選項群組

您可以建立新的選項群組，從預設選項群組衍生其設定。然後將一或多個選項新增至新的選項群組。或者，如果您已經有一個現有選項群組，您可以將包含其所有選項的選項群組複製到新的選項群組。如需詳細資訊，請參閱 [刪除選項群組](#)。

當您建立新的選項群組後，它不會有選項。如需了解如何新增選項群組的選項，請參閱 [將選項新增至選項群組](#)。新增所需選項後，您可以將選項群組與資料庫執行個體建立關聯。如此一來，這些選項便可在資料庫執行個體上使用。如需將選項群組與資料庫執行個體建立關聯的相關資訊，請參閱 [使用選項群組](#) 中引擎的文件。

主控台

建立選項群組的其中一種方法是使用 AWS Management Console。

使用主控台建立新的選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 視窗中，執行下列動作：
 - a. 在 [名稱] 中，輸入 AWS 帳戶中唯一的選項群組名稱。名稱僅可包含字母、數字與連字號。

- b. 在 Description (說明) 欄位中，輸入選項群組的簡要說明。用於顯示用途的說明。
 - c. 針對 Engine (引擎)，選擇您想要使用的資料庫引擎。
 - d. 在 Major engine version (主要引擎版本) 中，選擇您要使用的主要資料庫引擎版本。
5. 請選擇 Create (建立) 以繼續進行。若要取消操作，請改為選擇 Cancel (取消)。

AWS CLI

若要建立選項群組，請使用具有下列必要參數的 AWS CLI [create-option-group](#) 指令。

- --option-group-name
- --engine-name
- --major-engine-version
- --option-group-description

Example

以下範例會建立名為 `testoptiongroup` 的選項群組，該群組與 Oracle Enterprise Edition 資料庫引擎關聯。該說明包含在引號中。

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name testoptiongroup \  
  --engine-name oracle-ee \  
  --major-engine-version 19 \  
  --option-group-description "Test option group for Oracle Database 19c EE"
```

在 Windows 中：

```
aws rds create-option-group ^  
  --option-group-name testoptiongroup ^  
  --engine-name oracle-ee ^-  
  --major-engine-version 19 ^  
  --option-group-description "Test option group for Oracle Database 19c EE"
```

RDS API

如要建立選項群組，請呼叫 Amazon RDS API [CreateOptionGroup](#) 操作。包含以下參數：

- OptionGroupName
- EngineName
- MajorEngineVersion
- OptionGroupDescription

刪除選項群組

您可以使用 AWS CLI 或 Amazon RDS API 複製選項組。複製選項群組可以很方便。例如，當您已建立現有的選項群組，並希望將其中大部份的自訂參數及值併入新的選項群組時。您還可以複製在生產中使用的選項群組，然後修改複本以測試其他選項設定。

Note

您目前無法將選項群組複製到其他「AWS 區域」。

AWS CLI

若要複製選項群組，請使用 AWS CLI [複製選](#)項群組指令。包含下列必要選項：

- --source-option-group-identifier
- --target-option-group-identifier
- --target-option-group-description

Example

以下範例會建立名為 new-option-group 的選項群組，該群組為選項群組 my-option-group 的本機複本。

對於LinuxmacOS、或Unix：

```
aws rds copy-option-group \
```

```
--source-option-group-identifier my-option-group \  
--target-option-group-identifier new-option-group \  
--target-option-group-description "My new option group"
```

在 Windows 中：

```
aws rds copy-option-group ^  
--source-option-group-identifier my-option-group ^  
--target-option-group-identifier new-option-group ^  
--target-option-group-description "My new option group"
```

RDS API

若要複製選項群組，請呼叫 Amazon RDS API [CopyOption群組](#) 操作。包含下列必要參數。

- SourceOptionGroupIdentifier
- TargetOptionGroupIdentifier
- TargetOptionGroupDescription

將選項新增至選項群組

您可以將選項新增至現有選項群組。新增所需選項後，您可以將選項群組與資料庫執行個體關聯，以便使用資料庫執行個體上的選項。如需關聯選項群組與資料庫執行個體的資訊，請參閱在 [使用選項群組](#) 中列出的特定資料庫引擎文件。

在兩種情況下必須立即套用選項群組變更：

- 當您要新增一個新增或更新連接埠值的選項時 (例如 OEM 選項)。
- 使用包含連接埠值的選項新增或刪除選項群組時。

在這類情況下，請在主控制台選擇 Apply Immediately (立即套用) 選項。或是您可以在使用 AWS CLI 時納入 `--apply-immediately` 選項，或在使用 Amazon RDS API 時將 `ApplyImmediately` 參數設定為 `true`。不包含連接埠值的選項可以立即套用，或在資料庫執行個體的下一次維護時段套用。

Note

如果您指定安全群組，以作為選項群組中的選項值，請修改選項群組來管理安全群組。您無法藉由修改資料庫執行個體來變更或移除此安全群組。此外，安全群組不會出現在 AWS

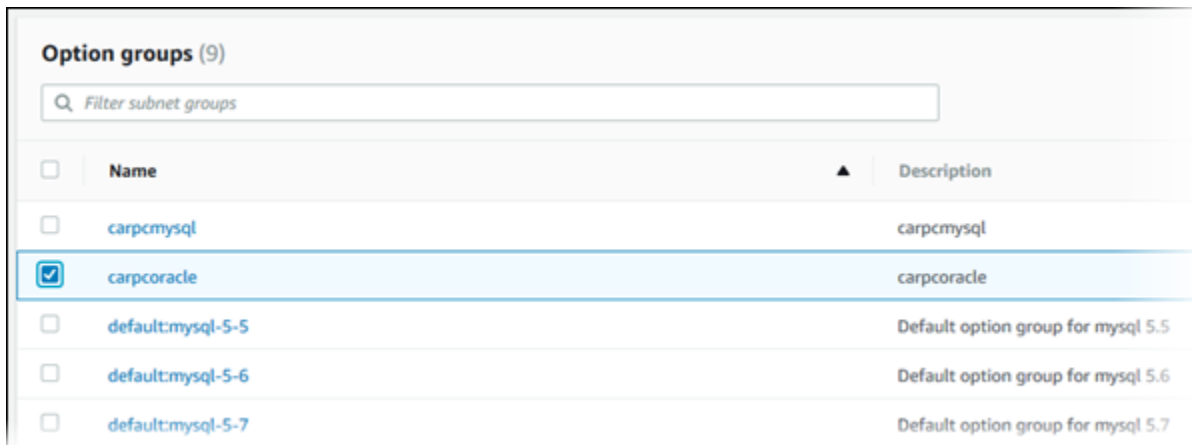
CLI 命令的輸出 AWS Management Console 或資料庫執行個體詳細資料中 describe-db-instances。

主控台

您可以使用 AWS Management Console 將選項新增至選項群組。

使用主控台將選項新增至選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您要修改的選項群組，然後選擇 Add option (新增選項)。



4. 在 Add option (新增選項) 視窗中，執行下列作業：
 - a. 選擇您要新增的選項。根據您選擇的選項，您可能需要提供其他值。例如，當您選擇 OEM 選項時，您必須也輸入連接埠值並指定一個安全群組。
 - b. 若要在所有關聯的資料庫執行個體新增選項時隨即啟用選項，請在 Apply Immediately (立即套用) 選擇 Yes (是)。如果您選擇了 No (不) (預設)，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。

Add Option

Option details

Option group name
carpcoracle

Option
Name of Option you want to add to this group
OEM

Port
The port number, if applicable, to use when connecting to the Option
1158

Security Groups
A list of VPC or DB Security Groups for which this Option is enabled
Choose security groups
default X

Apply Immediately [info](#)
 Yes
 No

Cancel Add Option

5. 當您滿意設定後，選擇 Add option (新增選項)。

AWS CLI

若要將選項新增至選項群組，請使用您要新 AWS CLI [增的選項來執行新增選項至選項群組](#) 命令。若要在所有關聯的資料庫執行個體立即啟用新選項，請將 `--apply-immediately` 參數納入。依預設，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。包含下列必要參數：

- `--option-group-name`

Example

下列範例會將具有 America/Los_Angeles 設定的 Timezone 選項新增至名為的選項群組，testoptiongroup 並立即啟用該選項。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name testoptiongroup \  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/  
Los_Angeles}]" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name testoptiongroup ^  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/  
Los_Angeles}]" ^  
  --apply-immediately
```

命令結果類似以下所示：

```
...{  
  "OptionName": "Timezone",  
  "OptionDescription": "Change time zone",  
  "Persistent": true,  
  "Permanent": false,  
  "OptionSettings": [  
    {  
      "Name": "TIME_ZONE",  
      "Value": "America/Los_Angeles",  
      "DefaultValue": "UTC",  
      "Description": "Specifies the timezone the user wants to change the  
system time to",  
      "ApplyType": "DYNAMIC",  
      "DataType": "STRING",  
      "AllowedValues": "Africa/Cairo,...",  
      "IsModifiable": true,  
      "IsCollection": false  
    }  
  ],  
  "DBSecurityGroupMemberships": [],  
  "VpcSecurityGroupMemberships": []  
}...
```


Example

以下範例將 Oracle OEM 選項新增至選項群組。其中也指定自訂連接埠及一對 Amazon EC2 VPC 安全群組，用於該連接埠。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \
  --option-group-name testoptiongroup \
  --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" \
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^
  --option-group-name testoptiongroup ^
  --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" ^
  --apply-immediately
```

命令結果類似以下所示：

```
OPTIONGROUP  False  oracle-ee  19  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
Test Option Group  testoptiongroup  vpc-test
OPTIONS Oracle 12c EM Express  OEM      False  False  5500
VPCSECURITYGROUPMEMBERSHIPS  active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS  active  sg-test2
```

Example

下列範例會將 Oracle 選項新增NATIVE_NETWORK_ENCRYPTION至選項群組，並指定選項設定。若未指定選項設定，則將使用預設值。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \
  --option-group-name testoptiongroup \
  --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"}],
```

```
{"Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256, AES192, DES"}], "OptionName": "NATIVE_NETWORK_ENCRYPTION",
  \
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^
  --option-group-name testoptiongroup ^
  --options "OptionSettings"=[{"Name": "SQLNET.ENCRYPTION_SERVER", "Value": "REQUIRED"},
{"Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256\, AES192\, DES"}], "OptionName": "NATIVE_NETWORK_ENCRYPTION",
  ^
  --apply-immediately
```

命令結果類似以下所示：

```
...{
  "OptionName": "NATIVE_NETWORK_ENCRYPTION",
  "OptionDescription": "Native Network Encryption",
  "Persistent": false,
  "Permanent": false,
  "OptionSettings": [
    {
      "Name": "SQLNET.ENCRYPTION_TYPES_SERVER",
      "Value": "AES256, AES192, DES",
      "DefaultValue":
"RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40",
      "Description": "Specifies list of encryption algorithms in order of
intended use",
      "ApplyType": "STATIC",
      "DataType": "STRING",
      "AllowedValues":
"RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40",
      "IsModifiable": true,
      "IsCollection": true
    },
    {
      "Name": "SQLNET.ENCRYPTION_SERVER",
      "Value": "REQUIRED",
      "DefaultValue": "REQUESTED",
      "Description": "Specifies the desired encryption behavior",
      "ApplyType": "STATIC",
      "DataType": "STRING",
      "AllowedValues": "ACCEPTED, REJECTED, REQUESTED, REQUIRED",
```

```
"IsModifiable": true,  
  "IsCollection": false  
},...
```

RDS API

若要使用 Amazon RDS API 將選項新增至選項群組，請使用您要新增的選項呼叫 [ModifyOption群組](#) 操作。若要在所有關聯的資料庫執行個體立即啟用新選項，請將 `ApplyImmediately` 參數納入並將其設為 `true`。依預設，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。包含下列必要參數：

- `OptionGroupName`

列出選項群組的選項和選項設定

您可列出選項群組所有的選項和選項設定。

主控台

您可以使 AWS Management Console 用列出選項群組的所有選項和選項設定。

列出選項群組的選項和選項設定

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇選項群組名稱以顯示其詳細資訊。將列出選項群組中的選項和選項設定。

AWS CLI

若要列示選項群組的選項和選項設定，請使用 AWS CLI [describe-option-groups](#) 指令。指定您想要檢視其選項和設定的選項群組名稱。如果未指定選項群組名稱，將說明所有的選項群組。

Example

下列範例列出所有選項群組的選項和選項設定。

```
aws rds describe-option-groups
```

Example

下列範例列出名為 `testoptiongroup` 的選項群組的選項和選項設定。

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

RDS API

若要列出某一選項群組的選項和選項設定，請使用 Amazon RDS API [DescribeOptionGroups](#) 操作。指定您想要檢視其選項和設定的選項群組名稱。如果未指定選項群組名稱，將說明所有的選項群組。

修改選項設定

新增有可修改之選項設定的選項後，您可隨時修改其設定。如果您變更選項群組中的選項或選項設定，這些變更將套用到與該選項群組關聯的所有資料庫執行個體。如需各種選項可用設定的詳細資訊，請參閱 [使用選項群組](#) 中引擎的文件。

在兩種情況下必須立即套用選項群組變更：

- 當您要新增一個新增或更新連接埠值的選項時 (例如 OEM 選項)。
- 使用包含連接埠值的選項新增或刪除選項群組時。

在這類情況下，請在主控台選擇 `Apply Immediately` (立即套用) 選項。或是您可以在使用 `--apply-immediately` 時納入 AWS CLI 選項，或在使用 RDS API 時將 `ApplyImmediately` 參數設定為 `true`。不包含連接埠值的選項可以立即套用，或在資料庫執行個體的下一次維護時段套用。

Note

如果您指定安全群組，以作為選項群組中的選項值，您可以修改選項群組來管理安全群組。您無法藉由修改資料庫執行個體來變更或移除此安全群組。此外，安全群組不會出現在 AWS CLI 命令的輸出 AWS Management Console 或資料庫執行個體詳細資料中 `describe-db-instances`。

主控台

您可以使用 AWS Management Console 修改選項設定。

使用主控台修改選項設定

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您要修改選項的選項群組，然後選擇 Modify option (修改選項)。
4. 在 Modify option (修改選項) 視窗的 Installed Options (已安裝選項) 中，選擇想要修改其設定的選項。依照您想要的進行變更。
5. 若要在新增選項時隨即啟用選項，請在 Apply Immediately (立即套用) 選擇 Yes (是)。如果您選擇了 No (不) (預設)，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。
6. 當您滿意設定後，選擇 Modify Option (修改選項)。

AWS CLI

若要修改選項設定，請將指 AWS CLI [add-option-to-option-group](#) 令與您要修改的選項群組和選項搭配使用。依預設，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。若要將變更立即套用到所有關聯的資料庫執行個體，請將 `--apply-immediately` 參數納入。若要修改選項設定，請用 `--settings` 引數。

Example

以下範例會修改 Oracle Enterprise Manager 資料庫控制 (OEM) 在名為 `testoptiongroup` 的選項群組中使用的連接埠並立即套用變更。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name testoptiongroup \  
  --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name testoptiongroup ^  
  --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^  
  --apply-immediately
```

命令結果類似以下所示：

```
OPTIONSGROUP   False oracle-ee 19 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
  Test Option Group   testoptiongroup
OPTIONS Oracle 12c EM Express OEM      False   False   5432
DBSECURITYGROUPMEMBERSHIPS default authorized
```

Example

下列範例會修改 Oracle 選項 NATIVE_NETWORK_ENCRYPTION 並變更選項設定。

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group \
  --option-group-name testoptiongroup \
  --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES,RC4_256"}], "OptionName":"NA
\
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^
  --option-group-name testoptiongroup ^
  --options "OptionSettings"=[{"Name":"SQLNET.ENCRYPTION_SERVER","Value"]="REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value"]="AES256\,AES192\,DES
\,RC4_256"}], "OptionName"="NATIVE_NETWORK_ENCRYPTION" ^
  --apply-immediately
```

命令結果類似以下所示：

```
OPTIONSGROUP   False oracle-ee 19 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
  Test Option Group   testoptiongroup
OPTIONS Oracle Advanced Security - Native Network Encryption
  NATIVE_NETWORK_ENCRYPTION      False   False
OPTIONSETTINGS
RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40 STATIC
STRING
  RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
Specifies list of encryption algorithms in order of intended use
```

True	True	SQLNET.ENCRYPTION_TYPES_SERVER	AES256,AES192,DES,RC4_256
OPTIONSETTINGS	ACCEPTED,REJECTED,REQUESTED,REQUIRED	STATIC	STRING REQUESTED
Specifies the desired encryption behavior			
False	True	SQLNET.ENCRYPTION_SERVER	REQUIRED
OPTIONSETTINGS	SHA1,MD5	STATIC	STRING SHA1,MD5
Specifies list of checksumming algorithms in order of intended use			
True	True	SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA1,MD5
OPTIONSETTINGS	ACCEPTED,REJECTED,REQUESTED,REQUIRED	STATIC	STRING
REQUESTED	Specifies the desired data integrity behavior		False True
SQLNET.CRYPTO_CHECKSUM_SERVER	REQUESTED		

RDS API

若要修改選項設定，請搭配您想修改的選項群組和選項使用 Amazon RDS [ModifyOptionGroup](#) API。依預設，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。若要將變更立即套用到所有關聯的資料庫執行個體，請將 `ApplyImmediately` 參數納入並將其設為 `true`。

從選項群組移除選項

有些選項可從選項群組中移除，但有些不行。像是持續選項便無法從選項群組中移除，除非與該選項群組關聯的所有資料庫執行個體取消關聯。永久選項則絕不可能從選項群組中移除。如需可移除選項的詳細資訊，請參閱在 [使用選項群組](#) 中列出的特定引擎文件。

如果移除選項群組中的所有選項，Amazon RDS 也不會刪除選項群組。與空白選項群組相關聯的資料庫執行個體會繼續與其關聯；只不過沒有任何作用中的選項。或者，若要從資料庫執行個體移除所有選項，您可將資料庫執行個體與預設 (空白) 選項群組關聯。

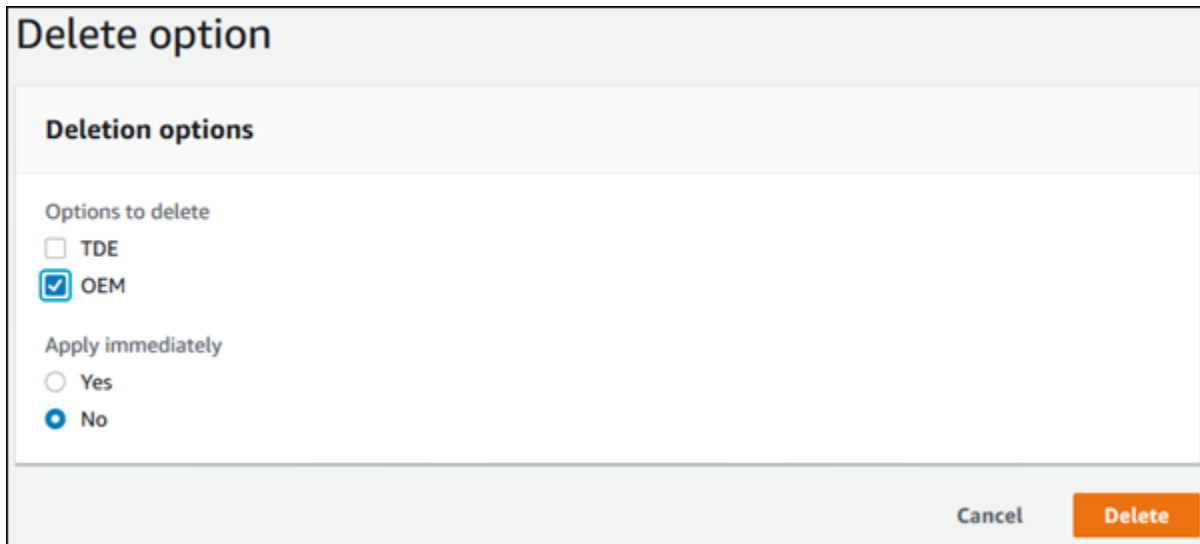
主控台

您可以使 AWS Management Console 用從選項群組中移除選項。

使用主控台從選項群組移除選項

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您要移除選項的選項群組，然後選擇 Delete option (刪除選項)。
4. 在 Delete option (刪除選項) 視窗中，執行下列作業：
 - 選擇您要刪除的選項的核取方塊。

- 若要在刪除後立即生效，請在 Apply immediately (立即套用) 中選擇 Yes (是)。如果您選擇了 No (不) (預設)，選項將會在下一個維護時段為每個關聯的資料庫執行個體刪除。



Delete option

Deletion options

Options to delete

TDE

OEM

Apply immediately

Yes

No

Cancel Delete

5. 當您滿意設定後，請選擇 Yes, Delete (是，刪除)。

AWS CLI

若要從選項群組中移除選項，請使用 AWS CLI [remove-option-from-option-group](#) 指令搭配您要刪除的選項。依預設，選項將會在下一個維護時段從每個關聯的資料庫執行個體中移除。若要立即套用變更，請納入 `--apply-immediately` 參數。

Example

以下範例會將 Oracle Enterprise Manager 資料庫控制 (OEM) 選項從名為 `testoptiongroup` 的選項群組中移除並立即套用變更。

對於 Linux/macOS、或 Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name testoptiongroup \  
  --options OEM \  
  --apply-immediately
```

在 Windows 中：


```
aws rds remove-option-from-option-group ^
  --option-group-name testoptiongroup ^
  --options OEM ^
  --apply-immediately
```

命令結果類似以下所示：

```
OPTIONGROUP    testoptiongroup oracle-ee    19    Test option group
```

RDS API

若要從選項群組移除選項，請使用 Amazon RDS API [ModifyOptionGroup](#) 動作。依預設，選項將會在下一個維護時段從每個關聯的資料庫執行個體中移除。若要立即套用變更，請納入 `ApplyImmediately` 參數並將其設為 `true`。

包含以下參數：

- `OptionGroupName`
- `OptionsToRemove.OptionName`

刪除選項群組

只有在選項群組符合下列條件時，您才可以刪除該選項群組：

- 它不與任何 Amazon RDS 資源相關聯。選項群組可與資料庫執行個體、手動資料庫快照，或是自動資料庫快照建立關聯。
- 它不是預設選項群組。

若要識別資料庫執行個體和資料庫快照所使用的選項群組，您可以使用下列 CLI 命令：

```
aws rds describe-db-instances \
  --query 'DBInstances[*].
[DBInstanceIdentifier,OptionGroupMemberships[].OptionGroupName]'
```

```
aws rds describe-db-snapshots | jq -r '.DBSnapshots[] | "\(.DBInstanceIdentifier),
\(.OptionGroupName)"' | sort | uniq
```

若您嘗試刪除與 RDS 資源建立相關聯的選項群組，就會傳回如下的錯誤。

```
An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

找出與選項群組具有關聯的 Amazon RDS 資源

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇選項群組名稱以顯示其詳細資訊。
4. 查看 Associated Instances and Snapshots (關聯執行個體及快照) 區段，找出具有關聯的 Amazon RDS 資源。

若資料庫執行個體與選項群組具有關聯，請修改資料庫執行個體使用不同的選項群組。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

如果手動資料庫快照與選項群組有關聯，請修改資料庫快照，使用不同的選項群組。您可以使用 AWS CLI [modify-db-snapshot](#) 命令執行此操作。

Note

您無法修改自動資料庫快照的選項群組。

主控台

刪除選項群組的方法之一是使用 AWS Management Console。

使用主控台刪除選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇選項群組。

4. 選擇 Delete group (刪除群組)。
5. 請在確認頁面選擇 Delete (刪除) 以完成刪除選項群組，或選擇 Cancel (取消) 以取消刪除。

AWS CLI

若要刪除選項群組，請搭配下列必要參數使用 AWS CLI [delete-option-group](#) 指令。

- `--option-group-name`

Example

下列範例會刪除名為 `testoptiongroup` 的選項群組。

對於LinuxmacOS、或Unix：

```
aws rds delete-option-group \  
  --option-group-name testoptiongroup
```

在 Windows 中：

```
aws rds delete-option-group ^  
  --option-group-name testoptiongroup
```

RDS API

如要刪除選項群組，請呼叫 Amazon RDS API [DeleteOptionGroup](#) 操作。請納入下列參數：

- `OptionGroupName`

使用參數群組

資料庫參數指定資料庫的配置方式。例如，資料庫參數可以指定要配置給資料庫的資源量 (例如記憶體)。

您可利用參數群組建立資料庫執行個體和多可用區域資料庫叢集的關聯性，以此管理資料庫組態。Amazon RDS 使用預設設定來定義參數群組。您也可以使用自訂設定，定義自己的參數群組。

Note

某些資料庫引擎提供附加功能，您可將其作為選項群組中的選項新增至您的資料庫。如需選項群組的詳細資訊，請參閱 [使用選項群組](#)。

主題

- [參數群組概觀](#)
- [在資料庫執行個體中使用資料庫參數群組](#)
- [使用多可用區域資料庫叢集的資料庫叢集參數群組](#)
- [比較資料庫參數群組](#)
- [指定資料庫參數](#)

參數群組概觀

資料庫參數群組扮演引擎組態值的容器，以套用至一或多個資料庫執行個體。

資料庫叢集參數群組僅適用於多可用區域資料庫叢集。在多可用區域資料庫叢集中，資料庫叢集參數群組中的設定會套用至叢集中的所有資料庫執行個體。資料庫引擎和資料庫引擎版本的預設資料庫參數群組會用於資料庫叢集中的每個資料庫執行個體。

主題

- [預設和自訂參數群組](#)
- [靜態和動態資料庫執行個體參數](#)
- [靜態和動態資料庫叢集參數](#)
- [字元集參數](#)
- [支援的參數和參數值](#)

預設和自訂參數群組

如果建立的資料庫執行個體未指定資料庫參數群組，則會使用預設的資料庫參數群組。同樣地，如果您建立多可用區域資料庫叢集，而未指定資料庫叢集參數群組，資料庫叢集將使用預設的資料庫叢集參數群組。每個預設的參數群組將依引擎、運算等級和執行個體分配儲存包含資料庫引擎預設值和 Amazon RDS 系統預設值。

您無法修改預設參數群組的參數設定。相反地，您可以執行下列作業：

1. 建立新的參數群組。
2. 變更所需參數的設定。並非參數群組中的所有資料庫引擎參數都有資格進行修改。
3. 修改資料庫執行個體或資料庫叢集，以建立新參數群組的關聯。

將新的資料庫參數群組與資料庫執行個體建立關聯時，關聯會立即發生。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。如需修改多可用區域資料庫叢集的詳細資訊，請參閱[修改多可用區域資料庫叢集](#)。

Note

如果您已將資料庫執行個體修改為使用自訂參數群組，並啟動資料庫執行個體，RDS 會在啟動過程中自動重新啟動資料庫執行個體。

RDS 只會在資料庫執行個體重新啟動後，才會在新關聯的參數群組中套用修改後的靜態和動態參數。不過，如果您在將資料庫參數群組與資料庫執行個體建立關聯之後修改該群組中的動態參數，則會立即套用這些變更，而不需重新開機。如需變更資料庫叢集參數群組的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

如果您在資料庫參數群組內更新參數，變更會應用到所有和參數群組建立連結的資料庫執行個體。同樣地，如果您在多可用區域資料庫叢集參數群組內更新參數，變更會套用到所有與資料庫叢集參數群組相關聯的 Aurora 資料庫叢集。

如果您不想從頭開始建立參數群組，可以使用 AWS CLI [copy-db-parameter-group](#) 指令或 [copy-db-cluster-parameter-group](#) 指令複製現有的參數群組。在某些情況下，您可能會發現複製參數群組很有用。例如，您可能想要將其大部份現有資料庫參數群組的自訂參數及值併入新的資料庫參數群組。

靜態和動態資料庫執行個體參數

資料庫執行個體參數可為靜態或動態。它們不同之處如下：

- 當您變更靜態參數並儲存資料庫參數群組時，參數變更會在您手動重新啟動相關聯的資料庫執行個體後生效。若為靜態參數，主控台一律將 `pending-reboot` 用於 `ApplyMethod`。
- 變更動態參數時，參數變更預設會立即生效，無需重新啟動。當您使用變更 AWS Management Console 資料庫執行個體參數值時，它一律會用 `immediateApplyMethod` 於動態參數。若要將參數變更延遲到重新啟動相關聯的資料庫執行個體之後，請使用 AWS CLI 或 RDS API。將 `ApplyMethod` 設定為 `pending-reboot` 以進行參數變更。

Note

在適 `pending-reboot` 用於 SQL 伺服器資料庫執行個體的 RDS 上的 AWS CLI 或 RDS API 中使用動態參數會產生錯誤。在 RDS for SQL Server 上使用 `apply-immediately`。

如需使用變更參數值的 AWS CLI 詳細資訊，請參閱 [modify-db-parameter-group](#)。如需有關使用 RDS API 變更參數值的詳細資訊，請參閱 [修改ParameterGroup](#) 資料庫。

如果資料庫執行個體未在其相關聯的資料庫參數群組中使用最新的變更，主控台會將資料庫參數群組的狀態顯示為 `pending-reboot`。此狀態不會在下一個維護時段期間造成自動重新啟動。如欲對該資料庫執行個體套用最新參數變更，請手動重新啟動資料庫執行個體。

靜態和動態資料庫叢集參數

資料庫叢集參數可為靜態或動態。它們不同之處如下：

- 當您變更靜態參數並儲存資料庫叢集參數群組時，參數變更會在您手動重新啟動相關聯的資料庫叢集後生效。若為靜態參數，主控台一律將 `pending-reboot` 用於 `ApplyMethod`。
- 變更動態參數時，參數變更預設會立即生效，無需重新啟動。當您使用變更 AWS Management Console 更資料庫叢集參數值時，它一律會用 `immediateApplyMethod` 於動態參數。若要將參數變更延遲到相關聯的資料庫叢集重新開機之後，請使用 AWS CLI 或 RDS API。將 `ApplyMethod` 設定為 `pending-reboot` 以進行參數變更。

如需有關使用變更參數值 AWS CLI 的詳細資訊，請參閱 [modify-db-cluster-parameter-group](#)。如需有關使用 RDS API 變更參數值的詳細資訊，請參閱 [修改ClusterParameterGroup](#) 資料庫。

字元集參數

在建立資料庫執行個體或多可用區域資料庫叢集之前，先在參數群組中設定與字元集或資料庫定序相關的任何參數。也會在其中建立資料庫之前這樣做。透過此方式，您可以確保預設資料庫和新資料庫皆使用您指定的字元集和定序值。如果您變更字元集和定序參數，參數變更將不會套用到現有資料庫。

針對部分資料庫引擎，您可以使用 ALTER DATABASE 命令變更現有資料庫的字元集或定序值，例如：

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

如需變更資料庫的字元集或定序值的詳細資訊，請參閱適用於您資料庫引擎的文件。

支援的參數和參數值

若要判斷資料庫引擎的支援參數，您可以檢視資料庫執行個體或資料庫叢集所使用的資料庫參數群組和資料庫叢集參數群組中的參數。如需詳細資訊，請參閱 [檢視資料庫參數群組的參數值](#) 及 [檢視資料庫叢集參數群組的參數值](#)。

在許多情況下，您可使用運算式、公式和函數來指定整數及布林值參數。函數可包含數學的對數表達式。但是，並非所有參數都支援參數值的表達式、公式和函數。如需詳細資訊，請參閱 [指定資料庫參數](#)。

未正確設定參數群組中的參數，可能產生各種意外影響，包括效能降低和系統不穩定。修改資料庫參數時請務必謹慎，在修改參數群組之前，請備份資料。在將這些參數群組變更套用到生產資料庫執行個體或資料庫叢集之前，請在測試資料庫執行個體或資料庫叢集上嘗試參數群組設定變更。

在資料庫執行個體中使用資料庫參數群組

資料庫執行個體會使用資料庫參數群組。以下各節介紹資料庫執行個體參數群組的設定和管理。

主題

- [建立資料庫參數群組](#)
- [將資料庫參數群組與資料庫執行個體建立關聯](#)
- [修改資料庫參數群組中的參數](#)
- [將資料庫參數群組中的參數重設為其預設值](#)
- [複製資料庫參數群組](#)
- [列出資料庫參數群組](#)

- [檢視資料庫參數群組的參數值](#)
- [刪除資料庫參數群組](#)

建立資料庫參數群組

您可以使用 AWS Management Console、或 RDS API 建立新的資料庫參數群組。AWS CLI

下列限制適用於資料庫參數群組名稱：

- 名稱必須為 1 到 255 個字母、數字或連字號。

預設參數群組名稱可以包括句點，例如 `default.mysql8.0`。不過，自訂參數群組名稱不能包括句點。

- 第一個字元必須是字母。
- 名稱不能以連字號結尾或包含兩個連續連字號。

主控台

建立資料庫參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在「參數群組名稱」中，輸入新資料庫參數群組的名稱。
5. 在說明中，輸入新資料庫參數群組的說明。
6. 針對「引擎類型」，選擇您的資料庫引擎。
7. 對於「參數群組族群」，請選擇資料庫參數群組族群。
8. 對於「類型」，選擇「資料庫參數群組」。
9. 選擇建立。

AWS CLI

若要建立資料庫參數群組，請使用 AWS CLI [create-db-parameter-group](#) 指令。下列範例將為 MySQL 8.0 版建立一個名為 `mydbparametergroup` 的資料庫參數群組，其說明為「My new parameter group (我的新參數群組)」。

包含下列必要參數：

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

若要列出所有可用的參數群組系列，請使用下列命令：

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

Note

輸出包含重覆項目。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL8.0 \  
  --description "My new parameter group"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name mydbparametergroup ^  
  --db-parameter-group-family MySQL8.0 ^  
  --description "My new parameter group"
```

此命令會產生類似下列的輸出：

```
DBPARAMETERGROUP mydbparametergroup mysql8.0 My new parameter group
```

RDS API

若要建立資料庫參數群組，請使用 RDS API [CreateDBParameterGroup](#) 操作。

包含下列必要參數：

- DBParameterGroupName
- DBParameterGroupFamily
- Description

將資料庫參數群組與資料庫執行個體建立關聯

您可以使用自訂設定，建立自己的資料庫參數群組。您可以使用 AWS Management Console、或 RDS API 將資料庫參數群組與資料庫執行個體建立關聯。AWS CLI您可以在建立或修改資料庫執行個體時執行此動作。

如需建立資料庫參數群組的資訊，請參閱[建立資料庫參數群組](#)。如需建立資料庫執行個體的相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

Note

當您建立新資料庫參數群組與資料庫執行個體的關聯時，只有在資料庫執行個體重新開機之後，才會套用修改過的靜態參數和動態參數。不過，如果您在將資料庫參數群組與資料庫執行個體建立關聯之後修改該群組中的動態參數，則會立即套用這些變更，而不需重新開機。

主控台

將資料庫參數群組與資料庫執行個體建立關聯

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 變更 DB parameter group (資料庫參數群組) 設定。
5. 選擇 Continue (繼續)，並檢查修改的摘要。
6. (選用) 選擇 Apply immediately (立即套用) 以立即套用變更。在某些情況下，選擇此選項會導致停機。如需詳細資訊，請參閱 [排程修改設定](#)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB instance (修改資料庫執行個體) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要將資料庫參數群組與資料庫執行個體建立關聯，請使用 AWS CLI [modify-db-instance](#) 指令搭配下列選項：

- `--db-instance-identifier`
- `--db-parameter-group-name`

下面的範例將 `mydbpg` 資料庫參數群組與 `database-1` 資料庫執行個體建立關聯。透過 `--apply-immediately`，即可立即套用變更。使用 `--no-apply-immediately` 在下次維護時段套用變更。如需詳細資訊，請參閱[排程修改設定](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier database-1 \  
  --db-parameter-group-name mydbpg \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier database-1 ^  
  --db-parameter-group-name mydbpg ^  
  --apply-immediately
```

RDS API

若要將資料庫參數群組與資料庫執行個體建立關聯，請使用 RDS API [ModifyDBInstance](#) 操作搭配下列參數：

- `DBInstanceName`
- `DBParameterGroupName`

修改資料庫參數群組中的參數

您可修改客戶建立的資料庫參數群組中的參數值；但無法變更預設資料庫參數群組中的參數值。客戶建立的資料庫參數群組中的參數變更會套用到與該資料庫參數群組關聯的所有資料庫執行個體。

某些參數的變更會立即套用到資料庫執行個體，而不需要重新啟動。其他參數的變更只有在資料庫執行個體重新啟動之後才會套用。RDS 主控台會在 Configuration (組態) 標籤上，顯示與資料庫執行個體相關聯的資料庫參數群組的狀態。例如，假設資料庫執行個體未使用其相關聯之資料庫參數群組的最新變更。若是如此，RDS 主控台會顯示狀態為 pending-reboot 的資料庫參數群組。如欲對該資料庫執行個體套用最新參數變更，請手動重新啟動資料庫執行個體。

The screenshot shows the Amazon RDS console interface. At the top, there are navigation tabs: Connectivity & security, Monitoring, Logs & events, Configuration (highlighted with a red box), Maintenance & backups, and Tags. Below the tabs, the 'Instance' section is visible. The 'Configuration' tab is selected, displaying various instance details. The 'Parameter group' field is highlighted with a red box, showing 'test-sqlserver-se-2017 (pending-reboot)'. Other details include DB instance id (database-2), Engine version (14.00.3281.6.v1), DB name (-), License model (License Included), Collation (SQL_Latin1_General_CP1_CI_AS), Option groups (test-se-2017), ARN (arn:aws:rds:us-west-2: [redacted]:db:database-2), Resource id (db-[redacted]), Created time (Wed Dec 04 2019 14:22:38 GMT-0500 (Eastern Standard Time)), Deletion protection (Disabled), Instance class (db.r4.large), vCPU (2), RAM (15.25 GB), Master username (admin), IAM db authentication (Not Enabled), Multi AZ (Yes (Mirroring)), and Secondary Zone (us-west-2d).

Configuration	Instance class
DB instance id database-2	Instance class db.r4.large
Engine version 14.00.3281.6.v1	vCPU 2
DB name -	RAM 15.25 GB
License model License Included	Availability
Collation SQL_Latin1_General_CP1_CI_AS	Master username admin
Option groups test-se-2017	IAM db authentication Not Enabled
ARN arn:aws:rds:us-west-2-[redacted]:db:database-2	Multi AZ Yes (Mirroring)
Resource id db-[redacted]	Secondary Zone us-west-2d
Created time Wed Dec 04 2019 14:22:38 GMT-0500 (Eastern Standard Time)	
Parameter group test-sqlserver-se-2017 (pending-reboot)	
Deletion protection Disabled	

主控台

若要修改資料庫參數群組中的參數

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇要修改的參數群組名稱。
4. 針對 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。
5. 變更您要修改的參數值。您可用對話方塊右上角的箭頭鍵來捲動參數。

您無法變更預設參數群組中的值。

6. 選擇儲存變更。

AWS CLI

若要修改資料庫參數群組，請搭配下列必要選項使用 AWS CLI [modify-db-parameter-group](#) 指令：

- `--db-parameter-group-name`
- `--parameters`

以下範例將修改名為 `mydbparametergroup` 的資料庫參數群組的 `max_connections` 和 `max_allowed_packet` 值。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --parameters  
  "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \  
  
  "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name mydbparametergroup ^
  --parameters
  "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^
  "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

該命令會產生類似以下的輸出：

```
DBPARAMETERGROUP mydbparametergroup
```

RDS API

若要修改資料庫參數群組，請使用 RDS API [ModifyDBParameterGroup](#) 操作，並搭配下列必要參數：

- DBParameterGroupName
- Parameters

將資料庫參數群組中的參數重設為其預設值

您可以將客戶建立的資料庫參數群組中的參數值重設為其預設值。客戶建立的資料庫參數群組中的參數變更會套用到與該資料庫參數群組關聯的所有資料庫執行個體。

使用主控台時，您可以將特定參數重設為其預設值。不過，您無法輕易地一次重設資料庫參數群組中的所有參數。使用 AWS CLI 或 RDS API 時，您可以將特定參數重設為其預設值。您也可以一次重設資料庫參數群組中的所有參數。

某些參數的變更會立即套用到資料庫執行個體，而不需要重新啟動。其他參數的變更只有在資料庫執行個體重新啟動之後才會套用。RDS 主控台會在 Configuration (組態) 標籤上，顯示與資料庫執行個體相關聯的資料庫參數群組的狀態。例如，假設資料庫執行個體未使用其相關聯之資料庫參數群組的最新變更。若是如此，RDS 主控台會顯示狀態為 pending-reboot 的資料庫參數群組。如欲對該資料庫執行個體套用最新參數變更，請手動重新啟動資料庫執行個體。

Connectivity & security | Monitoring | Logs & events | **Configuration** | Maintenance & backups | Tags

Instance

Configuration	Instance class
DB instance id database-2	Instance class db.r4.large
Engine version 14.00.3281.6.v1	vCPU 2
DB name -	RAM 15.25 GB
License model License Included	Availability
Collation SQL_Latin1_General_CP1_CI_AS	Master username admin
Option groups test-se-2017	IAM db authentication Not Enabled
ARN arn:aws:rds:us-west- XXXXXXXXXX :db:database-2	Multi AZ Yes (Mirroring)
Resource id db- XXXXXXXXXX	Secondary Zone us-west-2d
Created time Wed Dec 04 2019 14:22:38 GMT-0500 (Eastern Standard Time)	
Parameter group test-sqlserver-se-2017 (pending-reboot)	
Deletion protection Disabled	

Note

在預設資料庫參數群組中，參數始終設定為其預設值。

主控台

將資料庫參數群組中的參數重設為其預設值

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇參數群組。
4. 針對 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。
5. 選擇您要重設為預設值的參數。您可用對話方塊右上角的箭頭鍵來捲動參數。

您無法重設預設參數群組中的值。

6. 選擇「Reset (重設)」，然後選擇「Reset parameters (重設參數)」來確認。

AWS CLI

若要重設資料庫參數群組中的部分或全部參數，請使用具有下列必要選項的 AWS CLI [reset-db-parameter-group](#) 命令：`--db-parameter-group-name`

若要重設資料庫參數群組中的所有參數，請指定 `--reset-all-parameters` 選項。若要重設特定參數，請指定 `--parameters` 選項。

下列範例會將資料庫參數群組中所有名為 `mydbparametergroup` 的參數重設為其預設值。

Example

對於LinuxmacOS、或Unix：

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --reset-all-parameters
```

在 Windows 中：

```
aws rds reset-db-parameter-group ^  
  --db-parameter-group-name mydbparametergroup ^  
  --reset-all-parameters
```


下列範例會在名為 `mydbparametergroup` 的資料庫參數群組中，將 `max_connections` 與 `max_allowed_packet` 選項重設為其預設值。

Example

對於Linux/macOS、或Unix：

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds reset-db-parameter-group ^  
  --db-parameter-group-name mydbparametergroup ^  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

該命令會產生類似以下的輸出：

```
DBParameterGroupName mydbparametergroup
```

RDS API

若要將資料庫參數群組中的參數重設為其預設值，請使用 RDS API [ResetDBParameterGroup](#) 命令與下列必要參數：DBParameterGroupName。

若要重設資料庫參數群組中的所有參數，請將 `ResetAllParameters` 參數設定為 `true`。若要重設特定參數，請指定 `Parameters` 參數。

複製資料庫參數群組

您可複製所建立的自訂資料庫參數群組。複製參數群組可以是便利的解決方案。範例是在您已建立資料庫參數群組，並希望將其大部份的自訂參數及值併入新的資料庫參數群組時。您可以使用複製資料庫參數群組 AWS Management Console。您也可以使用命 AWS CLI [copy-db-parameter-group](#) 令或 RDS API [複製資料庫ParameterGroup](#) 作業。

複製資料庫參數群組後，請等待至少 5 分鐘，然後再建立第一個使用該資料庫參數群組做為預設參數群組的資料庫執行個體。執行此動作允許 Amazon RDS 在使用參數群組前，完整完成複製動作。這對

建立資料庫執行個體預設資料庫時的關鍵參數尤其重要。例如 `character_set_database` 參數定義的預設資料庫字元集。使用 [Amazon RDS 主控台](#) 的「參數群組」選項或命 `describe-db-parameters` 令來確認資料庫參數群組已建立。

Note

您無法複製預設參數群組。但您可以依照預設參數群組建立新的參數群組。
您無法將資料庫參數群組複製到不同的 AWS 帳戶 或 AWS 區域。

主控台

複製資料庫參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇您要複製的自訂參數群組。
4. 針對 Parameter group actions (參數群組動作)，選擇 Copy (複製)。
5. 在 New DB parameter group identifier (新資料庫參數群組識別符) 中，輸入新參數群組的名稱。
6. 在 Description (描述) 中，輸入新參數群組的描述。
7. 請選擇 Copy (複製)。

AWS CLI

若要複製資料庫參數群組，請搭配下列必要選項使用 AWS CLI [copy-db-parameter-group](#) 指令：

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

以下範例會建立名為 `mygroup2` 的新資料庫參數群組，該群組為資料庫參數群組 `mygroup1` 的複本。

Example

對於 Linux/macOS、或 Unix：

```
aws rds copy-db-parameter-group \  
  --source-db-parameter-group-identifier mygroup1 \  
  --target-db-parameter-group-identifier mygroup2 \  
  --target-db-parameter-group-description "DB parameter group 2"
```

在 Windows 中：

```
aws rds copy-db-parameter-group ^  
  --source-db-parameter-group-identifier mygroup1 ^  
  --target-db-parameter-group-identifier mygroup2 ^  
  --target-db-parameter-group-description "DB parameter group 2"
```

RDS API

若要複製資料庫參數群組，請使用 RDS API [CopyDBParameterGroup](#) 參數並搭配下列必要參數：

- SourceDBParameterGroupIdentifier
- TargetDBParameterGroupIdentifier
- TargetDBParameterGroupDescription

列出資料庫參數群組

您可以列出為您的 AWS 帳戶建立的資料庫參數群組。

Note

當您為特定資料庫引擎和版本建立資料庫執行個體時，會從預設的參數範本自動建立預設的參數群組。這些預設參數群組包含偏好的參數設定，無法修改。當您建立自訂參數群組時，您可以修改參數設定。

主控台

列出 AWS 帳戶的所有資料庫參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

資料庫參數群組隨即會出現在清單中。

AWS CLI

若要列出 AWS 帳戶的所有資料庫參數群組，請使用 AWS CLI [describe-db-parameter-groups](#) 指令。

Example

以下範例列出 AWS 帳戶的所有可用資料庫參數群組。

```
aws rds describe-db-parameter-groups
```

此命令會傳回類似以下的回應：

```
DBPARAMETERGROUP  default.mysql8.0    mysql8.0  Default parameter group for MySQL8.0
DBPARAMETERGROUP  mydbparametergroup mysql8.0  My new parameter group
```

以下範例說明 mydbparamgroup1 參數群組。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-parameter-groups \
  --db-parameter-group-name mydbparamgroup1
```

在 Windows 中：

```
aws rds describe-db-parameter-groups ^
  --db-parameter-group-name mydbparamgroup1
```

此命令會傳回類似以下的回應：

```
DBPARAMETERGROUP  mydbparametergroup1  mysql8.0  My new parameter group
```

RDS API

若要列出 AWS 帳戶的所有資料庫參數群組，請使用 RDS API [DescribeDBParameterGroups](#) 作業。

檢視資料庫參數群組的參數值

您可以從資料庫參數群組取得所有參數與其值的清單。

主控台

檢視資料庫參數群組的參數值

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

資料庫參數群組隨即會出現在清單中。

3. 選擇參數群組的名稱，以查看其參數清單。

AWS CLI

若要檢視資料庫參數群組的參數值，請使用具有下列必要參數的 AWS CLI [describe-db-parameters](#) 命令。

- `--db-parameter-group-name`

Example

以下範例列出資料庫參數群組 `mydbparametergroup` 的參數和參數值。

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

此命令會傳回類似以下的回應：

DBPARAMETER	Parameter Name	Parameter Value	Source	Data Type
Apply Type	Is Modifiable			
DBPARAMETER	allow-suspicious-udfs		engine-default	boolean
static	false			
DBPARAMETER	auto_increment_increment		engine-default	integer
dynamic	true			
DBPARAMETER	auto_increment_offset		engine-default	integer
dynamic	true			
DBPARAMETER	binlog_cache_size	32768	system	integer
dynamic	true			

DBPARAMETER	socket	/tmp/mysql.sock	system	string
static	false			

RDS API

若要檢視資料庫參數群組的參數值，請搭配下列必要參數使用 RDS API [DescribeDBParameters](#) 命令。

- DBParameterGroupName

刪除資料庫參數群組

您可以使用 AWS Management Console、AWS CLI 或 RDS API 刪除資料庫參數群組。只有當參數群組與資料庫執行個體沒有關聯時，才有資格刪除參數群組。

主控台

刪除資料庫參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

資料庫參數群組隨即會出現在清單中。

3. 選擇要刪除的參數群組名稱。
4. 選擇動作，然後選擇刪除。
5. 檢閱參數群組名稱，然後選擇「刪除」。

AWS CLI

若要刪除資料庫參數群組，請使用具有下列必要參數的 AWS CLI [delete-db-parameter-group](#) 命令。

- --db-parameter-group-name

Example

下列範例會刪除名為 mydb 參數群組/參數群組的資料庫參數群組。

```
aws rds delete-db-parameter-group --db-parameter-group-name mydbparametergroup
```

RDS API

若要刪除資料庫參數群組，請使用 RDS API [DeleteDBParameterGroup](#) 命令搭配下列必要參數。

- DBParameterGroupName

使用多可用區域資料庫叢集的資料庫叢集參數群組

多可用區域資料庫叢集使用資料庫叢集參數群組。以下各節介紹資料庫叢集參數群組的設定和管理。

主題

- [建立資料庫叢集參數群組](#)
- [修改資料庫叢集參數群組中的參數](#)
- [重設資料庫叢集參數群組中的參數](#)
- [複製資料庫叢集參數群組](#)
- [列出資料庫叢集參數群組](#)
- [檢視資料庫叢集參數群組的參數值](#)
- [刪除資料庫叢集參數群組](#)

建立資料庫叢集參數群組

您可以使用 AWS Management Console、或 RDS API 建立新的資料庫叢集參數群組。AWS CLI

建立資料庫叢集參數群組後，請等待至少 5 分鐘，然後再建立使用該資料庫叢集參數群組的資料庫叢集。執行此動作允許 Amazon RDS 完整建立參數群組，再將其供新資料庫叢集使用。您可以使用 [Amazon RDS 主控台](#) 的 Parameter groups (參數群組) 頁面或 [describe-db-cluster-parameters](#) 命令確認是否已建立資料庫叢集參數群組。

下列限制適用於資料庫叢集參數群組名稱：

- 名稱必須為 1 到 255 個字母、數字或連字號。

預設參數群組名稱可以包括句點，例如 `default.mysql5.7`。不過，自訂參數群組名稱不能包括句點。

- 第一個字元必須是字母。
- 名稱不能以連字號結尾或包含兩個連續連字號。

主控台

建立資料庫叢集參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。

Create parameter group (建立參數群組) 視窗隨即出現。

4. 在 Parameter group family (參數群組系列) 清單中，選擇資料庫參數群組系列
5. 在 [類型] 清單中，選取 [資料庫叢集參數群組]。
6. 在 Group name (群組名稱) 方塊中輸入新資料庫叢集參數群組的名稱。
7. 在 Description (描述) 方塊中輸入新資料庫叢集參數群組的描述。
8. 選擇建立。

AWS CLI

若要建立資料庫叢集參數群組，請使用 AWS CLI [create-db-cluster-parameter-group](#) 指令。

以下範例將為 RDS for MySQL 8.0 版建立名為 mydbclusterparametergroup，描述為 My new cluster parameter group (我的新叢集參數群組) 的資料庫叢集參數群組。

包含下列必要參數：

- `--db-cluster-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

若要列出所有可用的參數群組系列，請使用下列命令：

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```


Note

輸出包含重覆項目。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup \  
  --db-parameter-group-family mysql8.0 \  
  --description "My new cluster parameter group"
```

在 Windows 中：

```
aws rds create-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name mydbclusterparametergroup ^  
  --db-parameter-group-family mysql8.0 ^  
  --description "My new cluster parameter group"
```

此命令會產生類似下列的輸出：

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterparametergroup",  
    "DBParameterGroupFamily": "mysql8.0",  
    "Description": "My new cluster parameter group",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterparametergroup2"  
  }  
}
```

RDS API

若要建立資料庫叢集參數群組，請使用 RDS API [CreateDBClusterParameterGroup](#) 動作。

包含下列必要參數：

- `DBClusterParameterGroupName`
- `DBParameterGroupFamily`

- Description

修改資料庫叢集參數群組中的參數

您可以修改客戶建立的資料庫叢集參數群組中參數值。您無法變更預設資料庫叢集參數群組中的參數值。客戶建立的資料庫叢集參數群組中的參數變更會套用到與該資料庫叢集參數群組關聯的所有資料庫叢集。

主控台

修改資料庫叢集參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇您要修改的參數群組。
4. 針對 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。
5. 變更您想要修改的參數值。您可用對話方塊右上角的箭頭鍵來捲動參數。

您無法變更預設參數群組中的值。

6. 選擇儲存變更。
7. 重新啟動叢集，以將變更套用至叢集。

AWS CLI

若要修改資料庫叢集參數群組，請使用具有下列必要參數的 AWS CLI [modify-db-cluster-parameter-group](#) 命令：

- `--db-cluster-parameter-group-name`
- `--parameters`

以下範例將修改名稱為 `mydbclusterparametergroup` 的資料庫叢集參數群組的 `server_audit_logging` 和 `server_audit_logs_upload` 值。

Example

對於 Linux macOS、或 Unix：

```
aws rds modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup \  
  --parameters  
  "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \  
  
  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name mydbclusterparametergroup ^  
  --parameters  
  "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^  
  
  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

該命令會產生類似以下的輸出：

```
DBCLUSTERPARAMETERGROUP mydbclusterparametergroup
```

RDS API

若要修改資料庫叢集參數群組，請使用 RDS API [ModifyDBClusterParameterGroup](#) 命令並搭配下列必要參數：

- DBClusterParameterGroupName
- Parameters

重設資料庫叢集參數群組中的參數

您可以將參數重設為客戶建立的資料庫叢集參數群組中的預設值。客戶建立的資料庫叢集參數群組中的參數變更會套用到與該資料庫叢集參數群組關聯的所有資料庫叢集。

Note

在預設的資料庫叢集參數群組中，參數永遠設定為其預設值。

主控台

將資料庫叢集參數群組中的參數重設為其預設值

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇參數群組。
4. 針對 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。
5. 選擇您要重設為預設值的參數。您可用對話方塊右上角的箭頭鍵來捲動參數。

您無法重設預設參數群組中的值。

6. 選擇「Reset (重設)」，然後選擇「Reset parameters (重設參數)」來確認。
7. 重新啟動。

AWS CLI

若要將資料庫叢集參數群組中的參數重設為預設值，請搭配下列必要選項使用 AWS CLI [reset-db-cluster-parameter-group](#) 命令：`--db-cluster-parameter-group-name`

若要重設資料庫叢集參數群組中的所有參數，請指定 `--reset-all-parameters` 選項。若要重設特定參數，請指定 `--parameters` 選項。

下列範例會將資料庫參數群組中所有名為 `mydbparametergroup` 的參數重設為其預設值。

Example

對於LinuxmacOS、或Unix：

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbparametergroup \  
  --reset-all-parameters
```

在 Windows 中：

```
aws rds reset-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name mydbparametergroup ^  
  --reset-all-parameters
```

以下範例會將名為 `mydbclusterparametergroup` 的資料庫叢集參數群組中的 `server_audit_logging` 和 `server_audit_logs_upload` 重設為其預設值。

Example

對於LinuxmacOS、或Unix：

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup \  
  --parameters "ParameterName=server_audit_logging,ApplyMethod=immediate" \  
  "ParameterName=server_audit_logs_upload,ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds reset-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name mydbclusterparametergroup ^  
  --parameters  
  "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^  
  
  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

該命令會產生類似以下的輸出：

```
DBClusterParameterGroupName mydbclusterparametergroup
```

RDS API

若要將資料庫叢集參數群組中的參數重設為其預設值，請使用 RDS API

[ResetDBClusterParameterGroup](#) 命令與下列必要參數：DBClusterParameterGroupName。

若要重設資料庫叢集參數群組中的所有參數，請將 `ResetAllParameters` 參數設定為 `true`。若要重設特定參數，請指定 `Parameters` 參數。

複製資料庫叢集參數群組

您可複製所建立的自訂資料庫叢集參數群組。如果您已建立資料庫叢集參數群組，並且希望在新資料庫叢集參數群組中包含該組中的大多數自訂參數和值，則複製參數群組是一種方便的解決方案。[您可以使用複製 AWS CLI-db 叢集參數群組命令或 RDS API CopyDB 群組作業來複製資料庫叢集參數群組。](#) [ClusterParameter](#)

複製資料庫叢集參數群組後，請等待至少 5 分鐘，然後再建立使用該資料庫叢集參數群組的資料庫叢集。執行此動作允許 Amazon RDS 完整複製參數群組，再將其供新資料庫叢集使用。您可以使用

[Amazon RDS 主控台](#)的 Parameter groups (參數群組) 頁面或 [describe-db-cluster-parameters](#) 命令確認是否已建立資料庫叢集參數群組。

Note

您無法複製預設參數群組。但您可以依照預設參數群組建立新的參數群組。
您無法將資料庫叢集參數群組複製到不同的 AWS 帳戶 或 AWS 區域。

主控台

複製資料庫叢集參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇您要複製的自訂參數群組。
4. 針對 Parameter group actions (參數群組動作)，選擇 Copy (複製)。
5. 在 New DB parameter group identifier (新資料庫參數群組識別符) 中，輸入新參數群組的名稱。
6. 在 Description (描述) 中，輸入新參數群組的描述。
7. 請選擇 Copy (複製)。

AWS CLI

若要複製資料庫叢集參數群組，請搭配下列必要參數使用 AWS CLI [copy-db-cluster-parameter-group](#) 命令：

- `--source-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-description`

以下範例會建立名為 mygroup2 的新資料庫叢集參數群組，該群組為資料庫叢集參數群組 mygroup1 的複本。

Example

對於LinuxmacOS、或Unix：

```
aws rds copy-db-cluster-parameter-group \  
  --source-db-cluster-parameter-group-identifier mygroup1 \  
  --target-db-cluster-parameter-group-identifier mygroup2 \  
  --target-db-cluster-parameter-group-description "DB parameter group 2"
```

在 Windows 中：

```
aws rds copy-db-cluster-parameter-group ^  
  --source-db-cluster-parameter-group-identifier mygroup1 ^  
  --target-db-cluster-parameter-group-identifier mygroup2 ^  
  --target-db-cluster-parameter-group-description "DB parameter group 2"
```

RDS API

若要複製資料庫叢集參數群組，請使用 RDS API [CopyDBClusterParameterGroup](#) 操作並搭配下列必要參數：

- SourceDBClusterParameterGroupIdentifier
- TargetDBClusterParameterGroupIdentifier
- TargetDBClusterParameterGroupDescription

列出資料庫叢集參數群組

您可以列出為您的 AWS 帳戶建立的資料庫叢集參數群組。

Note

當您為特定資料庫引擎和版本建立資料庫叢集時，會從預設的參數範本自動建立預設的參數群組。這些預設參數群組包含偏好的參數設定，無法修改。當您建立自訂參數群組時，您可以修改參數設定。

主控台

列出 AWS 帳戶的所有資料庫叢集參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

資料庫叢集參數群組會以 DB cluster parameter group (資料庫叢集參數群組) 出現在 Type (類型) 清單中。

AWS CLI

若要列出 AWS 帳戶的所有資料庫叢集參數群組，請使用 AWS CLI [describe-db-cluster-parameter-groups](#) 指令。

Example

以下範例列出 AWS 帳戶的所有可用資料庫叢集參數群組。

```
aws rds describe-db-cluster-parameter-groups
```

以下範例說明 mydbclusterparametergroup 參數群組。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-cluster-parameter-groups \  
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

在 Windows 中：

```
aws rds describe-db-cluster-parameter-groups ^  
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

此命令會傳回類似以下的回應：

```
{  
  "DBClusterParameterGroups": [  
    {  
      "DBClusterParameterGroupName": "mydbclusterparametergroup2",  
      "DBParameterGroupFamily": "mysql8.0",  
      "Description": "My new cluster parameter group",  
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterparametergroup"  
    }  
  ]  
}
```


RDS API

若要列出 AWS 帳戶的所有資料庫叢集參數群組，請使用 RDS API [DescribeDBClusterParameterGroups](#) 動作。

檢視資料庫叢集參數群組的參數值

您可以從資料庫叢集參數群組取得所有參數與其值的清單。

主控台

檢視資料庫叢集參數群組的參數值

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。

資料庫叢集參數群組會以 DB cluster parameter group (資料庫叢集參數群組) 出現在 Type (類型) 清單中。

3. 選擇資料庫叢集參數群組的名稱，以查看參數清單。

AWS CLI

若要檢視資料庫叢集參數群組的參數值，請搭配下列必要參數使用 AWS CLI [describe-db-cluster-parameters](#) 命令。

- `--db-cluster-parameter-group-name`

Example

以下範例列出資料庫叢集參數群組 `mydbclusterparametergroup` 的參數和參數值，為 JSON 格式。

此命令會傳回類似以下的回應：

```
aws rds describe-db-cluster-parameters --db-cluster-parameter-group-name mydbclusterparametergroup
```

```
{  
  "Parameters": [  

```

```

    {
      "ParameterName": "activate_all_roles_on_login",
      "ParameterValue": "0",
      "Description": "Automatically set all granted roles as active after the
user has authenticated successfully.",
      "Source": "engine-default",
      "ApplyType": "dynamic",
      "DataType": "boolean",
      "AllowedValues": "0,1",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot",
      "SupportedEngineModes": [
        "provisioned"
      ]
    },
    {
      "ParameterName": "allow-suspicious-udfs",
      "Description": "Controls whether user-defined functions that have only an
xxx symbol for the main function can be loaded",
      "Source": "engine-default",
      "ApplyType": "static",
      "DataType": "boolean",
      "AllowedValues": "0,1",
      "IsModifiable": false,
      "ApplyMethod": "pending-reboot",
      "SupportedEngineModes": [
        "provisioned"
      ]
    },
    ...

```

RDS API

若要檢視資料庫叢集參數群組的參數值，請搭配下列必要參數使用 RDS API

[DescribeDBClusterParameters](#) 命令。

- DBClusterParameterGroupName

在某些情況下，不會顯示允許的參數值。這些一律為參數，其中來源是資料庫引擎預設值。

若要檢視這些參數的值，您可以執行下列 SQL 陳述式：

- MySQL：

```
-- Show the value of a particular parameter
mysql$ SHOW VARIABLES LIKE '%parameter_name%';

-- Show the values of all parameters
mysql$ SHOW VARIABLES;
```

- PostgreSQL :

```
-- Show the value of a particular parameter
postgresql=> SHOW parameter_name;

-- Show the values of all parameters
postgresql=> SHOW ALL;
```

刪除資料庫叢集參數群組

您可以使用 AWS Management Console、AWS CLI 或 RDS API 刪除資料庫叢集參數群組。資料庫叢集參數群組參數群組參數群組只有在未與資料庫叢集相關聯時才有資格刪除。

主控台

刪除參數群組的步驟

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。參數群組會顯示在清單中。
3. 選擇要刪除的資料庫叢集參數群組名稱。
4. 選擇動作，然後選擇刪除。
5. 檢閱參數群組名稱，然後選擇「刪除」。

AWS CLI

若要刪除資料庫叢集參數群組，請搭配下列必要參數使用 AWS CLI [delete-db-cluster-parameter-group](#) 命令。

- --db-parameter-group-name

Example

下列範例會刪除名為 mydb 參數群組/參數群組的資料庫叢集參數群組。

```
aws rds delete-db-cluster-parameter-group --db-parameter-group-name mydbparametergroup
```

RDS API

若要刪除資料庫叢集參數群組，請使用 RDS API [DeleteDBClusterParameterGroup](#) 命令搭配下列必要參數。

- DBParameterGroupName

比較資料庫參數群組

您可以使用 AWS Management Console 來檢視兩個資料庫參數群組之間的差異。

指定的參數群組必須都是資料庫參數群組，或兩者都必須是資料庫叢集參數群組。當資料庫引擎和版本相同，也是如此。例如，您無法比較 aurora-mysql8.0 (Aurora MySQL 版本 3) 資料庫參數群組和 aurora-mysql8.0 資料庫叢集參數群組。

您可以比較 Aurora MySQL 與 RDS for MySQL 資料庫參數群組，即使是不同的版本也可以，但無法比較 Aurora PostgreSQL 與 RDS for PostgreSQL 資料庫參數群組。

若要比較兩個 DB 參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 在清單中，選擇您要比較的兩個參數群組。

Note

若要將預設參數群組與自訂參數群組進行比較，請先在「預設」頁籤上選擇預設參數群組，然後在「自訂」頁籤上選擇自訂參數群組。

4. 從「動作」中選擇「比較」。

指定資料庫參數

資料庫參數類型包含下列各項：

- Integer
- 布林值
- 字串
- Long
- Double
- 時間戳記
- 其他定義資料類型的物件
- integer、Boolean、string、long、double、timestamp 或 object 參數類型的值陣列

您還可以使用運算式、公式和函數來指定整數及布林值參數。

對於 Oracle 引擎，您可以使用 `DBInstanceClassHugePagesDefault` 公式變數來指定布林值資料庫參數。請參閱[資料庫參數公式變數](#)。

對於 PostgreSQL 引擎，您可以使用運算式來指定布林值資料庫參數。請參閱[布林值資料庫參數運算式](#)。

內容

- [資料庫參數公式](#)
 - [資料庫參數公式變數](#)
 - [資料庫參數公式運算子](#)
- [資料庫參數函數](#)
- [布林值資料庫參數運算式](#)
- [資料庫參數對數運算式](#)
- [資料庫參數值範例](#)

資料庫參數公式

資料庫參數公式是可以解析成整數值或布林值的運算式。以括號 `{}` 括住運算式。您可以針對資料庫參數值使用公式，或將公式指定為資料庫參數函數的引數。

語法

```
{FormulaVariable}  
{FormulaVariable*Integer}  
{FormulaVariable*Integer/Integer}  
{FormulaVariable/Integer}
```

資料庫參數公式變數

每個公式變數都會傳回整數或布林值。變數的名稱區分大小寫。

AllocatedStorage

返回表示資料磁碟區大小的整數 (字元組)。

資料庫 InstanceClassHugePagesDefault

傳回布林值。目前僅支援 Oracle 引擎。

如需詳細資訊，請參閱 [對 RDS for Oracle 執行個體開啟巨型分頁](#)。

資料庫 InstanceClassMemory

傳回一個整數，表示資料庫程序可用的記憶體位元組數。此數字是在內部從資料庫執行個體類別的總記憶體量開始計算得出的。其中，計算會減去為作業系統和管理執行個體的 RDS 程序保留的記憶體。因此，此數字永遠會低於 [資料庫執行個體類別](#) 上執行個體類別表中顯示的記憶體容量。確切值取決於幾個綜合因素。其中包括執行個體類別、資料庫引擎，以及要套用至 RDS 執行個體還是屬於 Aurora 叢集的執行個體。

DBInstanceVCPU

傳回表示 Amazon RDS 管理執行個體所用虛擬中央處理單元 (vCPU) 數目的整數。目前，它僅支援 RDS PostgreSQL 引擎。

EndPointPort

返回表示連線至資料庫執行個體時所用連接埠的整數。

TrueIfReplica

如果資料庫執行個體為僅供讀取複本，則傳回 1，若不是，則傳回 0。這是 MySQL 中 `read_only` 參數的預設值。

資料庫參數公式運算子

資料庫參數公式支援兩種運算子：除法和乘法。

除法運算子：/

用除數除以被除數，並傳回整數之商。商中的小數不進位，直接截斷。

語法

```
dividend / divisor
```

被除數和除數引數必須為整數運算式。

乘法運算子：*

乘以運算式，並傳回運算式的乘積。運算式中的小數不進位，直接截斷。

語法

```
expression * expression
```

兩個運算式都必須為整數。

資料庫參數函數

您可以將資料庫參數函數的引數指定為整數或公式。每個函數至少必須有一個引數。以逗號分隔的清單指定多個引數。清單不能有任何空成員，例如 `argument1`、`argument3`。函數名稱區分大小寫。

IF

傳回引數。

目前僅支援 Oracle 引擎，且唯一支援的第一個引數為

`{DBInstanceClassHugePagesDefault}`。如需更多詳細資訊，請參閱 [對 RDS for Oracle 執行個體開啟巨型分頁](#)。

語法

```
IF(argument1, argument2, argument3)
```

若第一個引數計算結果為 true，傳回第二個引數。否則，傳回第三個引數。

GREATEST

從整數或參數公式清單中傳回最大值。

語法

```
GREATEST(argument1, argument2,...argumentn)
```

傳回整數。

LEAST

從整數或參數公式清單中傳回最小值。

語法

```
LEAST(argument1, argument2,...argumentn)
```

傳回整數。

SUM

新增指定整數或參數公式的值。

語法

```
SUM(argument1, argument2,...argumentn)
```

傳回整數。

布林值資料庫參數運算式

布林值資料庫參數運算式解析為 1 或 0 的布林值。該運算式包含在引號中。

Note

布林值資料庫參數運算式僅支援 PostgreSQL 引擎。

語法

```
"expression operator expression"
```

這兩個運算式必須解析為整數。運算式可以是下列項目：

- 整數常數
- 資料庫參數公式
- 資料庫參數函數
- 資料庫參數變數

布林值資料庫參數運算式支援以下不等式運算子：

大於運算子：>

語法

```
"expression > expression"
```

小於運算子：<

語法

```
"expression < expression"
```

大於或等於運算符：>=、=>

語法

```
"expression >= expression"  
"expression => expression"
```

小於或等於運算子：<=、=<

語法

```
"expression <= expression"  
"expression =< expression"
```

Example 使用布林值資料庫參數運算式

下列布林值資料庫參數運算式範例會將參數公式的結果與整數作比較。這樣做是為了修改 PostgreSQL 資料庫執行個體的布林值資料庫參數 `wal_compression`。參數運算式會將 vCPU 的數目與值 2 作比較。如果 vCPU 的數目大於 2，則 `wal_compression` 資料庫參數設定為 `true`。

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \  
--parameters "ParameterName=wal_compression,ParameterValue=\"{DBInstanceVCPU} > 2\" "
```

資料庫參數對數運算式

您可以將整數資料庫參數值設定為對數運算式。以括號 `{}` 括住運算式。例如：

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

`log` 函數表示對數底數 2。此範例還會使用 `DBInstanceClassMemory` 公式變數。請參閱[資料庫參數公式變數](#)。

Note

目前，您不能使用除了整數以外的任何值指定 MySQL `innodb_log_file_size` 參數。

資料庫參數值範例

這些範例顯示使用資料庫參數值的公式、函數和運算式。

Warning

未正確設定資料庫參數群組中的參數，可能產生各種意外影響。這些影響可能包括降低效能和系統不穩定。修改資料庫參數時請謹慎，在修改資料庫參數群組之前，請備份您的資料。在將這些參數群組變更套用至生產資料庫執行個體之前 `point-in-time-restores`，先嘗試使用建立的測試資料庫執行個體上的參數群組變更。

Example 使用資料庫參數函數 GREATEST

您可以在 Oracle 進程參數中指定 `GREATEST` 函數。使用該函數將使用者進程的數目設定為 80 或 `DBInstanceClassMemory` 除以 9,868,951。

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example 使用資料庫參數函數 LEAST

您可以在 MySQL LEAST 參數值中指定 `max_binlog_cache_size` 函數。使用該函數將交易可在 MySQL 執行個體中使用的快取大小上限設為 1 MB 或 `DBInstanceClass/256` (取較小者)。

```
LEAST({DBInstanceClassMemory/256},10485760)
```

使用 Amazon RDS 資料庫執行個體設定建立 Amazon ElastiCache 快取

ElastiCache 是完全受控的記憶體內快取服務，可提供微秒的讀取和寫入延遲，以支援彈性的即時使用案例。ElastiCache 可協助您加速應用程式和資料庫效能。您可以用 ElastiCache 作不需要資料耐久性的使用案例的主要資料存放區，例如遊戲排行榜、串流和資料分析。ElastiCache 有助於消除與部署和管理分散式運算環境相關的複雜性。如需詳細資訊，請參閱[常見 ElastiCache 使用案例 ElastiCache 以及如何協助 Memcached](#) 和[常見 ElastiCache 使用案例 ElastiCache 以及如何為 Redis 提供協助](#)。您可以使用 Amazon RDS 控制台創建 ElastiCache 緩存。

您可以使用兩種 ElastiCache 格式操作 Amazon。您可以開始使用無伺服器快取，或選擇自行設計快取叢集。如果您選擇設計自己的快取叢集，ElastiCache 可以同時使用 Redis 和 Memcached 引擎。如果您不確定要使用哪個引擎，請參閱[比較 Memcached 和 Redis](#)。有關 Amazon 的更多信息 ElastiCache，請參閱 [Amazon ElastiCache 用戶指南](#)。

主題

- [執行個體設定的 ElastiCache 快取建立概觀](#)
- [使用執行個體的設定建立 ElastiCache 快取](#)

執行個體設定的 ElastiCache 快取建立概觀

您可以使用與新建立的或現有的 RDS 資料庫執行個體相同的組態設定，從 Amazon RDS 建立 ElastiCache 快取。

將 ElastiCache 快取與執行個體建立關聯的一些使用案例：

- 與 RDS ElastiCache 搭配使用而不是單獨在 RDS 上執行，您可以節省成本並改善效能。

例如，您可以節省高達 55% 的成本，並獲得高達 80 倍的讀取效能，只要 ElastiCache 搭配使用適用於 MySQL 的 RDS，即可獲得高達 80 倍的讀取效能。

- 對於不需要資料耐久性的應用程式，您可以使用 ElastiCache 快取做為主要資料存放區。您使用 Redis 或 Memcached 的應用程式幾乎不需要修改即可使 ElastiCache 用。

當您從 RDS 建立 ElastiCache 快取時，ElastiCache 快取會從關聯的 RDS 資料庫執行個體繼承下列設定：

- ElastiCache 連線設定

- **ElastiCache 安全性設定**

您也可以根據需要設定快取組態設定。

ElastiCache 在您的應用程式中設定

您的應用程式必須設定為使用 ElastiCache 快取。您也可以根據需求設定應用程式使用快取策略，藉此最佳化和改善快取效能。

- 若要存取您的 ElastiCache 快取並開始使用，請參閱 [Redis 版 Amazon ElastiCache 入門](#) 和 [Memcached ElastiCache 的 Amazon 入門](#)。
- 如需快取策略的詳細資訊，請參閱 [快取策略與最佳實務](#) (若為 Memcached) 和 [快取策略與最佳實務](#) (若為 Redis)。
- 如需 Redis 叢集中高可用性的 ElastiCache 相關資訊，請參閱 [使用複寫群組的高可用性](#)。
- 您可能會產生與備份儲存、區域內或跨區域的資料傳輸或使用相關的費用 AWS Outposts。如需定價詳細資訊，請參閱 [Amazon ElastiCache 定價](#)。

使用執行個體的設定建立 ElastiCache 快取

您可以使用繼承自 資料庫 RDS 資料庫執行個體建立 ElastiCache 快取。

使用資料庫執行個體的設定建立 ElastiCache 快取

1. 若要建立資料庫執行個體，請遵循 [建立 Amazon RDS 資料庫執行個體](#) 中的指示。
2. 建立 RDS 資料庫執行個體後，主控台會顯示建議的附加元件視窗。選取 [使用您的資料庫設定從 RDS 建立 ElastiCache 叢集]。

對於現有資料庫，在「資料庫」頁面中，選取所需的資料庫執行個體。在動作下拉式功能表中，選擇建立 ElastiCache 叢集以在 RDS 中建立與現有 RDS 資料庫執行個體具有相同設定的 ElastiCache 快取。

在 ElastiCache 組態區段中，來源資料庫識別碼會顯示 ElastiCache 快取繼承設定的資料庫執行個體。

3. 選擇要建立 Redis 叢集還是 Memcached 叢集。如需詳細資訊，請參閱 [比較 Memcached 與 Redis](#)。

ElastiCache cluster configuration [Info](#)

Source DB identifier
mysqlforlambda

Cluster type

Redis

Memcached

Deployment option

Serverless cache - new
Use to quickly create a cache that automatically scales to meet application traffic demands, with no servers to manage.

Design your own cache
Use to create a cache by selecting node type, size, and count.

- 在此之後，選擇您要建立無伺服器快取或設計您自己的快取。如需詳細資訊，請參閱[選擇部署選項](#)。

如果您選擇「無伺服器快取」：

- 在「快取」設定中，輸入「名稱」和「說明」的值。
- 在 [檢視預設設定] 下，保留預設設定，以建立快取和資料庫執行個體之間的連線。
- 您也可以選擇 [自訂預設設定] 來編輯預設設定。選取ElastiCache 連線設定、ElastiCache 安全性設定和使用上限。

- 如果您選擇設計自己的緩存：

- 如果您選擇 Redis 叢集，請選擇是否要保持叢集模式 [啟用] 或 [停用]。如需詳細資訊，請參閱[複寫：Redis \(叢集模式已停用\) 與 Redis \(叢集模式已啟用\)](#)。
- 輸入名稱、描述和引擎版本的值。

對於引擎版本，建議的預設值是最新的引擎版本。您也可以為最符合您需求的 ElastiCache 快取選擇 Engine 版本。

- 在節點類型選項中選擇節點類型。如需詳細資訊，請參閱[管理節點](#)。

如果您選擇在叢集模式設定為已啟用的情況下建立 Redis 叢集，請在碎片數目選項中輸入碎片數目 (分割區/節點群組)。

在複本數目中輸入每個碎片的複本數目。

Note

選取的節點類型、碎片數目以及複本數量都會影響您的快取效能和資源成本。確定這些設定符合您的資料庫需求。如需定價資訊，請參閱 [Amazon ElastiCache 定價](#)。

- d. 選取ElastiCache 連線設定和ElastiCache 安全性設定。您可以根據需求保留預設設定或自訂這些設定。
6. 驗證 ElastiCache 快取的預設和繼承設定。部分設定在建立後無法變更。

Note

RDS 可能會調整 ElastiCache 快取的備份視窗，以符合 60 分鐘的最低視窗需求。來源資料庫的備份視窗會保持不變。

7. 準備就緒後，請選擇 [建立 ElastiCache 快取]。

主控台會顯示 ElastiCache 快取建立的確認標題。請按照標題中的連結前往主 ElastiCache 控制台以檢視快取詳細資料。ElastiCache 控制台顯示新創建的 ElastiCache 緩存。

管理 Amazon RDS 資料庫執行個體

在下列內容，您可以找到管理和維護 Amazon RDS 資料庫執行個體的指示。

主題

- [暫時停止 Amazon RDS 資料庫執行個體](#)
- [啟動之前已停止的 Amazon RDS 資料庫執行個體](#)
- [自動連線 AWS 運算資源和資料庫執行個體](#)
- [修改 Amazon RDS 資料庫執行個體](#)
- [維持資料庫執行個體](#)
- [升級資料庫執行個體引擎版本](#)
- [重新命名資料庫執行個體](#)
- [重新啟動中的資料庫執行個體](#)
- [使用資料庫執行個體僅供讀取複本](#)
- [標記 Amazon RDS 資源](#)
- [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)
- [使用適合 Amazon RDS 資料庫執行個體的儲存體](#)
- [刪除資料庫執行個體](#)

暫時停止 Amazon RDS 資料庫執行個體

您可以間歇性地停止資料庫執行個體以進行暫時測試或每日開發活動。最常見的使用案例是成本最佳化。

Note

在某些情況下，停止資料庫執行個體需要很長時間。要停止資料庫執行個體並立即重新啟動，請重新啟動資料庫執行個體。如需詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

主題

- [停止資料庫執行個體的使用案例](#)
- [支援的資料庫引擎、執行個體類別和區域](#)
- [停止多可用區部署中的資料庫執行個體](#)
- [如何停止資料庫執行個體](#)
- [停止資料庫執行個體的限制](#)
- [選項和參數群組考量](#)
- [公用 IP 位址考量](#)
- [暫時停止資料庫執行個體：基本步驟](#)

停止資料庫執行個體的使用案例

停止和啟動資料庫執行個體比建立資料庫快照、刪除資料庫執行個體，然後在要存取執行個體時還原快照更快。停止執行個體的常見使用案例如下：

- 成本最佳化 — 對於非生產資料庫，您可以暫時停止 Amazon RDS 資料庫執行個體以節省成本。執行個體停止時，不需支付資料庫執行個體小時的費用。

Important

資料庫執行個體停止時，您需要支付佈建儲存 (包含佈建 IOPS) 的費用。您也需支付備份儲存 (包含所指定保留時段內的手動快照和自動備份) 的費用。不過，您無須支付資料庫執行個體小時數的費用。如需詳細資訊，請參閱 [帳單常見問答集](#)。

- **每日開發** — 如果您基於開發目的維護資料庫執行個體，您可以在需要時啟動執行個體，然後在不需要時關閉執行個體。
- **測試** — 您可能需要臨時資料庫執行個體來測試備份和復原程序、移轉、應用程式升級或相關活動。在這些使用案例中，您可以在不需要時停止資料庫執行個體。
- **訓練** — 如果您要在 RDS 中進行訓練，則可能需要在訓練階段期間啟動資料庫執行個體，然後將其關閉。

支援的資料庫引擎、執行個體類別和區域

您可以停止和啟動執行下列資料庫引擎的 Amazon RDS 資料庫執行個體：

- Db2
- MariaDB
- Microsoft SQL 服務器，包括 RDS 自定義 SQL 服務器
- MySQL
- Oracle
- PostgreSQL

所有 AWS 區域和資料庫執行個體類別都支援停止和啟動資料庫執行個體。

停止多可用區部署中的資料庫執行個體

您可以在異地同步備份部署中停止和啟動資料庫執行個體。注意下列限制：

- 您只能在資料庫引擎支援的情況下建立異地同步備份部署。如需引擎支援的詳細資訊，請參閱 [Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎](#)。
- 適用於 SQL Server 的 RDS 不支援停止異地同步備份部署中的資料庫執行個體。如需詳細資訊，請參閱 [Microsoft SQL Server 異地同步備份部署限制、備註和建議](#)。
- 停止資料庫執行個體可能需要很長一段時間。如果在先前的容錯移轉之後至少有一個備份，則可以透過容錯移轉作業執行重新開機來加速停止作業。如需詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

如何停止資料庫執行個體

停止操作會在下列階段發生：

1. 資料庫執行個體啟動正常關機程序。

資料庫執行個體的狀態變更為 `stopping`。

2. 執行個體會停止執行，最長為連續 7 天。

資料庫執行個體的狀態變更為 `stopped`。

已停止資料庫執行個體的特性

處於停止狀態時，您的資料庫執行個體具有下列特性：

- 停止的資料庫執行個體會保留下列項目
 - 執行個體 ID
 - 網域名稱伺服器 (DNS) 端點
 - 參數群組
 - 安全群組
 - Option group (選項群組)
 - Amazon S3 交易日誌 (point-in-time還原所需)

當您重新啟動資料庫執行個體時，它會保有與停止時相同的組態。

- 所有的儲存磁碟區仍會繼續連結至資料庫執行個體，其中的資料也會保留。RDS 會刪除儲存在資料庫執行個體 RAM 中的任何資料。

資料庫執行個體停止時，您需要支付佈建儲存 (包含佈建 IOPS) 的費用。您也需支付備份儲存 (包含所指定保留時段內的手動快照和自動備份) 的費用。

- RDS 會移除擱置中的動作 (包括排程的維護更新)，但資料庫執行個體的選項群組或資料庫參數群組的擱置動作除外。

Note

偶爾 RDS for PostgreSQL 資料庫執行個體關閉時不會完全清理資料。如果發生這種情況，之後重新啟動執行個體時執行個體會執行復原程序。這是資料庫引擎的預期行為，目的是保護資料庫完整性。某些以記憶體為基礎的統計數字和計數器不會保留歷史記錄，且會在重新啟動後重新初始化，以擷取往後的作業工作負載。

自動重啟已停止的資料庫執行個體

如果資料庫執行個體停止連續七天後，您沒有手動將它啟動，RDS 會自動啟動您的資料庫執行個體。如此一來，您的執行個體就不會落後於任何必要的維護更新。若要了解如何依排程停止和啟動執行個體，請參閱[如何使用 Step Functions 停止 Amazon RDS 執行個體超過 7 天？](#)。

停止資料庫執行個體的限制

以下為停止和啟動資料庫執行個體的一些限制：

- 您無法在異地同步備份部署中停止適用於 SQL Server 的 RDS 資料庫執行個體。
- 您不能停止擁有僅供讀取複本的資料庫執行個體，或本身就是僅供讀取複本的資料庫執行個體。
- 您無法修改已停止的資料庫執行個體。
- 您無法刪除已和停止的資料庫執行個體建立關聯的選項群組。
- 您無法刪除已和停止的資料庫執行個體建立關聯的資料庫參數群組。
- 在多可用區部署中，啟動資料庫執行個體之後，可能會切換主要和次要可用區域。

RDS Custom for SQL Server 適用額外的限制。如需詳細資訊，請參閱[啟動與停止 RDS Custom for SQL Server 資料庫執行個體](#)。

選項和參數群組考量

如果資料庫執行個體已關聯至選項群組，則您無法從選項群組移除持續選項 (包括永久選項)。此功能同樣適用於任何狀態為 `stopping`、`stopped` 或 `starting` 的資料庫執行個體。

您可以變更與已停止的資料庫執行個體相關聯的選項群組或資料庫參數群組。不過，直到您下次啟動資料庫執行個體後才會發生變更。若您選擇立即套用變更，則變更將在您啟動資料庫執行個體時發生。否則，在啟動資料庫執行個體後，變更要在下一次維護時段才會發生。

公用 IP 位址考量

當您停止資料庫執行個體時，它將保留 DNS 端點。如果您停止具有公有 IP 地址的資料庫執行個體，Amazon RDS 會釋放其公有 IP 位址。重新啟動資料庫執行個體時，其具有不同的公有 IP 地址。

Note

您應該一律使用 DNS 端點而非使用 IP 地址來連接到資料庫執行個體。

暫時停止資料庫執行個體：基本步驟

您可以使用 AWS Management Console、AWS CLI、或 RDS API 停止資料庫。

主控台

停止資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要停止的資料庫執行個體。
3. 針對 Actions (動作)，選擇 Stop temporarily (暫時停止)。
4. 在 Stop DB instance temporarily (暫時停止資料庫執行個體) 中，選取資料庫執行個體將在 7 天後自動重新啟動的確認。
5. (選用) 選取 Save the DB instance in a snapshot (將資料庫執行個體儲存在快照中)，然後輸入 Snapshot name (快照名稱) 的快照名稱。如果您想在資料庫執行個體停止前為其建立快照，請選擇此選項。
6. 選擇 Stop temporarily (暫時停止) 來停止資料庫執行個體，或選擇 Cancel (取消) 來取消操作。

AWS CLI

若要使用停止資料庫執行個體 AWS CLI，請使用下列選項呼叫 [stop-db-instance](#) 命令：

- `--db-instance-identifier` – 資料庫執行個體名稱。

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

RDS API

若要使用 Amazon RDS API 停止資料庫執行個體，請搭配下列參數呼叫 [StopDBInstance](#) 操作：

- `DBInstanceIdentifier` – 資料庫執行個體名稱。

啟動之前已停止的 Amazon RDS 資料庫執行個體

您可以暫停 Amazon RDS 資料庫執行個體以節省金錢。在資料庫執行個體停止之後，您可以再次將其啟動並開始使用。關於停止和開始資料庫執行個體的詳細資訊，請參閱 [暫時停止 Amazon RDS 資料庫執行個體](#)。

當您啟動先前停止的資料庫執行個體時，該資料庫執行個體會保留某些資訊。此資訊是 ID、網域名稱、伺服器 (DNS) 端點、參數群組、安全群組和選項群組。當您啟動一個停止的執行個體時，將一律視為一個完整執行個體小時收費。

主控台

啟動資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要啟動的資料庫執行個體。
3. 針對 Actions (動作)，選擇 Start (啟動)。

AWS CLI

若要使用 AWS CLI 來啟動資料庫執行個體，請搭配下列選項呼叫 [start-db-instance](#) 命令：

- `--db-instance-identifier` – 資料庫執行個體名稱。

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

RDS API

若要使用 Amazon RDS API 啟動資料庫執行個體，請搭配下列參數呼叫 [StartDBInstance](#) 操作：

- `DBInstanceIdentifier` – 資料庫執行個體名稱。

自動連線 AWS 運算資源和資料庫執行個體

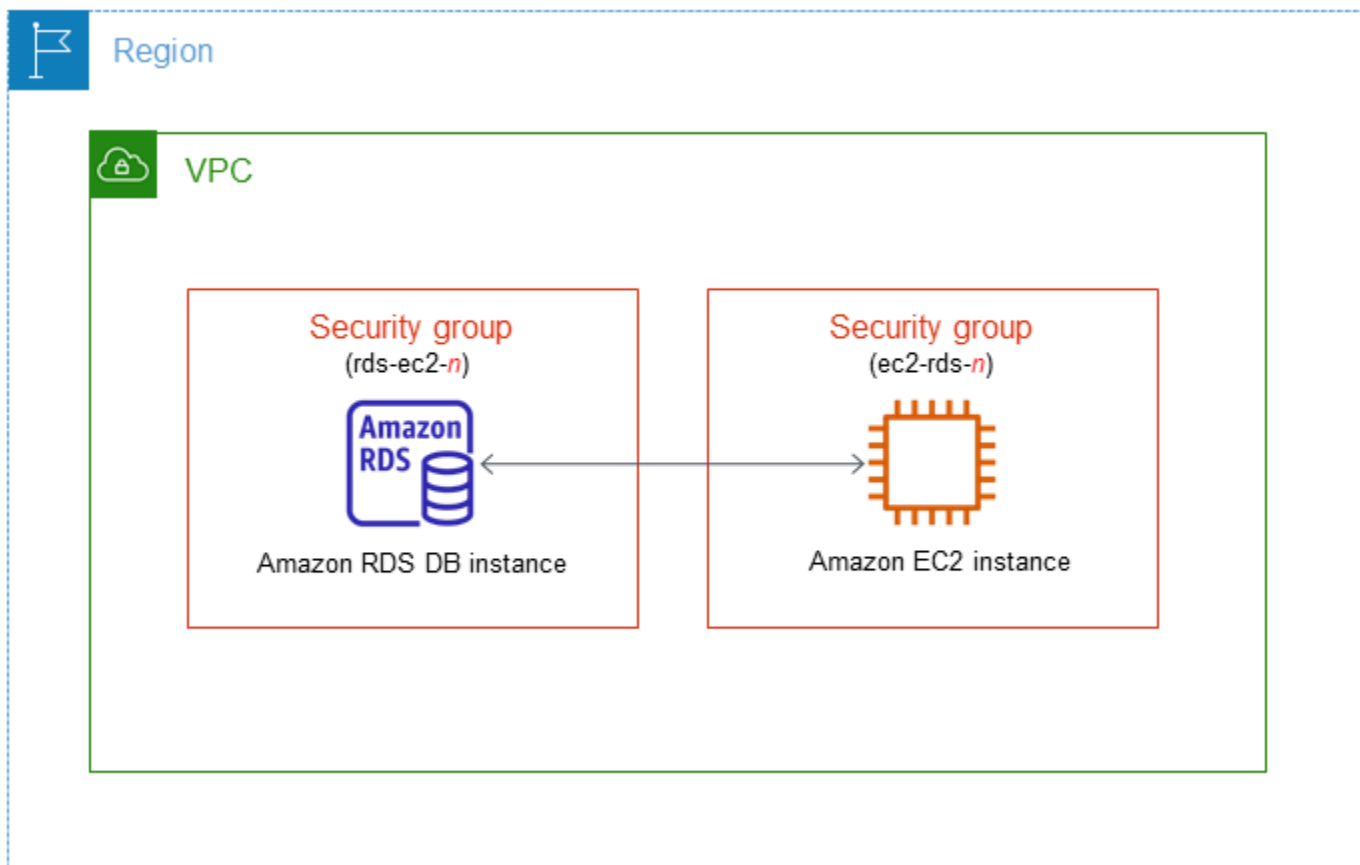
您可以自動連接資料庫執行個體和 AWS 運算資源，例如 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和 AWS Lambda 函數。

主題

- [自動連線 EC2 執行個體和資料庫執行個體](#)
- [自動連線 Lambda 函數和資料庫執行個體](#)

自動連線 EC2 執行個體和資料庫執行個體

您可以使用 Amazon RDS 主控台，簡化 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體與資料庫執行個體之間的連線設定。通常，您的資料庫執行個體位於私有子網路中，而 EC2 執行個體位於 VPC 內的公有子網路中。您可以在 EC2 執行個體上使用 SQL 用戶端，連線到資料庫執行個體。EC2 執行個體也可以執行存取私有資料庫執行個體的 Web 伺服器或應用程式。如需詳細了解多可用區域資料庫叢集與 EC2 執行個體之間的連線設定，請參閱 [the section called “連線 EC2 執行個體和多可用區域資料庫叢集”](#)。



如果您想要連線至與資料庫執行個體不在相同 VPC 中的 EC2 執行個體，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 中的案例。

主題

- [與 EC2 執行個體自動連線的概要](#)
- [自動連線 EC2 執行個體和 RDS 資料庫](#)
- [檢視已連線的運算資源](#)
- [連線至執行特定資料庫引擎的資料庫執行個體](#)

與 EC2 執行個體自動連線的概要

當您在 EC2 執行個體與 RDS 資料庫之間自動設定連線時，Amazon RDS 會為您的 EC2 執行個體以及 RDS 資料庫設定 VPC 安全群組。

以下是連線 EC2 執行個體與 RDS 資料庫的要求：

- EC2 執行個體必須與 RDS 資料庫存在於相同的 VPC 中。

如果沒有 EC2 執行個體存在於相同的 VPC 中，則主控台會提供一個連結來建立該執行個體。

- 設定連線的使用者必須擁有執行下列 Amazon EC2 操作的許可：

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

如果資料庫執行個體和 EC2 執行個體位於不同的可用區域，您的帳戶可能會產生跨可用區域成本。

當您設定 EC2 執行個體的連線時，Amazon RDS 會根據與 RDS 資料庫和 EC2 執行個體相關聯之安全群組的目前組態執行動作，如下表所述。

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
<p>有一或多個與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code> (其中 <i>n</i> 是數字)。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-n</code> (其中 <i>n</i> 是數字)。符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其具有 RDS 資料庫的 VPC 安全群組做為來源。</p>	<p>RDS 不會採取任何動作。</p> <p>已在 EC2 執行個體與 RDS 資料庫之間自動設定連線。由於 EC2 執行個體與 RDS 資料庫之間已存在連線，因此不會修改安全群組。</p>
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。 有一或多個與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與 EC2 執行個體連線。Amazon RDS 無法使用沒有傳入規則的安全群組，並將具有 EC2 執行個體的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。修改範例包括新增規則或變更現有規則的連線埠。 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-n</code>。 有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-n</code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與 RDS 資料庫連線。Amazon RDS 無法使用沒有傳出規則的安全群組，並將具有 RDS 資料庫的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。 	<p>RDS action: create new security groups</p>
<p>有一或多個與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-n</code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與 RDS 資料庫連線。Amazon RDS 無法使用沒有傳出規則的安</p>	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
	<p>全群組，並將具有 RDS 資料庫的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。</p>	
<p>有一或多個與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>存在用於連線的有效 EC2 安全群組，但與 EC2 執行個體沒有相關聯。此安全群組具有符合模式 <code>ec2-rds-<i>n</i></code> 的名稱。尚未將其修改。其只包含一個傳出規則，具有 RDS 資料庫的 VPC 安全群組做為來源。</p>	<p>RDS action: associate EC2 security group</p>
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。 有一或多個與 RDS 資料庫相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與 EC2 執行個體連線。Amazon RDS 無法使用沒有傳入規則的安全群組，並將具有 EC2 執行個體的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。 	<p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-<i>n</i></code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其具有 RDS 資料庫的 VPC 安全群組做為來源。</p>	<p>RDS action: create new security groups</p>

RDS 動作：建立新的安全群組

Amazon RDS 會採取下列動作：

- 建立符合模式 `rds-ec2-n` 的新安全群組。此安全群組具有一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。此安全群組與 RDS 資料庫相關聯，並可讓 EC2 執行個體存取 RDS 資料庫。
- 建立符合模式 `ec2-rds-n` 的新安全群組。此安全性群組具有以 RDS 資料庫資料庫之 VPC 安全性群組做為目標的輸出規則。此安全群組與 EC2 執行個體相關聯，並可讓 EC2 執行個體將流量傳送到 RDS 資料庫。

RDS 動作：關聯 EC2 安全群組

Amazon RDS 會將有效的現有 EC2 安全群組與 EC2 執行個體建立關聯。此安全群組允許 EC2 執行個體將流量傳送到 RDS 資料庫。

自動連線 EC2 執行個體和 RDS 資料庫

在設定 EC2 執行個體與 RDS 資料庫之間的連線之前，請確定您符合 [與 EC2 執行個體自動連線的概要](#) 中所述的要求。

如果您在設定連線之後變更這些安全群組，這些變更可能會影響 EC2 執行個體與 RDS 資料庫之間的連線。

Note

您只能使用 AWS Management Console，在 EC2 執行個體與 RDS 資料庫之間自動設定連線。您無法使用 AWS CLI 或 RDS API 自動設定連線。

自動連線 EC2 執行個體與 RDS 資料庫

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇 RDS 資料庫。
3. 從動作，選擇設定 EC2 連線。

Set up EC2 connection (設定 EC2 連線) 頁面即會出現。

4. 在 Set up EC2 connection (設定 EC2 連線) 頁面上，選擇 EC2 執行個體。

Set up EC2 connection [Info](#)

Select EC2 instance

Database
database-test1

EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-1234567890abcdef0
ec2-database-connect us-east-1c

[Create EC2 instance](#)

Cancel [Continue](#)

如果相同 VPC 中沒有 EC2 執行個體，請選擇 [Create EC2 instance](#) (建立 EC2 執行個體) 來建立執行個體。在此情況下，請確定新 EC2 執行個體與 RDS 資料庫位於相同的 VPC 中。

5. 選擇繼續。

[Review and confirm](#) (檢閱並確認) 頁面即會出現。

Review and confirm

Connection summary [Info](#)

You are setting up a connection between RDS database [database-test1](#) and EC2 instance [i-1234567890abcdef0](#).



Bold indicates an addition being made to set up a connection.

Changes to RDS database: database-test1

Attribute	Current value	New value
Security group	default	default, rds-ec2-1

Changes to EC2 instance: i-1234567890abcdef0

Attribute	Current value	New value
Security group	launch-wizard-5	launch-wizard-5, ec2-rds-1

Cancel

Previous

Confirm and set up

- 在 Review and confirm (檢閱並確認) 頁面上，檢閱 RDS 將做出以設定與 EC2 執行個體連線的變更。

如果變更正確，請選擇確認並設定。

如果變更不正確，請選擇 Previous (上一步) 或 Cancel (取消)。

檢視已連線的運算資源

您可以使用檢視連線 AWS Management Console 至 RDS 資料庫 Aurora 資料庫的計算資源。顯示的資源包括已自動設定的運算資源連線。您可以使用下列方式自動設定與運算資源的連線：

- 您可以在建立資料庫時選取運算資源。

如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#) 及 [建立多可用區域資料庫叢集](#)。

- 您可以設定現有資料庫與運算資源之間的連線。

如需詳細資訊，請參閱 [自動連線 EC2 執行個體和 RDS 資料庫](#)。

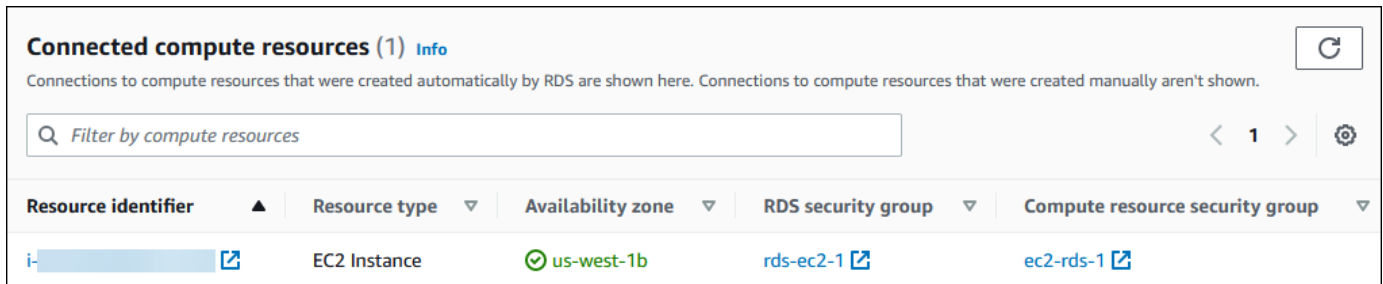
列出的運算資源不包括已手動連線至資料庫的運算資源。例如，您可以透過將規則新增至與資料庫關聯的 VPC 安全群組，來允許運算資源手動存取資料庫。

針對要列出的運算資源，必須滿足下列條件：

- 與運算資源相關聯的安全群組，其名稱符合模式 `ec2-rds-n` (其中 *n* 是數字)。
- 與運算資源相關聯的安全群組具有傳出規則，其連接埠範圍設為 RDS 資料庫所使用的連接埠。
- 與運算資源相關聯的安全群組具有傳出規則，其來源設為與 RDS 資料庫相關聯的安全群組。
- 與 RDS 資料庫相關聯的安全群組，其名稱符合模式 `rds-ec2-n` (其中 *n* 是數字)。
- 與 RDS 資料庫相關聯的安全群組具有傳入規則，其連接埠範圍設為 RDS 資料庫所使用的連接埠。
- 與 RDS 資料庫相關聯的安全群組具有傳入規則，其來源設為與運算資源相關聯的安全群組。

若要檢視連線至 RDS 資料庫的運算資源。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇 RDS 資料庫的名稱。
3. 在 Connectivity & security (連線和安全) 索引標籤上，檢視 Connected compute resources (已連線的運算資源)。



Resource identifier	Resource type	Availability zone	RDS security group	Compute resource security group
i-	EC2 Instance	us-west-1b	rds-ec2-1	ec2-rds-1

連線至執行特定資料庫引擎的資料庫執行個體

如需連線至執行特定資料庫引擎之資料庫執行個體的相關資訊，請遵循資料庫引擎的指示：

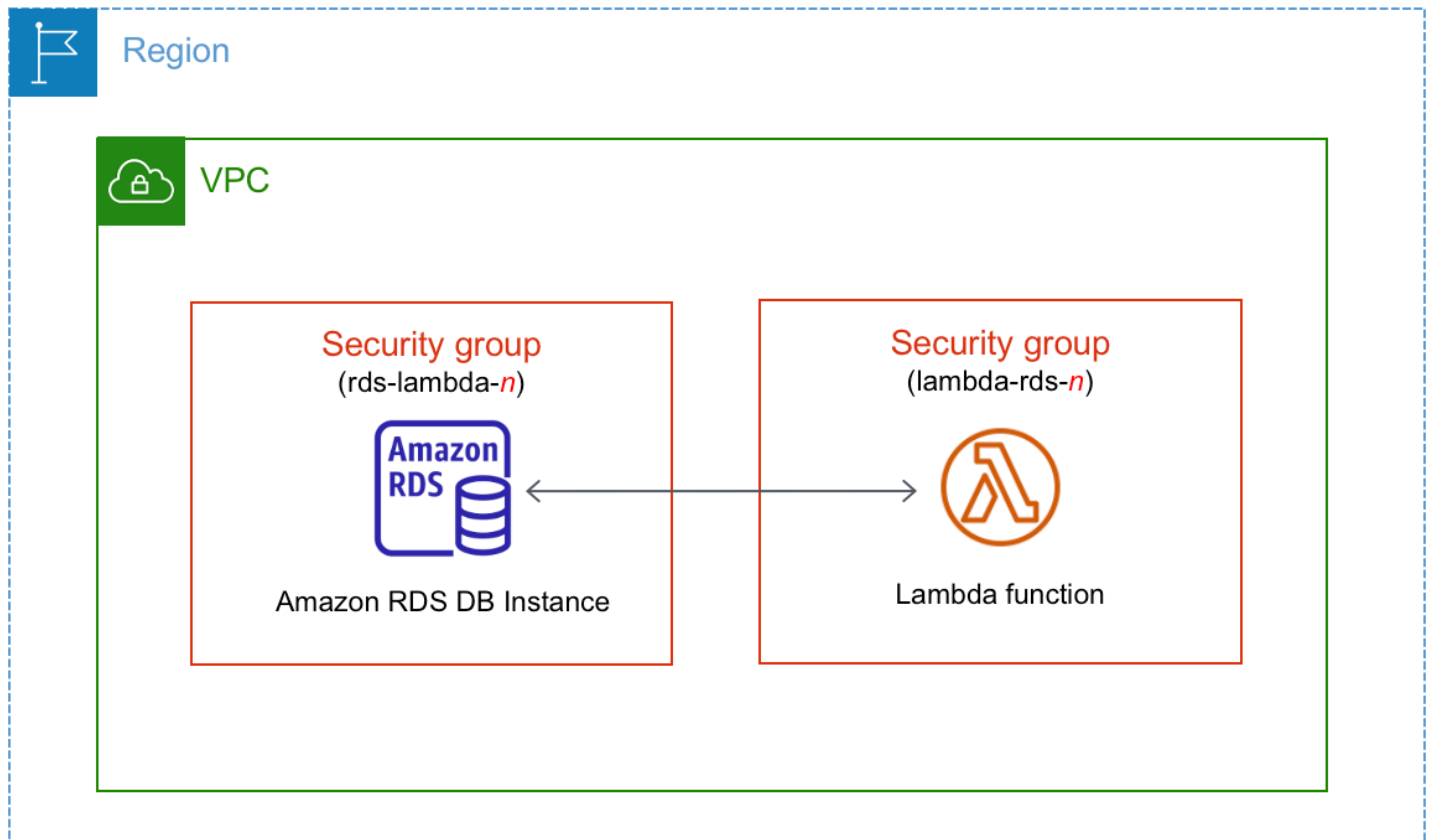
- [連接至執行 MariaDB 資料庫引擎的資料庫執行個體](#)
- [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)
- [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)
- [連線至 RDS for Oracle 資料庫執行個體](#)
- [連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)

自動連線 Lambda 函數和資料庫執行個體

您可以使用 Amazon RDS 主控台，簡化 Lambda 函數與資料庫執行個體之間的連線設定。通常，您的資料庫執行個體位於 VPC 內的私有子網路中。Lambda 函數可供應用程式用來存取您的私有資料庫執行個體。

如需詳細了解多可用區域資料庫叢集與 Lambda 函數之間的連線設定，請參閱 [the section called “正在連線 Lambda 函數和多可用區域資料庫叢集”](#)。

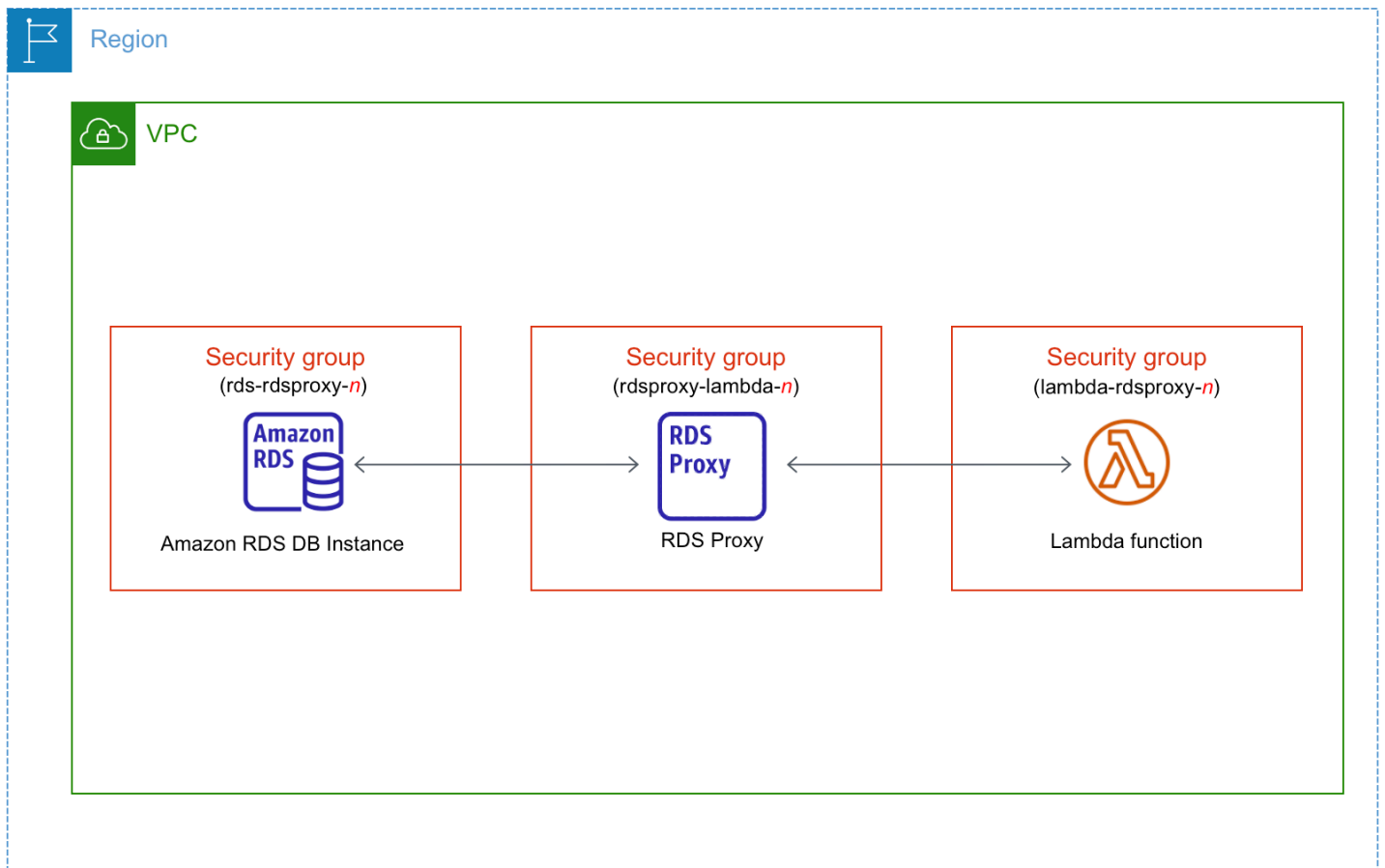
下圖顯示您的資料庫執行個體和 Lambda 函數之間的直接連線。



您可以透過 RDS Proxy 設定 Lambda 函數和資料庫執行個體來改善資料庫效能和復原能力。通常，Lambda 函數經常進行短暫的資料庫連線，這些連線將受益於 RDS Proxy 提供的連線集區。您可以利用 Lambda 函數中已有的任何 AWS Identity and Access Management (IAM) 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫憑證。如需更多詳細資訊，請參閱 [使用 Amazon RDS Proxy](#)。

當您使用主控台與現有代理連線時，Amazon RDS 會更新代理安全群組，以允許來自資料庫執行個體和 Lambda 函數的連線。

您也可以從相同的主控台頁面建立新的代理。當您在主控台中建立代理時，若要存取資料庫執行個體時，您必須輸入您的資料庫憑證或選取 AWS Secrets Manager 密碼。



主題


- [使用 Lambda 函數自動連線概觀](#)
- [自動連線 Lambda 函數和 RDS 資料庫](#)
- [檢視已連線的運算資源](#)

使用 Lambda 函數自動連線概觀

以下是連線 Lambda 函數與 RDS 資料庫執行個體 的要求：

- Lambda 函數必須存在於與資料庫執行個體相同的 VPC 內。
- 設定連線的使用者必須擁有執行下列 Amazon RDS、Amazon EC2、Lambda、Secrets Manager 和 IAM 操作的許可：
 - Amazon RDS
 - `rds:CreateDBProxies`
 - `rds:DescribeDBInstances`

- `rds:DescribeDBProxies`
- `rds:ModifyDBInstance`
- `rds:ModifyDBProxy`
- `rds:RegisterProxyTargets`
- Amazon EC2
 - `ec2:AuthorizeSecurityGroupEgress`
 - `ec2:AuthorizeSecurityGroupIngress`
 - `ec2:CreateSecurityGroup`
 - `ec2>DeleteSecurityGroup`
 - `ec2:DescribeSecurityGroups`
 - `ec2:RevokeSecurityGroupEgress`
 - `ec2:RevokeSecurityGroupIngress`
- Lambda
 - `lambda:CreateFunctions`
 - `lambda:ListFunctions`
 - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
 - `secretsmanager:CreateSecret`
 - `secretsmanager:DescribeSecret`
- IAM
 - `iam:AttachPolicy`
 - `iam:CreateRole`
 - `iam:CreatePolicy`
- AWS KMS
 - `kms:describeKey`

 Note

如果資料庫執行個體和 Lambda 函數位於不同的可用區域，您的帳戶可能會產生跨可用區域成本。

當您在 Lambda 函數與 RDS 資料庫之間自動設定連線時，Amazon RDS 會為您的函數以及資料庫執行個體設定 VPC 安全群組。如果您使用 RDS Proxy，則 Amazon RDS 也會為代理設定 VPC 安全群組。Amazon RDS 會根據與資料庫執行個體、Lambda 函數和代理相關聯之安全群組的目前組態執行動作，如下表所述。

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果代理已連線至您的資料庫執行個體，RDS 將檢查相關聯代理的 TargetHealth 是否為 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code> (其中 <i>n</i> 是數字)。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳出規則，其使用資料庫執行個體的 VPC 安全群組或代理做為目的地。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code> (其中 <i>n</i> 是數字)。</p> <p>符合模式的安全群組尚未修改。此安全群組具有傳入和傳出規則，其中包含 Lambda 函數和資料庫執行個體的 VPC 安全群組。</p>	<p>Amazon RDS 不會採取任何動作。</p> <p>Lambda 函數、代理 (選用) 和資料庫執行個體之間已自動設定連線。由於函數、代理與資料庫之間已存在連線，因此不會修改安全群組。</p>
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。 有一或多個與 Lambda 函數相關 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。 有一或多個與代理相關聯的安全群組，其名稱符 	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<ul style="list-style-type: none"> 有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。不過，這些安全群組都不能用於與 Lambda 函數的連線。 <p>Amazon RDS 無法使用沒有一個傳入規則 (使用 Lambda 函數的 VPC 安全群組或代理做為來源) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。修改範例包括新增規則或變更現有規則的連線埠。</p>	<p>聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體連線。</p> <p>Amazon RDS 無法使用沒有傳出規則的安全群組，並將具有資料庫執行個體或代理的 VPC 安全群組做為目的地。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>合 <code>rdsproxy-lambda-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體或 Lambda 函數連線。</p> <p>Amazon RDS 無法使用沒有傳入和傳出規則 (包含資料庫執行個體和 Lambda 函數的 VPC 安全群組) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。</p>	

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。</p> <p>不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體連線。Amazon RDS 無法使用沒有傳出規則的安全群組，並將具有資料庫執行個體或代理的 VPC 安全群組做為目的地。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。</p> <p>不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體或 Lambda 函數連線。Amazon RDS 無法使用沒有傳入和傳出規則 (包含資料庫執行個體和 Lambda 函數的 VPC 安全群組) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>存在用於連線的有效 Lambda 安全群組，但與 Lambda 函數沒有相關聯。此安全群組具有符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code> 的名稱。尚未將其修改。只有一個輸出規則，其中的 VPC 安全群組資料庫執行個體或代理做為目的地。</p>	<p>存在用於連線的有效代理安全群組，但與代理沒有相關聯。此安全群組具有符合模式 <code>rdsproxy-lambda-<i>n</i></code> 的名稱。尚未將其修改。具有包含資料庫執行個體的 VPC 安全群組的傳入和傳出規則，和 Lambda 函數。</p>	<p>RDS action: associate Lambda security group</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。 有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。不過，Amazon RDS 無法使用這些安全群組中的任一個與 Lambda 函數或代理連線。 <p>Amazon RDS 無法使用沒有一個傳入規則 (使用 Lambda 函數的 VPC 安全群組或代理做為來源) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個輸出規則，其中的 VPC 安全群組資料庫執行個體或代理做為目的地。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。</p> <p>符合模式的安全群組尚未修改。此安全群組具有傳入和傳出規則，其中包含資料庫執行個體和 Lambda 函數的 VPC 安全群組。</p>	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。 有一或多個與資料庫執行個體相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者如果相關聯代理的 TargetHealth 是 AVAILABLE。不過，Amazon RDS 無法使用這些安全群組中的任一個與 Lambda 函數或代理連線。 <p>Amazon RDS 無法使用沒有一個傳入規則 (使用 Lambda 函數的 VPC 安全群組或代理做為來源) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。 有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體連線。 <p>Amazon RDS 無法使用沒有傳出規則的安全群組，並將具有資料庫執行個體或代理的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。 有一或多個與代理相關聯的安全群組，其名稱符合 <code>rdsproxy-lambda-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與資料庫執行個體或 Lambda 函數連線。 <p>Amazon RDS 無法使用沒有傳入和傳出規則 (包含資料庫執行個體和 Lambda 函數的 VPC 安全群組) 的安全群組。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>RDS action: create new security groups</p>

RDS 動作：建立新的安全群組

Amazon RDS 會採取下列動作：

- 建立符合模式的新安全群組 `rds-lambda-n` 或者 `rds-rdsproxy-n` (如果您選擇使用 RDS Proxy)。此安全群組具有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。與資料庫執行個體關聯的此一安全群組可讓函數或代理存取資料庫執行個體。
- 建立符合模式 `lambda-rds-n` 或 `lambda-rdsproxy-n` 的新安全群組。此安全群組有一個輸出規則，其中的 VPC 安全群組資料庫執行個體或代理做為目的地。此安全群組與 Lambda 函數相關聯，並允許函數將流量傳送至資料庫執行個體或通過代理傳送流量。
- 建立符合模式 `rdsproxy-lambda-n` 的新安全群組。此安全群組具有傳入和傳出規則，其中包含資料庫執行個體和 Lambda 函數的 VPC 安全群組的傳入和傳出規則。

RDS 動作：與 Lambda 相關聯的安全群組

Amazon RDS 將有效、現有的 Lambda 安全群組與 Lambda 函數建立關聯。此安全群組允許函數將流量傳送至資料庫執行個體或通過代理傳送流量。

自動連線 Lambda 函數和 RDS 資料庫

您可以使用 Amazon RDS 主控台將 Lambda 函數自動連線到您的資料庫執行個體。簡化資源之間建立連線的過程。

您也可以使用 RDS Proxy 將代理包含在連線中。Lambda 函數經常進行短暫的資料庫連線，這些連線將受益於 RDS Proxy 提供的連線集區。您還可以使用 Lambda 函數中已有的任何 IAM 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫憑證。

您可以使用設定 Lambda 連線頁面，將現有的資料庫執行個體連線至新的和現有的 Lambda 函數。設定程序會自動為您設定所需的安全群組。

在設定 Lambda 函數和資料庫執行個體之間的連線之前，請確定：

- 你的 Lambda 函數和資料庫執行個體在相同 VPC 中。
- 您擁有正確的使用者帳戶權限。有關需求的詳細資訊，請參閱 [使用 Lambda 函數自動連線概觀](#)。

如果您在設定連線之後變更安全群組，這些變更可能會影響 Lambda 函數與資料庫執行個體之間的連線。

Note

您可以自動設定資料庫執行個體和僅在 AWS Management Console 中 Lambda 函數之間的連線。若要連線一個 Lambda 函數，資料庫執行個體必須處於可用狀態。

若要自動連線 Lambda 函數和資料庫執行個體

<result>

在您確認設定之後，Amazon RDS 會開始連線 Lambda 函數、RDS Proxy (如果您使用代理) 程序，以及資料庫執行個體。主控台會顯示連線詳細資訊對話方塊，此對話方塊會列出允許資源之間連線的安全群組變更。

</result>

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇資料庫，然後選擇您要將其連線至 Lambda 函數的資料庫執行個體。
3. 針對動作，選擇設定 Lambda 連線。
4. 在設定 Lambda 連線頁面中的選取 Lambda 函數，執行下列任一操作：
 - 如果您在相同的 VPC 中擁有現有 Lambda 函數資料庫執行個體，選擇選擇現有函數，然後選擇函數。
 - 如果您在同一個 VPC 中沒有 Lambda 函數，請選擇建立新函數，然後輸入函數名稱。預設執行期會設定為 Nodejs.18。完成連線設定後，您可以在 Lambda 主控台中修改新 Lambda 函數的設定。
5. (選用) 在 RDS Proxy 中，選取使用 RDS Proxy 連線，然後執行下列任一項操作：
 - 若您要使用現有的代理，請選擇選擇現有的代理，然後選擇代理。
 - 如果您沒有代理，並且您希望 Amazon RDS 自動為您建立代理，請選擇建立新的代理。然後，為資料庫憑證，執行下列任一操作：
 - a. 選擇資料庫使用者名稱和密碼，然後為您的資料庫執行個體輸入使用者名稱和密碼。
 - b. 選擇 Secrets Manager 密碼。然後，針對選取密碼，選擇一個 AWS Secrets Manager 密碼。如果您沒有 Secrets Manager 密碼，請選擇建立新的 Secrets Manager 密碼，以[建立新的密碼](#)。建立密碼之後，針對選取密碼，選擇新的密碼。

建立新的代理之後，請選擇選擇現有的代理，然後選擇代理。請注意，您的代理可能需要一些時間才能用於連線。

6. (選用) 展開連線摘要並為您的資源驗證反白顯示的更新。
7. 選擇 Set up (設定)。

檢視已連線的運算資源

您可以使用 AWS Management Console 檢視連線至資料庫執行個體的 Lambda 函數。顯示的資源包括 Amazon RDS 自動設定的運算資源連線。

列出的運算資源不含手動連線至資料庫執行個體。例如，您可以透過將規則新增至與資料庫關聯的 VPC 安全群組，來允許運算資源手動存取您的資料庫執行個體。

若要讓主控台列出 Lambda 函數，必須符合下列條件：

- 與運算資源相關聯的安全群組，其名稱符合模式 `lambda-rds-n` 或 `lambda-rdsproxy-n` (其中 *n* 是數字)。
- 與運算資源相關聯的安全群組具有傳出規則，其連線埠範圍設為資料庫執行個體或關聯代理所使用的連線埠。傳出規則的目的地必須設定為與資料庫執行個體或關聯代理相關聯的安全群組。
- 如果組態包含代理，則附加到資料庫相關聯代理的安全群組，名稱符合模式 `rdsproxy-lambda-n` (其中 *n* 是數字)。
- 與函數相關聯的安全群組具有傳出規則，其連線埠設定為資料庫執行個體或相關的代理所使用的連線埠。目的地必須設定為與資料庫執行個體或關聯代理相關聯的安全群組。

若要檢視自動連線至資料庫執行個體的運算資源

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇資料庫，然後選擇資料庫執行個體。
3. 在連線與安全性索引標籤中，檢視已連線的運算資源中的運算資源。

修改 Amazon RDS 資料庫執行個體

您可以變更資料庫執行個體的設定，以完成新增額外儲存體或變更資料庫執行個體類別等任務。在本主題中，您可以瞭解如何修改 Amazon RDS 資料庫執行個體，以及瞭解資料庫執行個體的設定。

建議您先在測試執行個體上測試任何變更，然後再修改生產執行個體。這樣做可協助您充分了解每個變更的影響。這在升級資料庫版本時特別重要。

對資料庫執行個體進行的大多數修改，您可以立即套用或延遲到下一個維護時段才進行。而諸如參數群組變更等部分修改作業，則要求您手動重新啟動資料庫執行個體，才能使變更生效。

Important

部分修改作業會導致系統停機；因為 Amazon RDS 必須重新啟動資料庫執行個體，才能使變更生效。因此，請務必先檢視修改作業對資料庫與應用程式所造成的影響，再修改資料庫執行個體設定。

主控台

修改資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 變更您要的任何設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。
5. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看修改摘要。
6. (選用) 選擇 Apply immediately (立即套用) 以立即套用變更。在某些情況下，選擇此選項會導致停機。如需詳細資訊，請參閱 [排程修改設定](#)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB instance (修改資料庫執行個體) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要使用修改資料庫執行個體 AWS CLI，請呼叫命 [modify-db-instance](#) 令。指定資料庫執行個體識別符，以及您要修改選項的值。如需每個選項的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

Example

以下程式碼會修改 mydbinstance，將備份保留期設為 1 週 (7 天)。程式碼可藉由使用 `--deletion-protection` 來啟用刪除保護。若要停用刪除保護，請使用 `--no-deletion-protection`。使用 `--no-apply-immediately`，會在下一次維護時段期間由系統套用變更。使用 `--apply-immediately` 可立即套用變更。如需詳細資訊，請參閱 [排程修改設定](#)。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --backup-retention-period 7 \  
  --deletion-protection \  
  --no-apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --backup-retention-period 7 ^  
  --deletion-protection ^  
  --no-apply-immediately
```

RDS API

若要使用 Amazon RDS API 來修改資料庫執行個體，請呼叫 [ModifyDBInstance](#) 操作。指定資料庫執行個體識別符，以及您要修改設定的參數。如需每項參數的相關資訊，請參閱 [資料庫執行個體的設定](#)。

排程修改設定

修改資料庫執行個體時，您可以決定何時進行修改。

Schedule modifications

When to apply modifications

- Apply during the next scheduled maintenance window
Current maintenance window: April 10, 2024 05:28 - 05:58 (UTC-04:00)
- Apply immediately
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

若要立即套用變更，而不是在下一個維護時段中套用，請選擇中的「立即套用」選項 AWS Management Console。或者，在 `--apply-immediately` 使用 Amazon RDS API 呼叫 AWS CLI 或將 `ApplyImmediately` 參數設定為 `true` 時，您也可以使用參數。

如果您不選擇立即套用變更，RDS 會將變更放入擱置的修改佇列中。在下一個維護時段期間，RDS 會套用佇列中的任何暫止變更。若您選擇立即套用變更，則系統會一併套用待定修改佇列中的新變更及任何變更內容。

若要查看下一個維護時段擱置的修改，請使用 [describe-db-instances](#) AWS CLI 指令並檢查 `PendingModifiedValues` 欄位。

Important

如果有任何擱置的修改需要資料庫執行個體暫時無法使用 (停機)，選擇 `apply immediately` (立即套用) 選項可能會導致非預期的停機時間。

當您選擇立即套用變更時，任何待定的修改也會立即套用，而不會等到下一次維護時段才套用。

若您不想在下一個維護時段套用待定變更，您可修改資料庫執行個體以還原該變更。您可以使用 AWS CLI 並指定 `--apply-immediately` 選項來執行此操作。

即使您選擇延遲變更，針對某些資料庫設定的變更仍會立即套用。若要查看不同的資料庫設定如何影響立即套用設定，請參閱 [資料庫執行個體的設定](#)。

資料庫執行個體的設定

在下表中，您可以找到有關可以修改和不可修改之設定的詳細資訊。也可以找到何時可以套用變更，以及變更是否會導致資料庫執行個體停機。若使用多可用區域這類 Amazon RDS 功能，您之後修改資料庫執行個體時，可以最大限度減少停機時間。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。

您可以使用主控台、[modify-db-instance](#) CLI 命令或 [ModifyDBInstance](#) RDS API 操作來修改資料庫執行個體。

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>配置儲存</p> <p>要配置給資料庫執行個體的儲存體 (以 GiB 為單位)。您只能增加配置的儲存體。您無法減少配置的儲存體。</p> <p>您無法修改部分舊版資料庫執行個體的儲存體，或是由舊版資料庫快照還原的資料庫執行個體。如果資料庫執行個體不符合資格，主控台中的 Allocated storage (配置的儲存體) 選項就會停用。您可以檢查是否可以通過使用 CLI 命令 describe-valid-db-instance-修改分配更多的存儲。此命令會傳回您資料庫執行個體的有效儲存選項。</p> <p>如果資料庫執行個體狀態為 storage-optimization，則您無法修改已配置的儲存體。如果已在過去六小時修改資料庫執行個體，則您也無法修改其已配置的儲存體。</p> <p>允許的儲存體上限視資料庫引擎及儲存體類型而定。如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體。</p>	<p>CLI 選項：</p> <p><code>--allocated-storage</code></p> <p>RDS API 參數：</p> <p>Allocated Storage</p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。但是，進行變更時可能會降低效能。</p>	<p>所有資料庫引擎</p>
<p>架構組態</p> <p>允許多個租戶資料庫駐留在資料庫執行個體的組態。目前，只</p>	<p>CLI 選項：</p> <p><code>--multi-tenant (CDB</code></p>	<p>系統會立即進行變更。</p>	<p>進行此變更時，不會發生停機。</p>	<p>Oracle</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>有 RDS for Oracle 容器資料庫 (CDB) 支援此設定。</p> <p>如果 CDB 位於單一租戶組態中，您可以修改它，以使用多租戶組態。在此組態中，您可以使用 RDS API 建立 1—30 個租用戶資料庫，具體取決於資料庫版本和任何必要的選項授權。不支援應用程式 PDB 和 Proxy PDB。多租戶組態是永久的，這表示您以後無法將 CDB 轉換回單一租戶組態。</p> <div data-bbox="115 940 597 1493" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Amazon RDS 功能稱為「multi-tenant」而不是「multitenant」，因為它是 RDS 平台的一項功能，而不只是 Oracle 資料庫引擎而已。「Oracle 多租戶」一詞僅指與內部部署和 RDS 部署相容的 Oracle 資料庫架構。</p> </div> <p>如需詳細資訊，請參閱 RDS for Oracle CDB 概觀。</p>	<p>架構的多租戶組態)</p> <p>--no-multi-tenant (CDB 架構的單一租戶組態)</p> <p>API 參數：</p> <p>MultiTenant</p>			

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>架構設定</p> <p>Oracle 資料庫的架構：CDB 或非 CDB。如果您選擇 Oracle 多租戶架構，RDS for Oracle 會將您的非 CDB 轉換為使用單一租戶組態的 CDB。</p> <p>您的資料庫必須是執行 Oracle 資料庫 19c 且具有 2021 年 4 月或更高 RU 的非 CDB，才支援此設定。轉換後，您的 CDB 會包含一個初始插入式資料庫 (PDB)。架構變更是永久的，這表示您無法將 CDB 轉換回非 CDB。</p> <div data-bbox="115 1081 596 1444" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>若要將單一租戶組態中的 CDB 轉換為多租戶組態，請再次修改 CDB 執行個體，然後為架構組態選擇多租戶組態。</p> </div> <p>如需詳細資訊，請參閱 CDB 架構的單一租戶組態。</p>	<p>CLI 選項：</p> <p><code>--engine oracle-ee-cdb</code> (Oracle 多租戶)</p> <p><code>--engine oracle-se2-cdb</code> (Oracle 多租戶)</p> <p>API 參數：</p> <p>Engine</p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>Oracle</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Auto minor version upgrade (自動次要版本升級)</p> <p>選擇 [啟用 auto 動次要版本升級]，讓您的資料庫執行個體在推出時自動接收偏好的次要資料庫引擎版本升級。這是預設行為。Amazon RDS 會在維護時段執行自動次要版本升級。如果您未選擇「啟用 auto 動次要版本升級」，則當新的次要版本可用時，資料庫執行個體不會自動升級。</p> <p>如需詳細資訊，請參閱 自動升級次要引擎版本。</p>	<p>CLI 選項：</p> <pre>--auto-minor-version-upgrade --no-auto-minor-version-upgrade</pre> <p>RDS API 參數：</p> <pre>AutoMinorVersionUpgrade</pre>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>
<p>備份複寫</p> <p>選擇 [啟用複寫至其他 AWS 區域]，在其他區域建立備份以進行災難復原。</p> <p>然後選擇其他備份的 Destination Region (目的地區域)。</p>	<p>無法在修改資料庫執行個體時使用。如需使用 AWS CLI 或 RDS API 啟用跨區域備份的相關資訊，請參閱啟用跨區域自動備份。</p>	<p>系統會盡快以非同步的方式來套用變更。</p>	<p>進行此變更時，不會發生停機。</p>	<p>Oracle、PostgreSQL、SQL Server</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Backup retention period (備份保留期間)</p> <p>自動備份保留的天數。若要停用自動備份，請將備份保留期設為 0。</p> <p>如需詳細資訊，請參閱 備份簡介。</p> <div data-bbox="115 747 597 1108" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>如果您使用 AWS Backup 來管理備份，則不適用此選項。如需相關資訊 AWS Backup，請參閱 AWS Backup 開發人員指南。</p> </div>	<p>CLI 選項：</p> <p><code>--backup-retention-period</code></p> <p>RDS API 參數：</p> <p><code>BackupRetentionPeriod</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，並將設定從非零值變更為另一個非零值，則系統會盡快以非同步的方式來套用變更。否則，在下一次維護時段期間，系統才會進行變更。</p>	<p>如果將 0 變更為非零值，或將非零值變更為 0，便會發生停機。</p> <p>這適用於單一可用區和多個可用區資料庫執行個體。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>備份時段</p> <p>進行自動資料庫備份的時間範圍。備份時段是國際標準時間 (UTC) 格式的開始時間和以小時為單位的持續時間。</p> <p>如需詳細資訊，請參閱 備份簡介。</p> <div data-bbox="115 800 597 1161" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>如果您使 AWS Backup 用管理備份，則不會顯示此選項。如需相關資訊 AWS Backup，請參閱開 AWS Backup 發人員指南。</p> </div>	<p>CLI 選項：</p> <p><code>--preferred-backup-window</code></p> <p>RDS API 參數：</p> <p>PreferredBackupWindow</p>	<p>系統會盡快以非同步的方式來套用變更。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>
<p>憑證授權單位</p> <p>資料庫執行個體所使用之伺服器憑證的憑證授權單位 (CA)。</p> <p>如需詳細資訊，請參閱。</p>	<p>CLI 選項：</p> <p><code>--ca-certificate-identifier</code></p> <p>RDS API 參數：</p> <p>CACertificateIdentifier</p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>只有在資料庫引擎不支援無需重新啟動即可輪換時，才會發生停機。您可以使用指 describe-db-engine-versions AWS CLI 令來判斷 DB 引擎是否支援旋轉而不重新啟動。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Copy tags to snapshots (將標籤複製到快照)</p> <p>如果您有任何資料庫執行個體標籤，啟用此選項，以便在您建立資料庫快照時複製這些標籤。</p> <p>如需詳細資訊，請參閱 標記 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <p><code>--copy-tags-to-snapshot</code> 或 <code>--no-copy-tags-to-snapshot</code> *</p> <p>RDS API 參數：</p> <p>CopyTagsToSnapshot</p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>
<p>Database port (資料庫連線埠)</p> <p>您要用來存取資料庫執行個體的連接埠。</p> <p>與資料庫執行個體相關聯之選項群組中的選項所指定的任何連接埠值，都不可與該連接埠值相符。</p> <p>如需詳細資訊，請參閱 連接至 Amazon RDS 資料庫執行個體。</p>	<p>CLI 選項：</p> <p><code>--db-port-number</code></p> <p>RDS API 參數：</p> <p>DBPortNumber</p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>資料庫執行個體會立即重新啟動。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>DB engine version (資料庫引擎版本)</p> <p>您要使用的資料庫引擎的版本。升級生產資料庫執行個體之前，建議您先在測試資料庫執行個體上測試升級程序。這樣做有助於確認其持續時間並驗證您的應用程式。</p> <p>如需詳細資訊，請參閱 升級資料庫執行個體引擎版本。</p>	<p>CLI 選項：</p> <p><code>--engine-version</code></p> <p>RDS API 參數：</p> <p><code>EngineVersion</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>所有資料庫引擎</p>
<p>DB instance class (資料庫執行個體類別)</p> <p>您要使用的資料庫執行個體類別。</p> <p>如需詳細資訊，請參閱 資料庫執行個體類別。</p>	<p>CLI 選項：</p> <p><code>--db-instance-class</code></p> <p>RDS API 參數：</p> <p><code>DBInstanceClass</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>DB instance identifier (資料庫執行個體識別符)：</p> <p>新的資料庫執行個體識別符。此值會以小寫字母字串的形式儲存。</p> <p>如需重新命名資料庫執行個體的效果詳細資訊，請參閱重新命名資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--new-db-instance-identifier</pre> <p>RDS API 參數：</p> <pre>NewDBInstanceIdentifier</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>除非您的資料庫引擎版本支援動態 SSL 上傳，否則在此變更期間將發生停機。若要判斷您的版本是否需要重新啟動，請執行下列 AWS CLI 命令：</p> <pre>aws rds describe-db-engine-versions \ --default-only \ --engine <i>your-db-engine</i> \ --query 'DBEngineVersions[*].SupportsCertificateRotationWithoutRestart'</pre>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>DB parameter group (資料庫參數群組)</p> <p>要與資料庫執行個體建立關聯的資料庫參數群組。</p> <p>如需詳細資訊，請參閱 使用參數群組。</p>	<p>CLI 選項：</p> <p><code>--db-parameter-group-name</code></p> <p>RDS API 參數：</p> <p><code>DBParameterGroupName</code></p>	<p>新資料庫參數群組與資料庫執行個體的關聯會立即發生。</p>	<p>將新的資料庫參數群組與資料庫執行個體建立關聯時，不會發生停機時間。</p> <p>資料庫參數群組的關聯性與參數群組中參數變更的應用程式不同。RDS 只有在您手動重新啟動資料庫執行個體後，才會在新關聯的群組中套用修改的靜態和動態參數設定。但是，如果您在將資料庫參數群組與資料庫執行個體建立關聯之後修改動態參數，則這些參數設定會立即套用，而不需要重新開機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
			如需詳細資訊，請參閱 使用參數群組 及 重新啟動中的資料庫執行個體 。	
<p>專用日誌磁碟區</p> <p>使用專用日誌磁碟區 (DLV)，將資料庫交易日誌儲存在儲存體磁碟區上，此儲存體磁碟區與包含資料庫資料表的磁碟區不同。</p> <p>如需詳細資訊，請參閱 使用專用日誌磁碟區 (DLV)。</p>	<p>CLI 選項：</p> <p><code>-dedicated-log-volume</code></p> <p>RDS API 參數：</p> <p><code>DedicatedLogVolume</code></p>	<p>資料庫執行個體重新啟動時會套用變更。</p>	<p>資料庫執行個體重新啟動時會停機。</p>	<p>MariaDB、MySQL、PostgreSQL</p>
<p>刪除保護</p> <p>啟用刪除保護可避免您的資料庫執行個體遭意外刪除。</p> <p>如需詳細資訊，請參閱 刪除資料庫執行個體。</p>	<p>CLI 選項：</p> <p><code>--deletion-protection --no-deletion-protection</code></p> <p>RDS API 參數：</p> <p><code>DeletionProtection</code></p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Enhanced Monitoring (增強型監控)</p> <p>Enable Enhanced Monitoring (啟用增強型監控)，以針對資料庫執行個體執行所在的作業系統即時收集指標。</p> <p>如需詳細資訊，請參閱 使用增強型監控來監控作業系統指標。</p>	<p>CLI 選項：</p> <p><code>--monitoring-interval</code> 與 <code>--monitoring-role-arn</code></p> <p>RDS API 參數：</p> <p><code>MonitoringInterval</code>、與 <code>MonitoringRoleArn</code></p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>IAM DB authentication (IAM 資料庫身分驗證)</p> <p>Enable IAM DB authentication (啟用 IAM 資料庫身分驗證)，透過使用者和角色驗證資料庫使用者。</p> <p>如需詳細資訊，請參閱 適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證。</p>	<p>CLI 選項：</p> <pre>--enable-iam-database-authentication --no-enable-iam-database-authentication</pre> <p>RDS API 參數：</p> <pre>EnableIAMDatabaseAuthentication</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。</p>	<p>僅限 MariaDB、MySQL 和 PostgreSQL</p>
<p>Kerberos 身分驗證</p> <p>選擇要將資料庫執行個體移至的 Active Directory。目錄必須在此作業執行之前便存在。如果已經選取目錄，則可以指定 None (無)，從其目前目錄移除資料庫執行個體。</p> <p>如需詳細資訊，請參閱 Kerberos 身分驗證。</p>	<p>CLI 選項：</p> <pre>--domain 與 --domain-iam-role-name</pre> <p>RDS API 參數：</p> <pre>Domain、與 DomainIAMRoleName</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生短暫的停機。</p>	<p>僅限 Microsoft SQL Server、MySQL、Oracle 及 PostgreSQL</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>License model (授權模式)</p> <p>選擇bring-your-own-license使用您的 Db2 和甲骨文的許可證。</p> <p>選擇 license-included (已包含授權) 來使用 Microsoft SQL Server 或 Oracle 的一般授權合約。</p> <p>如需詳細資訊，請參閱Amazon 適用於 Db2 的授權選項、Amazon RDS 上的 Microsoft SQL Server 授權及RDS for Oracle 授權選項。</p>	<p>CLI 選項：</p> <p><code>--license-model</code></p> <p>RDS API 參數：</p> <p><code>LicenseModel</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>僅限 Microsoft SQL Server 及 Oracle</p>
<p>Log exports (日誌匯出)</p> <p>要發佈到 Amazon CloudWatch 日誌的資料庫日誌檔類型。</p> <p>如需詳細資訊，請參閱 將資料庫日誌發佈至 Amazon CloudWatch Logs。</p>	<p>CLI 選項：</p> <p><code>--cloudwatch-logs-export-configuration</code></p> <p>RDS API 參數：</p> <p><code>CloudwatchLogsExportConfiguration</code></p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Maintenance window (維護時段)</p> <p>進行系統維護的時間範圍。系統維護包括升級 (如適用)。維護時段是國際標準時間 (UTC) 格式的開始時間和以小時為單位的持續時間。</p> <p>如果您將時段設定為目前時間，則目前時間與結束時段必須至少間隔 30 分鐘。此計時有助於確保系統成功套用任何待定變更。</p> <p>如需詳細資訊，請參閱 Amazon RDS 維護時段。</p>	<p>CLI 選項：</p> <p><code>--preferred-maintenance-window</code></p> <p>RDS API 參數：</p> <p>Preferred MaintenanceWindow</p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>如果您有一個或多個會導致停機的擱置動作，並將維護時段變更至涵蓋目前時間，則系統會立即套用這些擱置動作，最終導致停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>管理中的主要認證 AWS Secrets Manager</p> <p>選取 Manage master credentials in AWS Secrets Manager (管理 AWS Secrets Manager 中的主要憑證) 以秘密管理 Secrets Manager 中的主要使用者密碼。</p> <p>選擇性地選擇要用來保護機密的 KMS 金鑰。從您帳戶中的 KMS 金鑰進行選擇，或輸入來自不同帳戶的金鑰。</p> <p>如果 RDS 已在管理資料庫執行個體的主要使用者密碼，您可以選擇 Rotate secret immediately (立即輪換機密) 來輪換主要使用者密碼。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager。</p>	<p>CLI 選項：</p> <pre>--manage-master-user-password --no-manage-master-user-password</pre> <pre>--master-user-secret-kms-key-id</pre> <pre>--rotate-master-user-password --no-rotate-master-user-password</pre> <p>RDS API 參數：</p> <pre>ManageMasterUserPassword</pre>	<p>如果您開啟或關閉自動主要使用者密碼管理，變更會立即發生。此變更會忽略立即套用設定。</p> <p>如果您要輪換主要使用者密碼，則必須指定立即套用變更。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
	MasterUserSecretKeyId RotateMasterUserPassword			
<p>Multi-AZ deployment (異地同步備份部署)</p> <p>Yes (是) 以便在多個可用區域部署您的資料庫執行個體。否則為 No (否)。</p> <p>如需詳細資訊，請參閱 設定及管理多可用區部署。</p>	CLI 選項： --multi-az --no-multi-az RDS API 參數： MultiAZ	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。但是，可能會對效能產生影響。如需詳細資訊，請參閱 將資料庫執行個體修改為多可用區域資料庫執行個體部署。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Network type (網路類型)</p> <p>受資料庫執行個體支援的 IP 地址通訊協定。</p> <p>IPv4，指定資源只能透過網際網路通訊協定第 4 版 (IPv4) 定址通訊協定與資料庫執行個體進行通訊。</p> <p>雙堆疊模式，指定資源可以透過 IPv4、網際網路通訊協定第 6 版 (IPv6) 或兩者與資料庫執行個體進行通訊。如果您有任何資源必須透過 IPv6 定址通訊協定與您的資料庫執行個體進行通訊，請使用雙堆疊模式。此外，請確保將 IPv6 CIDR 區塊與您指定的資料庫子網路群組中的所有子網路相關聯。</p> <p>如需詳細資訊，請參閱 Amazon RDS IP 定址。</p>	<p>CLI 選項：</p> <p><code>--network-type</code></p> <p>RDS API 參數：</p> <p><code>NetworkType</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，可能會發生停機。</p>	<p>所有資料庫引擎</p>
<p>新主要使用者密碼</p> <p>主要使用者的密碼。密碼必須包含 8–41 個英數字元。</p>	<p>CLI 選項：</p> <p><code>--master-user-password</code></p> <p>RDS API 參數：</p> <p><code>MasterUserPassword</code></p>	<p>系統會盡快以非同步的方式來套用變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Option group (選項群組)</p> <p>要與資料庫執行個體相關聯的選項群組。</p> <p>如需詳細資訊，請參閱 使用選項群組。</p>	<p>CLI 選項：</p> <p><code>--option-group-name</code></p> <p>RDS API 參數：</p> <p><code>OptionGroupName</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。有一種情形例外，那就是將 MariaDB 稽核外掛程式新增至 RDS for MariaDB 或 RDS for MySQL 資料庫執行個體，這樣做可能會造成中斷。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Performance Insights (績效詳情)</p> <p>Enable Performance Insights (啟用績效詳情) 會監控您的資料庫執行個體負載，讓您可分析資料庫效能並對其進行故障排除。</p> <p>績效詳情不適用於某些資料庫引擎版本和資料庫執行個體類別。如果您的資料庫執行個體無法使用 Performance Insights (績效詳情) 區段，則不會顯示在主控台中。</p> <p>如需詳細資訊，請參閱 在 Amazon RDS 上使用績效詳情監控資料庫負載 及 Performance Insights 支援 Amazon RDS 資料庫引擎和執行個體類別。</p>	<p>CLI 選項：</p> <pre>--enable-performance-insights --no-enable-performance-insights</pre> <p>RDS API 參數：</p> <pre>EnablePerformanceInsights</pre>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>除了 Db2 以外的全部</p>
<p>Performance Insights AWS KMS key (績效詳情)</p> <p>用於加密 Performance Insights 資料的 AWS KMS 金鑰識別碼。AWS KMS key 金鑰識別碼是 Amazon 資源名稱 (ARN)、金鑰 AWS KMS 鑰識別碼或 KMS 金鑰的金鑰別名。</p> <p>如需詳細資訊，請參閱 開啟和關閉 Amazon RDS 的 Performance Insights。</p>	<p>CLI 選項：</p> <pre>--performance-insights-kms-key-id</pre> <p>RDS API 參數：</p> <pre>PerformanceInsightsKMSKeyId</pre>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>除了 Db2 以外的全部</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>績效詳情保留期間</p> <p>Performance Insights 資料的保留時間，單位為天。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 Performance Insights 的定價和資料保留。</p> <p>如需詳細資訊，請參閱 開啟和關閉 Amazon RDS 的 Performance Insights。</p>	<p>CLI 選項：</p> <pre>--performance-insights-retention-period</pre> <p>RDS API 參數：</p> <pre>PerformanceInsightsRetentionPeriod</pre>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>除了 Db2 以外的全部</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>處理器功能</p> <p>為資料庫執行個體的資料庫執行個體類別指定的 CPU 核心數和每個核心的執行緒數目。</p> <p>如需詳細資訊，請參閱 針對 RDS for Oracle 中的資料庫執行個體類別設定處理器。</p>	<p>CLI 選項：</p> <pre>--processor-features 與 --use-default-processor-features --no-use-default-processor-features</pre> <p>RDS API 參數：</p> <pre>ProcessorFeatures、與 UseDefaultProcessorFeatures</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>僅限 Oracle</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>佈建 IOPS</p> <p>資料庫執行個體的新佈建 IOPS (每秒輸入/輸出作業數) 值。只有在您針對 Storage type (儲存體類型) 選擇下列其中一項時，才能使用此設定：</p> <ul style="list-style-type: none"> • General purpose SSD (gp3) (一般用途 SSD (gp3)) • Provisioned IOPS SSD (io1) (佈建 IOPS SSD (io1)) • 佈建的 IOPS 固態硬碟 (io2) <p>如需詳細資訊，請參閱 the section called “佈建 IOPS 儲存體” 及 the section called “gp3 儲存裝置 (建議使用)”。</p>	<p>CLI 選項：</p> <p><code>--iops</code></p> <p>RDS API 參數：</p> <p>Iops</p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>公用存取</p> <p>Publicly accessible (可公開存取) 用來給予資料庫執行個體一個公有 IP 地址，這表示可在 VPC 外加以存取。資料庫執行個體也必須位於 VPC 的公有子網路中，才能公開存取。</p> <p>Not publicly accessible (不可公開存取) 將使得資料庫執行個體只能從 VPC 內部存取。</p> <p>如需詳細資訊，請參閱 在 VPC 中的網際網路中隱藏資料庫執行個體。</p> <p>若要從其 VPC 外部連線到資料庫執行個體，資料庫執行個體必須可公開存取。亦須使用資料庫執行個體安全群組的傳入規則授予存取權。此外，必須滿足其他要求。如需詳細資訊，請參閱 無法連線至 Amazon RDS 資料庫執行個體。</p> <p>如果您的資料庫執行個體無法公開存取，您也可以使用 AWS Site-to-Site VPN 連線或連線，從私人網路存取該執行個體。如需詳細資訊，請參閱 網際網路流量隱私權。</p>	<p>CLI 選項：</p> <pre>--publicly-accessible --no-publicly-accessible</pre> <p>RDS API 參數：</p> <p>PubliclyAccessible</p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>安全群組</p> <p>要與資料庫執行個體相關聯的 VPC 安全群組。</p> <p>如需詳細資訊，請參閱 使用安全群組控制存取。</p>	<p>CLI 選項：</p> <p><code>--vpc-security-group-ids</code></p> <p>RDS API 參數：</p> <p><code>VpcSecurityGroupId</code></p>	<p>系統會盡快以非同步的方式來套用變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>
<p>儲存體自動調整規模</p> <p>Enable storage autoscaling (啟用儲存體自動調整規模) 可讓 Amazon RDS 在需要避免資料庫執行個體耗盡儲存空間時，自動增加儲存體。</p> <p>使用 Maximum storage threshold (儲存體閾值上限) 可設定 Amazon RDS 的上限，以自動增加資料庫執行個體的儲存體。預設為 1,000 GiB。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS 儲存體自動擴展自動管理容量。</p>	<p>CLI 選項：</p> <p><code>--max-allocated-storage</code></p> <p>RDS API 參數：</p> <p><code>MaxAllocatedStorage</code></p>	<p>系統會立即進行變更。此設定會忽略立即套用設定。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>Storage throughput (訊息輸送量)</p> <p>資料庫執行個體的新儲存體輸送量值。只有在您針對 Storage type (儲存體類型) 選擇 General purpose SSD (gp3) (一般用途 SSD (gp3)) 時，才能使用此設定。</p> <p>如需詳細資訊，請參閱 the section called “gp3 儲存裝置 (建議使用)”。</p>	<p>CLI 選項：</p> <p><code>--storage-throughput</code></p> <p>RDS API 參數：</p> <p><code>StorageThroughput</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。</p>	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>儲存體類型</p> <p>您要使用的儲存體類型。</p> <p>如果您選擇 General Purpose SSD (gp3) (一般用途 SSD (gp3))，您可以在 Advanced settings (進階設定) 下佈建額外的 Provisioned IOPS (佈建 IOPS) 和 Storage throughput (儲存體輸送量)。</p> <p>如果您選擇佈建 IOPS 固態硬碟 (io1) 或佈建 IOPS 固態硬碟 (io2)，請輸入佈建的 IOPS 值。</p> <p>在 Amazon RDS 開始修改您的資料庫執行個體以變更儲存體大小或類型之後，六小時內您都無法提交另一個請求來變更儲存體大小、效能或類型。</p> <p>如需詳細資訊，請參閱 Amazon RDS 儲存類型。</p>	<p>CLI 選項：</p> <p><code>--storage-type</code></p> <p>RDS API 參數：</p> <p><code>StorageType</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>下列所有變更都會導致程序開始時短暫停機。之後，當變更生效時，您就可以正常使用資料庫。</p> <ul style="list-style-type: none"> 從 General Purpose (SSD) (一般用途 (SSD)) 或 Provisioned IOPS (SSD) (佈建 IOPS (SSD)) 到 Magnetic (磁帶)。 從 Magnetic (磁帶) 到 General Purpose (SSD) (一般用途 (SSD)) 或 Provisioned IOPS (SSD) (佈建 IOPS (SSD))。 	<p>所有資料庫引擎</p>

主控台設定和說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註	支援的資料庫引擎
<p>DB subnet group (資料庫子網路群組)</p> <p>資料庫執行個體的資料庫子網路群組。您可以使用此設定將資料庫執行個體移動至不同的 VPC。</p> <p>如需詳細資訊，請參閱 Amazon VPC 和 Amazon RDS。</p>	<p>CLI 選項：</p> <p><code>--db-subnet-group-name</code></p> <p>RDS API 參數：</p> <p><code>DBSubnetGroupName</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>	<p>所有資料庫引擎</p>

維持資料庫執行個體

Amazon RDS 會定期在 Amazon RDS 資源上執行維護。維護通常涉及更新資料庫執行個體中的以下資源：

- 基礎硬體
- 基礎作業系統 (OS)
- 資料庫引擎版本

作業系統更新大多是因為安全性問題。你應該盡快完成更新。

進行某些維護項目時，Amazon RDS 需要將您的資料庫執行個體短暫離線。需要資源離線的維護項目包括必要的作業系統或資料庫修補。所需的修補程式僅會針對與安全性和執行個體可靠性相關的修補程式自動安排。這類修補不常發生，通常每隔幾個月進行一次。維護僅需片刻的時間即可完成。

您已選擇不立即套用的延遲資料庫執行個體修改，亦將會在下一次維護時段套用。例如，您可能選擇在維護時段變更資料庫執行個體類別或參數群組。您使用待定重新開機設定所指定的這類修改不會顯示在待定維護清單中。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

若要查看下一個維護時段擱置的修改，請使用[描述be-db-instance](#) AWS CLI 命令並檢查欄位。PendingModifiedValues

主題

- [檢視待處理的維護](#)
- [套用資料庫執行個體的更新](#)
- [異地同步備份部署的維護](#)
- [Amazon RDS 維護時段](#)
- [調整偏好的資料庫執行個體維護時段](#)
- [使用強制作業系統更新](#)

檢視待處理的維護

使用 RDS 主控台或 RDS API 檢視資料庫執行個體是否有可用的 AWS CLI 維護更新。若有可用的更新，Amazon RDS 主控台就會在資料庫執行個體的 Maintenance (維護) 欄中表示，如下所示。

Current activity	Maintenance	VPC	Multi-AZ
0 Connections	none	vpc-2aed394c	No
0 Connections	next window	vpc-2aed394c	No
0.02 Sessions	none	vpc-2aed394c	No

如果資料庫執行個體沒有可用的維護更新，欄的值就會是 none (無)。

如果資料庫執行個體有可用的維護更新，就可能是以下的欄值：

- 必要 – 維護動作將套用至資源，無法無限期延遲。
- available (可用) – 維護動作可用，但不會自動套用至資源。您可手動套用。
- next window (下個時段) – 維護動作會在下個維護時段中套用到資源。
- In progress (進行中) – 維護動作正在套用至資源。

若有可用的更新，您可以採取以下其中一個動作：

- 如果維護值為 next window (下個時段)，請由 Actions (動作) 選擇 Defer upgrade (延遲升級) 以延遲維護項目。如果維護動作已經啟動，則無法延遲維護動作。
- 立即套用維護項目
- 排定要在下一個維護時段中啟動的維護項目。
- 不採取動作。

若要進行此項動作，請選擇資料庫執行個體顯示其詳細資訊，然後選擇 Maintenance & backups (維護及備份)。隨即出現待定維護項目。

The screenshot displays the 'Maintenance & backups' tab in the Amazon RDS console. It is divided into two main sections: 'Maintenance' and 'Pending maintenance (1)'. The 'Maintenance' section includes three key items: 'Auto minor version upgrade' which is 'Enabled', a 'Maintenance window' of 'mon:11:28-mon:11:58 UTC (GMT)', and 'Pending maintenance' set to 'next window'. The 'Pending maintenance (1)' section features a table with one entry, a search filter, and buttons for 'Apply now' and 'Apply at next maintenance window'.

Description	Type	Status	Apply date
Automatic minor version upgrade to postgres 9.6.11	db-upgrade	next window	February 25th 2019, 3:28:00 am UTC-8 (local)

維護時段決定等待中的操作何時開始，但不限制這些操作的總執行時間。維護操作不保證在維護時段結束之前完成，可能持續到超過指定的結束時間。如需更多詳細資訊，請參閱 [Amazon RDS 維護時段](#)。

您也可以執行 [describe-pending-maintenance-actions](#) AWS CLI 命令，檢視資料庫執行個體是否有可用的維護更新。

套用資料庫執行個體的更新

透過 Amazon RDS，即可自行選擇套用維護操作的時機。您可以使用 RDS 主控台 AWS Command Line Interface (AWS CLI) 或 RDS API 來決定 Amazon RDS 何時套用更新。

Note

對於 RDS for SQL Server，您可以停止並啟動資料庫執行個體，或者再次縱向擴展和縮減資料庫執行個體類別，以套用對基礎作業系統的更新。

主控台

管理資料庫執行個體的更新內容

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇具有必要更新的資料庫執行個體。
4. 針對 Actions (動作)，選擇下列其中一項：
 - 立即升級
 - 在下一個時段升級

Note

如果您選擇 Upgrade at next window (在下一個時段升級)，但稍後想要延遲更新，則可以選擇 Defer upgrade (延遲升級)。如果維護動作已經啟動，則無法延遲維護動作。若要取消維護動作，請修改資料庫執行個體並停用 Auto minor version upgrade (自動次要版本升級)。

AWS CLI

若要將擱置中的更新套用至資料庫執行個體，請使用應用暫停維護 [AWS CLI](#) 動作指令。

Example

對於 Linux macOS、或 Unix：

```
aws rds apply-pending-maintenance-action \  
  --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \  
  --apply-action system-update \  
  --opt-in-type immediate
```

在 Windows 中：

```
aws rds apply-pending-maintenance-action ^  
  --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^  
  --apply-action system-update ^  
  --opt-in-type immediate
```

Note

若要延期維護動作，請指定 `undo-opt-in` 的 `--opt-in-type`。如果維護動作已經啟動，則無法指定 `undo-opt-in` 的 `--opt-in-type`。

若要取消維護動作，請執行 [modify-db-instance](#) AWS CLI 命令，並指定 `--no-auto-minor-version-upgrade`。

若要傳回至少有一個擱置中更新的資源清單，請使用[描述暫停維護動作命](#) AWS CLI 令。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-pending-maintenance-actions \  
  --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

在 Windows 中：

```
aws rds describe-pending-maintenance-actions ^  
  --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

您也可以指定 `describe-pending-maintenance-actions` AWS CLI 命令的 `--filters` 參數，傳回資料庫執行個體的資源清單。`--filters` 命令的格式為：`Name=filter-name, Value=resource-id, ...`。

以下為篩選條件中，`Name` 參數可接受的值：

- `db-instance-id` – 可接受資料庫執行個體識別符或 Amazon Resource Name (ARN) 的清單。在系統所傳回的資料庫執行個體待處理維護動作清單中，只會包含以這些識別符或 ARN 識別的項目。
- `db-cluster-id` – 可接受資料庫叢集識別符或 Amazon Aurora ARN 的清單。在系統所傳回的資料庫叢集待處理維護動作清單中，只會包含以這些識別符或 ARN 識別的項目。

例如，下方範例將傳回 `sample-instance1` 與 `sample-instance2` 資料庫執行個體的待處理維護動作。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-pending-maintenance-actions \  
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

在 Windows 中：

```
aws rds describe-pending-maintenance-actions ^  
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

RDS API

若要將更新內容套用至資料庫執行個體，請呼叫 Amazon RDS API

[ApplyPendingMaintenanceAction](#) 操作。

若要傳回至少具有一項待處理更新的資源清單，請呼叫 Amazon RDS API

[DescribePendingMaintenanceActions](#) 操作。

異地同步備份部署的維護

將資料庫執行個體當作多可用區部署來執行時，可以進一步減少維護事件的影響。此結果是因為 Amazon RDS 會遵循下列步驟套用作業系統更新：

1. 在待命時執行維護。
2. 將待命提升為主要。
3. 在原本的主要執行個體上執行維護作業，使其成為新的備用執行個體。

如果在多可用區部署中升級資料庫執行個體的資料庫引擎，則 Amazon RDS 會同時修改主要及次要資料庫執行個體。在此情況下，多可用區部署中的主要及次要資料庫執行個體在升級期間無法使用。此操作會造成停機，直到升級完成為止。停機持續時間依資料庫執行個體大小而有所不同。

如果需要套用基礎作業系統修補程式，則需要短暫的異地同步備份容錯移轉，才能將修補程式套用至主要資料庫執行個體。此容錯移轉通常持續時間不到一分鐘。

如果您的資料庫執行個體執行 RDS 版 MySQL、RDS PostgreSQL 或適用於 MariaDB 的 RDS，您可以使用藍/綠部署將升級所需的停機時間降到最低。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。如果您在異地同步備份部署中升級適用於 SQL Server 的 RDS 或適用於 SQL Server 的 RDS 自訂資料庫執行個體，則 Amazon RDS 會執行滾動式升級，因此您只會在容錯移轉期間發生中斷。如需詳細資訊，請參閱 [多可用區域和記憶體內最佳化的考量](#)。

如果您的資料庫執行個體在多可用區部署中執行 RDS for SQL Server，您可以使用下列其中一種方法，套用對基礎作業系統的更新：

- 將資料庫執行個體類別修改為不同的大小，然後將其修改回原始大小。
- 縱向擴展資料庫執行個體大小，然後縮減回原始大小。
- 將資料庫執行個體從多可用區修改為單一可用區、停止並啟動資料庫執行個體，然後將執行個體變更為多可用區。

如需異地同步備份部署的詳細資訊，請參閱 [設定及管理多可用區部署](#)。

Amazon RDS 維護時段

維護時段是每週套用任何系統變更的時間間隔。每個資料庫執行個體都有每週的維護時段。維護時段是控制何時進行修改和軟體修補的機會。

套用維護作業的期間，RDS 會使用資料庫執行個體上的部分資源。您可能會發現，該操作對效能會造成些許影響。在極少數情況下，資料庫執行個體需要執行異地同步備份容錯移轉，才能完成維護更新作業。

若在特定某週排定維護事件，系統將在指定的 30 分鐘維護時段內啟動該事件。此外，多數維護事件也能在 30 分鐘的維護時段內完成，但較大型的維護事件可能需要 30 分鐘以上才能完成。當資料庫執行個體停止時，維護時段會暫停。

30 分鐘的維護時段是從每個區域之 8 小時時段內隨機選取的。若您在資料庫執行個體建立期間，沒有指定維護時段，則 RDS 會在一週內隨機選取一天，並指派 30 分鐘的維護時段。

您可以在下方找到每個區域適用的時段，且系統會從中指派預設維護時段。

區域名稱	區域	時間區塊
美國東部 (俄亥俄)	us-east-2	上午 3 時至 11 時 (UTC)
美國東部 (維吉尼亞北部)	us-east-1	上午 3 時至 11 時 (UTC)
美國西部 (加利佛尼亞北部)	us-west-1	上午 6 時至下午 2 時 (UTC)
美國西部 (奧勒岡)	us-west-2	上午 6 時至下午 2 時 (UTC)
非洲 (開普敦)	af-south-1	上午 3 時至 11 時 (UTC)
亞太區域 (香港)	ap-east-1	06:00–14:00 UTC
亞太區域 (海德拉巴)	ap-south-2	06:30–14:30 UTC
亞太區域 (雅加達)	ap-southeast-3	08:00–16:00 UTC
亞太區域 (墨爾本)	ap-southeast-4	上午 11 時至下午 7 時 (UTC)
亞太區域 (孟買)	ap-south-1	上午 6 時至下午 2 時 (UTC)
亞太區域 (大阪)	ap-northeast-3	下午 10 時至 11 時 59 分 (UTC)
亞太區域 (首爾)	ap-northeast-2	下午 1 時至 9 時 (UTC)
亞太區域 (新加坡)	ap-southeast-1	下午 2 時至 10 時 (UTC)
亞太區域 (雪梨)	ap-southeast-2	中午 12 時至下午 8 時 (UTC)
亞太區域 (東京)	ap-northeast-1	下午 1 時至 9 時 (UTC)
加拿大 (中部)	ca-central-1	03:00–11:00 UTC
加拿大西部 (卡加利)	ca-west-1	18:00–02:00 UTC
中國 (北京)	cn-north-1	上午 6 時至下午 2 時 (UTC)
中國 (寧夏)	cn-northwest-1	上午 6 時至下午 2 時 (UTC)

區域名稱	區域	時間區塊
歐洲 (法蘭克福)	eu-central-1	下午 9 時至上午 5 時 (UTC)
歐洲 (愛爾蘭)	eu-west-1	下午 10 時至上午 6 時 (UTC)
歐洲 (倫敦)	eu-west-2	22:00–06:00 UTC
歐洲 (米蘭)	eu-south-1	上午 2 時至 10 時 (UTC)
歐洲 (巴黎)	eu-west-3	下午 11 時 59 分至上午 7 時 29 分 (UTC)
歐洲 (西班牙)	eu-south-2	上午 2 時至 10 時 (UTC)
歐洲 (斯德哥爾摩)	eu-north-1	下午 11 時至上午 7 時 (UTC)
歐洲 (蘇黎世)	eu-central-2	上午 2 時至 10 時 (UTC)
以色列 (特拉維夫)	il-central-1	03:00–11:00 UTC
中東 (巴林)	me-south-1	06:00–14:00 UTC
中東 (阿拉伯聯合大公國)	me-central-1	上午 5 時至下午 1 時 (UTC)
南美洲 (聖保羅)	sa-east-1	上午 12 時至 8 時 (UTC)
AWS GovCloud (美國東部)	us-gov-east-1	下午 5 時至上午 1 時 (UTC)
AWS GovCloud (美國西部)	us-gov-west-1	上午 6 時至下午 2 時 (UTC)

調整偏好的資料庫執行個體維護時段

維護時段應落在使用量最低的時段，因此可能需要不時進行調整。除非系統套用變更 (例如資料庫執行個體類別中的變更) 及要求停機，否則這段時間無法使用您的叢集或執行個體。無法使用資料庫執行個體的時間，僅為執行必要變更所需的最短時間。

在以下範例中，您可以針對資料庫執行個體調整偏好的維護時段。

對於此範例，我們將假設名為 `mydbinstance` 的資料庫執行個體存在，且偏好的維護時段為「UTC 時間週日 05:00 至週日 06:00」。

主控台

調整偏好的維護時段

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選取您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 在 Maintenance (維護) 區段中，更新維護時段。

Note

資料庫執行個體的維護時段和備份時段不能重疊。若您輸入的維護時段值與備份時段重疊，即會出現錯誤訊息。

5. 選擇 Continue (繼續)。

在確認頁面上，檢閱您的變更。

6. 選取 Apply immediately (立即套用)，就能立刻將變更套用至維護時段。
7. 選擇修改資料庫執行個體以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要調整偏好的維護時段，請使用具有下列參數的 AWS CLI [modify-db-instance](#) 指令：

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

在下列程式碼範例中，系統會將維護時段設為 UTC 時間週二上午 4:00 至 4:30。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \  
--db-instance-identifier mydbinstance \  
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

在 Windows 中：

```
aws rds modify-db-instance ^  
--db-instance-identifier mydbinstance ^  
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

RDS API

若要調整偏好的維護時段，請搭配下列參數執行 Amazon RDS API [ModifyDBInstance](#) 操作：

- DBInstanceIdentifier
- PreferredMaintenanceWindow

使用強制作業系統更新

適用於 Db2 的 RDS、適用於 MariaDB 的 RDS、MySQL 版 RDS、適用於 PostgreSQL 的 RDS 以及適用於甲骨文資料庫執行個體的 RDS 偶爾需要作業系統更新。Amazon RDS 將作業系統升級至較新版本，以改善資料庫效能和客戶的整體安全狀態。通常，更新大約需要 10 分鐘。作業系統更新不會變更資料庫執行個體的資料庫引擎版本或資料庫執行個體類別。

作業系統更新可以是選用的，也可以是強制的。

- 可以隨時套用選用更新。雖然這些更新是選用的，但建議您定期套用更新，讓 RDS 機群保持最新狀態。RDS 不會自動套用這些更新。

若要在新的選用作業系統修補程式可用時收到通知，您可以訂閱安全修補事件類別中的 [RDS-EVENT-0230](#)。如需訂閱 RDS 事件的相關資訊，請參閱 [訂閱 Amazon RDS 事件通知](#)。

Note

RDS-EVENT-0230 不適用於作業系統發行版升級。

Note

如果您收到 RDS for SQL Server 資料庫執行個體的 RDS-EVENT-0230，則無法透過 `apply-pending-maintenance` 動作套用作業系統更新。如需詳細資訊，請參閱 [套用資料庫執行個體的更新](#)。

- 強制更新為必要的更新，且有套用日期。請規劃將更新排程在此套用日期之前。在指定的套用日期之後，Amazon RDS 會在您指派的其中一個維護時段期間自動將資料庫執行個體的作業系統升級至最新版本。

Note

為了履行各種合規義務，可能需要將所有選用與強制更新保持為最新的狀態。建議您在維護時段期間定期套用 RDS 提供的所有更新。

您可以使用 AWS Management Console 或取 AWS CLI 得作業系統升級類型的相關資訊。

主控台

若要取得更新資訊，請使用 AWS Management Console

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選取資料庫執行個體。
3. 選擇 Maintenance & backups (維護和備份)。
4. 在待定維護區段中，尋找作業系統更新，然後檢查狀態值。

在中 AWS Management Console，選擇性更新的維護狀態已設定為可用，而且沒有套用日期，如下圖所示。

The screenshot shows the 'Maintenance & backups' tab in the AWS console. Under the 'Maintenance' section, the 'Pending maintenance' status is 'available'. Below this, a table lists pending maintenance actions:

Description	Type	Status	Apply date
New Operating System update is available	system-update	available	-

強制更新將其維護 Status (狀態) 設定為 required (必要) , 且具有 Apply date (套用日期) , 如下圖所示。

The screenshot shows the 'Maintenance & backups' tab in the AWS console. Under the 'Maintenance' section, the 'Pending maintenance' status is 'required'. Below this, a table lists pending maintenance actions:

Description	Type	Status	Apply date
New Operating System update is available	system-update	required	August 31, 2022, 12:00:00 AM UTC

AWS CLI

若要從中取得更新資訊 AWS CLI , 請使用 [描述](#)- 暫停維護動作指令。

```
aws rds describe-pending-maintenance-actions
```

強制作業系統更新包括 AutoAppliedAfterDate 值與 CurrentApplyDate 值。選用作業系統更新未包含這些值。

以下輸出顯示強制性作業系統更新。

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
```

```
"AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
"CurrentApplyDate": "2022-08-31T00:00:00+00:00",
"Description": "New Operating System update is available"
}
]
}
```

以下輸出顯示選用作業系統更新。

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

作業系統更新的可用性

作業系統更新為資料庫引擎版本與資料庫執行個體類別所特有。因此，資料庫執行個體會在不同的時間接收或需要更新。當根據引擎版本和執行個體類別，有作業系統更新可供您的資料庫執行個體使用時，該更新會出現在主控台中。它也可以通過運行 AWS CLI [描述-暫停維護操作命令](#)或通過調用 [RDS API 操作](#)來查看。[DescribePendingMaintenanceActions](#) 如果執行個體有可用的更新，您可以依照[套用資料庫執行個體的更新](#)中的指示更新作業系統。

升級資料庫執行個體引擎版本

Amazon RDS Amazon 提供每個受支援的資料庫引擎的更新版本，因此您可以保留資料庫執行個體 up-to-date。較新版本包含資料庫引擎的錯誤修正、安全性強化和其他改善。Amazon RDS 支援新版本的資料庫引擎時，您可以選擇如何和何時升級資料庫執行個體。

升級有兩種：主要版本升級和次要版本升級。一般而言，主要引擎版本升級可能引進與現有應用程式不相容的變更。反之，次要版本升級只包含與現有應用程式回溯相容的變更。

若是多可用區域資料庫叢集，只有 RDS for PostgreSQL 可支援主要版本升級。支援多可用區域資料庫叢集的所有引擎都支援次要版本升級。如需詳細資訊，請參閱 [the section called “升級多可用區域資料庫叢集的引擎版本”](#)。

每個資料庫引擎有明確的版本編號順序。例如，RDS for MySQL 5.7 和 8.0 是主要引擎版本，而從任何 5.7 版升級到任何 8.0 版是主要版本升級。RDS for MySQL 5.7.22 和 5.7.23 版是次要版本，而從 5.7.22 升級到 5.7.23 是次要版本升級。

Important

升級資料庫執行個體時，您無法修改資料庫執行個體。升級期間，資料庫執行個體狀態為 upgrading。

如需特定資料庫引擎的主要和次要版本升級的詳細資訊，根據您的資料庫引擎而定，請參閱以下文件：

- [升級 MariaDB 資料庫引擎](#)
- [升級 Microsoft SQL Server 資料庫引擎](#)
- [升級 MySQL 資料庫引擎](#)
- [升級 RDS for Oracle 資料庫引擎](#)
- [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)

對於主要版本升級，您必須透過、或 RDS API 手動修改資料庫引擎版本。AWS Management Console AWS CLI 若是次要版本升級，您可以手動修改引擎版本，或選擇啟用自動次要版本升級選項。

Note

資料庫引擎升級需要停機。您可以使用藍/綠部署，將資料庫執行個體升級所需的停機時間降至最低。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

主題

- [手動升級引擎版本](#)
- [自動升級次要引擎版本](#)

手動升級引擎版本

若要手動升級資料庫執行個體的引擎版本 AWS Management Console，您可以使用 AWS CLI、或 RDS API。

主控台

使用主控台升級資料庫執行個體的引擎版本

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要升級的資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 在 DB engine version (資料庫引擎版本) 中，選擇新版本。
5. 選擇 Continue (繼續)，並檢查修改的摘要。
6. 決定何時排定升級。若要立即套用變更，請選擇 Apply immediately (立即套用)。在某些情況下，選擇此選項會導致停機。如需更多詳細資訊，請參閱 [排程修改設定](#)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB instance (修改資料庫執行個體) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要升級資料庫執行個體的引擎版本，請使用 CLI [modify-db-instance](#) 命令。指定下列參數：

- `--db-instance-identifier` – 資料庫執行個體名稱。
- `--engine-version` – 會以此資料庫引擎版本編號為目標進行升級。

如需有效引擎版本的相關資訊，請使用 AWS CLI [describe-db-engine-versions](#) 指令。

- `--allow-major-version-upgrade` – 升級主要版本。
- `--no-apply-immediately` – 在下次維護時段套用變更。若要立即套用變更，請使用 `--apply-immediately`。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --engine-version new_version \  
  --allow-major-version-upgrade \  
  --no-apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --engine-version new_version ^  
  --allow-major-version-upgrade ^  
  --no-apply-immediately
```

RDS API

若要升級資料庫執行個體的引擎版本，請使用 [ModifyDBInstance](#) 動作。指定下列參數：

- `DBInstanceIdentifier` – 資料庫執行個體的名稱，例如 *mydbinstance*。
- `EngineVersion` – 會以此資料庫引擎版本編號為目標進行升級。如需有效引擎版本的相關資訊，請使用 [描述 B 作EngineVersions](#) 業。
- `AllowMajorVersionUpgrade` – 是否允許主要版本升級。若要這麼做，請設為 `true` 值。
- `ApplyImmediately` – 指出是否要立即套用變更，或等到下個維護時段再套用。若要立即套用變更，請將值設為 `true`。若要在下一次維護時段套用變更，請將值設為 `false`。

自動升級次要引擎版本

次要引擎版本是在主要引擎版本內對於資料庫引擎版本的更新。例如，主要引擎版本可能是 9.6，而其內部有次要引擎版本 9.6.11 和 9.6.12。

如果要讓 Amazon RDS 自動升級資料庫的資料庫引擎版本，您可以對該資料庫啟用自動次要版本升級。

適用於 SQL 伺服器的 RDS 目前不支援自動次要版本更新。

主題

- [自動次要版本升級的運作方式](#)
- [開啟自動次要版本升級](#)
- [判斷維護更新的可用性](#)
- [尋找自動次要版本升級目標](#)

自動次要版本升級的運作方式

當符合下列條件時，Amazon RDS 會將次要引擎版本指定為偏好的次要引擎版本：

- 資料庫執行的資料庫引擎次要版本低於偏好的次要引擎版本。

您可以透過查看資料庫詳細資訊頁面的組態，以尋找目前資料庫執行個體的引擎版本，或是執行 CLI 命令 `describe-db-instances`。

- 資料庫已啟用自動次要版本升級。

RDS 會排程升級在維護時段自動執行。在自動升級期間，RDS 會執行下列基本步驟：

1. 執行預先檢查，確定資料庫狀況良好且可以升級
2. 升級資料庫引擎
3. 執行升級後任務
4. 將資料庫升級標示為完成

自動升級會導致停機。停機時間的長度取決於多種因素，包括資料庫引擎類型和資料庫的大小。

開啟自動次要版本升級

當您執行以下任務時，您可以控制是否對資料庫執行個體啟用自動次要版本升級：

- [建立資料庫執行個體](#)
- [修改資料庫執行個體](#)
- [建立僅供讀取複本](#)
- [從快照還原資料庫執行個體](#)
- [將資料庫執行個體還原至指定的時間](#)
- [從 Amazon S3 匯入資料庫執行個體](#) (適用於 Amazon S3 上的 MySQL 備份)

當您執行以下任務時，您可以控制是否對資料庫執行個體啟用自動次要版本升級，方法如下：

- 使用主控台，設定 Auto minor version upgrade (自動次要版本升級) 選項。
- 使用 AWS CLI，設定選 `--auto-minor-version-upgrade` | `--no-auto-minor-version-upgrade` 項。
- 使用 RDS API，設定 `AutoMinorVersionUpgrade` 參數。

判斷維護更新的可用性

若要判斷資料庫執行個體是否有維護更新 (例如資料庫引擎版本升級)，您可以使用主控台或 RDS API。AWS CLI 您也可以手動升級資料庫引擎版本，並調整維護時段。如需詳細資訊，請參閱 [維持資料庫執行個體](#)。

尋找自動次要版本升級目標

您可以使用下列 AWS CLI 命令來判斷特定中指定次要資料庫引擎版本的目前自動次要升級目標版本 AWS 區域。這個命令可能的 `--engine` 值，列在 [CreateDBInstance](#) 內 `Engine` 參數的說明中。

對於 Linux macOS、或 Unix：

```
aws rds describe-db-engine-versions \  
--engine engine \  
--engine-version minor-version \  
--region region \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  

```

```
--output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^
--engine engine ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

例如，下列 AWS CLI 命令會決定美國東部 (俄亥俄) AWS 區域 (us-east-2) 中 MySQL 次要版本 8.0.11 的自動次要升級目標。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

輸出類似如下。

```
-----
| DescribeDBEngineVersions |
+-----+-----+
| AutoUpgrade | EngineVersion |
```

```
+-----+-----+
| False   | 8.0.15   |
| False   | 8.0.16   |
| False   | 8.0.17   |
| False   | 8.0.19   |
| False   | 8.0.20   |
| False   | 8.0.21   |
| True    | 8.0.23 |
| False   | 8.0.25   |
+-----+-----+
```

在此範例中，此 AutoUpgrade 值是 True (若為 MySQL 版本 8.0.23)。因此，自動次要升級目標是 MySQL 版本 8.0.23，其已在輸出中反白顯示。

Important

如果您打算在不久將 RDS for PostgreSQL 資料庫執行個體遷移至 Aurora PostgreSQL 資料庫叢集，我們強烈建議您在規劃階段的早期關閉資料庫執行個體的自動次要版本升級。如果 RDS for PostgreSQL 版本尚未得到 Aurora PostgreSQL 支援，遷移至 Aurora PostgreSQL 可能會延遲。如需 Aurora PostgreSQL 版本的相關資訊，請參閱 [Amazon Aurora PostgreSQL 的引擎版本](#)。

重新命名資料庫執行個體

您可以使用 AWS Management Console、AWS CLI `modify-db-instance` 命令或 Amazon RDS API `ModifyDBInstance` 動作重新命名資料庫執行個體。重新命名資料庫執行個體可以發揮深遠的效果。以下是您重新命名資料庫執行個體前的考量事項清單。

- 重新命名資料庫執行個體時，資料庫執行個體的端點會隨之變更，因為 URL 包含您指派給資料庫執行個體的名稱。您應該將流量從舊的 URL 重新導向到新的 URL。
- 重新命名資料庫執行個體時，資料庫執行個體所用的舊 DNS 名稱會立即刪除，不過其會保留在快取內幾分鐘。重新命名的資料庫執行個體的新 DNS 名稱會在大約 10 分鐘後生效。重新命名的資料庫執行個體必須等到新名稱生效後才可使用。
- 重新命名執行個體後，您將無法使用現有的資料庫執行個體名稱。
- 與資料庫執行個體關聯的所有僅供讀取複本在重新命名後仍會保持與該執行個體的關聯。例如，假設您有一個為生產資料庫服務的資料庫執行個體，且該執行個體有多個關聯的僅供讀取複本。如果您命名資料庫執行個體，接著在生產環境中將執行個體換成資料庫快照，重新命名後的資料庫執行個體仍有與其關聯的僅供讀取複本。
- 如果您重複使用資料庫執行個體名稱，與資料庫執行個體名稱關聯的指標和事件也會保留。例如，如果您提升僅供讀取複本，並將其重新命名為與先前主要資料庫執行個體相同的名稱，與主要資料庫執行個體相關聯的事件和指標將變成與重新命名後的執行個體相關聯。
- 資料庫執行個體標籤隨資料庫執行個體保留，無論是否重新命名。
- 重新命名的資料庫執行個體將保留其資料庫快照。

Note

資料庫執行個體是在雲端執行的隔離資料庫環境。資料庫執行個體可以託管多個資料庫，也可以託管具有多個結構描述的單一 Oracle 資料庫。如需有關變更資料庫名稱的資訊，請參閱資料庫引擎相關文件。

重新命名以取代現有的資料庫執行個體

重新命名資料庫執行個體的最常見原因是您要升級僅供讀取複本，或是從資料庫快照或 point-in-time 復原 (PITR) 還原資料。透過重新命名資料庫的方式，您便能取代資料庫執行個體，而無需變更參考資料庫執行個體的任何應用程式碼。在這些情況下，您需執行下列動作：

1. 停止傳入主要資料庫執行個體的所有流量。這牽涉到重新導向流量，避免其存取資料庫執行個體上的資料庫，或您可用某些其他方式來避免流量存取資料庫執行個體上的資料庫。
2. 重新命名主要資料庫執行個體，名稱應指出其已不再是主要資料庫執行個體，如本主題稍後說明。
3. 從資料庫快照還原，或提升僅供讀取複本，以建立新的主要資料庫執行個體，然後將新執行個體命名為先前主要資料庫執行個體的名稱。
4. 將任何僅供讀取複本與新的主要資料庫執行個體建立關聯。

如果要刪除舊的主要資料庫執行個體，您也必須負責刪除舊的主要資料庫執行個體任何不想要的資料庫快照。

如需提升僅供讀取複本的相關資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

Important

資料庫執行個體會重新命名後重新開機。

主控台

重新命名資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想要重新命名的資料庫執行個體。
4. 選擇 Modify (修改)。
5. 在 Settings (設定) 中，針對 DB instance identifier (資料庫執行個體識別符)，輸入新名稱。
6. 選擇 Continue (繼續)。
7. 若要立即套用變更，請選擇 Apply immediately (立即套用)。在某些情況下，選擇此選項會導致停機。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
8. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB Instance (修改資料庫執行個體) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要重新命名資料庫執行個體，請使用 AWS CLI 命令 [modify-db-instance](#)。用資料庫執行個體的新名稱提供目前的 `--db-instance-identifier` 值和 `--new-db-instance-identifier` 參數。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier DBInstanceIdentifier \  
  --new-db-instance-identifier NewDBInstanceIdentifier
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier DBInstanceIdentifier ^  
  --new-db-instance-identifier NewDBInstanceIdentifier
```

RDS API

若要重新命名資料庫執行個體，請搭配下列參數呼叫 Amazon RDS API 操作 [ModifyDBInstance](#)：

- `DBInstanceIdentifier` — 執行個體的目前名稱
- `NewDBInstanceIdentifier` — 執行個體的新名稱

重新啟動中的資料庫執行個體

您可以在 RDS 資料庫執行個體上停止和啟動資料庫服務，稱為重新開機的單一作業。

Note

本主題僅適用於重新啟動資料庫執行個體。如需重新啟動異地同步備份資料庫叢集的指示，請參閱[the section called “重新啟動多可用區域資料庫叢集”](#)。

主題

- [在資料庫體的使用案例](#)
- [重新啟動資料庫體的運作](#)
- [異地同步備份部署中的資料庫執行個體重新啟動方式](#)
- [在資料庫體的考量](#)
- [在資料庫體的必要](#)
- [重新啟動資料庫體：基本步驟](#)

在資料庫體的使用案例

通常，出於維護原因，您將資料庫執行個體重新開機，以便變生效。以下是常見的使用案例：

- 建立新資料庫參數群組關聯 — 當您將新的資料庫參數群組與資料庫執行個體建立關聯時，RDS 只會在資料庫執行個體重新啟動後套用修改後的靜態和動態參數。但是，如果您在將資料庫參數群組與資料庫執行個體產生關聯之後修改動態參數，則這些變更會立即套用，而無需重新開機。如需詳細資訊，請參閱 [使用參數群組](#)。
- 將變更套用至現有資料庫參數群組中的靜態參數 — 變更靜態參數並儲存資料庫參數群組時，與主控台中此參數群組相關聯的資料庫執行個體狀態會變更為待處理重新開機。只有在關聯的資料庫執行個體重新啟動後，參數變更才會生效。當您變更現有參數群組中的動態參數時，變更依預設會立即生效，而不需要重新開機。

Note

擱置中重新啟動狀態不會導致在下一個維護時段期間自動重新啟動。若要將最新參數變更套用至資料庫執行個體，請手動重新啟動資料庫執行個體。如需參數群組的詳細資訊，請參閱[使用參數群組](#)。

- 測試異地同步備份容錯移轉 — 異地同步備份資料庫叢集的測試策略可能需要重新啟動主資料庫執行個體，以啟動容錯移轉至其他可用區域。
- 疑難排解 — 您可能遇到需要重新開機的效能或其他作業問題。例如，您的資料庫執行個體可能沒有回應。

重新啟動資料庫體的運作

Amazon RDS 重新啟動資料庫執行個體時，會執行下列順序任務：

1. 停止資料庫執行個體上的資料庫服務
2. 啟動資料庫執行個體上的資料庫服務

重新啟動過程會導致短暫中斷。在此中斷期間，資料庫執行個體狀態會重新啟動。單一可用區域部署和多可用區域資料庫執行個體部署都會發生中斷，即使您使用容錯移轉來重新開機也一樣。

異地同步備份部署中的資料庫執行個體重新啟動方式

如果 Amazon RDS 資料庫執行個體位於異地同步備份部署中，您可以透過容錯移轉重新開機。此作業對於模擬資料庫執行個體故障或在容錯移轉後將作業還原至原始可用區域非常有用。

在透過容錯移轉進行重新開機期間，Amazon RDS 會執行下列

- 突然中斷資料庫。資料庫執行個體及其用戶端工作階段可能沒有足夠時間正常地關閉。

Warning

為避免資料遺失的可能性，建議您在進行容錯移轉並重新啟動之前，先停止資料庫執行個體上的交易。

- 自動切換到另一個 AZ 中的備用複本。AZ 變更可能在數分鐘內 AWS Management Console 未反映在 AWS CLI 和 RDS API 的呼叫中。

- 更新資料庫執行個體的 DNS 記錄，以指向待命資料庫執行個體。因此，您需要清除和重新建立資料庫執行個體任何現有的連線。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。
- 在重新開機後建立 Amazon RDS 事件。

在適用於 Microsoft SQL Server 的 RDS 上，容錯移轉只會重新啟動主資料庫執行個體。在容錯移轉之後，主要資料庫執行個體會成為新的次要資料庫執行個體。多可用區域執行個體的參數可能無法更新。若要在無容錯移轉的情況下重新啟動，主要和次要資料庫執行個體會在重新啟動後更新參數。若資料庫執行個體沒有回應，建議您重新開機，不要進行容錯移轉。

在資料庫體的考量

重新啟動執行個體之前，請考慮下列事項：

- 對於具有僅供讀取複本的資料庫執行個體，您可以獨立重新啟動來源資料庫執行個體及其僅供讀取複本。重新開機完成後，複寫會自動繼續。
- 重新開機時間取決於當機復原程序、重新開機時的資料庫活動，以及特定資料庫引擎的行為。若要縮短重新開機時間，建議您在重新開機期間盡量減少資料庫活動。此技術可減少在途交易的回復活動。

在資料庫體的必要

請確定您符合下列先決條件：

- 您的資料庫執行個體必須處於 available 狀態。您的資料庫可能因為多種原因而無法使用，例如進行中的備份、先前要求的修改或維護時段作業。
- 如果您強制容錯移轉到不同的可用區域，您的資料庫執行個體必須設定為異地同步備份。
- 如果您強制容錯移轉到不同的 AZ，建議您先停止資料庫執行個體上的交易，以防止可能的資料遺失。

重新啟動資料庫體：基本步驟

您可以使用 AWS Management Console、AWS CLI 或 RDS API 重新啟動資料庫執行個體。

主控台

重新啟動資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要重新啟動的資料庫執行個體。
3. 針對 Actions (動作)，選擇 Reboot (重新啟動)。

[重新啟動資料庫執行個體] 頁面

4. (選用) 選擇 Reboot with failover? (重新啟動並容錯移轉?)，以強制容錯移轉到另一個 AZ。
5. 選擇 Reboot (重新啟動)，以重新啟動您的資料庫執行個體。

或者，選擇 Cancel (取消)。

AWS CLI

若要使用重新啟動資料庫執行個體 AWS CLI，請呼叫 [reboot-db-instance](#) 命令。

Example 簡單的重新啟動

對於LinuxmacOS、或Unix：

```
aws rds reboot-db-instance \  
  --db-instance-identifier mydbinstance
```

在 Windows 中：

```
aws rds reboot-db-instance ^  
  --db-instance-identifier mydbinstance
```

Example 使用容錯移轉重新啟動

若要在異地同步備份資料庫叢集中強制容錯移轉從一個 AZ 到另一個 AZ，請使用 `--force-failover` 參數。

對於LinuxmacOS、或Unix：

```
aws rds reboot-db-instance \  
  --force-failover
```

```
--db-instance-identifier mydbinstance \  
--force-failover
```

在 Windows 中：

```
aws rds reboot-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --force-failover
```

RDS API

若要使用 Amazon RDS API 重新啟動資料庫執行個體，請呼叫 [RebootDBInstance](#) 操作。

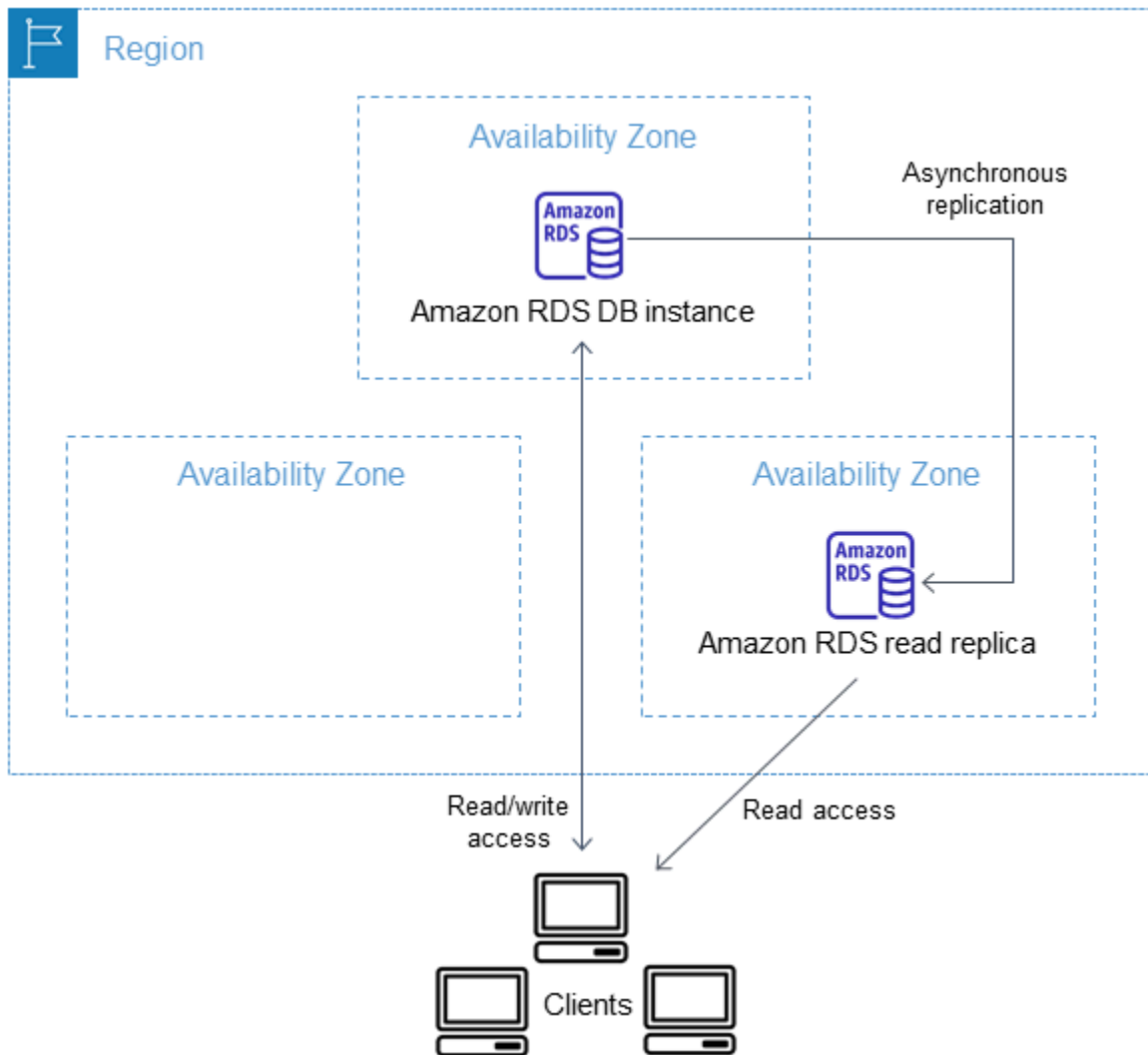
使用資料庫執行個體僅供讀取複本

僅供讀取複本是資料庫執行個體的唯一讀副本。您可以將查詢從應用程式路由到僅供讀取複本，以減少主要資料庫執行個體的負載。如此一來，您可以彈性地擴展超越單一資料庫執行個體的容量限制，以處理高讀取量的資料庫工作負載。

若要從來源資料庫執行個體建立僅供讀取複本，Amazon RDS 會使用資料庫引擎的內建複寫功能。如需搭配特定引擎使用僅供讀取複本的資訊，請參閱下列各節：

- [使用 MariaDB 僅供讀取複本](#)
- [使用 Amazon RDS 中的 Microsoft SQL Server 的僅供讀取複本](#)
- [使用 MySQL 僅供讀取複本](#)
- [使用適用於 Amazon RDS for Oracle 的僅供讀取複本](#)
- [使用 Amazon RDS for PostgreSQL 的僅供讀取複本](#)

在您從來源資料庫執行個體建立僅供讀取複本之後，來源會成為主要資料庫執行個體。當您對主要資料庫執行個體進行更新時，Amazon RDS 會以非同步方式將它們複製到僅供讀取複本。下圖顯示複寫到不同可用區域 (AZ) 中僅供讀取複本的來源資料庫執行個體。用戶端具有主要資料庫執行個體的讀取/寫入存取權，以及複本的唯一讀存取



主題

- [Amazon RDS 僅供讀取複本的概觀](#)
- [建立僅供讀取複本](#)
- [提升僅供讀取複本為獨立的資料庫執行個體](#)
- [監控僅供讀取複本](#)
- [以不同的方式建立僅供讀取複本 AWS 區域](#)

Amazon RDS 僅供讀取複本的概觀

下列各節討論資料庫執行個體僅供讀取複本。如需多可用區域資料庫叢集僅供讀取複本的詳細資訊，請參閱 [the section called “使用多可用區域資料庫叢集僅供讀取複本”](#)。

主題

- [僅供讀取複本的使用案例](#)
- [僅供讀取複本的運作方式](#)
- [多可用區部署中的僅供讀取複本](#)
- [跨區域僅供讀取複本](#)
- [資料庫引擎的僅供讀取複本之間的差異](#)
- [僅供讀取複本儲存類型](#)
- [從複本建立複本的限制](#)
- [刪除複本時的考量](#)

僅供讀取複本的使用案例

為指定的來源資料庫執行個體部署一或多個僅供讀取複本，在各種鏈中可能都有意義，包括：

- 擴展超越單一資料庫執行個體的運算或 I/O 容量，以處理高讀取量的資料庫工作負載。您可以將此多餘讀取流量指向到一或多個僅供讀取複本。
- 在來源資料庫執行個體不可用時，為讀取流量提供服務。在某些情況下，如果您的來源資料庫執行個體無法處理 I/O 請求 (例如，由於備份或計劃維護而暫停 I/O)。在這些情況下，您可以將讀取流量導向到您的僅供讀取複本。請注意，在這種使用案例中，因為來源資料庫執行個體不可用，僅供讀取複本中的資料可能為「過時」狀態。
- 商業報告或資料倉儲案例，您可能希望針對僅供讀取複本執行商業報告查詢，而非針對生產資料庫執行個體。
- 實作災難復原。您可以在主要資料庫執行個體失敗時，提升僅供讀取複本為獨立的執行個體，以做為災難復原解決方案。

僅供讀取複本的運作方式

建立僅供讀取複本時，應先指定現有資料庫執行個體做為來源。然後 Amazon RDS 會取得來源執行個體的快照，並從該快照建立唯讀執行個體。Amazon RDS 會使用非同步複寫方法，以供資料庫引擎更新僅供讀取複本，而不論主要資料庫執行個體是否有變更。

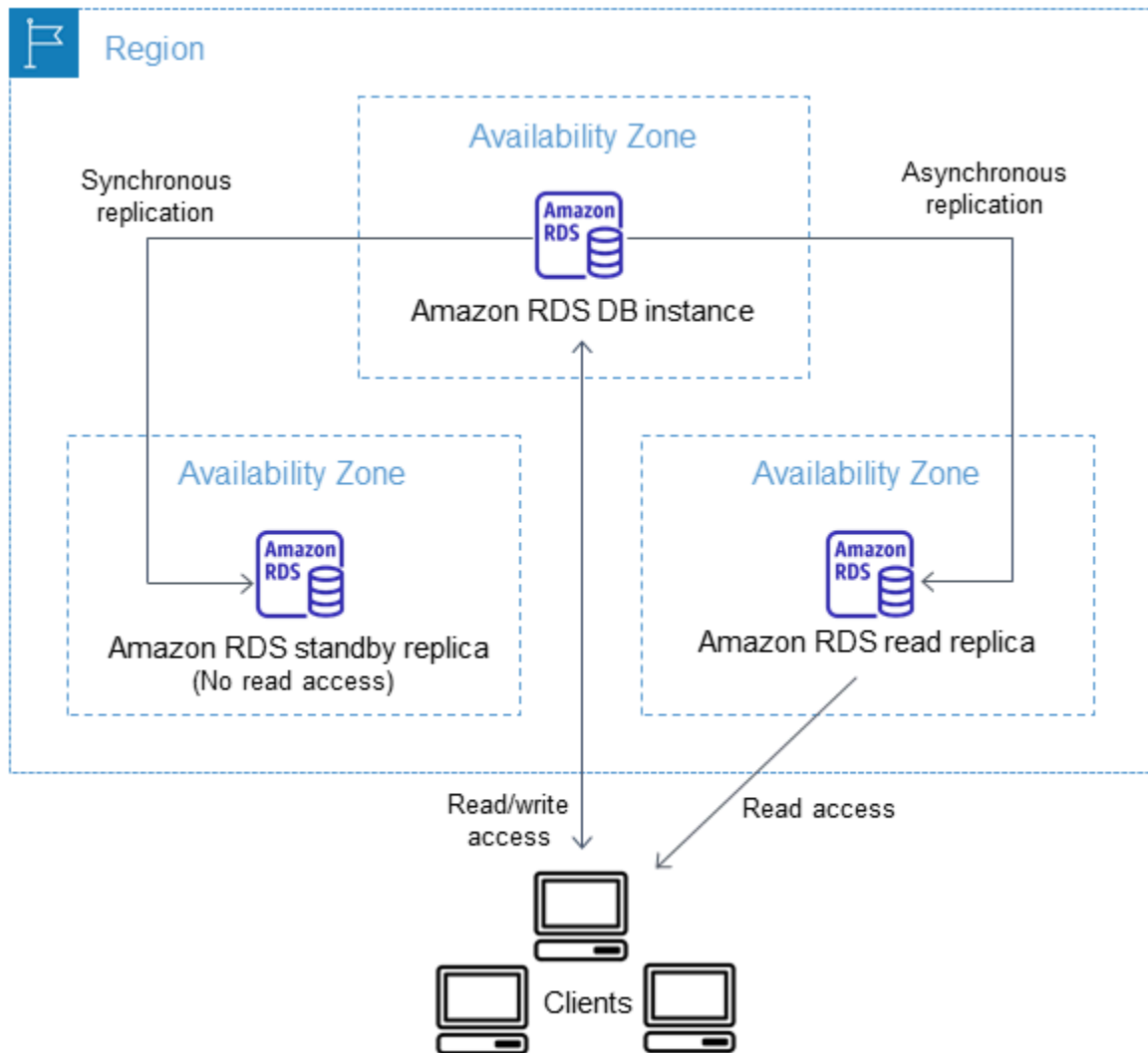
僅供讀取複本做為資料庫執行個體運作，並允許唯讀連接。範例是 RDS for Oracle 資料庫引擎，其支援掛載模式中的複本資料庫。掛載複本不接受使用者連線，因此無法提供唯讀工作負載。掛載複本的主要用途是跨區域災難復原。如需詳細資訊，請參閱 [使用適用於 Amazon RDS for Oracle 的僅供讀取複本](#)。

應用程式會連線至僅供讀取複本，如同它們對任何資料庫執行個體所做一般。Amazon RDS 會複寫來源資料庫執行個體中的所有資料庫。

多可用區部署中的僅供讀取複本

您可以在異地同步備份部署中，為資料庫執行個體設定僅供讀取複本，該執行個體還具有設定為高可用性的待命複本。使用待命複本進行複寫是同步的。與僅供讀取複本不同，待命複本無法提供讀取流量。

在下列案例中，用戶端對某個 AZ 中的主要資料庫執行個體具有讀取/寫入存取權。主要執行個體會以非同步方式將更新複製到第二個 AZ 中的僅供讀取複本，並以同步方式將它們複製到第三個 AZ 中的待命複本。用戶端對僅供讀取複本只有讀取存取權。



如需高可用性和待命複本的相關資訊，請參閱[設定及管理多可用區部署](#)。

跨區域僅供讀取複本

在某些情況下，僅供讀取複本位於與其主資料庫執行個體不同 AWS 區域 的位置。在這些情況下，Amazon RDS 會在主要資料庫執行個體與僅供讀取複本之間設定安全通訊管道。Amazon RDS 會建立啟用 AWS 安全通道所需的任何安全組態，例如新增安全群組項目。如需跨區域僅供讀取複本的相關資訊，請參閱[以不同的方式建立僅供讀取複本 AWS 區域](#)。

本章中的資訊適用於在與來源資料庫執行個體相 AWS 區域 同或單獨 AWS 區域建立 Amazon RDS 僅供讀取複本。以下資訊並不適用於在 Amazon EC2 執行個體上執行的執行個體，或內部部署的執行個體上設定複寫。

資料庫引擎的僅供讀取複本之間的差異

由於 Amazon RDS 資料庫引擎實作複寫方式的不同，有幾點重大差異您應該要知道，如下表所示。

功能或行為	MySQL 和 MariaDB	Oracle	PostgreSQL	SQL Server
什麼是複寫方法？	邏輯複寫。	實體複寫。	實體複寫。	實體複寫。
如何清除交易日誌？	RDS for MySQL 和 RDS for MariaDB 會保留任何未套用的二進位日誌。	如果主要資料庫執行個體沒有跨區域僅供讀取複本，Amazon RDS for Oracle 在來源資料庫執行個體上會保留至少兩小時的交易日誌。兩小時之後，或過了存檔日誌保留時數設定後 (視何者較長而定)，將從來源資料庫執行個體中清除日誌。只有在日誌已成功套用至資料庫後，才會在過了存檔日誌保留時數設定之後從僅供讀取複本中清除日誌。	PostgreSQL 有 <code>wal_keep_segments</code> 參數指出保留多少個預先寫入日誌 (WAL) 檔案，以提供檔案給僅供讀取複本。參數值指定了要保留的日誌數量。	如果不再需要次要複本，便會截斷主要複本上交易記錄檔的虛擬記錄檔 (VLF)。在複本中強化記錄檔記錄時，只能將 VLF 標示為非作用中。無論主要複本中的磁碟子系統有多快，交易記錄檔

功能或行為	MySQL 和 MariaDB	Oracle	PostgreSQL	SQL Server
		<p>在某些情況下，一個主要資料庫執行個體可能有一或多個跨區域僅供讀取複本。如果是這樣，Amazon RDS for Oracle 會將交易日誌保留在來源資料庫執行個體上，直到傳輸並套用至所有跨區域僅供讀取複本為止。</p> <p>如需設定封存記錄保留時數的相關資訊，請參閱保留封存的redo日誌。</p>		<p>都會保留 VLF，直到速度最慢的複本強化它為止。</p>
能將複本變成可寫入嗎？	<p>是。您可將 MySQL 或 MariaDB 僅供讀取複本變成可寫入的。</p>	<p>否。Oracle 僅供讀取複本是實體複製，且 Oracle 並不允許在僅供讀取複本中進行寫入。您可提升僅供讀取複本以將其變為可寫入。提升的僅供讀取副本會將複製的資料置於提升請求提出的時機。</p>	<p>否。PostgreSQL 僅供讀取複本為實體複本，且 PostgreSQL 不允許將僅供讀取複本變成可寫入。</p>	<p>否。SQL Server 僅供讀取複本是實體副本，也不允許寫入。您可提升僅供讀取複本以將其變為可寫入。提升的僅供讀取副本會將複製的資料置於提升請求提出的時機。</p>

功能或行為	MySQL 和 MariaDB	Oracle	PostgreSQL	SQL Server
能否在複本上執行備份？	是。RDS for MySQL 或 RDS for MariaDB 僅供讀取複本上支援自動備份和手動快照。	是。RDS for Oracle 僅供讀取複本上支援自動備份和手動快照。	是，您可以建立 RDS for PostgreSQL 僅供讀取複本的手動快照。RDS for PostgreSQL 14.1 及更新版本僅支援僅供讀取複本的自動備份。您無法為早於 14.1 的 RDS for PostgreSQL 版本開啟 PostgreSQL 僅供讀取複本的自動備份。對於 RDS for PostgreSQL 13 及更早版本，若您想要備份僅供讀取複本，則請從僅供讀取複本建立快照。	否。RDS for SQL Server 僅供讀取複本上不支援自動備份和手動快照。
能否使用平行複寫？	是。所有支援的 MariaDB 和 MySQL 版本皆允許平行複寫執行緒。	是。重做記錄資料會一律從主要資料庫平行傳輸到其所有僅供讀取副本。	否。PostgreSQL 使用單一程序來處理複寫。	是。重做記錄資料會一律從主要資料庫平行傳輸到其所有僅供讀取副本。

功能或行為	MySQL 和 MariaDB	Oracle	PostgreSQL	SQL Server
您可以將複本保持為已掛載而非唯讀狀態嗎？	否。	是。掛載複本的主要用途是跨區域災難復原。掛載複本不需要 Active Data Guard 授權。如需更多詳細資訊，請參閱 使用適用於 Amazon RDS for Oracle 的僅供讀取複本 。	否。	否。

僅供讀取複本儲存類型

根據預設，僅供讀取複本建立時使用與來源資料庫執行個體相同的儲存類型。但您仍可以依照下表中的選項建立使用與來源資料庫執行個體不同儲存類型的僅供讀取複本。

來源資料庫執行個體儲存體類型	來源資料庫執行個體儲存體配置	僅供讀取複本儲存體類型選項
佈建 IOPS	100 GiB–64 TiB	佈建 IOPS、一般用途、磁性
一般用途	100 GiB–64 TiB	佈建 IOPS、一般用途、磁性
一般用途	<100 GiB	一般用途、磁性
磁帶	100 GiB–6 TiB	佈建 IOPS、一般用途、磁性
磁帶	<100 GiB	一般用途、磁性

Note

當您提升僅供讀取複本的配置儲存空間時，其至少必須提升 10%。如果您嘗試以低於 10% 的值增加該值，便會發生錯誤。

從複本建立複本的限制

Amazon RDS 不支援循環複寫。您不能將資料庫執行個體設定為現有資料庫執行個體的複寫來源。您只能從現有的資料庫執行個體建立新的僅供讀取複本。例如，如果 **MySourceDBInstance** 建立複本到 **ReadReplica1**，您不能將 **ReadReplica1** 設定為向 **MySourceDBInstance** 建立複本。

對於 MRDS for MariaDB 和 RDS for MySQL，以及對於 RDS for PostgreSQL 的某些版本，您可從現有僅供讀取複本建立僅供讀取複本。例如，您可以從現有複本 **ReadReplica1** 建立新的僅供讀取複本 **ReadReplica2**。對於 RDS for Oracle 和 RDS for SQL Server，您無法從現有的僅供讀取複本建立僅供讀取複本。

刪除複本時的考量

如果您不再需要僅供讀取複本，可以使用刪除資料庫執行個體的相同機制明確刪除它們。如果您刪除來源資料庫執行個體而未刪除其僅供讀取複本 AWS 區域，則每個僅供讀取複本都會提升為獨立資料庫執行個體。如需刪除資料庫執行個體的相關資訊，請參閱[刪除資料庫執行個體](#)。如需僅供讀取複本提升的相關資訊，請參閱[提升僅供讀取複本為獨立的資料庫執行個體](#)。

如果您有跨區域僅供讀取複本，請參閱[跨區域複寫考量](#)以瞭解刪除跨區域僅供讀取複本來源資料庫執行個體的相關資料。

建立僅供讀取複本

您可以使用 AWS Management Console、AWS CLI 或 RDS API 從現有資料庫執行個體建立僅供讀取複本。若要建立僅供讀取複本，需指定 `SourceDBInstanceIdentifier`，也就是您想要複製之來源資料庫執行個體的資料庫執行個體識別符。

建立僅供讀取複本時，Amazon RDS 會對來源資料庫執行個體建立資料庫快照，接著開始複寫。資料庫快照作業開始時，來源資料庫執行個體會遇到非常短暫的 I/O 暫停。I/O 懸吊通常持續約一秒鐘。假如來源資料庫執行個體為異地同步備份部署，則可避免 I/O 暫停，因為此時快照是從次要資料庫執行個體建立。

長時間執行的作用中交易會使建立僅供讀取複本的流程變慢。建議您等到長時間執行交易完成後，再建立僅供讀取複本。如果從同一個來源資料庫執行個體平行建立多個僅供讀取複本，Amazon RDS 只會在一開始的建立動作時擷取一個快照。

建立僅供讀取複本時有一些問題需要考慮。首先，務必要將備份保留期設定為 0 以外的值，才能啟用來源資料庫執行個體的自動備份。若僅供讀取複本能讓其他僅供讀取複本做為來源資料庫執行個體，則亦適用此要求。如要啟用 RDS for MySQL 僅供讀取複本的自動備份，首先請先建立該僅供讀取複本，然後修改該複本以啟用自動備份。

Note

在中 AWS 區域，我們強烈建議您在以 Amazon VPC 為基礎的相同虛擬私有雲端 (VPC) 中建立所有僅供讀取複本做為來源資料庫執行個體。如果您在與來源資料庫執行個體不同的 VPC 中建立僅供讀取複本，則複本和 RDS 系統之間的無類別網域間路由 (CIDR) 範圍可能會重疊。CIDR 重疊會使複本不穩定，這會對連線到該複本的應用程式產生負面影響。如果您在建立僅供讀取複本時收到錯誤，請選擇不同的目的地資料庫子網路群組。如需詳細資訊，請參閱 [在 VPC 中使用資料庫執行個體](#)。

沒有直接的方法可以 AWS 帳戶 使用控制台或在另一個僅供讀取複本中建立僅供讀取複本 AWS CLI。

主控台

若要從來源資料庫執行個體建立僅供讀取複本

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇要作為僅供讀取複本來源的資料庫執行個體。
4. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
5. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入僅供讀取複本的名稱。
6. 選擇您的執行個體組態。建議您使用與僅供讀取複本來源資料庫執行個體相同或更大的資料庫執行個體類別和儲存體類型。
7. 對於 AWS 區域，指定僅供讀取複本的目的地區域。
8. 對於儲存體，指定配置的儲存體大小，以及您是否要使用儲存體自動擴展。


如果您的來源資料庫執行個體不在最新的儲存體組態上，則可以使用升級儲存體檔案系統組態選項。您可以啟用此設定，將僅供讀取複本的儲存體檔案系統升級為偏好的組態。如需詳細資訊，請參閱 [the section called “升級儲存體檔案系統”](#)。

9. 對於可用性，選擇是否要在另一個可用區域中建立複本的待命複本，以支援複本的容錯移轉。

Note

建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。

10. 指定其他資料庫執行個體設定。如需每項可用設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。
11. 若要建立加密的僅供讀取複本，請展開其他組態並指定下列設定：
 - a. 選擇 Enable encryption (啟用加密)。
 - b. 對於 AWS KMS key，選擇 KMS 金鑰的 AWS KMS key 識別碼。

 Note

來源資料庫執行個體必須加密。若要進一步了解如何加密來源資料庫執行個體，請參閱 [加密 Amazon RDS 資源](#)。

12. 選擇 Create read replica (建立僅供讀取複本)。

建立讀取複本之後，您可以在 RDS 主控台的 Databases (資料庫) 頁面上看到它。它在 Role (角色) 欄中顯示 Replica (複本)。

AWS CLI

若要從來源資料庫執行個體建立僅供讀取複本，請使用「[建立-DB 執行個體-唯讀複本](#)」AWS CLI 指令。此範例也會設定配置的儲存體大小、啟用儲存體自動擴展，並將檔案系統升級為偏好的組態。

您可以指定其他設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --source-db-instance-identifier mydbinstance \  
  --allocated-storage 100 \  
  --max-allocated-storage 1000 \  
  --upgrade-storage-config
```

在 Windows 中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^
```

```
--source-db-instance-identifier mydbinstance ^  
--allocated-storage 100 ^  
--max-allocated-storage 1000 ^  
--upgrade-storage-config
```

RDS API

若要從來源 MySQL、MariaDB、Oracle、PostgreSQL 或 SQL Server 資料庫執行個體建立僅供讀取複本，請呼叫 Amazon RDS API [CreateDBInstanceReadReplica](#) 操作，其中包含下列必要參數：

- DBInstanceIdentifier
- SourceDBInstanceIdentifier

提升僅供讀取複本為獨立的資料庫執行個體

您可以將僅供讀取複本提升為獨立的資料庫執行個體。如果來源資料庫執行個體擁有多個僅供讀取複本，將其中一個僅供讀取複本提升為資料庫執行個體，並不會影響其他的複本。

升級僅供讀取複本時，RDS 會重新啟動資料庫執行個體，然後再提供資料庫執行個體。該提升程序可耗費數分鐘或更長的時間來完成，視僅供讀取複本大小而定。



提升僅供讀取複本的使用案例

您可能會想要將僅供讀取複本升級到獨立資料庫執行個體，原因如下：

- 實作故障復原 – 您可用僅供讀取複本提升，做為主要資料庫執行個體故障時的資料復原結構描述。這種方法可用於補強同步複寫、自動故障偵測和容錯移轉。

如果您已了解非同步複寫的後果和限制，但仍想使用僅供讀取複本提升做為資料復原，可以這樣做。要進行此操作，請先建立僅供讀取複本，然後監控主要資料庫執行個體的故障。若發生故障，請執行下列程序：

1. 提升僅供讀取複本。
 2. 將資料庫流量引導至提升的資料庫執行個體。
 3. 以提升的資料庫執行個體做為來源，建立替換的僅供讀取複本。
- 升級儲存體組態 - 如果來源資料庫執行個體不在偏好的儲存體組態上，您可以建立執行個體的僅供讀取複本，並升級儲存體檔案系統組態。此選項會將僅供讀取複本的檔案系統遷移至偏好的組態。您接著可以將僅供讀取複本提升為獨立的執行個體。

您可以使用此選項，克服舊版 32 位元檔案系統對儲存體和檔案大小的擴展限制。如需詳細資訊，請參閱 [the section called “升級儲存體檔案系統”](#)。

只有在您的來源資料庫執行個體不在最新的儲存體組態上時，或只有在您要在相同請求內修改資料庫執行個體類別時，才能使用此選項。

- 分片 – 分片採用「不共用」架構，基本上就是將一個大型資料庫分成多個較小的資料庫。拆分資料庫的一種常見方式，就是將未加入同一個查詢的資料表拆分到不同的主機。另一個方法是將資料表複製到多個主機，接著使用雜湊演算法來決定哪個主機將接收特定的更新。您可建立與每個分片 (較小的資料庫) 對應的僅供讀取複本，然後在您決定將其轉換為獨立分片時加以提升。接著您可根據需求調整金鑰空間 (如果要拆分資料列) 或分散資料表到每個分片。
- 執行 DDL 操作 (僅限 MySQL 和 MariaDB) – 像是建立或重建索引等 DDL 操作需要一段時間，且會大幅降低資料庫執行個體的效能。一旦僅供讀取複本與其主要資料庫執行個體同步後，您便能在 MySQL 或 MariaDB 僅供讀取複本上執行這些操作。接著您可提升僅供讀取複本，並指示應用程式使用提升的執行個體。

Note

如果您的僅供讀取複本是 RDS for Oracle 資料庫執行個體，您可以執行轉換，而不是升級。在轉換過程中，來源資料庫執行個體會變成新複本，而複本會成為新的來源資料庫執行個體。如需詳細資訊，請參閱 [執行 Oracle Data Guard 容錯移轉](#)。

提升僅供讀取複本的特性

升級僅供讀取複本之後，它會停止作為僅供讀取複本運作，並成為獨立資料庫執行個體。新的獨立資料庫執行個體具有下列特性：

- 獨立資料庫執行個體會保留提升前僅供讀取複本的選項群組和參數群組。
- 您可以從獨立資料庫執行個體建立僅供讀取複本，並執行 point-in-time 還原作業。
- 您無法將資料庫執行個體用作複寫目標，因為它不再是僅供讀取複本。

升級僅供讀取複本的先決條件

升級僅供讀取複本之前，請執行下列動作：

- 檢閱您的備份策略：
 - 我們建議您啟用備份並至少完成一個備份。備份持續期間為自上次備份後資料庫變更的數量函數。
 - 如果您已啟用僅供讀取複本的備份，請設定自動備份時間，以免日常備份干擾到僅供讀取複本的提升。
 - 請確定您的僅供讀取複本沒有backing-up狀態。僅供讀取複本處於此狀態時，您無法升級僅供讀取複本。
- 停止將任何交易寫入主資料庫執行個體，然後等待 RDS 將所有更新套用至僅供讀取複本。

資料庫更新在主要資料庫執行個體上發生之後，會在僅供讀取複本上發生。複寫延遲可能會有很大差異。使用 [Replica Lag](#) 指標以確定針對僅供讀取複本進行的所有更新時間。

- (僅限 MySQL 和 MariaDB) 若要在升級 MySQL 或 MariaDB 僅供讀取複本之前對其進行變更，請在僅供讀取複本的資料庫 `read_only` 參數群組中將參數設定為。接著您便能執行所有必要的 DDL 操作，像是在僅供讀取複本上建立索引。在僅供讀取複本上擷取快照，不會影響主要資料庫執行個體的效能。

提升僅供讀取複本：基本步驟

以下步驟顯示了提升僅供讀取複本至資料庫執行個體的一般流程：

1. 使用 Amazon RDS 主控台、AWS CLI 命令 [promote-read-replica](#) 或 [PromoteReadReplica](#) Amazon RDS API 作業上的升級選項來提升僅供讀取複本。

Note

提升程序可能需要幾分鐘來完成。升級僅供讀取複本時，RDS 會停止複寫並重新啟動僅供讀取複本。當重新啟動完成，該僅供讀取複本便可用做新的資料庫執行個體。

2. (選用) 將新的資料庫執行個體修改為異地同步備份部署。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#) 及 [設定及管理多可用區部署](#)。

主控台

若要將僅供讀取複本提升為獨立的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。

Databases (資料庫) 窗格隨即出現。每個僅供讀取複本會在 Role (角色) 欄中顯示 Replica (複本)。

3. 選擇您要提升的僅供讀取複本。
4. 針對 Actions (動作)，選擇 Promote (提升)。
5. 在提升僅供讀取複本頁面上，輸入新升級資料庫執行個體的備份保留期和備份時間。
6. 滿意所有設定時，請選擇 Continue (繼續)。
7. 在確認頁面上，選擇 Promote Read Replica (提升僅供讀取複本)。

AWS CLI

若要將僅供讀取複本升級為獨立資料庫執行個體，請使用 AWS CLI [promote-read-replica](#) 指令。

Example

對於LinuxmacOS、或Unix：

```
aws rds promote-read-replica \  
  --db-instance-identifier myreadreplica
```

在 Windows 中：

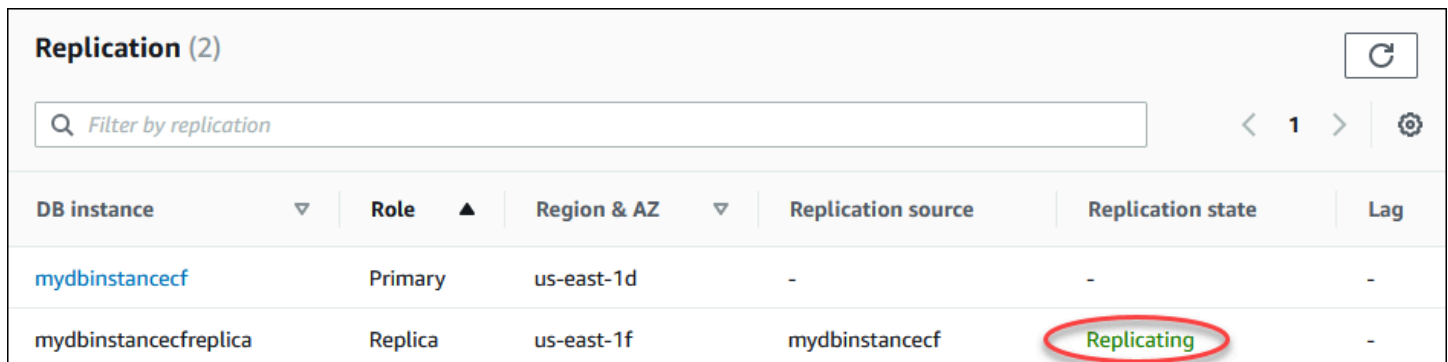
```
aws rds promote-read-replica ^
  --db-instance-identifier myreadreplica
```

RDS API

若要將僅供讀取複本提升為獨立的資料庫執行個體，請使用所需參數 `DBInstanceIdentifier` 呼叫 Amazon RDS API [PromoteReadReplica](#) 操作。

監控僅供讀取複本

您可用多種方式監控僅供讀取複本的狀態。Amazon RDS 主控台在僅供讀取複本詳細內容的 **Connectivity & security (連線和安全)** 的 **Replication (複寫)** 區段中顯示僅供讀取複本的狀態。若要檢視僅供讀取複本的詳細資訊，請在 Amazon RDS 主控台的資料庫執行個體清單中選擇僅供讀取複本的名稱。



DB instance	Role	Region & AZ	Replication source	Replication state	Lag
mydbinstancecf	Primary	us-east-1d	-	-	-
mydbinstancecfreplica	Replica	us-east-1f	mydbinstancecf	Replicating	-

您也可以使用 AWS CLI `describe-db-instances` 命令或 Amazon RDS API `DescribeDBInstances` 操作查看僅供讀取複本的狀態。

僅供讀取複本的狀態可以是下列其中一個：

- **replicating (複寫中)**–僅供讀取複本正在成功複寫中。
- **複寫效能降級 (僅限 SQL Server 和 PostgreSQL)** – 複本正在接收來自主要執行個體的資料，但一或多個資料庫可能無法取得更新。例如，當複本正在設定新建立的資料庫時，就會發生這種情況。在藍/綠部署的藍色環境中進行不支援的 DDL 或大型物件變更時，也可能會發生這種情況。

除非在降級狀態期間發生錯誤，否則狀態不會從 `replication degraded` 轉換為 `error`。

- **error (錯誤)**– 複寫發生錯誤。請檢查 Amazon RDS 主控台內的 **Replication Error (複寫錯誤)** 欄位或事件記錄，以判斷確切的錯誤。如需有關複寫錯誤疑難排解的詳細資訊，請參閱 [對 MySQL 僅供讀取複本問題進行故障診斷](#)。

- terminated (已終止) (僅限 MariaDB、MySQL 或 PostgreSQL) – 已終止複寫。此錯誤是因複寫作業停止連續超過 30 天所導致 (不論手動停止或由於複寫錯誤)。在此情況下，Amazon RDS 會終止主要資料庫執行個體與所有僅供讀取複本之間的複寫。Amazon RDS 會執行此操作以避免來源資料庫執行個體的儲存需求增加以及長期容錯移轉。

複寫中斷可能影響儲存，因為記錄的大小與數量會因為大量錯誤訊息寫入記錄而成長。複寫中斷也會影響故障復原，因為 Amazon RDS 在復原期間需要時間來維護及處理大量記錄。

- terminated (已終止) (僅限 Oracle) – 已終止複寫。如果複寫已停止超過 8 小時，則會發生此情況，因為僅供讀取複本上沒有足夠的儲存體。在此情況下，Amazon RDS 會終止主要資料庫執行個體與受影響的僅供讀取複本之間的複寫。此狀態為終端狀態，且僅供讀取複本必須重新建立。
- stopped (已停止) (僅限 MariaDB 或 MySQL) – 複寫因客戶提出要求而停止。
- replication stop point set (設定了複寫停止點) (僅限 MySQL)–使用 [mysql.rds_start_replication_until](#) 預存程序設定了客戶啟動的停止點，而且複寫正在進行中。
- replication stop point set (達到複寫停止點) (僅限 MySQL)–使用 [mysql.rds_start_replication_until](#) 預存程序設定了客戶啟動的停止點，複寫因達到停止點而停止。

您可以查看複寫資料庫執行個體的位置，並且檢查其複寫狀態。在 RDS 主控台的 Databases (資料庫) 頁面上，它會在 Role (角色) 欄中顯示 Primary (主要)。選擇其資料庫執行個體名稱。在其詳細資訊頁面上，在 Connectivity & security (連線能力和安全性) 索引標籤中，其複寫狀態位於 Replication (複寫) 下面。

監控複寫延遲

您可以 CloudWatch 透過檢視 Amazon RDS ReplicaLag 指標來監控 Amazon 中的複寫延遲。

如為 MariaDB 和 MySQL，ReplicaLag 指標會回報 Seconds_Behind_Master 命令的 SHOW REPLICAS STATUS 欄位值。MySQL 和 MariaDB 複寫延遲的常見原因如下：

- 網路停機。
- 寫入僅供讀取複本上的資料表仍有索引。若未將僅供讀取複本的 read_only 參數設為 0，可能會中斷複寫。
- 使用非交易儲存引擎 (例如 MyISAM)。僅 MySQL 上的 InnoDB 儲存引擎和 MariaDB 上的 XtraDB 儲存引擎支援複寫。

Note

MariaDB 和 MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICATION STATUS`。如果您使用的 MariaDB 是 10.5 之前的版本或使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

當 `ReplicaLag` 指標到達 0，複本即已跟上主要資料庫執行個體。如果 `ReplicaLag` 指標傳回 -1，表示複寫目前為非作用中。`ReplicaLag = -1` 相當於 `Seconds_Behind_Master = NULL`。

對於 Oracle，`ReplicaLag` 指標是 `Apply Lag` 值與目前時間和套用延遲 `DATUM_TIME` 值之差的總和。`DATUM_TIME` 值是僅供讀取複本從其來源資料庫執行個體接收資料的最後時間。如需詳細資訊，請參閱 Oracle 文件中的 [V\\$DATAGUARD_STATS](#)。

對於 SQL Server，`ReplicaLag` 指標是落後的資料庫延遲上限 (以秒為單位)。例如，如果您有兩個分別延遲 5 秒和 10 秒的資料庫，則 `ReplicaLag` 為 10 秒。`ReplicaLag` 指標會傳回下列查詢的值。

```
SELECT MAX(secondary_lag_seconds) max_lag FROM sys.dm_hadr_database_replica_states;
```

如需詳細資訊，請參閱 Microsoft 文件中的 [secondary_lag_seconds](#)。

如果 RDS 無法判斷延遲，例如在複本設定期間或是僅供讀取複本處於 `ReplicaLag` 狀態的時間，-1 會傳回 `error`。

Note

在新資料庫在僅供讀取複本上可供存取之前，將不會列入延遲計算中。

對於 PostgreSQL，`ReplicaLag` 指標會傳回下列查詢的值。

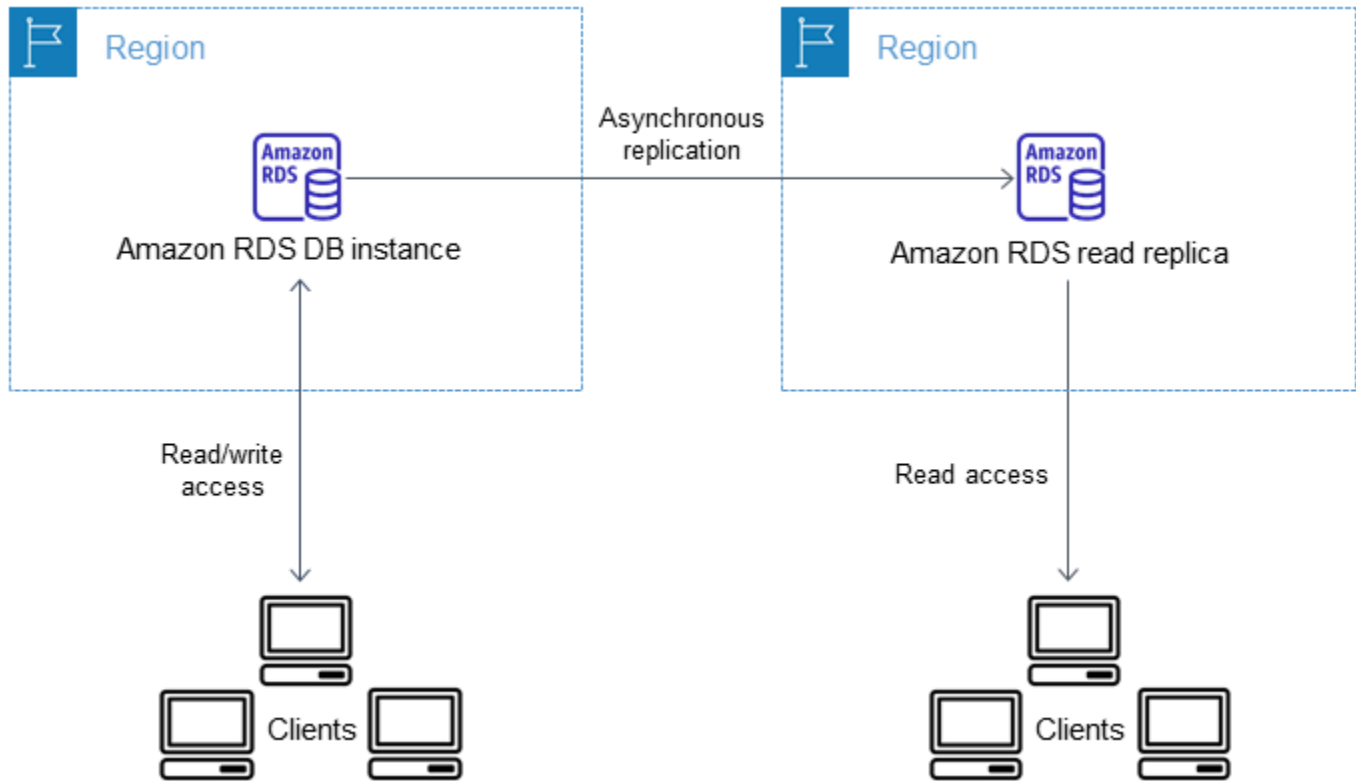
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

PostgreSQL 9.5.2 版和更新版本使用實體複寫槽來管理在來源執行個體上的預先寫入日誌 (WAL) 保留期。對於各個跨區域的僅供讀取複本執行個體來說，Amazon RDS 會建立實體複寫槽，並將其與執行個體建立關聯。兩個 Amazon CloudWatch 指標 `Transaction Logs Disk Usage`，`Oldest Replication Slot Lag` 並顯示在收到的 WAL 資料以及 WAL 資料使用多少儲存空間方面，最滯後的複本有多遠。`Transaction Logs Disk Usage` 值可在跨區域僅供讀取複本嚴重延遲時大幅增加。

如需使用監視資料庫執行個體的詳細資訊 CloudWatch，請參閱[使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。

以不同的方式建立僅供讀取複本 AWS 區域

使用 Amazon RDS，您可以在與來源資料庫執行個體不同 AWS 區域的僅供讀取複本中建立僅供讀取複本。



您可以建立不同的僅供讀取複本 AWS 區域 來執行下列動作：

- 改善災難復原功能。
- 將讀取作業擴展到 AWS 區域 更接近使用者的位置。
- 讓您更輕鬆地從一個資料中心遷移 AWS 區域 到另一個資料中心 AWS 區域。

在與來源執行個體不同 AWS 區域的僅供讀取複本中建立僅供讀取複本類似 AWS 區域。您可以使用 AWS Management Console、執行[create-db-instance-read-replica](#)命令或呼叫 [CreateDBInstanceReadReplica](#) API 作業。

Note

若要在不同 AWS 區域 於來源資料庫執行個體中建立加密的僅供讀取複本，必須加密來源資料庫執行個體。

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需跨區域複寫的版本 and 區域可用性的相關資訊，請參閱[Amazon RDS 中跨區域僅供讀取複本支援的區域和資料庫引擎](#)。

建立跨區域僅供讀取複本

下列程序說明如何從不同 AWS 區域中的來源 MariaDB、Microsoft SQL Server、MySQL、Oracle 或 PostgreSQL 資料庫執行個體建立僅供讀取複本。

主控台

您可以跨 AWS 區域 使用建立僅供讀取複本 AWS Management Console。

使用主控台建立僅 AWS 區域 供讀取複本

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇要作為僅供讀取複本來源的 MariaDB、Microsoft SQL Server、MySQL、Oracle 或 PostgreSQL 資料庫執行個體。
4. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
5. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入僅供讀取複本的名稱。
6. 選擇 Destination Region (目的地區域)。
7. 選擇您想要使用的執行個體規格。建議您使用與僅供讀取複本相同或更大的資料庫執行個體類別和儲存體類型。
8. 若要在另一個中建立加密的僅供讀取複本 AWS 區域：
 - a. 選擇 Enable encryption (啟用加密)。
 - b. 對於 AWS KMS key，選擇目的地中 KMS 金鑰的 AWS KMS key 識別碼 AWS 區域。

Note

若要建立加密的僅供讀取複本，來源資料庫執行個體也必須加密。若要進一步了解如何加密來源資料庫執行個體，請參閱[加密 Amazon RDS 資源](#)。

- 請選擇其他選項，例如 storage autoscaling (儲存體自動擴展)。
- 選擇 Create read replica (建立僅供讀取複本)。

AWS CLI

若要從不同 AWS 區域中的來源 MySQL、Microsoft SQL Server、MariaDB、Oracle 或 PostgreSQL 資料庫執行個體建立僅供讀取複本，您可以使用 [create-db-instance-read-replica](#) 命令。在這種情況下，您可以[create-db-instance-read-replica](#)從想要的 AWS 區域 位置使用僅供讀取複本 (目的地區域)，並為來源資料庫執行個體指定 Amazon 資源名稱 (ARN)。ARN 專門識別在 Amazon Web Services 中建立的資源。

例如，假如您的來源資料庫執行個體位在 US East (N. Virginia) 區域，則 ARN 看起來類似如下。

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

如需 ARN 的詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

若要在不同於來源資料庫執行個體 AWS 區域 中建立僅供讀取複本，您可以從目的地使用 AWS CLI [create-db-instance-read-replica](#)命令 AWS 區域。在另一個 AWS 區域中建立僅供讀取複本時需要下列參數：

- `--region`— 建立僅供讀取複本的目 AWS 區域 的地。
- `--source-db-instance-identifier` – 來源資料庫執行個體的資料庫執行個體識別符。此識別符必須是來源 AWS 區域的 ARN 格式。
- `--db-instance-identifier` – 指定目的地 AWS 區域中僅供讀取複本的識別符。

Example 跨區域僅供讀取複本

下列程式碼會透過 US East (N. Virginia) 區域中的來源資料庫執行個體，在 美國西部 (奧勒岡) 區域中建立僅供讀取複本。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --region us-west-2 \  
  --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

在 Windows 中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^  
  --region us-west-2 ^  
  --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

在另一個 AWS 區域中建立加密的僅供讀取複本時還需要下列參數：

- `--kms-key-id`— 用來加密目的地中僅供讀取複本的 KMS 金鑰 AWS KMS key 識別碼 AWS 區域。

Example 加密的跨區域僅供讀取複本

下列程式碼會透過 US East (N. Virginia) 區域中的來源資料庫執行個體，在美國西部 (奧勒岡) 區域中建立加密的僅供讀取複本。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --region us-west-2 \  
  --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance \  
  --kms-key-id my-us-west-2-key
```

在 Windows 中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^  
  --region us-west-2 ^  
  --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance ^
```

```
--kms-key-id my-us-west-2-key
```

在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 區域之間建立加密的僅供讀取複本時，需要使用此 `--source-region` 選項。若為 `--source-region`，請指定來源資料庫執行個體的 AWS 區域。

若未指定 `--source-region`，請指定 `--pre-signed-url` 值。presigned URL (預先簽章的 URL) 為包含對來源 AWS 區域中呼叫 `create-db-instance-read-replica` 命令之 Signature 第 4 版簽章請求的 URL。如要進一步了解 `pre-signed-url` 選項，請參閱《AWS CLI 命令參考》中的 [create-db-instance-read-replica](#)。

RDS API

若要在不同的來源 MySQL、Microsoft SQL 伺服器、甲骨文或 PostgreSQL 資料庫執行個體建立僅供讀取複本 AWS 區域，您可以呼叫 Amazon RDS API 作業建立資料庫複本。InstanceRead 在這種情況下，您可以從想要的 AWS 區域 位置呼叫 [CreateDB InstanceRead 複本](#) (目的地區域)，並為來源資料庫執行個體指定 Amazon 資源名稱 (ARN)。ARN 專門識別在 Amazon Web Services 中建立的資源。

若要在不同 AWS 區域 於來源資料庫執行個體中建立加密的僅供讀取複本，您可以從目的地使用 Amazon RDS API [CreateDBInstanceReadReplica](#) 操作 AWS 區域。若要在另一個僅供讀取複本中建立加密的僅供讀取複本 AWS 區域，您必須指定的值 PreSignedURL。PreSignedURL 應包含在中建立僅供讀取複本的來源 AWS 區域 中呼叫 [CreateDBInstanceReadReplica](#) 作業的要求。要進一步了解 PreSignedUrl，請參閱 [CreateDBInstanceReadReplica](#)。

例如，假如您的來源資料庫執行個體位在 US East (N. Virginia) 區域，則 ARN 看起來類似如下。

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

如需 ARN 的詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

Example

```
https://us-west-2.rds.amazonaws.com/  
?Action=CreateDBInstanceReadReplica  
&KmsKeyId=my-us-east-1-key  
&PreSignedUrl=https%253A%252F%252F%252Frds.us-west-2.amazonaws.com%252F%253FAction%253DCreateDBInstanceReadReplica%2526DestinationRegion%253Dus-east-1
```



```

%2526KmsKeyId%253Dmy-us-east-1-key
%2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%123456789012%25253Adb%25253Amydbinstance
%2526SignatureMethod%253DHmacSHA256
%2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
%2526Version%253D2014-10-31
%2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
%2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
%2526X-Amz-Date%253D20161117T215409Z
%2526X-Amz-Expires%253D3600
%2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
%2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
&DBInstanceIdentifier=myreadreplica
&SourceDBInstanceIdentifier=&region-arn;rds:us-east-1:123456789012:db:mydbinstance
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<&AWS; Access Key ID>
&Signature=<Signature>

```

Amazon RDS 如何執行跨區域複寫

Amazon RDS 使用下列流程建立跨區域僅供讀取複本。根據 AWS 區域 所涉及的資料和資料庫中的資料量而定，此程序可能需要數小時才能完成。您可在建立跨區域僅供讀取複本時利用此資訊來判斷流程的進度：

1. Amazon RDS 開始將來源資料庫執行個體設定為複寫來源，並將狀態設為 `modifying` (修改中)。
2. Amazon RDS 會開始在目標中設定指定的僅供讀取複本，AWS 區域 並將狀態設定為 `building`。
3. Amazon RDS 對來源 AWS 區域中的來源資料庫執行個體建立自動化資料庫快照。資料庫快照名稱的格式為 `rds:<InstanceID>-<timestamp>`，其中 `<InstanceID>` 為來源執行個體的識別符，`<timestamp>` 為複製開始的日期和時間。例如，`rds:mysourceinstance-2013-11-14-09-24` 是從 `mysourceinstance` 執行個體於 2013-11-14-09-24 所建立。在建立自動化資料庫快照期間，來源資料庫執行個體狀態會保持為 `modifying` (修改中)，僅供讀取複本的狀態保持為 `creating` (建立中)，而資料庫快照的狀態則為 `creating` (建立中)。主控台中資料庫快照頁面的進度欄將回報資料庫快照建立的進度。資料庫快照完成時，資料庫快照和來源資料庫執行個體的状态將同時設為 `available` (可用)。

4. Amazon RDS 開始進行初始資料傳輸的跨區域快照複製。快照複本會列為目的地中的自動快照 AWS 區域，其狀態為建立。其名稱與來源資料庫快照的名稱相同。資料庫快照的進度欄將顯示複製的進度。一旦複製完成，資料庫快照的狀態會變成 available (可用)。
5. 然後 Amazon RDS 會將複製的資料庫快照用於僅供讀取複本上的初始資料載入。在此階段，僅供讀取複本會列在目的地的資料庫執行個體清單中，狀態為 creating (建立中)。一旦載入完成，僅供讀取複本的狀態會變成 available (可用)，並將刪除資料庫快照複本。
6. 僅供讀取複本變成可供使用狀態之後，Amazon RDS 會開始複寫從建立僅供讀取複本操作起始起，對來源執行個體進行的變更。在此階段期間，僅供讀取複本的複寫延遲時間將大於 0。

如需複寫延遲時間的資訊，請參閱 [監控僅供讀取複本](#)。

跨區域複寫考量

在內部執行複寫的所有考量都 AWS 區域 適用於跨區域複寫。在 AWS 區域間複寫時，適用下列的額外考量：

- 來源資料庫執行個體可在多個 AWS 區域有跨區域僅供讀取複本。由於來源 VPC 的存取控制清單 (ACL) 項目數目有限制，RDS 無法保證跨區域僅供讀取複本資料庫執行個體超過五個。
- 您可以在 GovCloud (美國東部) 和 GovCloud (美國西部) 區域之間進行複製，但不能進出 GovCloud (US)。
- 您只能從不是其他 Amazon RDS 資料庫執行個體之僅供讀取複本的來源 Amazon RDS 資料庫執行個體建立跨區域 Amazon RDS 僅供讀取複本。此限制不適用於 MariaDB 和 MySQL 資料庫執行個體。
- 對於與來源執行個體不同 AWS 區域 的僅供讀取複本，您可以預期會看到更高層級的延遲時間。此延遲時間是由於區域資料中心之間有較長的網路通道。
- 如為跨區域僅供讀取複本，任何指定 `--db-subnet-group-name` 參數的建立僅供讀取複本命令皆必須指定同一 VPC 的資料庫子網路群組。
- 在大多數情況下，僅供讀取複本使用指定資料庫引擎的預設資料庫參數群組和資料庫選項群組。

對於 MySQL 和 Oracle DB 引擎，您可以在指 AWS CLI 令 `--db-parameter-group-name` 選項中為僅供讀取複本指定自訂參數群組 [create-db-instance-read-replica](#)。使用 AWS Management Console 時，您無法指定自訂參數群組。

- 僅供讀取複本使用預設安全群組。
- 若為 MariaDB、Microsoft SQL Server、MySQL 及 Oracle DB 資料庫執行個體，刪除跨區域僅供讀取複本的來源資料庫執行個體時，僅供讀取複本會提升。

- 若為 MariaDB、MySQL 和 PostgreSQL 資料庫執行個體，刪除跨區域僅供讀取複本的來源資料庫執行個體時，僅供讀取複本的複寫狀態會設為 `terminated`。不會提升僅供讀取複本。

您必須手動提升僅供讀取複本，或將其刪除。

請求跨區域僅供讀取複本

若要與來源區域通訊，以請求建立跨區域僅供讀取複本，請求者 (IAM 角色或 IAM 使用者) 必須具有來源資料庫執行個體和來源區域的存取權。

請求者的 IAM 政策中的某些條件可能會導致請求失敗。下列範例假設來源資料庫執行個體位於 美國東部 (俄亥俄)，而僅供讀取複本則會在 US East (N. Virginia) 中建立。以下範例顯示請求者 IAM 政策中導致請求失敗的條件：

- 請求者的政策具有 `aws:RequestedRegion` 的條件。

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:RequestedRegion": "us-east-1"
  }
}
```

由於政策不允許存取來源區域，請求失敗。若要成功請求，請同時指定來源區域和目的地區域。

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:RequestedRegion": [
      "us-east-1",
      "us-east-2"
    ]
  }
}
```

- 請求者的政策不允許存取來源資料庫執行個體。

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

若是成功請求，請同時指定來源執行個體和複本。

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
  "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
  "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]
...
```

- 請求者的政策拒絕 `aws:ViaAWSService`。

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
  "Bool": {"aws:ViaAWSService": "false"}
}
```

與來源區域的通訊由 RDS 代表請求者進行。對於成功的請求，請勿拒絕 AWS 服務撥打的電話。

- 請求者的政策具有 `aws:SourceVpc` 或 `aws:SourceVpce` 的條件。

這些請求可能會失敗，因為當 RDS 呼叫遠端區域時，其不是來自指定的 VPC 或 VPC 端點。

如果您需要使用會導致請求失敗的先前條件之一，則可以在政策中包含 `aws:CalledVia` 的第二個陳述式，以使請求成功。例如，您可以如下所示使用 `aws:CalledVia` 搭配 `aws:SourceVpce`：

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
```

```
"Resource": "*",
"Condition": {
  "Condition" : {
    "ForAnyValue:StringEquals" : {
      "aws:SourceVpce": "vpce-1a2b3c4d"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "rds:CreateDBInstanceReadReplica"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
        "rds.amazonaws.com"
      ]
    }
  }
}
}
```

如需詳細資訊，請參閱 IAM 使用者指南 中的 [IAM 中的政策和許可](#)。

授權僅供讀取複本

跨區域資料庫僅供讀取複本建立請求之後傳回 success，RDS 會在後台啟動複本建立。系統會建立 RDS 存取來源資料庫執行個體的授權。此授權會將來源資料庫執行個體連結至僅供讀取複本，並允許 RDS 僅複製到指定的僅供讀取複本。

授權由 RDS 使用服務連結 IAM 角色中的 `rds:CrossRegionCommunication` 許可來驗證。如果授權複本，RDS 會與來源區域通訊並完成複本建立。

對於之前未透過 `CreateDBInstanceReadReplica` 請求授權的資料庫執行個體，RDS 將無法存取。僅供讀取複本建立完成時，會撤銷授權。

RDS 使用服務連結的角色來驗證來源區域中的授權。如果您在複製建立程序期間刪除服務連結的角色，則複本則會失敗。

如需詳細資訊，請參閱 IAM 使用者指南中的 [使用服務連結角色](#)。

使用 AWS Security Token Service 認證

來自 global AWS Security Token Service (AWS STS) 端點的會話令牌僅在 AWS 區域 默認情況下啟用 (商業區域) 的會話令牌有效。如果您使用中 `assumeRole` API 作業的認證 AWS STS，如果來源區域是選擇加入的區域，請使用地區端點。否則，請求將失敗。發生這種情況是因為您的憑據在兩個區域都必須有效，這對於選擇加入區域僅在使用區域 AWS STS 端點時才有效。

若要使用全域端點，請確保操作時在這兩個區域均已將其啟用。在帳號設定 `Valid in all AWS #` 中將全域端點 AWS STS 設定為。

相同的規則適用於預先簽章 URL 參數中的登入資料。

如需詳細資訊，請參閱 [AWS STS](#) 《IAM 使用者指南》AWS 區域中的「管理」。

跨區域複寫成本

跨區域複寫傳輸的資料會衍生 Amazon RDS 數據傳輸費。對於傳出來源 AWS 區域的資料，這些跨區域複寫動作會產生費用：

- 當您建立僅供讀取複本時，Amazon RDS 會取得來源執行個體的快照，並將快照傳輸至僅供讀取複本 AWS 區域。
- 對於在來源資料庫中進行的每項資料修改，Amazon RDS 都會將資料從來源傳輸 AWS 區域 到僅供讀取複本 AWS 區域。

如需資料傳輸定價的詳細資訊，請參閱 [Amazon RDS 定價](#)。

如為 MySQL 和 MariaDB 執行個體，可透過減少建立的跨區域僅供讀取複本數量來降低資料傳輸成本。例如，假設您在一個來源資料庫執行個體中有一個，AWS 區域 而且想要在另 AWS 區域一個執行個體中有三個僅供讀取複本。在此情況下，您只從來源資料庫執行個體建立一個僅供讀取複本。另兩個僅供讀取複本則從第一個僅供讀取複本建立，而非來源資料庫執行個體。

例如，如果您有一 `source-instance-1` 個 AWS 區域，則可以執行以下操作：

- `read-replica-1` 在新的中創建 AWS 區域，指定 `source-instance-1` 為源。
- 從 `read-replica-2` 建立 `read-replica-1`。
- 從 `read-replica-3` 建立 `read-replica-1`。

在此範例中，您只需要為從 `source-instance-1` 傳輸到 `read-replica-1` 的資料付費。從 `read-replica-1` 傳輸至其他另兩個僅供讀取複本的資料，因為位於相同 AWS 區域中，因無需付費。如果您直接從 `source-instance-1` 建立所有三個複本，那麼您將為所有三個複本的資料傳輸付費。

標記 Amazon RDS 資源

Amazon RDS 標籤是您定義並與 Amazon RDS 資源 (例如資料庫執行個體或資料庫快照) 建立關聯的名稱-值組。此名稱叫做金鑰。或者，您可以為鍵提供值。

您可以使用 AWS Management Console、AWS CLI、或 Amazon RDS API 在 Amazon RDS 資源上新增、列出和刪除標籤。使用 CLI 或 API 時，請務必提供 RDS 資源的 Amazon Resource Name (ARN) 來使用。如需建構 ARN 的詳細資訊，請參閱 [建構 Amazon RDS 的 ARN](#)。

主題

- [為什麼要使用 Amazon RDS 資源標籤？](#)
- [Amazon RDS 資源標籤的工作原理](#)
- [標記 Amazon RDS 資源的最佳實務](#)
- [在 Amazon RDS 中管理標籤](#)
- [將標籤複製到資料庫快照](#)
- [教學課程：使用標籤指定要停止的資料庫執行個體](#)

為什麼要使用 Amazon RDS 資源標籤？

您可以使用標籤來執行下列作業：

- 依應用程式、專案、部門、環境等對 RDS 資源進行分類。例如，您可以使用標籤鍵來定義品類，其中標籤值是此品類中的項目。您可以建立標籤 `environment=prod`。或者，您可以定義的標籤金鑰 `project` 和標籤值 `Salix`，這表示已將 Amazon RDS 資源指派給 `Salix` 專案。
- 自動化資源管理工作。例如，您可以為加上標籤的例證建立維護時段，`environment=prod` 這些例證與加上標籤的例證視窗不同 `environment=test`。您也可以為標記的執行個體設定自動資料庫快照 `environment=prod`。
- 控制 IAM 政策中對 RDS 資源的存取。您可以使用全域 `aws:ResourceTag/tag-key` 條件索引鍵來執行此操作。例如，政策可能只允許 `DBAdmin` 群組中的使用者修改標記為的資料庫執行個體 `environment=prod`。如需使用 IAM 政策管理已標記資源存取權的相關資訊，請參閱 AWS Identity [Amazon RDS 的 Identity and access management](#) and [Access Management 使用者指南中的](#) [和控制 AWS 資源存取](#)。
- 根據標籤監視資源。例如，您可以為標記為的資料庫執行個體建立 Amazon CloudWatch 儀表板 `environment=prod`。

- 透過將類似標記資源的費用分組來追蹤成本。例如，如果您使用標記與 Salix 專案相關聯的 RDS 資源 `project=Salix`，您可以產生成本報告，並將費用分配給此專案。如需詳細資訊，請參閱 [在 Amazon RDS 中使用標籤的 AWS 計費方式](#)。

Amazon RDS 資源標籤的工作原理

AWS 不對您的標籤應用任何語義含義。標籤會嚴格解譯為字元字串。

主題

- [Amazon RDS 中的標籤組](#)
- [Amazon RDS 中的標籤結構](#)
- [符合標記資格的 Amazon RDS 資源](#)
- [在 Amazon RDS 中使用標籤的 AWS 計費方式](#)

Amazon RDS 中的標籤組

每個 Amazon RDS 資源都有一個稱為標籤集的容器。容器包含指派給資源的所有標籤。資源只有一個標籤組。

標籤組包含 0—50 個標籤。如果您將標籤新增至 RDS 資源，而其擁有與現有標籤相同的金鑰，則新值會覆寫舊值。

Amazon RDS 中的標籤結構

RDS 標籤的結構如下所示：

標籤鍵

索引鍵是標籤的必要名稱。字串值的長度必須為 1—128 個 Unicode 字元，且不能加上或前置字元。aws:rds:字串只能包含 Unicode 字母、數字、空白字元、`_`、`.`、`:`、`/`、`=`、`+`、`-`、`和@`。Java 正則表達式是 `"^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$"`。標籤鍵區分大小寫。因此，鍵 `project` 和 `Project` 是不同的。

索引鍵對於標籤組而言是唯一的。例如，您不能在標籤組中擁有金鑰組中的金鑰配對，其金鑰相同但具有不同的值，例如 `project=Trinity` 和 `project=Xanadu`。

標籤值

該值是標籤的可選字符串值。字符串的長度必須為 1—256 個萬國碼字元。字符串只能包含 Unicode 字母、數字、空白字元、`_`、`.`、`:`、`/`、`=`、`+`、`-`、和`@`。Java 正則表達式是"`^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`". 標籤值區分大小寫。因此，值`prod`和`Prod`是不同的。

值不需要在標籤集中是唯一的，並且可以為 `null`。例如，您可以在 `project=Trinity` 和 `cost-center=Trinity` 標籤集中有一個鍵值對。

符合標記資格的 Amazon RDS 資源

您可以標記以下 Amazon RDS 資源：

- 資料庫執行個體
- 資料庫叢集
- 資料庫叢集端點
- 僅供讀取複本
- 資料庫快照
- 資料庫叢集快照
- 預留資料庫執行個體
- 事件訂閱
- 資料庫選項群組
- 資料庫參數群組
- 資料庫叢集參數群組
- 資料庫子網路群組
- RDS Proxy
- RDS Proxy 端點

Note

目前無法使用 AWS Management Console 標記 RDS Proxy 和 RDS Proxy 端點。

- 藍/綠部署
- 零 ETL 整合 (預覽)

在 Amazon RDS 中使用標籤的 AWS 計費方式

使用標籤來組織帳 AWS 單，以反映您自己的成本結構。要做到這一點，註冊以獲取包含標籤鍵值的 AWS 帳戶 帳單。接著，若要查看合併資源的成本，請根據具有相同標籤鍵值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。如需詳細資訊，請參閱《AWS Billing》使用者指南中的[使用成本分配標籤](#)。

成本分配標籤如何搭配資料庫快照搭配

您可以將標籤新增至資料庫快照。不過，您的帳單不會反映這個分組。若要套用至資料庫快照的成本配置標記，必須符合下列條件：

- 標籤必須附加至父資料庫執行個體。
- 父資料庫執行個體必須與資料庫快照相同 AWS 帳戶。
- 父資料庫執行個體必須與資料庫快照相同 AWS 區域。

如果資料庫快照不在與父資料庫執行個體相同的區域中，則資料庫叢集快照會被視為孤立。孤立快照的成本會彙總到單一未標記的行項目中。滿足下列條件時，跨帳戶資料庫快照不會被視為孤立：

- 它們與父資料庫執行個體位於相同的區域中。
- 父資料庫執行個體由來源帳戶擁有。

Note

如果父資料庫執行個體由不同帳戶擁有，則成本分配標籤不適用於目標帳戶中的跨帳戶快照。

標記 Amazon RDS 資源的最佳實務

當您使用標籤時，我們建議您遵守下列最佳做法：

- 標籤使用的文件慣例會遵循組織中的所有專案團隊。特別是，請確保名稱具有描述性且一致。例如，在格式上標準化，`environment:prod`而不是使`env:production`用標記某些資源。

Important

請勿將個人識別資訊 (PII) 或其他機密或敏感資訊儲存在標籤中。

- 自動標記以確保一致性。例如，您可以使用下列技巧：
 - 在 AWS CloudFormation 範本中包含標籤。當您使用範本建立資源時，會自動標記資源。
 - 使用 AWS Lambda 函數定義和應用標籤。
 - 建立 SSM 文件，其中包含將標籤新增至 RDS 資源的步驟。
- 僅在必要時使用標籤。您最多可以為單一 RDS 資源新增 50 個標籤，但最佳做法是避免不必要的標籤擴散和複雜性。
- 定期檢閱標籤以確保相關性和準確性。視需要移除或修改過時的標籤。
- 請考慮使用中的標 AWS 籤編輯器建立標籤 AWS Management Console。您可以使用標籤編輯器同時將標籤新增至多個支援的資 AWS 源，包括 RDS 資源。如需詳細資訊，請參閱 AWS Resource Groups 使用者指南中的 [標籤編輯器](#)。

在 Amazon RDS 中管理標籤

您可以執行下列作業：

- 在建立資源時建立標籤，例如執行指 AWS CLI 令時 `create-db-instance`。
- 使用指令將標籤新增至現有資源 `add-tags-to-resource`。
- 使用指令列出與特定資源相關聯的標籤 `list-tags-for-resource`。
- 使用指令更新標籤 `add-tags-to-resource`。
- 使用指令從資源中移除標籤 `remove-tags-from-resource`。

下列程序顯示如何在與資料庫執行個體相關的資源上執行一般標記作業。請注意，標籤會進行快取，以供授權使用。因此，當您在 Amazon RDS 資源上新增或更新標籤時，可能需要經過幾分鐘的時間才能進行修改。

主控台

設定 Amazon RDS 資源標籤的程序類似為所有資源設定標籤的程序。以下程序會說明如何為 Amazon RDS 資料庫執行個體新增標籤。

新增標籤至資料庫資訊個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

Note

若要篩選 Databases (資料庫) 窗格中的資料庫執行個體清單，請在 Filter databases (篩選資料庫) 中輸入文字字串。只有包含此字串的資料庫執行個體會顯示出來。

3. 選擇您要標記的資料庫執行個體名稱，以顯示其詳細資訊。
4. 在詳細內容區段，向下捲動至 Tags (標籤) 區段。
5. 選擇 Add (新增)。此時將會顯示 Add tags (新增標籤) 視窗。

Tag key	Value
<input type="text"/>	<input type="text"/>

6. 輸入 Tag key (標籤索引鍵) 和 Value (值) 的值。
7. 若要新增另一個標籤，您可以選擇 Add another Tag (新增另一個標籤)，然後輸入其 Tag key (標籤索引鍵) 和 Value (值) 的值。

視需要重複此步驟。

8. 選擇 Add (新增)。

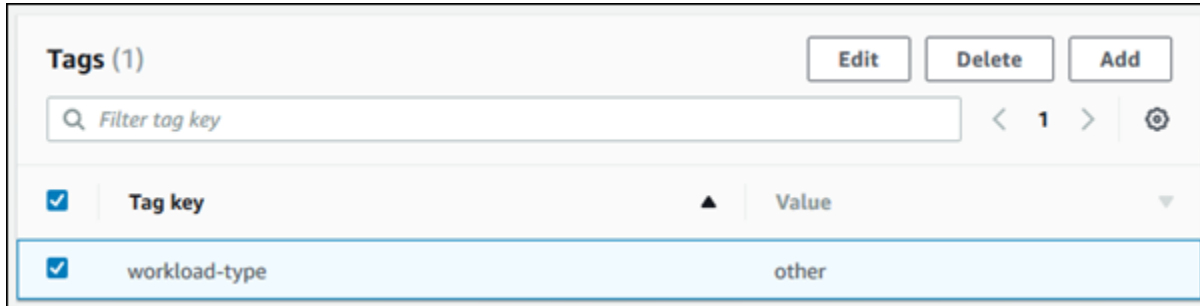
從資料庫資訊個體刪除標籤

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

Note

若要篩選 Databases (資料庫) 窗格中的資料庫執行個體清單，請在 Filter databases (篩選資料庫) 方塊中輸入文字字串。只有包含此字串的資料庫執行個體會顯示出來。

3. 選擇資料庫執行個體的名稱，以顯示其詳細資訊。
4. 在詳細內容區段，向下捲動至 Tags (標籤) 區段。
5. 選擇您要刪除的標籤。



6. 選擇 Delete (刪除)，接著在 Delete tags (刪除標籤) 視窗中選擇 Delete (刪除)。

AWS CLI

您可以使用 AWS CLI 新增、列出或移除資料庫執行個體的標籤。

- 若要將一個或多個標籤新增至 Amazon RDS 資源，請使用 AWS CLI 指令 [add-tags-to-resource](#)。
- 若要列出 Amazon RDS 資源上的標籤，請使用 AWS CLI 指令 [list-tags-for-resource](#)。
- 若要從 Amazon RDS 資源中移除一個或多個標籤，請使用 AWS CLI 指令 [remove-tags-from-resource](#)。

若要進一步了解如何建構必要的 ARN，請參閱 [建構 Amazon RDS 的 ARN](#)。

RDS API

您可以使用 Amazon RDS API 新增、列出或移除資料庫執行個體的標籤。

- 若要為 Amazon RDS 資源新增標籤，請使用 [AddTagsToResource](#) 操作。
- 若要列出指派給 Amazon RDS 資源的標籤，請使用 [ListTagsForResource](#)。
- 若要移除 Amazon RDS 資源的標籤，請使用 [RemoveTagsFromResource](#) 操作。

若要進一步了解如何建構必要的 ARN，請參閱[建構 Amazon RDS 的 ARN](#)。

搭配 XML 使用 Amazon RDS API 時，標籤使用以下結構描述：

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Trinity</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>Jones</Value>
    </Tag>
  </TagSet>
</Tagging>
```

下表列出允許的 XML 標籤及其特性。Key 和的值區分 Value 大小寫。例如，project=Trinity 和 PROJECT=Trinity 是不同的標籤。

標記元素	描述
TagSet	標籤集是一個容器，其中存放指派給 Amazon RDS 資源的所有標籤。每個資源只能有一個標籤集。您 TagSet 只能透過 Amazon RDS API 使用。
標籤	標籤是使用者定義的鍵值組。標籤集內可以有 1 到 50 個標籤。
Key	<p>鍵是標籤的必要名稱。如需限制，請參閱Amazon RDS 中的標籤結構。</p> <p>字串值長度可以是 1 到 128 個 Unicode 字元，不可在前面加上 aws: 或 rds:。該字串僅能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「-」（Java regex：<code>^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*\$</code>）。</p> <p>鍵在標籤集內必須是唯一的。舉例來說，標籤集不能擁有索引鍵相同，但值不同的索引鍵組，例如：project/Trinity 和 project/Xanadu。</p>
Value	<p>值是標籤的選用值。如需限制，請參閱Amazon RDS 中的標籤結構。</p> <p>字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上 aws: 或 rds:。該字串僅能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「-」（Java regex：<code>^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*\$</code>）。</p>

標記元素	描述
	標籤組中的值不必是唯一的，並且可以是 null。例如，在 project/Trinity 及 cost-center/Trinity 標籤集中，均能擁有一個索引鍵/值組。

將標籤複製到資料庫快照

當您建立或還原資料庫執行個體時，您可以指定標籤從資料庫執行個體複製到資料庫執行個體的快照。複製標籤可確保資料庫快照的中繼資料符合來源資料庫執行個體的中繼資料。它也確保資料庫快照的任何存取政策也符合來源資料庫執行個體的存取政策。

您可以指定標籤複製到資料庫快照以用於下列動作：

- 建立資料庫執行個體。
- 還原資料庫執行個體。
- 建立僅供讀取複本。
- 複製資料庫快照。

大多數情況下，預設為不複製標籤。但是從資料庫快照還原資料庫執行個體時，RDS Proxy 會檢查您是否指定了新標籤。如果有指定，便會將新標籤新增到還原的資料庫執行個體中。如果沒有新標籤，RDS 會在建立快照時將來源資料庫執行個體中的標籤新增到還原的資料庫執行個體。

為了防止將來源資料庫執行個體中的標籤新增到還原的資料庫執行個體，建議您在還原資料庫執行個體時指定新標籤。

Note

在某些情況下，您可能會包含建立 [AWS CLI db-snapshot](#) 指令的 `--tags` 參數值。或者，您可以至少提供一個標籤給 [CreateDBSnapshot](#) API 作業。在這些情況下，RDS 不會將標籤從來源資料庫執行個體複製到新的資料庫快照。即使來源資料庫執行個體開啟了 `--copy-tags-to-snapshot` (CopyTagsToSnapshot) 選項，此功能仍適用。

如果採取此方法，您可以從資料庫快照建立資料庫執行個體的複本。這種方法可避免新增不適用於新資料庫執行個體的標籤。您可以使用 AWS CLI `create-db-snapshot` 命令 (或 `CreateDBSnapshot` RDS API 作業) 建立資料庫快照集。在建立資料庫快照之後，您可以依照本主題稍後所述來新增標籤。

教學課程：使用標籤指定要停止的資料庫執行個體

本教學課程假設您在開發或測試環境中有多個資料庫執行個體。您需要將這些資料庫執行個體保留數天。有些資料庫執行個體會在一夜之間執行測試，而其他資料庫執行個體則可以在一夜之間停止，

下列教學課程說明如何將標籤指派給適合在夜間停止的資料庫執行個體。本教學課程說明指令碼如何偵測哪些資料庫執行個體具有標籤，然後停止已標記的資料庫執行個體。在這個範例中，鍵值對的值部分並不重要。stoppable 標籤的存在表示資料庫執行個體具有此使用者定義的屬性。

在下列教學課程中，用於標記的命令和 API 可與 ARN 搭配使用，這可讓 RDS 在 AWS 區域、AWS 帳戶和可能具有相同短名稱的不同類型資源之間順暢運作。您可以在於資料庫執行個體上操作的 CLI 命令中指定 ARN 而不是資料庫執行個體 ID。

若要指定要停止哪些資料庫執行個體

1. 確定您要指定為可停止的資料庫執行個體 ARN。

在下列範例中，請將您自己的資料庫執行個體名稱取代####-DB 執行個體。在使用 ARN 參數的後續命令中，替換您自己的資料庫執行個體 ARN。ARN 包含您自己的 AWS 帳戶 ID 和資料庫執行個體所在 AWS 地區的名稱。

```
$ aws rds describe-db-instances --db-instance-identifier dev-test-db-instance \  
  --query "[*].{DBInstance:DBInstanceArn}" --output text  
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
```

2. 將標籤 `stoppable` 新增至此資料庫執行個體。

您會選擇此標籤的名稱。由於此範例會將標籤視為存在或不存在的屬性，因此會省略 `Value=` 參數的 `--tags` 部分。這種方法表示您可以避免設計命名慣例，對名稱中的所有相關資訊進行編碼。在這類慣例中，您可能會對資料庫執行個體名稱或其他資源名稱中的資訊進行編碼。

```
$ aws rds add-tags-to-resource \  
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \  
  --tags Key=stoppable
```

3. 確認標籤存在於資料庫執行個體中。

下列命令會以 JSON 格式和純定位鍵分隔文字擷取資料庫執行個體的標籤資訊。

```
$ aws rds list-tags-for-resource \  
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
```



```

{
  "TagList": [
    {
      "Key": "stoppable",
      "Value": ""
    }
  ]
}
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --
output text
TAGLIST stoppable

```

4. 停止指定為的所有資料庫執行個體stoppable。

下列範例會建立列出所有資料庫執行個體的文字檔。shell 命令會循環瀏覽清單，並檢查每個資料庫執行個體是否使用相關屬性標記，並aws rds stop-db-instance針對每個資料庫執行個體執行命令執行。

```

$ aws rds describe-db-instances --query "*[].[DBInstanceArn]" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
  match="$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
stoppable)"
  if [[ ! -z "$match" ]]
  then
    echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
    dbid=$(echo $arn | sed -e 's/.*://')
    aws rds stop-db-instance --db-instance-identifier $dbid
  fi
done

DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is
tagged as stoppable. Stopping it now.
{
  "DBInstance": {
    "DBInstanceIdentifier": "dev-test-db-instance",
    "DBInstanceClass": "db.t3.medium",
    ...

```

您可以在每天結束時執行類似前面的指令碼，以確保停止非必要的資料庫執行個體。您也可以使用 `cron` 之類的公用程式來排程要執行的任務，例如每天晚上檢查。例如，如果某些資料庫執行個體錯誤地保持執行中狀態，您可以執行此操作。在這裡，您可以微調準備要檢查的資料庫執行個體清單的命令。

下列命令會產生資料庫執行個體的清單，但只會產生 `available` 狀態的執行個體清單。指令碼可以忽略已停止的資料庫執行個體，因為它們會有不同的狀態值，例如 `stopped` 或 `stopping`。

```
$ aws rds describe-db-instances \
  --query '*[].[DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus]|[?
DBInstanceStatus == `available`]|[].[DBInstanceArn:DBInstanceArn]' \
  --output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

Tip

您可以使用指派標籤，並尋找具有這些標籤的資料庫執行個體，以其他方式降低成本。例如，採取此案例搭配用於開發和測試的資料庫執行個體。在此情況下，您可以指定一些要在每天結束時刪除的資料庫執行個體。或者，您可以指定它們，在預期的低使用量期間將其資料庫執行個體變更為小型資料庫執行個體類別。

在 Amazon RDS 中使用 Amazon Resource Name (ARN)

在 Amazon Web Services 中建立的資源，都是用 Amazon 資源名稱 (ARN) 做為唯一識別符。當您在執行特定的 Amazon RDS 操作時，必須要指定可識別 Amazon RDS 資源的唯一 ARN。舉例來說，建立 RDS 資料庫執行個體僅供讀取複本時，請務必提供來源資料庫執行個體的 ARN。

建構 Amazon RDS 的 ARN

在 Amazon Web Services 中建立的資源，都是用 Amazon 資源名稱 (ARN) 做為唯一識別符。若要建構 Amazon RDS 資源的 ARN，則可使用下列語法。

```
arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
		rds-fips.us-east-2.api.aws	HTTPS
		rds.us-east-2.api.aws	HTTPS
		rds-fips.us-east-2.amazonaws.com	HTTPS
美國東部 (維吉尼亞 北部)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
美國西部 (加利佛尼 亞北部)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
美國西部 (奧勒岡)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
非洲 (開普敦)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
亞太區域 (香港)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS
亞太區域 (海德拉巴)	ap-south-2	rds.ap-south-2.amazonaws.com	HTTPS
		rds.ap-south-2.api.aws	HTTPS
亞太區域 (雅加達)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com	HTTPS
		rds.ap-southeast-3.api.aws	HTTPS
亞太區域 (墨爾本)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com	HTTPS
		rds.ap-southeast-4.api.aws	HTTPS
亞太區域 (孟買)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS
		rds.ap-south-1.api.aws	HTTPS
亞太區域 (大阪)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS
		rds.ap-northeast-3.api.aws	HTTPS
亞太區域 (首爾)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS
		rds.ap-northeast-2.api.aws	HTTPS
亞太區域 (新加坡)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
		rds.ap-southeast-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
亞太區域 (雪梨)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
		rds.ap-southeast-2.api.aws	HTTPS
亞太區域 (東京)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
加拿大 (中部)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
加拿大西部 (卡加利)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
歐洲 (法蘭克福)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
歐洲 (愛爾蘭)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
歐洲 (倫敦)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
歐洲 (米蘭)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS
歐洲 (巴黎)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
歐洲 (西班牙)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
歐洲 (斯德哥爾摩)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
歐洲 (蘇黎世)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
以色列 (特拉維夫)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
中東 (巴林)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS
中東 (阿拉伯聯合大公國)	me-central-1	rds.me-central-1.amazonaws.com	HTTPS
		rds.me-central-1.api.aws	HTTPS
南美洲 (聖保羅)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
		rds.sa-east-1.api.aws	HTTPS
AWS GovCloud (美國東部)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS
		rds.us-gov-east-1.api.aws	HTTPS
AWS GovCloud (美國西部)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS
		rds.us-gov-west-1.api.aws	HTTPS

下表顯示您在為特定 Amazon RDS 資源類型建構 ARN 時應使用的格式。

資源類型	ARN 格式
資料庫執行個體	<p>arn:aws:rds:<region>:<account> :db:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :db:my-mysql-instance-1</pre>
資料庫叢集	<p>arn:aws:rds:<region>:<account> :cluster:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :cluster: my-aurora-cluster-1</pre>
事件訂閱	<p>arn:aws:rds:<region>:<account> :es:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :es:my-subscription</pre>
資料庫選項群組	<p>arn:aws:rds:<region>:<account> :og:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :og:my-og</pre>
DB parameter group (資料庫參數群組)	<p>arn:aws:rds:<region>:<account> :pg:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :pg:my-param-enable-logs</pre>

資源類型	ARN 格式
DB cluster parameter group (資料庫叢集參數群組)	<p>arn:aws:rds:<region>:<account> :cluster-pg:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :cluster-pg: my-cluster-param-timezone</pre>
預留資料庫執行個體	<p>arn:aws:rds:<region>:<account> :ri:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :ri:my-reserved-postgresql</pre>
資料庫安全群組	<p>arn:aws:rds:<region>:<account> :secgrp:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :secgrp:my-public</pre>
自動化的資料庫快照	<p>arn:aws:rds:<##> : <##> : snapshot:rds : <##></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :snapshot:rds: my-mysql-db-2019-07-22-07-23</pre>
自動化的資料庫叢集快照	<p>arn:aws:rds:<##> : <##> : cluster-snapshot:rds : <##></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot:rds: my-aurora-cluster-2019-07-22-16-16</pre>

資源類型	ARN 格式
手動資料庫快照	<p>arn:aws:rds:<region>:<account> :snapshot:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :snapshot: my-mysql-db-snap</pre>
手動資料庫叢集快照	<p>arn:aws:rds:<region>:<account> :cluster-snapshot:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot: my-aurora-cluster-snap</pre>
資料庫子網路群組	<p>arn:aws:rds:<region>:<account> :subgrp:<name></p> <p>例如：</p> <pre>arn:aws:rds: us-east-2 :123456789012 :subgrp:my-subnet-10</pre>

取得現有的 ARN

您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 RDS API 取得 RDS 資源的 ARN。

主控台

若要從中取得 ARN AWS Management Console，請瀏覽至您想要 ARN 的資源，然後檢視該資源的詳細資訊。

例如，您可以從資料庫執行個體詳細資訊的組態標籤取得資料庫執行個體的 ARN。

AWS CLI

若要從特定 RDS 資源取得 ARN，您可以使用該資源的 describe 命令。AWS CLI 下表顯示了每個 AWS CLI 命令，以及用於獲取 ARN 命令的 ARN 屬性。

AWS CLI 命令	ARN 屬性
describe-event-subscriptions	EventSubscription 阿恩
describe-certificates	CertificateArn
describe-db-parameter-groups	DB ParameterGroup 阿恩
describe-db-cluster-parameter-groups	資料庫 ClusterParameter GroupArn
describe-db-instances	資料庫 InstanceArn
describe-db-security-groups	DB SecurityGroup 阿恩
describe-db-snapshots	資料庫 SnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	預備分享 InstanceArn
describe-db-subnet-groups	DB SubnetGroup 阿恩
describe-option-groups	OptionGroup 阿恩
describe-db-clusters	資料庫 ClusterArn
describe-db-cluster-snapshots	DB ClusterSnapshot 阿恩

例如，下列 AWS CLI 命令會取得資料庫執行個體的 ARN。

Example

對於 Linux/macOS、或 Unix：

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "*[].[DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn]"
```

在 Windows 中：

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

該命令的輸出與以下內容相似：

```
[
  {
    "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",
    "DBInstanceIdentifier": "instance_id"
  }
]
```

RDS API

若要取得特定 RDS 資源的 ARN，則可呼叫下列 RDS API 操作，並搭配使用下方所示的 ARN 屬性。

RDS API 操作	ARN 屬性
DescribeEvent 訂閱	EventSubscription 阿恩
DescribeCertificates	CertificateArn
描述 B ParameterGroups	DB ParameterGroup 阿恩
描述 B 群組 ClusterParameter	資料庫 ClusterParameter GroupArn
DescribeDBInstances	資料庫 InstanceArn
描述 B SecurityGroups	DB SecurityGroup 阿恩
DescribeDBSnapshots	資料庫 SnapshotArn
DescribeEvents	SourceArn
DescribeReserved 數據庫實例	預備分享 InstanceArn
描述 B SubnetGroups	DB SubnetGroup 阿恩

RDS API 操作	ARN 屬性
DescribeOption群組	OptionGroup阿恩
DescribeDBClusters	資料庫 ClusterArn
描述 B ClusterSnapshots	DB ClusterSnapshot 阿恩

使用適合 Amazon RDS 資料庫執行個體的儲存體

若要指定資料如何儲存於 Amazon RDS 中，您需要在建立或修改資料庫執行個體時，選擇儲存類型並提供儲存體大小。之後，您可以藉由修改資料庫執行個體，來增加儲存量或改變儲存類型。如需工作負載適合之儲存類型的詳細資訊，請參閱 [Amazon RDS 儲存類型](#)。

主題

- [增加資料庫執行個體的儲存容量](#)
- [使用 Amazon RDS 儲存體自動擴展自動管理容量](#)
- [升級資料庫執行個體的儲存體檔案系統](#)
- [修改佈建 IOPS SSD 儲存體的設定](#)
- [I/O 密集型儲存體修改](#)
- [修改一般用途 SSD \(gp3\) 儲存體的設定](#)
- [使用專用日誌磁碟區 \(DLV\)](#)

增加資料庫執行個體的儲存容量

如果您需要更多空間來儲存資料，可以擴展現有資料庫執行個體的儲存容量。若要如此做，您可以使用 Amazon RDS 管理主控台、Amazon RDS API 或 AWS Command Line Interface (AWS CLI)。如需儲存體限制的相關資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

Note

擴展 Amazon RDS for Microsoft SQL Server 資料庫執行個體的儲存體，僅支援一般用途 SSD 或佈建 IOPS SSD 儲存類型。

若要監控資料庫執行個體的可用儲存容量，以便在必要時做出回應，建議您建立 Amazon CloudWatch 警示。如需設定 CloudWatch 鬧鐘的詳細資訊，請參閱 [使用 CloudWatch 鬧鐘](#)。

擴展儲存體通常不會導致資料庫執行個體的任何中斷或效能下降。修改完資料庫執行個體的儲存空間大小之後，其狀態呈現為 storage-optimization (儲存最佳化)。

Note

儲存體最佳化可能需要幾個小時。在六 (6) 小時或在執行個體上完成儲存體最佳化 (以時間較長者為準) 後，您才能進行進一步的儲存體修改。您可以在中檢視儲存區最佳化進度，AWS Management Console 或使用 [描述-db-instance 命](#) AWS CLI 命。

但若有 SQL Server 資料庫執行個體，且自 2017 年 11 月起即未修改儲存組態，則為特殊情況。在此情況中，當您修改資料庫執行個體增加配置的儲存體時，可能會發生幾分鐘的短暫停機。停機之後，資料庫執行個體會在线上，但處於 storage-optimization 狀態。儲存最佳化期間效能可能會降低。

Note

儲存體配置完畢後，您無法縮減資料庫執行個體的儲存量。當您嘗試增加配置儲存時，其必須至少是 10%。如果您嘗試以低於 10% 的值增加該值，便會發生錯誤。

主控台

增加資料庫執行個體的儲存容量

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。
4. 選擇 Modify (修改)。
5. 針對 Allocated storage (配置儲存) 輸入新的值。必須大於目前的值。

Storage type

General Purpose (SSD) ▼

Allocated storage

16384

GiB

This instance supports multiple storage ranges between 20 and 16384 GiB. [See all](#)**Scaling your instance storage can:**

- Deplete the initial General Purpose (SSD) I/O credits, leading to longer conversion times. [Learn more](#)
- Impact instance performance until operation completes. [Learn more](#)

6. 選擇 Continue (繼續) 以前往下一個畫面。
7. 若要對資料庫執行個體立即套用儲存變更，請在 Scheduling of modifications (修改排程) 區段中選擇 Apply immediately (立即套用)。或選擇 Apply during the next scheduled maintenance window (在下一個排定的維護時段套用) 在下個維護時段套用變更。
8. 若對設定感到滿意，請選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要增加資料庫執行個體的儲存空間，請使用 AWS CLI 指令 [modify-db-instance](#)。設定下列參數：

- `--allocated-storage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `--apply-immediately` – 使用 `--apply-immediately` 可立即套用儲存變更。

或使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。套用變更後會造成立即停機。

如需儲存體的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

RDS API

若要增加資料庫執行個體的儲存體，請使用 Amazon RDS API 操作 [ModifyDBInstance](#)。設定下列參數：

- `AllocatedStorage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `ApplyImmediately` – 將此選項設為 `True` 可立即套用儲存變更。將此選項設為 `False` (預設值) 在下個維護時段套用變更。套用變更後會造成立即停機。

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

使用 Amazon RDS 儲存體自動擴展自動管理容量

如果您的工作負載是不可預測，您可啟用 Amazon RDS 資料庫執行個體的儲存體自動擴展。若要如此做，您可以使用 Amazon RDS 主控台、Amazon RDS API 或 AWS CLI。

例如，您可針對使用者快速採用的行動遊戲應用程式使用此功能。在此例中，快速增加的工作負載可能會超過可用的資料庫儲存體。為免必須手動擴充資料庫儲存體規模，您可使用 Amazon RDS 儲存體自動擴展。

在啟用儲存體自動擴展時，當 Amazon RDS 偵測到您用完可用資料庫空間，它會自動向上擴展您的儲存體。當下列因素符合時，Amazon RDS 會開始修改已啟用自動擴展的資料庫執行個體儲存體：

- 免費的可用空間小於或等於 10% 的已配置儲存體。
- 低儲存體狀況至少持續五分鐘。
- 自上次修改儲存後，或在執行個體上完成儲存體最佳化後 (以時間較長者為準)，至少已經過了六個小時。

額外的儲存體以下列任何較大一項增量：

- 10 GiB
- 目前配置之儲存體的 10%
- 依據過去一小時的 `FreeStorageSpace` 指標，預測儲存體於接下來 7 小時內超過目前配置的儲存體大小。有關指標的詳細資訊，請參閱[使用 Amazon 進行監控 CloudWatch](#)。

最大儲存閾值是您為自動擴展資料庫執行個體設定的限制。它有下列限制條件：

- 您必須將最大儲存閾值設定為至少比目前分配的儲存多 10%。我們建議將其設定為至少 26% 以上，以避免接收 [event notification](#) 儲存體大小接近最大儲存閾值。

例如，如果您的資料庫執行個體分配了 1000 GiB 的儲存，則將最大儲存閾值設定為至少 1100 GiB。否則，您會收到一個錯誤，例如無效的 `engine_name` 最大儲存大小。但是，我們建議您將最大儲存閾值設定為至少 1260 GiB，以避免此事件。

- 對於使用佈建 IOPS (io1 或 io2 區塊快速) 儲存的資料庫執行個體，IOPS 與最大儲存臨界值的比率 (以 GiB 為單位) 必須在特定範圍內。如需詳細資訊，請參閱 [佈建 IOPS SSD 儲存體](#)。
- 您無法為已啟用自動規模調整的執行個體，將儲存體閾值上限設為大於配置的儲存體上限。

例如，db.m5.xlarge 上的 SQL Server Standard Edition 預設配置給執行個體的儲存為 20 GiB (最小)，配置的儲存上限為 16,384 GiB。自動擴展的預設儲存體閾值上限為 1,000 GiB。如果您使用此預設值，執行個體不會自動調整到超過 1,000 GiB。即使配置給執行個體的儲存體上限為 16,384 GiB，也是如此。

Note

建議您根據使用模式和客戶需求，仔細選擇最大儲存閾值。如果使用模式有任何差異，當自動調整預測非常高的臨界值時，最大儲存閾值可以防止將儲存體縮放為意外高的值。將資料庫執行個體自動擴展後，就無法減少其配置的儲存體。

主題

- [限制](#)
- [啟用新資料庫執行個體的儲存體自動擴展](#)
- [變更資料庫執行個體的儲存體自動擴展設定](#)
- [關閉資料庫執行個體的儲存體自動擴展](#)

限制

下列限制適用於儲存體自動擴展：

- 如果儲存增量會超出儲存體上限閾值，則不會自動調整規模。
- 自動擴展時，RDS 會為後續自動擴展作業預測儲存體大小。若預測後續作業超過最大儲存體閾值，則 RDS 會自動擴展至最大儲存體閾值。
- 自動擴展無法完全防止大量資料載入的儲存已滿情況。這是因為在六 (6) 小時或在執行個體上完成儲存體最佳化之前 (以時間較長者為準)，無法進一步修改儲存。

如果您執行大型資料載入，且自動擴展未提供足夠的空間，資料庫可能會維持儲存已滿狀態數小時。這可能會損害資料庫。

- 如果您啟動儲存體擴展操作的時間和 Amazon RDS 啟動自動擴展操作的時間一樣，會先執行您的儲存體修改操作。自動擴展已取消。
- 自動調度資源無法減少配置的儲存空間。儲存體配置完畢後，您無法縮減資料庫執行個體的儲存量。
- 自動擴展不能與磁性儲存搭配使用。
- 自動擴展不能與下列可排序儲存體少於 6 TiB 的前一代執行個體類別搭配使用：
db.m3.large、db.m3.xlarge 及 db.m3.2xlarge。
- 自動調度資源操作不會記錄。AWS CloudTrail 如需詳細資訊 CloudTrail，請參閱 [在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)。

雖然自動擴展可幫助您動態增加 Amazon RDS 資料庫執行個體上的儲存體，但您仍應將資料庫執行個體的初始儲存體設定為適合您一般工作負載的大小。

啟用新資料庫執行個體的儲存體自動擴展

當您建立新的 Amazon RDS 資料庫執行個體時，您可選擇是否啟用儲存體自動擴展。您也可以設定 Amazon RDS 可為資料庫執行個體配置的儲存體上限。

Note

當您複製已啟用儲存體自動擴展的 Amazon RDS 資料庫執行個體時，複製的執行個體不會自動繼承該設定。新資料庫執行個體的配置儲存量 and 原始執行個體一樣。如果複製的執行個體儲存體需求持續增加，您可再次開啟新執行個體的儲存體自動擴展。

主控台

啟用新資料庫執行個體的儲存體自動擴展

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇您要建立資料庫執行個體的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇建立資料庫。在 Select engine (選取引擎) 頁面上選擇資料庫引擎，並指定資料庫執行個體資訊，如 [Amazon RDS 入門](#) 所述。

5. 在 Storage autoscaling (儲存體自動擴展) 區段中，設定資料庫執行個體的 Maximum storage threshold (儲存體閾值上限) 值。
6. 指定其餘的資料庫執行個體資訊，如[Amazon RDS 入門](#)所述。

AWS CLI

若要啟用新資料庫執行個體的儲存自動調度資源功能，請使用命 AWS CLI 令 [create-db-instance](#)。設定下列參數：

- `--max-allocated-storage` – 開啟儲存體自動擴展，並設定儲存體大小上限 (GiB)。

若要確認 Amazon RDS 儲存自動調度資源功能可用於您的資料庫執行個體，請使用命 AWS CLI [describe-valid-db-instance-modifications](#) 令。若要先依執行個體類別勾選再建立執行個體，請使用 [describe-orderable-db-instance-options](#) 命令。檢查傳回值的以下欄位：

- `SupportsStorageAutoscaling` – 指出是資料庫執行個體還是執行個體類別支援儲存體自動擴展。

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

RDS API

若要啟用新資料庫執行個體的儲存體自動擴展，請使用 Amazon RDS API 操作 [CreateDBInstance](#)。設定下列參數：

- `MaxAllocatedStorage` – 開啟 Amazon RDS 儲存體自動擴展，並設定儲存體大小上限 (GiB)。

若要驗證 Amazon RDS 儲存體自動擴展是否可供資料庫執行個體使用，請針對現有的執行個體使用 Amazon RDS API [DescribeValidDbInstanceModifications](#) 操作，或先使用 [DescribeOrderableDBInstanceOptions](#) 操作再建立執行個體。檢查傳回值的以下欄位：

- `SupportsStorageAutoscaling` – 指出資料庫執行個體是否支援儲存體自動擴展。

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

變更資料庫執行個體的儲存體自動擴展設定

您可開啟現有 Amazon RDS 資料庫執行個體的儲存體自動擴展。您也可以變更 Amazon RDS 可為資料庫執行個體配置的儲存體上限。

主控台

變更資料庫執行個體的儲存體自動擴展設定

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要修改的資料庫執行個體，然後選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 變更 Autoscaling (自動擴展) 區段的儲存體限制。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
5. 當所有變更都如您所想時，請選擇 Continue (繼續) 並查看修改內容。
6. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB Instance (修改資料庫執行個體) 儲存變更。如果都不正確，請選擇 Back (上一步) 編輯變更，或選擇 Cancel (取消) 取消變更。

立即會變更儲存體自動調整規則限制。此設定會忽略 Apply immediately (立即套用) 設定。

AWS CLI

若要變更資料庫執行個體的儲存自動調度資源設定，請使用指 AWS CLI 令 [modify-db-instance](#)。設定下列參數：

- `--max-allocated-storage` – 設定儲存體大小上限 (GiB)。如果此值大於 `--allocated-storage` 參數，即開啟儲存體自動擴展。如果此值與 `--allocated-storage` 參數相同，即關閉儲存體自動擴展。

若要確認 Amazon RDS 儲存自動調度資源功能可用於您的資料庫執行個體，請使用命 AWS CLI [describe-valid-db-instance-modifications](#) 令。若要先依執行個體類別勾選再建立執行個體，請使用 [describe-orderable-db-instance-options](#) 命令。檢查傳回值的以下欄位：

- `SupportsStorageAutoscaling` – 指出資料庫執行個體是否支援儲存體自動擴展。

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

RDS API

若要變更資料庫執行個體的儲存體自動擴展設定，請使用 Amazon RDS API 操作 [ModifyDBInstance](#)。設定下列參數：

- MaxAllocatedStorage – 設定儲存體大小上限 (GiB)。

若要驗證 Amazon RDS 儲存體自動擴展是否可供資料庫執行個體使用，請針對現有的執行個體使用 Amazon RDS API [DescribeValidDbInstanceModifications](#) 操作，或先使用 [DescribeOrderableDBInstanceOptions](#) 操作再建立執行個體。檢查傳回值的以下欄位：

- SupportsStorageAutoscaling – 指出資料庫執行個體是否支援儲存體自動擴展。

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

關閉資料庫執行個體的儲存體自動擴展

如果不再需要 Amazon RDS 自動增加 Amazon RDS 資料庫執行個體的儲存體，您可關閉儲存體自動擴展。如此做之後，您仍然可以手動增加資料庫執行個體的儲存量。

主控台

關閉資料庫執行個體的儲存體自動擴展

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要修改的資料庫執行個體，然後選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 清除 Storage autoscaling (儲存體自動擴展) 區段中的 Enable storage autoscaling (啟用儲存體自動擴展) 核取方塊。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
5. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看修改內容。
6. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB Instance (修改資料庫執行個體) 儲存變更。如果都不正確，請選擇 Back (上一步) 編輯變更，或選擇 Cancel (取消) 取消變更。

立即會變更儲存體自動調整規則限制。此設定會忽略 Apply immediately (立即套用) 設定。

AWS CLI

若要關閉資料庫執行個體的儲存自動調度資源功能，請使用 AWS CLI 命令 [modify-db-instance](#) 和下列參數：

- `--max-allocated-storage` – 指定值等於 `--allocated-storage` 的設定，可防止指定資料庫執行個體進一步的 Amazon RDS 儲存體自動擴展。

如需儲存體的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

RDS API

使用 Amazon RDS API 操作 [ModifyDBInstance](#) 來關閉資料庫執行個體的儲存體自動擴展。設定下列參數：

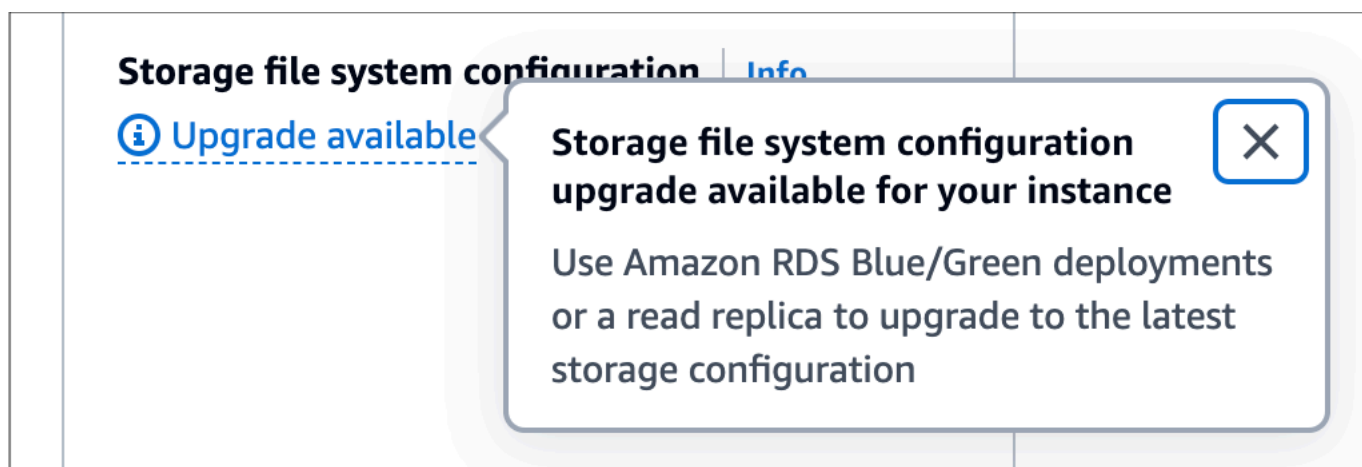
- `MaxAllocatedStorage` – 指定值等於 `AllocatedStorage` 的設定，可防止指定資料庫執行個體進一步的 Amazon RDS 儲存體自動擴展。

如需儲存體的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

升級資料庫執行個體的儲存體檔案系統

大多數 RDS 資料庫執行個體都提供 64 TiB 的儲存體大小上限，MariaDB 用於雲端資料庫、MySQL 和 PostgreSQL 資料庫。某些舊型 32 位元檔案系統具有較低的儲存容量。若要判斷資料庫執行個體的儲存容量，您可以使用 [描述驗證的 DB 執行個體修改命](#) AWS CLI 令。

如果 RDS 偵測到其中一個資料庫執行個體正在執行舊版檔案系統 (具有 16 TiB 儲存體大小、2 TiB 檔案大小限制或未最佳化寫入的檔案系統)，RDS 主控台即會通知您，檔案系統組態符合升級資格。您可以在資料庫執行個體詳細資訊頁面的儲存體面板上檢查資料庫執行個體的升級資格。



如果您的資料庫執行個體符合檔案系統升級的資格，您可以使用下列兩種方式之一來執行升級：

- 建立藍/綠部署，並指定升級儲存體檔案系統組態。此選項會將綠色環境中的檔案系統升級為偏好的組態。然後，您可以轉換藍/綠部署，將綠色環境提升至新的生產環境。如需詳細說明，請參閱 [the section called “建立藍/綠部署”](#)。
- 建立資料庫執行個體僅供讀取複本，並指定升級儲存體檔案系統組態。此選項會將僅供讀取複本的檔案系統升級為偏好的組態。您接著可以將僅供讀取複本提升為獨立的執行個體。如需詳細說明，請參閱 [the section called “建立僅供讀取複本”](#)。

升級儲存體組態是 I/O 密集型操作，且會導致僅供讀取複本和藍/綠部署的建立時間延長。如果來源資料庫執行個體使用佈建 IOPS SSD (io1 或 io2 Block Express) 儲存，且您佈建了綠色環境或僅供讀取複本的執行個體大小為 4xlarge 或更大，則儲存升級程序會更快。涉及一般用途 SSD (gp2) 儲存體的儲存升級可能會耗盡您的 I/O 用量餘額，導致升級時間更長。如需詳細資訊，請參閱 [the section called “資料庫執行個體儲存體”](#)。

在儲存體升級程序期間，資料庫引擎無法使用。如果來源資料庫執行個體的儲存耗用量大於或等於已配置儲存體大小的 90%，並且啟用儲存體自動調度資源，則儲存體升級程序會將綠色執行個體或僅供讀取複本配置的儲存大小增加 10%。如果停用儲存區自動調度資源功能，則在升級期間不會增加儲存區大小。

修改佈建 IOPS SSD 儲存體的設定

您可使用 Amazon RDS 主控台、AWS CLI 或 Amazon RDS API 來修改使用佈建 IOPS SSD 儲存體的資料庫執行個體設定。指定儲存類型、配置的儲存體，以及您需要的佈建 IOPS 數量。範圍視您的資料庫引擎和執行個體類型而定。

雖然您可以縮減執行個體中佈建的 IOPS 數量，但您無法減少儲存體大小。

在大多數情況下，擴展儲存體完全不需要停機，也不會降低伺服器的效能。修改完資料庫執行個體的儲存空間 IOPS 之後，其狀態呈現為 storage-optimization (儲存最佳化)。

Note

儲存體最佳化可能需要幾個小時。在六 (6) 小時或在執行個體上完成儲存體最佳化 (以時間較長者為準) 後，您才能進行進一步的儲存體修改。

如需每個資料庫引擎可用的已配置儲存體和佈建 IOPS 範圍的相關資訊，請參閱 [佈建 IOPS SSD 儲存體](#)。

主控台

變更資料庫執行個體的佈建 IOPS。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

若要篩選資料庫執行個體清單，請在 Filter databases (篩選資料庫) 中輸入文字字串，供 Amazon RDS 用來篩選結果。只有名稱中包含此字串的資料庫執行個體會顯示出來。

3. 選擇您想要修改且擁有佈建 IOPS 的資料庫執行個體。
4. 選擇 Modify (修改)。
5. 在 [修改資料庫執行個體] 頁面上，針對儲存類型選擇佈建 IOPS SSD (io1) 或佈建 IOPS SSD (io2)。
6. 針對 Provisioned IOPS (佈建 IOPS)，輸入一值。

如果您針對 Allocated storage (已配置儲存體) 或 Provisioned IOPS (佈建 IOPS) 指定的值超出其他參數支援的限制，則會顯示警告訊息。此訊息提供其他參數所需的數值範圍。

7. 選擇 Continue (繼續)。
8. 在 Scheduling of modifications (修改排程) 區段中選擇 Apply immediately (立即套用)，以將變更立即套用至資料庫執行個體。或選擇 Apply during the next scheduled maintenance window (在下一個排定的維護時段套用) 在下個維護時段套用變更。
9. 檢閱要變更的參數，並選擇 Modify DB instance (修改資料庫執行個體) 以完成修改。

配置儲存或佈建 IOPS 的新值都會出現在 Status (狀態) 欄中。

AWS CLI

若要變更資料庫執行個體的佈建 IOPS 設定，請使用命 AWS CLI 令 [modify-db-instance](#)。設定下列參數：

- `--storage-type`— 針 `io2` 對已佈建 IOPS 設定為 `io1` 或。
- `--allocated-storage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `--iops` – 資料庫執行個體佈建 IOPS 的新數量，以每秒輸入/輸出操作量表示。
- `--apply-immediately` – 使用 `--apply-immediately` 立即套用變更。使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。

RDS API

如要變更資料庫執行個體的佈建 IOPS 設定，請使用 Amazon RDS API 操作 [ModifyDBInstance](#)。設定下列參數：

- `StorageType`— 針 `io2` 對已佈建 IOPS 設定為 `io1` 或。
- `AllocatedStorage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `Iops` – 資料庫執行個體的新 IOPS 速率，以每秒輸入/輸出操作量表示。
- `ApplyImmediately` – 將此選項設為 `True` 立即套用變更。將此選項設為 `False` (預設值) 在下個維護時段套用變更。

I/O 密集型儲存體修改

Amazon RDS 資料庫執行個體將 Amazon Elastic Block Store (EBS) 磁碟區用於資料庫和日誌儲存體的 Amazon RDS 資料庫執行個體。根據所請求儲存的數量，Amazon RDS 會自動將多個 Amazon EBS 磁碟區分成等量以增強效能。具有 SSD 儲存類型的 RDS 資料庫執行個體由 RAID 0 組態中的一個或四個分段的 Amazon EBS 磁碟區提供支援。根據設計，RDS 資料庫執行個體的儲存修改作業對進行中資料庫作業的影響最小。

在大多數情況下，儲存擴展修改會完全卸載到 Amazon EBS 層，而且對資料庫是透明的。此程序通常會在幾分鐘內完成。但是，某些較舊的 RDS 儲存磁碟區需要不同的程序來修改大小、佈建 IOPS 或儲存區類型。這涉及使用潛在 I/O 密集型操作來製作數據的完整副本。

如果符合下列任一因素，則儲存裝置修改會使用 I/O 密集型作業：

- 來源儲存類型為磁性。磁性儲存裝置不支援彈性磁碟區修改。
- RDS 資料庫執行個體不在一或四磁碟區的 Amazon EBS 配置上。您可以使用增強型監控指標，檢視 RDS 資料庫執行個體上使用中的 Amazon EBS 磁碟區數量。如需詳細資訊，請參閱 [在 RDS 主控台中檢視作業系統指標](#)。
- 修改要求的目標大小會將配置的儲存區增加到超過 400 GiB (適用於 MariaDB、MySQL 和 PostgreSQL 執行個體的 RDS)，以及適用於 Oracle 的 RDS 的 200 GiB。儲存自動調度資源作業將資料庫執行個體的配置儲存大小增加到超過這些閾值時，也會產生相同的效果。

如果您的儲存修改涉及 I/O 密集型作業，它會消耗 I/O 資源並增加資料庫執行個體的負載。若要將 I/O 密集型作業 (包含一般用途 SSD (gp2)) 的 I/O 密集型作業，並導致更長的轉換時間，並導致更長的轉換時間。

我們建議您將這些儲存區修改要求排程在尖峰時段以外的最佳作法，以協助縮短完成儲存區修改作業所需的時間。或者，您也可以建立資料庫執行個體的僅供讀取複本，並在僅供讀取複本上執行儲存體修改。然後將僅供讀取複本提升為主要資料庫執行個體。如需詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

如需詳細資訊，請參閱《[當我嘗試增加分配的儲存時，為什麼 Amazon RDS 資料庫執行個體卡在修改狀態？](#)》

修改一般用途 SSD (gp3) 儲存體的設定

您可以使用 Amazon RDS 主控台或 Amazon RDS API 修改使用一般用途 SSD (gp3) 儲存的資料庫執行個體的設定。AWS CLI 指定您需要的儲存體類型、配置的儲存體、佈建 IOPS 數量，以及儲存體輸送量。

雖然您可以減少資料庫執行個體的佈建 IOPS 和儲存輸送量，但無法縮減儲存大小。

在大多數情況下，擴展儲存體並不需要任何中斷。修改完資料庫執行個體的儲存空間 IOPS 之後，其狀態呈現為 storage-optimization (儲存最佳化)。在儲存體最佳化期間，您可以預期延遲提高，但仍在個位數字毫秒範圍內。對儲存體進行修改後，資料庫執行個體完全可以正常運作。

Note

在儲存空間在執行個體上完成最佳化之後，必須經過 6 個小時，您才能在進行進一步的儲存空間修改。

如需每個資料庫引擎可用的已配置儲存體、佈建 IOPS、儲存體輸送量的範圍相關資訊，請參閱 [gp3 儲存裝置 \(建議使用\)](#)。

主控台

變更資料庫執行個體的儲存體效能設定

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

若要篩選資料庫執行個體清單，請在 Filter databases (篩選資料庫) 中輸入文字字串，供 Amazon RDS 用來篩選結果。只有名稱中包含此字串的資料庫執行個體會顯示出來。

3. 選擇具有您想要修改之 gp3 儲存體的資料庫執行個體。

4. 選擇 Modify (修改)。
5. 在 Modify DB Instance (修改資料庫執行個體) 頁面上，選擇 General Purpose SSD (gp3) (一般用途 SSD (gp3)) 作為 Storage type (儲存體類型)，然後執行下列動作：
 - a. 針對 Provisioned IOPS (佈建 IOPS)，選擇一值。

如果您針對 Allocated storage (已配置儲存體) 或 Provisioned IOPS (佈建 IOPS) 指定的值超出其他參數支援的限制，則警告訊息便會出現。此訊息提供其他參數所需的數值範圍。
 - b. 在 Storage throughput (儲存體輸送量) 中，選擇一值。

如果您針對 Provisioned IOPS (佈建 IOPS) 或 Storage throughput (儲存體輸送量) 指定的值超出其他參數支援的限制，則警告訊息便會出現。此訊息提供其他參數所需的數值範圍。
6. 選擇 Continue (繼續)。
7. 在 Scheduling of modifications (修改排程) 區段中選擇 Apply immediately (立即套用)，以將變更立即套用至資料庫執行個體。或選擇 Apply during the next scheduled maintenance window (在下一個排定的維護時段套用) 在下一個維護時段套用變更。
8. 檢閱要變更的參數，並選擇 Modify DB instance (修改資料庫執行個體) 以完成修改。

佈建 IOPS 的新值會出現在 Status (狀態) 欄中。

AWS CLI

若要變更資料庫執行個體的儲存效能設定，請使用 AWS CLI 指令 [modify-db-instance](#)。設定下列參數：

- `--storage-type` – 設為 `gp3` 表示一般用途 SSD (gp3)。
- `--allocated-storage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `--iops` – 資料庫執行個體佈建 IOPS 的新數量，以每秒輸入/輸出操作量表示。
- `--storage-throughput`— 資料庫執行個體的新儲存輸送量，以表示 MiBps。
- `--apply-immediately` – 使用 `--apply-immediately` 立即套用變更。使用 `--no-apply-immediately` (預設值) 在下一個維護時段套用變更。

RDS API

若要變更資料庫執行個體的儲存體效能設定，請使用 Amazon RDS API 操作 [ModifyDBInstance](#)。設定下列參數：

- `StorageType` – 設為 `gp3` 表示一般用途 SSD (`gp3`)。
- `AllocatedStorage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `Iops` – 資料庫執行個體的新 IOPS 速率，以每秒輸入/輸出操作量表示。
- `StorageThroughput`— 資料庫執行個體的新儲存輸送量，以表示 MiBps。
- `ApplyImmediately` – 將此選項設為 `True` 立即套用變更。將此選項設為 `False` (預設值) 在下個維護時段套用變更。

使用專用日誌磁碟區 (DLV)

您可以針對使用佈建 IOPS (PIOPS) 儲存的資料庫執行個體使用專用記錄磁碟區 (DLV)。DLV 會將 PostgreSQL 資料庫交易記錄檔和 MySQL/MariaDB 重做記錄檔和二進位記錄移至儲存磁碟區 (與包含資料庫表格的磁碟區分開)。DLV 使交易寫入記錄更有效率且一致。DLV 非常適合具有大型配置儲存體、每秒高 I/O (IOPS) 需求或延遲敏感工作負載的資料庫。

DLV 可用於 PIOPS 儲存體 (`io1` 和 `io2` 區塊快速)，而且建立的固定大小為 1,000 GiB 和 3,000 個佈建 IOPS。

Amazon RDS 支援以下版本 AWS 區域的所有 DLV：

- MariaDB 10.6.7 和更新的 10 版本
- MySQL 8.0.28 和更新的 8.版本
- PostgreSQL 13.10 和更新的 13 版本、14.7 和更新的 14 版本，以及 15.2 和更新的 15 版本

RDS 支援 DLV 搭配多可用區部署。修改或建立異地同步備份執行個體時，會同時為主要和次要執行個體建立 DLV。

RDS 支援 DLV 搭配僅供讀取複本。如果主要資料庫執行個體已啟用 DLV，則在啟用 DLV 之後建立的所有僅供讀取複本也將具有 DLV。在切換到 DLV 之前建立的任何僅供讀取複本將不會啟用它，除非明確修改為要這樣做。建議在啟用 DLV 之前附加至主要執行個體的所有僅供讀取複本，您也須將其手動修改為具有 DLV。

Note

針對 5 TiB 或更高的資料庫組態，建議使用專用日誌磁碟區。

如需每個資料庫引擎可用的已配置儲存體、佈建 IOPS、儲存體輸送量的範圍相關資訊，請參閱 [佈建 IOPS SSD 儲存體](#)。

建立資料庫執行個體時啟用 DLV

您可以使用 AWS Management Console、AWS CLI、或 RDS API 建立已啟用 DLV 的資料庫執行個體。

主控台

在新的資料庫執行個體上啟用 DLV

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇建立資料庫。
3. 在 [建立資料庫執行個體] 頁面上，選擇支援 DLV 的資料庫引擎。
4. 對於存儲：
 - a. 選擇佈建的 IOPS 固態硬碟 (io1) 或佈建 IOPS 固態硬碟 (io2)。
 - b. 輸入您想要的已配置儲存區和已佈建 IOPS。
 - c. 展開專用記錄磁碟區，然後選取 [開啟專用記錄磁碟區]。

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

Provisioned IOPS SSD (io2)
Low latency, highly durable, I/O intensive storage

Allocated storage [Info](#)

100 GiB

The minimum value is 100 GiB and the maximum value is 65,536 GiB

i After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. [Learn more](#)

Provisioned IOPS [Info](#)

3000 IOPS

The minimum value is 1,000 IOPS and the maximum value is 160,000 IOPS. The IOPS to GiB ratio must be between 0.5 and 1,000

i Your actual IOPS might vary from the amount that you provisioned based on your database workload and instance type. [Learn more](#)

▶ **Storage autoscaling**

▼ **Dedicated Log Volume**

Dedicated Log Volume [Info](#)
Dedicated Log Volumes store database transaction logs on a dedicated volume to improve write performance for latency sensitive workloads. There is additional cost associated with this feature.

Turn on Dedicated Log Volume

i We recommend this for larger databases with latency sensitivity.

5. 視需要選擇其他設定。

6. 選擇建立資料庫。

建立資料庫之後，「專用記錄磁碟區」的值就會顯示在資料庫詳細資訊頁面的「組態」頁籤上。

CLI

若要在使用佈建 IOPS 儲存建立資料庫執行個體時啟用 DLV，請使用 [AWS CLI 指令建立 db-instance](#)。設定下列參數：

- `--dedicated-log-volume`— 啟用專用的記錄檔磁碟區。

- `--storage-type`— 針io2對已佈建 IOPS 設定為io1或。
- `--allocated-storage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `--iops`— 資料庫執行個體的佈建 IOPS 數量，以每秒 I/O 作業表示。

RDS API

若要在使用佈建 IOPS 儲存建立資料庫執行個體時啟用 DLV，請使用 [Amazon RDS API 操作建 CreateDBInstance](#)。設定下列參數：

- `DedicatedLogVolume`— 設定為啟true用專用記錄磁碟區。
- `StorageType`— 針io2對已佈建 IOPS 設定為io1或。
- `AllocatedStorage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。
- `Iops`— 資料庫執行個體的 IOPS 速率，以每秒 I/O 作業數表示。

在現有資料庫執行個體上啟用 DLV

您可以使用 AWS Management Console AWS CLI、或 RDS API 修改資料庫執行個體以啟用 DLV。

修改資料庫執行個體的 DLV 設定後，您必須重新啟動資料庫執行個體。

主控台

在現有資料庫執行個體上啟用 DLV

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

若要篩選資料庫執行個體清單，請在 Filter databases (篩選資料庫) 中輸入文字字串，供 Amazon RDS 用來篩選結果。只有名稱中包含此字串的資料庫執行個體會顯示出來。

3. 選擇您要修改的已佈建 IOPS 儲存體的資料庫執行個體。
4. 選擇 Modify (修改)。
5. 在修改資料庫執行個體頁面上：
 - 對於儲存，請展開專用記錄磁碟區，然後選取開啟專用記錄磁碟區。
6. 選擇繼續。

7. 選擇立即套用，立即將變更套用至資料庫執行個體。或選擇 Apply during the next scheduled maintenance window (在下一個排定的維護時段套用) 在下個維護時段套用變更。
8. 檢閱要變更的參數，並選擇 Modify DB instance (修改資料庫執行個體) 以完成修改。

專用記錄磁碟區的新值會顯示在資料庫詳細資訊頁面的 [組態] 索引標籤上。

CLI

若要使用佈建 IOPS 儲存在現有資料庫執行個體上啟用或停用 DLV，請使用命令。AWS CLI [modify-db-instance](#) 設定下列參數：

- `--dedicated-log-volume`— 啟用專用的記錄檔磁碟區。
使用 `--no-dedicated-log-volume` (預設值) 停用專用記錄磁碟區。
- `--apply-immediately` – 使用 `--apply-immediately` 立即套用變更。
使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。

RDS API

若要使用佈建 IOPS 儲存體，在現有資料庫執行個體上啟用或停用 DLV，請使用 Amazon RDS API 操作 [ModifyDBInstance](#)。設定下列參數：

- `DedicatedLogVolume`— 將此選項設定 `true` 為啟用專用記錄磁碟區。
將此選項設定 `false` 為停用專用記錄磁碟區。這是預設值。
- `ApplyImmediately` – 將此選項設為 `True` 立即套用變更。
將此選項設為 `False` (預設值) 在下個維護時段套用變更。

刪除資料庫執行個體

您可以使用 AWS Management Console、或 RDS API 刪除資料庫執行個體。AWS CLI 如果您想刪除 Aurora 資料庫叢集中的資料庫執行個體，請參閱[刪除 Aurora 資料庫叢集和資料庫執行個體](#)。

主題

- [刪除資料庫執行個體的先決條件](#)
- [刪除資料庫執行個體時的考量](#)
- [刪除資料庫執行個體](#)

刪除資料庫執行個體的先決條件

在嘗試刪除資料庫執行個體前，請務必關閉刪除保護功能。預設會開啟使用主控台建立之資料庫執行個體的刪除保護功能。

如果您的資料庫執行個體已開啟刪除保護，您可以透過修改執行個體設定將其關閉。在資料庫詳細資訊頁面中選擇 [修改]，或呼叫[modify-db-instance](#)指令。此操作不會導致中斷。如需詳細資訊，請參閱[資料庫執行個體的設定](#)。

刪除資料庫執行個體時的考量

刪除資料庫執行個體會影響執行個體復原性、備份可用性和僅供讀取複本狀態。請考慮以下問題：

- 您可以選擇是否建立最終資料庫快照。您有下列選項：
 - 如果您建立最終快照，可以使用它來還原已刪除的資料庫執行個體。RDS 會保留您先前建立的最終快照和任何手動快照。如果您的資料庫執行個體未處於 Available 狀態，則無法建立其最終資料庫快照。如需詳細資訊，請參閱[檢視 Aurora 叢集中 資料庫執行個體狀態](#)。
 - 如果您沒有拍攝最終快照，刪除執行個體的速度會更快。缺點是沒有最終快照存在，您可以稍後還原。如果您決定還原已刪除的資料庫執行個體，請保留自動備份，或使用較早的手動快照將資料庫執行個體還原到先前快照的時間點。
- 您可以選擇是否保留自動備份。您有下列選項：
 - 如果您保留自動備份，在您刪除時 RDS 會將其保留對資料庫執行個體生效的保留期間。您可以使用自動備份將資料庫執行個體還原到保留期內的某個時間 (但不超過保留期)。無論您是否選擇建立最終資料庫快照，保留期都有效。若要刪除保留的自動備份，請參閱[刪除保留的自動備份](#)。
 - 保留的自動備份和手動快照會產生帳單費用，直到將其刪除為止。如需詳細資訊，請參閱[保留成本](#)。

- 如果您不保留自動備份，RDS 會刪除與資料庫執行個體相 AWS 區域 同的自動備份。您無法復原這些備份。如果您的自動備份已複製到另一個 AWS 區域，即使您不選擇保留自動備份，RDS 仍會將其保留。如需詳細資訊，請參閱 [將自動備份複製到另一個 AWS 區域](#)。

Note

通常，如果有建立最終資料庫快照，則不需要保留自動備份。

- 刪除資料庫執行個體時，RDS 不會刪除手動資料庫快照。如需詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。
- 如果您想刪除所有 RDS 資源，請注意下列資源會產生費用：
 - 資料庫執行個體
 - 資料庫快照
 - 資料庫叢集

如果您購買了預留執行個體，則會根據您購買執行個體時所同意的合約計費。如需詳細資訊，請參閱 [Amazon RDS 的預留資料庫執行個體](#)。您可以使用取得所有資 AWS 源的帳單資訊 AWS Cost Explorer。如需詳細資訊，請參閱 [使用分析成本 AWS Cost Explorer](#)。

- 如果刪除具有相同僅供讀取複本的資料庫執行個體 AWS 區域，則每個僅供讀取複本都會自動升級為獨立資料庫執行個體。如需詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。如果您的資料庫執行個體有不同的僅 [跨區域複製考量](#) 供讀取複本 AWS 區域，請參閱以取得刪除跨區域僅供讀取複本之來源資料庫執行個體的相關資訊。
- 當資料庫執行個體的狀態為 `deleting`，其 CA 憑證值不會顯示在 RDS 主控台或 AWS CLI 命令或 RDS API 作業的輸出中。如需憑證授權機構憑證的詳細資訊，請參閱。
- 刪除資料庫執行個體所需的時間會因備份保留期間 (也就是要刪除的備份數量)、刪除的資料量，以及是否拍攝最終快照而有所不同。

刪除資料庫執行個體

您可以使用 AWS Management Console、或 RDS API 刪除資料庫執行個體。AWS CLI 您必須執行下列動作：

- 提供資料庫執行個體的名稱
- 啟用或停用讓您建立執行個體最終資料庫快照的選項
- 啟用或停用讓您保留自動備份的選項

Note

開啟刪除保護時，無法刪除資料庫執行個體。如需詳細資訊，請參閱 [刪除資料庫執行個體的先決條件](#)。

主控台

刪除資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要刪除的資料庫執行個體。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。
4. 若要為資料庫執行個體建立最終的資料庫快照，請選擇 Create final snapshot? (是否建立最終快照?)。
5. 如果您選擇建立最終快照，請輸入最終快照名稱。
6. 若要保留自動備份，請選擇 Retain automated backups (保留自動備份)。
7. 在方塊中輸入 **delete me**。
8. 選擇 Delete (刪除)。

AWS CLI

若要尋找帳戶中資料庫執行個體的執行個體 ID，請呼叫以下 [describe-db-instances](#) 指令：

```
aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier]' --output text
```

若要使用刪除資料庫執行個體 AWS CLI，請使用下列選項呼叫 [delete-db-instance](#) 命令：

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` 或 `--skip-final-snapshot *`

Example 有最終快照但沒有保留的自動備份

對於 Linux/macOS、或 Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier mydbinstance \  
  --final-db-snapshot-identifier mydbinstancefinalsnapshot \  
  --delete-automated-backups
```

在 Windows 中：

```
aws rds delete-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --final-db-snapshot-identifier mydbinstancefinalsnapshot ^  
  --delete-automated-backups
```

Example 有保留的自動備份但沒有最終快照

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier mydbinstance \  
  --skip-final-snapshot \  
  --no-delete-automated-backups
```

在 Windows 中：

```
aws rds delete-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --skip-final-snapshot ^  
  --no-delete-automated-backups
```

RDS API

若要使用 Amazon RDS API 刪除資料庫執行個體，請搭配下列參數呼叫 [DeleteDBInstance](#) 操作：

- DBInstanceIdentifier
- FinalDBSnapshotIdentifier 或 SkipFinalSnapshot *

設定及管理多可用區部署

多可用區域部署可以有一個待命或兩個待命資料庫執行個體。當部署具有一個備用資料庫執行個體時，稱為多可用區域資料庫執行個體部署。多可用區域資料庫執行個體部署有一個待命資料庫執行個體，可提供容錯移轉支援，但是不提供讀取流量。當部署有兩個待命資料庫執行個體時，它稱為多可用區域資料庫叢集部署。多可用區域資料庫叢集部署具有待命資料庫執行個體，可提供容錯移轉支援，也可提供讀取流量。

您可以使用 AWS Management Console 來確定異地同步備份部署是多可用區域資料庫執行個體部署，還是多可用區域資料庫叢集部署。在導覽窗格中，選擇 Databases (資料庫)，然後選擇 DB identifier (資料庫識別符)。

- 多可用區域資料庫執行個體部署具有以下特徵：
 - 資料庫執行個體只有一行。
 - Role (角色) 的值為 Instance (執行個體) 或 Primary (主要)。
 - Multi-AZ (多可用區域) 的值為 Yes (是)。
- 多可用區域資料庫叢集部署具有以下特徵：
 - 有一個叢集層級行，其下有三個資料庫執行個體行。
 - 對於叢集層級行，Role (角色) 的值為 Multi-AZ DB cluster (多可用區域資料庫叢集)。
 - 對於每個執行個體層級行，Role (角色) 的值為 Writer instance (寫入器執行個體) 或 Reader instance (讀取器執行個體)。
 - 對於每個執行個體層級行，Multi-AZ (多可用區域) 的值為 3 Zones (3 個區)。

主題

- [多可用區域資料庫執行個體部署](#)
- [多可用區域資料庫叢集部署](#)

此外，下列主題同時適用於資料庫執行個體和多可用區域資料庫叢集：

- [the section called “標記 RDS 資源”](#)
- [the section called “使用 ARN”](#)
- [the section called “使用儲存體”](#)
- [the section called “維持資料庫執行個體”](#)

- [the section called “升級引擎版本”](#)

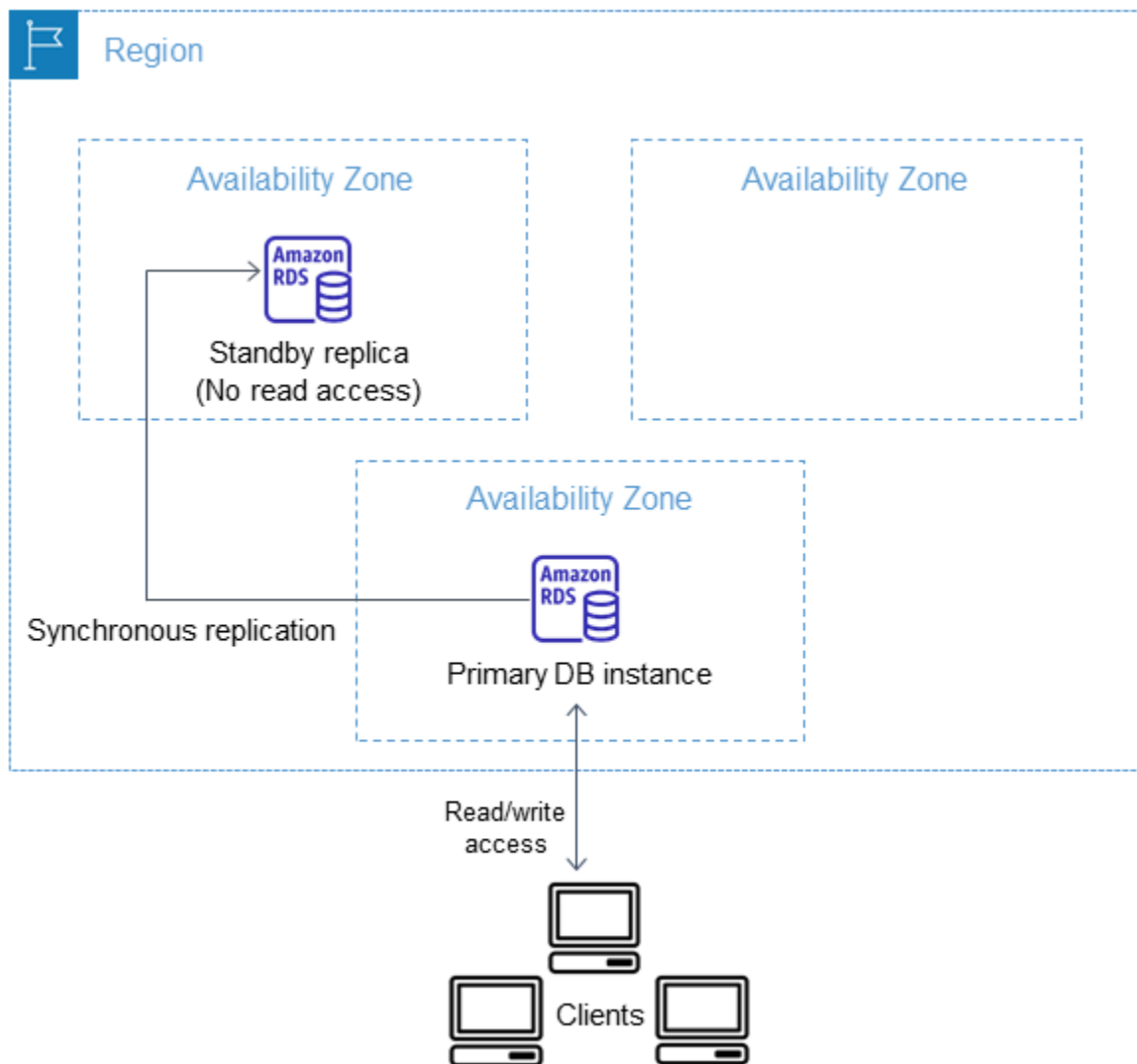
多可用區域資料庫執行個體部署

使用多可用區域部署搭配單一待命資料庫執行個體，Amazon RDS 可為資料庫執行個體提供高可用性和容錯移轉支援。這種類型的部署稱為多可用區域資料庫執行個體部署。Amazon RDS 使用幾個不同技術來提供這個容錯移轉支援。MariaDB、MySQL、Oracle、PostgreSQL 和 RDS Custom for SQL Server 資料庫執行個體的多可用區域部署會使用 Amazon 容錯移轉技術。而 SQL Server 資料庫執行個體則會使用 SQL Server 資料庫鏡像 (DBM) 技術或 Always On 可用性群組 (AG)。如需 SQL Server 版本異地同步備份支援的相關資訊，請參閱[Amazon RDS for Microsoft SQL Server 的異地同步備份部署](#)。如需針對多可用區域使用 RDS Custom for SQL Server 的相關資訊，請參閱[管理 RDS Custom for SQL Server 的多可用區部署](#)。

在 Multi-AZ 資料庫執行個體部署中，Amazon RDS 會自動佈建，並在不同的可用區域中維持同步待命複本。主要資料庫執行個體會跨可用區域，同步複寫到待命複本，提供資料備援並且降低系統備份時的延遲遽增發生等功能。執行具有高可用性的資料庫執行個體，可以在規劃好的系統維護期間增強可用性。它還有助於在資料庫執行個體失敗和可用區域中斷時保護資料庫。如需可用區域的詳細資訊，請參閱[區域、可用區域和 Local Zones](#)。

Note

高可用性選項不是唯讀案例的擴展解決方案。您無法使用待命複本來提供讀取流量。若要提供唯讀流量，請改用多可用區域資料庫叢集或僅供讀取複本。如需建立多可用區域資料庫叢集的詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。如需僅供讀取複本的詳細資訊，請參閱[使用資料庫執行個體僅供讀取複本](#)。



使用 RDS 主控台，只要在建立資料庫執行個體時指定多可用區域，即可輕鬆建立多可用區域資料庫執行個體部署。您可以使用主控台，藉由修改資料庫執行個體並指定多可用區域選項，將現有資料庫執行個體轉換為多可用區域資料庫執行個體部署。您也可以使用 AWS CLI 或 Amazon RDS API 指定異地同步備份資料庫執行個體部署。使用 [create-db-instance](#) 或 [modify-db-instance](#) CLI 命令，或 [CreateDBInstance](#) 或 [ModifyDBInstance](#) API 作業。

RDS 主控台會顯示待命複本的可用區域 (稱為次要 AZ)。您也可以使用 [describe-db-instances](#) CLI 命令或 [DescribeDBInstances](#) API 作業來尋找次要 AZ。

相較於單一可用區域部署，使用多可用區域資料庫執行個體部署的資料庫執行個體會增加的寫入和遞交延遲。這可能是因為發生的同步資料複寫。如果您的部署容錯移轉至待命複本，您可能會發生延遲變更，不過 AWS 是以可用區域之間的低延遲網路連線而設計。對於生產應用程式，我們建議使用佈建 IOPS (每秒輸入/輸出操作) 來取得快速、一致的效能。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

將資料庫執行個體修改為多可用區域資料庫執行個體部署

如果您的資料庫執行個體是在單一可用區域部署中，而想要將其修改為多可用區域資料庫執行個體部署 (用於 Amazon Aurora 以外的引擎)，Amazon RDS 會執行幾個動作：

1. 建立主要資料庫執行個體的 Amazon Elastic Block Store (EBS) 磁碟區快照。
2. 從快照為待命複本建立新磁碟區。這些磁碟區會在背景中初始化，並在資料完全初始化之後達到最大磁碟區效能。
3. 開啟主要複本和待命複本磁碟區之間的同步區塊層級複本。

Important

使用快照建立待命執行個體可避免從單一可用區轉換為多可用區域的停機時間，但在轉換為異地同步備份期間和之後可能會對效能產生影響。對於對寫入延遲敏感的工作負載而言，這個影響可能很重大。

雖然此功能可以從快照中還原大量磁碟區，但由於是同步複寫，可能會導致 I/O 作業延遲。此延遲可能會影響您的資料庫效能。我們強烈建議您不要在生產資料庫執行個體上執行異地同步備份轉換。

若要避免對目前提供敏感工作負載服務的資料庫執行個體造成效能影響，請建立僅供讀取複本並在僅供讀取複本上啟用備份。將僅供讀取複本轉換為異地同步備份，並執行將資料載入僅供讀取複本磁碟區的查詢 (在兩個異地上)。然後會將僅供讀取複本提升為主要資料庫執行個體。

如需詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

有兩種方式可將資料庫執行個體修改為多可用區域資料庫執行個體部署：

主題

- [使用 RDS 主控台將其轉換為多可用區域資料庫執行個體部署](#)
- [將資料庫執行個體修改為多可用區域資料庫執行個體部署](#)

使用 RDS 主控台將其轉換為多可用區域資料庫執行個體部署

您可以使用 RDS 主控台，將資料庫執行個體轉換為多可用區域資料庫執行個體部署。

您只能使用主控台來完成轉換。若要使用 AWS CLI 或 RDS API，請遵循中的指示 [將資料庫執行個體修改為多可用區域資料庫執行個體部署](#)。

使用 RDS 主控台將其轉換為多可用區域資料庫執行個體部署

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 從 Actions (動作) 中，選擇 Convert to Multi-AZ deployment (轉換為多可用區部署)。
4. 在確認頁面上，選擇 Apply immediately (立即套用)，來立即套用變更。選擇此選項並不會產生停機時間，但可能會對效能產生影響。或者，您也可以選擇在下個維護時段套用更新。如需詳細資訊，請參閱 [排程修改設定](#)。
5. 選擇 Convert to Multi-AZ (轉換為多可用區域)。

將資料庫執行個體修改為多可用區域資料庫執行個體部署

您可以採取下列方法，將資料庫執行個體修改為多可用區域資料庫執行個體部署：

- 使用 RDS 主控台，修改資料庫執行個體，並將 Multi-AZ deployment (多可用區部署) 設為 Yes (是)。
- 使用 AWS CLI，呼叫指 [modify-db-instance](#) 令，然後設定 `--multi-az` 選項。
- 使用 RDS API，呼叫 [ModifyDBInstance](#) 操作，並將 MultiAZ 參數設為 true。

如需修改資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。修改完成後，Amazon RDS 會觸發一個表示該程序已完成的事件 (RDS-EVENT-0025)。您可以監控 Amazon RDS 事件。如需事件的詳細資訊，請參閱 [使用 Amazon RDS 事件通知](#)。

Amazon RDS 的容錯移轉程序

如果您已啟用多可用區域，當您的資料庫執行個體因為基礎設施瑕疵而發生計劃中或非計劃的中斷時，Amazon RDS 會自動切換到另一個可用區域中的待命複本。完成容錯移轉所需的時間取決於主要資料庫執行個體失效時的資料庫活動和其他條件。通常容錯移轉時間是 60–120 秒。不過，大型交易或冗長復原程序可能會增加容錯移轉時間。容錯移轉完成後，RDS 控制台使用者界面可能需要多一點時間來反映新的可用區域。

Note

重新啟動資料庫執行個體時，您可以手動強制容錯移轉。如需詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

Amazon RDS 會自動處理容錯移轉，因此您可以盡快恢復資料庫操作，而無須人為介入管理。如果發生下表所述的任何條件，主要資料庫執行個體會自動切換至待命複本。您可以在事件日誌中檢視這些容錯移轉原因。

容錯移轉原因	描述
<p>以 RDS 資料庫執行個體為基礎的作業系統正在離線操作中進行修補。</p>	<p>作業系統修補或安全更新的維護期間觸發容錯移轉。</p> <p>如需詳細資訊，請參閱 維持資料庫執行個體。</p>
<p>RDS 異地同步備份執行個體的主要主機狀況不良。</p>	<p>多可用區域資料庫執行個體部署偵測到主要資料庫執行個體受損並容錯移轉。</p>
<p>由於網路連線中斷，RDS 異地同步備份執行個體的主要主機無法連線。</p>	<p>RDS 監控偵測到主要資料庫執行個體的網路連線失敗，並觸發容錯移轉。</p>
<p>RDS 執行個體已由客戶修改。</p>	<p>RDS 資料庫執行個體修改觸發了容錯移轉。</p> <p>如需詳細資訊，請參閱 修改 Amazon RDS 資料庫執行個體。</p>
<p>RDS 異地同步備份主要執行個體忙碌且沒有回應。</p>	<p>主要資料庫執行個體沒有回應。我們建議您進行下列動作：</p> <ul style="list-style-type: none"> • 檢查事件和記 CloudWatch 錄檔是否有過多的 CPU、記憶體或交換空間使用量。如需詳細資訊，請參閱 使用 Amazon RDS 事件通知 及 建立由 Amazon RDS 事件觸發的規則。 • 評估您的工作負載，以判斷您是否正在使用適當的資料庫執行個體類別。如需詳細資訊，請參閱 資料庫執行個體類別。 • 針對即時作業系統指標，使用增強型監控。如需詳細資訊，請參閱 使用增強型監控來監控作業系統指標。 •

容錯移轉原因	描述
	<p>使用績效詳情協助分析影響資料庫執行個體效能的任何問題。如需詳細資訊，請參閱 在 Amazon RDS 上使用績效詳情監控資料庫負載。</p> <p>如需這些建議的詳細資訊，請參閱 在 Amazon RDS 中監控指標的概觀 和 Amazon RDS 的最佳實務。</p>
<p>以 RDS 異地同步備份執行個體主要主機為基礎的儲存磁碟區發生故障。</p> <p>使用者請求資料庫執行個體的容錯移轉。</p>	<p>多可用區域資料庫執行個體部署在主要資料庫執行個體上偵測到儲存問題並容錯移轉。</p> <p>您重新啟動了資料庫執行個體，並選擇了 Reboot with failover (使用容錯移轉重新啟動)。</p> <p>如需詳細資訊，請參閱 重新啟動中的資料庫執行個體。</p>

若要判斷您的多可用區域資料庫執行個體是否已容錯移轉，您可以執行下列動作：

- 設定資料庫事件訂閱，透過電子郵件或 SMS 通知您已啟動容錯移轉。如需事件的詳細資訊，請參閱 [使用 Amazon RDS 事件通知](#)。
- 您可以使用 RDS 主控台或 API 操作來檢視資料庫事件。
- 您可以使用 RDS 主控台和 API 操作，檢視多可用區域資料庫執行個體部署的目前狀態。

有關如何回應容錯移轉、減少恢復時間以及其他 Amazon RDS 最佳實務，詳細資訊請參閱 [Amazon RDS 的最佳實務](#)。

設定 DNS 名稱查詢的 JVM TTL

容錯移轉機制會自動將資料庫執行個體的網域名稱系統 (DNS) 記錄變更為指向待命資料庫執行個體。因此，您必須重新建立資料庫執行個體任何現有的連線。在 Java 虛擬機器 (JVM) 環境中，基於 Java DNS 快取機制的運作方式，您可能需要重新配置 JVM 設定。

JVM 會快取 DNS 名稱查詢。當 JVM 將主機名稱解析為 IP 位址時，會將 IP 位址快取一段指定的時間段，稱為 time-to-live(TTL)。

由於 AWS 資源使用偶爾會變更的 DNS 名稱項目，因此建議您將 JVM 設定為不超過 60 秒的 TTL 值。如此可確保當資源的 IP 位址變更時，您的應用程式可以透過重新查詢 DNS 來接收並使用資源的新 IP 位址。

在一些 Java 組態上，則會設定 JVM 預設 TTL，以便其在重新啟動 JVM 之前絕對不會重新整理 DNS 項目。因此，如果 AWS 資源的 IP 位址在應用程式仍在執行時發生變更，則在您手動重新啟動 JVM 並重新整理快取 IP 資訊之前，該資源無法使用該資源。在此情況下，設定 JVM 的 TTL 至為關鍵，以便其定期重新整理其快取的 IP 資訊。

您可以擷取 [networkaddress.cache.ttl](#) 屬性值來取得 JVM 預設 TTL：

```
String ttl = java.security.Security.getProperty("networkaddress.cache.ttl");
```

Note

預設 TTL 可能會視 JVM 的版本以及是否已安裝安全管理員而異。許多 JVM 提供的預設 TTL 少於 60 秒。如果您使用此類 JVM (而非安全管理員)，則可忽略本主題的其餘內容。如需 Oracle 中安全管理員的詳細資訊，請參閱 Oracle 文件中的 [安全管理員](#)。

若要修改 JVM 的 TTL，請設定 `networkaddress.cache.ttl` 屬性值。根據您的需求，使用下列其中一種方法：

- 若要為使用 JVM 的所有應用程式全域設定屬性值，請在 `networkaddress.cache.ttl` 檔案中設定 `$JAVA_HOME/jre/lib/security/java.security`。

```
networkaddress.cache.ttl=60
```

- 若要僅針對您的應用程式進行適當的本機設定，請在建立任何網路連線之前，在您應用程式的初始化程式碼中設定 `networkaddress.cache.ttl`。

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```


多可用區域資料庫叢集部署

異地同步備份資料庫叢集部署是具有兩個可讀取複本資料庫執行個體的 Amazon RDS 半同步、高可用性部署模式。多可用區域資料庫叢集具有寫入器資料庫執行個體和兩個讀取器資料庫執行個體，位於相同 AWS 區域中的三個個別可用區域中。多可用區域資料庫叢集相較於多可用區域資料庫執行個體部署，可提供高可用性、增加讀取工作負載的容量，以及降低寫入延遲。

您可以遵循[將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#)中的指示，將資料從內部部署資料庫匯入至多可用區域資料庫叢集。

您可以為多可用區域資料庫叢集購買預留資料庫執行個體。如需詳細資訊，請參閱[多可用區域資料庫叢集的預留資料庫執行個體](#)。

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 Amazon RDS 搭配多個可用區域資料庫叢集的版本和可用性的詳細資訊，請參閱[Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎](#)。

主題

- [異地同步備份資料庫叢集執行個體類別](#)
- [多可用區域資料庫叢集概觀](#)
- [使用管理異地同步備份資料庫叢集 AWS Management Console](#)
- [使用多可用區域資料庫叢集的參數群組](#)
- [升級多可用區域資料庫叢集的引擎版本](#)
- [使用 RDS Proxy 搭配多可用區資料庫叢集](#)
- [複本延遲和多可用區域資料庫叢集](#)
- [多可用區域資料庫叢集的容錯移轉程序](#)
- [建立多可用區域資料庫叢集](#)
- [連線至多可用區域資料庫叢集](#)
- [自動連線 AWS 運算資源和多可用區域資料庫叢集](#)
- [修改多可用區域資料庫叢集](#)
- [重新命名多可用區域資料庫叢集](#)
- [重新啟機多可用區域資料庫叢集和讀取器資料庫執行個體](#)
- [使用多可用區域資料庫叢集僅供讀取複本](#)

- [搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫](#)
- [刪除多可用區域資料庫叢集](#)
- [異地同步備份資料庫叢集限制](#)

Important

多可用區域資料庫叢集與 Aurora 資料庫叢集不同。如需 Aurora 資料庫叢集的詳細資訊，請參閱 [《Amazon Aurora 使用者指南》](#)。

異地同步備份資料庫叢集執行個體類別

下列資料庫執行個體類別支援異地同步備份資料庫叢集部

署：db.m5ddb.m6gddb.m6idddb.m6idndb.r5ddb.r6gd、db.x2iedn、db.r6id、db.r6idn、和db.c6gd。

Note

c6gd 實體類別是唯一支援執行個體大小的實體類別。medium

如需資料庫執行個體類別的詳細資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

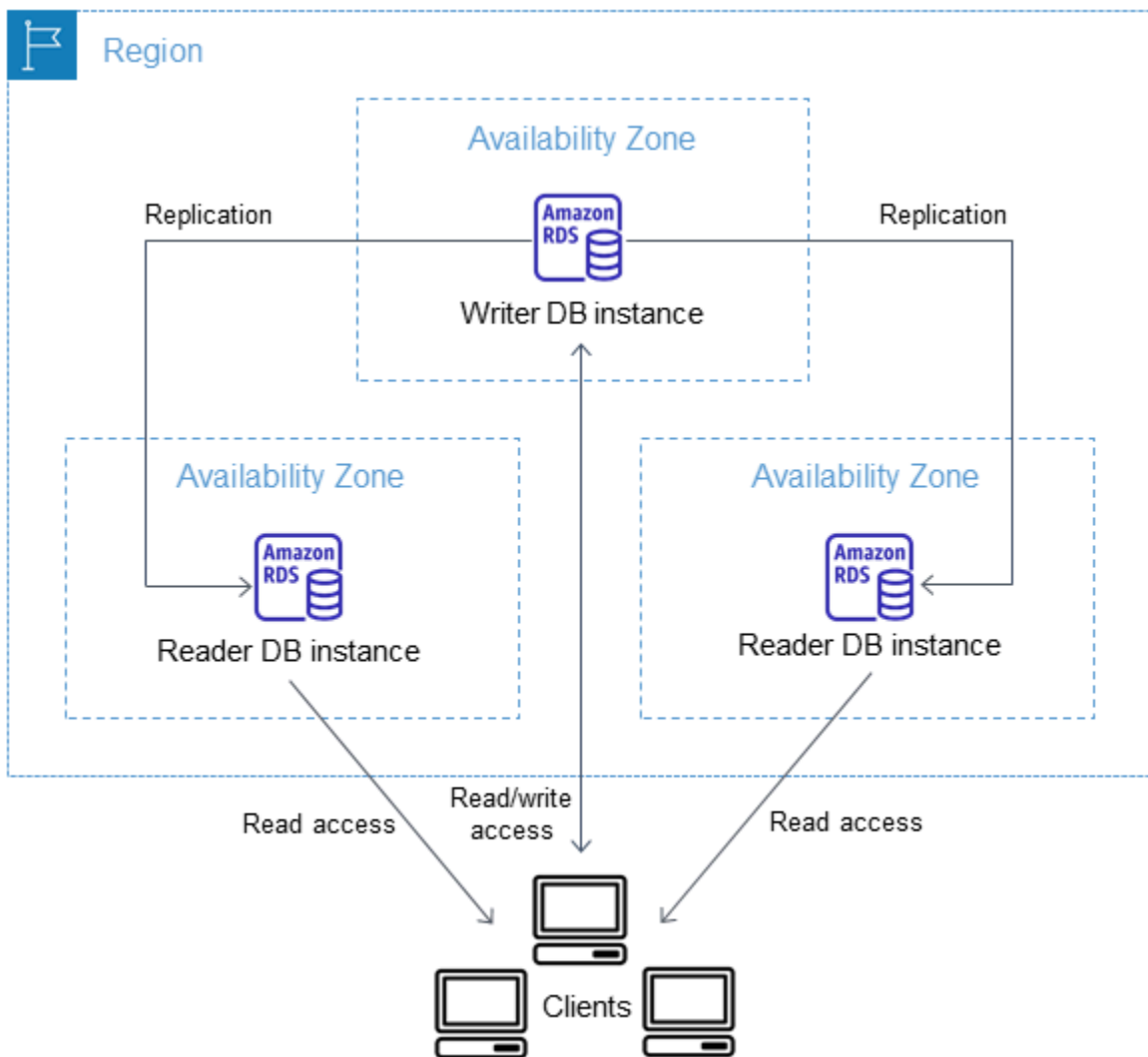
多可用區域資料庫叢集概觀

有了多可用區域資料庫叢集，Amazon RDS 會使用資料庫引擎的原生複寫功能，將寫入器資料庫執行個體的資料複寫到兩個讀取器資料庫執行個體。對寫入器資料庫執行個體進行變更時，會將它傳送至每個讀取器資料庫執行個體。

多可用區域資料庫叢集部署使用半同步複寫，需要至少一個讀取器資料庫執行個體的確認才能遞交變更。它不需要確認事件已在所有複本上完全執行並遞交。

讀取器資料庫執行個體做為自動容錯移轉目標，並提供讀取流量，以增加應用程式讀取輸送量。如果您的寫入器資料庫執行個體發生服務中斷，RDS 會管理容錯移轉到其中一個讀取器資料庫執行個體。RDS 會根據哪個讀取器資料庫執行個體有最近的變更記錄來執行此作業。

下圖顯示多可用區域資料庫叢集。



與多可用區域資料庫執行個體部署相比，多可用區域資料庫叢集通常具有較低的寫入延遲。此外也允許在讀取器資料庫執行個體上執行唯讀工作負載。RDS 主控台會顯示寫入器資料庫執行個體的可用區域和讀取器資料庫執行個體的可用區域。您也可以使用 [describe-db-clusters](#) CLI 命令或 [DescribeDBClusters](#) API 操作來尋找此資訊。

⚠ Important

為了防止 RDS for MySQL 多可用區域資料庫叢集發生複寫錯誤，強烈建議所有資料表都具有主索引鍵。

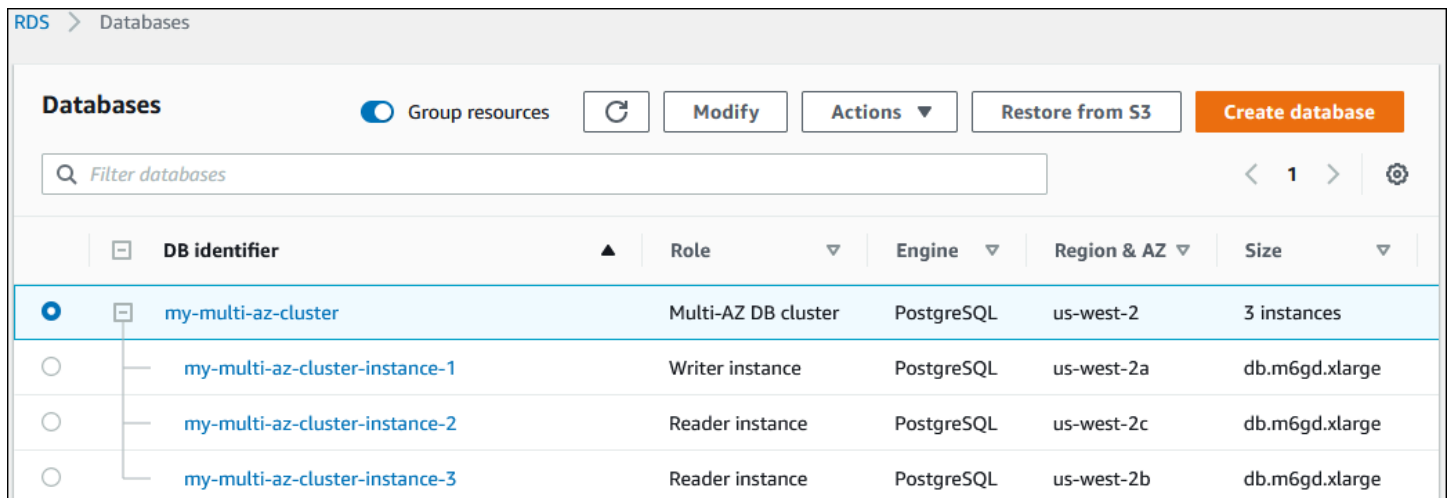
使用管理異地同步備份資料庫叢集 AWS Management Console

您可以使用主控台來管理多可用區域資料庫叢集。

若要使用主控台管理多可用區域資料庫叢集

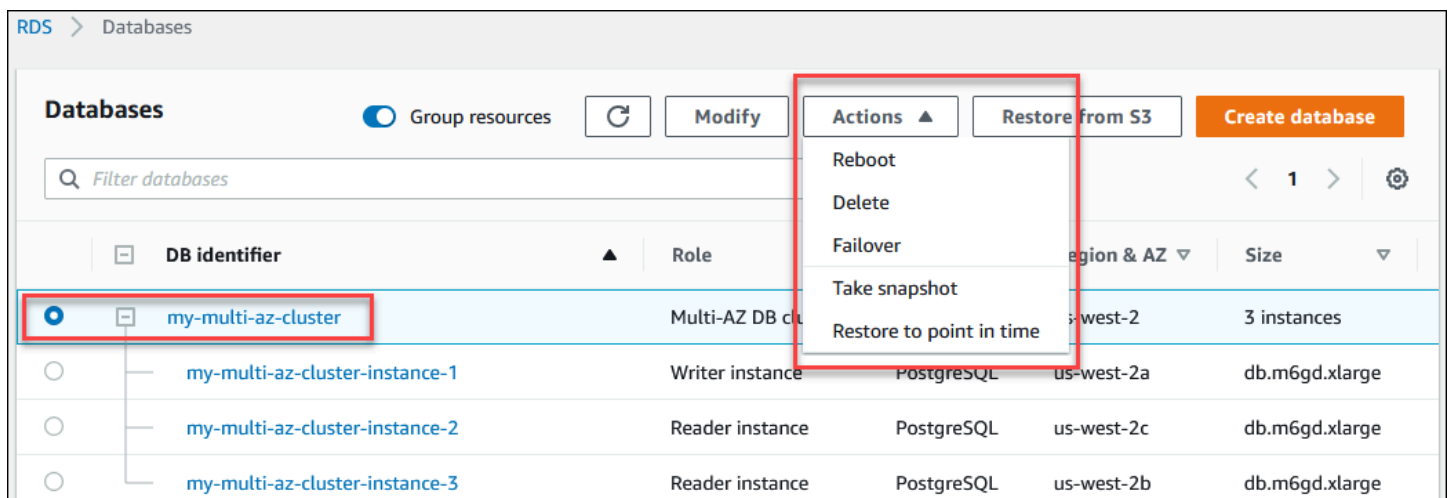
1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要管理的多可用區域資料庫叢集。

下圖顯示主控台中的多可用區域資料庫叢集。

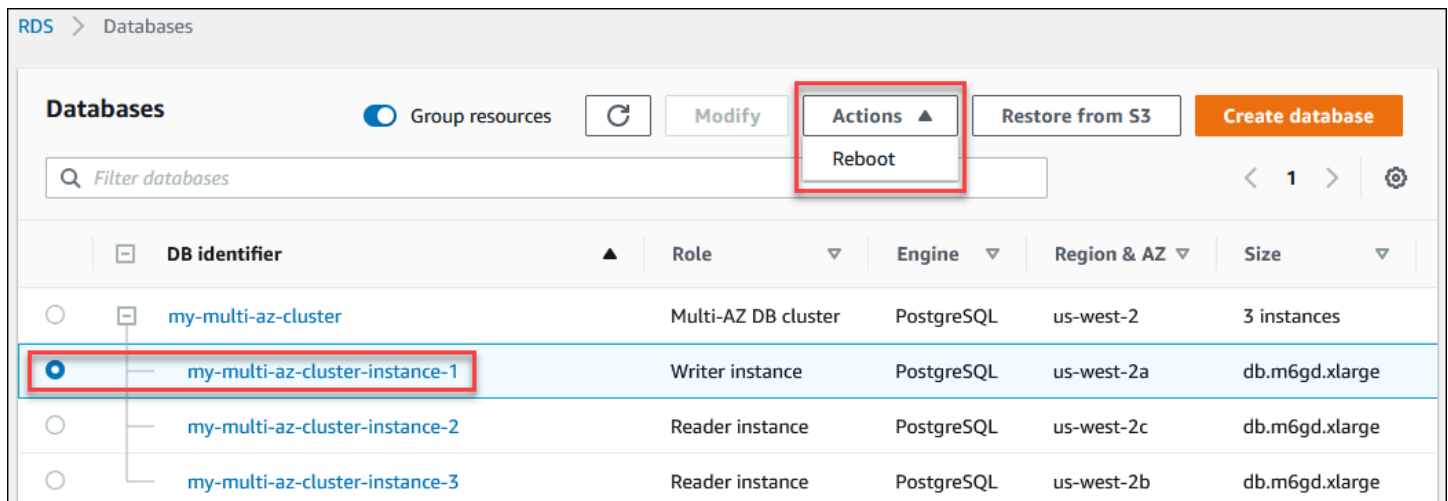


Actions (動作) (動作) 功能表中可用的動作取決於已選取多可用區域資料庫叢集或已選取叢集中的資料庫執行個體。

選擇多可用區域資料庫叢集以檢視叢集詳細資料，並在叢集層級執行動作。



在多可用區域資料庫叢集中選擇資料庫執行個體，以檢視資料庫執行個體詳細資料並在資料庫執行個體層級執行動作。



使用多可用區域資料庫叢集的參數群組

在多可用區域資料庫叢集中，資料庫叢集參數群組做為引擎組態值的容器，這些值會套用到多可用區域資料庫叢集的每個資料庫執行個體。

在多可用區域資料庫叢集中，DB parameter group (資料庫參數群組) 會設定為資料庫引擎和資料庫引擎版本的預設資料庫參數群組。資料庫叢集參數群組中的設定用於叢集中的所有資料庫執行個體。

如需參數群組的詳細資訊，請參閱 [the section called “使用資料庫叢集參數群組”](#)。

升級多可用區域資料庫叢集的引擎版本

Amazon RDS 提供每個受支援的資料庫引擎的更新版本，讓您的異地同步備份資料庫叢集保持最新狀態。Amazon RDS 支援新版的資料庫引擎時，您可以選擇如何和何時升級多可用區域資料庫叢集。

您可以執行兩種升級：

主要版本升級

主要引擎版本升級可能會導致與現有應用程式不相容的變更。當您啟動主要版本升級時，Amazon RDS 會同時升級讀取器和寫入器執行個體。因此，在升級完成之前，您的資料庫叢集可能無法使用。

次要版本升級

次要版本升級只包含與現有應用程式回溯相容的變更。當您啟動次要版本升級時，Amazon RDS 會先一次升級一個讀取器資料庫執行個體。然後，其中一個讀取器資料庫執行個體會切換為新的寫入器資料庫執行個體。然後，Amazon RDS 會升級舊的寫入器執行個體 (現在是讀取器執行個體)。

升級期間的停機時間僅限於其中一個讀取器資料庫執行個體成為新寫入器資料庫執行個體所需的時間。此停機時間就像是自動容錯移轉。如需詳細資訊，請參閱 [the section called “多可用區域資料庫叢集的容錯移轉程序”](#)。請注意，異地同步備份資料庫叢集的複本延遲可能會影響停機時間。如需詳細資訊，請參閱 [the section called “複本延遲和多可用區域資料庫叢集”](#)。

對於 RDS 異地 PostgreSQL 資料庫叢集僅供讀取複本，Amazon RDS 會一次升級一個叢集成員執行個體。讀取器和寫入器叢集角色在升級期間不會切換。因此，Amazon RDS 升級叢集寫入器執行個體時，資料庫叢集可能會遇到停機時間。

Note

異地同步備份資料庫叢集次要版本升級的停機時間通常為 35 秒。與 RDS Proxy 搭配使用時，您可以進一步將停機時間減少到一秒或更短的時間。如需詳細資訊，請參閱 [使用 RDS Proxy](#)。或者，您可以使用開放原始碼資料庫代理伺服器 (例如 [ProxSQL](#)) 或 MySQL 的 [AWS JDBC 驅動程式](#)。 [PgBouncer](#)

目前，Amazon RDS 僅支援 RDS 異地同步備份資料庫叢集的主要版本升級。Amazon RDS 支援所有支援異地同步備份資料庫叢集的資料庫引擎進行次要版本升級。

Amazon RDS 不會自動升級多可用區域資料庫叢集讀取複本。對於次要版本升級，您必須先手動升級所有僅供讀取複本，然後升級叢集。否則，升級會遭到封鎖。當您執行叢集的主要版本升級時，所有讀取複本的複寫狀態都會變更為已終止。升級完成後，您必須刪除並重新建立讀取複本。如需詳細資訊，請參閱 [the section called “監控僅供讀取複本”](#)。

多可用區域資料庫叢集的引擎版本升級程序，與資料庫執行個體引擎版本的升級程序相同。如需說明，請參閱 [the section called “升級引擎版本”](#)。唯一的不同之處在於，使用 AWS Command Line Interface (AWS CLI) 時，您可以使用 [修改-db-cluster](#) 命令並指定 `--db-cluster-identifier` 參數 (連同參數)。 `--allow-major-version-upgrade`

如需有關主要和次要版本升級的詳細資訊，請參閱下列資料庫引擎的文件：

- [the section called “升級 PostgreSQL 資料庫引擎”](#)
- [the section called “升級 MySQL 資料庫引擎”](#)

使用 RDS Proxy 搭配多可用區資料庫叢集

您可以使用 Amazon RDS 代理為異地同步備份資料庫叢集建立代理。透過使用 RDS Proxy，您的應用程式可以集區和共用資料庫連線，以改善其擴充能力。每個代理執行連接多工，也稱為連接重複使

用。RDS Proxy 運用「多工」，使用一個基礎資料庫連線執行一筆交易的所有操作。RDS Proxy 也可以將異地同步備份資料庫叢集次要版本升級的停機時間縮短至一秒或更少。如需有關 RDS Proxy 優勢的詳細資訊，請參閱 [使用 RDS Proxy](#)。

若要為多可用區資料庫叢集設定 Proxy，請在建立叢集時選擇建立 RDS Proxy。如需建立和管理 RDS Proxy 端點的指示，請參閱 [the section called “使用 RDS Proxy 端點”](#)。

複本延遲和多可用區域資料庫叢集

複本延遲指的是寫入器資料庫執行個體上的最新交易，與多可用區域資料庫叢集中讀取器資料庫執行個體上最新套用交易之間的時間差異。Amazon 指 CloudWatch 標 `ReplicaLag` 代表此時差。如需 CloudWatch 測量結果的詳細資訊，請參閱 [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。

雖然多可用區域資料庫叢集可達到高寫入效能，但由於以引擎為基礎的的複寫性質，複本延遲仍可能出現。任何容錯移轉都必須先解決複本延遲情況，然後才能提升新的寫入器資料庫執行個體，因此監控和管理複本延遲是需考慮的因素。

對於 RDS for MySQL 多可用區域資料庫叢集，容錯移轉時間取決於其餘兩個讀取器資料庫執行個體的複本延遲。兩個讀取器資料庫執行個體必須套用未套用的交易，然後才能將其中一個提升到新的寫入器資料庫執行個體。

對於 RDS for PostgreSQL 多可用區域資料庫叢集，容錯移轉時間取決於其餘兩個讀取器資料庫執行個體的最低複本延遲。具有最低複本延遲的讀取器資料庫執行個體必須套用未套用的交易，然後才能將其提升到新的寫入器資料庫執行個體。

如需說明如何在複本延遲超過設定時間量時建立 CloudWatch 警示的教學課程，請參閱 [教學課程：為多可用區域資料庫叢集複本延遲建立 Amazon CloudWatch 警示](#)。

複本延遲的常見原因

通常，如果寫入工作負載過高，讀取器資料庫執行個體就無法有效套用交易，進而發生複本延遲。多種工作負載可能會發生暫時或連續的複本延遲。以下是一些常見原因的例子：

- 寫入器資料庫執行個體上進行高寫入並行或大批次更新，導致讀取器資料庫執行個體上的套用程序落後。
- 繁重的讀取工作負載使用一或多個讀取器資料庫執行個體上的資源。執行緩慢或大型查詢可能會影響套用程序，且可能導致複本延遲。
- 修改大量資料或 DDL 陳述式的交易有時會導致複本延遲暫時增加，因為資料庫必須保留遞交順序。

緩解複本延遲

針對 RDS for MySQL 和 RDS for PostgreSQL 的多可用區域資料庫叢集，可以藉由減少寫入器資料庫執行個體上的負載來緩解複本延遲情況。您還可以使用流量控制來減少複本延遲。流量控制的運作機制是調節寫入器資料庫執行個體上的寫入作業，確保複本延遲不會繼續無限制增長。寫入調節的運作機制是在交易結尾新增延遲，藉此降低寫入器資料庫執行個體的寫入輸送量。雖然流量控制不能保證消除延遲，但可以幫助減少許多工作負載的整體延遲。下列各節提供了有關將流量控制與 RDS for MySQL 和 RDS for PostgreSQL 搭配使用的資訊。

利用流量控制緩解 RDS for MySQL 的複本延遲

當您使用 RDS for MySQL 多可用區域資料庫叢集時，預設會使用動態參數

`rpl_semi_sync_master_target_apply_lag` 開啟流量控制。此參數指定您要用於複本延遲的上限。當複本延遲接近此設定的限制時，流量控制會限制寫入器資料庫執行個體上的寫入交易，以嘗試將複本延遲限制在指定值以下。在某些情況下，複本延遲可能會超過指定限制。根據預設，此參數設定為 120 秒。若要關閉流量控制，請將此參數設定為 86,400 秒 (一天) 的最大值。

若要查看流量控制注入的目前延遲，請執行下列查詢以顯示參數

`Rpl_semi_sync_master_flow_control_current_delay`。

```
SHOW GLOBAL STATUS like '%flow_control%';
```

您的輸出應該類似以下內容：

```
+-----+-----+
| Variable_name                | Value |
+-----+-----+
| Rpl_semi_sync_master_flow_control_current_delay | 2010  |
+-----+-----+
1 row in set (0.00 sec)
```

Note

延遲以微秒為單位顯示。

為 RDS for MySQL 多可用區域資料庫叢集開啟 Performance Insights 後，您可以監控與指示查詢由流量控制延遲的 SQL 陳述式對應的等待事件。流量控制引入延遲時，您可以查看與 Performance Insights 儀表板上的 SQL 陳述式對應的等待事件 `/wait/synch/cond/semisync/`

`semi_sync_flow_control_delay_cond`。若要查看這些指標，請確認已開啟 Performance Schema。如需 Performance Insights 的詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。

利用流量控制緩解 RDS for PostgreSQL 的複本延遲

在使用 RDS for PostgreSQL 多可用區域資料庫叢集時，流量控制功能會部署為擴充功能。它會為資料庫叢集中的所有資料庫執行個體啟動背景工作者。預設情況下，讀取器資料庫執行個體上的背景工作者，會向寫入器資料庫執行個體上的背景工作者傳達目前的複本延遲情況。如果任何讀取器資料庫執行個體上的延遲超過兩分鐘，寫入器資料庫執行個體上的背景工作者就會在交易結尾新增延遲。若要控制延遲閾值，請使用參數 `flow_control.target_standby_apply_lag`。

流量控制調節 PostgreSQL 程序時，Extension 會等待 `pg_stat_activity` 中的事件，而績效詳情會指出此情況。函數 `get_flow_control_stats` 顯示有關目前正在新增多少延遲的詳細資訊。

流量控制有助於大多數線上交易處理 (OLTP) 工作負載，這些工作負載具有短暫但高度並行的交易。如果延遲是由長時間執行的交易 (如批次操作) 引起的，流量控制提供的效益就沒有這麼大。

若要關閉流量控制功能，請從 `shared_preload_libraries` 移除擴充功能並重新啟動資料庫執行個體。

多可用區域資料庫叢集的容錯移轉程序

如果您的多可用區域資料庫叢集中的寫入器資料庫執行個體發生計劃內或計劃外中斷，Amazon RDS 會自動容錯移轉到不同可用區域中的讀取器資料庫執行個體。完成容錯移轉所需的時間取決於寫入器資料庫執行個體失效時的資料庫活動和其他條件。容錯移轉時間通常低於 35 秒。當兩個讀取器資料庫執行個體都套用來自失敗寫入器的未完成交易時，容錯移轉就會完成。容錯移轉完成後，RDS 控制台使用者界面可能需要多一點時間來反映新的可用區域。

主題

- [自動容錯移轉](#)
- [手動容錯移多可用區域資料庫叢集](#)
- [判斷多可用區域資料庫叢集是否已容錯移轉](#)
- [設定 DNS 名稱查詢的 JVM TTL](#)

自動容錯移轉

Amazon RDS 會自動處理容錯移轉，因此您可以盡快恢復資料庫操作，而無須人為介入管理。若要容錯移轉，寫入器資料庫執行個體會自動切換至讀取器資料庫執行個體。

手動容錯移多可用區域資料庫叢集

如果您手動容錯移轉異地同步備份資料庫叢集，RDS 會先終止主要資料庫執行個體。然後，內部監控系統會偵測到主要資料庫執行個體運作狀況不佳，並提升可讀取的複本資料庫執行個體。容錯移轉時間通常低於 35 秒。

您可以使用 AWS Management Console、或 RDS API 手動容錯移轉異地同步備份資料庫叢集。AWS CLI

主控台

若要手動容錯移轉多可用區域資料庫叢集

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要容錯移轉的多可用區域資料庫叢集。
4. 針對 Actions (動作)，選擇 Failover (容錯移轉)。

便會顯示容錯移轉資料庫叢集頁面

5. 選擇 Failover (容錯移轉) 以確認手動容錯移轉。

AWS CLI

若要手動容錯移轉異地同步備份資料庫叢集，請使用 AWS CLI 指令 [容錯移轉](#) db-cluster。

Example

```
aws rds failover-db-cluster --db-cluster-identifier mymultiadbcluster
```

RDS API

若要手動容錯移轉多可用區域資料庫叢集，請呼叫 Amazon RDS API [FailoverDBCluster](#) 並指定 DBClusterIdentifier。

判斷多可用區域資料庫叢集是否已容錯移轉

若要判斷您的多可用區域資料庫叢集是否已容錯移轉，您可以執行下列動作：

- 設定資料庫事件訂閱，透過電子郵件或 SMS 通知您已啟動容錯移轉。如需事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。
- 使用 Amazon RDS 主控台或 API 操作來檢視資料庫事件。
- 使用 Amazon RDS 主控台和 RDS API 檢視異地同步備份資料庫叢集的目前狀態。AWS CLI

有關如何回應容錯移轉、減少恢復時間以及其他 Amazon RDS 最佳實務，詳細資訊請參閱[Amazon RDS 的最佳實務](#)。

設定 DNS 名稱查詢的 JVM TTL

容錯移轉機制會自動將資料庫執行個體的網域名稱系統 (DNS) 記錄變更為指向讀取器資料庫執行個體。因此，您必須重新建立資料庫執行個體任何現有的連線。在 Java 虛擬機器 (JVM) 環境中，基於 Java DNS 快取機制的運作方式，您可能需要重新配置 JVM 設定。

JVM 會快取 DNS 名稱查詢。當 JVM 將主機名稱解析為 IP 地址時，它會在指定的時間段內快取 IP 地址，稱為存留時間 (TTL)。

由於 AWS 資源使用偶爾會變更的 DNS 名稱項目，因此建議您將 JVM 設定為不超過 60 秒的 TTL 值。如此可確保當資源的 IP 位址變更時，您的應用程式可以透過重新查詢 DNS 來接收並使用資源的新 IP 位址。

在一些 Java 組態上，則會設定 JVM 預設 TTL，以便其在重新啟動 JVM 之前絕對不會重新整理 DNS 項目。因此，如果 AWS 資源的 IP 位址在應用程式仍在執行時發生變更，則在您手動重新啟動 JVM 並重新整理快取 IP 資訊之前，該資源無法使用該資源。在此情況下，設定 JVM 的 TTL 至為關鍵，以便其定期重新整理其快取的 IP 資訊。

Note

預設 TTL 可能會視 JVM 的版本以及是否已安裝安全管理員而異。許多 JVM 提供的預設 TTL 少於 60 秒。如果您使用此類 JVM (而非安全管理員)，則可忽略本主題的其餘內容。如需 Oracle 中安全管理員的詳細資訊，請參閱 Oracle 文件中的[安全管理員](#)。

若要修改 JVM 的 TTL，請設定 [networkaddress.cache.ttl](#) 屬性值。根據您的需求，使用下列其中一種方法：

- 若要為使用 JVM 的所有應用程式全域設定屬性值，請在 `networkaddress.cache.ttl` 檔案中設定 `$JAVA_HOME/jre/lib/security/java.security`。

```
networkaddress.cache.ttl=60
```

- 若要僅針對您的應用程式進行適當的本機設定，請在建立任何網路連線之前，在您應用程式的初始化程式碼中設定 `networkaddress.cache.ttl`。

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

建立多可用區域資料庫叢集

多可用區域資料庫叢集在三個不同的可用區域中有一個寫入器資料庫執行個體和兩個讀取器資料庫。多可用區域資料庫叢集相較於多可用區域部署，可提供高可用性、增加讀取工作負載的容量以及更低的延遲。如需建立多可用區域資料庫叢集的詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。

Note

多可用區域資料庫叢集僅支援 MySQL 和 PostgreSQL 資料庫引擎。

資料庫叢集先決條件

Important

您必須先完成 [設定 Amazon RDS](#) 中的任務，然後才能建立多可用區域資料庫叢集。

以下是建立多可用區域資料庫叢集須完成的先決條件。

主題

- [設定資料庫叢集的網路](#)
- [其他先決條件](#)

設定資料庫叢集的網路

您只能在以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中建立多可用區資料庫叢集。它必須位於至少具 AWS 區域有三個可用區域的中。您為資料庫叢集選擇的資料庫子網路群組必須至少包含三個可用區域。此設定可確保資料庫叢集中的每個資料庫執行個體都位於不同的可用區域。

若要在同一 VPC 中設定新資料庫叢集與 Amazon EC2 執行個體之間的連線，則可在建立資料庫叢集期間這麼做。若要在同一 VPC 中從 EC2 執行個體以外的資源連線到資料庫叢集，請手動設定網路連線。

主題

- [設定與 EC2 執行個體的自動網路連線](#)
- [手動設定網路](#)

設定與 EC2 執行個體的自動網路連線

建立異地同步備份資料庫叢集時，可以使用設 AWS Management Console 定 EC2 執行個體和新資料庫叢集之間的連線。當您這樣做時，RDS 會自動設定您的 VPC 和網路設定。資料庫叢集會在與 EC2 執行個體相同的 VPC 中建立，讓 EC2 執行個體可以存取資料庫叢集。

以下是將 EC2 執行個體與資料庫叢集連接的要求：

- EC2 執行個體必須存在於中，AWS 區域 才能建立資料庫叢集。

如果中沒有 EC2 執行個體 AWS 區域，則主控台會提供建立 EC2 執行個體的連結。

- 建立資料庫執行個體的使用者必須擁有執行下列操作的許可：

- `ec2:AssociateRouteTable`
- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateRouteTable`
- `ec2:CreateSubnet`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

使用此選項建立私有資料庫叢集。資料庫叢集使用僅包含私有子網路的資料庫子網路群組，以限制對 VPC 內資源的存取。

若要將 EC2 執行個體連接至資料庫叢集，請在 Create database (建立資料庫) 頁面上的 Connectivity (連線) 區段，選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源)。

Connectivity [Info](#)
↻

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 Instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

Choose EC2 instances
▼

當您選擇 **Connect to an EC2 compute resource (連線至 EC2 運算資源)** 時，RDS 會自動設定下列選項。除非您透過選擇 **Don't connect to an EC2 compute resource (不要連線至 EC2 運算資源)**，選擇不設定與 EC2 執行個體的連線，否則無法變更設定。

主控台選項	自動設定
Virtual Private Cloud (VPC)	RDS 會將 VPC 設定為與 EC2 執行個體關聯的 VPC。
DB subnet group (資料庫子網路群組)	<p>RDS 需要在相同可用區域中具有私有子網路的資料庫子網路群組作為 EC2 執行個體。如果存在符合此要求的資料庫子網路群組，則 RDS 會使用現有的資料庫子網路群組。依預設，此選項會設為 Automatic setup (自動設定)。</p> <p>當您選擇 Automatic setup (自動設定)，且沒有符合此需求的資料庫子網路群組時，則會發生下列動作。RDS 在三個可用區域中使用三個可用的私有子網路，其中一個可用區域與 EC2 執行個體相同。如果可用區域中無法使用私有子網路，RDS 會在可用區域中建立私有子網路。RDS 接著建立資料庫子網路群組。</p> <p>當私有子網路可用時，RDS 會使用與該子網路相關聯的路由表，並將其建立的任何子網路新增至此路由表。當沒有可用的私有子</p>

主控台選項	自動設定
	<p>網路時，RDS 會建立沒有網際網路閘道存取權的路由表，並將其建立的子網路新增至路由表。</p> <p>RDS 也可讓您使用現有的資料庫子網路群組。如果想要使用您選擇的現有資料庫子網路群組，請選取 Choose existing (選擇現有的)。</p>
公用存取	<p>RDS 選擇 No (否)，以便無法公開存取資料庫叢集。</p> <p>為了安全起見，最佳實務是保持資料庫為私有，並確保無法從網際網路存取該資料庫。</p>
VPC security group (firewall) (VPC 安全群組 (防火牆))	<p>RDS 會建立與資料庫叢集關聯的新安全群組。安全群組已命名為 rds-ec2-<i>n</i>，其中 <i>n</i> 是數字。此安全群組包含以 EC2 VPC 安全群組 (防火牆) 做為來源的傳入規則。與資料庫叢集關聯的此一安全群組可讓 EC2 執行個體存取資料庫叢集。</p> <p>RDS 也會建立與 EC2 執行個體關聯的新安全群組。安全群組命名為 ec2-rds-<i>n</i>，其中 <i>n</i> 是數字。此安全群組包含傳出規則，並將資料庫叢集的 VPC 安全群組做為來源。此安全群組允許 EC2 執行個體將流量傳送到資料庫叢集。</p> <p>您可以選擇 Create New (建立新的)，並輸入新安全群組的名稱，以新增其他的新安全群組。</p> <p>您可以選擇 Choose existing (選擇現有)，然後選取要新增的安全群組，以新增現有的安全群組。</p>
可用區域	<p>RDS 針對異地同步備份資料庫叢集部署中的一個資料庫執行個體選擇 EC2 執行個體的可用區域。RDS 會為其他兩個資料庫執行個體隨機選擇不同的可用區域。寫入器資料庫執行個體建立在與 EC2 執行個體相同的可用區域中。如果發生容錯移轉，且寫入器資料庫執行個體位於不同的可用區域，則可能會產生跨可用區域成本。</p>

如需這些設定的詳細資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。

如果您在建立資料庫叢集之後變更這些設定，這些變更可能會影響 EC2 執行個體和資料庫叢集之間的連線。

手動設定網路

若要在同一 VPC 中從 EC2 執行個體以外的資源連線到資料庫叢集，請手動設定網路連線。如果您使用建立異地同步備份資料庫叢集，您可以 AWS Management Console 讓 Amazon RDS 自動為您建立 VPC。或者，您可以使用現有的 VPC 或為您的多可用區域資料庫叢集建立新的 VPC。在至少三個可用區域的每一個區域中，您的 VPC 必須有至少一個子網路，您才能與多可用區域資料庫叢集搭配使用。如需 VPC 的相關資訊，請參閱 [Amazon VPC](#) 和 [Amazon RDS](#)。

如果您沒有預設 VPC 或尚未建立 VPC，而且您不打算使用主控台，請執行下列動作：

- 在您要部署資料庫叢集的區域中至少三個可用區域中，每個子網路中 AWS 至少有一個子網路建立 VPC。如需詳細資訊，請參閱 [在 VPC 中使用資料庫執行個體](#)。
- 指定可授權連線到資料庫叢集的 VPC 安全群組。如需詳細資訊，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#) 及 [使用安全群組控制存取](#)。
- 指定 RDS 資料庫子網路群組，該子網路群組會在可由多可用區域資料庫叢集使用的 VPC 中，至少定義三個子網路。如需詳細資訊，請參閱 [使用資料庫子網路群組](#)。

如需有關套用至多可用區域資料庫叢集的限制資訊，請參閱 [異地同步備份資料庫叢集限制](#)。

如果您想要連線至與多可用區域資料庫叢集不在相同 VPC 中的資源，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 中的適當案例。

其他先決條件

您必須考量下列其他先決條件，才能建立多可用區域資料庫叢集：

- 若要連線到 AWS 使用 AWS Identity and Access Management (IAM) 登入資料，您的 AWS 帳戶必須具有特定的 IAM 政策。這些會授與執行 Amazon RDS 操作所需的許可。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。

如果您使用 IAM 存取 RDS 主控台，請先使用 IAM 使用 AWS Management Console 者登入資料登入。接著前往 RDS 主控台：<https://console.aws.amazon.com/rds/>。

- 若要為資料庫叢集量身打造組態參數，則必須使用所需的參數設定來指定資料庫叢集參數群組。如需建立或修改資料庫叢集參數群組的詳細資訊，請參閱 [使用多可用區域資料庫叢集的參數群組](#)。

- 決定要為資料庫叢集指定的 TCP/IP 連接埠號碼。某些公司的防火牆會封鎖與這些預設連接埠的連線。如果您的公司防火牆會封鎖預設連接埠，請為您的資料庫叢集選擇另一個連接埠。資料庫叢集中的所有資料庫執行個體都使用相同的連接埠。
- 如果資料庫的主要引擎版本到達 RDS 標準 Support 結束日期，您必須使用延伸支援 CLI 選項或 RDS API 參數。如需詳細資訊，請參閱[建立多可用區域資料庫叢集的設定](#)。

建立資料庫叢集

您可以使用 AWS Management Console、或 RDS API 建立異地同步備份資料庫叢集。AWS CLI

主控台

您可以藉由選擇 Availability and durability (可用性與持久性) 區段中的 Multi-AZ DB cluster (多可用區域資料庫叢集)，建立多可用區域資料庫叢集。

使用主控台建立多可用區域資料庫叢集

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在的右上角 AWS Management Console，選擇您要 AWS 區域 在其中建立資料庫叢集的。

如需支援異地同步備份 AWS 區域 資料庫叢集的相關資訊，請參閱[異地同步備份資料庫叢集限制](#)。

3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。

若要建立多可用區域資料庫叢集，請確定已選取 Standard Create (標準建立) 且未選取 Easy Create (輕鬆建立)。

5. 在 Engine type (引擎類型) 中，選擇 MySQL 或 PostgreSQL。
6. 在 Version (版本) 中，選擇資料庫引擎版本。

如需有關支援多可用區域資料庫叢集的資料庫引擎叢集的詳細資訊，請參閱[異地同步備份資料庫叢集限制](#)。

7. 在 Templates (範本) 中，選擇您的部署的適當範本。
8. 在 Availability and durability (可用性與持久性) 中，選擇 Multi-AZ DB cluster (多可用區域資料庫叢集)。

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB cluster**
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance**
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance**
Creates a single DB instance with no standby DB instances.

9. 在 DB cluster identifier (資料庫叢集識別符) 中，輸入您的資料庫叢集的識別符。
10. 在 Master username (主要使用者名稱) 中，輸入主要使用者名稱，或保留預設設定。
11. 輸入您的主要密碼：
 - a. 在 Settings (設定) 區段，開啟 Credential Settings (登入資料設定)。
 - b. 如果您想要指定密碼，請清除 Auto generate a password (自動產生密碼) 方塊 (如果已選取)。
 - c. (選用) 變更 Master username (主要使用者名稱) 值。
 - d. 在 Master password (主要密碼) 和 Confirm password (確認密碼) 中輸入相同的密碼。
12. 在資料庫執行個體類別中，選擇資料庫執行個體類別。如需受支援資料庫執行個體類別的清單，請參閱[the section called “異地同步備份資料庫叢集執行個體類別”](#)。
13. (選用) 為此資料庫叢集設定與運算資源的連線。

您可以在建立資料庫叢集期間設定 Amazon EC2 執行個體和新資料庫叢集之間的連線。如需詳細資訊，請參閱 [設定與 EC2 執行個體的自動網路連線](#)。

14. 在 VPC 安全群組 (防火牆) 下的連線能力區段中，如果您選取建立新的，則系統會建立 VPC 安全群組並提供傳入規則，允許本機電腦的 IP 地址存取資料庫。
15. 在其餘區段，指定資料庫叢集設定。如需每項設定的相關資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。
16. 選擇 Create database (建立資料庫)。

如果您選擇使用自動產生的密碼，View credential details (檢視登入資料詳細資訊) 按鈕會出現在 Databases (資料庫) 頁面。

若要檢視資料庫叢集的主要使用者名稱和密碼，請選擇 View credential details (檢視登入資料詳細資訊)。

如要以主要使用者的身分連接至資料庫叢集，請使用出現的使用者名稱和密碼。

⚠ Important

您無法再次檢視主要使用者密碼。

17. 在 Databases (資料庫) 中，選擇新的資料庫叢集的名稱。

在 RDS 主控台上，新資料庫叢集的詳細資訊即會出現。在資料庫叢集建立完成且可供使用之前，資料庫叢集會處於 Creating (建立中) 狀態。狀態變更為 Available (可用) 時，您便可連線至資料庫叢集。根據資料庫叢集類別和配置儲存，新的資料庫叢集可能需要幾分鐘才能使用。

AWS CLI

使用建立異地同步備份資料庫叢集之前 AWS CLI，請確定符合所需的必要條件。其中包括建立 VPC 和 RDS 資料庫子網路群組。如需詳細資訊，請參閱 [資料庫叢集先決條件](#)。

若要使用建立異地同步備份資料庫叢集 AWS CLI，請呼叫 `aws rds create-db-cluster` 命令。請指定 `--db-cluster-identifier`。對於 `--engine` 選項，指定 `mysql` 或 `postgres`。

如需每個選項的詳細資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。

如需支援異地同步備份 AWS 區域資料庫叢集的、資料庫引擎和資料庫引擎版本的相關資訊，請參閱 [異地同步備份資料庫叢集限制](#)。

`aws rds create-db-cluster` 命令會建立資料庫叢集的寫入器資料庫執行個體和兩個讀取器資料庫執行個體。每個資料庫執行個體位於不同的可用區域。

例如，下列命令會建立新的 MySQL 8.0 多可用區域資料庫叢集，名為 `mysql-multi-az-db-cluster`。

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-db-cluster \
```

```
--db-cluster-identifier mysql-multi-az-db-cluster \  
--engine mysql \  
--engine-version 8.0.32 \  
--master-username admin \  
--manage-master-user-password \  
--port 3306 \  
--backup-retention-period 1 \  
--db-subnet-group-name default \  
--allocated-storage 4000 \  
--storage-type io1 \  
--iops 10000 \  
--db-cluster-instance-class db.m5d.xlarge
```

在 Windows 中：

```
aws rds create-db-cluster ^  
  --db-cluster-identifier mysql-multi-az-db-cluster ^  
  --engine mysql ^  
  --engine-version 8.0.32 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --port 3306 ^  
  --backup-retention-period 1 ^  
  --db-subnet-group-name default ^  
  --allocated-storage 4000 ^  
  --storage-type io1 ^  
  --iops 10000 ^  
  --db-cluster-instance-class db.m5d.xlarge
```

下列命令會建立 PostgreSQL 13.4 多可用區域資料庫叢集，名為 `postgresql-multi-az-db-cluster`。

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-db-cluster \  
  --db-cluster-identifier postgresql-multi-az-db-cluster \  
  --engine postgres \  
  --engine-version 13.4 \  
  --manage-master-user-password \  
  --master-username postgres \  
  --storage-type io1
```

```
--port 5432 \  
--backup-retention-period 1 \  
--db-subnet-group-name default \  
--allocated-storage 4000 \  
--storage-type io1 \  
--iops 10000 \  
--db-cluster-instance-class db.m5d.xlarge
```

在 Windows 中：

```
aws rds create-db-cluster ^  
--db-cluster-identifier postgresql-multi-az-db-cluster ^  
--engine postgres ^  
--engine-version 13.4 ^  
--manage-master-user-password ^  
--master-username postgres ^  
--port 5432 ^  
--backup-retention-period 1 ^  
--db-subnet-group-name default ^  
--allocated-storage 4000 ^  
--storage-type io1 ^  
--iops 10000 ^  
--db-cluster-instance-class db.m5d.xlarge
```

RDS API

在可以使用 RDS API 建立多可用區域資料庫叢集之前，您必須先滿足所需的先決條件，例如建立 VPC 和 RDS 資料庫子網路群組。如需詳細資訊，請參閱 [資料庫叢集先決條件](#)。

若要使用 RDS API 建立多可用區域資料庫叢集，請呼叫 [CreateDBCluster](#) 操作。請指定 `DBClusterIdentifier`。對於 `Engine` 參數，指定 `mysql` 或 `postgresql`。

如需每個選項的詳細資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。

`CreateDBCluster` 操作會建立資料庫叢集的寫入器資料庫執行個體和兩個讀取器資料庫執行個體。每個資料庫執行個體位於不同的可用區域。

建立多可用區域資料庫叢集的設定

如需當您建立多可用區域資料庫叢集時所選擇設定的詳細資訊，請參閱下表。如需有關 AWS CLI 選項的詳細資訊，請參閱 [建立 db 叢集](#)。如需 RDS API 參數的詳細資訊，請參閱 [CreateDBCluster](#)。

主控台設定	設定說明	CLI 選項和 RDS API 參數
配置儲存	針對資料庫叢集中每個資料庫執行個體配置的儲存量 (以 GiB 為單位)。如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。	CLI 選項： --allocated-storage API 參數： AllocatedStorage
Auto minor version upgrade (自動次要版本升級)	Enable auto minor version upgrade (啟用自動次要版本升級)，讓資料庫叢集自動在可用時接收偏好次要資料庫引擎版本升級。Amazon RDS 會在維護時段執行自動次要版本升級。	CLI 選項： --auto-minor-version-upgrade --no-auto-minor-version-upgrade API 參數： AutoMinorVersionUpgrade
Backup retention period (備份保留期間)	您想要資料庫叢集之自動備份保留的天數。對於多可用區域資料庫叢集，此值必須設定為 1 或以上。 如需詳細資訊，請參閱 備份簡介 。	CLI 選項： --backup-retention-period API 參數： BackupRetentionPeriod
備份時段	Amazon RDS 自動備份資料庫叢集的時段。除非要在特定時段備份資料庫，否則請使用預設值 No Preference (無偏好設定)。 如需詳細資訊，請參閱 備份簡介 。	CLI 選項： --preferred-backup-window API 參數： PreferredBackupWindow
憑證授權單位	資料庫叢集所使用之伺服器憑證的憑證授權單位 (CA)。 如需詳細資訊，請參閱。	CLI 選項： --ca-certificate-identifier

主控台設定	設定說明	CLI 選項和 RDS API 參數
		CLI 選項和 RDS API 參數 RDS API 參數： CACertificateIdentifier
Copy tags to snapshots (將標籤複製到快照)	此選項可在您建立快照時將任何資料庫叢集標籤複製到某個資料庫快照。 如需詳細資訊，請參閱 標記 Amazon RDS 資源 。	CLI 選項： -copy-tags-to-snapshot -no-copy-tags-to-snapshot RDS API 參數： CopyTagsToSnapshot
Database authentication (資料庫身分驗證)	針對多可用區域資料庫叢集，僅支援密碼身分驗證。	無，因為密碼身分驗證是預設值。
Database port (資料庫連線埠)	您想要透過其存取資料庫叢集的連接埠。會顯示預設連接埠。 連接埠在建立資料庫叢集之後就無法變更。 某些公司的防火牆會封鎖與這些預設連接埠的連線。如果您的公司防火牆會封鎖預設連接埠，請為您的資料庫叢集輸入另一個連接埠。	CLI 選項： --port RDS API 參數： Port
DB cluster identifier (資料庫叢集識別符)	資料庫叢集的名稱。以您為內部部署伺服器命名的相同方，為您的資料庫叢集命名。您的資料庫叢集識別碼最多可包含 63 個英數字元，而且在所選 AWS 區域中的帳戶必須是唯一的。	CLI 選項： --db-cluster-identifier RDS API 參數： DBClusterIdentifier

主控台設定	設定說明	CLI 選項和 RDS API 參數
DB instance class (資料庫執行個體類別)	<p>多可用區域資料庫叢集中每個資料庫執行個體的運算和記憶體容量，例如 db.m5d.xlarge 。</p> <p>可能的話，選擇足夠大的資料庫執行個體類別，讓一般查詢工作集可以保留在記憶體中。當工作集保留在記憶體中時，系統可以避免寫入至磁碟，因而可改善效能。</p> <p>如需受支援資料庫執行個體類別的清單，請參閱 the section called “異地同步備份資料庫叢集執行個體類別”。</p>	CLI 選項： <pre>--db-cluster-instance-class</pre> RDS API 參數： DBClusterInstanceClass
DB cluster parameter group (資料庫叢集參數群組)	<p>要與資料庫叢集建立關聯的資料庫叢集參數群組。</p> <p>如需詳細資訊，請參閱 使用多可用區域資料庫叢集的參數群組。</p>	CLI 選項： <pre>--db-cluster-parameter-group-name</pre> RDS API 參數： DBClusterParameterGroupName
DB engine version (資料庫引擎版本)	<p>您要使用的資料庫引擎版本。</p>	CLI 選項： <pre>--engine-version</pre> RDS API 參數： EngineVersion

主控台設定	設定說明	CLI 選項和 RDS API 參數
DB cluster parameter group (資料庫叢集參數群組)	<p>要與資料庫叢集關聯的資料庫執行個體參數群組。</p> <p>如需詳細資訊，請參閱 使用多可用區域資料庫叢集的參數群組。</p>	<p>CLI 選項：</p> <pre>--db-cluster-parameter-group-name</pre> <p>RDS API 參數：</p> <p>DBClusterParameterGroupName</p>
DB subnet group (資料庫子網路群組)	<p>您要用於資料庫叢集的資料庫子網路群組。</p> <p>選取 Choose existing (選擇現有的) 來使用現有的資料庫子網路群組。然後，從 Existing DB subnet groups (現有資料庫子網路群組) 下拉式清單中選擇所需的子網路群組。</p> <p>選擇 Automatic setup (自動設定) 以讓 RDS 選取相容的資料庫子網路群組。如果不存在，RDS 會為您的叢集建立新的子網路群組。</p> <p>如需詳細資訊，請參閱 使用資料庫子網路群組。</p>	<p>CLI 選項：</p> <pre>--db-subnet-group-name</pre> <p>RDS API 參數：</p> <p>DBSubnetGroupName</p>
刪除保護	<p>啟用刪除保護可避免您的資料庫叢集遭意外刪除。如果您使用主控台建立生產資料庫叢集，預設會啟用刪除保護功能。</p> <p>如需詳細資訊，請參閱 刪除資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--deletion-protection</pre> <pre>--no-deletion-protection</pre> <p>RDS API 參數：</p> <p>DeletionProtection</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
加密	<p>Enable Encryption (啟用加密) 以開啟此資料庫叢集的靜態加密。</p> <p>多可用區域資料庫叢集的加密預設為開啟。</p> <p>如需詳細資訊，請參閱 加密 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <p>--kms-key-id</p> <p>--storage-encrypted</p> <p>--no-storage-encrypted</p> <p>RDS API 參數：</p> <p>KmsKeyId</p> <p>StorageEncrypted</p>
Enhanced Monitoring (增強型監控)	<p>Enable enhanced monitoring (啟用增強型監控) 以針對資料庫叢集執行所在的作業系統開啟指標即時收集。</p> <p>如需詳細資訊，請參閱 使用增強型監控來監控作業系統指標。</p>	<p>CLI 選項：</p> <p>--monitoring-interval</p> <p>--monitoring-role-arn</p> <p>RDS API 參數：</p> <p>MonitoringInterval</p> <p>MonitoringRoleArn</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
初始資料庫名稱	<p>資料庫叢集上資料庫的名稱。如果您未提供名稱，Amazon RDS 不會在 MySQL 的資料庫叢集上建立資料庫。不過，它確實會在 PostgreSQL 的資料庫叢集上建立資料庫。名稱不能是資料庫引擎的保留字。它具有其他限制，取決於資料庫引擎。</p> <p>MySQL :</p> <ul style="list-style-type: none"> • 必須包含 1–64 個英數字元。 <p>PostgreSQL :</p> <ul style="list-style-type: none"> • 名稱必須包含 1–63 個英數字元。 • 開頭必須是字母或底線。後續字元可以是字母、底線或數字 (0-9)。 • 初始資料庫名為 postgres。 	<p>CLI 選項 :</p> <p>--database-name</p> <p>RDS API 參數 :</p> <p>DatabaseName</p>
Log exports (日誌匯出)	<p>要發佈到 Amazon CloudWatch 日誌的資料庫日誌檔類型。</p> <p>如需詳細資訊，請參閱 將資料庫日誌發佈至 Amazon CloudWatch Logs。</p>	<p>CLI 選項 :</p> <p>-enable-cloudwatch-logs-exports</p> <p>RDS API 參數 :</p> <p>EnableCloudwatchLogsExports</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
Maintenance window (維護時段)	<p>將擱置中修改套用至資料庫叢集的 30 分鐘時段。如果不在意時段，請選擇 No Preference (無偏好設定)。</p> <p>如需詳細資訊，請參閱 Amazon RDS 維護時段。</p>	<p>CLI 選項：</p> <pre>--preferred-maintenance-window</pre> <p>RDS API 參數：</p> <pre>PreferredMaintenanceWindow</pre>
管理中的主要認證 AWS Secrets Manager	<p>選取 Manage master credentials in AWS Secrets Manager (管理 AWS Secrets Manager 中的主要憑證) 以秘密管理 Secrets Manager 中的主要使用者密碼。</p> <p>選擇性地選擇要用來保護機密的 KMS 金鑰。從您帳戶中的 KMS 金鑰進行選擇，或輸入來自不同帳戶的金鑰。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager。</p>	<p>CLI 選項：</p> <pre>--manage-master-user-password --no-manage-master-user-password</pre> <pre>--master-user-secret-kms-key-id</pre> <p>RDS API 參數：</p> <pre>ManageMasterUserPassword</pre> <pre>MasterUserSecretKmsKeyId</pre>
Master password (主要密碼)	<p>主要使用者帳戶的密碼。</p>	<p>CLI 選項：</p> <pre>--master-user-password</pre> <p>RDS API 參數：</p> <pre>MasterUserPassword</pre>

主控台設定	設定說明	CLI 選項和 RDS API 參數
主要使用者名稱	<p>做為具有所有資料庫權限之主要使用者名稱來登入資料庫叢集的名稱。</p> <ul style="list-style-type: none">• 可含有 1–16 個英數字元和底線。• 第一個字元必須是字母。• 不能是資料庫引擎的保留字。 <p>您無法在多可用區域資料庫叢集建立之後變更主要使用者名稱。</p> <p>如需授予主要使用者權限的詳細資訊，請參閱 主要使用者帳戶權限。</p>	<p>CLI 選項：</p> <p><code>--master-username</code></p> <p>RDS API 參數：</p> <p><code>MasterUsername</code></p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
Performance Insights (績效詳情)	<p>Enable Performance Insights (啟用績效詳情) 會監控您的資料庫叢集負載，讓您可分析資料庫效能並對其進行疑難排解。</p> <p>選擇保留期間，用來判斷要保留多少績效詳情資料歷史記錄。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 Performance Insights 的定價和資料保留。</p> <p>選擇一個主金鑰，用來保護用於加密此資料庫磁碟區的金鑰。從您帳戶中的數個主金鑰進行選擇，或輸入來自不同帳戶之金鑰。</p> <p>如需詳細資訊，請參閱 在 Amazon RDS 上使用績效詳情監控資料庫負載。</p>	<p>CLI 選項：</p> <pre>--enable-performance-insights</pre> <pre>--no-enable-performance-insights</pre> <pre>--performance-insights-retention-period</pre> <pre>--performance-insights-kms-key-id</pre> <p>RDS API 參數：</p> <pre>EnablePerformanceInsights</pre> <pre>PerformanceInsightsRetentionPeriod</pre> <pre>PerformanceInsightsKMSKeyId</pre>
佈建 IOPS	<p>初始配置給資料庫叢集的佈建 IOPS 數量 (每秒輸入/輸出作業)。</p>	<p>CLI 選項：</p> <pre>--iops</pre> <p>RDS API 參數：</p> <pre>Iops</pre>

主控台設定	設定說明	CLI 選項和 RDS API 參數
公用存取	<p>Publicly accessible (可公開存取) 用來給予資料庫叢集一個公有 IP 地址，這表示可在 VPC 外加以存取。資料庫叢集也必須位於 VPC 的公有子網路中，才能公開存取。</p> <p>Not publicly accessible (不可公開存取) 將使得資料庫叢集只能從 VPC 內部存取。</p> <p>如需詳細資訊，請參閱 在 VPC 中的網際網路中隱藏資料庫執行個體。</p> <p>若要從其 VPC 外部連線到資料庫叢集，資料庫叢集必須可公開存取。此外，必須使用資料庫叢集安全群組的傳入規則授予存取權，且必須符合其他需求。如需詳細資訊，請參閱 無法連線至 Amazon RDS 資料庫執行個體。</p> <p>如果您的資料庫叢集無法公開存取，您可以使用 AWS Site-to-Site VPN 連線或連 AWS Direct Connect 線從私人網路存取該叢集。如需詳細資訊，請參閱 網際網路流量隱私權。</p>	<p>CLI 選項：</p> <p>--publicly-accessible</p> <p>--no-publicly-accessible</p> <p>RDS API 參數：</p> <p>PubliclyAccessible</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
RDS 延伸 Support	<p>選取「啟用 RDS 延伸 Support 援」以允許支援的主要引擎版本在 RDS 標準支援結束日期之後繼續執行。</p> <p>當您建立資料庫叢集時，Amazon RDS 預設為 RDS 延伸 Support。若要防止在 RDS 標準 Support 結束日期之後建立新的資料庫叢集，並避免 RDS 延伸支援的費用，請停用此設定。在 RDS 延伸 Support 定價開始日期之前，您現有的資料庫叢集不會產生費用。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS 延長支援。</p>	<p>CLI 選項：</p> <pre>--engine-lifecycle-support</pre> <p>RDS API 參數：</p> <pre>EngineLifecycleSupport</pre>
Storage throughput (訊息輸送量)	<p>資料庫叢集的儲存體輸送量值。只有在您為儲存類型選擇一般用途 SSD (gp3) 時，才能看見此設定。</p> <p>此設定無法設定，且會根據您指定的 IOPS 自動設定。</p> <p>如需詳細資訊，請參閱 gp3 儲存裝置 (建議使用)。</p>	<p>系統會自動計算此值，而且沒有 CLI 選項。</p>
RDS Proxy	<p>選擇 Create an RDS Proxy (建立 RDS 代理)，為您的資料庫叢集建立一個代理。Amazon RDS 會自動為代理建立 IAM 角色和 Secrets Manager 機密。</p>	<p>無法在建立資料庫叢集時使用。</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數
儲存體類型	<p>資料庫叢集的儲存類型。</p> <p>僅支援一般用途固態硬碟 (gp3)、佈建 IOPS (io1) 和佈建 IOPS 固態硬碟 (io2) 儲存裝置。</p> <p>如需詳細資訊，請參閱 Amazon RDS 儲存類型。</p>	<p>CLI 選項：</p> <p>--storage-type</p> <p>RDS API 參數：</p> <p>StorageType</p>
Virtual Private Cloud (VPC)	<p>要與此資料庫叢集相關聯的 VPC (以 Amazon VPC 服務為基礎)。</p> <p>如需詳細資訊，請參閱 Amazon VPC 和 Amazon RDS。</p>	對於 CLI 及 API，您指定 VPC 安全性群組 ID。
VPC security group (firewall) (VPC 安全群組 (防火牆))	<p>要與資料庫叢集建立關聯的安全群組。</p> <p>如需詳細資訊，請參閱 VPC 安全群組概觀。</p>	<p>CLI 選項：</p> <p>--vpc-security-group-ids</p> <p>RDS API 參數：</p> <p>VpcSecurityGroupIds</p>

建立多可用區域資料庫叢集時不適用的設定

AWS CLI 命令 [create-db-cluster](#) 和 RDS API 作業中的下列設定 [CreateDBCluster](#) 不適用於異地同步備份資料庫叢集。

您也無法在主控台中為多可用區域資料庫叢集指定這些設定。

AWS CLI 設置	RDS API 設定
--availability-zones	AvailabilityZones
--backtrack-window	BacktrackWindow
--character-set-name	CharacterSetName

AWS CLI 設置	RDS API 設定
<code>--domain</code>	Domain
<code>--domain-iam-role-name</code>	DomainIAMRoleName
<code>--enable-global-write-forwarding</code> <code>--no-enable-global-write-forwarding</code>	EnableGlobalWriteForwarding
<code>--enable-http-endpoint</code> <code>--no-enable-http-endpoint</code>	EnableHttpEndpoint
<code>--enable-iam-database-authentication</code> <code>--no-enable-iam-database-authentication</code>	EnableIAMDatabaseAuthentication
<code>--global-cluster-identifier</code>	GlobalClusterIdentifier
<code>--option-group-name</code>	OptionGroupName
<code>--pre-signed-url</code>	PreSignedUrl
<code>--replication-source-identifier</code>	ReplicationSourceIdentifier
<code>--scaling-configuration</code>	ScalingConfiguration

連線至多可用區域資料庫叢集

多可用區域資料庫叢集具有三個資料庫執行個體，而非單一資料庫執行個體。每個連線會由特定資料庫執行個體處理。連線到多可用區域資料庫叢集時，您指定的主機名稱和連接埠會指向稱為端點的完全限定網域名稱。多可用區域資料庫叢集會使用端點機制，抽象處理這些連線，因此您不需準確指定資料庫叢集中要連線的資料庫執行個體。因此，在某些資料庫執行個體無法使用時，您不需將所有主機名稱硬式編碼或編寫您自己的邏輯，就可以重新路由連線。

寫入器端點會連接至資料庫叢集的寫入器資料庫執行個體 (支援讀取和寫入操作)。讀取器端點連接到僅支援讀取操作的兩個讀取器資料庫執行個體中的任何一個。

使用端點，您可以根據您的使用案例將每個連線對應至適當的資料庫執行個體或資料庫執行個體群組。例如，若要執行 DDL 和 DML 陳述式，您可以連接至屬於寫入器資料庫執行個體的任何資料庫執行個體。若要執行查詢，您可以連接至讀取器端點，而多可用區域資料庫叢集會在所有讀取器資料庫執行個體間自動管理連線。針對診斷或調校，您可以連接至特定資料庫執行個體端點來檢查關於特定資料庫執行個體的詳細資訊。

如需連線至資料庫執行個體的詳細資訊，請參閱 [連接至 Amazon RDS 資料庫執行個體](#)。

主題

- [多可用區域資料庫叢集端點的類型](#)
- [檢視多可用區域資料庫叢集的端點](#)
- [使用叢集端點](#)
- [使用讀取者端點](#)
- [使用執行個體端點](#)
- [多可用區域資料庫端點如何與高可用性搭配使用](#)
- [使用驅動 AWS 程式連接異地同步備份資料庫叢集](#)

多可用區域資料庫叢集端點的類型

端點代表包含主機地址的唯一識別符。下列類型的端點可從多可用區域資料庫叢集取得。

叢集端點

多可用區域資料庫叢集的叢集端點 (或寫入器端點) 會連接至該資料庫叢集的目前寫入器資料庫執行個體。此端點是唯一可執行寫入操作 (如 DDL 和 DML 陳述式) 的端點。此端點也可以執行讀取操作。

每個多可用區域資料庫叢集具有一個叢集端點和一個寫入器資料庫執行個體。

您會對資料庫叢集上的所有寫入操作，包括插入、更新、刪除和 DDL 變更使用該叢集端點。您也可以對讀取操作 (例如查詢) 使用叢集端點。

如果資料庫叢集目前的寫入器資料庫執行個體失敗，多可用區域資料庫叢集會自動容錯移轉至新的寫入器資料庫執行個體。容錯移轉期間，資料庫叢集會繼續從新的寫入器資料庫執行個體對叢集端點提供連接請求，將對服務的中斷降到最低。

下列示範多可用區域資料庫叢集的叢集端點。

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com
```

讀取器端點

多可用區域資料庫叢集的讀取器端點可為資料庫叢集的唯一讀連線提供支援。對讀取操作 (例如 SELECT 查詢) 使用讀取器端點。藉由在讀取器資料庫執行個體上處理這些陳述式，此端點可降低寫入器資料庫執行個體上的額外負荷。它也可協助叢集擴展處理同時 SELECT 查詢的能力。每個多可用區域資料庫叢集有一個讀取器端點。

讀取器端點會將每個連線請求傳送到其中一個讀取器資料庫執行個體。當您針對工作階段使用讀取器端點時，您只能執行唯讀陳述式，例如該工作階段中的 SELECT。

下列範例說明多可用區域資料庫叢集的讀取器端點。讀取器端點的僅供讀取的意圖由叢集端點名稱內的 `-ro` 表示。

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com
```

執行個體端點

執行個體端點會連接至多可用區域資料庫叢集內的特定資料庫執行個體。資料庫叢集中的每個資料庫執行個體都有自己唯一的執行個體端點。因此，資料庫叢集目前的寫入器資料庫執行個體會有一個執行個體端點，而資料庫叢集中的每個讀取器資料庫執行個體也都會有一個執行個體端點。

執行個體端點透過與資料庫叢集的連接提供直接控制。此控制可以協助您處理使用叢集端點或讀取器端點可能不適當的案例。例如，根據工作負載類型而定，您的用戶應用程式可能要更精細的負載平衡。在此情況下，您可以設定多個用戶端來連接至資料庫叢集中的不同讀取器資料庫執行個體，以分配讀取工作負載。

下列範例說明多可用區域資料庫叢集中資料庫執行個體的執行個體端點。

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com
```

檢視多可用區域資料庫叢集的端點

在中 AWS Management Console，您會在每個異地同步備份資料庫叢集的詳細資料頁面上看到叢集端點和讀取器端點。您可以在每個資料庫執行個體的詳細資訊頁面中查看執行個體端點。

使用時 AWS CLI，您會在 [describe-db-clusters](#) 命令的輸出中看到寫入器和讀取器端點。例如，下列命令會顯示目前 AWS 區域中所有叢集的端點屬性。

```
aws rds describe-db-cluster-endpoints
```

使用 Amazon RDS API，您可以透過呼叫 [描述ClusterEndpoints](#) 動作擷取端點。輸出也會顯示 Amazon Aurora 資料庫叢集端點 (如果存在)。

使用叢集端點

每個多可用區域叢集有單一內建叢集端點，其名稱和其他屬性是由 Amazon RDS 管理。您無法建立、刪除或修改這類型的端點。

您會在管理資料庫叢集、執行擷取、轉換、載入 (ETL) 操作，或是開發和測試應用程式時使用叢集端點。叢集端點會連接至叢集的寫入器資料庫執行個體。寫入器資料庫執行個體是您可以建立資料表和索引、執行 INSERT 陳述式，以及執行其他 DDL 和 DML 操作的唯一資料庫執行個體。

叢集端點所指向的實體 IP 地址，會在容錯移轉機制將新資料庫執行個體提升為叢集的資料庫執行個體時變更。如果您使用任何形式的連線集區或其他多工處理，請準備好清除或減少任何快取 DNS time-to-live 資訊的。這麼做可確保您不會嘗試對變得無法使用或在容錯移轉之後現在為唯讀的資料庫執行個體建立讀寫連線。

使用讀取者端點

您可以為多可用區域資料庫叢集的唯一讀連接使用讀取器端點。此端點可協助您的資料庫叢集處理查詢密集型工作負載。讀取器端點是您提供給應用程式以在叢集上執行報告或其他唯讀操作的端點。讀取器端點會將連線傳送到多可用區域資料庫叢集中的可用讀取器資料庫執行個體。

每個多可用區域叢集有單一內建讀取器端點，其名稱和其他屬性是由 Amazon RDS 管理。您無法建立、刪除或修改這類型的端點。

使用執行個體端點

多可用區域資料庫叢集中的每個資料庫執行個體會有其自己的內建執行個體端點，其名稱和其他屬性是由 Amazon RDS 管理。您無法建立、刪除或修改這類型的端點。透過多可用區域資料庫叢集，您通常會比執行個體端點更常使用寫入器和讀取器端點。

在 day-to-day 操作中，使用執行個體端點的主要方法是診斷會影響異地同步備份資料庫叢集中某個特定資料庫執行個體的容量或效能問題。連接至特定資料庫執行個體時，您可以檢查其狀態變數、指標等等。這麼做可幫助您判斷該資料庫執行個體中所發生情況，與叢集中其他資料庫執行個體所發生情況的差異。

多可用區域資料庫端點如何與高可用性搭配使用

若為高可用性很重要的多可用區域資料庫叢集，請針對讀寫或一般用途連線使用寫入器端點，以及針對唯讀連線使用讀取器端點。寫入者和讀取者端點比執行個體端點更能善加管理資料庫執行個體容錯移轉。與執行個體端點不同的是，如果叢集中的資料庫執行個體變得無法使用，寫入者和讀取者端點會自動變更它們要連線到哪個資料庫執行個體。

如果資料庫叢集的寫入器資料庫執行個體失敗，Amazon RDS 會自動容錯移轉至新的寫入器資料庫執行個體。它藉由將讀取器資料庫執行個體升級為新的寫入器資料庫執行個體來執行這個操作。如果發生容錯移轉，您可以使用寫入器端點來重新連接至新提升的寫入器資料庫執行個體。或者您可以使用讀取器端點來重新連接至資料庫叢集中的其中一個讀取器資料庫執行個體。在容錯移轉期間，在讀取器資料庫執行個體提升為新的寫入器資料庫執行個體之後，讀取器端點可能會短暫直接連接至資料庫叢集的新寫入器資料庫執行個體。如果您要設計自己的應用程式邏輯來管理執行個體端點的連線，則可手動或以程式設計方式探索資料庫叢集中可用資料庫執行個體的結果集。

使用驅動 AWS 程式連接異地同步備份資料庫叢集

驅動程式 AWS 套件的設計旨在提供更快的切換和容錯移轉時間，以及使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合身分進行身份驗證的支援。AWS 驅動程式仰賴監視資料庫叢集狀態，並瞭解叢集拓撲來判斷新的寫入器。這種方法可將切換和容錯移轉時間縮短為個位數秒，而開放原始碼驅動程式則需要數十秒。

隨著新的服務功能推出，驅動程序 AWS 套件的目標是內置支持這些服務功能。

使用 Amazon Web Services (AWS) JDBC 驅動程式連接到異地同步備份資料庫叢集

Amazon Web Services (AWS) JDBC 驅動程式被設計為進階 JDBC 包裝函式，可協助應用程式利用叢集資料庫的功能。這個包裝器是補充和擴展現有 JDBC 驅動程序的功能。該驅動程序與以下社區驅動程序兼容：

- MySQL 連接器
- MariaDB 連接器/J
- PGJBC

若要安裝 AWS JDBC 驅動程式，請附加 AWS JDBC 驅動程式 .jar 檔案 (位於應用程式中CLASSPATH)，並保留對個別社群驅動程式的參考。更新相應的連接 URL 前綴，如下所示：

- jdbc:mysql:// 設定為 jdbc:aws-wrapper:mysql://
- jdbc:mariadb:// 設定為 jdbc:aws-wrapper:mariadb://
- jdbc:postgresql:// 設定為 jdbc:aws-wrapper:postgresql://

如需有關 AWS JDBC 驅動程式的詳細資訊以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS\) JDBC 驅動程式 GitHub 儲存庫](#)。

使用 Amazon Web Services 連線到異地同步備份資料庫叢集 (AWS) Python 驅動程式

Amazon Web Services (AWS) Python 驅動程序被設計為一個先進的 Python 包裝。此包裝器是補充並擴展了開源 Psycopg 驅動程序的功能。AWS Python 驅動程式支援 3.8 及更高版本。您可以使用 pip 命令以及 psycopg 開放原始碼套件來安裝套件。aws-advanced-python-wrapper

有關 AWS Python 驅動程序的更多信息以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS \) Python 驅動程序 GitHub 儲存庫](#)。

自動連線 AWS 運算資源和多可用區域資料庫叢集

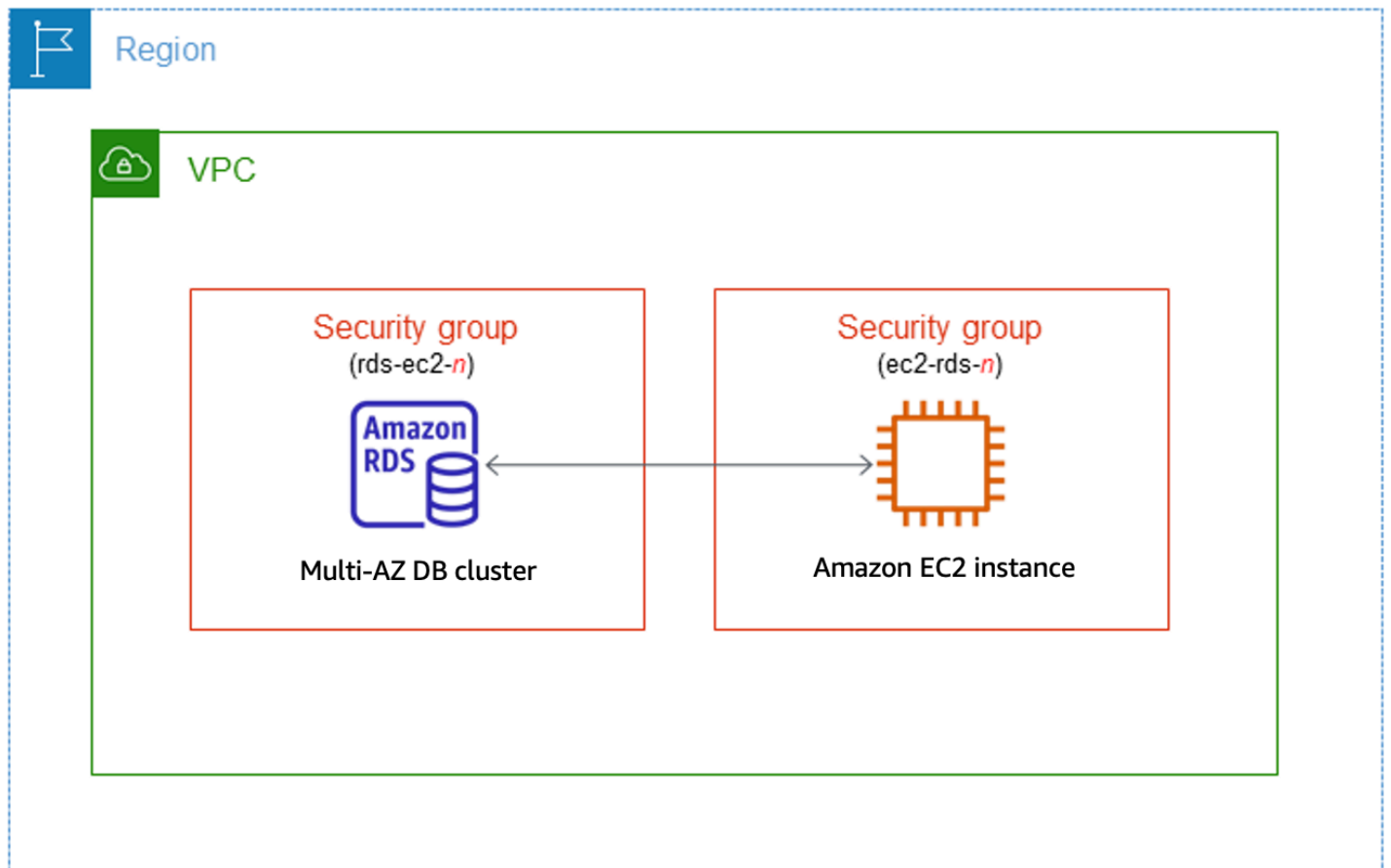
您可以自動連接多可用區域資料庫叢集和 AWS 運算資源，例如 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和 AWS Lambda 函數。

主題

- [自動連線 EC2 執行個體和多可用區域資料庫叢集](#)
- [自動連線 Lambda 函數和多可用區域資料庫叢集](#)

自動連線 EC2 執行個體和多可用區域資料庫叢集

您可以使用 Amazon RDS 主控台，簡化 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體與多可用區域資料庫叢集之間的連線設定。通常，您的多可用區域資料庫叢集位於私有子網路中，而 EC2 執行個體位於 VPC 內的公有子網路中。您可以在 EC2 執行個體上使用 SQL 用戶端，連線到多可用區域資料庫叢集。EC2 執行個體也可以執行存取私有多可用區資料庫叢集的 Web 伺服器或應用程式。



如果您想要連線至與多可用區域資料庫叢集不在相同 VPC 中的 EC2 執行個體，請參閱 [the section called “在 VPC 中存取資料庫執行個體的案例”](#) 中的案例。

主題

- [與 EC2 執行個體自動連線的概要](#)
- [自動連線 EC2 執行個體和多可用區域資料庫叢集](#)
- [檢視已連線的運算資源](#)

與 EC2 執行個體自動連線的概要

當您在 EC2 執行個體與多可用區域資料庫叢集之間自動設定連線時，Amazon RDS 會為您的 EC2 執行個體以及資料庫叢集設定 VPC 安全群組。

以下是將 EC2 執行個體與多可用區域資料庫叢集連線的要求：

- EC2 執行個體必須與多可用區域資料庫叢集存在於相同的 VPC 中。

如果沒有 EC2 執行個體存在於相同的 VPC 中，主控台會提供一個連結來建立該執行個體。

- 設定連線的使用者必須擁有執行下列 EC2 操作的許可：

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

當您設定 EC2 執行個體的連線時，Amazon RDS 會根據與多可用區域資料庫叢集和 EC2 執行個體相關聯之安全群組的目前組態行動，如下表所述。

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
有一或多個與多可用區域資料庫叢集相關聯的安全群組，其	有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合	Amazon RDS 不會採取任何動作。

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
<p>名稱符合模式 <code>rds-ec2-<i>n</i></code> (其中 <i>n</i> 是數字)。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>模式 <code>rds-ec2-<i>n</i></code> (其中 <i>n</i> 是數字)。符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其具有多可用區域資料庫叢集的 VPC 安全群組做為來源。</p>	<p>已在 EC2 執行個體與多可用區域資料庫叢集之間自動設定連線。由於 EC2 執行個體與 RDS 資料庫之間已存在連線，因此不會修改安全群組。</p>
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。 有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。不過，這些安全群組都不能用於與 EC2 執行個體的連線。如果安全群組沒有一個傳入規則，具有 EC2 執行個體的 VPC 安全群組做為來源，則無法使用該安全群組。即使已修改安全群組，也無法使用。修改範例包括新增規則或變更現有規則的連接埠。 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-<i>n</i></code>。 有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-<i>n</i></code>。不過，這些安全群組都不能用於與多可用區域資料庫叢集的連線。如果安全群組沒有傳出規則，具有多可用區域資料庫叢集的 VPC 安全群組做為來源，則無法使用安全群組。即使已修改安全群組，也無法使用。 	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>ec2-rds-<i>n</i></code>。不過，這些安全群組都不能用於與多可用區域資料庫叢集的連線。如果安全群組沒有傳出規則，具有多可用區域資料庫叢集的 VPC 安全群組做為來源，則無法使用安全群組。即使已修改安全群組，也無法使用。</p>	<p>RDS action: create new security groups</p>
<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-<i>n</i></code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。</p>	<p>存在用於連線的有效 EC2 安全群組，但與 EC2 執行個體沒有相關聯。此安全群組具有符合模式 <code>rds-ec2-<i>n</i></code> 的名稱。尚未將其修改。其只包含一個傳出規則，具有多可用區域資料庫叢集的 VPC 安全群組做為來源。</p>	<p>RDS action: associate EC2 security group</p>

目前 RDS 安全群組組態	目前 EC2 安全群組組態	RDS 動作
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。 有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。不過，這些安全群組都不能用於與 EC2 執行個體的連線。如果安全群組沒有一個傳入規則，具有 EC2 執行個體的 VPC 安全群組做為來源，則無法使用該安全群組。即使已修改安全群組，也無法使用。 	<p>有一或多個與 EC2 執行個體相關聯的安全群組，其名稱符合模式 <code>rds-ec2-n</code>。符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其具有多可用區域資料庫叢集的 VPC 安全群組做為來源。</p>	<p>RDS action: create new security groups</p>

RDS 動作：建立新的安全群組

Amazon RDS 會採取下列動作：

- 建立符合模式 `rds-ec2-n` 的新安全群組。此安全群組具有一個傳入規則，其具有 EC2 執行個體的 VPC 安全群組做為來源。與多可用區域資料庫叢集關聯的此一安全群組可讓 EC2 執行個體存取多可用區域資料庫叢集。
- 建立符合模式 `ec2-rds-n` 的新安全群組。此安全群組包含一個傳出規則，其具有多可用區域資料庫叢集的 VPC 安全群組做為來源。此安全群組與 EC2 執行個體相關聯，並可讓 EC2 執行個體將流量傳送到多可用區域資料庫叢集。

RDS 動作：關聯 EC2 安全群組

Amazon RDS 會將有效的現有 EC2 安全群組與 EC2 執行個體建立關聯。此安全群組允許 EC2 執行個體將流量傳送到多可用區域資料庫叢集。

自動連線 EC2 執行個體和多可用區域資料庫叢集

在設定 EC2 執行個體與 RDS 資料庫之間的連線之前，請確定您符合 [與 EC2 執行個體自動連線的概要](#) 中所述的要求。

如果您在設定連線之後變更這些安全群組，這些變更可能會影響 EC2 執行個體與 RDS 資料庫之間的連線。

Note

您只能使用 AWS Management Console，在 EC2 執行個體與 RDS 資料庫之間自動設定連線。您無法使用 AWS CLI 或 RDS API 自動設定連線。

自動連線 EC2 執行個體與 RDS 資料庫

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇 RDS 資料庫。
3. 從動作，選擇設定 EC2 連線。

Set up EC2 connection (設定 EC2 連線) 頁面即會出現。

4. 在 Set up EC2 connection (設定 EC2 連線) 頁面上，選擇 EC2 執行個體。

Set up EC2 connection [Info](#)

Select EC2 instance

Database
database-test1

EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-1234567890abcdef0
ec2-database-connect us-east-1c

[Create EC2 instance](#)

Cancel **Continue**

如果相同 VPC 中沒有 EC2 執行個體，請選擇 [Create EC2 instance](#) (建立 EC2 執行個體) 來建立執行個體。在此情況下，請確定新 EC2 執行個體與 RDS 資料庫位於相同的 VPC 中。

5. 選擇繼續。

[Review and confirm](#) (檢閱並確認) 頁面即會出現。

Review and confirm

Connection summary [Info](#)

You are setting up a connection between RDS database [database-test1](#) and EC2 instance [i-1234567890abcdef0](#).



Bold indicates an addition being made to set up a connection.

Changes to RDS database: database-test1

Attribute	Current value	New value
Security group	default	default, rds-ec2-1

Changes to EC2 instance: i-1234567890abcdef0

Attribute	Current value	New value
Security group	launch-wizard-5	launch-wizard-5, ec2-rds-1

Cancel

Previous

Confirm and set up

- 在 Review and confirm (檢閱並確認) 頁面上，檢閱 RDS 將做出以設定與 EC2 執行個體連線的變更。

如果變更正確，請選擇確認並設定。

如果變更不正確，請選擇 Previous (上一步) 或 Cancel (取消)。

檢視已連線的運算資源

您可以使用檢視連線 AWS Management Console 至 RDS 資料庫 Aurora 資料庫的計算資源。顯示的資源包括已自動設定的運算資源連線。您可以使用下列方式自動設定與運算資源的連線：

- 您可以在建立資料庫時選取運算資源。

如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#) 及 [建立多可用區域資料庫叢集](#)。

- 您可以設定現有資料庫與運算資源之間的連線。

如需詳細資訊，請參閱 [自動連線 EC2 執行個體和 RDS 資料庫](#)。

列出的運算資源不包括已手動連線至資料庫的運算資源。例如，您可以透過將規則新增至與資料庫關聯的 VPC 安全群組，來允許運算資源手動存取資料庫。

針對要列出的運算資源，必須滿足下列條件：

- 與運算資源相關聯的安全群組，其名稱符合模式 `ec2-rds-n` (其中 *n* 是數字)。
- 與運算資源相關聯的安全群組具有傳出規則，其連接埠範圍設為 RDS 資料庫所使用的連接埠。
- 與運算資源相關聯的安全群組具有傳出規則，其來源設為與 RDS 資料庫相關聯的安全群組。
- 與 RDS 資料庫相關聯的安全群組，其名稱符合模式 `rds-ec2-n` (其中 *n* 是數字)。
- 與 RDS 資料庫相關聯的安全群組具有傳入規則，其連接埠範圍設為 RDS 資料庫所使用的連接埠。
- 與 RDS 資料庫相關聯的安全群組具有傳入規則，其來源設為與運算資源相關聯的安全群組。

若要檢視連線至 RDS 資料庫的運算資源。

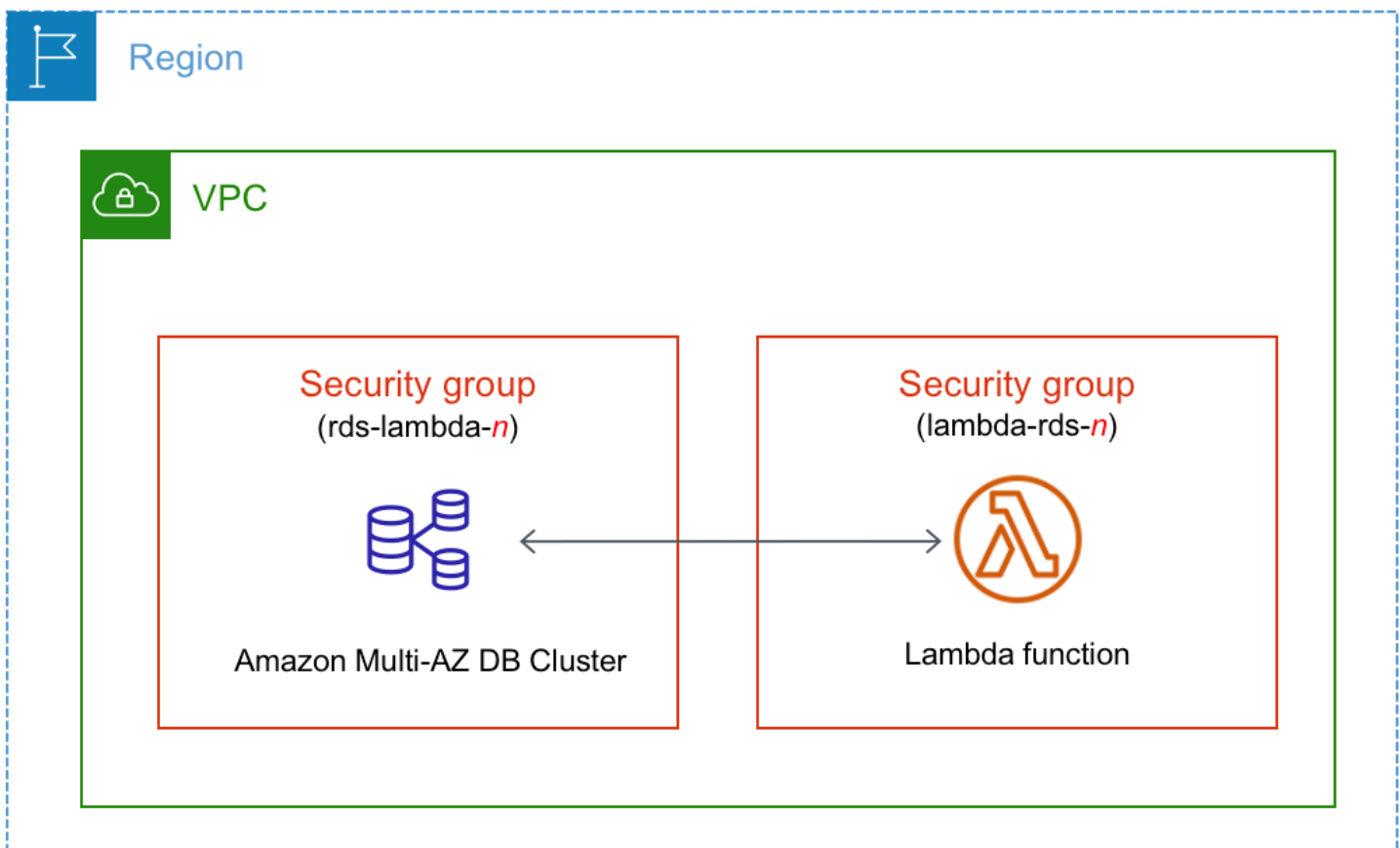
1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇 RDS 資料庫的名稱。
3. 在 Connectivity & security (連線和安全) 索引標籤上，檢視 Connected compute resources (已連線的運算資源)。

Connected compute resources (1) Info				
Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.				
<input type="text" value="Filter by compute resources"/>				
Resource identifier	Resource type	Availability zone	RDS security group	Compute resource security group
i-	EC2 Instance	us-west-1b	rds-ec2-1	ec2-rds-1

自動連線 Lambda 函數和多可用區域資料庫叢集

您可以使用 RDS 主控台，簡化 Lambda 函數與多可用區域資料庫叢集之間的連線設定。您可以使用 RDS 主控台，簡化 Lambda 函數與多可用區域資料庫叢集之間的連線設定。通常，您的多可用區域資料庫叢集位於 VPC 內的私有子網路中。應用程式可使用 Lambda 函數來存取您的私有多可用區資料庫叢集。

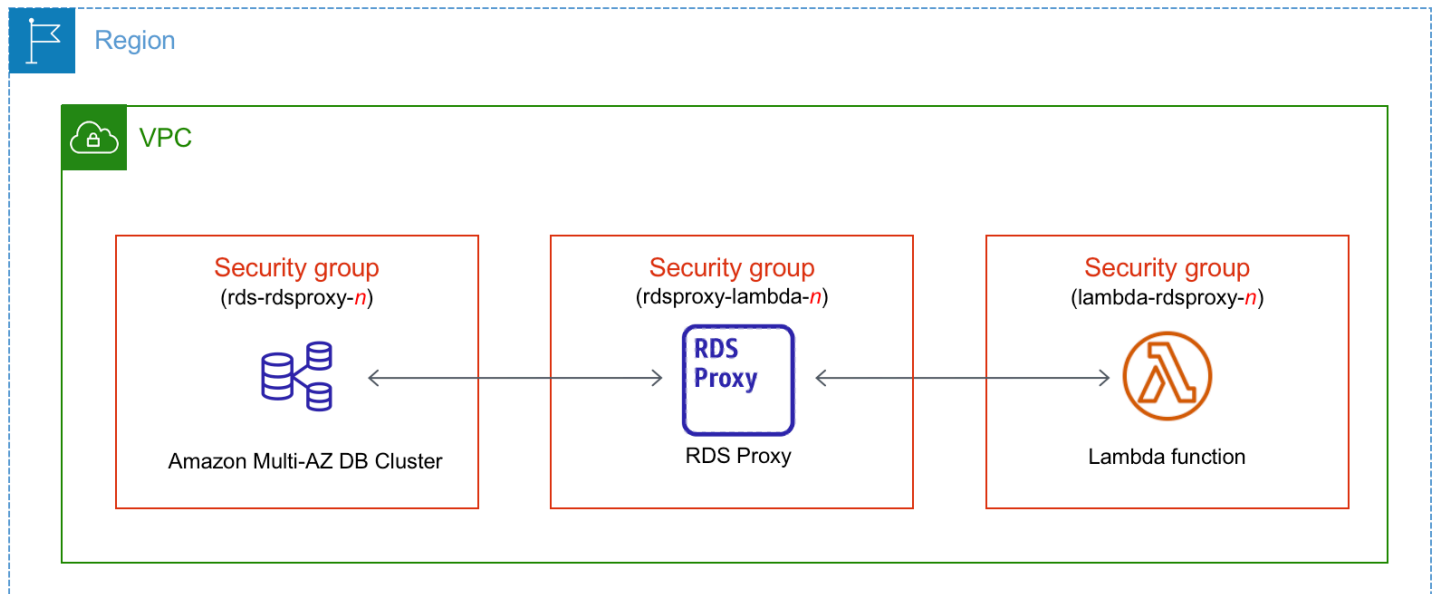
下圖顯示多可用區域資料庫叢集與 Lambda 函數之間的直接連線。



您可以透過 RDS Proxy 設定 Lambda 函數與資料庫之間的連線來改善資料庫效能和復原能力。通常，Lambda 函數經常進行短暫的資料庫連線，這些連線將受益於 RDS Proxy 提供的連線集區。您可

以利用 Lambda 函數中已有的任何 IAM 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫憑證。如需更多詳細資訊，請參閱 [使用 Amazon RDS Proxy](#)。

您可以使用主控台自動為您的連線建立代理。您也可以選取現有的代理。主控台會更新代理安全群組，以允許來自資料庫和 Lambda 函數的連線。您可以輸入資料庫憑證，或選取存取資料庫所需的 Secret Manager 密碼。



主題

- [使用 Lambda 函數自動連線概觀](#)
- [自動連線 Lambda 函數和多可用區域資料庫叢集](#)
- [檢視已連線的運算資源](#)

使用 Lambda 函數自動連線概觀

當您在 Lambda 函數與多可用區域資料庫叢集之間自動設定連線時，Amazon RDS 會為您的 Lambda 函數以及資料庫叢集設定 VPC 安全群組。

以下是將 Lambda 函數與多可用區域資料庫叢集連接的要求：

- Lambda 函數必須與多可用區域資料庫叢集存在於相同的 VPC 中。

如果沒有 Lambda 函數存在於相同的 VPC 中，主控台會提供一個連結來建立該執行個體。

- 設定連線的使用者必須擁有執行下列 Amazon RDS、Amazon EC2、Lambda、Secrets Manager 和 IAM 操作的許可：

- Amazon RDS
 - `rds:CreateDBProxies`
 - `rds:DescribeDBInstances`
 - `rds:DescribeDBProxies`
 - `rds:ModifyDBInstance`
 - `rds:ModifyDBProxy`
 - `rds:RegisterProxyTargets`
- Amazon EC2
 - `ec2:AuthorizeSecurityGroupEgress`
 - `ec2:AuthorizeSecurityGroupIngress`
 - `ec2:CreateSecurityGroup`
 - `ec2>DeleteSecurityGroup`
 - `ec2:DescribeSecurityGroups`
 - `ec2:RevokeSecurityGroupEgress`
 - `ec2:RevokeSecurityGroupIngress`
- Lambda
 - `lambda:CreateFunctions`
 - `lambda:ListFunctions`
 - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
 - `secretsmanager:CreateSecret`
 - `secretsmanager:DescribeSecret`
- IAM
 - `iam:AttachPolicy`
 - `iam:CreateRole`
 - `iam:CreatePolicy`
- AWS KMS
 - `kms:describeKey`

當您在 Lambda 函數與多可用區域資料庫叢集之間設定連線時，Amazon RDS 會為您的函數以及多可用區域資料庫叢集設定 VPC 安全群組。如果您使用 RDS Proxy，則 Amazon RDS 也會為代理設定 VPC 安全群組。Amazon RDS 會根據與安全群組相關聯的目前組態執行動作資料庫執行個體多可用區域資料庫叢集、Lambda 函數和代理，如下表所述。

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>Amazon RDS 不會採取任何動作，因為所有資源的安全群組都遵循正確的命名模式，並且擁有正確的傳入和傳出規則。</p>	<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code> (其中 <i>n</i> 是數字)，或者 TargetHealth 關聯的代理是 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code> (其中 <i>n</i> 是數字)。</p> <p>符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其使用多可用區域資料庫叢集或代理的 VPC 安全群組做為目的地。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code> (其中 <i>n</i> 是數字)。</p> <p>符合模式的安全群組尚未修改。此安全群組包含傳入和傳出規則，其具有 Lambda 函數和多可用區域資料庫叢集的 VPC 安全群組。</p>
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。 	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。 有一或多個與代理相關聯的安全 	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<ul style="list-style-type: none"> 有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。不過，Amazon RDS 無法使用這些安全群組中的任一個與 Lambda 函數連線。 <p>Amazon RDS 無法使用沒有傳入規則的安全群組，並將 Lambda 函數或代理 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。修改範例包括新增規則或變更現有規則的連接埠。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集連線。</p> <p>Amazon RDS 無法使用沒有傳出規則的安全群組，其具有多可用區域資料庫叢集的 VPC 安全群組做為來源。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>群組，其名稱符合 <code>rdsproxy-lambda-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集或 Lambda 函數連線。</p> <p>Amazon RDS 無法使用沒有傳入和傳出規則的安全群組，並與多可用區域資料庫叢集的 VPC 安全群組和 Lambda 函數搭配使用。Amazon RDS 也無法使用經過修改的安全群組。</p>	

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。</p> <p>不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集連線。Amazon RDS 無法使用沒有傳出規則的安全群組，其具有多可用區域資料庫叢集或代理的 VPC 安全群組做為目的地。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。</p> <p>不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集或 Lambda 函數連線。Amazon RDS 無法使用沒有傳入和傳出規則的安全群組，並與多可用區域資料庫叢集的 VPC 安全群組和 Lambda 函數搭配使用。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。</p> <p>符合模式的安全群組尚未修改。此安全群組只有一個傳入規則，其具有 Lambda 函數或代理的 VPC 安全群組做為來源。</p>	<p>存在用於連線的有效 Lambda 安全群組，但與 Lambda 函數沒有相關聯。此安全群組具有符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code> 的名稱。尚未將其修改。其只包含一個傳出規則，具有多可用區域資料庫叢集或代理的 VPC 安全群組做為目的地。</p>	<p>存在用於連線的有效代理安全群組，但與代理沒有相關聯。此安全群組具有符合模式 <code>rdsproxy-lambda-<i>n</i></code> 的名稱。尚未將其修改。包含傳入和傳出規則，其具有多可用區域資料庫叢集和 Lambda 函數的 VPC 安全群組。</p>	<p>RDS action: associate Lambda security group</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。 有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-lambda-<i>n</i></code>，或者 TargetHealth 相關聯的代理是 AVAILABLE。不過，Amazon RDS 無法使用這些安全群組中的任一個與 Lambda 函數或代理連線。 <p>Amazon RDS 無法使用沒有傳入規則的安全群組，並將 Lambda 函數或代理 VPC 安全群組做為來源。Amazon</p>	<p>有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。</p> <p>符合模式的安全群組尚未修改。此安全群組只包含一個傳出規則，其具有多可用區域資料庫叢集的 VPC 安全群組做為目的地。</p>	<p>有一或多個與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。</p> <p>符合模式的安全群組尚未修改。此安全群組包含傳入和傳出規則，其具有多可用區域資料庫叢集和 Lambda 函數的 VPC 安全群組。</p>	<p>RDS action: create new security groups</p>

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
RDS 也無法使用經過修改的安全群組。			

目前 RDS 安全群組組態	目前 Lambda 安全群組組態	目前代理安全群組組態	RDS 動作
<p>有一或多個與多可用區域資料庫叢集相關聯的安全群組，其名稱符合模式 <code>rds-rdsproxy-<i>n</i></code> (其中 <i>n</i> 是數字)。</p>	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。 有一或多個與 Lambda 函數相關聯的安全群組，其名稱符合模式 <code>lambda-rds-<i>n</i></code> 或 <code>lambda-rdsproxy-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集連線。 <p>Amazon RDS 無法使用沒有傳出規則的安全群組，其具有多可用區域資料庫叢集或代理的 VPC 安全群組做為目的地。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>以下任一種條件均適用：</p> <ul style="list-style-type: none"> 沒有與代理相關聯的安全群組，其名稱符合模式 <code>rdsproxy-lambda-<i>n</i></code>。 有一或多個與代理相關聯的安全群組，其名稱符合 <code>rdsproxy-lambda-<i>n</i></code>。不過，Amazon RDS 無法使用這些安全群組中的任一個與多可用區域資料庫叢集或 Lambda 函數連線。 <p>Amazon RDS 無法使用沒有傳入和傳出規則的安全群組，並與多可用區域資料庫叢集的 VPC 安全群組和 Lambda 函數搭配使用。Amazon RDS 也無法使用經過修改的安全群組。</p>	<p>RDS action: create new security groups</p>

RDS 動作：建立新的安全群組

Amazon RDS 會採取下列動作：

- 建立符合模式 `rds-lambda-n` 的新安全群組。此安全群組具有以 Lambda 函數或代理的 VPC 安全群組做為來源的傳入規則。與多可用區域資料庫叢集關聯的此一安全群組可讓函數或代理存取多可用區域資料庫叢集。
- 建立符合模式 `lambda-rds-n` 的新安全群組。此安全群組包含一個傳出規則，其具有多可用區域資料庫叢集或代理的 VPC 安全群組做為來源。此安全群組與 Lambda 函數相關聯，可讓 Lambda 函數將流量傳送到多可用區域資料庫叢集或透過代理傳送流量。
- 建立符合模式 `rdsproxy-lambda-n` 的新安全群組。此安全群組包含傳入和傳出規則，其具有多可用區域資料庫叢集和 Lambda 函數的 VPC 安全群組。

RDS 動作：與 Lambda 相關聯的安全群組

Amazon RDS 將有效、現有的 Lambda 安全群組與 Lambda 函數建立關聯。此安全群組允許函數將流量傳送到多可用區域資料庫叢集或透過 Proxy 傳送流量。

自動連線 Lambda 函數和多可用區域資料庫叢集

您可以使用 Amazon RDS 主控台將 Lambda 函數自動連線至多可用區域資料庫叢集。簡化資源之間建立連線的過程。

您也可以使用 RDS Proxy 將代理包含在連線中。Lambda 函數經常進行短暫的資料庫連線，這些連線將受益於 RDS Proxy 提供的連線集區。您還可以使用 Lambda 函數中已有的任何 IAM 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫憑證。

您可以使用設定 Lambda 連線頁面，將現有的多可用區域資料庫叢集連線至新的和現有的 Lambda 函數。設定程序會自動為您設定所需的安全群組。

在設定 Lambda 函數和多可用區域資料庫叢集之間的連線之前，請確保：

- 您的 Lambda 函數和多可用區域資料庫叢集位於相同的 VPC 中。
- 您擁有正確的使用者帳戶權限。有關需求的詳細資訊，請參閱 [使用 Lambda 函數自動連線概觀](#)。

如果您在設定連線之後變更安全群組，這些變更可能會影響 Lambda 函數與多可用區域資料庫叢集之間的連線。

Note

您只能在 AWS Management Console 中設定多可用區域資料庫叢集與 Lambda 函數之間的自動連線。若要連接 Lambda 函數，多可用區域資料庫叢集中的所有執行個體都必須處於可用狀態。

自動連線 Lambda 函數和多可用區域資料庫叢集

<result>

確認設定後，Amazon RDS 會開始連線至 Lambda 函數、RDS Proxy (如果您使用代理) 和多可用區域資料庫叢集的程序。主控台會顯示連線詳細資訊對話方塊，此對話方塊會列出允許資源之間連線的安全群組變更。

</result>

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中選擇資料庫，然後選擇您要連線至 Lambda 函數的多可用區域資料庫叢集。
3. 針對動作，選擇設定 Lambda 連線。
4. 在設定 Lambda 連線頁面中的選取 Lambda 函數，執行下列任一操作：
 - 如果您在與多可用區域資料庫叢集相同的 VPC 中擁有現有的 Lambda 函數，請選擇選擇現有函數，然後選擇函數。
 - 如果您在同一個 VPC 中沒有 Lambda 函數，請選擇建立新函數，然後輸入函數名稱。預設執行期會設定為 Nodejs.18。完成連線設定後，您可以在 Lambda 主控台中修改新 Lambda 函數的設定。
5. (選用) 在 RDS Proxy 中，選取使用 RDS Proxy 連線，然後執行下列任一項操作：
 - 若您要使用現有的代理，請選擇選擇現有的代理，然後選擇代理。
 - 如果您沒有代理，並且您希望 Amazon RDS 自動為您建立代理，請選擇建立新的代理。然後，為資料庫憑證，執行下列任一操作：
 - a. 選擇資料庫使用者名稱和密碼，然後為您的多可用區資料庫叢集輸入使用者名稱和密碼。
 - b. 選擇 Secrets Manager 密碼。然後，針對選取密碼，選擇一個 AWS Secrets Manager 密碼。如果您沒有 Secrets Manager 密碼，請選擇建立新的 Secrets Manager 密碼，以[建立新的機密](#)。建立密碼之後，針對選取密碼，選擇新的機密。

建立新的代理之後，請選擇選擇現有的代理，然後選擇代理。請注意，您的代理可能需要一些時間才能連線。

6. (選用) 展開連線摘要並為您的資源驗證反白顯示的更新。
7. 選擇 Set up (設定)。

檢視已連線的運算資源

您可以使用 AWS Management Console 檢視連線至多可用區域資料庫叢集的運算資源。顯示的資源包括 Amazon RDS 自動設定的運算資源連線。

列出的運算資源不包含手動連線至多可用區域資料庫叢集的運算資源。例如，您可以透過將規則新增至與叢集關聯的 VPC 安全群組，來允許運算資源手動存取您的多可用區域資料庫。

若要讓主控台列出 Lambda 函數，必須符合下列條件：

- 與運算資源相關聯的安全群組，其名稱符合模式 `lambda-rds-n` 或 `lambda-rdsproxy-n` (其中 *n* 是數字)。
- 與運算資源相關聯的安全群組具有傳出規則，其連接埠範圍設為多可用區域資料庫叢集或關聯代理所使用的連接埠。傳出規則目的地必須設定為與多可用區域資料庫叢集或關聯代理相關聯的安全群組。
- 與資料庫叢集附加的代理安全群組，其名稱符合模式 `rds-rdsproxy-n` (其中 *n* 是數字)。
- 與函數相關聯的安全群組具有傳出規則，其中連接埠設定為多可用區域資料庫叢集或相關 Proxy 所使用的連接埠。目的地必須設定為與多可用區域資料庫叢集或關聯代理相關聯的安全群組。

檢視自動連線至多可用區域資料庫叢集的運算資源

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇資料庫，然後選擇多可用區域資料庫叢集。
3. 在連線與安全性索引標籤中，檢視已連線的運算資源中的運算資源。

修改多可用區域資料庫叢集

多可用區域資料庫叢集在三個不同的可用區域中有一個寫入器資料庫執行個體和兩個讀取器資料庫。多可用區域資料庫叢集相較於多可用區域部署，可提供高可用性、增加讀取工作負載的容量以及更低的延遲。如需建立多可用區域資料庫叢集的詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。

您可以修改多可用區域資料庫叢集以變更其設定。您也可以在多可用區域資料庫叢集上執行操作，例如擷取其快照。

Important

您無法修改異地同步備份資料庫叢集中的資料庫執行個體。所有修改都必須在數據庫集群級別進行。您唯一可以在異地同步備份資料庫叢集內的資料庫執行個體上執行的作業是將其重新啟動。

您可以使用 AWS Management Console、或 RDS API 修改異地同步備份資料庫叢集。AWS CLI

主控台

若要修改多可用區域資料庫叢集

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的多可用區域資料庫叢集。
3. 選擇 Modify (修改)。Modify DB cluster (修改資料庫叢集) 頁面隨即出現。
4. 變更您要的任何設定。如需每項設定的相關資訊，請參閱 [修改多可用區域資料庫叢集的設定](#)。
5. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看修改摘要。
6. (選用) 選擇 Apply immediately (立即套用) 以立即套用變更。在某些情況下，選擇此選項會導致停機。如需詳細資訊，請參閱 [立即套用變更](#)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB cluster (修改資料庫叢集) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要使用修改異地同步備份資料庫叢集 AWS CLI，請呼叫指[modify-db-cluster](#)令。指定資料庫叢集識別符，以及您要修改選項的值。如需每個選項的詳細資訊，請參閱[修改多可用區域資料庫叢集的設定](#)。

Example

以下程式碼會修改 `my-multi-az-dbcluster`，將備份保留期設為 1 週 (7 天)。程式碼可藉由使用 `--deletion-protection` 來開啟刪除保護。若要關閉刪除保護，請使用 `--no-deletion-protection`。使用 `--no-apply-immediately`，會在下一次維護時段期間由系統套用變更。使用 `--apply-immediately` 可立即套用變更。如需詳細資訊，請參閱 [立即套用變更](#)。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
  --db-cluster-identifier my-multi-az-dbcluster \  
  --backup-retention-period 7 \  
  --deletion-protection \  
  --no-apply-immediately
```

在 Windows 中：

```
aws rds modify-db-cluster ^  
  --db-cluster-identifier my-multi-az-dbcluster ^  
  --backup-retention-period 7 ^  
  --deletion-protection ^  
  --no-apply-immediately
```

RDS API

若要使用 Amazon RDS API 修改多可用區域資料庫叢集，請呼叫 [ModifyDBCluster](#) 操作。指定資料庫叢集識別符，以及您要修改設定的參數。如需每項參數的相關資訊，請參閱 [修改多可用區域資料庫叢集的設定](#)。

立即套用變更

修改多可用區域資料庫叢集時，您可以立即套用變更。若要立即套用變更，從 AWS Management Console 選擇 Apply immediately (立即套用) 選項。或者，您可以在 `--apply-immediately` 使用 Amazon RDS API 呼叫 AWS CLI 或將 `ApplyImmediately` 參數設定為 `true` 時使用此選項。

若您未選擇立即套用變更，則系統會將變更內容放入待定修改佇列。在下次維護時段期間，系統便會套用佇列中的任何待定變更。若您選擇立即套用變更，則系統會一併套用待定修改佇列中的新變更及任何變更內容。

Important

如果有任何擱置的修改需要資料庫叢集暫時無法使用 (停機)，選擇立即套用選項可能會導致非預期的停機時間。

當您選擇立即套用變更時，任何待定的修改也會立即套用，而不會等到下一次維護時段才套用。

若您不想在下一個維護時段套用待定變更，您可修改資料庫執行個體以還原該變更。您可以使用 AWS CLI 並指定 `--apply-immediately` 選項來執行此操作。

即使您選擇延遲變更，針對某些資料庫設定的變更仍會立即套用。若要查看不同的資料庫設定如何影響立即套用設定，請參閱 [修改多可用區域資料庫叢集的設定](#)。

修改多可用區域資料庫叢集的設定

如需可以用來修改多可用區域資料庫叢集之設定的詳細資訊，請參閱下表。如需有關 AWS CLI 選項的更多資訊，請參閱 [modify-db-cluster](#)。如需 RDS API 參數的詳細資訊，請參閱 [ModifyDBCluster](#)。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
配置儲存	針對資料庫叢集中每個資料庫執行個體配置的儲存量 (以 GiB 為單位)。如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。	CLI 選項： <code>--allocated-storage</code> RDS API 參數： <code>AllocatedStorage</code>	如果您選擇立即套用變更，則會立即執行。 如果您未選擇立即套用變更，則會在下一個維護時段執行。	進行此變更時，不會發生停機。
Auto minor version	Enable auto minor version upgrade (啟用)	CLI 選項：	系統會立即進行變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
upgrade (自動次要版本升級)	自動次要版本升級), 讓資料庫叢集自動在可用時接收偏好次要資料庫引擎版本升級。Amazon RDS 會在維護時段執行自動次要版本升級。	<pre>--auto-minor-version-upgrade</pre> <pre>--no-auto-minor-version-upgrade</pre> <p>RDS API 參數 :</p> <pre>AutoMinorVersionUpgrade</pre>		
Backup retention period (備份保留期間)	<p>您想要資料庫叢集之自動備份保留的天數。針對任何重要的資料庫叢集, 請將此值設為 1 或更大。</p> <p>如需詳細資訊, 請參閱 備份簡介。</p>	<p>CLI 選項 :</p> <pre>--backup-retention-period</pre> <p>RDS API 參數 :</p> <pre>BackupRetentionPeriod</pre>	<p>如果您選擇立即套用變更, 則會立即執行。</p> <p>如果您未選擇立即套用變更, 並將設定從非零值變更為另一個非零值, 則系統會盡快以非同步的方式來套用變更。否則, 在下次維護時段期間, 系統才會進行變更。</p>	<p>如果將 0 變更為非零值, 或將非零值變更為 0, 便會發生停機。</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
備份時段	<p>Amazon RDS 自動備份資料庫叢集的時段。除非要在特定時段備份資料庫，否則請使用預設值 No Preference (無偏好設定)。</p> <p>如需詳細資訊，請參閱 備份簡介。</p>	<p>CLI 選項：</p> <p>--preferred-backup-window</p> <p>RDS API 參數：</p> <p>PreferredBackupWindow</p>	系統會盡快以非同步的方式來套用變更。	進行此變更時，不會發生停機。
憑證授權單位	<p>資料庫叢集所使用之伺服器憑證的憑證授權單位 (CA)。</p> <p>如需詳細資訊，請參閱。</p>	<p>CLI 選項：</p> <p>--ca-certificate-identifier</p> <p>RDS API 參數：</p> <p>CACertificateIdentifier</p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	只有在資料庫引擎不支援無需重新啟動即可輪換時，才會發生停機。您可以使用指 describe-db-engine-versions AWS CLI 令來判斷 DB 引擎是否支援旋轉而不重新啟動。
Copy tags to snapshot (將標籤複製到快照)	<p>此選項可在您建立快照時將任何資料庫叢集標籤複製到某個資料庫快照。</p> <p>如需詳細資訊，請參閱 標記 Amazon RDS 資源。</p>	<p>CLI 選項：</p> <p>-copy-tags-to-snapshot</p> <p>-no-copy-tags-to-snapshot</p> <p>RDS API 參數：</p> <p>CopyTagsToSnapshot</p>	系統會立即進行變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
Database authentication (資料庫身分驗證)	針對多可用區域資料庫叢集，僅支援密碼身分驗證。	無，因為密碼身分驗證是預設值。	如果您選擇立即套用變更，則會立即執行。 如果您未選擇立即套用變更，則會在下一個維護時段執行。	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
DB cluster identifier (資料庫叢集識別符)	<p>資料庫叢集識別符此值會以小寫字母字串的形式儲存。</p> <p>當您變更資料庫叢集識別符時，資料庫叢集端點會變更。資料庫叢集中資料庫執行個體的識別符和端點也會變更。新的資料庫叢集名稱必須是唯一的。長度上限為 63 個字元。</p> <p>資料庫叢集中資料庫執行個體的名稱會變更，以與資料庫叢集的新名稱相對應。新的資料庫執行個體名稱不能與現有資料庫執行個體的名稱相同。例如，如果您將資料庫叢集名稱變更為 maz，資料庫執行個體名稱可能會變更為 maz-instance-1。在此情況下，不能有名為 maz-instance-1 的現有資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--new-db-cluster-identifier</pre> <p>RDS API 參數：</p> <pre>NewDBClusterIdentifier</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，不會發生停機。</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
	<p>如需詳細資訊，請參閱 重新命名多可用區域資料庫叢集。</p>			
<p>資料庫叢集執行個體類別</p>	<p>多可用區域資料庫叢集中每個資料庫執行個體的運算和記憶體容量，例如 db.r6gd.xlarge。</p> <p>可能的話，選擇足夠大的資料庫執行個體類別，讓一般查詢工作集可以保留在記憶體中。當工作集保留在記憶體中時，系統可以避免寫入至磁碟，因而可改善效能。</p> <p>如需詳細資訊，請參閱 the section called “異地同步備份資料庫叢集執行個體類別”。</p>	<p>CLI 選項：</p> <p><code>--db-cluster-instance-class</code></p> <p>RDS API 參數：</p> <p><code>DBClusterInstanceClass</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	<p>進行此變更時，會發生停機。</p>

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
DB cluster parameter group (資料庫叢集參數群組)	<p>要與資料庫叢集建立關聯的資料庫叢集參數群組。</p> <p>如需詳細資訊，請參閱 使用多可用區域資料庫叢集的參數群組。</p>	<p>CLI 選項：</p> <pre>--db-cluster-parameter-group-name</pre> <p>RDS API 參數：</p> <pre>DBClusterParameterGroupName</pre>	系統會立即變更參數群組。	進行此變更時，不會發生停機。當您變更參數群組時，某些參數的變更會立即套用到多可用區域資料庫叢集的資料庫執行個體，而不需要重新啟動。其他參數的變更只有在資料庫執行個體重新啟動之後才會套用。
DB engine version (資料庫引擎版本)	您要使用的資料庫引擎版本。	<p>CLI 選項：</p> <pre>--engine-version</pre> <p>RDS API 參數：</p> <pre>EngineVersion</pre>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	進行此變更時，會發生停機。
刪除保護	<p>啟用刪除保護可避免您的資料庫叢集遭意外刪除。</p> <p>如需詳細資訊，請參閱 刪除資料庫執行個體。</p>	<p>CLI 選項：</p> <pre>--deletion-protection</pre> <pre>--no-deletion-protection</pre> <p>RDS API 參數：</p> <pre>DeletionProtection</pre>	系統會立即進行變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
Maintenance window (維護時段)	<p>將擱置中修改套用至資料庫叢集的 30 分鐘時段。如果不在意時段，請選擇 No Preference (無偏好設定)。</p> <p>如需詳細資訊，請參閱 Amazon RDS 維護時段。</p>	<p>CLI 選項：</p> <p>--preferred-maintenance-window</p> <p>RDS API 參數：</p> <p>Preferred MaintenanceWindow</p>	系統會立即進行變更。此設定會忽略立即套用設定。	如果您有一個或多個會導致停機的擱置動作，並將維護時段變更至涵蓋目前時間，則系統會立即套用這些擱置動作，最終導致停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
管理中的主要認證 AWS Secrets Manager	<p>選取 Manage master credentials in AWS Secrets Manager (管理 AWS Secrets Manager 中的主要憑證) 以秘密管理 Secrets Manager 中的主要使用者密碼。</p> <p>選擇性地選擇要用來保護機密的 KMS 金鑰。從您帳戶中的 KMS 金鑰進行選擇，或輸入來自不同帳戶的金鑰。</p> <p>如果 RDS 已在管理資料庫叢集的主要使用者密碼，您可以選擇 Rotate secret immediately (立即輪換機密) 來輪換主要使用者密碼。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS 和密碼管理 AWS Secrets Manager。</p>	<p>CLI 選項：</p> <pre>--manage-master-user-password --no-manage-master-user-password</pre> <pre>--master-user-secret-kms-key-id</pre> <pre>--rotate-master-user-password --no-rotate-master-user-password</pre> <p>RDS API 參數：</p> <pre>ManageMasterUserPassword</pre> <pre>MasterUserSecretKeyId</pre> <pre>RotateMasterUserPassword</pre>	<p>如果您開啟或關閉自動主要使用者密碼管理，變更會立即發生。此變更會忽略立即套用設定。</p> <p>如果您要輪換主要使用者密碼，則必須指定立即套用變更。</p>	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
新主要使用者密碼	主要使用者帳戶的密碼。	CLI 選項： --master-user-password RDS API 參數： MasterUserPassword	系統會盡快以非同步的方式來套用變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。
佈建 IOPS	初始配置給資料庫叢集的佈建 IOPS 數量 (每秒輸入/輸出作業) 。	CLI 選項： --iops RDS API 參數： Iops	如果您選擇立即套用變更，則會立即執行。 如果您未選擇立即套用變更，則會在下一個維護時段執行。	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
公用存取	<p>Publicly accessible (可公開存取) 用來給予資料庫叢集一個公有 IP 地址，這表示可在 Virtual Private Cloud (VPC) 外加以存取。資料庫叢集也必須位於 VPC 的公有子網路中，才能公開存取。</p> <p>Not publicly accessible (不可公開存取) 將使得資料庫叢集只能從 VPC 內部存取。</p> <p>如需詳細資訊，請參閱 在 VPC 中的網際網路中隱藏資料庫執行個體。</p> <p>若要從其 VPC 外部連線到資料庫叢集，資料庫叢集必須可公開存取。此外，必須使用資料庫叢集安全群組的傳入規則授予存取權，且必須符合其他需求。如需詳細資訊，請參閱 無</p>	無法在修改資料庫叢集時使用。	系統會立即進行變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。

主控台 設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
	<p>法連線至 Amazon RDS 資料庫執行個體。</p> <p>如果您的資料庫叢集無法公開存取，您可以使用 AWS Site-to-Site VPN 連線或連 AWS Direct Connect 線從私人網路存取該叢集。如需詳細資訊，請參閱 網際網路流量隱私權。</p>			
儲存體 類型	<p>資料庫叢集的儲存類型。</p> <p>僅支援一般用途固態硬碟 (gp3)、佈建 IOPS (io1) 和佈建 IOPS 固態硬碟 (io2) 儲存裝置。</p> <p>如需詳細資訊，請參閱 Amazon RDS 儲存類型。</p>	<p>CLI 選項：</p> <p><code>--storage-type</code></p> <p>RDS API 參數：</p> <p><code>StorageType</code></p>	<p>如果您選擇立即套用變更，則會立即執行。</p> <p>如果您未選擇立即套用變更，則會在下一個維護時段執行。</p>	進行此變更時，不會發生停機。

主控台設定	設定說明	CLI 選項和 RDS API 參數	進行變更時的狀態	停機時間備註
VPC 安全群組	<p>要與資料庫叢集建立關聯的安全群組。</p> <p>如需詳細資訊，請參閱 VPC 安全群組概觀。</p>	<p>CLI 選項：</p> <pre>--vpc-security-group-ids</pre> <p>RDS API 參數：</p> <pre>VpcSecurityGroupIds</pre>	系統會盡快以非同步的方式來套用變更。此設定會忽略立即套用設定。	進行此變更時，不會發生停機。

修改多可用區域資料庫叢集時不適用的設定

AWS CLI 命令 [modify-db-cluster](#) 和 RDS API 作業 [ModifyDBCluster](#) 中的下列設定不適用於異地同步備份資料庫叢集。

您也無法在主控台中為多可用區域資料庫叢集修改這些設定。

AWS CLI 設置	RDS API 設定
<code>--backtrack-window</code>	BacktrackWindow
<code>--cloudwatch-logs-export-configuration</code>	CloudwatchLogsExportConfiguration
<code>--copy-tags-to-snapshot</code> <code>--no-copy-tags-to-snapshot</code>	CopyTagsToSnapshot
<code>--db-instance-parameter-group-name</code>	DBInstanceParameterGroupName
<code>--domain</code>	Domain
<code>--domain-iam-role-name</code>	DomainIAMRoleName

AWS CLI 設置	RDS API 設定
<code>--enable-global-write-forwarding</code> <code>--no-enable-global-write-forwarding</code>	<code>EnableGlobalWriteForwarding</code>
<code>--enable-http-endpoint</code> <code>--no-enable-http-endpoint</code>	<code>EnableHttpEndpoint</code>
<code>--enable-iam-database-authentication</code> <code>--no-enable-iam-database-authentication</code>	<code>EnableIAMDatabaseAuthentication</code>
<code>--option-group-name</code>	<code>OptionGroupName</code>
<code>--port</code>	<code>Port</code>
<code>--scaling-configuration</code>	<code>ScalingConfiguration</code>
<code>--storage-type</code>	<code>StorageType</code>

重新命名多可用區域資料庫叢集

您可以使用 AWS Management Console、AWS CLI `modify-db-cluster` 命令或 Amazon RDS API `ModifyDBCluster` 操作，重新命名多可用區域資料庫叢集。重新命名多可用區域資料庫叢集會產生重大影響。以下是您重新命名多可用區域資料庫叢集前的考量清單。

- 當您重新命名多可用區域資料庫叢集時，多可用區域資料庫叢集的叢集端點會變更。這些端點之所以變更，因為它們包含您指派給多可用區域資料庫叢集的名稱。您可以將流量從舊端點重新導向至新端點。如需建立多可用區域資料庫叢集端點的詳細資訊，請參閱 [連線至多可用區域資料庫叢集](#)。
- 當您重新命名多可用區域資料庫叢集時，多可用區域資料庫叢集所使用的舊 DNS 名稱會加以刪除，但其會保留在快取內幾分鐘。重新命名的多可用區域資料庫叢集的新 DNS 名稱會在大約 2 分鐘內生效。重新命名的多可用區域資料庫叢集必須等到新名稱生效後才可使用。
- 重新命名叢集時，您無法使用現有的多可用區域資料庫叢集名稱。
- 如果您重複使用資料庫叢集名稱，與多可用區域資料庫叢集名稱相關聯的指標和事件會保留下來。
- 多可用區域資料庫叢集標籤會與多可用區域資料庫叢集一起保留，無論重新命名與否。
- 重新命名的多可用區域資料庫叢集會保留其資料庫叢集快照。

Note

多可用區域資料庫叢集是在雲端中執行的隔離資料庫環境。多可用區域資料庫叢集可以託管多個資料庫。如需有關變更資料庫名稱的資訊，請參閱資料庫引擎相關文件。

重新命名以取代現有的多可用區域資料庫叢集

重新命名異地同步備份資料庫叢集的最常見案例包括從資料庫叢集快照還原資料或執行 point-in-time 復原 (PITR)。透過重新命名多可用區域資料庫叢集，您可以取代多可用區域資料庫叢集，而無需變更參考多可用區域資料庫叢集的任何應用程式碼。在這些情況下，完成下列步驟：

1. 停止所有流量進入多可用區域資料庫叢集。您可以重新導向流量，以免其存取多可用區域資料庫叢集上的資料庫，或選擇其他方式，來防止流量存取多可用區域資料庫叢集上的資料庫。
2. 重新命名現有的多可用區域資料庫叢集。
3. 透過從資料庫叢集還原或復原至某個時間點，來建立新的多可用區域資料庫叢集。然後，為新的多可用區域資料庫叢集提供先前的多可用區域資料庫叢集的名稱。

如果您刪除舊的多可用區域資料庫叢集，則必須負責刪除舊的多可用區域資料庫叢集的任何不需要的資料庫叢集快照。

主控台

重新命名多可用區域資料庫叢集

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要重新命名的多可用區域資料庫叢集。
4. 選擇 Modify (修改)。
5. 在 Settings (設定) 中，針對 DB cluster identifier (資料庫叢集識別符)，輸入新名稱。
6. 選擇 Continue (繼續)。
7. 若要立即套用變更，請選擇 Apply immediately (立即套用)。在某些情況下，選擇此選項會導致停機。如需詳細資訊，請參閱 [立即套用變更](#)。
8. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify cluster (修改叢集) 以儲存您的變更。
或者，選擇 Back (上一步) 以編輯您的變更，或是選擇 Cancel (取消) 以捨棄您的變更。

AWS CLI

若要重新命名異地同步備份資料庫叢集，請使用指AWS CLI令 [modify-db-cluster](#)。使用多可用區域資料庫叢集的新名稱提供目前的 `--db-cluster-identifier` 值和 `--new-db-cluster-identifier` 參數。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
  --db-cluster-identifier DBClusterIdentifier \  
  --new-db-cluster-identifier NewDBClusterIdentifier
```

在Windows中：

```
aws rds modify-db-cluster ^  
  --db-cluster-identifier DBClusterIdentifier ^
```

```
--new-db-cluster-identifier NewDBClusterIdentifier
```

RDS API

若要重新命名多可用區域資料庫叢集，請搭配下列參數呼叫 Amazon RDS API 操作 [ModifyDBCluster](#)：

- `DBClusterIdentifier` – 資料庫叢集的現有名稱。
- `NewDBClusterIdentifier` – 資料庫叢集的新名稱。

重新啟機多可用區域資料庫叢集和讀取器資料庫執行個體

您可能需要重新啟動多可用區域資料庫叢集，這通常是為了維護的原因。例如，假如您進行了某些修改，或變更與資料庫叢集相關聯的資料庫叢集參數群組，則要重新啟動資料庫叢集。這樣才可讓變更生效。

如果資料庫叢集未在其關聯的資料庫叢集參數群組中使用最新的變更，AWS Management Console 會顯示資料庫叢集參數群組，其狀態為 pending-reboot (等待重新啟動)。pending-reboot (等待重新啟動) 參數群組狀態不會在下一個維護時段期間造成自動重新啟動。若要對該資料庫叢集套用最新參數變更，請手動重新啟動資料庫叢集。如需參數群組的詳細資訊，請參閱[使用多可用區域資料庫叢集的參數群組](#)。

重新啟動資料庫叢集，將會重新啟動資料庫引擎服務。重新啟動資料庫叢集會暫時中斷，在此期間，資料庫叢集狀態設定為 rebooting (重新啟動中)。

如果資料庫叢集不是 Available (可用) 狀態，就無法重新啟動資料庫叢集。資料庫無法使用可能由多種原因導致，例如，備份進行中、先前要求的修改，或維護時段的動作。

重新啟動資料庫叢集所需的時間取決於損毀復原程序、重新啟動當時的資料庫活動，以及特定資料庫叢集的行為。若要改善重新啟動時間，建議在重新啟動程序期間盡可能減少資料庫活動。減少資料庫活動，可降低傳輸中交易的轉返活動。

Important

多可用區域資料庫叢集不支援使用容錯移轉重新啟動。當您重新啟動多可用區域資料庫叢集的寫入器執行個體時，不會影響該資料庫叢集中的讀取器資料庫執行個體，也不會發生容錯移轉。重新啟動讀取器資料庫執行個體時，不會發生容錯移轉。若要容錯移轉多可用區域資料庫叢集，請在主控台中選擇 Failover (容錯移轉)，呼叫 AWS CLI 命令 [failover-db-cluster](#)，或呼叫 API 操作 [FailoverDBCluster](#)。

主控台

若要重新啟動資料庫叢集

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要重新啟動的多可用區域資料庫叢集。
3. 針對 Actions (動作)，選擇 Reboot (重新啟動)。

Reboot DB cluster (重新啟動資料庫叢集) 頁面隨即出現。

4. 選擇 Reboot (重新啟動) 以重新啟動您的資料庫叢集。

或者選擇 Cancel (取消)。

AWS CLI

若要藉由使用 AWS CLI 重新啟動多可用區域資料庫叢集，請呼叫 [reboot-db-cluster](#) 命令。

```
aws rds reboot-db-cluster --db-cluster-identifier mymulti-az-db-cluster
```

RDS API

若要藉由使用 Amazon RDS API 重新啟動多可用區域資料庫叢集，請呼叫 [RebootDBCluster](#) 操作。

使用多可用區域資料庫叢集僅供讀取複本

資料庫叢集僅供讀取複本是一種特別的叢集類型，您可以從來源資料庫執行個體建立。建立僅供讀取複本後，對主要資料庫執行個體所做的任何更新，會以非同步方式複製到多可用區域資料庫叢集僅供讀取複本。您可以將讀取查詢從應用程式路由到僅供讀取複本，以減少主要資料庫執行個體的負載。使用僅供讀取複本，您可以彈性地擴展超越單一資料庫執行個體的容量限制，以處理高讀取量的資料庫工作負載。

您也可以從多可用區域資料庫叢集建立一或多個資料庫執行個體的僅供讀取複本。資料庫執行個體僅供讀取複本可讓您將過多的讀取流量導向僅供讀取複本，進而擴充超越來源多可用區域資料庫叢集的運算或 I/O 容量。目前，您無法從現有的多可用區域資料庫叢集建立多可用區域資料庫叢集的僅供讀取複本。

主題

- [使用僅供讀取複本遷移至多可用區域資料庫叢集](#)
- [從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本](#)

使用僅供讀取複本遷移至多可用區域資料庫叢集

若要將單一可用區域部署或多可用區域資料庫執行個體部署遷移至多可用區域資料庫叢集部署，並減少停機時間，您可以建立多可用區域資料庫執行個體的僅供讀取複本。對於來源，您可以在單一可用區域部署中指定資料庫執行個體，或在多可用區域資料庫執行個體部署中指定主要資料庫執行個體。資料庫執行個體可以在遷移至多可用區域資料庫叢集期間處理寫入交易。

建立多可用區域資料庫叢集僅供讀取複本之前，請考量以下項目：

- 來源資料庫執行個體必須位於支援多可用區域資料庫叢集的版本。如需詳細資訊，請參閱 [Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎](#)。
- 多可用區域資料庫叢集僅供讀取複本必須與其來源位於相同的主要版本，並位於相同或更新的次要版本。
- 您必須將備份保留期設定為 0 以外的值，才能開啟來源資料庫執行個體上的自動備份。
- 來源資料庫執行個體的配置儲存體必須為 100 GiB 或更高。
- 對於 RDS for MySQL，必須將來源資料庫執行個體的 `gtid-mode` 和 `enforce_gtid_consistency` 參數都設定為 ON。您必須使用自訂參數群組，而不是預設參數群組。如需詳細資訊，請參閱 [the section called “使用資料庫參數群組”](#)。
- 長時間執行的作用中交易會使建立僅供讀取複本的流程變慢。建議您等到長時間執行交易完成後，再建立僅供讀取複本。

- 如果您刪除多可用區域資料庫叢集的來源資料庫執行個體，僅供讀取複本會提升為獨立的多可用區域資料庫叢集。

建立和提升多可用區域資料庫叢集僅供讀取複本

您可以使用 AWS Management Console、AWS CLI 或 RDS API 建立和升級異地同步備份資料庫叢集僅供讀取複本。

Note

強烈建議您根據來源資料庫執行個體的 Amazon VPC，在相同的虛擬私有雲端 (VPC) 中建立所有僅供讀取複本。

如果您在來源資料庫執行個體的不同 VPC 中建立僅供讀取複本，則無類別網域間路由 (CIDR) 範圍可能會在複本和 Amazon RDS 系統之間重疊。CIDR 重疊會使複本不穩定，這會對連線到該複本的應用程式產生負面影響。如果您在建立僅供讀取複本時收到錯誤，請選擇不同的目的地資料庫子網路群組。如需詳細資訊，請參閱 [在 VPC 中使用資料庫執行個體](#)。

主控台

若要使用僅供讀取複本，將單一可用區域部署或多可用區域資料庫執行個體部署遷移至多可用區域資料庫叢集，請使用 AWS Management Console 完成下列步驟。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 建立多可用區域資料庫叢集僅供讀取複本。
 - a. 在導覽窗格中，選擇 Databases (資料庫)。
 - b. 選擇要作為僅供讀取複本來源的資料庫執行個體。
 - c. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
 - d. 針對 Availability and durability (可用性與持久性)，選擇 Multi-AZ DB cluster (多可用區域資料庫叢集)。
 - e. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入僅供讀取複本的名稱。
 - f. 在其餘區段，指定資料庫叢集設定。如需設定的相關資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。
 - g. 選擇 Create read replica (建立僅供讀取複本)。
3. 備妥時，將僅供讀取複本提升為獨立的多可用區域資料庫叢集：

- a. 停止任何交易寫入至僅供來源資料庫執行個體，然後等待針對僅供讀取複本所做的所有更新。
資料庫更新在主要資料庫執行個體上發生之後，會在僅供讀取複本上發生。此複寫延遲可能會有很大不同。使用 `ReplicaLag` 指標以確定針對僅供讀取複本進行的所有更新時間。如需複本延遲的詳細資訊，請參閱 [監控僅供讀取複本](#)。
- b. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
- c. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。
Databases (資料庫) 窗格隨即出現。每個僅供讀取複本會在 Role (角色) 欄中顯示 Replica (複本)。
- d. 選擇您要提升的多可用區域資料庫叢集僅供讀取複本。
- e. 針對 Actions (動作)，選擇 Promote (提升)。
- f. 在 Promote read replica (提升僅供讀取複本) 頁面上，輸入新提升之多可用區域資料庫叢集的備份保留期和備份時段。
- g. 依您所需完成設定後，選擇 Promote read replica (提升僅供讀取複本)。
- h. 等待提升的多可用區域資料庫叢集狀態成為 Available。
- i. 引導您的應用程式使用提升的多可用區域資料庫叢集。

選擇性地刪除單一可用區域部署或多可用區域資料庫執行個體部署 (如果不再需要)。如需說明，請參閱 [刪除資料庫執行個體](#)。

AWS CLI

若要使用僅供讀取複本，將單一可用區域部署或多可用區域資料庫執行個體部署遷移至多可用區域資料庫叢集，請使用 AWS CLI 完成下列步驟。

1. 建立多可用區域資料庫叢集僅供讀取複本。

若要從來源資料庫執行個體建立僅供讀取複本，請使用 AWS CLI 指令 [create-db-cluster](#)。若為 `--replication-source-identifier`，請指定來源資料庫執行個體的 Amazon Resource Name (ARN)。

對於 Linux macOS、或 Unix：

```
aws rds create-db-cluster \
```

```
--db-cluster-identifier mymultiadbcluster \  
--replication-source-identifier arn:aws:rds:us-  
east-2:123456789012:db:mydbinstance \  
--engine postgres \  
--db-cluster-instance-class db.m5d.large \  
--storage-type io1 \  
--iops 1000 \  
--db-subnet-group-name defaultvpc \  
--backup-retention-period 1
```

在Windows中：

```
aws rds create-db-cluster ^  
--db-cluster-identifier mymultiadbcluster ^  
--replication-source-identifier arn:aws:rds:us-  
east-2:123456789012:db:mydbinstance ^  
--engine postgres ^  
--db-cluster-instance-class db.m5d.large ^  
--storage-type io1 ^  
--iops 1000 ^  
--db-subnet-group-name defaultvpc ^  
--backup-retention-period 1
```

2. 停止任何交易寫入至僅供來源資料庫執行個體，然後等待針對僅供讀取複本所做的所有更新。

資料庫更新在主要資料庫執行個體上發生之後，會在僅供讀取複本上發生。此複寫延遲可能會有很大不同。使用 Replica Lag 指標以確定針對僅供讀取複本進行的所有更新時間。如需複本延遲的詳細資訊，請參閱 [監控僅供讀取複本](#)。

3. 備妥時，將僅供讀取複本提升為獨立的多可用區域資料庫叢集。

若要提升多可用區域資料庫叢集僅供讀取複本，請使用 AWS CLI 命令 [promote-read-replica-db-cluster](#)。若為 `--db-cluster-identifier`，請指定多可用區域資料庫叢集僅供讀取複本的識別符。

```
aws rds promote-read-replica-db-cluster --db-cluster-identifier mymultiadbcluster
```

4. 等待提升的多可用區域資料庫叢集狀態成為 Available。
5. 引導您的應用程式使用提升的多可用區域資料庫叢集。

選擇性地刪除單一可用區域部署或多可用區域資料庫執行個體部署 (如果不再需要)。如需說明，請參閱 [刪除資料庫執行個體](#)。

RDS API

若要使用僅供讀取複本，將單一可用區域部署或多可用區域資料庫執行個體部署遷移至多可用區域資料庫叢集，請使用 RDS API 完成下列步驟。

1. 建立多可用區域資料庫叢集僅供讀取複本。

若要建立多可用區域資料庫叢集僅供讀取複本，請使用 [CreateDBCluster](#) 操作搭配必要參數 `DBClusterIdentifier`。若為 `ReplicationSourceIdentifier`，請指定來源資料庫執行個體的 Amazon Resource Name (ARN)。

2. 停止任何交易寫入至僅供來源資料庫執行個體，然後等待針對僅供讀取複本所做的所有更新。

資料庫更新在主要資料庫執行個體上發生之後，會在僅供讀取複本上發生。此複寫延遲可能會有很大不同。使用 `Replica Lag` 指標以確定針對僅供讀取複本進行的所有更新時間。如需複本延遲的詳細資訊，請參閱 [監控僅供讀取複本](#)。

3. 備妥時，將僅供讀取複本提升為獨立的多可用區域資料庫叢集。

若要提升多可用區域資料庫叢集僅供讀取複本，請使用 [PromoteReadReplicaDBCluster](#) 操作搭配必要參數 `DBClusterIdentifier`。指定多可用區域資料庫叢集僅供讀取複本的識別符。

4. 等待提升的多可用區域資料庫叢集狀態成為 `Available`。
5. 引導您的應用程式使用提升的多可用區域資料庫叢集。

選擇性地刪除單一可用區域部署或多可用區域資料庫執行個體部署 (如果不再需要)。如需說明，請參閱 [刪除資料庫執行個體](#)。

建立多可用區域資料庫叢集僅供讀取複本的限制

下列限制適用於從單一可用區域部署或多可用區域資料庫執行個體部署建立多可用區域叢集僅供讀取複本。

- 您無法在不同於擁有來源資料庫執行個體 AWS 帳戶 體的異地同步備份資料庫叢集僅供讀取複本中建立。AWS 帳戶
- 您無法在不同於來源資料庫執行個體中建立異地同 AWS 區域 步備份資料庫叢集僅供讀取複本。
- 您無法將多可用區域資料庫叢集僅供讀取複本復原至某個時間點。
- 儲存體加密必須在來源資料庫執行個體和多可用區域資料庫叢集上具有相同的設定。

- 如果來源資料庫執行個體已加密，則多可用區域資料庫叢集僅供讀取複本必須使用相同的 KMS 金鑰加密。
- 如果來源資料庫執行個體使用一般用途 SSD (gp3) 儲存，且配置的儲存體少於 400 GiB，則無法修改異地同步備份資料庫叢集僅供讀取複本的佈建 IOPS。
- 若要在來源資料庫執行個體上執行次要版本升級，您必須先在多可用區域資料庫叢集僅供讀取複本上執行次要版本升級。
- 當您在 RDS for PostgreSQL 異地同步備份資料庫叢集僅供讀取複本上執行次要版本升級時，讀取器資料庫執行個體在升級後不會切換至寫入器資料庫執行個體。因此，Amazon RDS 升級寫入器執行個體時，您的資料庫叢集可能會遇到停機時間。
- 您無法在異地同步備份資料庫叢集僅供讀取複本上執行主要版本升級。
- 您可以在多可用區域資料庫叢集僅供讀取複本的來源資料庫執行個體上執行主要版本升級，但複寫至僅供讀取複本會停止且無法重新啟動。
- 多可用區域資料庫叢集僅供讀取複本不支援階層式僅供讀取複本。
- 對於 RDS for PostgreSQL，多可用區域資料庫叢集僅供讀取複本無法容錯移轉。

從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本

您可以從多可用區域資料庫叢集建立資料庫執行個體的僅供讀取複本，以便擴充超越叢集的運算或 I/O 容量，用以處理高讀取量的資料庫工作負載。您可以將此多餘讀取流量指向到一或多個資料庫執行個體僅供讀取複本。您也可以使用僅供讀取複本，從多可用區域資料庫叢集遷移至資料庫執行個體。

若要建立僅供讀取複本，請指定多可用區域資料庫叢集作為複寫來源。多可用區域資料庫叢集的其中一個讀取器執行個體永遠是複寫的來源，而不是寫入器執行個體。此條件可確保複本始終與來源叢集同步，即使在容錯移轉的情況下也是如此。

主題

- [比較讀取器資料庫執行個體和資料庫執行個體僅供讀取複本](#)
- [考量事項](#)
- [建立資料庫執行個體僅供讀取複本](#)
- [提升資料庫執行個體僅供讀取複本](#)
- [從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本的限制](#)

比較讀取器資料庫執行個體和資料庫執行個體僅供讀取複本

多可用區域資料庫叢集的資料庫執行個體僅供讀取複本在下列方面與多可用區域資料庫叢集的讀取器資料庫執行個體不同：

- 讀取器資料庫執行個體會做為自動容錯移轉目標，而資料庫執行個體僅供讀取複本不是。
- 讀取器資料庫執行個體必須先確認來自寫入器資料庫執行個體的變更，才能遞交變更。然而，對於資料庫執行個體僅供讀取複本，不需要確認，更新就會以非同步方式複製到僅供讀取複本。
- 讀取器資料庫執行個體一律與多可用區域資料庫叢集的寫入器資料庫執行個體共用相同的執行個體類別、儲存類型和引擎版本。但是，資料庫執行個體僅供讀取複本不一定需要與來源叢集共用相同的組態。
- 您可將資料庫執行個體僅供讀取複本提升為獨立資料庫執行個體。您無法將多可用區域資料庫叢集的讀取器資料庫執行個體提升為獨立執行個體。
- 讀取器端點僅會將請求路由至多可用區域資料庫叢集的讀取器資料庫執行個體。它絕不會將請求路由到資料庫執行個體僅供讀取複本。

如需有關讀取器和寫入器資料庫執行個體的詳細資訊，請參閱 [the section called “多可用區域資料庫叢集概觀”](#)。

考量事項

從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本之前，請考量以下項目：

- 當您建立資料庫執行個體僅供讀取複本時，它必須與其來源叢集位於相同的主要版本，並位於相同或更新的次要版本。建立僅供讀取複本之後，您可以選擇性地將僅供讀取複本升級到比來源叢集更高的次要版本。
- 建立資料庫執行個體僅供讀取複本時，配置的儲存體必須與來源多可用區域資料庫叢集的配置儲存體相同。您可以在建立僅供讀取複本之後變更已配置的儲存體。
- 對於 RDS for MySQL，來源多可用區域資料庫叢集的 `gtid-mode` 參數必須設定為 `ON`。如需詳細資訊，請參閱 [the section called “使用資料庫叢集參數群組”](#)。
- 長時間執行的作用中交易會使建立僅供讀取複本的流程變慢。建議您等到長時間執行交易完成後，再建立僅供讀取複本。
- 如果您刪除資料庫執行個體僅供讀取複本的來源多可用區域資料庫叢集，它所寫入的任何僅供讀取複本都會提升為獨立的資料庫執行個體。

建立資料庫執行個體僅供讀取複本

您可以使用 AWS Management Console、AWS CLI 或 RDS API 從異地同步備份資料庫叢集建立資料庫執行個體僅供讀取複本。

Note

強烈建議您根據來源多可用區域資料庫叢集的 Amazon VPC，在相同的虛擬私有雲端 (VPC) 中建立所有僅供讀取複本。

如果您在與來源多可用區域資料庫叢集不同的 VPC 中建立僅供讀取複本，則無類別域間路由 (CIDR) 範圍可能會在複本和 RDS 系統之間重疊。CIDR 重疊會使複本不穩定，這會對連線到該複本的應用程式產生負面影響。如果您在建立僅供讀取複本時收到錯誤，請選擇不同的目的地資料庫子網路群組。如需詳細資訊，請參閱 [the section called “在 VPC 中使用資料庫執行個體”](#)。

主控台

若要從多可用區域資料庫叢集建立資料庫執行個體，請使用 AWS Management Console 完成下列步驟。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇要作為僅供讀取複本來源的多可用區域資料庫叢集。
4. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
5. 對於複本來源，請確定已選取正確的多可用區域資料庫叢集。
6. 在資料庫識別符中，輸入僅供讀取複本的名稱。
7. 在其餘區段，指定資料庫執行個體的設定。如需設定的相關資訊，請參閱 [the section called “可用設定”](#)。

Note

資料庫執行個體僅供讀取複本配置的儲存體必須與來源多可用區域資料庫叢集配置的儲存體相同。

8. 選擇 Create read replica (建立僅供讀取複本)。

AWS CLI

若要從異地同步備份資料庫叢集建立資料庫執行個體僅供讀取複本，請使用指 AWS CLI 令 [create-db-instance-read-replica](#)。若為 `--source-db-cluster-identifier`，請指定多可用區域資料庫叢集的識別符。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --source-db-cluster-identifier mymultiazdbcluster
```

在Windows中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^  
  --source-db-cluster-identifier mymultiazdbcluster
```

RDS API

若要從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本，請使用 [CreateDBInstanceReadReplica](#) 操作。

提升資料庫執行個體僅供讀取複本

如果您不再需要資料庫執行個體僅供讀取複本，您可以將其提升為獨立的資料庫執行個體。當您提升僅供讀取複本時，資料庫執行個體將在可使用前重新啟動。如需說明，請參閱 [the section called “提升僅供讀取複本”](#)。

如果您使用僅供讀取複本，將多可用區域資料庫叢集部署遷移至單一可用區域或多可用區域資料庫執行個體部署，請務必停止正在寫入來源資料庫叢集的任何交易。然後，等候對僅供讀取複本進行的所有更新。在多可用區域資料庫叢集的其中一個讀取器資料庫執行個體上進行資料庫更新後，才會在僅供讀取複本上發生資料庫更新。此複寫延遲可能會有很大不同。使用 `ReplicaLag` 指標以確定針對僅供讀取複本進行的所有更新時間。如需複本延遲的詳細資訊，請參閱 [the section called “監控僅供讀取複本”](#)。

提升僅供讀取複本之後，請等待提升資料庫執行個體的狀態變成 `Available`，接著再指示您的應用程式使用提升後的資料庫執行個體。或者，如果不再需要，您可以選擇刪除多可用區域資料庫叢集部署。如需說明，請參閱 [the section called “刪除多可用區域資料庫叢集”](#)。

從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本的限制

下列限制適用於從多可用區域資料庫叢集部署建立資料庫執行個體僅供讀取複本。

- 您無法在不同於擁有來源異地同步備份資料庫叢集 AWS 帳戶 的資料庫執行個體僅供讀取複本中建立。AWS 帳戶
- 您無法在不同於來源異地同 AWS 區域 步備份資料庫叢集中建立資料庫執行個體僅供讀取複本。
- 您無法將資料庫執行個體僅供讀取複本復原至某個時間點。
- 儲存體加密必須在來源多可用區域資料庫叢集和資料庫執行個體僅供讀取複本上具有相同的設定。
- 如果來源多可用區域資料庫叢集已加密，則資料庫執行個體僅供讀取複本必須使用相同的 KMS 金鑰來加密。
- 若要在來源多可用區域資料庫叢集上執行次要版本升級，您必須先在資料庫執行個體僅供讀取複本上執行次要版本升級。
- 資料庫執行個體僅供讀取複本不支援階層式僅供讀取複本。
- 對於 RDS for PostgreSQL，來源多可用區域資料庫叢集必須執行 PostgreSQL 13.11、14.8 或 15.2.R2 版或更新版本，才能建立資料庫執行個體僅供讀取複本。
- 您可以在資料庫執行個體僅供讀取複本的來源多可用區域資料庫叢集上執行主要版本升級，但複寫至僅供讀取複本會停止且無法重新啟動。

搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫

透過搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫，您可以複寫和同步個別資料表，而不是整個資料庫執行個體。邏輯複寫會使用發佈與訂閱模型，將來源中的變更複寫到一或多個收件人。其運作方式為使用 PostgreSQL 預寫日誌 (WAL) 中的變更記錄。如需詳細資訊，請參閱 [the section called “邏輯複寫”](#)。

當您在多可用區域資料庫叢集的寫入器資料庫執行個體上建立新的邏輯複寫插槽時，插槽會以非同步方式複製到叢集中的每個讀取器資料庫執行個體。讀取器資料庫執行個體上的插槽會持續與寫入器資料庫執行個體上的插槽同步。

執行 RDS for PostgreSQL 14.8-R2 版及更新版本，以及 15.3-R2 版及更新版本的多可用區域資料庫叢集支援邏輯複寫。

Note

除了原生 PostgreSQL 邏輯複寫功能之外，執行 RDS for PostgreSQL 的多可用區域資料庫叢集也支援 pglogical 延伸模組。

如需 PostgreSQL 邏輯複寫的詳細資訊，請參閱 PostgreSQL 文件中的 [邏輯複寫](#)。

主題

- [必要條件](#)
- [設定邏輯複寫](#)
- [限制與建議](#)

必要條件

若要為多可用區域資料庫叢集設定 PostgreSQL 邏輯複寫，您必須符合下列先決條件。

- 您的使用者帳戶必須是 `rds_superuser` 群組的成員並具有 `rds_superuser` 權限。如需詳細資訊，請參閱 [the section called “了解 PostgreSQL 角色和許可”](#)。
- 您的多可用區域資料庫叢集必須與自訂資料庫叢集參數群組相關聯，以便您可以設定下列程序中所述的參數值。如需詳細資訊，請參閱 [the section called “使用資料庫叢集參數群組”](#)。

設定邏輯複寫

若要為多可用區域資料庫叢集設定邏輯複寫，請在相關聯的資料庫叢集參數群組內啟用特定參數，然後建立邏輯複寫插槽。

Note

從 PostgreSQL 第 16 版開始，您可以使用異地同步備份資料庫叢集的讀取器資料庫執行個體進行邏輯複寫。

為 RDS for PostgreSQL 多可用區域資料庫叢集設定邏輯複寫

1. 開啟與 RDS for PostgreSQL 多可用區域資料庫叢集相關聯的自訂資料庫叢集參數群組。
2. 在參數搜尋欄位中，找出 `rds.logical_replication` 靜態參數並將其值設為 1。此參數變更可能會增加 WAL 產生，因此只有在使用邏輯插槽時才啟用該變更。
3. 做為此變更的一部分，請設定下列資料庫叢集參數。
 - `max_wal_senders`
 - `max_replication_slots`
 - `max_connections`

根據您預期的用量，您可能也需要變更下列參數的值。不過，在許多情況下，預設值就足夠了。

- `max_logical_replication_workers`
 - `max_sync_workers_per_subscription`
4. 重新啟動多可用區域資料庫叢集，讓參數值生效。如需說明，請參閱[the section called “重新啟動多可用區域資料庫叢集”](#)。
 5. 如 [the section called “使用邏輯複寫槽”](#) 中所述，在多可用區域資料庫叢集的寫入器資料庫執行個體上建立邏輯複寫插槽。此程序要求您指定解碼外掛程式。目前，RDS for PostgreSQL 支援隨附於 PostgreSQL 的 `test_decoding`、`wal2json` 和 `pgoutput` 外掛程式。

插槽會以非同步方式複製到叢集中的每個讀取器資料庫執行個體。

6. 驗證多可用區域資料庫叢集的所有讀取器資料庫執行個體上的插槽狀態。若要這麼做，請檢查所有讀取器資料庫執行個體上的 `pg_replication_slots` 檢視，並確定當應用程式主動取用邏輯變更時，`confirmed_flush_lsn` 狀態正在進行。

下列命令示範如何檢查讀取器資料庫執行個體上的複寫狀態。

```
% psql -h test-postgres-instance-2.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
  slot_name  | slot_type | confirmed_flush_lsn
-----+-----+-----
 logical_slot | logical  | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
  slot_name  | slot_type | confirmed_flush_lsn
-----+-----+-----
 logical_slot | logical  | 32/D8003628
(1 row)

% psql -h test-postgres-instance-3.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
  slot_name  | slot_type | confirmed_flush_lsn
-----+-----+-----
 logical_slot | logical  | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
  slot_name  | slot_type | confirmed_flush_lsn
-----+-----+-----
 logical_slot | logical  | 32/D8003628
(1 row)
```

在完成複寫任務之後，您應該停止複寫程序、捨棄複寫插槽，並關閉邏輯複寫。若要關閉邏輯複寫，請修改您的資料庫叢集參數群組，並將 `rds.logical_replication` 的值設回 0。重新啟動叢集，讓參數變生效。

限制與建議

下列限制和建議適用於將邏輯複寫與執行 PostgreSQL 第 16 版的異地同步備份資料庫叢集搭配使用：

- 您只能使用寫入器資料庫執行個體來建立或卸除邏輯複寫插槽。例如，CREATE SUBSCRIPTION命令必須使用主機連接字串中的叢集寫入器端點。
- 您必須在任何資料表同步處理或重新同步處理期間使用叢集寫入器端點。例如，您可以使用下列命令重新同步處理新增的資料表：

```
Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=writer-endpoint  
Postgres=>ALTER SUBSCRIPTION subscription-name REFRESH PUBLICATION
```

- 您必須等待表格同步處理完成，才能使用讀取器資料庫執行個體進行邏輯複寫。您可以使用目[pg_subscription_rel](#)錄表格來監視表格同步化。srsubstate資料行設定為 ready (r) 時，表格同步處理即完成。
- 我們建議在初始資料表同步處理完成後，針對邏輯複寫連線使用執行個體端點。下列指令可將複寫卸載至其中一個讀取器資料庫執行個體，以減少寫入器資料庫執行個體的負載：

```
Postgres=>ALTER SUBSCRITPION subscription-name CONNECTION host=reader-instance-endpoint
```

您不能一次在多個資料庫執行個體上使用相同的插槽。當兩個或兩個以上的應用程式從叢集中的不同資料庫執行個體複製邏輯變更時，某些變更可能會因叢集容錯移轉或網路問題而遺失。在這些情況下，您可以在主機連接字串中使用執行個體端點進行邏輯複寫。使用相同組態的其他應用程式會顯示下列錯誤訊息：

```
replication slot slot_name is already active for PID x providing immediate feedback.
```

- 使用pglogical擴充功能時，您只能使用叢集寫入器端點。延伸模組具有已知的限制，可在資料表同步處理期間建立未使用的邏輯複寫過時的複寫插槽會保留預寫記錄 (WAL) 檔案，並可能導致磁碟空間問題。

刪除多可用區域資料庫叢集

您可以使用 AWS Management Console、或 RDS API 刪除異地同步備份資料庫叢集。AWS CLI

刪除異地同步備份資料庫叢集所需的時間可能會因下列因素而有所不同：

- 您的備份保留期（也就是說，要刪除多少備份）。
- 刪除了多少數據。
- 是否拍攝最終快照。

異地同步備份資料庫叢集上的刪除保護必須先停用，然後才能刪除它。如需詳細資訊，請參閱 [the section called “刪除資料庫執行個體的先決條件”](#)。您可以修改異地同步備份資料庫叢集來停用刪除保護。如需詳細資訊，請參閱 [the section called “修改多可用區域資料庫叢集”](#)。

主控台

若要刪除多可用區域資料庫叢集

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要刪除的多可用區域資料庫叢集。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。
4. 選擇 Create final snapshot? (是否建立最終快照?)，為多可用區域資料庫叢集建立最終資料庫快照。

如果您建立最終快照，請針對 Final snapshot name (最終快照名稱) 輸入名稱。

5. 選擇 Retain automated backups (保留自動備份) 來保留自動備份。
6. 在方塊中輸入 **delete me**。
7. 選擇 Delete (刪除)。

AWS CLI

若要使用刪除異地同步備份資料庫叢集 AWS CLI，請使用下列選項呼叫 [delete-db-cluster](#) 命令：

- `--db-cluster-identifier`
- `--final-db-snapshot-identifier` 或 `--skip-final-snapshot`

Example 具有最終快照

對於LinuxmacOS、或Unix：

```
aws rds delete-db-cluster \  
  --db-cluster-identifier mymultiadbcluster \  
  --final-db-snapshot-identifier mymultiadbclusterfinalsnapshot
```

在 Windows 中：

```
aws rds delete-db-cluster ^  
  --db-cluster-identifier mymultiadbcluster ^  
  --final-db-snapshot-identifier mymultiadbclusterfinalsnapshot
```

Example 沒有最終快照

對於LinuxmacOS、或Unix：

```
aws rds delete-db-cluster \  
  --db-cluster-identifier mymultiadbcluster \  
  --skip-final-snapshot
```

在 Windows 中：

```
aws rds delete-db-cluster ^  
  --db-cluster-identifier mymultiadbcluster ^  
  --skip-final-snapshot
```

RDS API

若要使用 Amazon RDS API 刪除多可用區域資料庫叢集，請搭配下列參數呼叫 [DeleteDBCluster](#) 操作：

- DBClusterIdentifier
- FinalDBSnapshotIdentifier 或 SkipFinalSnapshot *

異地同步備份資料庫叢集限制

Multi-AZ 資料庫叢集在三個不同的可用區域中有一個寫入器資料庫執行個體和兩個讀取器資料庫。多可用區域資料庫叢集相較於多可用區域部署，可提供高可用性、增加讀取工作負載的容量以及更低的延遲。如需建立多可用區域資料庫叢集的詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。

異地同步備份資料庫叢集適用下列限制。

- 多可用區域資料庫叢集不支援下列功能：
 - IPv6 連線 (雙堆疊模式)
 - 跨區域自動備份
 - IAM 資料庫身分驗證和 Kerberos 驗證
 - 修改連接埠。或者，您可以將多可用區域資料庫叢集還原到某個時間點，並指定不同的連接埠。
 - 選項群組
 - 已刪除叢集的 Point-in-time-recovery (PITR)
 - 將異地同步備份資料庫叢集快照資料匯出到 S3 儲存貯體，或從 S3 儲存貯體還原異地同步備份資料庫叢集快照
 - 透過設定配置的最大儲存空間來自動調度儲存貯體 或者，您可以手動擴展儲存貯體。
 - 停止和啟動異地同步備份資料庫叢集
 - 複製多可用區域資料庫叢集的快照
 - 加密未加密的多可用區域資料庫叢集
- RDS for MySQL 多可用區域資料庫叢集不支援複寫到外部目標資料庫。
- RDS for MySQL 多可用區域資料庫叢集僅支援下列系統預存程序：
 - `mysql.rds_rotate_general_log`
 - `mysql.rds_rotate_slow_log`
 - `mysql.rds_show_configuration`
 - `mysql.rds_set_external_master_with_auto_position`
- 異地同步備 PostgreSQL 資料庫叢集不支援下列擴充功能：和 `aws_s3_pg_transport`
- RDS for PostgreSQL 多可用區域資料庫叢集不支援針對傳出網路存取使用自訂 DNS 伺服器。

使用 Amazon RDS 延長支援

使用 Amazon RDS 延長支援，就可以在 RDS 標準支援結束日期過後，繼續在主要引擎版本上執行您的資料庫，但須另付費用。在 RDS 標準 Support 結束日期，Amazon RDS Amazon 會自動在 RDS 擴展支援中註冊您的資料庫。自動註冊 RDS 延伸 Support 不會變更資料庫引擎，也不會影響資料庫執行個體的正常運作時間或效能。

此付費方案可讓您有更多時間升級至受支援的主要引擎版本。

例如，RDS for MySQL 5.7 版的 RDS 標準支援結束日期為 2024 年 2 月 29 日。不過，您尚未準備好在該日期之前手動升級至 MySQL 8.0 版的 RDS。在這種情況下，Amazon RDS 會在 2024 年 2 月 29 日在 RDS 擴展 Support 中自動註冊您的資料庫，您可以繼續執行 MySQL 版 5.7 版。自 2024 年 3 月 1 日起，Amazon RDS 會自動向您收取 RDS 擴展 Support 的費用。

RDS 延伸 Support 在主要引擎版本的 RDS 標準支援結束日期之後最多可提供。在此之後，如果您尚未將主要引擎版本升級到受支援的版本，Amazon RDS Amazon 將自動升級您的主要引擎版本。我們建議您盡快升級至支援的主要引擎版本。

主題

- [Amazon RDS 擴展 Support 概述](#)
- [使用 Amazon RDS 延伸 Support 建立資料庫執行個體或異](#)
- [在 Amazon RDS 延伸 Support 中檢視資料庫執行個體或異地同步備份資的註冊](#)
- [使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)

Amazon RDS 擴展 Support 概述

在 RDS 標準 Support 結束日期之後，Amazon RDS Amazon 會自動在 RDS 擴展支援中註冊您的資料庫。如果您尚未執行該版本，Amazon RDS 會自動將您的資料庫執行個體升級到 RDS 結束標準支援日期之前發佈的最後一個次要版本。在您的主要引擎版本的 RDS RDS Amazon 不會升級您的次要版本。

您可以使用已達到 RDS 標準支援結束日期的主要引擎版本建立新資料庫。RDS 會自動在 RDS 延伸 Support 中註冊這些新資料庫，並向您收取此供應項目的費用。

如果您在 RDS 標準 Support 援結束日期之前升級至仍然符合 RDS 標準支援的引擎，Amazon RDS Amazon 將不會在 RDS 延伸支援中註冊您的引擎。

如果您嘗試還原與超過 RDS 標準 Support 結束日期的引擎相容的資料庫快照，但尚未註冊 RDS 延伸支援，則 Amazon RDS Amazon 將嘗試升級快照，使其與仍在 RDS 標準支援下的最新引擎版本相

容。如果還原失敗，Amazon RDS Amazon 會使用與快照相容的版本，自動在 RDS 延伸 Support 中註冊您的引擎。

您可以隨時結束 RDS 延伸 Support 的註冊。若要結束註冊，請將每個已註冊的引擎升級至仍在 RDS 標準支援下的較新引擎版本。RDS 延伸 Support 註冊將在您完成升級至仍在 RDS 標準支援下的較新引擎版本當天生效。

主題

- [Amazon RDS 擴展 Support 費用](#)
- [具備 Amazon RDS 延伸 Support 的版本](#)
- [Amazon RDS Amazon 和客戶的責任與 Amazon RDS 擴展 Support](#)

Amazon RDS 擴展 Support 費用

從 RDS 標準支援結束日期的第二天開始，所有註冊參加 RDS 延伸 Support 援的引擎都會產生費用。如需 RDS 標準支援結束日期的[資訊](#)，請參閱[支援的 MySQL 主要版本和發行星事曆](#)。RDS 延伸 Support 費用適用於異地同步備份部署中的待命執行個體。

當您執行下列其中一個動作時，RDS 延伸 Support 的額外費用會自動停止：

- 升級至標準支援涵蓋的引擎版本。
- 刪除執行超過 RDS 標準支援結束日期之主要版本的資料庫。

如果您的目標引擎版本 future 進入 RDS 延伸 Support，則費用將重新啟動。

例如，適用於 PostgreSQL 的 RDS 11 將於 2024 年 3 月 1 日進入延伸 Support，但費用要到 2024 年 4 月 1 日才開始。您可以在 2024 年 4 月 30 日將您的 RDS 11 資料庫升級至適用於 Postgre 的 RDS。您可以在此資料庫執行 PostgreSQL RDS 標準支援日期超過 2025 年 2 月 28 日的 RDS 標準支援結束日期。自 2025 年 3 月 1 日起，您的資料庫將再次產生 RDS 延伸 Support 費用。

如需詳細資訊，請參閱 [Amazon RDS for MySQL 定價](#) 和 [Amazon RDS for PostgreSQL 定價](#)。

避免向 Amazon RDS 擴展 Support 收費

若要這麼做，請使用 AWS CLI 或 RDS API。

在中 AWS CLI，`open-source-rds-extended-support-disabled` 為 `--engine-lifecycle-support` 選項指定。在 RDS API 中，`open-source-rds-extended-support-`

disabled 為 LifeCycleSupport 參數指定。如需詳細資訊，請參閱 [建立資料庫執行個體或異地同步備份資料庫叢集或叢集](#) 或 [將資料庫執行個體或異地同步備份資料庫叢集還原叢集](#)。

具備 Amazon RDS 延伸 Support 的版本

RDS 延伸 Support 僅適用於主要版本。它不適用於次要版本。

RDS 延伸 Support 適用於 MySQL 5.7 和 8.0 版，以及適用於適用於 PostgreSQL 於 RDS 的支援。如需詳細資訊 [支援的 MySQL 主要版本](#)，請參閱 [Amazon RDS for PostgreSQL 發行說明中 Amazon RDS for PostgreSQL 日曆](#)。

Amazon RDS 擴展 Support 版本命名

Amazon RDS Amazon 將在 RDS 擴展 Support 上針對引擎推出具有修正程式和 CVE 修補程式的新次要版本。如需詳細資訊，請參閱 [Amazon RDS 適用於 PostgreSQL 的版本注意事項中的適用於 MySQL 版 RDS 的 Amazon RDS 擴展 Support 版本和 Amazon RDS for PostgreSQL 擴展 Support 更新](#)。

這些次要版本的名稱將採用格式 major.minor-RDS.YYYYMMDD.patch.YYYYMMDD，例如 5.7.44-RDS.20240208.R2.20240210 (適用於 MySQL 版) 或 11.22-RDS.20240208.R2.20240210 (適用 PostgreSQL)。

重大的

對於 MySQL，主要版本號碼既是整數，也是版本號碼的第一個小數部分，例如 8.0。主要版本升級會增加版本號碼的主要部分。例如，從 5.7.44 升級到 8.0.33 是主要版本升級，其中 5.7 和 8.0 是主要版本號碼。

對於 PostgreSQL，主要版本號碼是整數，例如 11。

次要人民幣年月日

對於 MySQL，次要版本號碼是版本號碼的第三部分，例如，在 44-RDS.20240208 中 5.7.44-RDS.20240208。

對於 PostgreSQL，次要版本號碼是版本號碼的第二部分，例如中 22-RDS.20240208 的 11.22-RDS.20240208

日期是 Amazon RDS 創建 Amazon RDS 次要版本的日期。

補丁

修補程式版本是 Amazon RDS 建立 Amazon RDS 次要版本 (例如 R2 中的 5.7.44-RDS.20240208.R2 或 11.22-RDS.20240208.R2) 之後的日期。

Amazon RDS 修補程式版本包含在發行後新增至 Amazon RDS 次要版本的重要錯誤修正。

年月日

該日期是 Amazon RDS 創建修補程序版本的時間，例如，在 5.7.44-RDS.20240208.R2.20240210 或 11.22-RDS.20240208.R2.20240210

Amazon RDS 日期版本是一種安全修補程式，其中包含在發行後新增至次要版本的重要安全修正程式。它不包括任何可能改變引擎行為的修復程序。

Amazon RDS Amazon 和客戶的責任與 Amazon RDS 擴展 Support

下列內容說明 Amazon RDS Amazon 的責任，以及您對 RDS 延伸 Support 的責任。

主題

- [Amazon RDS Amazon 責任](#)
- [您的責任](#)

Amazon RDS Amazon 責任

在 RDS 標準 Support 結束日期之後，Amazon RDS 將為已註冊 RDS 延伸支援的引擎提供修補程式、錯誤修正和升級。這將發生長達 3 年，或直到您停止使用引擎為止，以先發生者為準。

這些修補程式適用於嚴重和高 CVE，如國家弱點資料庫 (NVD) CVE 嚴重性等級所定義。如需詳細資訊，請參閱[漏洞指標](#)。

您的責任

您必須負責套用針對已註冊 RDS 延伸 Support 的資料庫執行個體或異地同步備份資料庫叢集 Aurora DB 叢集提供的修補程式、錯誤修正和升級。Amazon RDS 保留隨時變更、更換或撤銷此類修補程式、錯誤修正和升級的權利。如果需要修補程式來解決安全或重大穩定性問題，Amazon RDS Amazon 保留使用修補程式更新資料庫執行個體或異地同步備份資料的權利，或要求您安裝修補程式。

您也必須負責在 RDS 延伸 Support 結束日期之前將引擎升級至較新的引擎版本。RDS 延伸 Support 結束日期通常是 RDS 標準支援日期社群生命週期 3 年。如需資料庫主要引擎版本的 RDS 延伸 Support 結束日期，請參閱[支援的 MySQL 主要版本](#)和[發行 Amazon RDS for PostgreSQL](#)。

如果您沒有升級引擎，則在 RDS 延伸 Support 結束日期之後，Amazon RDS Amazon 會嘗試將您的引擎升級到 RDS A 標準支援下支援的最新引擎版本。如果升級失敗，則 Amazon RDS Amazon 保留刪除資料庫執行個體或異地同步備份資料庫叢集的權利。但是，在執行此操作之前，Amazon RDS 將保留該引擎中的數據。

使用 Amazon RDS 延伸 Support 建立資料庫執行個體或異

當您建立資料庫執行個體或異地同步備份資料庫叢集時，請在主控台中選取「啟用 RDS 延伸 Support」，或使用 RDS API 中 AWS CLI 或參數中的「延伸支援」選項。

Note

如果您未指定「RDS 延伸 Support」設定，則 RDS 會預設為 RDS 延伸 Support。此預設行為會將資料庫的可用性維持在 RDS 標準支援結束日期之後。

主題

- [RDS 延伸 Support 的注意事項](#)
- [使用 RDS 延伸 Support 建立資料庫執行個體或異](#)

RDS 延伸 Support 的注意事項

在建立資料庫執行個體或異地同步備份資料庫叢集或之前，請考慮下列事項：

- 在 RDS 標準 Support 結束日期過後，您可以防止建立新的資料庫執行個體或新的異地同步備份資料庫叢集或，並避免 RDS 延伸支援費用。若要這麼做，請使用 AWS CLI 或 RDS API。在中 AWS CLI，open-source-rds-extended-support-disabled 為 --engine-lifecycle-support 選項指定。在 RDS API 中，open-source-rds-extended-support-disabled 為 LifecycleSupport 參數指定。如果您指定 open-source-rds-extended-support-disabled RDS 標準支援結束日期已過，則建立資料庫執行個體或異地同步備份資料庫叢集，將永遠失敗。
- RDS 延伸 Support 是在叢集層級設定。叢集的成員在 RDS 主控台、和 EngineLifecycleSupport RDS API 中永遠具有相同 --engine-lifecycle-support 的 RDS 擴充 Support 設定。AWS CLI

如需詳細資訊，請參閱 [MySQL 版本](#) 和 [發行適用於 Amazon RDS 的行事曆](#)。

使用 RDS 延伸 Support 建立資料庫執行個體或異

您可以使用 AWS Management Console、或 RDS API 建立具有 RDS 延伸支援版本的 Aurora 資料庫執行個體或異域叢集。AWS CLI

主控台

當您建體或異地同步備份資料庫叢集時，請在「引擎選項」區段中選取「啟用 RDS 延伸 Support」。

下圖顯示 [啟用 RDS 擴充 Support] 設定：

Enable RDS Extended Support [Info](#)

Amazon RDS Extended Support is a [paid offering](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

AWS CLI

當您執行「`)`」命令時，請指定選項以選取 RDS 擴充 Support。AWS CLI `open-source-rds-extended-support --engine-lifecycle-support` 依預設，此選項設定為 `open-source-rds-extended-support`。

若要防止在 RDS。如此一來，您就可以避免任何相關的 RDS 延伸 Support 費用。

RDS API

當您使用 CreateDBC ([異地同步備份資料庫叢集](#)) Amazon RDS API 作業時，請將參數設定為以選取 RDS 擴充 Support。EngineLifecycleSupport `open-source-rds-extended-support` 根據預設，此參數會設定為 `open-source-rds-extended-support`。

若要防止在 RDS。如此一來，您就可以避免任何相關的 RDS 延伸 Support 費用。

如需詳細資訊，請參閱下列主題：

- 若要建立資料庫執行個體，請遵循 [建立 Amazon RDS 資料庫執行個體](#) 中關於資料庫引擎的指示。
- 若要建立多可用區域資料庫叢集，請遵循 [建立多可用區域資料庫叢集](#) 中的資料庫引擎指示。

在 Amazon RDS 延伸 Support 中檢視資料庫執行個體或異地同步備份資的註冊

您可以使用 AWS Management Console、或 RDS API 在 RDS 延伸 Support 中檢視資料庫執行個體或異的註冊。AWS CLI

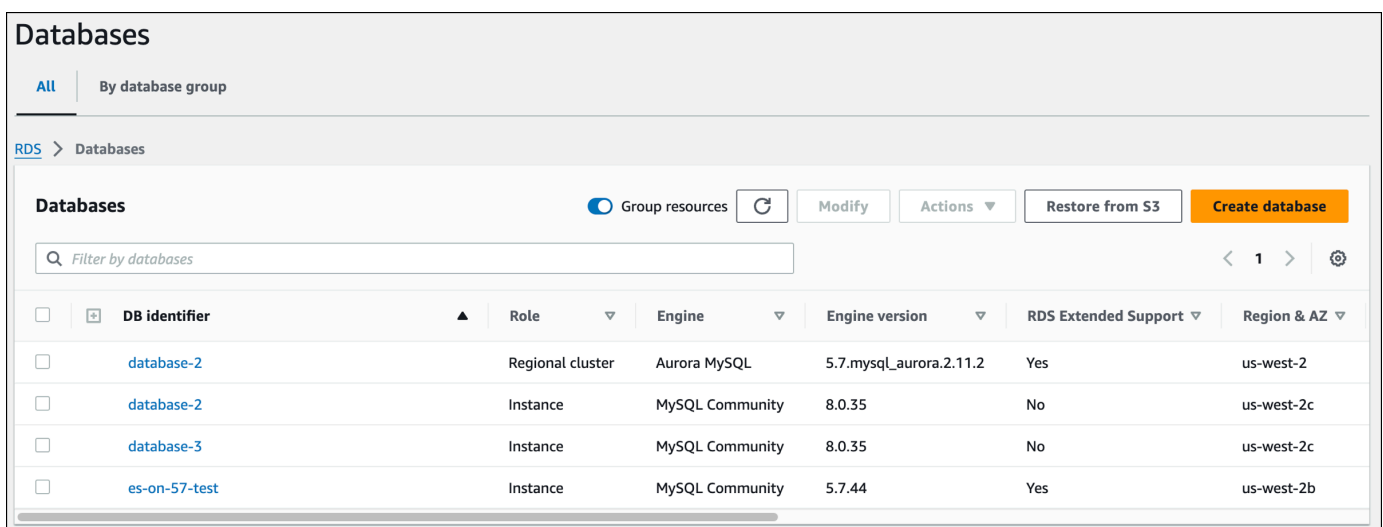
主控台

在 RDS 延伸 Support 中檢視資料庫執行個體或異地同步備份資料庫叢集的註冊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。RDS 延伸 Support 下的值表示 Aurora 資料庫叢集或全域叢集是否已在 RDS 延伸 Support 中註冊資料庫執行個體或異。如果未顯示任何值，則表示您的資料庫無法使用 RDS 延伸 Support。

Tip

如果沒有出現 RDS 延伸 Support 欄，請選擇喜好設定圖示，然後開啟 RDS 延伸 Support。



The screenshot shows the Amazon RDS console interface for the 'Databases' section. It includes a search bar, a table of database instances, and various action buttons like 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database'.

<input type="checkbox"/>	DB identifier	Role	Engine	Engine version	RDS Extended Support	Region & AZ
<input type="checkbox"/>	database-2	Regional cluster	Aurora MySQL	5.7.mysql_aurora.2.11.2	Yes	us-west-2
<input type="checkbox"/>	database-2	Instance	MySQL Community	8.0.35	No	us-west-2c
<input type="checkbox"/>	database-3	Instance	MySQL Community	8.0.35	No	us-west-2c
<input type="checkbox"/>	es-on-57-test	Instance	MySQL Community	5.7.44	Yes	us-west-2b

3. 您也可以每個資料庫的 [組態] 索引標籤上檢視註冊。選擇數據庫標識符下的數據庫。在 [組態] 索引標籤上，查看 [延伸 Support] 下方，查看資料庫是否已註冊。

The screenshot displays the configuration page for an Amazon RDS instance. The instance name is 'es-on-57-test'. The status is 'Available'. The instance class is 'db.t3.micro'. The engine is 'MySQL Community'. The region is 'us-west-2b'. The CPU usage is 3.23% and there are 0 connections. The 'Configuration' tab is selected, showing details such as 'DB instance ID: es-on-57-test', 'Engine version: 5.7.44', 'Instance class: db.t3.micro', 'vCPU: 2', 'RAM: 1 GB', 'Storage type: General Purpose SSD (gp2)', and 'Storage: 25 GiB'. The 'RDS Extended Support' is highlighted with a red box and is set to 'Enabled'.

AWS CLI

[若要使用檢視 RDS 延伸 Support 中資料庫的註冊 AWS CLI，請執行描述-DB 叢集或描述令。](#)

如果資料庫可使用 RDS 延伸 Support，則回應會包含參數 `EngineLifecycleSupport`。此值 `open-source-rds-extended-support` 表示。此值 `open-source-rds-extended-support-disabled` 表示已停用 RDS 延伸 Support 中資料庫執行個體或異地同步備份的註冊。

範例

下列命令會傳回所有資料庫執行個體的資訊：

```
aws rds describe-db-instances
```

下列回應顯示在資料庫執行個體上執行的 PostgreSQL 引擎 `database-1` 已在 RDS 延伸 Support 中註冊：

```
{
  "DBInstanceIdentifier": "database-1",
  "DBInstanceClass": "db.t3.large",
```

```
"Engine": "postgres",
...
"EngineLifecycleSupport": "open-source-rds-extended-support"
}
```

RDS API

如果資料庫可使用 RDS 延伸 Support，則回應會包含參數 `EngineLifecycleSupport`。此值 `open-source-rds-extended-support` 表示。此值 `open-source-rds-extended-support-disabled` 表示已停用 RDS 延伸 Support 中資料庫執行個體或異地同步備份的註冊。

使用 Amazon RDS 延伸 Support 將資料庫執行個體或異

將資料庫執行個體或異地同步備份資料庫叢集還原時，請在主控台中選取「啟用 RDS 延伸 Support」，或使用 RDS API 中 AWS CLI 或參數中的「延伸支援」選項。

Note

如果您未指定 RDS 延伸 Support 設定，RDS 會預設為 RDS 延伸 Support。此預設行為會將資料庫的可用性維持在 RDS 標準支援結束日期之後。

主題

- [RDS 延伸 Support 的注意事項](#)
- [使用 RDS 延伸 Support 將資料庫執行個體或異](#)

RDS 延伸 Support 的注意事項

將資料庫執行個體或異地同步備份資料庫叢集還原之前，請考慮下列事項：

- RDS 標準支援結束日期過後，如果您想要從 Amazon S3 還原資料庫執行個體或異，則只能使用 AWS CLI 或 RDS API 來執行此操作。[使用從 s3 還原-DB 叢集 AWS CLI 命令中的 `--engine-lifecycle-support` 選項，或復原 S3 RDS API 作業中的 `EngineLifecycleSupport` 參數。ClusterFrom](#)
- 如果您想要防止 RDS 還原您的資料庫至 RDS 擴充 Support 版本，請 `open-source-rds-extended-support-disabled` 在 AWS CLI 或 RDS API 中指定。如此一來，您就可以避免任何相關的 RDS 延伸 Support 費用。

如果您指定此設定，Amazon RDS Amazon 會自動將還原的資料庫升級到較新、受支援的主要版本。如果升級未能在升級前檢查，Amazon RDS Amazon 將安全地復原至 RDS 延伸 Support 引擎版本。此資料庫將維持在 RDS 延伸 Support 模式，而 Amazon RDS Amazon 會向您收取 RDS 延伸 Support 的費用，直到您手動升級資料庫為止。

例如，如果您在不使用 RDS 擴展 Support 的情況下還原 MySQL 5.7 快照，Amazon RDS 將嘗試自動將您的資料庫升級到 MySQL 8.0。如果此升級因為您需要解決的問題而失敗，Amazon RDS 會將資料庫復原至 MySQL 5.7。Amazon RDS 會將資料庫保留在 RDS 擴充 Support 中，直到您可以修正問題為止。例如，升級可能會因為儲存空間不足而失敗。修正問題之後，您必須啟動升級。在第一次嘗試升級資料庫之後，Amazon RDS 不會再次嘗試升級資料庫。

- RDS 延伸 Support 是在叢集層級設定。叢集的成員在 RDS 主控台、和 `EngineLifecycleSupport` RDS API 中永遠具有相同 `--engine-lifecycle-support` 的 RDS 擴充 Support 設定。AWS CLI

如需詳細資訊，請參閱 [MySQL 版本](#) 和 [發行適用於 Amazon RDS 的行事曆](#)。

使用 RDS 延伸 Support 將資料庫執行個體或異

您可以使用 AWS Management Console、或 RDS API 將資料庫執行個體或異。AWS CLI

主控台

當您將 體或異地同步備份資料庫叢集時，請在「引擎選項」區段中選取「啟用 RDS 延伸 Support」。

下圖顯示 [啟用 RDS 擴充 Support] 設定：

Enable RDS Extended Support [Info](#)

Amazon RDS Extended Support is a [paid offering](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

AWS CLI

當您執行從快照復命命令時，請指定選項以選取 RDS 擴充 Support。AWS CLI `open-source-rds-extended-support --engine-lifecycle-support`

如果您想要避免與 RDS 延伸 Support 相關的費用，請將選 `--engine-lifecycle-support` 項設定為 `open-source-rds-extended-support-disabled`。依預設，此選項設定為 `open-source-rds-extended-support`。

您也可以使用下列指 AWS CLI 令來指定此值：

- [restore-db-cluster-from-s3](#)
- [restore-db-cluster-to-point-in-time](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

RDS API

當您使用恢復快照或還原快ClusterFrom照 Amazon RDS API 作業時，請將參數設定為以選取 RDS 擴展 S support。EngineLifecycleSupport open-source-rds-extended-support

如果您想要避免與 RDS 延伸 Support 相關的費用，請將EngineLifecycleSupport參數設定為open-source-rds-extended-support-disabled。根據預設，此參數會設定為 open-source-rds-extended-support。

您也可以使用下列 RDS API 作業來指定此值：

- [恢復 B S3 ClusterFrom](#)
- [恢復時ClusterToPointIn間](#)
- [恢復 B S3 InstanceFrom](#)
- [恢復時InstanceToPointIn間](#)

如需還原資料庫執行個體或異地同步備份資料庫叢集的詳細資訊，請遵循中適用於資料庫引擎的[從資料庫快照還原](#)指示。

使用 Amazon RDS 藍/綠部署進行資料庫更新

藍/綠部署會將生產資料庫環境複製到個別已同步的預備環境。透過使用 Amazon RDS 藍/綠部署，您可以在預備環境中對資料庫進行變更，而不會影響生產環境。例如，您可以升級主要或次要資料庫引擎版本、變更資料庫參數，或在預備環境中進行結構描述變更。準備就緒後，您可以將測試環境提升為新的生產資料庫環境，停機時間通常不到一分鐘。

Note

目前，藍色/綠色部署僅支援 MariaDB 用於 RDS、MySQL 版 RDS 和適用於 PostgreSQL 的 RDS。如需了解 Amazon Aurora 可用性，請參閱《Amazon Aurora 使用者指南》中的[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

主題

- [Amazon RDS 藍/綠部署概觀](#)
- [建立藍/綠部署](#)
- [檢視藍/綠部署](#)
- [切換藍/綠部署](#)
- [刪除藍/綠部署](#)

Amazon RDS 藍/綠部署概觀

透過使用 Amazon RDS 藍/綠部署，您可以進行並測試資料庫變更，然後在生產環境中實作這些變更。藍/綠部署會建立一個複製生產環境的預備環境。在藍/綠部署中，藍色環境是目前的生產環境。綠色環境是預備環境。測試環境會使用邏輯複寫與目前的生產環境保持同步。

您可以對綠色環境中的 RDS 資料庫執行個體進行變更，而不會影響生產工作負載。例如，您可以升級主要或次要資料庫引擎版本、升級基礎檔案系統組態或在預備環境中變更資料庫參數。您可以在綠色環境中徹底測試變更。備妥後，您可以切換環境，將綠色環境提升為新的生產環境。切換通常只需不到一分鐘的時間，不會遺失資料，也不需要變更應用程式。

因為綠色環境是生產環境拓撲的副本，所以綠色環境包含資料庫執行個體所使用的功能。這些功能包含僅供讀取複本、儲存組態、資料庫快照、自動備份、Performance Insights 和增強型監控。如果藍色資料庫執行個體是多可用區域資料庫執行個體部署，則綠色資料庫執行個體也是多可用區域資料庫執行個體部署。

Note

目前，僅 RDS for MariaDB、RDS for MySQL 和 RDS for PostgreSQL 支援藍/綠部署。如需 Amazon Aurora 的可用性，請參閱 [Amazon Aurora 使用者指南中的使用 Amazon RDS 藍色/綠色部署進行資料庫更新](#)。

主題

- [區域和版本可用性](#)
- [使用 Amazon RDS 藍/綠部署的優點](#)
- [藍/綠部署的工作流程](#)
- [授予藍/綠部署操作的存取權](#)
- [藍/綠部署考量](#)
- [藍/綠部署的最佳實務](#)
- [藍/綠部署的限制](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需詳細資訊，請參閱 [the section called “藍/綠部署”](#)。

使用 Amazon RDS 藍/綠部署的優點

透過使用 Amazon RDS 藍/綠部署，您可以隨時掌握最新的安全修補程式、改善資料庫效能，以及採用較新的資料庫功能，其停機時間短暫且可預測。藍/綠部署可減少資料庫更新 (例如主要或次要引擎版本升級) 的風險和停機時間。

藍/綠部署提供下列優點：

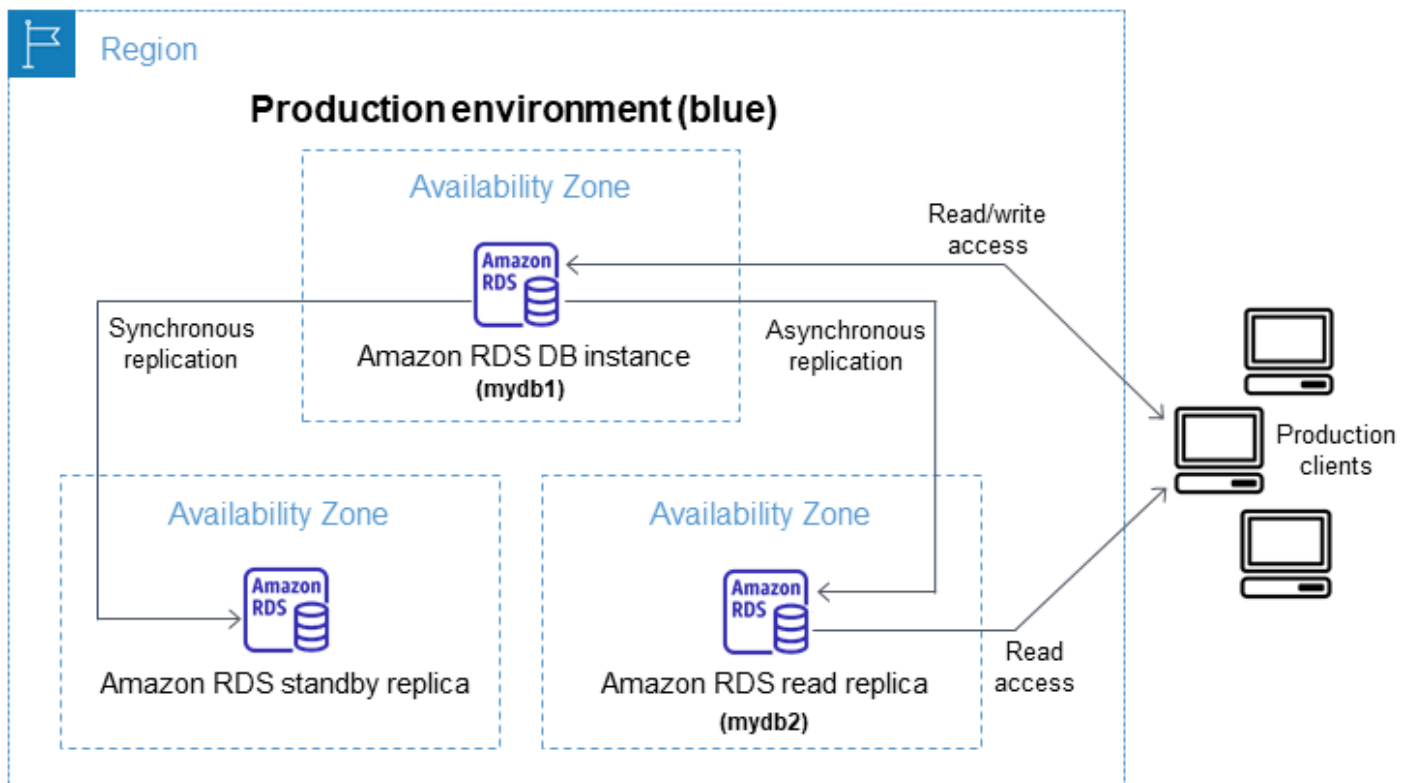
- 輕鬆建立生產就緒的預備環境。
- 自動將資料庫變更從生產環境複製到預備環境。
- 在安全預備環境中測試資料庫變更，而不會影響生產環境。
- 隨時掌握最新的資料庫修補程式和系統更新。
- 實作和測試較新的資料庫功能。
- 切換預備環境以成為新的生產環境，而無需對應用程式進行變更。
- 透過使用內建的防護機制安全切換。
- 消除切換期間的資料遺失情況。
- 通常在一分鐘內快速切換，取決於您的工作負載。

藍/綠部署的工作流程

當您使用藍/綠部署進行資料庫更新時，請完成下列主要步驟。

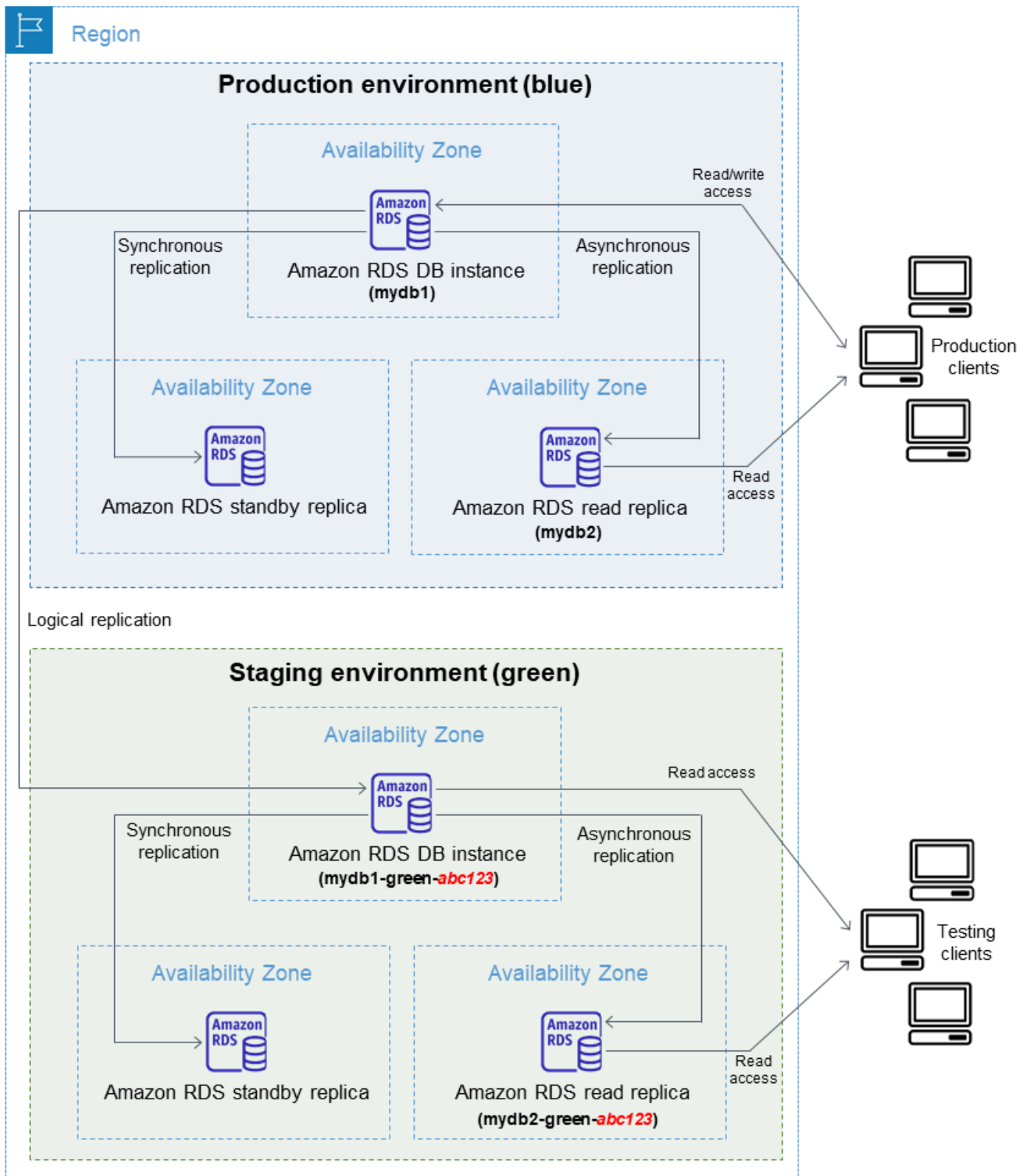
1. 識別需要更新的生產環境。

例如，此影像中的生產環境具有多可用區域資料庫執行個體部署 (mydb1) 和僅供讀取複本 (mydb2)。



2. 建立藍/綠部署。如需說明，請參閱[建立藍/綠部署](#)。

下圖顯示一個範例，說明如何從步驟 1 開始生產環境的藍/綠部署範例。在建立藍/綠部署時，RDS 會複製主要資料庫執行個體的完整拓撲和組態，以建立綠色環境。複製的資料庫執行個體名稱會附加 `-green-random-characters`。影像中的預備環境包含多可用區域資料庫執行個體部署 (`mydb1-green-abc123`) 和僅供讀取複本 (`mydb2-green-abc123`)。



建立藍/綠部署時，您可以升級資料庫引擎版本，並針對綠色環境中的資料庫執行個體指定不同的資料庫參數群組。RDS 也會設定從藍色環境中主要資料庫執行個體到綠色環境中主要資料庫執行個體的邏輯複寫。

在您建立藍/綠部署之後，綠色環境中的資料庫執行個體會預設為唯讀。

3. 如有需要，對預備環境進行其他變更。

例如，您可能會對資料庫進行結構描述變更，或變更綠色環境中一或多個資料庫執行個體所使用的資料庫執行個體類別。

如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

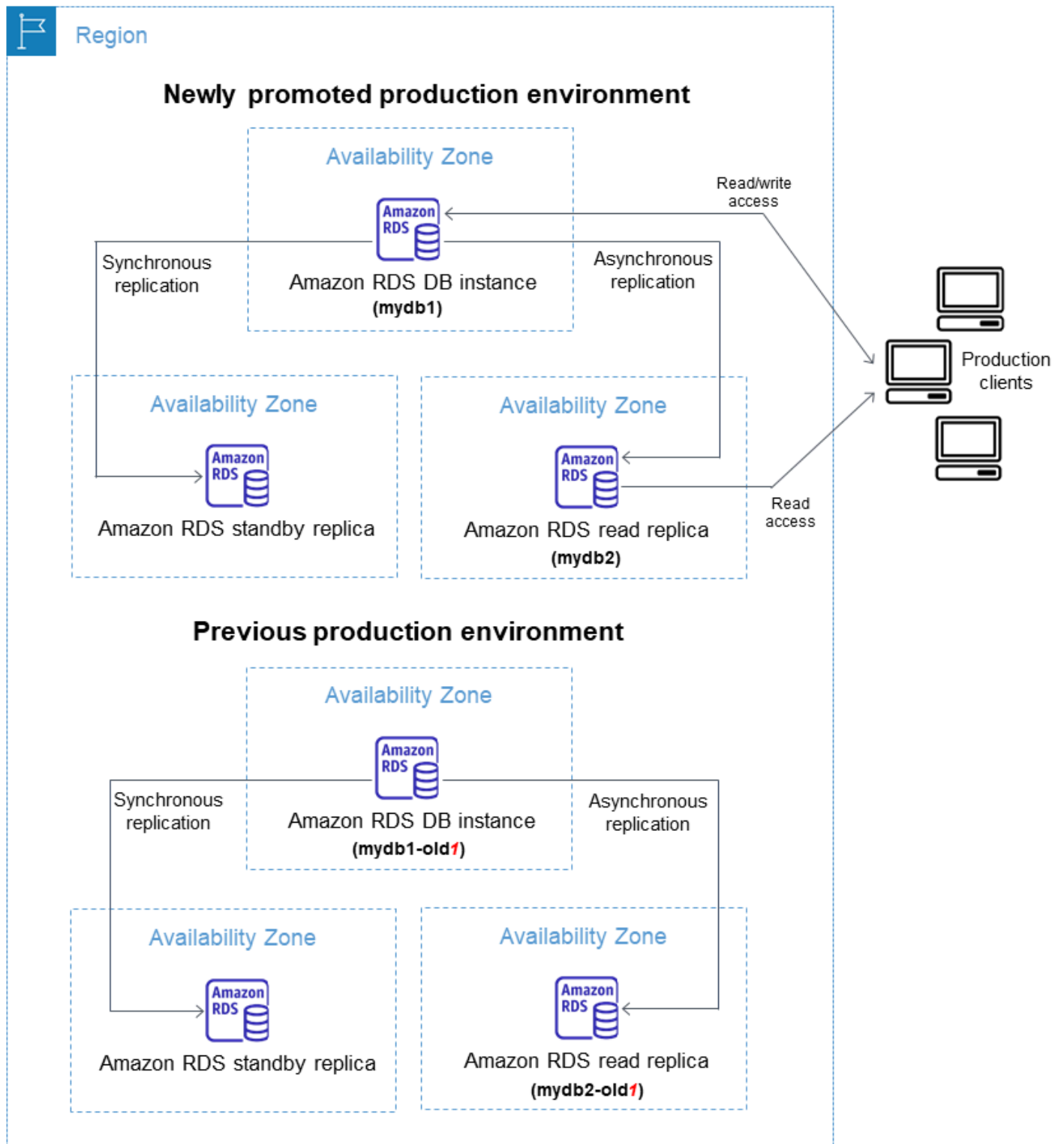
4. 測試您的預備環境。

在測試期間，建議您將綠色環境中的資料庫保持唯讀狀態。請小心在綠色環境中啟用寫入作業，因為這些作業可能會造成複寫衝突。它們也可能會在切換後於生產資料庫中產生非預期的資料。若要啟用 RDS for MySQL 的寫入作業，請將`read_only`參數設定為`0`，然後重新啟動資料庫執行個體。對於適用於 PostgreSQL 的 RDS，請`off`在工作階段層級將`default_transaction_read_only`參數設定為。

5. 備妥後，請切換以將預備環境提升為新的生產環境。如需說明，請參閱[切換藍/綠部署](#)。

切換會產生停機時間。停機時間通常不到一分鐘，但可能更長，取決於您的工作負載。

下圖顯示切換後的資料庫執行個體。



切換後，綠色環境中的資料庫執行個體會成為新的生產資料庫執行個體。目前生產環境中的名稱和端點會指派給新提升的生產環境，不需要對您的應用程式進行任何變更。因此，您的生產流量現在會流向新的生產環境。先前藍色環境中的資料庫執行個體會重新命名，方法是將 `-oldn` 附加至目前

名稱 (其中 n 是數字)。例如，假設藍色環境中資料庫執行個體的名稱為 mydb1。切換後，資料庫執行個體名稱可能是 mydb1-old1。

在影像的範例中，切換期間會發生下列變更：

- 名為 mydb1-green-abc123 的綠色環境多可用區域執行個體部署會成為名為 mydb1 的生產多可用區域執行個體部署。
 - 名為 mydb2-green-abc123 的綠色環境僅供讀取複本會成為生產僅供讀取複本 mydb2。
 - 名為 mydb1 的藍色環境多可用區域資料庫執行個體部署會成為 mydb1-old1。
 - 名為 mydb2 的藍色環境僅供讀取複本會成為 mydb2-old1。
6. 如果不再需要藍/綠部署，您可以將其刪除。如需說明，請參閱[刪除藍/綠部署](#)。

切換後，不會刪除先前的生產環境，以便您可以在必要時使用它進行迴歸測試。

授予藍/綠部署操作的存取權

使用者必須具有必要的許可，才能執行與藍/綠部署相關的操作。您可以建立 IAM 政策，許可使用者和角色對其需要的指定資源執行特定 API 操作。然後，您可以將這些政策連線至需要這些許可的 IAM 許可集或角色。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。

建立藍/綠部署的使用者必須具有執行下列 RDS 操作的許可：

- rds:AddTagsToResource
- rds:CreateDBInstanceReadReplica

切換藍/綠部署的使用者必須具有執行下列 RDS 操作的許可：

- rds:ModifyDBInstance
- rds:PromoteReadReplica

刪除藍/綠部署的使用者必須具有執行下列 RDS 操作的許可：

- rds>DeleteDBInstance

Amazon RDS 會代表您在測試環境中佈建和修改資源。這些資源包括使用內部定義命名慣例的資料庫執行個體。因此，附加的 IAM 政策不能包含部分資源名稱模式，例如 my-db-prefix-*. 僅支援萬用

字元 (*)。一般而言，我們建議使用資源標籤和其他支援的屬性來控制對這些資源的存取，而非萬用字元。如需詳細資訊，請參閱 [Amazon RDS 的動作、資源和條件金鑰](#)。

藍/綠部署考量

Amazon RDS 會使用每個資源的 `DbiResourceId` 追蹤藍/綠部署中的資源。此資源 ID 是資源的 AWS 區域唯一且不可變的識別碼。

資源 ID 與資料庫執行個體 ID 不同：

Instance


Configuration

DB instance ID
database-1

Engine version
8.0.28

DB name
-

License model
General Public License

Option groups
default:mysql-8-0  In sync

Amazon Resource Name (ARN)
arn:aws:rds:us-east-1:
:db:database-1

Resource ID
db-ZY2YAOOH4LWCKBYXVK6V7LI6VQ

當您切換藍/綠部署時，資源的名稱 (執行個體 ID) 會變更，但每個資源都保留相同的資源 ID。例如，資料庫執行個體識別符可能是藍色環境中的 mydb。切換後，相同的資料庫執行個體可能重新命名為 mydb-old1。不過，資料庫執行個體的資源 ID 在切換期間不會變更。因此，當綠色資源提升為新的生產資源時，其資源 ID 與先前位於生產環境中的藍色資源 ID 不符。

在切換藍/綠部署之後，請考慮將資源 ID 更新為新提升之生產資源的 ID，以取得與生產資源搭配使用的整合功能和服務。具體來說，請考慮下列更新：

- 如果您使用 RDS API 和資源 ID 執行篩選，請在切換後調整用於篩選的資源 ID。
- 如果您使 CloudTrail 用稽核資源，請調整的取用者，CloudTrail 以在切換後追蹤新的資源 ID。如需詳細資訊，請參閱 [在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)。
- 如果您使用 Performance Insights API，請在切換後調整 API 呼叫中的資源 ID。如需詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。

您可以在切換後監控具有相同名稱的資料庫，但其不包含切換前的資料。

- 如果您在 IAM 政策中使用資源 ID，請務必在必要時新增新提升之資源的資源 ID。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。
- 如果您的執行個體有關聯的 IAM 角色，請務必在切換後重新關聯這些角色。附加的角色不會自動複製到綠色環境。
- 如果您使用 [IAM 資料庫身份驗證](#) 對資料庫執行個體進行身分驗證，請確定用於資料庫存取的 IAM 政策同時具有政策 Resource 元素下方列出的藍色和綠色資料庫。這是必要的，以便在切換後連線到綠色資料庫。如需詳細資訊，請參閱 [the section called “建立並使用 IAM 政策進行 IAM 資料庫存取”](#)。
- 如果您使 AWS Backup 用管理藍/綠部署中資源的自動備份，請在切換 AWS Backup 後調整使用的資源 ID。如需詳細資訊，請參閱 [用 AWS Backup 於管理自動備份](#)。
- 如果您想要還原屬於藍/綠部署之資料庫執行個體的手動或自動資料庫快照，請確定透過檢查取得快照的時間來還原正確的資料庫快照。如需詳細資訊，請參閱 [從資料庫快照還原](#)。
- 如果您想要描述先前的藍色環境資料庫執行個體自動備份，或將其還原到某個時間點，請將資源 ID 用於操作。

因為資料庫執行個體的名稱會在切換期間變更，所以您無法將其先前的名稱用於 DescribeDBInstanceAutomatedBackups 或 RestoreDBInstanceToPointInTime 操作。

如需詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

- 若您在藍/綠部署的綠色環境中將僅供讀取複本新增至資料庫執行個體，則在切換時，新的僅供讀取複本不會取代藍色環境中的僅供讀取複本。不過，新的僅供讀取複本會在切換後保留在新的生產環境中。
- 在藍/綠部署的綠色環境中刪除資料庫執行個體時，您無法建立新的資料庫執行個體，以在藍/綠部署中取代該執行個體。

如果您使用與所刪除資料庫執行個體相同的名稱和 Amazon Resource Name (ARN) 建立的新資料庫執行個體，則它具有不同的 `DbiResourceId`，因此不屬於綠色環境。

如果您刪除綠色環境中的資料庫執行個體，則會產生下列行為：

- 如果藍色環境中存在名稱相同的資料庫執行個體，則它不會切換至綠色環境中的資料庫執行個體。此資料庫執行個體不會透過將 `-oldn` 新增至資料庫執行個體名稱來重新命名。
- 指向藍色環境中資料庫執行個體的任何應用程式都會在切換後繼續使用相同的資料庫執行個體。

相同的行為適用於資料庫執行個體和僅供讀取複本。

藍/綠部署的最佳實務

下列是藍/綠部署的最佳實務：

一般最佳實務

- 切換前，請在綠色環境中徹底測試資料庫執行個體。
- 在綠色環境中將您的資料庫保留唯讀狀態。建議您在綠色環境上小心啟用寫入操作，因為它們可能會在綠色環境中造成複寫衝突。它們也可能會在切換後於生產資料庫中產生非預期的資料。
- 使用藍/綠部署實作結構描述變更時，請僅進行複寫相容的變更。

例如，您可以在資料表結尾新增資料欄，而不會中斷從藍色部署到綠色部署的複寫。不過，結構描述變更 (例如重新命名資料欄或重新命名資料表) 會中斷綠色部署的複寫。

如需有關複寫相容變更的詳細資訊，請參閱 MySQL 文件中的 [在來源和複本上使用不同的資料表定義進行複寫](#)，以及 PostgreSQL 邏輯複寫文件中的 [限制](#)。

- 建立藍/綠部署之後，如有必要，請處理延遲載入。在切換之前，請確定資料載入完成。如需詳細資訊，請參閱 [在建立藍/綠部署時處理延遲載入](#)。
- 當您切換藍/綠部署時，請遵循切換最佳實務。如需詳細資訊，請參閱 [the section called “切換最佳實務”](#)。

RDS for MySQL 最佳實務

- 避免使用未針對複寫最佳化的非交易式儲存引擎，例如 MyISAM。
- 針對二進位日誌複寫最佳化僅供讀取複本，

例如，如果您的資料庫引擎版本支援它，請考慮在部署藍/綠部署之前，於生產環境中使用 GTID 複寫、平行複寫和免當機複寫。在切換藍/綠部署之前，這些選項會提升資料的一致性和耐久性。如需僅供讀取複本之 GTID 複寫的詳細資訊，請參閱 [使用 GTID 式複寫](#)。

RDS for PostgreSQL 最佳實務

- 如果您的資料庫有足夠的可釋放記憶體，請在藍色環境中增加 `logical_decoding_work_mem` DB 參數的值。這樣做允許減少磁碟上的解碼，並改用記憶體。您可以使用 `FreeableMemory` CloudWatch 指標監視可用內存。如需詳細資訊，請參閱 [the section called “Amazon RDS 的 CloudWatch Amazon 實例級指標”](#)。
- 建立藍/綠部署之前，請先將所有 PostgreSQL 延伸模組更新至最新版本。如需詳細資訊，請參閱 [the section called “升級 PostgreSQL 延伸”](#)。
- 如果您使用 `aws_s3` 延伸模組，請務必在建立綠色環境之後透過 IAM 角色將綠色資料庫執行個體存取權授與 Amazon S3。這允許匯入和匯出命令在轉換之後繼續運作。如需說明，請參閱 [the section called “設定對 Amazon S3 儲存貯體的存取權”](#)。
- 如果您為綠色環境指定較高的引擎版本，請在所有資料庫上執行 ANALYZE 作業以重新整理資料 `pg_statistic` 表。主要版本升級期間不會傳輸最佳化處理程式統計資料，因此您必須重新產生所有統計資料，以避免效能 如需主要版本升級期間的其他最佳作法，請參閱 [the section called “如何執行主要版本升級”](#)。
- 避免將觸發器設定為 `ENABLE REPLICA` 或 `ENABLE ALWAYS` 是在來源上使用觸發器來操作資料。否則，複製系統會傳播變更並執行觸發程序，從而導致重複。
- 長時間執行的交易可能會導致重大複本延遲 若要減少複本延遲，請考慮執行下列動作：
 - 減少可以延遲到綠色環境趕上藍色環境之後的長時間運行的事務。
 - 在建立藍/綠部署之前，在忙碌的資料表上啟動手動真空凍結作業。
 - 對於 PostgreSQL 版本 12 及更新版本，請停用大型或忙碌資料表上的 `index_cleanup` 參數，以提高藍色資料庫的正常維護速率。如需詳細資訊，請參閱 [the section called “盡快清空資料表”](#)。
- 緩慢的複寫會導致寄件者和接收者經常重新啟動，進而延遲同步處理。若要確保它們保持作用中狀態，請在藍色環境 0 中將 `wal_sender_timeout` 參數設定為，並在綠色環境 0 中將 `wal_receiver_timeout` 參數設定為來停用逾時。
- 若要防止從藍色環境中移除預寫記錄 (WAL) 區段，請針對 PostgreSQL 版本 13 及更低版本，將 `wal_keep_segments` 參數設定為 15625。對於版本 14 及更高版本，如果有足夠的可用存儲空間，請將 `wal_keep_size` 參數設置為 1 TiB。

藍/綠部署的限制

下列限制適用於藍/綠部署。

主題

- [藍/綠部署的一般限制](#)
- [藍/綠部署的PostgreSQL 擴充限制](#)
- [藍/綠部署中變更的限制](#)
- [藍/綠部署的 PostgreSQL 邏輯複寫限制](#)

藍/綠部署的一般限制

下列一般限制適用於藍/綠部署：

- MySQL 8.0.11 版至 8.0.13 版具有一個[社群錯誤](#)，其會阻止這些版本支援藍/綠部署。
- 支援下列版本的 RDS for PostgreSQL 做為升級來源和目標版本：11.21 及更高版本、12.16 及更高版本、13.12 及更高版本、14.9 及更高版本，以及 15.4 及更高版本。對於較低版本，您可以對支援的版本執行次要版本升級。
- 藍/綠部署不支援使用管理主要使用者密碼。AWS Secrets Manager
- 如果藍色資料庫上已啟用專用記錄磁碟區 (DLV)，則必須在所有資料庫執行個體 (包括僅供讀取複本) 上啟用該磁碟區。
- 對於 RDS for PostgreSQL，。
- 對於 PostgreSQL RDS，藍色環境資料庫執行個體不能是自我管理的邏輯來源 (發行者) 或複本 (訂閱者)。對於適用於 MySQL RDS，藍色環境資料庫執行個體不能是外部備份記錄複本。
- 在切換期間，藍色和綠色環境無法與 Amazon Redshift 進行零 ETL 整合。您必須先刪除整合再進行切換，然後重新建立整合。
- 建立藍/綠部署時，必須在綠色環境上停用事件排程器 (event_scheduler 參數)。這樣可以防止在綠色環境中產生事件並導致不一致。
- 藍色/綠色部署不支援適用於 MySQL 的 AWS JDBC 驅動程式。如需詳細資訊，請參閱的[已知限制](#) GitHub。
- 下列功能不支援藍/綠部署：
 - Amazon RDS Proxy
 - 階層式僅供讀取複本
 - 跨區域僅供讀取複本

- AWS CloudFormation
- Multi-AZ 資料庫叢集部署

支援藍/綠部署進行多可用區域資料庫執行個體部署。如需異地同步備份部署的詳細資訊，請參閱[設定及管理多可用區部署](#)。

藍/綠部署的PostgreSQL 擴充限制

下列限制適用於 PostgreSQL 延伸模組：

- 建立藍/綠部署時，必須在藍色環境上停用 `pg_partman` 延伸模組。延伸模組會執行 DDL 作業 (例如 `CREATE TABLE`)，其會中斷從藍色環境到綠色環境的邏輯複寫。
- 建立藍/綠部署之後，所有綠色資料庫上的 `pg_cron` 延伸模組必須保持停用狀態。延伸模組具有以超級使用者身分執行的背景工作者，且會略過綠色環境的唯讀設定，這可能會造成複寫衝突。
- 如果藍色資料庫執行個體設定為外部資料包裝函式 (FDW) 延伸模組的外部伺服器，您必須使用執行個體端點名稱，而非 IP 位址。這可讓組態在轉換之後保持運作狀態。
- 建立藍/綠部署時，必須在藍色環境上停用 `pglogical` 和 `pg_active` 延伸模組。將綠色環境提升為新的生產環境後，您可以再次啟用擴充功能。此外，藍色資料庫不能是外部執行個體的邏輯訂閱者。
- 如果您使用 `pgAudit` 擴充功能，它必須保留在藍色和綠色 `shared_preload_libraries` 資料庫執行個體的自訂資料庫參數群組上的共用程式庫 () 中。如需詳細資訊，請參閱 [the section called “設定 pgAudit 擴充功能”](#)。

藍/綠部署中變更的限制

下列是藍/綠部署中變更的限制：

- 您無法將未加密的資料庫執行個體變更為加密的資料庫執行個體。
- 您無法將加密的資料庫執行個體變更為未加密的資料庫執行個體。
- 您無法將藍色環境資料庫執行個體變更為高於其對應綠色環境資料庫執行個體的引擎版本。
- 藍色環境和綠色環境中的資源必須在同一 AWS 帳戶中。
- 對於 RDS for MySQL，如果來源資料庫與自訂選項群組相關聯，則您無法在建立藍/綠部署時指定主要版本升級。

在此情況下，您可以建立藍/綠部署，而無需指定主要版本升級。然後，您可以在綠色環境中升級資料庫。如需詳細資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

藍/綠部署的 PostgreSQL 邏輯複寫限制

藍/綠部署會使用邏輯複寫，讓預備環境與生產環境保持同步。PostgreSQL 具有某些與邏輯複寫相關的
限制，其會在針對 RDS for PostgreSQL 資料庫執行個體建立藍/綠部署時轉換為限制。

下表描述適用於 RDS for PostgreSQL 藍/綠部署的邏輯複寫限制。

限制	說明
資料定義語言 (DDL) 陳述式 (例如 CREATE TABLE 和 CREATE SCHEMA) 不會從藍色環境複寫到綠色環境。	<p>如果 Amazon RDS 在藍色環境中偵測到 DDL 變更，則您的綠色資料庫會進入複寫降級狀態。</p> <p>您會收到一個事件，通知您藍色環境中的 DDL 變更無法複寫到綠色環境。您必須刪除藍/綠部署和所有綠色資料庫，然後重新建立該部署。否則，您將無法轉換藍/綠部署。</p>
序列物件上的 NEXTVAL 作業不會在藍色環境與綠色環境之間同步。	<p>在轉換期間，Amazon RDS 會在綠色環境中遞增序列值，以符合藍色環境中的序列值。如果您有數千個序列，這可能會延遲轉換。</p>
藍色環境中的大型物件建立或修改並不會複寫到綠色環境。	<p>如果 Amazon RDS 在藍色環境中偵測到 <code>pg_largeobject</code> 系統資料表中儲存的大型物件建立或修改，則您的綠色資料庫將進入複寫降級狀態。</p> <p>RDS 會產生一個事件，通知您藍色環境中的大型物件變更無法複寫到綠色環境。您必須刪除藍/綠部署和所有綠色資料庫，然後重新建立該部署。否則，您將無法轉換藍/綠部署。</p>
具體化視觀表不會在綠色環境中自動重新整理。	<p>在藍色環境中重新整理具體化視觀表不會在綠色環境中重新整理這些視觀表。轉換之後，您可以排定具體化視觀表的重新整理。</p>
不允許在沒有主索引鍵的資料表上執行 UPDATE 和 DELETE 作業。	<p>在建立藍/綠部署之前，請確定資料庫執行個體中的所有資料表都有主索引鍵。</p>

如需詳細資訊，請參閱 PostgreSQL 邏輯複寫文件中的[限制](#)。

建立藍/綠部署

建立藍/綠部署時，您可以指定要在部署中複製的來源資料庫執行個體。您選擇的資料庫執行個體是生產資料庫執行個體，而且其會成為藍色環境中的主要資料庫執行個體。此資料庫執行個體會複製到綠色環境，而且 RDS 會設定從藍色環境中的資料庫執行個體複寫到綠色環境中的資料庫執行個體。

RDS 將藍色環境的拓撲複製到預備區域，以及複製其設定的功能。當藍色資料庫執行個體具有僅供讀取複本時，僅供讀取複本會複製為部署中綠色資料庫執行個體的僅供讀取複本。如果藍色資料庫執行個體是多可用區域資料庫執行個體部署，則會建立綠色資料庫執行個體，作為多可用區域資料庫執行個體部署。

主題

- [準備進行藍/綠部署](#)
- [在建立藍/綠部署時指定變更](#)
- [在建立藍/綠部署時處理延遲載入](#)
- [建立藍/綠部署](#)
- [建立藍/綠部署的設定](#)

準備進行藍/綠部署

根據執行個體執行的引擎而定，在建立藍/綠部署之前，您必須執行某些步驟。

主題

- [準備適用於 MySQL 的 RDS 資料庫執行個體以進行藍色/綠色部署](#)
- [為藍/綠部署準備 RDS for PostgreSQL 資料庫執行個體](#)

準備適用於 MySQL 的 RDS 資料庫執行個體以進行藍色/綠色部署

在為 RDS for MySQL 資料庫執行個體建立藍色/綠色部署之前，必須先啟用自動備份。如需說明，請參閱[the section called “啟用自動備份”](#)。

為藍/綠部署準備 RDS for PostgreSQL 資料庫執行個體

在針對 RDS for PostgreSQL 資料庫執行個體建立藍色/綠色部署之前，請務必執行下列動作：

- 將執行個體與已開啟邏輯複寫 (rds.logical_replication) 的自訂資料庫參數群組建立關聯。從藍色環境複寫到綠色環境時，需要邏輯複寫。如需說明，請參閱[the section called “修改資料庫參數群組中的參數”](#)。

由於藍/綠部署每個資料庫至少需要一個背景 Worker，因此請務必根據您的工作負載調整下列組態設定。如需調整每個設定的指示，請參閱 PostgreSQL 文件中的[組態設定](#)。

- max_replication_slots
- max_wal_senders
- max_logical_replication_workers
- max_worker_processes

啟用邏輯複寫並設定所有組態選項後，請務必重新啟動資料庫執行個體，以讓您的變更生效。藍/綠部署要求資料庫執行個體與資料庫參數群組同步，否則建立作業將會失敗。如需詳細資訊，請參閱[the section called “將資料庫執行個體重新開機”](#)。

- 請確認資料庫執行個體所執行的 RDS for PostgreSQL 版本與 RDS 藍/綠部署相容。如需相容版本的清單，請參閱[the section called “藍/綠部署”](#)。
- 確認資料庫執行個體不是外部複寫的來源或目標。如需詳細資訊，請參閱 [the section called “一般限制”](#)。
- 確定資料庫執行個體中的所有資料表都有主索引鍵。PostgreSQL 邏輯複寫不允許對沒有主索引鍵的資料表執行 UPDATE 或 DELETE 操作。
- 如果您使用的是觸發器，請確定它們不會干擾名稱以「rds」開頭的pg_catalog.pg_publicationpg_catalog.pg_subscription、和pg_catalog.pg_replication_slots物件的建立、更新和刪除。

在建立藍/綠部署時指定變更

建立藍/綠部署時，您可以在綠色環境中對資料庫執行個體進行下列變更。

您可以在部署之後，於綠色環境中對資料庫執行個體進行其他修改。例如，您可能會對資料庫進行結構描述變更，或變更綠色環境中一或多個資料庫執行個體所使用的資料庫執行個體類別。

如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

指定較高的引擎版本

如果想要測試資料庫引擎升級，您可以指定更高的引擎版本。轉換時，資料庫會升級至您指定的主要或次要資料庫引擎版本。

指定不同的資料庫參數群組

您可以測試參數變更如何影響綠色環境中的資料庫執行個體，或在升級時針對新的主要資料庫引擎版本指定參數群組。

如果您指定不同的資料庫參數群組，則指定的資料庫參數群組會與綠色環境中的所有資料庫執行個體相關聯。如果您指定不同的參數群組，則綠色環境中的每個資料庫執行個體都會與其對應藍色資料庫執行個體的參數群組相關聯。

啟用 RDS Optimized Writes

您可以使用藍/綠部署升級至支援 RDS Optimized Writes 的資料庫執行個體類別。您只能在使用受支援的資料庫執行個體類別建立的資料庫上啟用 RDS Optimized Writes。因此，此選項會建立使用受支援資料庫執行個體類別的綠色資料庫，可讓您在綠色資料庫執行個體上開啟 RDS Optimized Writes。

如果您要從不支援 RDS Optimized Writes 的資料庫執行個體類別升級為支援 RDS Optimized Writes 的執行個體類別，則也必須升級綠色資料庫執行個體的儲存組態。如需詳細資訊，請參閱 [the section called “升級儲存組態”](#)。

您只能升級主要綠色資料庫執行個體的資料庫執行個體類別。根據預設，綠色環境中的僅供讀取複本會繼承藍色環境中的資料庫執行個體設定。成功建立綠色環境後，您必須在綠色環境中手動修改僅供讀取複本的資料庫執行個體類別。

視藍色資料庫執行個體的引擎版本和執行個體類別而定，不支援某些執行個體類別升級。如需資料庫執行個體類別的詳細資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

升級儲存組態

如果您的藍色資料庫不在最新的儲存組態上，RDS 可以將綠色資料庫執行個體從舊的儲存組態 (32 位元檔案系統) 移轉至偏好的組態。您可以使用 RDS 藍/綠部署來克服舊版 32 位元檔案系統對儲存和檔案大小的擴展限制。此外，如果指定的資料庫執行個體類別支援 Optimized Writes，則此設定會將儲存組態變更為與 RDS Optimized Writes 相容。

Note

升級儲存組態是 I/O 密集型作業，會導致藍/綠部署的建立時間延長。如果藍色資料庫執行個體使用佈建 IOPS SSD (io1) 儲存體，並且您佈建執行個體大小為 4xlarge 或更大的綠色環境，則儲存升級程序會更快。涉及一般用途 SSD (gp2) 儲存體的儲存升級可能會耗盡您的 I/O 用量餘額，導致升級時間更長。如需詳細資訊，請參閱 [the section called “資料庫執行個體儲存體”](#)。

在儲存體升級程序期間，資料庫引擎無法使用。如果藍色資料庫執行個體的儲存耗用量大於或等於已配置儲存體大小的 90%，則儲存升級程序會將綠色執行個體配置的儲存大小增加 10%。

只有當您的藍色資料庫不在最新的儲存組態上，或您要變更相同要求中的資料庫執行個體類別時，才能使用此選項。

在建立藍/綠部署時處理延遲載入

當您建立藍/綠部署時，Amazon RDS 會透過從資料庫快照還原，在綠色環境中建立主要資料庫執行個體。在建立之後，綠色資料庫執行個體會繼續在背景中載入資料，這稱為延遲載入。如果資料庫執行個體具有僅供讀取複本，則這些複本也會從資料庫快照建立，並受制於延遲載入。

如果您存取尚未載入的資料，資料庫執行個體會立即從 Amazon S3 下載所請求的資料，然後繼續在背景載入剩餘的資料。如需更多資訊，請參閱 [Amazon EBS 快照](#)。

若要協助緩解延遲載入對要快速存取之資料表的影響，您可以執行涉及完整資料表掃描的作業，例如 `SELECT *`。此操作會允許 Amazon RDS 從 S3 下載所有已備份的資料表資料。

如果應用程式嘗試存取未載入的資料，則在載入資料時，應用程式可能會遭遇比正常情況更高的延遲。由於延遲載入而產生的更高延遲可能會導致對延遲敏感的工作負載效能不佳。

Important

如果您在資料載入完成之前切換藍/綠部署，您的應用程式可能會由於高延遲而遭遇效能問題。

建立藍/綠部署

您可以使用 AWS Management Console、或 RDS API 建立藍/綠部署。AWS CLI

主控台

建立藍/綠部署

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇您要將其複製到綠色環境的資料庫執行個體。
3. 選擇「動作」，「建立藍/綠部署」。

如果您選擇 RDS for PostgreSQL 資料庫執行個體，請檢閱並確認邏輯複寫限制。如需詳細資訊，請參閱 [the section called “PostgreSQL 邏輯複寫限制”](#)。

Create Blue/Green Deployment (建立藍/綠部署) 頁面即會出現。

Create Blue/Green Deployment: mydb1 Info

Create a Blue/Green Deployment that clones the resources of your current production environment (blue) to a staging environment (green). You can modify the green environment without affecting the blue environment. When you're ready, switch to the green environment to make it the current production environment.

Settings

Identifiers Info

Blue database identifiers Blue

Selected database identifiers in the current production environment. The databases in the green environment are generated automatically when the Blue/Green Deployment is created.

mydb1

mydb2

Blue/Green Deployment identifier

Type a name for your Blue/Green Deployment. The name must be unique across all Blue/Green Deployments owned by your AWS account in the current AWS Region.

The Blue/Green Deployment identifier is case-insensitive, but is stored as all lowercase (as in "mybgdeployment"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Blue/Green Deployment settings Info

Choose the engine version for green databases.

Choose the DB parameter group for green databases.

4. 檢閱藍色資料庫識別碼。請確定它們符合您預期在藍色環境中的資料庫執行個體。如果不符，請選擇 Cancel (取消)。
5. 針對 Blue/Green Deployment identifier (藍/綠部署識別符)，請輸入藍/綠部署的名稱。
6. 在其餘部分中，指定綠色環境的設定。如需每項設定的相關資訊，請參閱 [the section called “可用設定”](#)。

您可以在部署之後，於綠色環境中對資料庫進行其他修改。

7. 選擇建立暫存環境。

AWS CLI

若要使用建立藍/綠部署 AWS CLI，請使用建立藍-綠部署指令。如需每個選項的詳細資訊，請參閱[the section called “可用設定”](#)。

Example

對於Linux/macOS、或Unix：

```
aws rds create-blue-green-deployment \
  --blue-green-deployment-name my-blue-green-deployment \
  --source arn:aws:rds:us-east-2:123456789012:db:mydb1 \
  --target-engine-version 8.0.31 \
  --target-db-parameter-group-name mydbparametergroup
```

在 Windows 中：

```
aws rds create-blue-green-deployment ^
  --blue-green-deployment-name my-blue-green-deployment ^
  --source arn:aws:rds:us-east-2:123456789012:db:mydb1 ^
  --target-engine-version 8.0.31 ^
  --target-db-parameter-group-name mydbparametergroup
```

RDS API

若要使用 Amazon RDS API 建立藍色/綠色部署，請使用以下[CreateBlueGreenDeployment](#)操作。如需每個選項的詳細資訊，請參閱[the section called “可用設定”](#)。

建立藍/綠部署的設定

下表說明建立藍/綠部署時可以選擇的設定。如需有關 AWS CLI 選項的詳細資訊，請參閱[建立藍綠色部署](#)。如需 RDS API 參數的詳細資訊，請參閱[CreateBlueGreenDeployment](#)。

主控台設定	設定說明	CLI 選項和 RDS API 參數
藍色/綠色部署識別碼	藍/綠部署的名稱。	CLI 選項：

主控台設定	設定說明	CLI 選項和 RDS API 參數
		--blue-green-deployment-name API 參數： BlueGreenDeploymentName
藍色資料庫識別	您要複製到綠色環境的執行個體識別碼。使用 CLI 或 API 時，請指定執行個體 Amazon 資源名稱 (ARN)。	CLI 選項： --source API 參數： Source
綠色資料庫的資料庫參數群組	與綠色環境中的資料庫相關聯的參數群組。	CLI 選項： --target-db-parameter-group-name --target-db-cluster-parameter-group-name API 參數： TargetDBParameterGroupName TargetDBClusterParameterGroupName

主控台設定	設定說明	CLI 選項和 RDS API 參數
為綠色資料庫啟用最佳化寫入	<p>在綠色主要資料庫執行個體上啟用 RDS 最佳化寫入。如需詳細資訊，請參閱 the section called “啟用 RDS Optimized Writes”。</p> <p>如果您要從不支援 Optimized Writes 的資料庫執行個體類別變更為支援 Optimized Writes 的類別，您還需要執行儲存組態升級。如需詳細資訊，請參閱 the section called “升級儲存組態”。</p>	對於 CLI 和 API，指定支援 RDS 優化寫入的目標資料庫執行個體類別會自動在綠色主要資料庫執行個體上啟用該執行個體。
綠色資料庫的引擎版本	將綠色環境中的資料庫升級至指定的資料庫引擎版本。	<p>CLI 選項：</p> <pre>--target-engine-version</pre> <p>RDS API 參數：</p> <pre>TargetEngineVersion</pre>
綠色 DB 執行個體類別	<p>例如，綠色環境中每個資料庫執行個體的運算和記憶體容量 <code>db.m5d.xlarge</code>。</p> <p>只有當您為綠色資料庫啟用 RDS 最佳化寫入時，才會顯示此選項。</p>	<p>CLI 選項：</p> <pre>--target-db-instance-class</pre> <p>RDS API 參數：</p> <pre>TargetDBInstanceClass</pre>

主控台設定	設定說明	CLI 選項和 RDS API 參數
儲存設定升級	<p>選擇是否要升級儲存檔案系統組態。如果啟用此設定，RDS 會將綠色資料庫從舊儲存區檔案系統移轉至偏好的組態。</p> <p>只有當您的藍色資料庫不在最新的儲存組態上，或您要在相同要求中啟用 RDS Optimized Writes 時，才能使用此選項。</p> <p>如需詳細資訊，請參閱 the section called “升級儲存體檔案系統”。</p>	<p>CLI 選項：</p> <pre>--upgrade-target-storage-config</pre> <p>RDS API 參數：</p> <pre>UpgradeTargetStorageConfig</pre>

檢視藍/綠部署

您可以使用 AWS Management Console、AWS CLI 或 RDS API 檢視藍/綠部署的詳細資訊。

您也可以檢視和訂閱事件，以取得藍/綠部署的相關資訊。如需詳細資訊，請參閱 [藍/綠部署事件](#)。

主控台

檢視藍/綠部署的詳細資訊

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後在清單中尋找藍/綠部署。

	DB identifier	Role	Engine
○	mydb1 Blue	Primary	MySQL Community
○	mydb2 Blue	Replica	MySQL Community
○	my-blue-green-deployment	Blue/Green Deployment	-
○	mydb1-green-biuyjj Green	Primary	MySQL Community
○	mydb2-green-d8rdiv Green	Replica	MySQL Community

藍/綠部署的 Role (角色) 值是 Blue/Green Deployment (藍/綠部署)。

3. 選擇您要檢視以顯示其詳細資訊的藍/綠部署名稱。

每個索引標籤都有藍色部署的區段，以及綠色部署的區段。例如，在 [組態] 索引標籤上，如果您要在綠色環境中升級資料庫引擎版本，則在藍色環境和綠色環境中的資料庫引擎版本可能會有所不同。

下圖顯示 [連線與安全性] 索引標籤的範例：

RDS > Databases > mydb1 > my-blue-green-deployment

my-blue-green-deployment Refresh Modify Actions

Related

Filter by databases < 1 > Settings

	DB identifier	Role	Engine	Region & AZ
○	mydb1 Blue	Primary	MySQL Community	us-east-1f
○	mydb2 Blue	Replica	MySQL Community	us-east-1a
●	my-blue-green-deployment	Blue/Green Deployment	-	-
○	mydb1-green-wjsta5 Green	Primary	MySQL Community	us-east-1f

Connectivity & security | Monitoring | Logs & events | Configuration | Status | Tags | Recommendations

Blue connectivity and security Blue

Endpoint & port

Endpoint
mydb1.cbqv6h4bocho.us-east-1.rds.amazonaws.com

Port
3306

Green connectivity and security Green

Endpoint & port

Endpoint
mydb1-green-wjsta5.cbqv6h4bocho.us-east-1.rds.amazonaws.com

Port
3306

連線與安全性索引標籤也包含名為複寫的區段，其中顯示目前的邏輯複寫狀態，以及藍色環境與綠色環境之間的複本延遲。如果複寫狀態為 Replicating，則表示藍/綠部署成功複寫。

對於 RDS for PostgreSQL 藍色/綠色部署，如果您在藍色環境中進行不支援的 DDL 或大型物件變更，則複寫狀態可能會變更為 Replication degraded。如需詳細資訊，請參閱[the section called “PostgreSQL 邏輯複寫限制”](#)。

下圖顯示了「組態」索引標籤的範例：

Connectivity & security	Monitoring	Logs & events	Configuration	Status	Tags	Recommendations
Blue/Green Deployment						
DB identifier my-blue-green-deployment		Resource ID bgd-tuvaqsyncirljmml6				
Blue source database			Green source database			
Configuration			Configuration			
DB instance ID mydb1			DB instance ID mydb1-green-wjsta5			
Engine MySQL Community			Engine MySQL Community			
Engine version 8.0.35			Engine version 8.0.35			
DB name -			DB name -			
License model General Public License			License model General Public License			
Option groups default:mysql-8-0 ✔ In sync			Option groups default:mysql-8-0 ✔ In sync			
Amazon Resource Name (ARN) arn:aws:rds:us-east-1:478253424788:db:mydb1			Amazon Resource Name (ARN) arn:aws:rds:us-east-1:478253424788:db:mydb1-green-wjsta5			

下圖顯示「狀態」標籤的範例：

Connectivity & security | Monitoring | Logs & events | Configuration | **Status** | Tags | Recommendations

Green environment status (3)

Filter by Staging environment < 1 > ⚙️

Description	Status
Read Replica creation of the source	✔️ Completed
Backups configuration	🕒 In progress
Green topology creation	🕒 Pending

Switchover mapping (2)

Filter by Switchover mapping < 1 > ⚙️

Blue DB Instance ▲	Green DB Instance ▼	Role ▼	Status ▼
mydb1	mydb1-green-wjsta5	Primary	🕒 Provisioning
mydb2	Pending green DB instance	Replica	-

AWS CLI

若要使用檢視有關藍/綠部署的詳細資料AWS CLI，請使用指[describe-blue-green-deployments](#)令。

Example 透過篩選藍/綠部署的名稱來檢視其詳細資訊

使用[describe-blue-green-deployments](#)指令時，您可以在上篩選--blue-green-deployment-name。下列範例顯示藍/綠部署 (名為 *my-blue-green-deployment*) 的詳細資訊。

```
aws rds describe-blue-green-deployments --filters Name=blue-green-deployment-name,Values=my-blue-green-deployment
```

Example 透過指定藍/綠部署的識別符來檢視其詳細資訊

使用指[describe-blue-green-deployments](#)令時，您可以指定--blue-green-deployment-identifier。下列範例顯示識別符為 *bgd-1234567890abcdef* 的藍/綠部署的詳細資訊。

```
aws rds describe-blue-green-deployments --blue-green-deployment-identifier bgd-1234567890abcdef
```

RDS API

若要使用 Amazon RDS API 檢視藍/綠部署的詳細資訊，請使用 [DescribeBlueGreenDeployments](#) 操作並指定 `BlueGreenDeploymentIdentifier`。

切換藍/綠部署

切換可將綠色環境提升為新的生產環境。當綠色資料庫執行個體具有僅供讀取複本時，也會提升它們。切換前，生產流量會路由到藍色環境中的資料庫執行個體和僅供讀取複本。切換後，生產流量會路由到綠色環境中的資料庫執行個體和僅供讀取複本。

主題

- [切換逾時](#)
- [切換防護機制](#)
- [切換動作](#)
- [切換最佳實務](#)
- [切換前驗證 CloudWatch 指標](#)
- [切換藍/綠部署](#)
- [切換後](#)

切換逾時

您可以指定介於 30 秒與 3,600 秒(一小時)之間的切換逾時期間。如果切換所花費的時間超過指定的持續時間，則會復原任何變更，且不會對任一環境進行任何變更。預設逾時期間為 300 秒(五分鐘)。

切換防護機制

當您開始切換時，Amazon RDS 會執行一些基本檢查，以測試藍色和綠色環境是否準備好進行切換。這些檢查稱為切換防護機制。如果環境還沒有做好準備，這些切換防護機制可防止切換。因此，它們可避免超過預期的停機時間，並防止若切換開始，可能導致藍色和綠色環境之間遺失資料。

Amazon RDS 會在綠色環境上執行下列防護機制檢查：

- 複寫運作狀態 – 檢查綠色主要資料庫執行個體複寫狀態是否良好。綠色主要資料庫執行個體是藍色主要資料庫執行個體的複本。
- 複寫延遲 – 檢查綠色主要資料庫執行個體的複本延遲是否在轉換的允許限制內。允許的限制是以指定的逾時期間為基礎。複本延遲指出綠色主要資料庫執行個體落後於其藍色主要資料庫執行個體多

遠。如需詳細資訊，請參閱適用於 RDS for MySQL 的 [the section called “診斷和解決僅供讀取複本之間的延遲”](#)，以及適用於 RDS for PostgreSQL 的 [the section called “監控和調校複寫程序”](#)。

- 作用中寫入 – 確定綠色主要資料庫執行個體上沒有作用中寫入。

Amazon RDS 會在藍色環境上執行下列防護機制檢查：

- 外部複寫 — 對於版 RDS，請確定藍色環境不是自我管理的邏輯來源 (發行者) 或複本 (訂閱者)。如果是這樣，建議您將自我管理的複寫插槽和訂閱放置在藍色環境中的所有資料庫中，繼續進行轉換，然後重新建立它們以繼續複寫。對於適用於 MySQL RDS 和適用於 MariaDB 的 RDS，請檢查藍色資料庫是否不是外部備份記錄檔複本。如果是，請確保它沒有主動複製。
- 長時間執行的作用中寫入 – 確定藍色主要資料庫執行個體上沒有長時間執行的作用中寫入，因為這些寫入可能會增加複本延遲。
- 長時間執行的 DDL 陳述式 – 確定藍色主要資料庫執行個體上沒有長時間執行的 DDL 陳述式，因為這些陳述式可能會增加複本延遲。
- 不支援的 PostgreSQL 變更 – 對於 RDS for PostgreSQL 資料庫執行個體，請確定沒有任何 DDL 變更，也未在藍色環境上執行大型物件的新增或修改。如需詳細資訊，請參閱 [the section called “PostgreSQL 邏輯複寫限制”](#)。

如果 Amazon RDS 偵測到不支援的 PostgreSQL 變更，其會將複寫狀態變更為 Replication degraded，並通知您無法對藍/綠部署進行轉換。若要繼續進行轉換，建議您刪除並重新建立藍/綠部署和所有綠色資料庫。若要這樣做，請選擇動作、刪除綠色資料庫。

切換動作

當您切換藍/綠部署時，RDS 會執行下列動作：

1. 執行防護機制檢查，以驗證藍色和綠色環境是否已準備好進行切換。
2. 在這兩個環境中停止主要資料庫執行個體上的新寫入操作。
3. 捨棄與這兩個環境中資料庫執行個體的連線，而且不允許新的連線。
4. 等待複寫在綠色環境中趕上進度，以便綠色環境與藍色環境同步。
5. 在這兩個環境中重新命名資料庫執行個體。

RDS 會重新命名綠色環境中的資料庫執行個體，以符合藍色環境中對應的資料庫執行個體。例如，假設藍色環境中資料庫執行個體的名稱為 mydb。也會假設綠色環境中對應資料庫執行個體的名稱為 mydb-green-abc123。在切換期間，綠色環境中資料庫執行個體的名稱會變更為 mydb。

RDS 會重新命名藍色環境中的資料庫執行個體，方法是將 `-oldn` 附加至目前名稱 (其中 *n* 是數字)。例如，假設藍色環境中資料庫執行個體的名稱為 `mydb`。切換後，資料庫執行個體名稱可能是 `mydb-old1`。

RDS 也會重新命名綠色環境中的端點，以符合藍色環境中的對應端點，以便不需要應用程式變更。

6. 允許連線至這兩個環境中的資料庫。
7. 在新的生產環境中允許主要資料庫執行個體上的寫入操作。

轉換之後，先前的生產主要資料庫執行個體資料庫只允許讀取作業，直到您將 `read_only` 參數設定為資料庫執行個體 0 並重新啟動為止。

您可以使用 Amazon 監控轉換的狀態。EventBridge 如需詳細資訊，請參閱 [the section called “藍/綠部署事件”](#)。

如果您已在藍色環境中配置標籤，則這些標籤會在切換期間移至新的生產環境。先前的生產環境也會保留這些標籤。如需標籤的詳細資訊，請參閱 [標記 Amazon RDS 資源](#)。

如果切換開始，然後在完成前由於任何原因而停止，則會復原任何變更，且不會對任一環境進行任何變更。

切換最佳實務

在您轉換之前，我們強烈建議您完成下列任務，以遵守最佳實務：

- 徹底測試綠色環境中的資源。確保它們正常有效地執行。
- 監控相關的 Amazon CloudWatch 指標。如需詳細資訊，請參閱 [the section called “切換前驗證 CloudWatch 指標”](#)。
- 識別切換的最佳時間。

切換期間，會中斷這兩種環境中資料庫的寫入。識別生產環境上流量最低的時間。長時間執行的交易 (例如作用中 DDL) 可能會增加您的切換時間，因而延長生產工作負載的停機時間。

如果您的資料庫執行個體上有大量連線，請考慮在切換藍/綠部署之前，手動將其減少到應用程式所需的最低數量。達成此目標的方法之一是建立一項指令碼，用以監控藍/綠部署狀態，並在偵測到狀態變更至 `SWITCHOVER_IN_PROGRESS` 時開始清除連線。

- 確定這兩個環境中的資料庫執行個體處於 Available 狀態。
- 確定綠色環境中的主要資料庫執行個體運作良好且複寫中。

- 請確定您的網路和用戶端組態不會將 DNS 快取存留時間 (TTL) 增加到五秒以上，這是 RDS DNS 區域的預設值。
否則，應用程式會在切換之後繼續將寫入流量傳送至藍色環境。
- 在切換之前，請確定資料載入完成。如需詳細資訊，請參閱 [the section called “處理延遲載入”](#)。
- 對於 RDS 資料庫執行個體，請執行下列動作：
 - 檢閱邏輯複寫限制，並在切換之前採取任何必要的動作。如需詳細資訊，請參閱 [the section called “PostgreSQL 邏輯複寫限制”](#)。
 - 執行 ANALYZE 操作以重新整理 pg_statistics 資料表。這樣可以降低轉換後發生效能問題的風險。

Note

切換期間，您無法修改切換中包含的任何資料庫執行個體。

切換前驗證 CloudWatch 指標

在切換藍/綠部署之前，我們建議您在 Amazon CloudWatch 中檢查以下指標的值。

- ReplicaLag – 使用此指標識別綠色環境上目前的複寫延遲。為了減少停機時間，請在切換前確保此值趨近於零。
- DatabaseConnections – 使用此指標估計藍/綠部署的活動層級，並在切換前確認該值處於部署的可接受層級。如果績效詳情已開啟，則 DBLoad 為更準確的指標。

如需這些指標的詳細資訊，請參閱 [the section called “CloudWatch 適用於 RDS 的指標”](#)。

切換藍/綠部署

您可以使用 AWS Management Console、或 RDS API 切換藍/綠部署。AWS CLI

主控台

切換藍/綠部署

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要切換的藍/綠部署。

3. 針對 Actions (動作), 選擇 Switch over (切換)。

Switch over (切換) 頁面即會出現。

Switchover summary

You are about to switch over from Blue databases to Green databases. Check the settings of the Green databases to verify that they are ready for the switchover.

Blue databases Blue	Green databases Green
Identifiers mydb1 mydb2	Identifiers mydb1-green-biuyjj mydb2-green-d8rdiv
Engine version mysql 8.0.33	Engine version mysql 8.0.35
Option group default:mysql-8-0	Option group default:mysql-8-0
Parameter group default.mysql8.0	Parameter group default.mysql8.0
Size 400 GiB	Size 400 GiB
VPC sg-ee82bee3	VPC sg-ee82bee3
Multi-AZ us-east-1c	Multi-AZ us-east-1c
Storage type Provisioned IOPS SSD (io1)	Storage type Provisioned IOPS SSD (io1)
Storage file system configuration Info Current	Storage file system configuration Info Current

4. 在 Switch over (切換) 頁面上, 檢閱切換摘要。請確定這兩個環境中的資源符合您預期的資源。如果不符, 請選擇 Cancel (取消)。
5. 針對逾時設定, 輸入轉換的時間限制。
6. 如果您的執行個體正在執行 RDS for PostgreSQL, 請檢閱並確認轉換前建議。如需詳細資訊, 請參閱 [the section called “PostgreSQL 邏輯複寫限制”](#)。

7. 選擇 Switch over (切換)。

AWS CLI

若要使用切換藍/綠部署 AWS CLI，請搭配下列選項使用[切換-藍綠](#)部署命令：

- `--blue-green-deployment-identifier`— 指定藍/綠部署的資源 ID。
- `--switchover-timeout` – 指定切換的時間限制，以秒為單位。預設值為 300。

Example 切換藍/綠部署

對於LinuxmacOS、或Unix：

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-1234567890abcdef \  
  --switchover-timeout 600
```

在 Windows 中：

```
aws rds switchover-blue-green-deployment ^  
  --blue-green-deployment-identifier bgd-1234567890abcdef ^  
  --switchover-timeout 600
```

RDS API

若要使用 Amazon RDS API 切換藍/綠部署，請搭配下列參數使用[SwitchoverBlueGreenDeployment](#) 操作：

- `BlueGreenDeploymentIdentifier`— 指定藍/綠部署的資源 ID。
- `SwitchoverTimeout` – 指定切換的時間限制，以秒為單位。預設值為 300。

切換後

切換後，先前藍色環境中的資料庫執行個體會保留下來。標準成本適用於這些資源。藍色和綠色環境之間的複寫會停止。

RDS 會重新命名藍色環境中的資料庫執行個體，方法是將 `-oldn` 附加至目前資源名稱 (其中 *n* 是數字)。在您將 `read_only` 參數設定為 0 之前，資料庫執行個體是唯讀的。

	DB identifier ▲	Role ▼	Engine ▼
○	mydb1-old1 Old Blue	Primary	MySQL Community
○	mydb2-old1 Old Blue	Replica	MySQL Community
○	my-blue-green-deployment	Blue/Green Deployment	-
○	mydb1 New Blue	Primary	MySQL Community
○	mydb2 New Blue	Replica	MySQL Community

更新消費者的父節點

切換至 RDS for MariaDB 或 RDS for MySQL 藍/綠部署之後，如果藍色資料庫執行個體資料庫在切換前有任何外部複本或二進位記錄取用者，則必須在切換後更新其父節點，以維持複寫持續性。

切換之後，先前位於綠色環境中的入器資料庫執行個體會發出包含主記錄檔名稱和主記錄位置的事件。例如：

```
aws rds describe-events --output json --source-type db-instance --source-identifier db-instance-identifier

{
  "Events": [
    ...
    {
      "SourceIdentifier": "db-instance-identifier",
      "SourceType": "db-instance",
      "Message": "Binary log coordinates in green environment after switchover:
        file mysql-bin-changelog.000003 and position 804",
      "EventCategories": [],
      "Date": "2023-11-10T01:33:41.911Z",
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:db-instance-identifier"
    }
  ]
}
```


首先，請確定取用者或複本已套用舊藍色環境中的所有二進位記錄檔。然後，使用提供的二進制日誌坐標恢復消費者的應用程序。例如，如果您在 EC2 上執行 MySQL 複本，則可以使用以下 CHANGE MASTER TO 命令：

```
CHANGE MASTER TO MASTER_HOST='{new-writer-endpoint}', MASTER_LOG_FILE='mysql-bin-changelog.000003', MASTER_LOG_POS=804;
```

Note

如果取用者是 MariaDB 的另一個 RDS 或適用於 MariaDB 資料庫執行個體的 RDS，您可以依序執行下列預存程序：[the section called “mysql.rds_stop_replication”](#)、[the section called “mysql.rds_reset_external_master”](#)和。[the section called “mysql.rds_set_external_master”](#) [the section called “mysql.rds_start_replication”](#)

刪除藍/綠部署

您可以在切換藍/綠部署之前或之後將其刪除。

若您在切換藍/綠部署之前將其刪除，Amazon RDS 會選擇性地刪除綠色環境中的資料庫執行個體：

- 如果您選擇刪除綠色環境 (--delete-target) 中的資料庫執行個體，則必須關閉刪除保護。
- 如果未刪除綠色環境 (--no-delete-target) 中的資料庫執行個體，則會保留該執行個體，但它不再是藍/綠部署的一部分。在環境之間繼續複寫。

[切換](#)後，刪除綠色資料庫的選項無法在主控台中使用。使用刪除藍/綠部署時 AWS CLI，如果部署狀態為 SWITCHOVER_COMPLETED，則無法指定 --delete-target 選項。

Important

刪除藍/綠部署不會影響藍色環境。

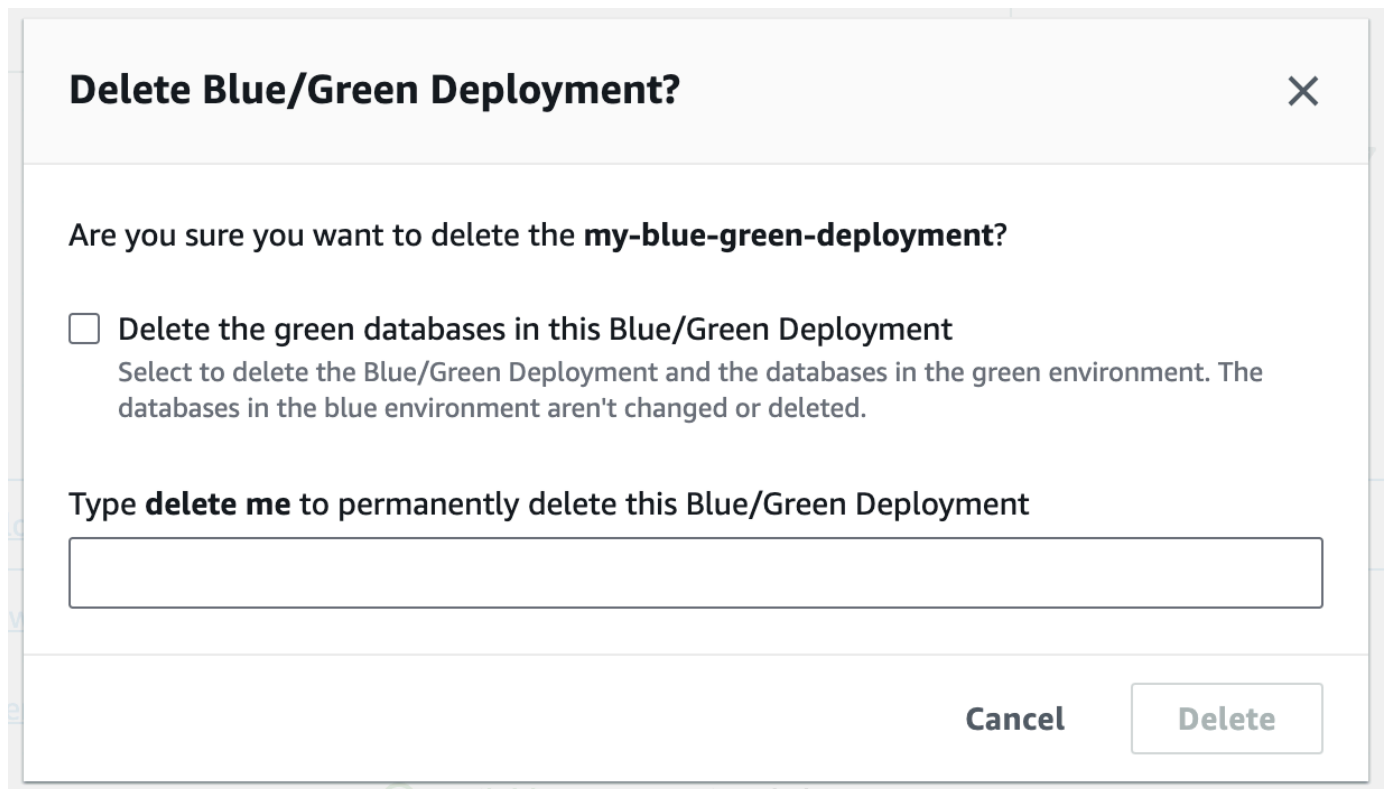
您可以使用 AWS Management Console、或 RDS API 刪除藍/綠部署。AWS CLI

主控台

刪除藍/綠部署

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要刪除的藍/綠部署。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。

Delete Blue/Green Deployment? (刪除藍/綠部署?) 視窗即會出現。



Delete Blue/Green Deployment? ✕

Are you sure you want to delete the **my-blue-green-deployment**?

Delete the green databases in this Blue/Green Deployment
Select to delete the Blue/Green Deployment and the databases in the green environment. The databases in the blue environment aren't changed or deleted.

Type **delete me** to permanently delete this Blue/Green Deployment

Cancel Delete

若要刪除綠色資料庫，請選取 Delete the green databases in this Blue/Green Deployment (刪除此藍/綠部署中的綠色資料庫)。

4. 在方塊中輸入 **delete me**。
5. 選擇 Delete (刪除)。

AWS CLI

若要使用刪除藍/綠部署 AWS CLI，請搭配下列選項使用 [delete-blue-green-deployment](#) 指令：

- `--blue-green-deployment-identifier`— 要刪除的藍/綠部署的資源 ID。
- `--delete-target` – 指定刪除綠色環境中的資料庫執行個體。如果藍/綠部署的狀態為 `SWITCHOVER_COMPLETED`，則您無法指定此選項。
- `--no-delete-target` – 指定保留綠色環境中的資料庫執行個體。

Example 刪除藍/綠部署以及綠色環境中的資料庫執行個體

對於LinuxmacOS、或Unix：

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-1234567890abcdef \  
  --delete-target
```

在 Windows 中：

```
aws rds delete-blue-green-deployment ^\  
  --blue-green-deployment-identifier bgd-1234567890abcdef ^\  
  --delete-target
```

Example 刪除藍/綠部署，但保留綠色環境中的資料庫執行個體

對於LinuxmacOS、或Unix：

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-1234567890abcdef \  
  --no-delete-target
```

在 Windows 中：

```
aws rds delete-blue-green-deployment ^\  
  --blue-green-deployment-identifier bgd-1234567890abcdef ^\  
  --no-delete-target
```

RDS API

若要使用 Amazon RDS API 刪除藍/綠部署，請搭配下列參數使用 [DeleteBlueGreenDeployment](#) 操作：

- `BlueGreenDeploymentIdentifier`— 要刪除的藍/綠部署的資源 ID。

- `DeleteTarget` – 指定 `TRUE` 以刪除綠色環境中的資料庫執行個體，或指定 `FALSE` 以保留它們。如果藍/綠部署的狀態為 `SWITCHOVER_COMPLETED`，則不能為 `TRUE`。

備份、還原和匯出資料

本節說明如何從 Amazon RDS 資料庫執行個體或異地同步備份資料庫叢集備份、還原和匯出資料。

主題

- [備份簡介](#)
- [管理自動備份](#)
- [管理手動備份](#)
- [從資料庫 快照還原](#)
- [複製資料庫快照](#)
- [共享資料庫 快照](#)
- [將資料庫快照資料匯出至 Amazon S3](#)
- [用 AWS Backup 於管理自動備份](#)

備份簡介

Amazon RDS 會在資料庫執行個體的備份時段建立並儲存資料庫執行個體或多可用區域資料庫叢集的自動備份。RDS 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。RDS 會根據您指定的備份保留期儲存資料庫執行個體的自動備份。如有需要，您可將資料庫執行個體復原至備份保留期間的任何時間點。

自動備份會遵循下列規則：

- 您的資料庫執行個體必須處於 available 狀態才能進行自動備份。當您的資料庫執行個體處於 available 以外的狀態，例如，storage_full，自動備份不會發生。
- 當相同資料庫在相同 AWS 區域中執行資料庫快照複製時，不會進行自動備份。

您也可以建立資料庫快照以手動備份您的資料庫執行個體。如需手動建立資料庫快照的詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

資料庫執行個體的第一個快照會包含完整的資料庫資料。相同資料庫的後續快照均採用遞增處理，這表示系統僅會儲存上次執行快照後有所變更的資料。

您可以複製自動和手動資料庫快照，並共用手動資料庫快照。如需複製資料庫快照的詳細資訊，請參閱 [複製資料庫快照](#)。如需共用資料庫快照的詳細資訊，請參閱 [共享資料庫快照](#)。

備份儲存體

每個儲存的 Amazon RDS 備份儲存體都 AWS 區域 是由該區域的自動備份和手動資料庫快照組成。總備份儲存空間等於該區域所有備份的儲存總和。將資料庫快照移到其他區域會增加目的地區域的備份儲存。備份儲存在 Amazon S3 中。

如需備份儲存成本的詳細資訊，請參閱 [Amazon RDS 定價](#)。

如果您選擇在刪除資料庫執行個體時保留自動備份，則自動備份會保留整個保留期間。如果您在刪除資料庫執行個體時沒有選擇 Retain automated backups (保留自動備份)，則所有自動備份會隨著資料庫執行個體一起刪除。自動備份在刪除之後就無法復原。如果您選擇讓 Amazon RDS 建立最終的資料庫快照，然後再刪除您的資料庫執行個體，您可以使用它來恢復您的資料庫執行個體。或者，您可以使用先前建立的手動快照。系統並不會刪除手動快照。每個區域最多可以有 100 個手動快照。

管理自動備份

本節說明如何管理資料庫執行個體和資料庫叢集的自動備份。

主題

- [備份時段](#)
- [Backup retention period \(備份保留期間\)](#)
- [啟用自動備份](#)
- [保留自動備份](#)
- [刪除保留的自動備份](#)
- [停用自動備份](#)
- [利用不受支援的 MySQL 儲存引擎進行自動備份](#)
- [利用不受支援的 MariaDB 儲存引擎進行自動備份](#)
- [將自動備份複製到另一個 AWS 區域](#)

備份時段

自動備份會每天在偏好的備份時段內執行。如果備份需要的時間超過所分配的備份時段，備份會在時段結束後繼續執行直到完成。備份時段不可與每週的資料庫執行個體或多可用區域資料庫叢集維護時段重疊。

在自動備份期間，儲存輸入/輸出可能會在備份程序初始化時短暫暫停 (通常在幾秒內)。在備份異地同步備份部署時，可能會有幾分鐘的延遲增加情形。若為 MariaDB、MySQL、Oracle 和 PostgreSQL，在備份異地同步備份部署期間，您的主要執行個體上不會暫停輸入/輸出活動，因為備份取自於待命執行個體。若為 SQL Server，在備份單一可用區和多可用區部署期間會短暫停止 I/O 活動，因為備份取自於主要執行個體。對於 Db2，即使備份是從待命狀態取得，I/O 活動也會在備份期間短暫暫停。

如果資料庫執行個體或叢集在應啟動備份時的工作負載繁重，有時候可能會略過自動備份。如果略過備份，您仍然可以執行 point-in-time-recovery (PITR)，並且在下一個備份時段仍會嘗試備份。如需 PITR 的詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

當您建立資料庫執行個體或多可用區域資料庫叢集時，若未指定偏好的備份時段，Amazon RDS 將會指派預設 30 分鐘的備份時段。該窗口是從 8 小時的時間段中隨機選擇每個 AWS 區域窗口。下表列出指派預設備份時段的每個 AWS 區域 時間區塊。

區域名稱	區域	時間區塊
美國東部 (俄亥俄)	us-east-2	上午 3 時至 11 時 (UTC)
美國東部 (維吉尼亞北部)	us-east-1	上午 3 時至 11 時 (UTC)
美國西部 (加利佛尼亞北部)	us-west-1	上午 6 時至下午 2 時 (UTC)
美國西部 (奧勒岡)	us-west-2	上午 6 時至下午 2 時 (UTC)
非洲 (開普敦)	af-south-1	上午 3 時至 11 時 (UTC)
亞太區域 (香港)	ap-east-1	06:00–14:00 UTC
亞太區域 (海德拉巴)	ap-south-2	06:30–14:30 UTC
亞太區域 (雅加達)	ap-southeast-3	08:00–16:00 UTC
亞太區域 (墨爾本)	ap-southeast-4	上午 11 時至下午 7 時 (UTC)
亞太區域 (孟買)	ap-south-1	下午 4 時 30 分至上午 12 時 30 分 (UTC)
亞太區域 (大阪)	ap-northeast-3	上午 12 時至上午 8 時 (UTC)
亞太區域 (首爾)	ap-northeast-2	下午 1 時至 9 時 (UTC)
亞太區域 (新加坡)	ap-southeast-1	下午 2 時至 10 時 (UTC)
亞太區域 (雪梨)	ap-southeast-2	中午 12 時至下午 8 時 (UTC)
亞太區域 (東京)	ap-northeast-1	下午 1 時至 9 時 (UTC)
加拿大 (中部)	ca-central-1	03:00–11:00 UTC
加拿大西部 (卡加利)	ca-west-1	18:00–02:00 UTC
中國 (北京)	cn-north-1	上午 6 時至下午 2 時 (UTC)
中國 (寧夏)	cn-northwest-1	上午 6 時至下午 2 時 (UTC)

區域名稱	區域	時間區塊
歐洲 (法蘭克福)	eu-central-1	下午 8 時至上午 4 時 (UTC)
歐洲 (愛爾蘭)	eu-west-1	下午 10 時至上午 6 時 (UTC)
歐洲 (倫敦)	eu-west-2	22:00–06:00 UTC
歐洲 (米蘭)	eu-south-1	上午 2 時至 10 時 (UTC)
歐洲 (巴黎)	eu-west-3	上午 7 時 29 分至下午 2 時 29 分 (UTC)
歐洲 (西班牙)	eu-south-2	上午 2 時至 10 時 (UTC)
歐洲 (斯德哥爾摩)	eu-north-1	下午 11 時至上午 7 時 (UTC)
歐洲 (蘇黎世)	eu-central-2	上午 2 時至 10 時 (UTC)
以色列 (特拉維夫)	il-central-1	03:00–11:00 UTC
中東 (巴林)	me-south-1	06:00–14:00 UTC
中東 (阿拉伯聯合大公國)	me-central-1	上午 5 時至下午 1 時 (UTC)
南美洲 (聖保羅)	sa-east-1	23:00–07:00 UTC
AWS GovCloud (美國東部)	us-gov-east-1	下午 5 時至上午 1 時 (UTC)
AWS GovCloud (美國西部)	us-gov-west-1	06:00–14:00 UTC

Backup retention period (備份保留期間)

您可以在建立資料庫執行個體或多可用區域資料庫叢集時設定備份保留期間。如果您使用 Amazon RDS API 建立資料庫執行個體 AWS CLI，或者如果未設定備份保留期，則預設備份保留期為一天。如果您使用主控台建立資料庫執行個體，預設備份保留期為 7 天。

在您建立資料庫執行個體或叢集之後，可修改備份保留期。您可以將資料庫執行個體的備份保留期設為 0 到 35 天。將備份保留期設為 0 會停用自動備份。對於異地同步備份資料庫叢集，您可以將備份保留期設定為 1 到 35 天。手動快照限制 (每個區域 100 個) 不適用於自動備份。

停止資料庫執行個體或叢集時，不會建立自動備份。如果資料庫執行個體已停止，備份的保留時間可超過備份保留期間。RDS 不包含計算備份保留期間時在 stopped 狀態所花費的時間。

Important

如果您將資料庫執行個體的備份保留期從 0 變更為非零值或從非零值變更為 0，就會發生中斷。

啟用自動備份

如果您的資料庫執行個體未啟用自動備份，您可以隨時啟用。您可以將備份保留期間設定為正的非零值以啟用自動備份。當自動備份啟用時，資料庫執行個體會離線，並立即建立備份。

Note

如果您在中管理備份 AWS Backup，則無法啟用自動備份。如需詳細資訊，請參閱 [用 AWS Backup 於管理自動備份](#)。

主控台

立即啟用自動備份

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇資料庫，然後選擇您要修改的資料庫執行個體或多可用區域資料庫叢集。
3. 選擇 Modify (修改)。
4. 針對 Backup retention period (備份保留期)，選擇非零正值，例如 3 天。
5. 選擇 Continue (繼續)。
6. 選擇 Apply immediately (立即套用)。
7. 選擇修改資料庫執行個體或修改叢集，以儲存您的變更並啟用自動備份。

AWS CLI

若要啟用自動備份，請使用 AWS CLI [modify-db-instance](#) 或 [modify-db-cluster](#) 指令。

包含以下參數：

- `--db-instance-identifier` (或多可用區域資料庫叢集的 `--db-cluster-identifier`)
- `--backup-retention-period`
- `--apply-immediately` 或 `--no-apply-immediately` *

在以下範例中，我們將備份保留期間設定為三天來啟用自動備份。變更會立即套用。

Example

對於Linux/macOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --backup-retention-period 3 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --backup-retention-period 3 ^  
  --apply-immediately
```

RDS API

若要啟用自動備份，請使用 RDS API [ModifyDBInstance](#) 或 [ModifyDBCluster](#) 操作，其中包含下列必要參數：

- `DBInstanceIdentifier` 或 `DBClusterIdentifier`
- `BackupRetentionPeriod`

檢視自動備份

若要檢視自動備份，請在導覽窗格中選擇 Automated backups (自動備份)。若要檢視與自動備份相關聯的個別快照，請在導覽窗格中選擇 Snapshots (快照)。或者，您可以描述與自動備份相關聯的個別快照。您可以從那裡的其中一個快照直接還原資料庫執行個體。

若要使用說明現有資料庫執行個體的自動備份 AWS CLI，請使用下列命令之一：

```
aws rds describe-db-instance-automated-backups --db-instance-  
identifier DBInstanceIdentifier
```

或

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

若要使用 RDS API 來描述自動備份，請呼叫 [DescribeDBInstanceAutomatedBackups](#) 動作並搭配下列其中一個參數：

- DBInstanceIdentifier
- DbiResourceId

保留自動備份

Note

您僅能保留資料庫執行個體的自動備份，不能保留多可用區域資料庫叢集的自動備份。

當您刪除資料庫執行個體時，您可以選擇保留自動備份。自動備份的保留天數等於在刪除時您為資料庫執行個體設定的備份保留期。

保留的自動備份包含來自資料庫執行個體的系統快照和交易日誌。還包含還原到作用中執行個體時所需的資料庫執行個體屬性，例如配置的儲存體和資料庫執行個體類別。

保留的自動備份和手動快照會產生帳單費用，直到將其刪除為止。如需詳細資訊，請參閱 [保留成本](#)。

您可以保留執行 Db2、MariaDB、MySQL、PostgreSQL、甲骨文和 Microsoft SQL 伺服器引擎的 RDS 執行個體的自動備份。

您可以使用 RDS API 和來還原或移除保留的自動備份 AWS CLI。AWS Management Console

主題

- [保留期間](#)
- [檢視保留的備份](#)
- [還原](#)
- [保留成本](#)
- [限制](#)

保留期間

保留的自動備份中的系統快照和交易日誌會像在來源資料庫執行個體中一樣過期。因為沒有為此執行個體建立新的快照或日誌，保留的自動備份最後會完全過期。事實上，根據您刪除來源執行個體時的保留期間設定，保留的自動備份存留多久取決於其最後一個系統快照是否還存在。在最後一個系統快照過期之後，系統會移除保留的自動備份。

您可以像刪除資料庫執行個體一樣來移除保留的自動備份。您可以使用主控台或 RDS API 操作 `DeleteDBInstanceAutomatedBackup`，以移除保留的自動備份。

最終快照與保留的自動備份彼此獨立。即使您保留自動備份，仍強烈建議您建立最終快照，因為保留的自動備份最後會過期。最終快照不會過期。

檢視保留的備份

若要檢視保留的自動備份，請在導覽窗格中選擇 Automated backups (自動備份)，然後選擇 Retained (保留)。若要檢視與保留的自動備份相關聯的個別快照，請在導覽窗格中選擇 Snapshots (快照)。或者，您可以描述與保留的自動備份相關聯的個別快照。您可以從那裡的其中一個快照直接還原資料庫執行個體。

若要使用描述保留的自動備份 AWS CLI，請使用下列命令：

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

若要使用 RDS API 來描述保留的自動備份，請呼叫 [DescribeDBInstanceAutomatedBackups](#) 動作並搭配 `DbiResourceId` 參數。

還原

如需有關從自動備份中還原資料庫執行個體的資訊，請參閱[將資料庫執行個體還原至指定的時間](#)。

保留成本

保留的自動備份所需的成本是其相關聯的系統快照的總儲存成本。交易日誌或執行個體中繼資料不會產生額外費用。備份的所有其他定價規則適用於可還原的執行個體。

例如，假設配置給執行中執行個體的總儲存體是 100 GB。也假設您有 50 GB 的手動快照，還有 75 GB 的系統快照，這些都與保留的自動備份相關聯。在此案例中，您只需要對額外 25 GB 的備份儲存付費，就像這樣： $(50 \text{ GB} + 75 \text{ GB}) - 100 \text{ GB} = 25 \text{ GB}$ 。

限制

保留的自動備份有下列限制：

- 一個 AWS 區域中保留的自動備份數目上限為 40 個。這不算在資料庫執行個體配額內。您可以同時有 40 個執行中的資料庫執行個體和額外 40 個保留的自動備份。
- 保留的自動備份不含參數或選項群組的相關資訊。
- 您可以將已刪除的執行個體還原到刪除當時的保留期間內的某個時間點。
- 您無法修改保留的自動備份。這是因為它包含您刪除來源執行個體時存在的系統備份、交易日誌和資料庫執行個體屬性。

刪除保留的自動備份

不再需要保留的自動備份時，刪除即可。

主控台

刪除保留的自動備份

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 在已保留標籤，選擇您要刪除的已保留自動備份。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 在確認頁面上，輸入 **delete me**，然後選擇 Delete (刪除)。

AWS CLI

您可以使用 AWS CLI 命令 [delete-db-instance-automated-backup](#) 與以下選項刪除保留的自動備份：

- `--dbi-resource-id` – 來源資料庫執行個體的資源識別符。

您可以執行 AWS CLI 指令 [describe-db-instance-automated-backup](#)，找到保留自動備份的來源資料庫執行個體的資源識別碼。

Example

下列範例刪除來源資料庫執行個體識別符為 `db-123ABCEXAMPLE` 的已保留自動備份。

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance-automated-backup \  
  --dbi-resource-id db-123ABCEXAMPLE
```

在 Windows 中：

```
aws rds delete-db-instance-automated-backup ^  
  --dbi-resource-id db-123ABCEXAMPLE
```

RDS API

您可以使用InstanceAutomatedBackup具有下列參數的 [Amazon RDS API 操作](#)刪除保留的自動備份：

- `DbiResourceId` – 來源資料庫執行個體的資源識別符。

您可以使用 Amazon RDS API 操作 [說明 B](#)，找到保留自動備份的來源資料庫執行個體的資源識別碼。[InstanceAutomatedBackups](#)

停用自動備份

在某些情況下，您可能想要暫時停用自動備份，例如載入大量資料時。

Important

我們強烈建議禁用自動備份，因為它會禁用 point-in-time 恢復。停用資料庫執行個體或多可用區域資料庫叢集的自動備份將會刪除該資料庫現有的所有自動備份。如果您停用，然後重新啟用自動備份，您只能從您重新啟用自動備份的時間開始還原。

主控台

立即停用自動備份

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇資料庫，然後選擇您要修改的資料庫執行個體或多可用區域資料庫叢集。
3. 選擇 Modify (修改)。
4. 針對 Backup retention period (備份保留期)，選擇 0 days (0 天)。
5. 選擇 Continue (繼續)。
6. 選擇 Apply immediately (立即套用)。
7. 選擇修改資料庫執行個體或修改叢集，以儲存您的變更並啟用自動備份。

AWS CLI

若要立即停用自動備份，請使用 [modify-db-instance](#) 或 [modify-db-cluster](#) 命令，並將備份保留期限設定為 0 --apply-immediately。

Example

下列範例會立即停用多可用區域資料庫叢集的自動備份。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --backup-retention-period 0 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-cluster ^  
  --db-cluster-identifier mydbcluster ^  
  --backup-retention-period 0 ^  
  --apply-immediately
```

若要知道修改何時生效，請針對資料庫執行個體呼叫 `describe-db-instances`，或針對多可用區域資料庫叢集呼叫 `describe-db-clusters`，直到備份保留期的值為 0 且 `mydbcluster mydbinstance` 狀態為可用。


```
aws rds describe-db-clusters --db-cluster-identifier mydcluster
```

RDS API

若要立即停用自動備份，請呼叫 [ModifyDBInstance](#) 或 [ModifyDBCluster](#) 操作與下列參數搭配：

- DBInstanceIdentifier = mydbinstance (或 DBClusterIdentifier = mydbcluster)
- BackupRetentionPeriod = 0

Example

```
https://rds.amazonaws.com/  
?Action=ModifyDBInstance  
&DBInstanceIdentifier=mydbinstance  
&BackupRetentionPeriod=0  
&SignatureVersion=2  
&SignatureMethod=HmacSHA256  
&Timestamp=2009-10-14T17%3A48%3A21.746Z  
&AWSAccessKeyId=<&AWS; Access Key ID>  
&Signature=<Signature>
```

利用不受支援的 MySQL 儲存引擎進行自動備份

就 MySQL 資料庫引擎而言，自動備份僅支援 InnoDB 儲存引擎。使用這些功能搭配其他 MySQL 儲存引擎 (包括 MyISAM) 可能導致從備份中還原時發生不可靠的行為。尤其，因為 MyISAM 之類的儲存引擎不支援可靠的損毀復原，萬一毀損，您的資料表可能會損壞。基於這個原因，我們鼓勵您使用 InnoDB 儲存引擎。

- 若要將現有的 MyISAM 資料表轉換為 InnoDB 資料表，您可以使用 ALTER TABLE 命令，例如：
`ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- 如果選擇使用 MyISAM，則您可以使用 REPAIR 命令，嘗試手動修復在損毀之後損壞的資料表。如需詳細資訊，請參閱 MySQL 文件中的 [REPAIR TABLE 陳述式](#)。不過，如同 MySQL 文件所述，您很有可能無法復原所有資料。
- 如果想要在還原之前建立 MyISAM 資料表的快照，請遵循下列步驟：
 1. 停止 MyISAM 資料表的所有活動 (亦即，關閉所有工作階段)。

您可以關閉所有工作階段，方法為針對從 SHOW FULL PROCESSLIST 命令傳回的每一個程序呼叫 [mysql.rds_kill](#) 命令。

2. 鎖定並清空您的每一個 MyISAM 資料表。例如，以下命令會鎖定並清空兩個名為 `myisam_table1` 和 `myisam_table2` 的資料表：

```
mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
```

3. 建立資料庫執行個體或多可用區域資料庫叢集的快照。完成快照時，請釋放鎖定並恢復 MyISAM 資料表上的活動。您可以使用以下命令，釋放資料表上的鎖定：

```
mysql> UNLOCK TABLES;
```

這些步驟會強制 MyISAM 將記憶體中存放的資料清空到磁碟，以確保您從資料庫快照還原時全新開始。如需建立資料庫快照的詳細資訊，請參閱[為單一可用區資料庫執行個體建立資料庫快照](#)。

利用不受支援的 MariaDB 儲存引擎進行自動備份

針對 MariaDB 資料庫引擎，僅 InnoDB 儲存引擎支援自動備份。使用這些功能搭配其他 MariaDB 儲存引擎 (包括 Aria) 可能導致從備份中還原時發生不可靠的行為。即使 Aria 是 MyISAM 的防當機替代方案，您的資料表仍可能在當機時毀損。基於這個原因，我們鼓勵您使用 InnoDB 儲存引擎。

- 若要將現有的 Aria 資料表轉換為 InnoDB 資料表，您可以使用 ALTER TABLE 命令。例如：ALTER TABLE *table_name* ENGINE=innodb, ALGORITHM=COPY;
- 如果選擇使用 Aria，則您可以嘗試使用 REPAIR TABLE 命令，手動修復在當機之後損壞的資料表。如需詳細資訊，請參閱 <http://mariadb.com/kb/en/mariadb/repair-table/>。
- 如果想要在還原之前建立 Aria 資料表的快照，請遵循下列步驟：
 1. 停止 Aria 資料表的所有活動 (亦即，關閉所有工作階段)。
 2. 鎖定並清空您的每一個 Aria 資料表。
 3. 建立資料庫執行個體或多可用區域資料庫叢集的快照。完成快照時，請釋放鎖定並恢復 Aria 資料表上的活動。這些步驟會強制 Aria 將記憶體中儲存的資料排至磁碟，進而確保在您從資料庫快照進行還原時全新開始。

將自動備份複製到另一個 AWS 區域

如需額外的災難復原功能，您可以設定 Amazon RDS 資料庫執行個體，將快照和交易日誌複寫到您選擇 AWS 區域 的目的地。為資料庫執行個體設定備份複寫時，RDS 會在資料庫執行個體上的所有快照和交易記錄準備就緒時，立即啟動其跨區域複製作業。

資料傳輸需支付資料庫快照複製費用。複製資料庫快照後，目的地區域的儲存體將收取標準費用。如需詳細資訊，請參閱 [RDS 定價](#)。

如需使用備份複寫的範例，請參閱 AWS 線上技術討論的使用 [Amazon RDS for Oracle 跨區域自動備份的受管災難復原](#)。

Note

異地同步備份資料庫叢集不支援自動備份複寫。

主題

- [區域和版本可用性](#)
- [來源和目的地 AWS 區域 支援](#)
- [啟用跨區域自動備份](#)
- [尋找複寫備份的相關資訊](#)
- [從複寫備份還原至指定的時間](#)
- [停用自動備份複寫](#)
- [刪除複寫的備份](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需跨區域自動備份的版本和區域可用性的相關資訊，請參閱 [支援的區域和資料庫引擎，可在 Amazon RDS 中進行跨區域自動備份](#)。

來源和目的地 AWS 區域 支援

支援下列項目之間的 Backup 複寫 AWS 區域。

來源區域	目的地區域可用
亞太區域 (孟買)	亞太區域 (新加坡) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
亞太區域 (大阪)	亞太區域 (東京)
亞太區域 (首爾)	亞太區域 (新加坡)、亞太區域 (東京) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
亞太區域 (新加坡)	亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (雪梨)、亞太區域 (東京) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
亞太區域 (雪梨)	亞太區域 (新加坡) 美國東部 (維吉尼亞北部)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)
亞太區域 (東京)	亞太區域 (大阪)、亞太區域 (首爾)、亞太區域 (新加坡) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
加拿大 (中部)	歐洲 (愛爾蘭) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)
中國 (北京)	中國 (寧夏)
中國 (寧夏)	中國 (北京)
Europe (Frankfurt)	歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (巴黎)、歐洲 (斯德哥爾摩) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
歐洲 (愛爾蘭)	加拿大 (中部) 歐洲 (法蘭克福)、歐洲 (倫敦)、歐洲 (巴黎)、歐洲 (斯德哥爾摩)

來源區域	目的地區域可用
	美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)
歐洲 (倫敦)	歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (巴黎)、歐洲 (斯德哥爾摩) 美國東部 (維吉尼亞北部)
歐洲 (巴黎)	歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (斯德哥爾摩) 美國東部 (維吉尼亞北部)
歐洲 (斯德哥爾摩)	歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (巴黎) 美國東部 (維吉尼亞北部)
南美洲 (聖保羅)	美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)
AWS GovCloud (美國東部)	AWS GovCloud (美國西部)
AWS GovCloud (美國西部)	AWS GovCloud (美國東部)
美國東部 (維吉尼亞北部)	亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (雪梨)、亞太區域 (東京) 加拿大 (中部) 歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (巴黎)、歐洲 (斯德哥爾摩) 南美洲 (聖保羅) 美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)

來源區域	目的地區域可用
美國東部 (俄亥俄)	亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (東京) 加拿大 (中部) 歐洲 (法蘭克福)、歐洲 (愛爾蘭) 南美洲 (聖保羅) 美國東部 (維吉尼亞北部)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)
美國西部 (加利佛尼亞北部)	亞太區域 (雪梨) 加拿大 (中部) 歐洲 (愛爾蘭) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡)
美國西部 (奧勒岡)	亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (雪梨)、亞太區域 (東京) 加拿大 (中部) 歐洲 (法蘭克福)、歐洲 (愛爾蘭) 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)

您也可以使用 `describe-source-regions` AWS CLI 指令找出哪些 AWS 區域 可以彼此複製。如需詳細資訊，請參閱 [尋找複寫備份的相關資訊](#)。

啟用跨區域自動備份

您可以使用 Amazon RDS 主控台在新的或現有的資料庫執行個體上啟用備份複寫。您也可以使用 `start-db-instance-automated-backups-replication` AWS CLI 命令或

StartDBInstanceAutomatedBackupsReplication RDS API 作業。您最多可以將 20 個備份複製到每個目 AWS 區域 的地 AWS 帳戶。

Note

若要能夠複製自動備份，請務必將其啟用。如需詳細資訊，請參閱 [啟用自動備份](#)。

主控台

您可以為新的或現有的資料庫執行個體啟用備份複製：

- 對於新的資料庫執行個體，請在啟動執行個體時將其啟用。如需詳細資訊，請參閱 [資料庫執行個體的設定](#)。
- 對於現有的資料庫執行個體，請使用下列程序。

為現有的資料庫執行個體啟用備份複製

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 在 Current Region (目前區域) 標籤上，選擇您要啟用備份複製的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Manage cross-Region replication (管理跨區域複製)。
5. 在 Backup replication (備份複製) 下，選擇 Enable replication to another AWS 區域 (啟用複製至另一個 AWS 區域)。
6. 選擇 Destination Region (目的地區域)。
7. 選擇 Replicated backup retention period (複製備份保留期間)。
8. 如果您已在來源資料庫執行個體上啟用加密，請選擇用 AWS KMS key 於加密備份或輸入金鑰 ARN。
9. 選擇 Save (儲存)。

在來源區域中，複製的備份會列在 Automated backups (自動備份) 頁面的 Current Region (目前區域) 標籤上。在目的地區域中，複製的備份會列在 Automated backups (自動備份) 頁面的 Replicated backups (複製備份) 標籤上。

AWS CLI

使用 [start-db-instance-automated-backups-replication](#) AWS CLI 指令啟用備份複寫。

下列 CLI 範例會將自動備份從 美國西部 (奧勒岡) 區域 中的資料庫執行個體複寫到 美國東部 (維吉尼亞北部) 區域。它也會使用目的地區域中的加密複寫 AWS KMS key 的備份。

啟用備份複寫

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds start-db-instance-automated-backups-replication \  
--region us-east-1 \  
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \  
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" \  
--backup-retention-period 7
```

在 Windows 中：

```
aws rds start-db-instance-automated-backups-replication ^  
--region us-east-1 ^  
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^  
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" ^  
--backup-retention-period 7
```

在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 區域之間加密備份時，需要使用此 `--source-region` 選項。若為 `--source-region`，請指定來源資料庫執行個體的 AWS 區域。

若未指定 `--source-region`，請務必指定 `--pre-signed-url` 值。presigned URL (預先簽章的 URL) 為包含對來源 AWS 區域中呼叫 `start-db-instance-automated-backups-replication` 命令之 Signature 第 4 版簽章請求的 URL。若要瞭解有關 `pre-signed-url` 此選項的更多資訊，請參閱《命令參考》中的 [start-db-instance-automated-backup-複寫](#)。AWS CLI

RDS API

使用具有下列參數的 [StartDBInstanceAutomatedBackupsReplication](#) RDS API 操作啟用備份複寫：

- Region
- SourceDBInstanceArn
- BackupRetentionPeriod
- KmsKeyId (選用)
- PreSignedUrl (如果使用 KmsKeyId 的話則為必要)

Note

如果加密備份，也必須包含預先簽章的 URL。如需有關預先簽章 URL 的詳細資訊，請參閱 Amazon Simple Storage Service API 參考中的[身分驗證請求：使用查詢參數 \(AWS 簽章第 4 版\)](#) 和 AWS 一般參考中的[簽章第 4 版簽署程序](#)。

尋找複寫備份的相關資訊

您可以使用下列 CLI 命令來尋找複寫備份的相關資訊：

- [describe-source-regions](#)
- [describe-db-instances](#)
- [describe-db-instance-automated-backups](#)

下列 describe-source-regions 範例列出 AWS 區域 可將自動備份複製到美國西部 (奧勒岡) 目的地區域的來源。

顯示來源區域的相關資訊

- 執行下列命令。

```
aws rds describe-source-regions --region us-west-2
```

輸出顯示備份可從 US East (N. Virginia) (但不能從美國東部 (俄亥俄) 或 美國西部 (加州北部)) 複寫到美國西部 (奧勒岡)。

```
{  
  "SourceRegions": [  

```

```
...
{
  "RegionName": "us-east-1",
  "Endpoint": "https://rds.us-east-1.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "us-east-2",
  "Endpoint": "https://rds.us-east-2.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": false
},
{
  "RegionName": "us-west-1",
  "Endpoint": "https://rds.us-west-1.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": false
}
]
```

下列 `describe-db-instances` 範例顯示資料庫執行個體的自動備份。

顯示資料庫執行個體的複寫備份

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-instances \
--db-instance-identifier mydatabase
```

在 Windows 中：

```
aws rds describe-db-instances ^
--db-instance-identifier mydatabase
```

輸出包括複寫的備份。

```
{
  "DBInstances": [
```

```

    {
      "StorageEncrypted": false,
      "Endpoint": {
        "HostedZoneId": "Z1PVIIF0B656C1W",
        "Port": 1521,
        ...

      "BackupRetentionPeriod": 7,
      "DBInstanceAutomatedBackupsReplications":
      [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7H0J4SIEXAMPLE"}]
    }
  ]
}

```

下列 `describe-db-instance-automated-backups` 範例顯示資料庫執行個體的自動備份。

顯示資料庫執行個體的自動備份

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-instance-automated-backups \
--db-instance-identifier mydatabase
```

在 Windows 中：

```
aws rds describe-db-instance-automated-backups ^
--db-instance-identifier mydatabase
```

輸出會顯示來源資料庫執行個體和 美國西部 (奧勒岡) 中的自動備份，並將備份複寫到 US East (N. Virginia)。

```

{
  "DBInstanceAutomatedBackups": [
    {
      "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
      "DbiResourceId": "db-L2IJCEXJP7XQ7H0J4SIEXAMPLE",
      "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE",

```

```

        "BackupRetentionPeriod": 7,
        "DBInstanceAutomatedBackupsReplications":
    [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7H0J4SIEXAMPLE"}]
        "Region": "us-west-2",
        "DBInstanceIdentifier": "mydatabase",
        "RestoreWindow": {
            "EarliestTime": "2020-10-26T01:09:07Z",
            "LatestTime": "2020-10-31T19:09:53Z",
        }
        ...
    }
}

```

下列 `describe-db-instance-automated-backups` 範例會使用 `--db-instance-automated-backups-arn` 選項顯示目的地區域中的複寫備份。

顯示複寫的備份

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-instance-automated-backups \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE"
```

在 Windows 中：

```
aws rds describe-db-instance-automated-backups ^
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE"
```

輸出會顯示 美國西部 (奧勒岡) 中的來源資料庫執行個體，以及 US East (N. Virginia) 中的複寫備份。

```

{
  "DBInstanceAutomatedBackups": [
    {
      "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
      "DbiResourceId": "db-L2IJCEXJP7XQ7H0J4SIEXAMPLE",
    }
  ]
}

```

```
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE",
    "Region": "us-west-2",
    "DBInstanceIdentifier": "mydatabase",
    "RestoreWindow": {
        "EarliestTime": "2020-10-26T01:09:07Z",
        "LatestTime": "2020-10-31T19:01:23Z"
    },
    "AllocatedStorage": 50,
    "BackupRetentionPeriod": 7,
    "Status": "replicating",
    "Port": 1521,
    ...
}
]
```

從複寫備份還原至指定的時間

您可以使用 Amazon RDS 主控台從複寫備份將資料庫執行個體還原至特定的時間點。您也可以使用 `restore-db-instance-to-point-in-time` AWS CLI 命令或 `RestoreDBInstanceToPointInTime` RDS API 作業。

如需 point-in-time 復原 (PITR) 的一般資訊，請參閱[將資料庫執行個體還原至指定的時間](#)。

Note

在適用於 SQL Server 的 RDS 上，複寫自動備份 AWS 區域時，不會複製選項群組。如果您已將自訂選項群組與 RDS for SQL Server 資料庫執行個體相關聯，則可以在目的地區域中重新建立該選項群組。然後還原目的地區域中的資料庫執行個體，並將自訂選項群組與其關聯。如需詳細資訊，請參閱[使用選項群組](#)。

主控台

從複寫備份將資料庫執行個體還原至指定的時間

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從區域選取器中選擇目的地區域 (將備份複製到的位置)。
3. 在導覽窗格中，選擇 Automated backups (自動備份)。

4. 在 Replicated backups (複寫備份) 標籤上，選擇要還原的資料庫執行個體。
5. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。
6. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇 Custom (自訂)，請輸入您希望執行個體還原到什麼日期和時間。

Note

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

7. 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入目標還原資料庫執行個體的名稱。
8. (選用) 視需要選擇其他選項，例如啟用自動調整規模。
9. 選擇 Restore to point in time (還原至時間點)。

AWS CLI

使用指 [restore-db-instance-to-point-in-time](#) AWS CLI 令建立新的資料庫執行個體。

從複寫備份將資料庫執行個體還原至指定的時間

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-automated-backups-arn "arn:aws:rds:us-  
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE" \  
  --target-db-instance-identifier mytargetdbinstance \  
  --restore-time 2020-10-14T23:45:00.000Z
```

在 Windows 中：

```
aws rds restore-db-instance-to-point-in-time ^  
  --source-db-instance-automated-backups-arn "arn:aws:rds:us-  
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE" ^  
  --target-db-instance-identifier mytargetdbinstance ^  
  --restore-time 2020-10-14T23:45:00.000Z
```

RDS API

若要將資料庫執行個體還原至指定的時間，請搭配下列參數呼叫 [RestoreDBInstanceToPointInTime](#) Amazon RDS API 操作：

- SourceDBInstanceAutomatedBackupsArn
- TargetDBInstanceIdentifier
- RestoreTime

停用自動備份複寫

您可以使用 Amazon RDS 主控台停止資料庫執行個體的備份複寫。您也可以使用 `stop-db-instance-automated-backups-replication` AWS CLI 命令或 `StopDBInstanceAutomatedBackupsReplication` RDS API 作業。

複寫備份的保留時間將視建立時所設定的備份保留期限而定。

主控台

從來源區域的 Automated backups (自動備份) 頁面停止備份複寫。

若要停止備份複寫至 AWS 區域

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從 Region selector (區域選取器) 選擇來源區域。
3. 在導覽窗格中，選擇 Automated backups (自動備份)。
4. 在 Current Region (目前區域) 標籤上，選擇您要停止備份複寫的資料庫執行個體。
5. 針對 Actions (動作)，選擇 Manage cross-Region replication (管理跨區域複寫)。
6. 在 Backup replication (備份複寫) 下，清除 Enable replication to another AWS 區域 (啟用複寫至另一個 AWS 區域) 核取方塊。
7. 選擇 Save (儲存)。

複寫的備份會列在目的地區域 Automated backups (自動備份) 頁面的 Retained (保留) 標籤上。

AWS CLI

使用 [stop-db-instance-automated-backups-replication](#) AWS CLI 命令停止備份複寫。

下列 CLI 範例會停止在 美國西部 (奧勒岡) 區域中複寫資料庫執行個體的自動備份。

停止備份複寫

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds stop-db-instance-automated-backups-replication \  
--region us-east-1 \  
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

在 Windows 中：

```
aws rds stop-db-instance-automated-backups-replication ^  
--region us-east-1 ^  
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

RDS API

使用具有下列參數的 [StopDBInstanceAutomatedBackupsReplication](#) RDS API 操作停止備份複寫：

- Region
- SourceDBInstanceArn

刪除複寫的備份

您可以使用 Amazon RDS 主控台刪除資料庫執行個體的複寫備份。您也可以使用delete-db-instance-automated-backups AWS CLI 命令或 DeleteDBInstanceAutomatedBackup RDS API 作業。

主控台

從 Automated backups (自動備份) 頁面刪除目的地區域中複製的備份。

刪除複寫的備份

- 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 從 Region selector (區域選取器) 選擇目的地區域。
3. 在導覽窗格中，選擇 Automated backups (自動備份)。
4. 在 Replicated backups (複寫備份) 標籤上，選擇要刪除複寫備份的資料庫執行個體。
5. 對於 Actions (動作)，請選擇 Delete (刪除)。
6. 在確認頁面上，輸入 **delete me**，然後選擇 Delete (刪除)。

AWS CLI

使用 [delete-db-instance-automated-backup](#) AWS CLI 命令刪除複製的備份。

您可以使用 [describe-db-instances](#) CLI 命令來尋找複寫備份的 Amazon Resource Name (ARN)。如需詳細資訊，請參閱 [尋找複寫備份的相關資訊](#)。

刪除複寫的備份

- 請執行下列其中一個命令：

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance-automated-backup \  
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-  
backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE"
```

在 Windows 中：

```
aws rds delete-db-instance-automated-backup ^  
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-  
backup:ab-L2IJCEXJP7XQ7H0J4SIEXAMPLE"
```

RDS API

使用 [DeleteDBInstanceAutomatedBackup](#) RDS API 操作搭配 `DBInstanceAutomatedBackupsArn` 參數刪除複寫的備份。

管理手動備份

本節說明如何管理資料庫執行個體和資料庫叢集的手動備份。

主題

- [為單一可用區資料庫執行個體建立資料庫快照](#)
- [建立多可用區域資料庫叢集快照](#)
- [刪除資料庫快照](#)

為單一可用區資料庫執行個體建立資料庫快照

Amazon RDS 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。在單一可用區資料庫執行個體中建立資料庫快照會導致輸入/輸出短暫停用；且根據資料庫執行個體的大小和類別，停用時間會持續幾秒鐘至幾分鐘。若為 MariaDB、MySQL、Oracle 和 PostgreSQL，在備份異地同步備份部署期間，您的主要執行個體上不會暫停輸入/輸出活動，因為備份取自於待命執行個體。若為 SQL Server，在備份異地同步備份部署期間會短暫暫停輸入/輸出活動。

建立資料庫快照時，您必須找出要進行備份的資料庫執行個體，並為該資料庫快照命名，使得您稍後可透過它進行還原。建立快照所需的時間量因資料庫的大小而異。由於快照包括整個儲存體磁碟區，檔案大小，例如暫存檔案，也會影響建立快照所需的時間量。

Note

您的資料庫執行個體必須處於 available 狀態，才能取得資料庫快照。
對於 PostgreSQL 資料庫執行個體，未記錄資料表中的資料可能無法從快照中還原。如需更多詳細資訊，請參閱 [使用 PostgreSQL 的最佳實務](#)。

與自動備份不同，手動快照不受備份保留期限的限制。快照不會過期。

針對非常長期的 MariaDB、MySQL 和 PostgreSQL 資料備份，建議您將快照資料匯出至 Amazon S3。如果資料庫引擎的主要版本不再受到支援，您則無法從快照還原至該版本。如需詳細資訊，請參閱 [將資料庫快照資料匯出至 Amazon S3](#)。

您可以使用 AWS Management Console、或 RDS API 建立資料庫快照集。AWS CLI

主控台

建立資料庫快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。

手動快照清單即會出現。

3. 選擇 Take Snapshot (擷取快照)。

Take DB Snapshot (建立資料庫快照) 視窗隨即顯示。

4. 選擇您要拍攝快照的資料庫執行個體。
5. 輸入快照名稱。
6. 選擇 Take Snapshot (擷取快照)。

[手動快照] 清單隨即出現，新資料庫快照的狀態顯示為Creating。之後，其狀態為 Available，您可以看到其建立時間。

AWS CLI

使用建立資料庫快照時 AWS CLI，您需要識別要備份的資料庫執行個體，然後為您的資料庫快照指定一個名稱，以便稍後從中還原。您可以使用具有以下參數的 AWS CLI [create-db-snapshot](#) 命令來執行此操作：

- `--db-instance-identifier`
- `--db-snapshot-identifier`

在此範例中，您會為稱為 *mydbinstance* 的資料庫執行個體建立稱為 *mydbsnapshot* 的資料庫快照。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-snapshot \  
  --db-instance-identifier mydbinstance \  
  --db-snapshot-identifier mydbsnapshot
```

在Windows中：

```
aws rds create-db-snapshot ^  
  --db-instance-identifier mydbinstance ^  
  --db-snapshot-identifier mydbsnapshot
```

RDS API

使用 Amazon RDS API 建立資料庫快照時，您必須找出要進行備份的資料庫執行個體，並為該資料庫快照命名，使得您稍後可透過它進行還原。您可以使用 Amazon RDS API [CreateDBSnapshot](#) 命令，並搭配下列參數來執行此動作：

- DBInstanceIdentifier
- DBSnapshotIdentifier

建立多可用區域資料庫叢集快照

當您建立多可用區域資料庫叢集快照時，請務必找出要進行備份的多可用區域資料庫叢集，並為該資料庫叢集快照命名，以便您稍後可透過它進行還原。您也可以共用多可用區域資料庫叢集快照。如需說明，請參閱[the section called “共享資料庫 快照”](#)。

您可以使用 AWS Management Console、或 RDS API 建立異地同步備份資料庫叢集快照。AWS CLI 主控台

建立資料庫叢集快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 在清單中，選擇您要擷取快照的多可用區域資料庫叢集。
4. 針對 Actions (動作)，選擇 Take snapshot (建立快照)。

Take DB Snapshot (建立資料庫快照) 視窗隨即顯示。

5. 在 Snapshot name (快照名稱) 中輸入快照的名稱。
6. 選擇 Take Snapshot (擷取快照)。

Snapshots (快照) 頁面隨即顯示，且新的多可用區域資料庫叢集快照的狀態顯示為 Creating。之後，其狀態為 Available，你可以看到其建立時間。

AWS CLI

您可以使用具有下列選項的 AWS CLI [create-db-cluster-snapshot](#) 指令來建立異地同步備份資料庫叢集快照：

- `--db-cluster-identifier`
- `--db-cluster-snapshot-identifier`

在此範例中，您為名為 *mymulti-az-db-clustersnapshot* 的資料庫叢集建立名為 *mymulti-az-cluster* 的多可用區域資料庫叢集快照。

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-db-cluster-snapshot \  
  --db-cluster-identifier mymultiazdbcluster \  
  --db-cluster-snapshot-identifier mymultiazdbclustersnapshot
```

在Windows中：

```
aws rds create-db-cluster-snapshot ^  
  --db-cluster-identifier mymultiazdbcluster ^  
  --db-cluster snapshot-identifier mymultiazdbclustersnapshot
```

RDS API

您可以使用具有下列參數的 Amazon RDS API 建立資料庫ClusterSnapshot操作來建立[異地同步備份資料庫叢集快照](#)：

- DBClusterIdentifier
- DBClusterSnapshotIdentifier

刪除多可用區域資料庫叢集快照

當您不再需要時，可以刪除由 Amazon RDS 管理的多可用區域資料庫快照。如需說明，請參閱 [the section called “刪除資料庫快照”](#)。

刪除資料庫快照

當您不再需要時，可以刪除由 Amazon RDS 管理的資料庫快照。

Note

使用 AWS Backup 主控台，刪除由 AWS Backup 管理的備份。如需 AWS Backup 的詳細資訊，請參閱《[AWS Backup 開發人員指南](#)》。

刪除資料庫快照

您可以使用 AWS Management Console、AWS CLI、或 RDS API 來刪除手動、共用或公開的資料庫快照。

要刪除共用或公開的快照，您必須登入擁有該快照的 AWS 帳戶。

如果您想刪除自動資料庫快照，而保留資料庫執行個體，請更改資料庫執行個體的備份保留週期為 0。當更改套用時，自動快照會被刪除。如果您不想等到下次的維護週期，您可以立即套用更改。更改完成之後，您可以透過設定備份週期的數值大於 0，重新啟用自動備份。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

保留的自動備份和手動快照會產生帳單費用，直到將其刪除為止。如需詳細資訊，請參閱[保留成本](#)。

如果您刪除資料庫執行個體，您可以透過移除自動備份資料庫執行個體來刪除自動資料庫快照。如需自動備份的相關資訊，請參閱[備份簡介](#)。

主控台

刪除資料庫快照

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。手動快照清單即會出現。
3. 選擇想要刪除的資料庫快照。
4. 針對 Actions (動作)，選擇 Delete Snapshot (刪除快照)。
5. 在確認頁面上，選擇 Delete (刪除)。

AWS CLI

您可以使用AWS CLI指令刪除資料庫快照[delete-db-snapshot](#)。

下列選項用來刪除資料庫快照。

- `--db-snapshot-identifier` – 資料庫快照的識別符。

Example

下列程式碼會刪除 `mydbsnapshot` 資料庫快照。

對於LinuxmacOS、或Unix：

```
aws rds delete-db-snapshot \  
  --db-snapshot-identifier mydbsnapshot
```

在Windows中：

```
aws rds delete-db-snapshot ^  
  --db-snapshot-identifier mydbsnapshot
```

RDS API

您可以使用 Amazon RDS API 操作 [DeleteDBSnapsho \(刪除資料庫快照\)](#) 來刪除資料庫快照。

下列參數用來刪除資料庫快照。

- `DBSnapshotIdentifier` – 資料庫快照的識別符。

從資料庫快照還原

本節說明如何從資料庫快照還原。

主題

- [參數群組考量](#)
- [安全群組考量](#)
- [選項群組考量](#)
- [資源標記考量事項](#)
- [Db2 考量](#)
- [Microsoft SQL Server 考量](#)
- [Oracle 資料庫考量事項](#)
- [從快照還原](#)
- [將資料庫執行個體還原至指定的時間](#)
- [將多可用區域資料庫叢集還原至指定時間](#)
- [從快照還原至多可用區域資料庫叢集](#)
- [從多可用區域資料庫叢集快照還原至資料庫執行個體](#)
- [教學課程：從資料庫快照還原 Amazon RDS 資料庫執行個體](#)

Amazon RDS 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。您可從資料庫快照還原來建立新的資料庫執行個體。您提供要從中還原之資料庫快照的名稱，然後提供一個從還原建立之新資料庫執行個體的名稱。您無法從資料庫快照還原為現有的資料庫執行個體；還原時會建立新的資料庫執行個體。

若還原的資料庫執行個體狀態為 `available`，您便可使用該執行個體。資料庫執行個體會背景繼續載入資料。這就是通常所稱的延遲載入。

如果您存取尚未載入的資料，資料庫執行個體會立即從 Amazon S3 下載所請求的資料，然後繼續在背景載入剩餘的資料。如需更多資訊，請參閱 [Amazon EBS 快照](#)。

若要協助緩解延遲載入對要快速存取之資料表的影響，您可以執行涉及完整資料表掃描的作業，例如 `SELECT *`。這會允許 Amazon RDS 從 S3 下載所有已備份的資料表資料。

您可以還原資料庫快照，並使用與來源資料庫快照不同的儲存類型。在此情況下，因為需要將資料遷移至新儲存類型的額外工作，使得還原程序較緩慢。如果您在磁帶儲存體間來回進行還原，遷移程序是最緩慢的。那是因為磁帶儲存體沒有佈建 IOPS 或一般用途 (SSD) 儲存體的 IOPS 功能。

您可以使用 AWS CloudFormation 從資料庫執行個體快照還原資料庫執行個體。如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的 [AWS::RDS::DBInstance](#)。

Note

您無法從同時共用和加密的資料庫快照還原資料庫執行個體。不過，您可以複製資料庫快照，再從副本還原資料庫執行個體。如需詳細資訊，請參閱 [複製資料庫快照](#)。

如需使用 RDS 延伸 Support 版本還原資料庫執行個體的相關資訊，請參閱 [使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)。

參數群組考量

建議您針對您建立的任何資料庫快照保留資料庫參數群組，這樣才能為還原的資料庫執行個體與正確的參數群組建立關聯。

預設的資料庫參數群組會與還原的執行個體建立關聯，除非您選擇不同執行個體。預設參數群組中沒有可用的自訂參數設定。

還原資料庫執行個體時，可以指定參數群組。

如需資料庫參數群組的詳細資訊，請參閱 [使用參數群組](#)。

安全群組考量

還原資料庫執行個體時，預設的虛擬私有雲端 (VPC)、資料庫子網路群組和 VPC 安全群組會與還原的執行個體建立關聯，除非您選擇不同的執行個體。

- 若您使用 Amazon RDS 主控台，可指定要與執行個體建立關聯的自訂 VPC 安全群組，或建立新的 VPC 安全群組。
- 如果您使用的是 AWS CLI，可以在 `restore-db-instance-from-db-snapshot` 命令中加入 `--vpc-security-group-ids` 選項，以指定要與執行個體關聯的自訂 VPC 安全性群組。
- 如果您是使用 Amazon RDS API，則可以在 `VpcSecurityGroupIds.VpcSecurityGroupId.N` 動作中包括 `RestoreDBInstanceFromDBSnapshot` 參數。

一旦還原完成且新的資料庫執行個體可用，您還可修改資料庫執行個體來變更 VPC 設定。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

選項群組考量

還原資料庫執行個體時，多數情況下預設的資料庫選項群組會與還原的資料庫執行個體建立關聯。

例外情況是來源資料庫執行個體與包含持續性或永久選項的選項群組相關聯時。例如，如果來源資料庫執行個體使用 Oracle 透明資料加密 (TDE)，還原的資料庫執行個體必須使用具有 TDE 選項的選項群組。

如果將資料庫執行個體還原至不同的 VPC 中，您必須執行以下其中一項操作以指派資料庫選項群組：

- 將該 VPC 群組的預設選項群組指派給執行個體。
- 指派與該 VPC 連結的另一個選項群組。
- 建立新的選項群組，並將其指派至資料庫執行個體。若使用 Oracle TDE 等持續性或永久選項，您必須建立包含持續性或永久選項的新選項群組。

如需資料庫選項群組的詳細資訊，請參閱 [使用選項群組](#)。

資源標記考量事項

從資料庫快照還原資料庫執行個體時，RDS Proxy 會檢查您是否指定了新標籤。如果有指定，便會將新標籤新增到還原的資料庫執行個體中。如果沒有新標籤，RDS 會在建立快照時將來源資料庫執行個體中的標籤新增到還原的資料庫執行個體。

如需詳細資訊，請參閱 [將標籤複製到資料庫快照](#)。

Db2 考量

使用 BYOL 模型時，您的 Amazon RDS for Db2 資料庫執行個體必須與包含您的和您 IBM Site ID 的 IBM Customer ID 否則，嘗試從快照還原資料庫執行個體將會失敗。如需詳細資訊，請參閱 [為 Db2 帶來您自己的授權](#) 及 [rdsadmin.restore_database](#)。

使用 Db2 授權直通 AWS Marketplace 模式時，您需要針對要使用的特定 IBM Db2 版本提供有效的 AWS Marketplace 訂閱。如果您還沒有，請在該 IBM Db2 版本中 [AWS Marketplace 訂閱 Db2](#)。如需詳細資訊，請參閱 [通過 Db2 許可證 AWS Marketplace](#)。

Microsoft SQL Server 考量

將 RDS for Microsoft SQL Server 資料庫快照還原至新執行個體時，您一律可還原至與您的快照相同的版本。在某些情況下，您也可以變更資料庫執行個體的版本。下列限制在您變更版本時可加以套用：

- 資料庫快照必須具有針對新版本配置的足夠儲存體。
- 僅支援下列版本變更：
 - 從 Standard Edition 至 Enterprise Edition
 - 從 Web Edition 至 Standard Edition 或 Enterprise Edition
 - 從 Express Edition 至 Web Edition、Standard Edition 或 Enterprise Edition

如果您想要透過還原快照，從某個版本變更為不支援的新版本，您可嘗試使用原生的備份和還原功能。SQL Server 會根據您在資料庫上啟用的 SQL Server 功能來驗證您的資料庫是否與新版本相容。如需詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

Oracle 資料庫考量事項

當您從資料庫快照還原 Oracle 資料庫時，請考慮下列事項：

- 在還原資料庫快照之前，您可以將其升級至更新的 Oracle 資料庫版本。如需詳細資訊，請參閱 [升級 Oracle 資料庫快照](#)。
- 如果您還原使用單一租戶組態之 CDB 執行個體的快照，則可以變更 PDB 名稱。當 CDB 執行個體使用多租戶組態時，您無法變更 PDB 名稱。如需詳細資訊，請參閱 [備份與還原 CDB](#)。
- 您不能更改 CDB 名稱，其始終是 RDSCDB。對於所有 CDB 執行個體，此 CDB 名稱都相同。
- 您無法直接與資料庫快照中的租戶資料庫互動。如果您還原使用多租戶組態之 CDB 執行個體的快照，則會還原其所有租戶資料庫。您可以使用 [describe-db-snapshot-tenant-databases](#)，在還原資料庫快照內的租戶資料庫之前先行檢查。
- 如果您使用 Oracle GoldenGate，請始終保留參數群組和 compatible 參數。當您從資料庫快照還原資料庫執行個體時，請指定具有相符或更大 compatible 值的一個參數群組。
- 您可以選擇在還原資料庫快照時重新命名資料庫。如果線上重做日誌的總大小大於 20GB，RDS 可能會將您的線上重做日誌大小重設為預設值 512MB (4 x 128MB)。較小的大小允許還原操作在合理的時間內完成。您可以稍後重新建立線上重做日誌並變更大小。

從快照還原

您可以使用 AWS Management Console、或 RDS API 從資料庫快照還原資 AWS CLI 料庫執行個體。

Note

您無法在還原資料庫執行個體時減少儲存數體的數量。當您嘗試增加配置儲存時，其必須至少是 10%。如果您嘗試以低於 10% 的值增加該值，便會發生錯誤。您無法在還原 RDS for SQL Server 資料庫執行個體時增加已配置的儲存體。

主控台

從資料庫快照還原資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。
5. 在 Restore snapshot (還原快照) 頁面上，對 DB Instance Identifier (資料庫執行個體識別符)，輸入您還原的資料庫執行個體名稱。
6. 指定其他設定，例如配置的儲存體大小。

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

7. 選擇 Restore DB instance (還原資料庫執行個體)。

AWS CLI

若要從資料庫快照還原資料庫執行個體，請使用 AWS CLI 指令從數據庫快照恢復-DB-執行個體。

在此範例中，您會從先前建立、名稱為 mydbsnapshot 的資料庫快照還原。您會還原至名稱為 mynewdbinstance 的新資料庫執行個體。此範例也會設定配置的儲存體大小。

您可以指定其他設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier mynewdbinstance \  
  --storage-size 100
```

```
--db-snapshot-identifier mydbsnapshot \  
--allocated-storage 100
```

在 Windows 中：

```
aws rds restore-db-instance-from-db-snapshot ^  
--db-instance-identifier mynewdbinstance ^  
--db-snapshot-identifier mydbsnapshot ^  
--allocated-storage 100
```

此命令會傳回類似以下的輸出：

```
DBINSTANCE mynewdbinstance db.t3.small MySQL 50 sa creating  
3 n 8.0.28 general-public-license
```

RDS API

若要從資料庫快照還原資料庫執行個體，請使用下列參數呼叫 Amazon RDS API 函數 [恢復資料 InstanceFrom庫快照](#)：

- DBInstanceIdentifier
- DBSnapshotIdentifier

將資料庫執行個體還原至指定的時間

您可以將資料庫執行個體還原到特定的時間點，建立新的資料庫執行個體而無需修改來源資料庫執行個體。

將資料庫執行個體還原至某個時間點時，您可以選擇預設的 Virtual Private Cloud (VPC) 安全群組。或者，可以將自訂 VPC 安全群組套用至您的資料庫執行個體。

還原的資料庫執行個體會自動與預設資料庫參數和選項群組產生關聯。不過，您可以在還原期間指定要套用的自訂參數群組和選項群組。

如果來源資料庫執行個體具有資源標籤，RDS 會將最新的標籤新增至還原的資料庫執行個體。

RDS 每五分鐘會將資料庫執行個體的交易日誌上傳至 Amazon S3。若要查看資料庫執行個體的最新可還原時間，請使用 AWS CLI [描述be-db-instance](#) 命令，並查看資料庫執行個體在欄位中傳回的 LatestRestorableTime 值。若要在 Amazon RDS 主控台中查看每個資料庫執行個體的最新可還原時間，請選擇自動備份。

您可以還原至備份保留期間內的任何時間點。若要查看每個資料庫執行個體的最早可還原時間，請在 Amazon RDS 主控台中選擇自動備份。

DB Name	Earliest restorable time	Latest restorable time	Engine	Encrypted
database-1	December 27th 2020, 9:42:48 am UTC	January 4th 2021, 6:25:01 pm UTC	sqlserver-se	No
database-1-sast	December 31st 2020, 9:18:52 am UTC	January 8th 2021, 2:44:01 pm UTC	sqlserver-ex	No
database-2	December 24th 2020, 11:38:43 am UTC	January 8th 2021, 2:46:01 pm UTC	sqlserver-se	Yes
database-3	December 31st 2020, 9:51:23 am UTC	January 8th 2021, 2:43:01 pm UTC	sqlserver-ex	No
database-6	December 31st 2020, 6:54:19 am UTC	January 8th 2021, 2:42:01 pm UTC	sqlserver-ex	No
database-7	January 1st 2021, 12:21:52 pm UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No
db4-5640	January 4th 2021, 7:11:04 pm UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No
myorclinstance-from-replicated-backup	December 24th 2020, 7:49:18 am UTC	January 8th 2021, 2:47:57 pm UTC	oracle-se2	No
test2-mysql-mag-maz	January 6th 2021, 6:42:52 am UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No

Note

建議您還原到相同或類似的資料庫執行個體大小，而且如果使用佈建 IOPS 儲存體做為來源資料庫執行個體，則包括 IOPS。例如，如果您選擇 IOPS 值不相容的資料庫執行個體大小，則可能會收到錯誤訊息。

如需使用 RDS 延伸 Support 版本還原資料庫執行個體的相關資訊，請參閱[使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)。

從時間點還原時，Amazon RDS 所使用的一些資料庫引擎有特殊考量。

- 如果您對 Amazon RDS for Db2 資料庫執行個體使用密碼身份驗證，則不會在日誌中擷取使用者管理動作 (包括 `rdsadmin.add_user`)。這些動作需要完整快照備份。

使用 BYOL 模型時，您的 RDS for Db2 資料庫執行個體必須與包含您 IBM Site ID 的和 IBM Customer ID 否則，嘗試將資料庫執行個體還原到特定時間點將會失敗。如需詳細資訊，請參閱 [為 Db2 帶來您自己的授權](#) 及 [rdsadmin.restore_database](#)。

使用 Db2 授權直通 AWS Marketplace 模式時，您需要針對要使用的特定 IBM Db2 版本提供有效的 AWS Marketplace 訂閱。如果您還沒有，請在該 IBM Db2 版本 [中 AWS Marketplace 訂閱 Db2](#)。如需詳細資訊，請參閱 [通過 Db2 許可證 AWS Marketplace](#)。

- 將 Oracle 資料庫執行個體還原至時間點時，您可以指定要由新的資料庫執行個體使用的 Oracle 資料庫引擎、授權模式和 DBName (SID)。
- 當您將 Microsoft SQL Server 資料庫執行個體還原至時間點時，該執行個體內的每個資料庫都會還原至執行個體內每個其他資料庫 1 秒內的時間點。可能無法一致地還原橫跨執行個體內多個資料庫的交易。
- 對於 SQL Server 資料庫執行個體，不支援 OFFLINE、EMERGENCY 和 SINGLE_USER 模式。將任何資料庫設為其中一種模式會導致整個資料庫執行個體的最新可還原時間停止向前移動。
- 某些動作 (例如變更 SQL Server 資料庫的復原模式) 可能會中斷用於 point-in-time 復原的記錄檔順序。在某些情況下，Amazon RDS 可以偵測到這個問題，並防止最近的可還原時間繼續前進。在其他情況下，例如當 SQL Server 資料庫使用 BULK_LOGGED 復原模式時，不會偵測到記錄順序中的中斷。如果日誌順序發生中斷，可能無法將 SQL Server 資料庫執行個體還至時間點。基於這些原因，Amazon RDS 不支援變更 SQL Server 資料庫的復原模型。

您也可以使用 AWS Backup 來管理 Amazon RDS 資料庫執行個體的備份。如果您的資料庫執行個體與中的備份計劃相關聯 AWS Backup，則會使用該備份計劃進行 point-in-time 復原。以 AWS Backup

建立的備份名稱結尾為 `awsbackup:AWS-Backup-job-number`。如需相關資訊 AWS Backup，請參閱 [AWS Backup 發人員指南](#)。

Note

本主題中的資訊適用於 Amazon RDS。如需還原 Amazon Aurora 資料庫叢集的資訊，請參閱 [將資料庫叢集還原至指定的時間](#)。

您可以使用 AWS Management Console、或 RDS API 將資料庫執行個體還原到 AWS CLI 某個時間點。

Note

您無法在還原資料庫執行個體時減少儲存數體的數量。當您嘗試增加配置儲存時，其必須至少是 10%。如果您嘗試以低於 10% 的值增加該值，便會發生錯誤。您無法在還原 RDS for SQL Server 資料庫執行個體時增加已配置的儲存體。

主控台

將資料庫執行個體還原至指定時間

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。自動備份顯示在 Current Region (目前區域) 索引標籤上。
3. 選擇您想要還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。

Restore to point in time (還原到時間點) 視窗隨即出現。

5. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇 Custom (自訂)，請輸入您希望執行個體叢集還原到什麼日期和時間。

Note

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

- 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入目標還原資料庫執行個體的名稱。名稱必須是唯一的。
- 視需要選擇其他選項，例如資料庫執行個體類別、儲存體以及是否要使用儲存體自動擴充功能。

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

- 選擇 Restore to point in time (還原至時間點)。

AWS CLI

若要將資料庫執行個體還原到指定的時間，請使用指 AWS CLI 令 [恢復-db-Instance 到時間點來建立新的資料庫執行個體](#)。此範例也會設定配置的儲存體大小，並啟用儲存體自動擴展。

此操作支援資源標記。使用 `--tags` 選項時，會忽略來源資料庫執行個體標籤，並使用提供的標籤。否則，會使用來源執行個體中的最新標籤。

您可以指定其他設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier mysourcedbinstance \  
  --target-db-instance-identifier mytargetdbinstance \  
  --restore-time 2017-10-14T23:45:00.000Z \  
  --allocated-storage 100 \  
  --max-allocated-storage 1000
```

在 Windows 中：

```
aws rds restore-db-instance-to-point-in-time ^  
  --source-db-instance-identifier mysourcedbinstance ^  
  --target-db-instance-identifier mytargetdbinstance ^  
  --restore-time 2017-10-14T23:45:00.000Z ^
```

```
--allocated-storage 100 ^  
--max-allocated-storage 1000
```

RDS API

若要將資料庫執行個體還原至指定的時間，請搭配下列參數呼叫 Amazon RDS API

[RestoreDBInstanceToPointInTime](#) 操作：

- SourceDBInstanceIdentifier
- TargetDBInstanceIdentifier
- RestoreTime

將多可用區域資料庫叢集還原至指定時間

您可將多可用區域資料庫叢集還原到特定的時間點，從而建立新的多可用區域資料庫叢集。

RDS 會持續將多可用區域資料庫叢集的交易日誌上傳至 Amazon S3。您可以還原至備份保留期間內的任何時間點。若要查看異地同步備份資料庫叢集的最早可還原時間，請使用指 AWS CLI [describe-db-clusters](#) 令。查看資料庫叢集的 `EarliestRestorableTime` 欄位中所傳回的值。若要查看多可用區域資料庫叢集的最新可還原時間，請查看資料庫叢集 `LatestRestorableTime` 欄位中傳回的值。

將異地同步備份資料庫叢集還原到某個時間點時，可以為異地同步備份資料庫叢集選擇預設的 VPC 安全群組，或將自訂 VPC 安全群組套用至異地同步備份資料庫叢集。

還原的多可用區域資料庫叢集會自動與預設資料庫叢集參數群組產生關聯。不過，您可以在還原期間指定自訂資料庫叢集參數群組來套用該群組。

如果來源資料庫叢集具有資源標記，則 RDS 會將最新標籤新增至還原的資料庫叢集。

Note

建議您還原到與來源資料庫叢集大小相同或相似的多可用區域資料庫叢集大小。如果使用的是佈建 IOPS 儲存，我們亦建議您使用相同或類似的 IOPS 值進行還原。例如，如果您選擇 IOPS 值不相容的資料庫叢集大小，則可能會收到錯誤訊息。

如果來源異地同步備份資料庫叢集使用一般用途 SSD (gp3) 儲存體，且配置的儲存體少於 400 GiB，則無法修改已還原資料庫叢集的佈建 IOPS。

如需使用 RDS 延伸 Support 版本還原異地同步備份資料庫叢集的相關資訊，請參閱 [使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)。

您可以使用 AWS Management Console、或 RDS API 將異地同步備份資料庫叢集還原到 AWS CLI 某個時間點。

主控台

若要將多可用區域資料庫叢集還原至指定時間

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要還原的多可用區域資料庫叢集。
4. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。

Restore to point in time (還原到時間點) 視窗隨即出現。

5. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇 Custom (自訂)，請輸入您希望多可用區域資料庫叢集還原到什麼日期和時間。

Note

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

6. 對於 DB cluster identifier (資料庫叢集識別符)，輸入還原之多可用區域資料庫叢集的名稱。
7. 在 Availability and durability (可用性與持久性) 中，選擇 Multi-AZ DB cluster (多可用區域資料庫叢集)。

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB cluster**
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance**
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance**
Creates a single DB instance with no standby DB instances.

8. 在 DB instance class (資料庫執行個體類別) 中，選擇資料庫執行個體類別。

目前，多可用區域資料庫叢集僅支援 db.m6gd 和 db.r6gd 資料庫執行個體類別。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

9. 在其餘區段，指定資料庫叢集設定。如需每項設定的相關資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。
10. 選擇 Restore to point in time (還原至時間點)。

AWS CLI

若要將異地同步備份資料庫叢集還原到指定的時間，請使用指 AWS CLI 令 [restore-db-cluster-to-](#) 建立新point-in-time的異地同步備份資料庫叢集。

目前，多可用區域資料庫叢集僅支援 db.m6gd 和 db.r6gd 資料庫執行個體類別。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-cluster-to-point-in-time \  
  --source-db-cluster-identifier mysourcemultiazdbcluster \  
  --db-cluster-identifier mytargetmultiazdbcluster \  
  --restore-to-time 2021-08-14T23:45:00.000Z \  
  --db-cluster-instance-class db.r6gd.xlarge
```

在 Windows 中：

```
aws rds restore-db-cluster-to-point-in-time ^  
  --source-db-cluster-identifier mysourcemultiazdbcluster ^  
  --db-cluster-identifier mytargetmultiazdbcluster ^  
  --restore-to-time 2021-08-14T23:45:00.000Z ^  
  --db-cluster-instance-class db.r6gd.xlarge
```

RDS API

若要將資料庫叢集還原到指定的時間，請使用下列參數呼叫 Amazon RDS API 還原 [B ClusterToPointInTime](#) 操作：

- SourceDBClusterIdentifier
- DBClusterIdentifier
- RestoreToTime

從快照還原至多可用區域資料庫叢集

您可以使用 AWS Management Console、或 RDS API 將快照還原到異地同步備份資料庫叢集。AWS CLI 您可以將這些類型的快照還原至多可用區域資料庫叢集：

- 單一可用區域部署的快照
- 使用單一資料庫執行個體進行異地同步備份資料庫叢集部署的快照
- 多可用區域資料庫叢集的快照

如需多可用區域部署的相關資訊，請參閱 [設定及管理多可用區部署](#)。

Tip

您可以透過還原快照，將單一可用區部署或異地同步備份資料庫叢集部署移轉至異地同步備份資料庫叢集部署。

如需使用 RDS 延伸 Support 版本還原異地同步備份資料庫叢集的相關資訊，請參閱 [使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)。

主控台

若要將快照還原至多可用區域資料庫叢集

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的快照。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。
5. 在 Restore snapshot (還原快照) 頁面的 Availability and durability (可用性與持久性) 中，選擇 Multi-AZ DB cluster (多可用區域資料庫叢集)。

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB cluster**
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance**
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance**
Creates a single DB instance with no standby DB instances.

6. 對於 DB cluster identifier (資料庫叢集識別符)，輸入還原之多可用區域資料庫叢集的名稱。
7. 在其餘區段，指定資料庫叢集設定。如需每項設定的相關資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。
8. 選擇 Restore DB instance (還原資料庫執行個體)。

AWS CLI

若要將快照還原到異地同步備份資料庫叢集，請使用指 AWS CLI 令 [restore-db-cluster-from-snapshot](#)。

在下例中，您會從先前建立的名稱為 mysnapshot 的快照還原。您會還原至名為 mynewmultiazdbcluster 的新多可用區域資料庫叢集。您也可以指定多可用區域資料庫叢集中資料庫執行個體所使用的資料庫執行個體類別。針對資料庫引擎指定 mysql 或 postgres。

針對 --snapshot-identifier 選項，您可以使用名稱或 Amazon 資源名稱 (ARN) 指定資料庫叢集快照。不過，您只能使用 ARN 指定資料庫快照。

對於 --db-cluster-instance-class 選項，為新的多可用區域資料庫叢集指定資料庫執行個體類別。多可用區域資料庫叢集僅支援特定的資料庫執行個體類別，例如 db.m6gd 和 db.r6gd 資料庫執行個體類別。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

您也可以指定其他選項。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-cluster-from-snapshot \
```

```
--db-cluster-identifier mynewmultiadbcluster \  
--snapshot-identifier mynsnapshot \  
--engine mysql/postgres \  
--db-cluster-instance-class db.r6gd.xlarge
```

在Windows中：

```
aws rds restore-db-cluster-from-snapshot ^  
--db-cluster-identifier mynewmultiadbcluster ^  
--snapshot-identifier mynsnapshot ^  
--engine mysql/postgres ^  
--db-cluster-instance-class db.r6gd.xlarge
```

在還原資料庫叢集之後，您可以將多可用區域資料庫叢集新增至安全群組，此安全群組與您用來建立快照 (如果適用) 的資料庫叢集或資料庫執行個體相關聯。完成此動作會提供先前資料庫叢集或資料庫執行個體的相同功能。

RDS API

若要將快照還原到異地同步備份資料庫叢集，請ClusterFromSnapshot使用下列參數呼叫 RDS API 作業[復原 B](#)：

- DBClusterIdentifier
- SnapshotIdentifier
- Engine

您也可以指定選用參數。

在還原資料庫叢集之後，您可以將多可用區域資料庫叢集新增至安全群組，此安全群組與您用來建立快照 (如果適用) 的資料庫叢集或資料庫執行個體相關聯。完成此動作會提供先前資料庫叢集或資料庫執行個體的相同功能。

從多可用區域資料庫叢集快照還原至資料庫執行個體

多可用區域資料庫叢集快照是資料庫叢集的儲存體磁碟區快照，其會備份整個資料庫叢集，而不只是個別的資料庫。您可以將多可用區域資料庫叢集快照還原至單一可用區域部署或多可用區資料庫執行個體部署。如需多可用區域部署的相關資訊，請參閱 [設定及管理多可用區部署](#)。

Note

您也可以將多可用區域資料庫叢集快照還原至新的多可用區域資料庫叢集。如需說明，請參閱 [從快照還原至多可用區域資料庫叢集](#)。

如需使用 RDS 延伸 Support 版本還原異地同步備份資料庫叢集的相關資訊，請參閱 [使用 Amazon RDS 延伸 Support 將資料庫執行個體或異](#)。

使用 AWS Management Console、AWS CLI、或 RDS API 將異地同步備份資料庫叢集快照還原至單一同步備份部署或異地同步備份資料庫執行個體部署。

主控台

將多可用區域資料庫叢集快照還原至單一可用區域部署或多可用區資料庫執行個體部署

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇快照。
3. 選擇想要從中還原的多可用區域資料庫叢集快照。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。
5. 在 Restore snapshot (還原快照) 頁面的 Availability and durability (可用性和持久性) 中，選擇下列其中一項：
 - Single DB instance (單一資料庫執行個體) - 將快照還原到沒有待命資料庫執行個體的資料庫執行個體。
 - Multi-AZ DB instance (多可用區域資料庫執行個體) - 將快照還原至具有一個主要資料庫執行個體和一個待命資料庫執行個體的多可用區域資料庫執行個體部署。
6. 對於 DB instance identifier (資料庫執行個體識別符)，輸入所還原資料庫執行個體的名稱。
7. 在其餘區段，指定資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。
8. 選擇 Restore DB instance (還原資料庫執行個體)。

AWS CLI

[若要將異地同步備份資料庫叢集快照還原至資料庫執行個體部署，請使用 AWS CLI 指令從 db-snapshot 還原-db-執行個體。](#)

在下列範例中，您會從先前建立的多可用區域資料庫叢集快照 (名為 `myclustersnapshot`) 還原。您會還原至主要資料庫執行個體名為 `mynewdbinstance` 的新多可用區域資料庫執行個體部署。對於 `--db-cluster-snapshot-identifier` 選項，指定多可用區域資料庫叢集快照的名稱。

對於 `--db-instance-class` 選項，指定資料庫執行個體類別，進行新的資料庫執行個體部署。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

您也可以指定其他選項。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier mynewdbinstance \  
  --db-cluster-snapshot-identifier myclustersnapshot \  
  --engine mysql \  
  --multi-az \  
  --db-instance-class db.r6g.xlarge
```

在 Windows 中：

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier mynewdbinstance ^  
  --db-cluster-snapshot-identifier myclustersnapshot ^  
  --engine mysql ^  
  --multi-az ^  
  --db-instance-class db.r6g.xlarge
```

在還原資料庫執行個體之後，您可以將其集新增至安全群組，此安全群組與您用來建立快照 (如果適用) 的多可用區域資料庫叢集相關聯。完成此動作會提供先前多可用區域資料庫叢集的相同功能。

RDS API

若要將異地同步備份資料庫叢集快照還原至資料庫執行個體部署，請使用下列參數呼叫 RDS API 作業 [恢復資料InstanceFrom庫快照](#)：

- DBInstanceIdentifier
- DBClusterSnapshotIdentifier
- Engine

您也可以指定選用參數。

在還原資料庫執行個體之後，您可以將其集新增至安全群組，此安全群組與您用來建立快照 (如果適用) 的多可用區域資料庫叢集相關聯。完成此動作會提供先前多可用區域資料庫叢集的相同功能。

教學課程：從資料庫快照還原 Amazon RDS 資料庫執行個體

通常，使用 Amazon RDS 時，您可能有一個偶爾使用的資料庫執行個體，但該執行個體不需全時段可用。例如，假設每一季您都會進行客戶問卷調查，其會使用 Amazon EC2 執行個體來託管客戶問卷調查網站。您也有用來存放問卷調查結果的資料庫執行個體。在此情況下，節省支出的方式就是在問卷調查完成後建立資料庫執行個體的資料庫快照。然後，您可以刪除資料庫執行個體，並在需要時將其還原以再次進行問卷調查。

還原資料庫執行個體時，您會提供要從中還原之資料庫快照的名稱。然後，您需提供從還原作業中建立的新資料庫執行個體名稱。

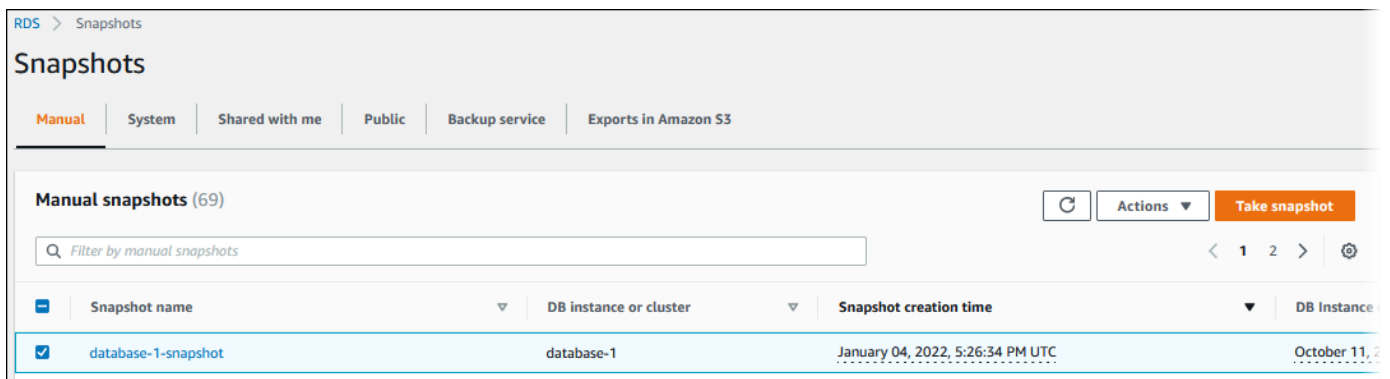
如需有關從快照還原資料庫執行個體的詳細資訊，請參閱 [從資料庫快照還原](#)。

從資料庫快照還原資料庫執行個體

在 AWS Management Console 中使用下列程序從快照還原。

從資料庫快照還原資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。



Restore snapshot (還原快照) 頁面隨即出現。

RDS > Snapshots > Restore snapshot

Restore snapshot

You are creating a new DB instance or DB cluster from a snapshot. The default VPC security group and parameter group are selected for the new DB instance or DB cluster, but you can change these settings.

DB instance settings

DB engine

SQL Server Express Edition ▼

License model

license-included ▼

Settings

DB snapshot ID
The identifier for the DB snapshot.
database-1-snapshot

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

5. 在 DB instance settings (資料庫執行個體設定) 底下，為 DB engine (資料庫引擎) 和 License model (授權模式) 使用預設的設定 (適用於 Oracle 或 Microsoft SQL Server)。
6. 在 Settings (設定) 下的 DB instance identifier (資料庫執行個體識別符) 中，輸入還原的資料庫執行個體要使用的唯一名稱，如 **mynewdbinstance**。

如果要從建立資料庫快照後即刪除的資料庫執行個體還原，可以使用該資料庫執行個體的名稱。

7. 在 可用性和持久性底下，選擇是否在另一個可用區域中建立備用執行個體。

對於本教學課程，請勿建立待命執行個體。

8. 在 Connectivity (連線) 底下，使用以下項目的預設設定：
 - Virtual Private Cloud (VPC) (虛擬私有雲端 (VPC))
 - DB subnet group (資料庫子網路群組)
 - 公用存取
 - VPC security group (firewall) (VPC 安全群組 (防火牆))
9. 選擇 DB instance class (資料庫執行個體類別)。

在本教學課程中，請選擇 Burstable classes (includes t classes) (高載類別 (包括 t 類別))，然後選擇 db.t3.small。

10. Encryption (加密) 請使用預設設定。

如果快照的來源資料庫執行個體已加密，還原的資料庫執行個體也會加密。無法使還原項目處於未加密狀態。

11. 展開頁面底部的 Additional configuration (其他組態)。

▼ Additional configuration
Database options, backup enabled, backtrack disabled, CloudWatch Logs, maintenance, delete protection disabled

Database options

DB parameter group [Info](#)
default.sqlserver-ex-15.0

Option group [Info](#)
default:sqlserver-ex-15-00

Collation [Info](#)

Backup

Copy tags to snapshots

Log exports
Select the log types to publish to Amazon CloudWatch Logs

Error log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Maintenance
Auto minor version upgrade [Info](#)

Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released.
The automatic upgrades occur during the maintenance window for the database.

Deletion protection

Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

12. 在 Database options (資料庫選項) 底下執行以下操作：

a. 選擇 DB parameter group (資料庫參數群組)。

在本教學課程中，請使用預設的參數群組。

b. 選擇 Option group (選項群組)。

在本教學課程中，請使用預設的選項群組。

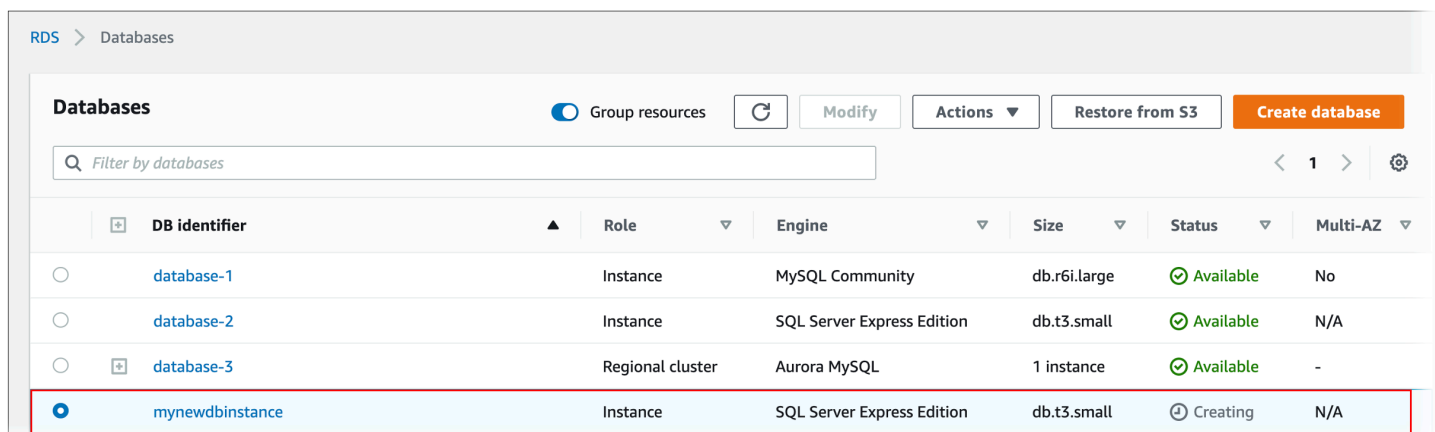
⚠ Important

在某些情況下，您可能會從使用持久性或永久性選項之資料庫執行個體的資料庫快照進行還原。若是如此，請務必選擇一個使用相同選項的選項群組。

- c. 在 Deletion protection (刪除保護) 選擇 Enable deletion protection (啟用刪除保護) 核取方塊。

13. 選擇 Restore DB instance (還原資料庫執行個體)。

Databases (資料庫) 頁面會顯示還原的資料庫執行個體，其狀態為 Creating。



RDS > Databases

Databases Group resources

Filter by databases < 1 >

<input type="checkbox"/>	DB identifier	Role	Engine	Size	Status	Multi-AZ
<input type="radio"/>	database-1	Instance	MySQL Community	db.r6i.large	Available	No
<input type="radio"/>	database-2	Instance	SQL Server Express Edition	db.t3.small	Available	N/A
<input type="radio"/>	database-3	Regional cluster	Aurora MySQL	1 instance	Available	-
<input checked="" type="radio"/>	mynewdbinstance	Instance	SQL Server Express Edition	db.t3.small	Creating	N/A

複製資料庫快照

透過 Amazon RDS，即可複製自動備份或手動資料庫快照。複製快照之後，副本即為手動快照。您可以製作自動備份或手動快照的多個複本，但每個複本都必須具有唯一的識別符。

您可以在其中複製快照 AWS 區域，也可以在其中複製快照 AWS 區域，也可以複製共用快照。

限制

以下是複製快照時的一些限制：

- 您無法在中國 (北京) 或中國 (寧夏) 區域來回複製快照。
- 您可以在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 之間複製快照。不過，您無法在這些 GovCloud (美國) 區域和非 (美國) 區域之間複製快照。 GovCloud
- 如果您在目標快照變成可用之前刪除來源快照，則快照副本可能會失敗。刪除來源快照之前，請確認目標快照的狀態為 AVAILABLE。
- 對於單一目的地區域，每一帳戶最多可有 20 個進行中的快照複製請求。
- 當您為相同來源資料庫執行個體請求多個快照副本時，其將會在內部排隊。稍後請求的副本在先前的快照副本完成前將不會啟動。如需詳細資訊，請參閱[為什麼我的 EC2 AMI 或 EBS 快照建立緩慢？](#) 在 AWS 知識中心。
- 視 AWS 區域 涉及的資料量和要複製的資料量而定，跨區域快照副本可能需要數小時才能完成。有時某個來源區域會出現大量跨區域快照複製請求。此時，在一些進行中的複製完成之前，Amazon RDS 可能會將該來源區域新提出的跨區域複製請求排入佇列。位於佇列中的複製請求不會顯示進度資訊。複製開始時才會顯示進度資訊。
- 當您啟動另一個複本時，如果某個複本仍處於待定狀態，則第二個複本在第一個複本完成之前不會開始。
- 您無法複製異地同步備份資料庫叢集的快照。

快照保留

Amazon RDS 會在多種情況下刪除自動備份：

- 在其保留期結束時。
- 在您停用資料庫執行個體的自動備份時。
- 在您刪除資料庫執行個體時。

如果想要讓自動備份保留期間更長，請複製自動備份來建立手動快照，即可保留到您刪除為止。如果手動快照超出預設儲存空間，則其可能產生 Amazon RDS 儲存成本。

如需備份儲存成本的詳細資訊，請參閱 [Amazon RDS 定價](#)。

複製共用快照

您可以複製其他人共用給您的快照 AWS 帳戶。在某些情況下，您可能會複製已從另一個人共用的加密快照 AWS 帳戶。在這些情況下，您必須擁有用 AWS KMS key 來加密快照的存取權。

Note

Amazon RDS 儲存成本適用於您複製的共用快照。Amazon RDS 可能會將來源資料庫執行個體的 ARN 附加到您複製的快照。

AWS 區域 如果快照未加密，您可以複製共用資料庫快照。不過，如果共用資料庫快照已加密，則您只能在相同的區域中複製快照。

Note

如果共用增量快照未加密，或使用與初始完整快照相同的 KMS 金鑰加密，AWS 區域 則支援複製相同的快照。如果您在複製後續快照時使用不同的 KMS 金鑰來加密，那些共用的快照就是完整的快照。如需詳細資訊，請參閱 [增量式快照複製](#)。

處理加密

您可以複製使用 KMS 金鑰所加密的快照。如果您複製加密快照，則快照的副本也必須加密。如果您複製相同的快照集中的加密快照 AWS 區域，您可以使用與原始快照相同的 KMS 金鑰加密副本。或者，您可以指定不同的 KMS 金鑰。

若要跨區域複製加密的快照，就必須指定在目的地 AWS 區域中有效的 KMS 金鑰。可以是特定區域專用的 KMS 金鑰，也可以是多區域金鑰。如需多區域 KMS 金鑰的詳細資訊，請參閱 [使用 AWS KMS 中的多區域金鑰](#)。

在整個複製過程中來源快照仍會保持加密狀態。如需更多詳細資訊，請參閱 [Amazon RDS 加密資料庫執行個體的限制](#)。

您也可以將未加密快照的副本加密。這樣可以快速將加密新增至先前未加密的資料庫執行個體。若要完成此操作，當您準備好加密資料庫執行個體時，可以建立其快照。然後建立該快照的複本，並指定 KMS 金鑰以加密該快照複本。之後，您可以從加密快照還原加密的資料庫執行個體。

增量式快照複製

增量式快照僅含相同資料庫執行個體最近一次拍攝快照之後所變更的資料。增量式快照複製速度較快，而且儲存成本較完整的快照複製低。

快照複本是否為增量，取決於最近完成的快照複本和來源快照。如果刪除了最近的快照複本，下一個複本則會是完整複本，而不是增量式複本。快照複本的類型與來源快照的類型相同。如果來源快照是增量快照，則快照複本將是增量快照。

當您複製快照時 AWS 帳戶，只有在符合下列所有條件的情況下，複製才會是增量副本：

- 最新的快照副本屬於相同的來源資料庫執行個體，仍然存在於目的地帳戶中。
- 目的地帳戶中快照的所有複本均未加密，或是已使用相同的 KMS 金鑰進行加密。
- 如果來源資料庫執行個體是多可用區域執行個體，則自上次擷取快照以來，尚未容錯移轉至另一個可用區域。

下列範例說明完整快照和增量快照之間的不同。這些範例同時適用於共用快照和非共用快照。

快照	加密金鑰	完整或增量快照
S1	K1	完整
S2	K1	S1 增量
S3	K1	S2 增量
S4	K1	S3 增量
S1 (S1C) 副本	K2	完整
S2 (S2C) 副本	K3	完整
S3 (S3C) 副本	K3	S2C 增量
S4 (S4C) 副本	K3	S3C 增量

快照	加密金鑰	完整或增量快照
S4 (S4C2) 副本 2	K4	完整

Note

在這些範例中，僅先前的快照仍存在時，快照 S2、S3 和 S4 才是增量快照。這也適用於副本。僅先前的副本仍存在時，快照副本 S3C 和 S4C 才是增量快照。

如需複製增量快照的相關資訊 AWS 區域，請參閱[完整複本和增量複本](#)。

跨區域快照複製

您可跨 AWS 區域複製資料庫快照。但是，跨區域快照複製具有特定的限制和考量。

請求跨區域資料庫快照複本

若要與來源區域通訊，以請求跨區域資料庫快照複本，請求者 (IAM 角色或 IAM 使用者) 必須具有來源資料庫快照和來源區域的存取權。

請求者的 IAM 政策中的某些條件可能會導致請求失敗。以下範例假設您正在將資料庫快照從 美國東部 (俄亥俄) 複製到 US East (N. Virginia)。以下範例顯示請求者 IAM 政策中導致請求失敗的條件：

- 請求者的政策具有 `aws:RequestedRegion` 的條件。

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:RequestedRegion": "us-east-1"
  }
}
```

由於政策不允許存取來源區域，請求失敗。若要成功請求，請同時指定來源區域和目的地區域。

```
...
"Effect": "Allow",
```

```

"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:RequestedRegion": [
      "us-east-1",
      "us-east-2"
    ]
  }
}

```

- 請求者的政策不允許存取來源資料庫快照。

```

...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...

```

若要成功要求，請同時指定來源快照和目的地快照。

```

...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
  "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
  "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...

```

- 請求者的政策拒絕 `aws:ViaAWSService`。

```

...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
  "Bool": {"aws:ViaAWSService": "false"}
}

```

與來源區域的通訊由 RDS 代表請求者進行。對於成功的請求，請勿拒絕 AWS 服務撥打的電話。

- 請求者的政策具有 `aws:SourceVpc` 或 `aws:SourceVpce` 的條件。

這些請求可能會失敗，因為當 RDS 呼叫遠端區域時，其不是來自指定的 VPC 或 VPC 端點。

如果您需要使用會導致請求失敗的先前條件之一，則可以在政策中包含 `aws:CalledVia` 的第二個陳述式，以使請求成功。例如，您可以如下所示使用 `aws:CalledVia` 搭配 `aws:SourceVpce`：

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
  "Condition": {
    "ForAnyValue:StringEquals" : {
      "aws:SourceVpce": "vpce-1a2b3c4d"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "rds:CopyDBSnapshot"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
        "rds.amazonaws.com"
      ]
    }
  }
}
```

如需詳細資訊，請參閱 IAM 使用者指南 中的 [IAM 中的政策和許可](#)。

授權快照複製

跨區域資料庫快照複製請求之後傳回 `success`，RDS 會在背景中啟動複本。系統會建立 RDS 存取來源快照的授權。此授權將來源資料庫快照連結至目標資料庫快照，並允許 RDS 僅複製到指定的目標快照。

授權由 RDS 使用服務連結 IAM 角色中的 `rds:CrossRegionCommunication` 許可來驗證。如果授權複本，RDS 會與來源區域通訊並完成複本。

對於之前未透過 CopyDBSnapshot 請求授權的資料庫快照，RDS 將無法存取。複製完成時，授權即會撤銷。

RDS 使用服務連結的角色來驗證來源區域中的授權。如果您在複製程序期間刪除服務連結的角色，複本則會失敗。

如需詳細資訊，請參閱 IAM 使用者指南中的[使用服務連結角色](#)。

使用 AWS Security Token Service 認證

來自 global AWS Security Token Service (AWS STS) 端點的會話令牌僅在 AWS 區域 默認情況下啟用 (商業區域) 的會話令牌有效。如果您使用中 assumeRole API 作業的認證 AWS STS，如果來源區域是選擇加入的區域，請使用地區端點。否則，請求將失敗。發生這種情況是因為您的憑據在兩個區域都必須有效，這對於選擇加入區域僅在使用區域 AWS STS 端點時才有效。

若要使用全域端點，請確保操作時在這兩個區域均已將其啟用。在帳號設定 Valid in all AWS # #中將全域端點 AWS STS 設定為。

相同的規則適用於預先簽章 URL 參數中的登入資料。

如需詳細資訊，請參閱《IAM 使用者指南》AWS 區域中的〈[管理 AWS STS](#)〉。

延遲和多個複製要求

視 AWS 區域 涉及的資料量和要複製的資料量而定，跨區域快照副本可能需要數小時才能完成。

有時某個來源 AWS 區域會出現大量跨區域快照複製請求。在這種情況下，Amazon RDS 可能會將來自該來源的新跨區域副本請求放入佇列，直到某些進行中的複本完成為止。位於佇列中的複製請求不會顯示進度資訊。複製開始時才會顯示進度資訊。

完整複本和增量複本

當您將快照複製到與來源快照不同 AWS 區域 的快照時，即使複製增量快照，第一個複本也是完整快照複本。完整的快照副本內含所有還原資料庫執行個體所需的資料及中繼資料。在第一個快照複製之後，您可以將相同資料庫執行個體的增量快照複製到相同的目標區域中 AWS 帳戶。如需增量快照的詳細資訊，請參閱[增量式快照複製](#)。

未加密和加密快照均支援增量快照複製。AWS 區域

當您複製快照時 AWS 區域，如果符合下列條件，則副本為增量副本：

- 之前已將快照複製到目的地區域。
- 最近的快照副本仍存在於目的地區域中。
- 目的地區域中快照的所有複本均未加密，或是已使用相同的 KMS 金鑰進行加密。

選項群組考量

資料庫選項群組特定 AWS 區域 於其建立的群組，而且您無法使用其中一個選項群組 AWS 區域 的選項群組 AWS 區域。

對於 Oracle 資料庫，您可以使用 AWS CLI 或 RDS API 從與您的 AWS 帳戶。您只能複製相同 AWS 區域內的選項群組。如果選項群組已複製到目的地帳戶，且自複製之後未對其進行任何變更，則不會複製該選項群組。如果之前已複製來源選項群組，但此群組自複製後已變更，則 RDS 會將新版本複製到目的地帳戶。不會複製預設選項群組。

當您跨區域複製快照時，您可以為快照指定新的選項群組。建議在複製快照之前準備新的選項群組。在目的地中 AWS 區域，建立與原始資料庫執行個體相同設定的選項群組。如果一個已經存在於新的 AWS 區域，你可以使用那個。

有時您可能會複製快照，而不是為快照指定新的選項群組。此時，在您還原快照時，資料庫執行個體會取得預設選項群組。如要提供原始的相同選項給新的資料庫執行個體，需要執行下列動作：

1. 在目的地中 AWS 區域，建立與原始資料庫執行個體相同設定的選項群組。如果一個已經存在於新的 AWS 區域，你可以使用那個。
2. 在目標中還原快照之後 AWS 區域，請修改新的資料庫執行個體，並從上一步新增新的或現有的選項群組。

參數群組考量

當您跨區域複製快照時，副本不包含原始資料庫執行個體所使用的參數群組。當您還原快照以建立新的資料庫執行個體時，該資料庫執行個體會取得在 AWS 區域 其中建立該執行個體的預設參數群組。如要提供原始的相同參數給新的資料庫執行個體，需要執行下列動作：

1. 在目的地中 AWS 區域，使用與原始資料庫執行個體相同的設定建立資料庫參數群組。如果一個已經存在於新的 AWS 區域，你可以使用那個。
2. 在目標中還原快照之後 AWS 區域，請修改新的資料庫執行個體，並從上一步新增新的或現有的參數群組。

複製資料庫快照

使用本主題中的程序來複製資料庫快照。如需複製快照的概觀，請參閱[複製資料庫快照](#)

對於每個快照 AWS 帳戶，您一次最多可以從一個快照複製 AWS 區域 到另一個資料庫快照。如果您將資料庫快照複製到另一個資料庫快照 AWS 區域，則會建立保留在其中的手動資料庫快照 AWS 區域。從來源複製資料庫快照 AWS 區域 會產生 Amazon RDS 資料傳輸費用。

如需資料傳輸定價的詳細資訊，請參閱 [Amazon RDS 定價](#)。

在新建立資料庫快照副本之後 AWS 區域，資料庫快照副本的行為與其中的所有其他資料庫快照相同。
AWS 區域

您可以使用 AWS Management Console、或 RDS API 複製資料庫快照集。 AWS CLI

主控台

下列程序會使用複製相同或跨區域中的加密 AWS 區域 或未加密資料庫快照。 AWS Management Console

複製資料庫快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選取您要複製的資料庫快照。
4. 針對 Actions (動作) 選擇 Copy snapshot (複製快照)。

Copy snapshot (複製快照) 頁面隨即出現。

RDS > Snapshots > Copy snapshot

Copy snapshot

Settings

Source DB Snapshot
DB Snapshot Identifier for the snapshot being copied.
db1-snapshot

Destination Region [Info](#)
US West (Oregon) ▼

New DB Snapshot Identifier
DB Snapshot Identifier for the new snapshot

Target Option Group (Optional)
No preference ▼

Copy Tags [Info](#)

i Please note that depending on the amount of data to be copied and the Region you choose, this operation could take several hours to complete and the display on the progress bar could be delayed until setup is complete.

Encryption

Encryption [Info](#)
 Enable Encryption
Choose to encrypt the copy of the source DB snapshot. Master key IDs and aliases appear in the list after they have been created using KMS. You cannot remove encryption from an encrypted DB snapshot.

Master key [Info](#)
(default) aws/rds ▼

Account

KMS key ID

[Cancel](#) [Copy snapshot](#)

5. 針對 Target option group (optional) (目標選項群組(選用))，選擇新的選項群組 (如果需要的話)。

如果要將快照從一個快照複製 AWS 區域 到另一個快照，而您的資料庫執行個體使用非預設選項群組，請指定此選項。

如果來源資料庫執行個體使用適用於 Oracle 或 Microsoft SQL Server 的透明資料加密，則您在跨區域複製時必須指定此選項。如需詳細資訊，請參閱 [選項群組考量](#)。

6. (選擇性) 若要將資料庫快照複製到其他快照 AWS 區域，請針對「目的地區域」選擇新的快照 AWS 區域。

Note

目的地 AWS 區域 必須具有與來源可用的相同資料庫引擎版本 AWS 區域。

7. 針對 New DB snapshot identifier (新的資料庫快照識別符)，輸入資料庫快照複本的名稱。

您可以製作自動備份或手動快照的多個複本，但每個複本都必須具有唯一的識別符。

8. (選用) 選取 Copy Tags (複製標籤)，將快照中的標籤和值複製到快照的副本。
9. (選用) 針對 Encryption (加密)，執行下列動作：

- a. 如果資料庫快照未加密，但您想要將複本加密，請選擇 啟用加密。

Note

如果資料庫快照已加密，您必須加密複本，才能勾選取核取方塊。

- b. 在 AWS KMS key 中，指定用來加密資料庫快照複本的 KMS 金鑰識別碼。

10. 選擇 Copy Snapshot (複製快照)。

AWS CLI

您可以使用 AWS CLI 指令複製資料庫快照集 [copy-db-snapshot](#)。如果您要將快照複製到新的快照 AWS 區域，請在新的中執行命令 AWS 區域。

下列選項用來複製資料庫快照。並非所有情況需要使用所有選項。請根據下列描述和範例來決定要使用的選項。

- `--source-db-snapshot-identifier` – 來源資料庫快照的識別符。
 - 如果來源快照與副本相 AWS 區域 同，請指定有效的資料庫快照識別碼。例如 `rds:mysql-instance1-snapshot-20130805`。

- 如果來源快照與副本相 AWS 區域 同，且已與您的快照共用 AWS 帳戶，請指定有效的資料庫快照 ARN。例如 `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`。
- 如果來源快照與副本 AWS 區域 不同，請指定有效的資料庫快照 ARN。例如：`arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`。
- 如果您要從共用的手動資料庫快照複製，此參數必須是共用資料庫快照的 Amazon 資源名稱 (ARN)。
- 如果您要複製加密快照，此參數必須為來源的 ARN 格式 AWS 區域，且必須符合 `PreSignedUrl` 參數 `SourceDBSnapshotIdentifier` 中的。
- `--target-db-snapshot-identifier` – 加密資料庫快照之新副本的識別符。
- `--copy-option-group` – 從已與您的 AWS 帳戶共用的快照中複製選項群組。
- `--copy-tags` – 包含複製標籤選項，可將快照中的標籤和值複製到快照的副本。
- `--option-group-name` – 要與快照的副本相關聯的選項群組。

如果要將快照從一個快照複製 AWS 區域 到另一個快照，而您的資料庫執行個體使用非預設選項群組，請指定此選項。

如果來源資料庫執行個體使用適用於 Oracle 或 Microsoft SQL Server 的透明資料加密，則您在跨區域複製時必須指定此選項。如需詳細資訊，請參閱 [選項群組考量](#)。

- `--kms-key-id` – 加密資料庫快照的 KMS 金鑰識別碼。KMS 金鑰識別碼是 KMS 金鑰的 Amazon 資源名稱 (ARN)、金鑰識別碼或金鑰別名。
 - 如果您從中複製加密的資料庫快照 AWS 帳戶，您可以指定此參數的值，以使用新的 KMS 金鑰加密副本。如果您不指定此參數的值，則會使用與來源資料庫快照相同的 KMS 金鑰，以加密資料庫快照的副本。
 - 如果您複製從另一個共用的加密資料庫快照集 AWS 帳戶，則必須指定此參數的值。
 - 如果您在複製未加密快照時指定此參數，則副本會進行加密。
 - 如果您將加密快照複製到其他快照 AWS 區域，則必須為目的地指定 KMS 金鑰 AWS 區域。KMS 金鑰專屬於建立金鑰 AWS 區域 的金鑰，您無法使用另一個 AWS 區域 金鑰的加密金鑰 AWS 區域。

Example 從未加密的快照，到相同區域

下列程式碼會建立具有新名稱的快照副本 `mydbsnapshotcopy`，與來源快照 AWS 區域 相同。建立複本時，原始快照上的資料庫選項群組和標籤都會複製到快照複本。

對於LinuxmacOS、或Unix：

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 \  
  --target-db-snapshot-identifier mydbsnapshotcopy \  
  --copy-option-group \  
  --copy-tags
```

在 Windows 中：

```
aws rds copy-db-snapshot ^  
  --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 ^  
  --target-db-snapshot-identifier mydbsnapshotcopy ^  
  --copy-option-group ^  
  --copy-tags
```

Example 從未加密的快照，跨區域

下列程式碼會使用新名稱建立快照的複本mydbsnapshotcopy，並 AWS 區域 在其中執行命令。

對於LinuxmacOS、或Unix：

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 \  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

在 Windows 中：

```
aws rds copy-db-snapshot ^  
  --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 ^  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

Example 從已加密的快照，跨區域

下列程式碼範例會在 US East (N. Virginia) 區域中，從 美國西部 (奧勒岡) 區域複製已加密的資料庫快照。在目的地 (us-east-1) 區域中執行命令。

對於Linux、macOS、或Unix：

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-  
instance1-snapshot-20161115 \  
  --target-db-snapshot-identifier mydbsnapshotcopy \  
  --kms-key-id my-us-east-1-key \  
  --option-group-name custom-option-group-name
```

在 Windows 中：

```
aws rds copy-db-snapshot ^  
  --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-  
instance1-snapshot-20161115 ^  
  --target-db-snapshot-identifier mydbsnapshotcopy ^  
  --kms-key-id my-us-east-1-key ^  
  --option-group-name custom-option-group-name
```

在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 區域之間複製加密快照時，需要此 `--source-region` 參數。若為 `--source-region`，請指定來源資料庫執行個體的 AWS 區域。

若未指定 `--source-region`，請指定 `--pre-signed-url` 值。presigned URL (預先簽章的 URL) 為包含對來源 AWS 區域中呼叫 `copy-db-snapshot` 命令之 Signature 第 4 版簽章請求的 URL。若要進一步瞭解該 `pre-signed-url` 選項，請參閱《AWS CLI 指令參考》[copy-db-snapshot](#) 中的〈〉。

RDS API

您可以使用 Amazon RDS API 操作 [CopyDBSnapshot](#) 來複製資料庫快照。如果您要將快照複製到新的快照 AWS 區域，請在新的快照中執行動作 AWS 區域。

下列參數用來複製資料庫快照。並非所有情況需要使用所有參數。請根據下列描述和範例來決定要使用的參數。

- `SourceDBSnapshotIdentifier` – 來源資料庫快照的識別符。
 - 如果來源快照與副本相 AWS 區域 同，請指定有效的資料庫快照識別碼。例如 `rds:mysql-instance1-snapshot-20130805`。
 - 如果來源快照與副本相 AWS 區域 同，且已與您的快照共用 AWS 帳戶，請指定有效的資料庫快照 ARN。例如 `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`。

- 如果來源快照與副本 AWS 區域不同，請指定有效的資料庫快照 ARN。例如：`arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`。
- 如果您要從共用的手動資料庫快照複製，此參數必須是共用資料庫快照的 Amazon 資源名稱 (ARN)。
- 如果您要複製加密快照，此參數必須為來源的 ARN 格式 AWS 區域，且必須符合 `PreSignedUrl` 參數 `SourceDBSnapshotIdentifier` 中的。
- `TargetDBSnapshotIdentifier` – 加密資料庫快照之新副本的識別符。
- `CopyOptionGroup` – 將此參數設為 `true`，將共用快照中的選項群組複製到快照的副本。預設值為 `false`。
- `CopyTags` – 將此參數設為 `true`，將快照中的標籤和值複製到快照的副本。預設值為 `false`。
- `OptionGroupName` – 要與快照的副本相關聯的選項群組。

如果要將快照從一個快照複製 AWS 區域 到另一個快照，而您的資料庫執行個體使用非預設選項群組，請指定此參數。

如果來源資料庫執行個體使用適用於 Oracle 或 Microsoft SQL Server 的透明資料加密，則您在跨區域複製時必須指定此參數。如需詳細資訊，請參閱 [選項群組考量](#)。

- `KmsKeyId` – 加密資料庫快照的 KMS 金鑰識別碼。KMS 金鑰識別碼是 KMS 金鑰的 Amazon 資源名稱 (ARN)、金鑰識別碼或金鑰別名。
 - 如果您從中複製加密的資料庫快照 AWS 帳戶，您可以指定此參數的值，以使用新的 KMS 金鑰加密副本。如果您不指定此參數的值，則會使用與來源資料庫快照相同的 KMS 金鑰，以加密資料庫快照的副本。
 - 如果您複製從另一個共用的加密資料庫快照集 AWS 帳戶，則必須指定此參數的值。
 - 如果您在複製未加密快照時指定此參數，則副本會進行加密。
 - 如果您將加密快照複製到其他快照 AWS 區域，則必須為目的地指定 KMS 金鑰 AWS 區域。KMS 金鑰專屬於建立金鑰的金鑰，您無法使用另一個 AWS 區域 金鑰的加密金鑰 AWS 區域。AWS 區域
- `PreSignedUrl` – 包含針對 `CopyDBSnapshot` API 作業的簽章版本 4 簽署要求的 URL，其中包含 AWS 區域 要複製的來源資料庫快照集。

當您使用 Amazon RDS API 從另一個 AWS 區域 快照複製加密的資料庫快照時，請指定此參數。當您使用 AWS CLI 從另一個 AWS 區域 複製加密資料庫快照時，您可以指定來源區域選項，而非此參數。

預先簽章的 URL 必須是對於 CopyDBSnapshot API 操作有效的請求，而此作業可於包含要複製之加密資料庫快照的來源 AWS 區域中執行。預先簽章的 URL 請求必須包含下列參數值：

- **DestinationRegion**— 將加密 AWS 區域的資料庫快照複製到的目標。AWS 區域 這與呼叫包含此預先簽署 URL 的 CopyDBSnapshot 作業相同。

例如，假設您將加密的資料庫快照從 us-west-2 區域複製到 us-east-1 區域。然後，您可以在 us-east-1 區域中呼叫 CopyDBSnapshot 操作，並提供一個預先簽章的 URL，其中包含在 us-west-2 區域中對 CopyDBSnapshot 操作的呼叫。在此範例中，預先簽章的 URL 中的 DestinationRegion 必須設為 us-east-1 區域。

- **KmsKeyId** – 金鑰的 KMS 金鑰識別碼，用於在目的地 AWS 區域中加密資料庫快照的複本。對於在目的地中呼叫的 CopyDBSnapshot 作業，以及預先簽署 URL 中包含的作業 AWS 區域，這是相同的識別碼。
- **SourceDBSnapshotIdentifier** – 要複製之加密快照的資料庫快照識別符。此識別符必須是來源 AWS 區域的 Amazon Resource Name (ARN) 格式。例如，如果您要從 us-west-2 區域複製加密的資料庫快照，則您的 SourceDBSnapshotIdentifier 看起來像下列範例：
.arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115

如需 Signature 第 4 版所簽署之請求的詳細資訊，請參閱下列資訊：

- [驗證請求：使用 Amazon 簡單儲存服務 API 參考中的查詢參數 \(AWS 簽章版本 4\)](#)
- [簽名版本 4 中的簽名過程](#) AWS 一般參考

Example 從未加密的快照，到相同區域

下列程式碼會建立具有新名稱的快照副本 mydbsnapshotcopy，與來源快照 AWS 區域相同。建立副本時，原始快照的所有標籤會複製到快照副本。

```
https://rds.us-west-1.amazonaws.com/  
?Action=CopyDBSnapshot  
&CopyTags=true  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805  
&TargetDBSnapshotIdentifier=mydbsnapshotcopy  
&Version=2013-09-09  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
```

```
&X-Amz-Date=20140429T175351Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example 從未加密的快照，跨區域

下列程式碼會在 美國西部 (加州北部) 區域中使用新名稱 `mydbsnapshotcopy` 來建立快照的複本。

```
https://rds.us-west-1.amazonaws.com/
?Action=CopyDBSnapshot
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&SourceDBSnapshotIdentifier=arn%3Aaws%3Aards%3Aus-east-1%3A123456789012%3Asnapshot%3Amysql-instance1-snapshot-20130805
&TargetDBSnapshotIdentifier=mydbsnapshotcopy
&Version=2013-09-09
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
&X-Amz-Date=20140429T175351Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example 從已加密的快照，跨區域

下列程式碼會在 US East (N. Virginia) 區域中使用新名稱 `mydbsnapshotcopy` 來建立快照的複本。

```
https://rds.us-east-1.amazonaws.com/
?Action=CopyDBSnapshot
&KmsKeyId=my-us-east-1-key
&OptionGroupName=custom-option-group-name
&PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F%253FAction%253DCopyDBSnapshot%2526DestinationRegion%253Dus-east-1%2526KmsKeyId%253Dmy-us-east-1-key%2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Aards%25253Aus-west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115%2526SignatureMethod%253DHmacSHA256%2526SignatureVersion%253D4%2526Version%253D2014-10-31%2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256%2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds%252Faws4_request%2526X-Amz-Date%253D20161117T215409Z
```

```
%2526X-Amz-Expires%253D3600  
%2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-  
content-sha256%253Bx-amz-date  
%2526X-Amz-Signature  
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&SourceDBSnapshotIdentifier=arn%3Aaws%3Aards%3Aus-west-2%3A123456789012%3Asnapshot  
%3Amysql-instance1-snapshot-20161115  
&TargetDBSnapshotIdentifier=mydbsnapshotcopy  
&Version=2014-10-31  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request  
&X-Amz-Date=20161117T221704Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

共享資料庫快照

使用 Amazon RDS 時，您可以透過下列方式共用手動資料庫快照：

- 共用手動資料庫快照 (無論是加密還是未加密) 都可 AWS 帳戶 以授權複製快照。
- 共用未加密的手動資料庫快照可讓授權直 AWS 帳戶 接從快照還原資料庫執行個體，而不是擷取該執行個體的副本並從中還原。但是，您無法從已共用又加密的資料庫快照來還原資料庫執行個體。不過，您可以複製資料庫快照，再從副本還原資料庫執行個體。

Note

若要共用自動資料庫快照，請複製該自動快照來建立手動資料庫快照，然後共用該複本。此程序也適用於 AWS Backup 產生的資源。

如需有關複製快照的詳細資訊，請參閱[複製資料庫快照](#)。如需有關從資料庫快照還原資料庫執行個體的詳細資訊，請參閱[從資料庫快照還原](#)。

您最多可與 20 個其他人共用手動快照 AWS 帳戶。

與其他人共用手動快照時，適用下列限制 AWS 帳戶：

- 使用 AWS Command Line Interface (AWS CLI) 或 Amazon RDS API 從共用快照還原資料庫執行個體時，必須將共用快照的 Amazon 資源名稱 (ARN) 指定為快照識別碼。
- 您無法共用使用具有永久或持久選項的選項群組的資料庫快照，但具有 Timezone 或 OLS 選項 (或兩者) 的 Oracle 資料庫執行個體除外。

無法從選項群組中移除永久選項。將具有持續選項的選項群組指派給資料庫執行個體後，就無法從資料庫執行個體中移除此選項群組。

下表列出永久和持續選項及其相關的資料庫引擎。

選項名稱	持續	永久	資料庫引擎
TDE	是	否	Microsoft SQL Server Enterprise Edition
TDE	是	是	Oracle Enterprise Edition

選項名稱	持續	永久	資料庫引擎
時區	是	是	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One 甲骨文標準版 2

針對 Oracle 資料庫執行個體，您可以複製具有 Timezone 或 OLS 選項 (或兩者) 的共用資料庫快照。作法是在您複製資料庫快照時，指定包含這些選項的目標選項群組。只有對執行 Oracle 12.2 版或更高版本的 Oracle 資料庫執行個體，OLS 才是永久和持續的選項。如需這些選項的詳細資訊，請參閱 [Oracle 時區](#) 和 [Oracle Label Security](#)。

- 您無法共用異地同步備份資料庫叢集的快照。

內容

- [共用快照](#)
- [共用公有快照](#)
 - [檢視其他人擁有的公開快照 AWS 帳戶](#)
 - [檢視您自己的公有快照](#)
 - [從已取代的資料庫引擎版本共用公開快照](#)
- [共用加密快照](#)
 - [建立客戶管理的金鑰並授予存取權](#)
 - [從來源帳戶複製並共用快照](#)
 - [複製目標帳戶中的共用快照](#)
- [停止快照共用](#)

共用快照

您可以使用 AWS Management Console、或 RDS API 共用 AWS CLI 資料庫快照集。

主控台

使用 Amazon RDS 主控台，您最多可以共用 20 個手動資料庫快照 AWS 帳戶。您也可以使用主控台來停止將手動快照共用給一或多個帳戶。

使用 Amazon RDS 主控台來分享手動資料庫 快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選取您要共享的手動快照。
4. 針對 Actions (動作)，選擇 Share snapshot (共用快照)。
5. 在 DB snapshot visibility (資料庫快照可見度) 中，選擇下列其中一個選項。
 - 如果來源未加密，請選擇公用以允許所有 AWS 帳戶從手動資料庫快照還原資料庫執行個體，或選擇「專用」，僅 AWS 帳戶 允許您指定從手動資料庫快照還原資料庫執行個體。

Warning

如果將資料庫快照可見性設定為 Public，則所有人都 AWS 帳戶 可以從手動資料庫快照還原資料庫執行個體，並可存取您的資料。請勿將任何包含私人資訊的手動資料庫快照以 Public (公有) 形式共用。

如需詳細資訊，請參閱 [共用公有快照](#)。

- 如果來源資料庫叢集已加密，DB snapshot visibility (資料庫快照可見度) 會設為 Private (私有)，因為加密快照無法以公有形式共用。

Note

使用預設值加密的快照 AWS KMS key 無法共用。如需如何解決此問題的相關資訊，請參閱 [共用加密快照](#)。

6. 在 AWS 帳戶 ID 中，輸入您要允許從手動快照還原資料庫執行個體的帳戶 AWS 帳戶 識別碼，然後選擇 [新增]。重複以上步驟以包含其他 AWS 帳戶 識別碼，最多 20 個 AWS 帳戶。

如果您在將 AWS 帳戶 識別碼新增至允許的帳戶清單時發生錯誤，可以選擇錯誤 AWS 帳戶 識別碼右側的 [刪除]，將其從清單中刪除。

7. 新增所有要允許還原手動快照的識別碼之後，請選擇 [儲存] 以儲存變更。AWS 帳戶

AWS CLI

若要共享資料庫快照，請使用 `aws rds modify-db-snapshot-attribute` 命令。使用 `--values-to-add` 參數可新增授權可還原手動快照的 ID 清單。AWS 帳戶

Example 與單一帳戶共用快照

下列範例會啟用 AWS 帳戶 識別碼123456789012來還原名為的資料庫快照集db7-snapshot。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

在 Windows 中：

```
aws rds modify-db-snapshot-attribute ^
```

```
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example 與多個帳戶共用快照

下列範例會啟用兩個 AWS 帳戶 識別碼，以111122223333及444455556666還原名為的資料庫快照集manual-snapshot1。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add {"111122223333","444455556666"}
```

在 Windows 中：

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-add "[\"111122223333\", \"444455556666\"]"
```

Note

使用 Windows 命令提示字元時，您必須在 JSON 程式碼中的雙引號 (") 開頭加上反斜線 (\)，以逸出雙引號。

若要列出 AWS 帳戶 已啟用的還原快照，請使用[describe-db-snapshot-attributes](#) AWS CLI 指令。

RDS API

您也可以使用 Amazon RDS API 與其他人共 AWS 帳戶 用手動資料庫快照。若要這樣做，請呼叫 [ModifyDBSnapshotAttribute](#) 作業。指定 restore forAttributeName，然後使用 ValuesToAdd 參數新增授權 AWS 帳戶 可還原手動快照的 ID 清單。

若要讓所有人都可以公開手動快照並可還原 AWS 帳戶，請使用值 all。但是，請注意不要 all 為任何包含您不想提供給所有人使用的私人資訊的手動快照新增值 AWS 帳戶。另外，也不要對加密快照指定 all，因為不支援將這種快照變成公有。

若要列出所有 AWS 帳戶 允許還原快照的項目，請使用 [DescribeDBSnapshotAttributes](#) API 作業。

共用公有快照

您也可以將未加密的手動快照共用為公開，讓所有 AWS 帳戶人都可以使用快照。確保在公開共用快照時，公用快照中不包含您的任何私人資訊。

當快照集公開共用時，它會授予所有 AWS 帳戶 權限複製快照集以及從中建立資料庫執行個體。

您無需對其他帳戶所擁有的公有快照備份儲存支付費用。您只需為您擁有的快照付費。

若您複製公有快照，則您擁有該副本。您需要為快照副本的備份儲存支付費用。若您從公用快照建立一個資料庫執行個體，則需支付該資料庫執行個體的費用。如需 Amazon RDS 定價資訊，請參閱 [Amazon RDS 產品頁面](#)。

您僅能刪除您擁有的公用快照。若要刪除共用或公用快照，請務必登入擁有 AWS 帳戶 該快照的快照。

檢視其他人擁有的公開快照 AWS 帳戶

您可以在 Amazon RDS 主控台的 [快照] 頁面的 [公用] 索引標籤上，檢視特定 AWS 區域中其他帳戶擁有的公開快照。您的快照 (您帳戶擁有的快照) 不會顯示在此索引標籤上。

如要檢視公有快照

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇 Public (公有) 索引標籤。

公有快照隨即顯示。您可在 Owner (擁有者) 資料欄中查看哪個帳戶擁有公有快照。

Note

您可能需要選取 Public snapshots (公用快照) 清單右上角的齒輪圖示來修改頁面偏好設定，以查看此資料欄。

檢視您自己的公有快照

您可以使用以下 AWS CLI 命令 (僅限 Unix) 來查看您在特定 AWS 區域 AWS 帳戶 中擁有的公共快照。

```
aws rds describe-db-snapshots --snapshot-type public --include-public |  
grep account_number
```

若您有公有快照，則傳回的輸出類似下列範例。

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mynapshot1",  
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mynapshot2",
```

Note

您可能會看到 `DBSnapshotIdentifier` 或 `SourceDBSnapshotIdentifier` 的重複項目。

從已取代的資料庫引擎版本共用公開快

不支援從已取代的資料庫引擎版本還原或複製公用快照集。

適用於甲骨 PostgreSQL 和 RDS 的資料庫引擎支援直接升級資料庫快照引擎版本。您可以升級快照，然後公開重新共享它們。如需詳細資訊，請參閱下列內容：

- [升級 Oracle 資料庫快照](#)
- [升級 PostgreSQL 資料庫快照引擎版本](#)

對於其他資料庫引擎，請執行下列步驟，讓現有不支援的公用快照集可用於還原或複製：

1. 將快照標記為私人。
2. 還原快照。
3. 將還原的資料庫執行個體升級至支援的引擎版本。
4. 建立新快照。
5. 公開重新共用快照。

共用加密快照

您可共用已使用 AES-256 加密演算法來「靜態」加密的資料庫快照，如 [加密 Amazon RDS 資源](#) 所述。

共用加密快照有下列限制：

- 您無法將加密快照以公有形式共用。
- 您無法共用使用透明資料加密 (TDE) 所加密的 Oracle 或 Microsoft SQL Server 快照。
- 您無法共用已使用共用快照的預設 KMS 金鑰加密的快照。AWS 帳戶

若要解決預設 KMS 金鑰問題，請執行下列工作：

1. [建立客戶管理的金鑰並授予存取權](#)。
2. [從來源帳戶複製並共用快照](#)。
3. [複製目標帳戶中的共用快照](#)。

建立客戶管理的金鑰並授予存取權

首先，建立與加密資料庫快照相同 AWS 區域的自訂 KMS 金鑰。在建立客戶管理的金鑰時，您可以為其他金鑰授予存取權 AWS 帳戶。

建立客戶管理的金鑰並授予其存取權

1. AWS Management Console 從來源登入 AWS 帳戶。
2. [請在以下位置開啟 AWS KMS 主控台](https://console.aws.amazon.com/kms)。 <https://console.aws.amazon.com/kms>
3. 若要變更 AWS 區域，請使用頁面右上角的「地區」選取器。
4. 在導覽窗格中，選擇 Customer managed keys (客戶受管金鑰)。
5. 選擇建立金鑰。
6. 在「設定」金鑰頁面上：
 - a. 選取「對稱」做為「金鑰類型」。
 - b. 對於金鑰使用，請選取加密並解密。
 - c. 展開 Advanced options (進階選項)。
 - d. 對於金鑰材料來源，選取 KMS。

- e. 對於「地區性」，選取「單一區域金鑰」。
 - f. 選擇下一步。
7. 在「新增標籤」頁面上：
- a. 針對別名，例如，輸入 KMS 金鑰的顯示名稱 **share-snapshot**。
 - b. (選擇性) 輸入 KMS 金鑰的說明。
 - c. (選擇性) 將標籤新增至您的 KMS 金鑰。
 - d. 選擇下一步。
8. 在定義金鑰管理許可頁面上，選擇下一步。
9. 在 [定義金鑰使用權限] 頁面上：
- a. 對於「其他」AWS 帳戶，選擇「新增其他 AWS 帳戶」
 - b. 輸入您要授與存取權的 ID。AWS 帳戶
- 您可以授予多個訪問權限 AWS 帳戶。
- c. 選擇下一步。
10. 檢閱您的 KMS 金鑰，然後選擇 [完成]。

從來源帳戶複製並共用快照

接下來，您可以使用客戶受管金鑰將來源資料庫快照複製到新的快照集。然後，您與目標共享它 AWS 帳戶。

複製和共用快照

1. AWS Management Console 從來源登入 AWS 帳戶。
2. 在以下位置打開 Amazon RDS 控制台 <https://console.aws.amazon.com/rds/>
3. 在導覽窗格中，選擇快照。
4. 選取要複製的資料庫快照。
5. 針對 Actions (動作) 選擇 Copy snapshot (複製快照)。
6. 在 [複製快照] 頁面上：
 - a. 在「目的地區域」中，選擇您 AWS 區域 在上一個程序中建立客戶管理金鑰的位置。
 - b. 在新資料庫快照識別碼中輸入資料庫快照副本的名稱。
 - c. 在中 AWS KMS key，選擇您建立的客戶管理金鑰。

RDS > Snapshots > Copy snapshot

Copy snapshot

Settings

Source DB Snapshot
DB Snapshot Identifier for the snapshot being copied.
[test-snapshot](#)

Destination Region [Info](#)
EU (Frankfurt) ▼

New DB Snapshot Identifier
DB Snapshot Identifier for the new snapshot
test-snapshot-copy
Must start with a letter and only contain letters, digits, or hyphens.

Copy tags [Info](#)

i Please note that depending on the amount of data to be copied and the Region you choose, this operation could take several hours to complete and the display on the progress bar could be delayed until setup is complete.

Encryption

Encryption [Info](#)
 Enable Encryption
Choose to encrypt the copy of the source DB snapshot. Master key IDs and aliases appear in the list after they have been created using KMS. You cannot remove encryption from an encrypted DB snapshot.

AWS KMS key [Info](#)
share-snapshot ▼

Account
[Redacted]

KMS key ID
[Redacted]

Cancel **Copy snapshot**

- d. 選擇 Copy Snapshot (複製快照)。
7. 當快照複本可用時，請選取它。
8. 針對 Actions (動作)，選擇 Share snapshot (共用快照)。
9. 在 [快照] 權限頁面上：

- a. 輸入您要與其共享快照副本的 AWS 帳戶 ID，然後選擇「新增」。
- b. 選擇儲存。

快照已共用。

複製目標帳戶中的共用快照

現在您可以複製目標中的共用快照 AWS 帳戶。

複製共用快照

1. AWS Management Console 從目標登入 AWS 帳戶。
2. 在以下位置打開 Amazon RDS 控制台 <https://console.aws.amazon.com/rds/>
3. 在導覽窗格中，選擇快照。
4. 選擇 [與我共享] 索引標籤。
5. 選取共用快照。
6. 針對 Actions (動作) 選擇 Copy snapshot (複製快照)。
7. 依照先前程序選擇複製快照的設定，但使用屬 AWS KMS key 於目標帳戶的設定。

選擇 Copy Snapshot (複製快照)。

停止快照共用

若要停止共用資料庫快照集，請從目標移除權限 AWS 帳戶。

主控台

若要停止與 AWS 帳戶

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選取您要停止共享的手動快照。
4. 選擇 Actions (動作)，然後選擇 Share snapshot (共用快照)。
5. 若要移除的權限 AWS 帳戶，請從授權 AWS 帳戶清單中針對該帳戶的帳號識別碼選擇 [刪除]。

6. 選擇儲存，以儲存變更。

CLI

若要從清單中移除 AWS 帳戶 識別碼，請使用 `--values-to-remove` 參數。

Example 停止快照共用

下列範例會防止 AWS 帳戶 識別碼 444455556666 還原快照集。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-snapshot-attribute \  
--db-snapshot-identifier manual-snapshot1 \  
--attribute-name restore \  
--values-to-remove 444455556666
```

在 Windows 中：

```
aws rds modify-db-snapshot-attribute ^  
--db-snapshot-identifier manual-snapshot1 ^  
--attribute-name restore ^  
--values-to-remove 444455556666
```

RDS API

若要移除的共用權限 AWS 帳戶，請使用 `AttributeName` 設定為 `restore` 和 `ValuesToRemove` 參數的 [ModifyDBSnapshotAttribute](#) 作業。若要將手動快照標示為私有，請從 `all` 屬性的值清單移除 `restore` 值。

將資料庫快照資料匯出至 Amazon S3

您可以將資料庫快照資料匯出至 Amazon S3 儲存貯體。匯出程序會在背景執行，不會影響您作用中資料庫執行個體的效能。

當您匯出資料庫快照時，Amazon RDS 會從快照擷取資料，並將其存放於 Amazon S3 儲存貯體中。資料會以壓縮且一致的 Apache Parquet 格式存放。

您可以匯出所有類型的資料庫快照，包括手動快照、自動系統快照，以及由服務建立的快照。AWS Backup 根據預設會匯出快照中的所有資料。但是，您可以選擇匯出特定資料庫、結構描述或資料表集。

匯出資料後，您可以直接透過 Amazon Athena 或 Amazon Redshift Spectrum 等工具分析匯出後的資料。如需使用雅典娜讀取實木地板資料的詳細資訊，請參閱 Amazon Athena 使用者指南 SerDe 中的 [鑲木地板](#)。如需有關使用 Redshift Spectrum 來讀取 Parquet 資料的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [從單欄式資料格式的 COPY](#)。

主題

- [區域和版本可用性](#)
- [限制](#)
- [匯出快照資料概觀](#)
- [設定對 Amazon S3 儲存貯體的存取權](#)
- [將資料庫快照匯出至 Amazon S3 儲存貯體](#)
- [監控快照匯出](#)
- [取消快照匯出任務](#)
- [Amazon S3 匯出任務的失敗訊息](#)
- [對 PostgreSQL 許可錯誤進行故障診斷](#)
- [檔案命名慣例](#)
- [匯出至 Amazon S3 儲存貯體時的資料轉換](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需具備將快照匯出至 S3 功能的版本和區域可用性的詳細資訊，請參閱 [支援的區域和資料庫引擎，可將快照匯出到 Amazon RDS 中的 S3](#)。

限制

將資料庫快照資料匯出至 Amazon S3 時有下列限制：

- 您無法同時針對相同的資料庫快照執行多個匯出任務。這同時適用於完整和部分匯出。
- 不支援從使用磁帶儲存的資料庫執行個體匯出快照。
- 匯出至 S3 不支援包含冒號 (:) 的 S3 前置詞。
- 在匯出過程中，S3 檔案路徑中的以下字元將轉換為底線 (_)：

```
\ ` " (space)
```

- 如果資料庫、結構描述或資料表的名稱中包含下列字元以外的字元，則不支援部分匯出。但是，您可以匯出整個資料庫快照。
 - 拉丁字母 (A–Z)
 - 數字 (0–9)
 - 美元符號 (\$)
 - 底線 (_)
- 資料庫資料表資料欄名稱不支援空格 () 和某些字元。資料行名稱中具備下列字元的資料表會在匯出時跳過：

```
, ; { } ( ) \n \t = (space)
```

- 匯出時會略過名稱中具備斜線 (/) 的表格。
- 匯出期間，系統會略過 RDS for PostgreSQL 的暫存和未記錄資料表。
- 若資料包含接近或超過 500 MB 的大型物件 (例如 BLOB 或 CLOB)，則匯出會失敗。
- 如果資料表包含接近或大於 2 GB 的大型資料列，則在匯出期間會略過該資料表。
- 對於部分匯出，ExportOnly清單的大小上限為 200 KB。
- 強烈建議您對每個匯出任務使用唯一的名稱。如果不使用唯一的任務名稱，可能會收到下列錯誤訊息：

ExportTaskAlreadyExists錯誤：呼叫 StartExportTask作業時發生錯誤 (ExportTaskAlreadyExists)：識別碼為 **xxxxx** 的匯出工作已存在。

- 您在將快照資料匯出至 S3 時可以刪除該快照，但在匯出任務完成之前，仍會向您收取該快照的儲存費用。
- 您無法將匯出的快照資料從 S3 還原到新的資料庫執行個體。

- 每個最多可以有五個正在進行的並行資料庫快照匯出工作 AWS 帳戶。

匯出快照資料概觀

您可以使用下列程序，將資料庫快照資料匯出至 Amazon S3 儲存貯體。如需詳細資訊，請參閱下列各節。

1. 識別要匯出的快照

使用現有的自動化或手動快照，或是建立資料庫執行個體的手動快照。

2. 設定對 Amazon S3 儲存貯體的存取。

「儲存貯體」是 Amazon S3 物件或檔案的容器。如要提供存取儲存貯體的資訊，請採取下列步驟：

- a. 識別要匯出快照的目標 S3 儲存貯體。S3 儲存貯體必須與快照位於相同的 AWS 區域。如需詳細資訊，請參閱 [識別要匯出的 Amazon S3 儲存貯體](#)。
 - b. 建立 AWS Identity and Access Management (IAM) 角色，授與 S3 儲存貯體的快照匯出任務存取權。如需詳細資訊，請參閱 [使用 IAM 角色提供對 Amazon S3 儲存貯體的存取權](#)。
- ### 3. 建立伺服器端加密 AWS KMS key 的對稱加密。快照匯出工作會使用 KMS 金鑰，在將匯出資料寫入 S3 時設定 AWS KMS 伺服器端加密。

KMS 金鑰政策必須同時包含 `kms:CreateGrant` 和 `kms:DescribeKey` 許可。如需在 Amazon RDS 中使用 KMS 金鑰的詳細資訊，請參閱 [AWS KMS key 管理](#)。

如果您的 KMS 金鑰原則中有拒絕陳述式，請務必明確排除 AWS 服務主體 `export.rds.amazonaws.com`。

您可以在 AWS 帳戶中使用 KMS 金鑰，也可以使用跨帳戶 KMS 金鑰。如需詳細資訊，請參閱 [使用跨帳戶 AWS KMS key 加密 Amazon S3 匯出](#)。

- ### 4. 使用主控台或 `start-export-task` CLI 命令將快照匯出至 Amazon S3。如需詳細資訊，請參閱 [將資料庫快照匯出至 Amazon S3 儲存貯體](#)。
- ### 5. 若要存取 Amazon S3 儲存貯體中您匯出的資料，請參閱《Amazon Simple Storage Service 使用者指南》中的 [上傳、下載及管理物件](#)。

設定對 Amazon S3 儲存貯體的存取權

如要將資料庫快照資料匯出至 Amazon S3 檔案，您首先要給予快照存取 Amazon S3 儲存貯體的許可。您接著可以建立 IAM 角色，允許 Amazon RDS 服務寫入 Amazon S3 儲存貯體。

主題

- [識別要匯出的 Amazon S3 儲存貯體](#)
- [使用 IAM 角色提供對 Amazon S3 儲存貯體的存取權](#)
- [使用跨帳戶 Amazon S3 儲存貯體](#)
- [使用跨帳戶 AWS KMS key 加密 Amazon S3 匯出](#)

識別要匯出的 Amazon S3 儲存貯體

識別要匯出資料庫快照的目標 Amazon S3 儲存貯體。使用現有的 S3 儲存貯體或建立新的 S3 儲存貯體。

Note

要匯出的 S3 儲存貯體必須與快照位於相同的 AWS 區域。

如需使用 Amazon S3 儲存貯體的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的下列內容：

- [如何檢視 S3 儲存貯體的屬性？](#)
- [如何啟用 Amazon S3 儲存貯體的預設加密？](#)
- [如何建立 S3 儲存貯體？](#)

使用 IAM 角色提供對 Amazon S3 儲存貯體的存取權

在您將資料庫快照資料匯出至 Amazon S3 前，請給予快照匯出任務對 Amazon S3 儲存貯體的存取許可。

若要授予此許可，請建立 IAM 政策，提供儲存貯體的存取權，然後建立 IAM 角色並將該政策附加至其中。稍後您會指派 IAM 角色給您的快照匯出任務。

⚠ Important

如果您計劃使 AWS Management Console 用匯出快照，您可以選擇在匯出快照時自動建立 IAM 政策和角色。如需說明，請參閱「[將資料庫快照匯出至 Amazon S3 儲存貯體](#)」。

給予資料庫快照任務對 Amazon S3 的存取權限

1. 建立 IAM 政策。此政策會提供儲存貯體和物件許可，允許您的快照匯出任務存取 Amazon S3。

在政策中，包含下列必要動作，允許將檔案從 Amazon RDS 傳輸至 S3 儲存貯體：

- s3:PutObject*
- s3:GetObject*
- s3:ListBucket
- s3>DeleteObject*
- s3:GetBucketLocation

在政策中，包含下列資源，以識別 S3 儲存貯體和該儲存貯體中的物件。以下資源清單會顯示用於存取 Amazon S3 的 Amazon Resource Name (ARN) 格式。

- arn:aws:s3:::*DOC-EXAMPLE-BUCKET*
- arn:aws:s3:::*DOC-EXAMPLE-BUCKET*/*

如需為 Amazon RDS 建立 IAM 政策的詳細資訊，請參閱 [建立並使用 IAM 政策進行 IAM 資料庫存取](#)。另請參閱《IAM 使用者指南》中的[教學：建立和連接您的第一個客戶受管原則](#)。

下列 AWS CLI 命令會建立以這些選項命名 ExportPolicy 的 IAM 政策。它授予一個名為 ## 示例桶的訪問權限。

📘 Note

在您建立政策後，請記下政策的 ARN。在後續步驟中將政策附加至 IAM 角色時，您會需要此 ARN。

```
aws iam create-policy --policy-name ExportPolicy --policy-document '{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ExportPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject*",
      "s3:ListBucket",
      "s3:GetObject*",
      "s3:DeleteObject*",
      "s3:GetBucketLocation"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
  }
]
}'

```

2. 建立 IAM 角色，讓 Amazon RDS 可以代表您擔任此 IAM 角色，以存取您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

下列範例示範如何使用 AWS CLI 命令建立名為的角色 `rds-s3-export-role`。

```

aws iam create-role --role-name rds-s3-export-role --assume-role-policy-document
'{"Statement": [{"Action": "sts:AssumeRole", "Effect": "Allow", "Principal": {"Service": "export.rds.amazonaws.com"}}]}'

```

3. 將您建立的 IAM 政策附加至您建立的 IAM 角色。

下列 AWS CLI 命令會將先前建立的原則附加至名為的角色 `rds-s3-export-role`。將 *your-policy-arn* 取代之您在稍早步驟中記下的政策 ARN。

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name rds-s3-export-role
```

使用跨帳戶 Amazon S3 儲存貯體

您可以跨 AWS 帳戶使用 Amazon S3 儲存貯體。若要使用跨帳戶儲存貯體，請新增儲存貯體政策，以將存取權授予您用於 S3 匯出的 IAM 角色。如需詳細資訊，請參閱[範例 2：儲存貯體擁有者授予跨帳戶儲存貯體許可](#)。

- 將儲存貯體政策附加至您的儲存貯體，如下列範例所示。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/Admin"
      },
      "Action": [
        "s3:PutObject*",
        "s3:ListBucket",
        "s3:GetObject*",
        "s3>DeleteObject*",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET",
        "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET/*"
      ]
    }
  ]
}
```

使用跨帳戶 AWS KMS key 加密 Amazon S3 匯出

您可以使用跨帳戶 AWS KMS key 來加密 Amazon S3 匯出。首先，將金鑰政策新增至本機帳戶，然後在外部帳戶中新增 IAM 政策。如需詳細資訊，請參閱[允許其他帳戶中的使用者使用 KMS 金鑰](#)。

使用跨帳戶 KMS 金鑰

1. 將金鑰政策新增至本機帳戶。

下列範例將本機帳戶 123456789012 中的許可給予外部帳戶 444455556666 中的 ExampleRole 和 ExampleUser。

```
{
  "Sid": "Allow an external account to use this KMS key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::444455556666:role/ExampleRole",
      "arn:aws:iam::444455556666:user/ExampleUser"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:RetireGrant"
  ],
  "Resource": "*"
}
```

2. 將 IAM 政策新增至外部帳戶。

以下 IAM 政策範例允許委託人使用帳戶 123456789012 中的 KMS 金鑰來進行密碼編譯操作。若要提供此許可給帳戶 444455556666 中的 ExampleRole 和 ExampleUser，請在該帳戶中[附加政策](#)給他們。

```
{
  "Sid": "Allow use of KMS key in account 123456789012",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",

```

```
    "kms:DescribeKey",
    "kms:RetireGrant"
  ],
  "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

將資料庫快照匯出至 Amazon S3 儲存貯體

每個最多可以有五個正在進行的並行資料庫快照匯出工作 AWS 帳戶。

Note

匯出 RDS 快照可能需要一段時間，依您的資料庫類型和大小而定。匯出任務會先還原和擴展整個資料庫，然後再將資料擷取到 Amazon S3。此階段期間的工作進度會顯示為 STARTING (開始)。當任務切換到將資料匯出到 S3 時，進度會顯示為 In progress (進行中)。

匯出完成所需的時間取決於儲存在資料庫中的資料。例如，如果資料表有分散均勻的數值主索引鍵或索引資料欄，則匯出速度最快。如果資料表不含適合分割的資料欄，或是資料表只在字串型資料欄上有一個索引，則會花更多時間。因為匯出使用較慢的單一執行緒處理序，所以將會耗費更長的時間。

您可以使用 AWS Management Console、或 RDS API 將資料庫快照匯出到 Amazon S3。AWS CLI

如果您使用 Lambda 函式匯出快照集，請將 `kms:DescribeKey` 動作新增至 Lambda 函式政策。如需詳細資訊，請參閱 [AWS Lambda 許可](#)。

主控台

Export to Amazon S3 (匯出至 Amazon S3) 主控台選項僅會針對可匯出至 Amazon S3 的快照顯示。由於下列原因，快照可能無法匯出：

- S3 匯出不支援此資料庫引擎。
- S3 匯出不支援此資料庫執行個體版本。
- 建立快照的 AWS 區域不支援 S3 匯出。

匯出資料庫快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 從標籤中，選擇您希望匯出的快照類型。
4. 在快照清單中，選擇您希望匯出的快照。
5. 針對 Actions (動作)，選擇 Export to Amazon S3 (匯出至 Amazon S3)。

隨即出現 Export to Amazon S3 (匯出至 Amazon S3) 視窗。

6. 針對 Export identifier (匯出識別符)，輸入名稱以識別匯出任務。這個值也會用來做為在 S3 儲存貯體中建立的檔案名稱。
7. 選擇匯出的資料：
 - 選擇 All (全部) 來匯出快照中的所有資料。
 - 選擇 Partial (部分) 來匯出快照的特定部分。若要識別要匯出的快照部分，請針對 Identifier (識別符) 輸入一或多個資料庫、結構描述或表格，並以空格分隔。

使用下列格式：

```
database[.schema][.table] database2[.schema2][.table2] ... databasen[.scheman]
[.tablen]
```

例如：

```
mydatabase mydatabase2.myschema1 mydatabase2.myschema2.mytable1
mydatabase2.myschema2.mytable2
```

8. 針對 S3 bucket (S3 儲存貯體)，選擇要匯出的儲存貯體。

如要將匯出資料指派給 S3 儲存貯體中的資料夾路徑，請針對 S3 prefix (S3 字首) 輸入選用的路徑。

9. 針對 IAM role (IAM 角色)，您可選擇授予對您所選擇 S3 儲存貯體寫入存取權限的角色，或建立新角色。
 - 如果您透過遵循 [使用 IAM 角色提供對 Amazon S3 儲存貯體的存取權](#) 中的步驟建立了角色，請選擇該角色。

- 如果您沒有建立授予您所選擇 S3 儲存貯體寫入存取權限的角色，請選擇 Create a new role (建立新角色) 以自動建立角色。接下來，為 IAM 角色名稱 (IAM role name) 中的角色輸入名稱。
10. 針對 AWS KMS key，輸入金鑰的 ARN 以加密匯出的資料。
 11. 選擇 Export to Amazon S3 (匯出至 Amazon S3)。

AWS CLI

若要使用將資料庫快照匯出到 Amazon S3 AWS CLI，請使用具有下列必要選項的[開始匯出任務](#)命令：

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

DOC/EX AMPLE ##### S3

Example

對於LinuxmacOS、或Unix：

```
aws rds start-export-task \
  --export-task-identifier my-snapshot-export \
  --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name \
  --s3-bucket-name DOC-EXAMPLE-BUCKET \
  --iam-role-arn iam-role \
  --kms-key-id my-key
```

在 Windows 中：

```
aws rds start-export-task ^
  --export-task-identifier my-snapshot-export ^
  --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name ^
  --s3-bucket-name DOC-EXAMPLE-BUCKET ^
  --iam-role-arn iam-role ^
  --kms-key-id my-key
```

範例輸出如下。

```
{
  "Status": "STARTING",
  "IamRoleArn": "iam-role",
  "ExportTime": "2019-08-12T01:23:53.109Z",
  "S3Bucket": "my-export-bucket",
  "PercentProgress": 0,
  "KmsKeyId": "my-key",
  "ExportTaskIdentifier": "my-snapshot-export",
  "TotalExtractedDataInGB": 0,
  "TaskStartTime": "2019-11-13T19:46:00.173Z",
  "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name"
}
```

如要為快照匯出提供 S3 儲存貯體中的資料夾路徑，請在 [start-export-task](#) 命令中包含 `--s3-prefix` 選項。

RDS API

若要使用 Amazon RDS API 將資料庫快照匯出到 Amazon S3，請使用具有下列必要參數的 [StartExport](#) [任務](#) 操作：

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

監控快照匯出

您可以使用 AWS Management Console、或 RDS API 監視資料庫快照匯出。AWS CLI

主控台

監控資料庫快照匯出

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。

3. 如要檢視快照匯出清單，請選擇 Exports in Amazon S3 (Amazon S3 中的匯出) 標籤。
4. 如要檢視特定快照匯出的相關資訊，請選擇匯出任務。

AWS CLI

若要使用監視資料庫快照集匯出 AWS CLI，請使用[描述-匯出工作命令](#)。

以下範例示範如何顯示您所有快照匯出的目前資訊。

Example

```
aws rds describe-export-tasks

{
  "ExportTasks": [
    {
      "Status": "CANCELED",
      "TaskEndTime": "2019-11-01T17:36:46.961Z",
      "S3Prefix": "something",
      "ExportTime": "2019-10-24T20:23:48.364Z",
      "S3Bucket": "DOC-EXAMPLE-BUCKET",
      "PercentProgress": 0,
      "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/
bPxRfiCYEXAMPLEKEY",
      "ExportTaskIdentifier": "anewtest",
      "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
      "TotalExtractedDataInGB": 0,
      "TaskStartTime": "2019-10-25T19:10:58.885Z",
      "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-
groups-test"
    },
    {
      "Status": "COMPLETE",
      "TaskEndTime": "2019-10-31T21:37:28.312Z",
      "WarningMessage": "{\"skippedTables\": [], \"skippedObjectives\": [], \"general
\": [{\"reason\": \"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\"}]}",
      "S3Prefix": "",
      "ExportTime": "2019-10-31T06:44:53.452Z",
      "S3Bucket": "DOC-EXAMPLE-BUCKET1",
      "PercentProgress": 100,
      "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
```

```

    "ExportTaskIdentifier": "thursday-events-test",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 263,
    "TaskStartTime": "2019-10-31T20:58:06.998Z",
    "SourceArn":
"arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
  },
  {
    "Status": "FAILED",
    "TaskEndTime": "2019-10-31T02:12:36.409Z",
    "FailureCause": "The S3 bucket edgcuc-export isn't located in the current
AWS Region. Please, review your S3 bucket name and retry the export.",
    "S3Prefix": "",
    "ExportTime": "2019-10-30T06:45:04.526Z",
    "S3Bucket": "DOC-EXAMPLE-BUCKET2",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
    "ExportTaskIdentifier": "wednesday-afternoon-test",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-10-30T22:43:40.034Z",
    "SourceArn":
"arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
  }
]
}

```

若要顯示特定快照匯出的相關資訊，請在 `--export-task-identifier` 命令中包含 `describe-export-tasks` 選項。若要篩選輸出，請包含 `--Filters` 選項。如需更多選項，請參閱 [describe-export-tasks](#) 命令。

RDS API

若要顯示使用 Amazon RDS API 匯出資料庫快照的相關資訊，請使用 [DescribeExport](#) 任務操作。

若要追蹤匯出工作流程的完成或起始其他工作流程，您可以訂閱 Amazon Simple Notification Service 主題。如需 Amazon SNS 的詳細資訊，請參閱 [使用 Amazon RDS 事件通知](#)。

取消快照匯出任務

您可以使用 AWS Management Console、或 RDS API 取消資料庫快照匯出工作。AWS CLI

Note

取消快照匯出任務不會移除任何已匯出到 Amazon S3 的資料。如需如何使用主控台刪除資料的資訊，請參閱[如何從 S3 儲存貯體刪除物件？](#) 如要使用 CLI 刪除資料，請使用 [delete-object](#) 命令。

主控台

取消快照匯出任務

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇 Exports in Amazon S3 (Amazon S3 中的匯出) 標籤。
4. 選擇您希望取消的快照匯出任務。
5. 選擇 Cancel (取消)。
6. 選擇確認頁面上的 Cancel export task (取消匯出任務)。

AWS CLI

若要使用取消快照匯出工作 AWS CLI，請使用取[消匯出工作指令](#)。命令需要 `--export-task-identifier` 選項。

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
  "Status": "CANCELING",
  "S3Prefix": "",
  "ExportTime": "2019-08-12T01:23:53.109Z",
  "S3Bucket": "DOC-EXAMPLE-BUCKET",
  "PercentProgress": 0,
  "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "ExportTaskIdentifier": "my_export",
  "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
  "TotalExtractedDataInGB": 0,
  "TaskStartTime": "2019-11-13T19:46:00.173Z",
```

```
"SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

RDS API

若要使用 Amazon RDS API 取消快照匯出任務，請搭配 `ExportTaskIdentifier` 參數使用 [CancelExport](#) 任務操作。

Amazon S3 匯出任務的失敗訊息

下表說明 Amazon S3 匯出任務失敗時傳回的訊息。

失敗訊息	描述
發生未知的內部錯誤。	因為不明的錯誤、例外或故障，所以任務失敗。
將匯出任務的中繼資料寫入 S3 儲存貯體 [儲存貯體名稱] 時發生未知的內部錯誤。	因為不明的錯誤、例外或故障，所以任務失敗。
RDS 匯出無法寫入匯出任務的中繼資料，因為無法擔任 IAM 角色 [角色 ARN]。	匯出任務會擔任您的 IAM 角色，以驗證是否允許將中繼資料寫入 S3 儲存貯體。如果任務無法擔任您的 IAM 角色，則會失敗。
RDS 匯出無法使用具有 KMS 金鑰 [金鑰 ID] 的 IAM 角色 [角色 ARN]，將匯出任務的中繼資料寫入 S3 儲存貯體 [儲存貯體名稱]。錯誤代碼：[錯誤代碼]	<p>缺少一或多個許可，因此匯出任務無法存取 S3 儲存貯體。收到下列其中一個錯誤碼時會引發此失敗訊息：</p> <ul style="list-style-type: none"> • <code>AWSSecurityTokenServiceException</code> 和錯誤代碼 <code>AccessDenied</code> • <code>AmazonS3Exception</code> 和錯誤代碼 <code>NoSuchBucket</code>、<code>AccessDenied</code>、<code>KMS.KMSInvalidStateException</code>、<code>403 Forbidden</code> 或 <code>KMS.DisabledException</code> <p>這些錯誤碼表示 IAM 角色、S3 儲存貯體或 KMS 金鑰的設定錯誤。</p>
IAM 角色 [角色 ARN] 未獲得授權在 S3 儲存貯體 [儲存貯體名稱] 上呼叫 [S3 動作]。檢閱您的許可，然後重試匯出。	IAM 政策設定錯誤。缺少 S3 儲存貯體上特定 S3 動作的許可，這會導致匯出任務失敗。

失敗訊息	描述
KMS 金鑰檢查失敗。請檢查 KMS 金鑰上的憑證，然後再試一次。	KMS 金鑰憑證檢查失敗。
S3 憑證檢查失敗。檢查您的 S3 儲存貯體與 IAM 政策的許可。	S3 憑證檢查失敗。
S3 儲存貯體 [儲存貯體名稱] 無效。可能不是位於目前的 AWS 區域或不存在。檢閱您的 S3 儲存貯體名稱並重試匯出。	S3 儲存貯體無效。
S3 儲存貯體 [儲存貯體名稱] 不在目前 AWS 區域中。檢閱您的 S3 儲存貯體名稱並重試匯出。	S3 儲存貯體位於錯誤的 AWS 區域。

對 PostgreSQL 許可錯誤進行故障診斷

將 PostgreSQL 資料庫匯出至 Amazon S3 時，您可能會看到 PERMISSIONS_DO_NOT_EXIST 錯誤，指出某些資料表已略過。您在建立資料庫執行個體時指定的超級使用者，若沒有存取這些資料表的許可，通常就會發生此錯誤。

若要修正此錯誤，請執行下列命令：

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

如需超級使用者權限的詳細資訊，請參閱[主要使用者帳戶權限](#)。

檔案命名慣例

特定表格的匯出資料會以 *base_prefix/files* 的格式儲存，其中基本前綴如下：

```
export_identifier/database_name/schema_name.table_name/
```

例如：

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```


檔案命名方式有兩種慣例：

- 目前慣例：

```
batch_index/part-partition_index-random_uuid.format-based_extension
```

批處理索引是一個序列號，表示從表中讀取的一批數據。如果我們無法將您的表分割成小塊以 parallel 導出，則會有多個批處理索引。如果您的表被分區為多個表，則會發生同樣的情況。將會有許多批次索引，主資料表的每個表格分割區各一個。

如果我們可以將您的表分割成小塊以 parallel 讀取，則只會有批處理索引1文件夾。

在批處理索引文件夾中，有一個或多個包含表數據的 Parquet 文件。實木複合地板文件名的前綴是 *part-partition_index*。如果您的資料表已分割，則會有多個以分割區索引開頭的檔案 00000。

分割區索引序列中可能會有間隙。發生這種情況是因為每個分區都是從表中的範圍查詢中獲得的。如果該分區的範圍內沒有數據，則跳過該序列號。

例如，假設資料行是 id 資料表的主索引鍵，而且其最小值和最大值為 100 和 1000。當我們嘗試使用九個分區導出此表時，我們使用 parallel 查詢來讀取它，如下所示：

```
SELECT * FROM table WHERE id <= 100 AND id < 200  
SELECT * FROM table WHERE id <= 200 AND id < 300
```

這應該生成九個文件，

從 *part-00000-random_uuid.gz.parquet* 到 *part-00008-random_uuid.gz.parquet*。但是，如果 200 和之間沒有 ID 的行 350，則其中一個完成的分區是空的，並且不會為其創建任何文件。在前面的例子中，*part-00001-random_uuid.gz.parquet* 不會創建。

- 較舊的慣例：

```
part-partition_index-random_uuid.format-based_extension
```

這與當前約定相同，但沒有前 *batch_index* 綴，例如：

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet  
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet  
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

檔案命名慣例可能會有所變更。因此，在讀取目標資料表時，建議您讀取資料表基本字首內的所有內容。

匯出至 Amazon S3 儲存貯體時的資料轉換

當您將資料庫快照匯出至 Amazon S3 儲存貯體時，Amazon RDS 會以 Parquet 格式轉換資料、匯出資料，以及存放資料。如需 Parquet 的詳細資訊，請參閱 [Apache Parquet](#) 網站。

Parquet 會將所有資料以下列其中一種基本類型存放：

- BOOLEAN
- INT32
- INT64
- INT96
- FLOAT
- DOUBLE
- BYTE_ARRAY – 長度可變的位元組陣列，也稱為二進位
- FIXED_LEN_BYTE_ARRAY – 長度固定的位元組陣列，用於值具備固定大小時

Parquet 的資料類型相當少，可減少讀取和寫入格式的複雜性。Parquet 提供擴充基本類型的邏輯類型。「邏輯類型」會實作為標註，並將資料存放在 LogicalType 中繼資料欄位中。邏輯類型標註會說明如何解譯基本類型。

當 STRING 邏輯類型標註 BYTE_ARRAY 類型時，會指出位元組陣列應解譯為 UTF-8 編碼的字元字串。在匯出任務完成後，Amazon RDS 會通知您是否發生任何字串轉換。匯出的基礎資料一律與來源資料相同。但是，由於 UTF-8 中的編碼存在差異，在 Athena 等工具中讀取時，有些字元的顯示方式可能會和來源中的顯示方式不同。

如需詳細資訊，請參閱 Parquet 文件中的 [Parquet Logical Type Definitions](#)。

主題

- [MySQL 和 MariaDB 資料類型對 Parquet 的映射](#)
- [PostgreSQL 資料類型對 Parquet 的映射](#)

MySQL 和 MariaDB 資料類型對 Parquet 的映射

下表顯示轉換資料並匯出至 Amazon S3 時，MySQL 和 MariaDB 資料類型到 Parquet 資料類型的映射。

來源資料類型	Parquet 基本類型	邏輯類型標註	轉換備註
數值資料類型			
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN _BYTE_ARRAY(9)	DECIMAL(20,0)	Parquet 只支援帶正負號的類型，因此映射需要額外的位元組 (8 加 1) 來存放 BIGINT_UNSIGNED 類型。
BIT	BYTE_ARRAY		
DECIMAL	INT32	DECIMAL(p,s)	如果來源值小於 2^{31} ，則會以 INT32 存放。
	INT64	DECIMAL(p,s)	如果來源值等於或大於 2^{31} ，但小於 2^{63} ，則會以 INT64 存放。
	FIXED_LEN _BYTE_ARRAY(N)	DECIMAL(p,s)	如果來源值等於或大於 2^{63} ，則會以 FIXED_LEN_BYTE_ARRAY(N) 存放。
	BYTE_ARRAY	STRING	Parquet 不支援大於 38 的 Decimal 精確度。Decimal 值會轉換成 BYTE_ARRAY 類型中的字串，並以 UTF8 編碼。

來源資料類型	Parquet 基本類型	邏輯類型標註	轉換備註
DOUBLE	DOUBLE		
FLOAT	DOUBLE		
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL(p,s)	如果來源值小於 2^{31} ，則會以 INT32 存放。
	INT64	DECIMAL(p,s)	如果來源值等於或大於 2^{31} ，但小於 2^{63} ，則會以 INT64 存放。
	FIXED_LEN_ARRAY(N)	DECIMAL(p,s)	如果來源值等於或大於 2^{63} ，則會以 FIXED_LEN_BYTE_ARRAY(N) 存放。
	BYTE_ARRAY	STRING	Parquet 不支援大於 38 的 Numeric 精確度。這個 Numeric 值會轉換成 BYTE_ARRAY 類型的字串，並以 UTF8 編碼。
SMALLINT	INT32		

來源資料類型	Parquet 基本類型	邏輯類型標註	轉換備註
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32		
字串資料類型			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	
LINESTRING	BYTE_ARRAY		
LOBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		
VARCHAR	BYTE_ARRAY	STRING	

來源資料類型	Parquet 基本類型	邏輯類型標註	轉換備註
日期和時間資料類型			
DATE	BYTE_ARRAY	STRING	日期會轉換成 BYTE_ARRAY 類型中的字串，並以 UTF8 編碼。
DATETIME	INT64	TIMESTAMP_MICROS	
TIME	BYTE_ARRAY	STRING	TIME 類型會轉換成 BYTE_ARRAY 類型中的字串，並以 UTF8 編碼。
TIMESTAMP	INT64	TIMESTAMP_MICROS	
YEAR	INT32		
幾何資料類型			
GEOMETRY	BYTE_ARRAY		
GEOMETRYCOLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		
POLYGON	BYTE_ARRAY		
JSON 資料類型			
JSON	BYTE_ARRAY	STRING	

PostgreSQL 資料類型對 Parquet 的映射

下表顯示轉換資料並匯出至 Amazon S3 時，PostgreSQL 資料類型到 Parquet 資料類型的映射。

PostgreSQL 資料類型	Parquet 基本類型	邏輯類型標註	映射備註
數值資料類型			
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	DECIMAL 類型會轉換成 BYTE_ARRAY 類型中的字串，並以 UTF8 編碼。 此轉換是為了避免因非數字 (NaN) 而導致的資料精確度和資料值複雜性。
DOUBLE PRECISION	DOUBLE		
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT_16	
SMALLSERIAL	INT32	INT_16	
字串和相關資料類型			
ARRAY	BYTE_ARRAY	STRING	

PostgreSQL 資料類型	Parquet 基本類型	邏輯類型標註	映射備註
			陣列會轉換成字串，並以 BINARY (UTF8) 編碼。 此轉換是為了避免因非數字 (NaN) 和時間資料值而導致的資料精確度複雜性。
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	
XML	BYTE_ARRAY	STRING	
日期和時間資料類型			
DATE	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	

PostgreSQL 資料類型	Parquet 基本類型	邏輯類型標註	映射備註
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
幾何資料類型			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	
POLYGON	BYTE_ARRAY	STRING	
JSON 資料類型			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	
其他資料類型			
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	網路資料類型
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	

PostgreSQL 資料類型	Parquet 基本類型	邏輯類型標註	映射備註
INET	BYTE_ARRAY	STRING	網路資料類型
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	N/A		
PG_LSN	BYTE_ARRAY	STRING	
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

用 AWS Backup 於管理自動備份

AWS Backup 這是一項全受管備份服務，可讓您輕鬆集中並自動化雲端和內部部署 AWS 服務之間的資料備份。您可以在 AWS Backup 中管理 Amazon RDS 資料庫的備份。

Note

由管理的備份視 AWS Backup 為手動資料庫快照集，但不會計入 RDS 的資料庫快照配額中。以 AWS Backup 建立的備份名稱結尾為 `awsbackup:backup-job-number`。

如需詳細資訊 AWS Backup，請參閱 [AWS Backup 開發人員指南](#)。

若要檢視由管理的備份 AWS Backup

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇 Backup service (備份服務) 標籤。

您的 B AWS Backup ackup 會列在 [備份服務快照] 底下。

監控 Amazon RDS 執行個體中的指標

在下列各節中，您可以找到 Amazon RDS 監控概觀及如何存取指標的說明。若要了解如何監控事件、日誌和資料庫活動串流，請參閱 [在 Amazon RDS 資料庫執行個體中監控事件、日誌和串流](#)。

主題

- [在 Amazon RDS 中監控指標的概觀](#)
- [檢視執行處理狀](#)
- [檢視和回應 Amazon RDS 建議](#)
- [在 Amazon RDS 主控台中檢視指標](#)
- [在 Amazon RDS 主控台中檢視組合指標](#)
- [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)
- [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)
- [使用 Amazon RDS 專用的 Amazon DevOps 大師分析性能異常](#)
- [使用增強型監控來監控作業系統指標](#)
- [Amazon RDS 的指標參考](#)

在 Amazon RDS 中監控指標的概觀

監控是維護 Amazon RDS 及您 AWS 解決方案可靠性、可用性和效能的重要部分。為了更輕鬆地偵錯多點故障，建議您從 AWS 解決方案的所有部分收集監控資料。

主題

- [監控計畫](#)
- [效能基準](#)
- [效能指導方針](#)
- [監控工具](#)

監控計畫

開始監控 Amazon RDS，請先建立監控計畫。此計畫應該回答下列問題：

- 監控目標是什麼？
- 監控哪些資源？
- 監控這些資源的頻率為何？
- 將使用哪些監控工具？
- 誰將執行監控任務？
- 發生問題時應該通知誰？

效能基準

若要達到監控目標，您需要建立一個基準。若要這麼做，請在 Amazon RDS 環境中以不同的時間、不同的負載條件下測量效能。您可以監控如下所示的指標：

- 網路輸送量
- 用戶端連線
- 用於讀取、寫入或中繼資料操作的 I/O
- 資料庫執行個體的爆量點數餘額

建議您儲存 Amazon RDS 的歷史效能資料。使用儲存的資料，您可以比較當前的效能與過去的趨勢。您也可以區分正常與異常的效能模式，並設計技術來解決問題。

效能指導方針

一般來說，效能指標的可接受值依據您應用程式相對於基準所執行的內容而定。調查距離基準的一致或趨勢變異。下列指標通常是效能問題的來源：

- 高 CPU 或 RAM 耗用量 – CPU 或 RAM 耗用量若符合應用程式的目標 (如輸送量或並行) 且預期的值較高時，這樣的消耗量值就可能是合理的。
- 磁碟空間消耗量 – 如果使用的空間持續保持在等於或高於總磁碟空間的 85%，請調查磁碟空間消耗量。看看從執行個體刪除資料或將資料封存至不同的系統來釋出空間是否可行。
- 網路流量 – 對於網路流量，請洽系統管理員，以了解您的網域網路和網際網路連線預期的輸送量。調查網路流量的傳輸量是否如預期一致地降低。
- 資料庫連線 – 如果您看到大量使用者連線，同時執行個體效能下降且回應時間延長，請考慮限制資料庫連線。資料庫執行個體使用者連接的最佳數量，將因執行個體類別和要執行的操作複雜性而不同。若要判定資料庫連線的數目，方法是將資料庫執行個體與 `User Connections` 參數設為 0 (無限制) 以外之值的參數群組建立關聯。您可以使用現有的參數群組或建立新的參數群組。如需更多詳細資訊，請參閱 [使用參數群組](#)。
- IOPS 指標 – IOPS 指標的預期值視磁碟規格和伺服器組態而定，因此請使用您的基準來了解何謂典型。調查值是否與您的基準一致地不同。為獲得最佳 IOPS 效能，請確定您的一般工作集將放入記憶體中，以將讀取和寫入操作降到最低。

當效能落在您建立的基準之外時，您可能需要進行變更，以最佳化工作負載的資料庫可用性。例如，您可能需要變更資料庫執行個體的執行個體類別。或者，您可能需要變用戶端可用的資料庫執行個體和讀取複本的數目。

監控工具

監控對於維護 Amazon RDS 及其他 AWS 解決方案的可靠性、可用性和效能至關重要。AWS 提供各種監控工具，可讓您監看 Amazon RDS、在發現錯誤時回報，並適時採取自動動作。

主題

- [自動化監控工具](#)
- [手動監控工具](#)

自動化監控工具

建議您盡可能自動化監控任務。

主題

- [Amazon RDS 執行個體狀態和建議](#)
- [Amazon RDS Amazon 極光的亞指標](#)
- [Amazon RDS Performance Insights 和作業系統監控](#)
- [整合服務](#)

Amazon RDS 執行個體狀態和建議

您可以使用下列自動化工具來監看 Amazon RDS，並在發生錯誤時回報：

- Amazon RDS 執行個體狀態 – 檢視執行個體目前狀態的詳細資訊，方法是使用 Amazon RDS 主控台、AWS CLI 或 RDS API。
- Amazon RDS 建議 — 回應針對資料庫資源提供的自動化建議，例如資料庫執行個體、僅供讀取複本，以及資料庫參數群組。如需詳細資訊，請參閱[檢視和回應 Amazon RDS 建議](#)。

Amazon RDS Amazon 極光的亞指標

Amazon RDS Amazon 與 Amazon 整合以 CloudWatch 提供額外的監控功能

- Amazon CloudWatch — 此服務可即時監控您的 AWS 資源和執行 AWS 的應用程式。您可以將以下 Amazon CloudWatch 功能與 Amazon RDS Amazon 搭配使用：
 - 亞馬遜 CloudWatch 指標 — Amazon RDS Amazon 會針對 CloudWatch 每個作用中資料庫自動將指標傳送到每分鐘。在中，您不會收到 Amazon RDS 指標的額外費用 CloudWatch。如需詳細資訊，請參閱 [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。
 - Amazon CloudWatch 警示 — 您可以在特定時間段內觀看單個 Amazon RDS 指標。然後，您可以根據相對於您所設定臨界值的指標值執行一或多個動作。如需詳細資訊，請參閱 [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。

Amazon RDS Performance Insights 和作業系統監控

您能夠使用下列自動化工具來監控 Amazon RDS 效能：

- Amazon RDS Performance Insights：評估資料庫的負載，以及判斷在何時何處採取動作。如需詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。
- Amazon RDS Enhanced Monitoring (Amazon RDS 增強型監控)：即時查看作業系統的指標。如需詳細資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

整合服務

以下 AWS 服務會與 Amazon RDS 整合：

- Amazon EventBridge 是一種無伺服器事件匯流排服務，可讓您輕鬆地將應用程式與各種來源的資料連接起來。如需詳細資訊，請參閱[監控 Amazon RDS 事件](#)。
- Amazon CloudWatch 日誌可讓您從 Amazon RDS Amazon 執行個體和其他來源監控 CloudTrail、存放和存取日誌檔。如需詳細資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。
- AWS CloudTrail 擷取您 AWS 帳戶 發出或代表發出的 API 呼叫和相關事件，並傳送日誌檔案至您指定的 Amazon S3 儲存貯體。如需詳細資訊，請參閱[在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)。
- 資料庫活動串流是 Amazon RDS 功能，可在 Oracle 資料庫執行個體中提供活動串 near-real-time 流。如需詳細資訊，請參閱[使用資料庫活動串流來監控 Amazon RDS](#)。

手動監控工具

您需要手動監視 CloudWatch 警報未涵蓋的項目。Amazon RDS AWS Trusted Advisor 和其他AWS主控台儀表板可提供您AWS環境狀態的 at-a-glance 檢視。CloudWatch建議您也檢查資料庫執行個體上的日誌檔。

- 從 Amazon RDS 主控台中，您可以監控資源的下列項目：
 - 資料庫執行個體的連線數目
 - 資料庫執行個體的讀取和寫入操作數量
 - 資料庫執行個體目前正在使用的儲存體數量
 - 正在針對資料庫執行個體使用的記憶體和 CPU 數量
 - 進出資料庫執行個體的網路流量
- 從 Trusted Advisor 儀表板中，您可以檢閱下列成本最佳化、安全性、容錯，以及效能提升檢查：
 - Amazon RDS 閒置資料庫執行個體
 - Amazon RDS 安全群組存取風險
 - Amazon RDS 備份
 - Amazon RDS 異地同步備份

如需這些檢查的詳細資訊，請參閱 [Trusted Advisor 最佳實務 \(檢查\)](#)。

- CloudWatch 主頁顯示：
 - 目前警示與狀態

- 警示與資源的圖表
- 服務運作狀態

此外，您可以使用執行 CloudWatch 以下操作：

- 建立 [自訂儀表板](#) 以監控您注重的服務。
- 用於疑難排解問題以及探索驅勢的圖形指標資料。
- 搜尋與瀏覽您的所有 AWS 資源指標。
- 建立與編輯要通知發生問題的警示。

檢視執行處理狀

您可以使用 Amazon RDS 主控台快速存取資料庫執行個體的狀態。

主題

- [檢視 Aurora 叢集中 資料庫執行個體狀態](#)

檢視 Aurora 叢集中 資料庫執行個體狀態

中的資料庫執行個體的状态指示資料庫執行個體的運作状态。您可以使用下列程序在 Amazon RDS 主控台中檢視體状态、AWS CLI 命令或 API 操作。

Note

Amazon RDS 也使用另一個状态，即顯示在 Amazon RDS 主控台的 Maintenance (維護) 欄中的 maintenance status (維護状态)。此值指出需要套用至資料庫執行個體的任何維護修補程式的状态。維護状态與資料庫執行個體的状态互不相關。如需維護状态的詳細資訊，請參閱[套用資料庫執行個體的更新](#)。

您可以在下表中找到資料庫執行個體的可能状态值。此表還列出計費方式為依資料庫執行個體和儲存計費、只針對儲存計費、或不計費。在全是資料庫執行個體状态的情況下，一律要為備份用途付費。

資料庫執行個體状态	計費	描述
可用性	計費	資料庫執行個體運作正常可供使用。
Backing-up (備份)	計費	目前正在備份資料庫執行個體。
Configuring-enhanced-monitoring (設定增強型監控)	計費	正在啟用或停用此資料庫執行個體的增強型監控。
Configuring-iam-database-auth	計費	AWS Identity and Access Management (IAM) 此資料庫執行個體的資料庫身份驗證正在啟用或停用。
Configuring-log-exports	計費	此資料庫執行個體已啟用或停用將 CloudWatch 日誌檔發佈到 Amazon 日誌。
Converting-to-vpc	計費	資料庫執行個體正在從不是在 Amazon Virtual Private Cloud (Amazon VPC) 中的資料庫執行個體，轉換成位於 Amazon VPC 中的資料庫執行個體。
正在建立	不計費	正在建立資料庫執行個體。資料庫執行個體正在建立時無法存取。

資料庫執行個體狀態	計費	描述
Delete-precheck	不計費	Amazon RDS 正在驗證僅供讀取複本是否狀態良好且可安全刪除。
正在刪除	不計費	正在刪除資料庫執行個體。
失敗	不計費	資料庫執行個體失效，Amazon RDS 無法復原它。執行point-in-time 還原至資料庫執行個體的最新可還原時間，以復原資料。
Inaccessible-encryption-credentials	不計費	無法存取或復原 AWS KMS key 用於加密或解密資料庫執行個體的資料庫執行個體。
Inaccessible-encryption-credentials-recoverable	針對儲存計費	無法存取用來加密或解密資料庫執行個體的 KMS 金鑰。但是，如果 KMS 金鑰處於作用中狀態，則重新啟動資料庫執行個體可以將其復原。 如需詳細資訊，請參閱 加密資料庫執行個體 。
Incompatible-network (不相容網路)	不計費	Amazon RDS 正在嘗試對資料庫執行個體執行復原動作，但失敗，因為 VPC 所處的狀態無法完成此動作。例如，如果子網路中的所有可用 IP 地址都在使用中，Amazon RDS 無法取得資料庫執行個體的 IP 地址，就會出現此狀態。
Incompatible-option-group (不相容選項群組)	計費	Amazon RDS 嘗試套用選項群組變更但不成功，且 Amazon RDS 無法轉返到上一個選項群組狀態。如需詳細資訊，請查看資料庫執行個體的 Recent Events (最近事件) 清單。例如，如果選項組包含 TDE 等選項，而資料庫執行個體沒有加密資訊，就會出現此狀態。
Incompatible-parameters (不相容參數)	計費	因為資料庫執行個體的資料庫參數群組中指定的參數與資料庫執行個體不相容，Amazon RDS 無法啟動資料庫執行個體。請回復參數變更，或使之與資料庫執行個體相容，以恢復存取您的資料庫執行個體。如需有關不相容參數的詳細資訊，請查看資料庫執行個體的 Recent Events (最近事件) 清單。

資料庫執行個體狀態	計費	描述
Incompatible-restore (不相容還原)	不計費	Amazon RDS 無法進行 point-in-time 還原。此狀態的常見原因包括使用臨時資料表，將 MyISAM 資料表與 MySQL 一起使用，或將 Aria 資料表與 MariaDB 一起使用。
Insufficient-capacity (容量不足)	不計費	Amazon RDS 無法建立您的執行個體，因為目前無法使用足夠的容量。若要在具有相同執行個體類型的相同可用區域中建立資料庫執行個體，請刪除您的資料庫執行個體，等候幾個小時，然後嘗試再次建立。或者，使用不同的執行個體類別或可用區域建立新的執行個體。
Maintenance (維護)	計費	Amazon RDS 正在將維護更新套用到資料庫執行個體。此狀態用於 RDS 事先排程的執行個體層級維護。
Modifying (正在修改)	計費	由於客戶請求修改資料庫執行個體，正在修改資料庫執行個體。
Moving-to-vpc	計費	正在將資料庫執行個體移到新的 Amazon Virtual Private Cloud (Amazon VPC)。
Rebooting (重新開機中)	計費	由於客戶請求或 Amazon RDS 程序需要重新啟動資料庫執行個體，正在重新啟動資料庫執行個體。
Resetting-master-credentials (重新設定主要登入資料)	計費	由於客戶請求重設資料庫執行個體的主要登入資料，正在重設主要登入資料。
重新命名	計費	由於客戶請求重新命名資料庫執行個體，正在重新命名資料庫執行個體。
Restore-error (還原錯誤)	計費	資料庫執行個體嘗試從快照還原到 point-in-time 或時發生錯誤。
啟動	針對儲存計費	正在啟動資料庫執行個體。

資料庫執行個體狀態	計費	描述
已停止	針對儲存計費	已停止資料庫執行個體。
Stopping (正在停止)	針對儲存計費	正在停止資料庫執行個體。
Storage-config-upgrade	計費	資料庫執行個體的儲存檔案系統組態正在升級中。此狀態僅適用於藍/綠部署內的綠色資料庫，或資料庫執行個體僅供讀取複本。
Storage-full (儲存已滿)	計費	資料庫執行個體已達到其儲存容量配置。這是一個緊急狀態，我們建議您立即解決此問題。做法是修改資料庫執行個體以擴展儲存空間。為了避免這種情況，請設置 Amazon CloudWatch 警報以在儲存空間不足時發出警告。
Storage-optimization (儲存最佳化)	計費	Amazon RDS 正在最佳化您資料庫執行個體的儲存空間。儲存最佳化程序通常很短，但有時甚至可能會超過 24 小時。 在儲存最佳化期間，資料庫執行個體仍然可用。儲存空間最佳化是一項不會影響執行個體可用性的背景程序。
Upgrading (正在升級)	計費	正在升級資料庫引擎版本。

主控台

檢視資料庫執行個體的狀態

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

Databases page (資料庫頁面) 會與資料庫執行個體清單一起顯示。針對每個資料庫執行個體，會顯示狀態值。

Databases			
<input type="text" value="Filter by databases"/>			
<input type="checkbox"/>	DB identifier	Role	Status
<input type="radio"/>	database-1	Instance	Stopped
<input type="radio"/>	database-2	Instance	Creating
<input type="radio"/>	database-3	Instance	Available
<input type="radio"/>	database-4	Instance	Available
<input type="radio"/>	database-5	Instance	Configuring-enhanced-monitoring

CLI

若要使用檢視資料庫執行個體及其狀態資訊 AWS CLI，請使用[描述-DB 執行個體命令](#)。例如，下面的 AWS CLI 命令列出了所有的數據庫實例信息。

```
aws rds describe-db-instances
```

若要檢視特定資料庫執行個體及其狀態，請搭配下列選項呼叫 [describe-db-instances](#) 命令：

- `DBInstanceIdentifier` – 資料庫執行個體名稱。

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

若只要檢視所有資料庫執行個體的狀態，請在中使用下列查詢 AWS CLI。

```
aws rds describe-db-instances --query 'DBInstances[*].  
[DBInstanceIdentifier,DBInstanceStatus]' --output table
```

API

若要使用 Amazon RDS API 檢視資料庫執行個體的狀態，請呼叫 [DescribeDBInstances](#) 操作。

檢視和回應 Amazon RDS 建議

Amazon RDS Amazon 針對資料庫資源 (例如資料庫執行個體、資料僅供讀取複本和資料庫參數群組) 提供自動化建議。這些建議會分析、資料庫執行個體組態、使用情形及效能資料，以提供最佳實務指南。

Amazon RDS Performance Insights 可監控特定指標，並透過分析指定資源可能有問題的層級來自動建立閾值。當新的測量結果值在指定期間內跨越預先定義的臨界值時，Performance Insights 會產生主動式建議。此建議有助於防止 future 的資料庫效能影響。例如，當連線到資料庫的工作階段未執行作用中的工作，但可能會封鎖資料庫資源時，針對 RDS for PostgreSQL 執行個體產生「交易閒置」建議。若要接收主動式建議，您必須開啟具有付費方案保留期的 Performance Insights。如需有關開啟 Performance Insights 的資訊，請參閱[開啟和關閉 Amazon RDS 的 Performance Insights](#)。如需 Performance Insights 見的定價和資料保留的相關資訊，請參閱[Performance Insights 的定價和資料保留](#)。

DevOpsGuru for RDS 監控某些指標，以檢測指標的行為何變得非常不尋常或異常。這些異常被報告為帶有建議的被動洞察。例如，DevOpsGuru for RDS 可能會建議您考慮增加 CPU 容量，或調查導致資料庫負載的等待事件。DevOpsRDS 的大師還提供基於閾值的主動式建議。對於這些建議，您必須開啟 RDS DevOps 專用的 Guru。如需開啟 RDS 版 DevOps Guru 的相關資訊，請參閱[開啟 DevOps Guru 並指定資源涵蓋範圍](#)。

建議將處於下列任一狀態：作用中、已關閉、擱置中或已解決。已解決的建議可在 365 天內使用。

您可以檢視或關閉建議。您可以立即套用以組態為基礎的作用中建議、在下一個維護時段中排程，或將其關閉。對於以閾值為基礎的主動式和以機器學習為基礎的被動式建議，您需要檢閱問題的建議原因，然後執行建議的動作來修正問題。

主題

- [檢視 Amazon RDS 建議](#)
- [回應 Amazon RDS 建議](#)

檢視 Amazon RDS 建議

Amazon RDS 會在資源建立或修改時，就資源產生建議。

以下區域支援以組態為基礎的建議：

- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)
- 亞太區域 (孟買)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太區域 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- Europe (Paris)
- 南美洲 (聖保羅)

您可以在下表中找到以組態為基礎的建議範例。

Type	描述	建議	需要停機	其他資訊
磁量正在使用中	您的資料庫執行個體使用磁性儲存。不建議大多數資料庫執行個體使用磁性儲存。選擇不同的儲存類型：一般用途 (SSD) 或佈建 IOPS。	選擇不同的儲存類型：一般用途 (SSD) 或佈建 IOPS。	是	Amazon EC2 文件中的 上一代磁碟區 。

Type	描述	建議	需要停機	其他資訊
資源自動備份已關閉	資料庫執行個體的自動備份功能不會開啟。建議使用自動備份，因為它們可以 point-in-time 恢復您的數據庫實例。	開啟保留期最多 14 天的自動備份。	是	啟用自動備份 在資料庫部落格上 揭開 Amazon RDS 備份儲存成本的神秘面紗 AWS
需要升級引擎次要版本	您的資料庫資源沒有執行最新的次要資料庫引擎版本。最新的次要版本包含最新的安全性修正和其他改進。	升級至最新的引擎版本。	是	升級資料庫執行個體引擎版本
增強型監控功能已關閉	您的資料庫資源未開啟增強型監控。增強型監控針對監控及疑難排解，提供即時的作業系統指標。	開啟增強型監控。	否	使用增強型監控來監控作業系統指標

Type	描述	建議	需要停機	其他資訊
儲存區加密已關閉	<p>Amazon RDS 使用您在金鑰管理服務 (AWS KMS) 中 AWS 管理的金鑰，為所有資料庫引擎支援靜態加密。在具有 Amazon RDS 加密的作用中資料庫執行個體上，儲存在儲存中的靜態資料會加密，類似於自動備份、僅供讀取複本和快照。</p> <p>如果在建立資料庫執行個體時未開啟加密，您必須先建立並還原資料庫執行個體已解密快照的加密副本，然後再開啟加密。</p>	為資料庫執行個體開啟靜態資料加密。	是	Amazon RDS 中的安全 複製資料庫快照
Performance Insights 已關閉	Performance Insights 可監控資料庫執行個體負載，以協助您分析和解決資料庫效能問題。我們建議您開啟 Performance Insights。	開啟績效詳情。	否	在 Amazon RDS 上使用績效詳情監控資料庫負載

Type	描述	建議	需要停機	其他資訊
資料庫執行個體關閉儲存自動調度資源	您的資料庫執行個體未開啟儲存自動調度資源功能。當資料庫工作負載增加時，RDS 儲存區會自動調整資源，以零停機時間自動擴展儲存容量。	使用指定的最大儲存臨界值開啟 Amazon RDS 儲存自動調度資源	否	使用 Amazon RDS 儲存體自動擴展自動管理容量
RDS 資源需要主要版本更新	不支援資料庫引擎目前主要版本的資料庫。我們建議您升級至包含新功能和增強功能的最新主要版本。	升級至資料庫引擎的最新主要版本。	是	升級資料庫執行個體引擎版本 使用 Amazon RDS 藍/綠部署進行資料庫更新
需要更新 RDS 資源執行個體類別	您的資料庫執行個體正在執行舊代資料庫執行個體類別。我們已將舊一代的資料庫執行個體類別取代為具有更高成本、效能或兩者兼具的資料庫執行個體類別。建議您使用新一代的資料庫執行個體類別執行資料庫執行個體。	升級資料庫執行個體類別。	是	資料庫執行個體類別的支援資料庫引擎
使用已包含授權的終止支援引擎版本的 RDS 資源	建議您將主要版本升級至 Amazon RDS 支援的最新引擎版本，以繼續使用目前的授權支援。目前的授權不支援資料庫的引擎版本。	我們建議您將資料庫升級到 Amazon RDS 中的最新支援版本，以繼續使用授權模型。	是	Oracle 主要版本升級

Type	描述	建議	需要停機	其他資訊
未使用異地同步備份部署的資料庫	建議您使用多可用區部署。多可用區部署可增強資料庫執行個體的可用性和耐久性。 。	設定受影響資料庫執行個體的異地同步備份	否 進行此變更時，不會發生停機。但是，可能會對效能產生影響。如需更多資訊，請參閱 將資料庫執行個體修改為多可用區域資料庫執行	Amazon RDS 異地同步備份的定價

Type	描述	建議	需要停機	其他資訊
			個體部署	
數據庫內存參數從默認值發散	<p>資料庫執行個體的記憶體參數與預設值明顯不同。這些設定可能會影響效能並導致錯誤。</p> <p>建議您將資料庫執行個體的自訂記憶體參數重設為資料庫參數群組中的預設值。</p>	將記憶體參數重設為預設值。	否	在 AWS 資料庫部落格上 設定適用於 MySQL 的 Amazon RDS for MySQL 效能參數的最佳實務
InnoDB_Change_Buffering 使用小於最佳值的參數	變更緩衝可讓 MySQL 資料庫執行個體延遲幾次寫入，這是維護次要索引所需的寫入。此功能在磁碟速度較慢的環境中非常有用。變更緩衝組態稍微改善了資料庫效能，但在升級期間造成當機復原延遲和較長的關機時間。	在資料庫 InnoDB_Change_Buffering 參數群組 NONE 中將參數值設定為。	否	在 AWS 資料庫部落格上 設定適用於 MySQL 的 Amazon RDS for MySQL 效能參數的最佳實務
查詢快取參數已開啟	當變更需要清除查詢快取時，您的資料庫執行個體將會停止。大部分工作負載並不會受益於查詢快取。MySQL 8.0 版已移除查詢快取。我們建議您將查詢_快取類型參數設定為 0。	在資料庫 query_cache_type 參數群組 0 中將參數值設定為。	是	在 AWS 資料庫部落格上 設定適用於 MySQL 的 Amazon RDS for MySQL 效能參數的最佳實務

Type	描述	建議	需要停機	其他資訊
log_output 參數設定為表格	當設定log_output 為時TABLE，使用的儲存空間會比設定log_output 為的時間多FILE。建議您將參數設定為FILE，以避免達到儲存區大小限制。	在資料庫log_output 參數群組FILE中將參數值設定為。	否	MySQL 資料庫日誌檔案
參數群組不使用大量頁面	大型分頁可以增加資料庫延展性，但您的資料庫執行個體並未使用大型分頁。建議您在資料庫執行use_large_pages 個體的資料庫參數群組ONLY中將參數值設定為。	在資料庫use_large_pages 參數群組ONLY中將參數值設定為。	是	對 RDS for Oracle 執行個體開啟巨型分頁
autovacuum 參數已關閉	資料庫執行個體資料庫的自動真空參數已關閉。關閉自動真空可增加工作台和指數膨脹，並影響效能。 建議您開啟資料庫參數群組中的自動真空功能。	開啟資料庫參數群組中的自動真空參數。	否	透過資料庫部落格了解 Amazon RDS for PostgreSQL 環境中的自動真空 AWS

Type	描述	建議	需要停機	其他資訊
synchronous_commit 參數已關閉	<p>當synchronous_commit 參數關閉時，資料可能會在資料庫損毀時遺失。資料庫的耐久性存在風險。</p> <p>建議您開啟synchronous_commit 參數。</p>	<p>開啟資料庫synchronous_commit 參數群組中的參數。</p>	是	<p>Amazon Aurora PostgreSQL 參數：資料庫部落格上的複寫、安全性和記錄 AWS</p>
track_counts 參數已關閉	<p>關閉track_counts 參數時，資料庫不會收集資料庫活動統計資料。自動清空功能需要這些統計資料才能正常運作。</p> <p>建議您將 track_counts 參數設定為 1。</p>	<p>將track_counts 參數設定為1。</p>	否	<p>執行階段統計資料</p>
enable_indexonlyscan 參數已關閉	<p>查詢規劃工具或最佳化工具在關閉時無法使用僅索引掃描計劃類型。</p> <p>建議您將enable_indexonlyscan 參數值設定為1。</p>	<p>將enable_indexonlyscan 參數值設定為1。</p>	否	<p>PostgreSQL 的供需規劃員方法組態</p>

Type	描述	建議	需要停機	其他資訊
enable_in dexscan 參 數已關閉	查詢規劃工具或最佳化工具在關閉索引掃描計劃類型時無法使用。 我們建議您將enable_in dexscan 值設定為1。	將enable_in dexscan 參數值設定為1。	否	PostgreSQL 的供需規劃員方法組態
innodb_f lush_log_a t_trx 參數 已關閉	資料庫執行個體的innodb_f lush_log_a t_trx 參數值不是安全值。此參數控制提交操作至磁碟的持續性。 建議您將 innodb_f lush_log_at_trx 參數設定為 1。	將innodb_f lush_log_a t_trx 參數值設定為1。	否	在 AWS 資料庫部落格上設定適用於 MySQL 的 Amazon RDS for MySQL 效能參數的最佳實務
sync_binl og 參數已關 閉	在資料庫執行個體中確認交易確認之前，不會強制執行二進位記錄到磁碟的同步處理。 建議您將sync_binl og 參數值設定為1。	將sync_binlog 參數值設定為1。	否	在 AWS 資料庫部落格上為 Amazon RDS for MySQL 設定複寫參數的最佳實務

Type	描述	建議	需要停機	其他資訊
innodb_stats_persistent 參數已關閉	<p>您的資料庫執行個體未設定將 InnoDB 統計資料保留於磁碟。如果不儲存統計資料，則每次執行個體重新啟動並存取資料表時，都會重新計算這些統計資料。這會導致查詢執行計劃的變化。您可以在資料表層級修改此全域參數的值。</p> <p>建議您將 innodb_stats_persistent 參數值設定為 0N。</p>	將 innodb_stats_persistent 參數值設定為 0N。	否	在 AWS 資料庫部落格上設定適用於 MySQL 的 Amazon RDS for MySQL 效能參數的最佳實務 最佳實務
innodb_open_files 參數低	<p>該 innodb_open_files 參數控制 InnoDB 一次可以打開的文件的數量。InnoDB 打開所有的日誌和系統表空間文件時 mysqld 正在運行。</p> <p>針對 InnoDB 一次能開啟的最大檔案數量，您的資料庫執行個體設定值很低。建議您將 innodb_open_files 參數設定為 65 的下限。</p>	將 innodb_open_files 參數設定為的最小值 65。	是	InnoDB 開啟檔案

Type	描述	建議	需要停機	其他資訊
max_user_connections 參數低	<p>針對每個資料庫帳戶能同時連線的數量上限，您的資料庫執行個體設定值很低。</p> <p>我們建議將max_user_connections 參數設定為大於的數字5。</p>	將max_user_connections 參數值增加到大於的數字5。	是	設定 MySQL 的帳號資源限制
僅供讀取複本以可寫入模式開啟	<p>您的資料庫執行個體具有可寫入模式的僅供讀取複本，可從用戶端進行更新。</p> <p>建議您將read_only 參數設定為，TrueIfReplica 以便僅供讀取複本不處於可寫入模式。</p>	將read_only 參數值設定為TrueIfReplica 。	否	在 AWS 資料庫部落格上為 Amazon RDS for MySQL 設定複寫參數的最佳實務

Type	描述	建議	需要停機	其他資訊
innodb_default_row_format 參數設定不安全	<p>您的資料庫執行個體遇到一個已知問題：在 MySQL 版本低於 8.0.26 中建立的資料表，當索引超過 767 個位元組時，row_format 設為 COMPACT 或 REDUNDANT 將無法存取且無法復原。</p> <p>建議您將 innodb_default_row_format 參數值設定為 DYNAMIC。</p>	將 innodb_default_row_format 參數值設定為 DYNAMIC。	否	MySQL 中的變化
general_log 參數已打開	<p>資料庫執行個體的一般記錄已開啟。此設定在疑難排解資料庫問題時很有用。不過，開啟一般記錄會增加 I/O 作業和配置的儲存空間量，這可能會導致爭用和效能降低。</p> <p>檢查您的一般記錄用法需求。建議您將 general_log 參數值設定為 0。</p>	檢查您的一般記錄用法需求。如果不是強制性的，建議您將 general_log 參數值設定為 0。	否	RDS for MySQL 資料庫日誌概觀

Type	描述	建議	需要停機	其他資訊
RDS 執行個體佈建不足的系統記憶體容量	建議您調整查詢以使用較少的記憶體，或使用配置記憶體較高的資料庫執行個體類型。當執行個體的記憶體不足時，資料庫效能會受到影響。	使用記憶體容量較高的資料庫執行個體	是	<p>在 AWS 資料庫部落格上垂直和水平擴展 Amazon RDS 執行個體</p> <p>Amazon RDS 實例類型</p> <p>Amazon RDS 定價</p>
針對系統 CPU 容量佈建不足的 RDS 執行個體	建議您調整查詢以使用較少的 CPU，或修改資料庫執行個體，以使用配置較高 vCPUs 的資料庫執行個體類別。資料庫執行個體 CPU 不足時，資料庫效能可能會下降。	使用 CPU 容量更高的資料庫執行個體	是	<p>在 AWS 資料庫部落格上垂直和水平擴展 Amazon RDS 執行個體</p> <p>Amazon RDS 實例類型</p> <p>Amazon RDS 定價</p>
RDS 資源未正確使用連線集區	我們建議您啟用 Amazon RDS 代理，以有效地集區和共用現有的資料庫連線。如果您已經為資料庫使用 Proxy，請正確設定它，以改善跨多個資料庫執行個體的連線共用和負載平衡。RDS Proxy 可協助降低連線耗盡和停機的風險，同時提升可用性和延展性。	啟用 RDS 代理或修改您現有的代理伺服器組態	否	<p>在 AWS 資料庫部落格上垂直和水平擴展 Amazon RDS 執行個體</p> <p>使用 Amazon RDS 代理</p> <p>Amazon RDS 代理定價</p>

Type	描述	建議	需要停機	其他資訊
RDS 實例正在創建過多的臨時對象	建議您調整工作負載以避免建立過多的暫存物件，或切換至支援最佳化讀取的 RDS 執行個體類別。RDS 最佳化讀取可改善涉及大量暫存物件和/或大型暫存物件的工作負載的資料庫效能。評估您的工作負載，以判斷使用具有 RDS 最佳化讀取的執行個體是否有益於您的資料庫	搭配 RDS 最佳化讀取使用資料庫執行個體類型	是	Amazon RDS 實例類型 使用 Amazon RDS 優化讀取來改善適用於 MySQL 的查詢效能 運用 Amazon RDS 最佳化讀取功能改善適用於 MariaDB 的 RDS 查詢效能 PostgreSQL 用 Amazon RDS 最佳化讀取功能改善 RDS 版的查詢效能
RDS 執行個體針對系統 IOPS 容量佈建不足	建議您調整資料庫工作負載以減少 IOPS，或將資料庫執行個體擴展為預設 IOPS 限制較高的類型。目前的資料庫執行個體無法支援佈建 IOPS，或資料庫工作負載具有高 IOPS 使用率。	使用預設 IOPS 限制較高的資料庫執行個體類型	是	Amazon RDS 實例類型 Amazon RDS 數據庫實例存儲 資料庫載入
RDS 執行個體佈建的 Amazon EBS 磁碟區不足	建議您調整資料庫工作負載以減少 IOPS 或增加資料庫的佈建 IOPS。當 IOPS 使用率接近佈建 IOPS 時，資料庫效能可能會下降。	為資料庫執行個體佈建更多 IOPS	是	Amazon RDS 實例類型 Amazon RDS 數據庫實例存儲 資料庫載入

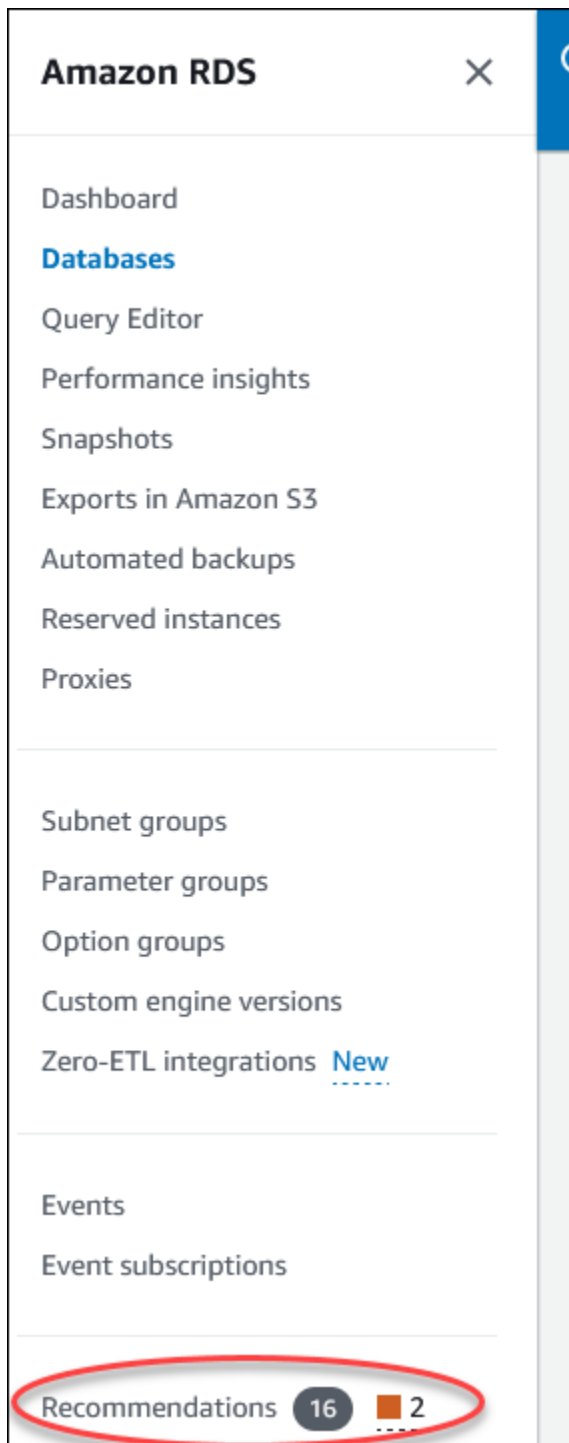
Type	描述	建議	需要停機	其他資訊
RDS 執行個體佈建不足的輸送量容量	建議您調整資料庫工作負載，以減少輸送量或增加資料庫的佈建輸送量。輸送量使用率接近佈建的輸送量時，資料庫效能可能會受到影響。	為資料庫執行個體佈建更多輸送量	是	Amazon RDS 實例類型 Amazon RDS 數據庫實例存儲 資料庫載入

使用 Amazon RDS 主控台，您可以檢視針對資料庫資源的 Amazon RDS 建議。

主控台

若要檢視 Amazon RDS Amazon 建議

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，執行下列任一項作業：
 - 選擇「建議」。您可以在「建議」旁邊找到適用於您資源的使用中建議數目，以及上個月產生的嚴重性最高的建議數目。若要尋找每個嚴重性的作用中建議數目，請選擇顯示最高嚴重性的數目。



依預設，「建議」頁面會顯示上個月的新建議清單。Amazon RDS Amazon 會針對您帳戶中的所有資源提供建議，並根據其嚴重性對建議進行排序。

Recommendations (16) Info

The list of recommendations which include best practices for resource configuration, threshold based insights when Performance Insights is using the paid tier, and anomalous DB load detection when DevOps Guru for RDS is turned on.

Filter by text or property (example: Severity) Active Last modified Last 1 month

Severity	Detection	Recommendation	Impact	Category	Start time
Medium	The InnoDB history list length increased sigr	Identify and address long-running transa Don't shut down the database	Queries may run : Shut-down may t	Performance e...	3 days ago
Medium	High DB Load on dgr-reactive-test-final-ins	Investigate 1 wait event Tune application workload	Reduced database p	Performance e...	21 days ago
Informational	18 resources don't have Enhanced Monitorir	Turn on Enhanced Monitoring	Reduced operational	Operational ex...	2 months ago

0 recommendations selected

您可以選擇建議來檢視頁面底端的段落，其中包含受影響的資源，以及如何套用建議的詳細資訊。

- 在「資料庫」頁面中，選擇資源的建議。

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations
aurora-mysql-cluster-instance-clone2-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	2 Informational
aurora-mysql-cluster-instance-clone2	Available	Writer instance	Aurora MySQL	us-west-2a	db.t3.small	1 Informational
database-1	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	2 Informational
database-1-instance-1	Available	Writer instance	Aurora MySQL	us-west-2c	db.r6g.2xlarge	1 Informational

[建議] 索引標籤會顯示所選資源的建議及其詳細資訊。

DB identifier Status Role Engine Region & AZ Size Recommendations

aurora-mysql-cluster-instance-clone2-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	2 Informational
aurora-mysql-cluster-instance-clone2	Available	Writer instance	Aurora MySQL	us-west-2a	db.t3.small	1 Informational

Connectivity & security Monitoring Logs & events Configuration Zero-ETL integrations Maintenance & backups Tags Recommendations

Recommendations (2) Info

Filter by text or property (example: Severity) Active Last modified Last 1 month

Severity	Detection	Recommendation	Impact	Category	Start time
Informational	1 resource doesn't have Enhanced Monitorir	Turn on Enhanced Monitoring	Reduced operational	Operational ex...	2 months ago
Informational	1 resource has only one DB instance	Add a reader DB instance to your DB cluster	Data availability at ri	Reliability	2 months ago

以下是建議的詳細資訊：

- 嚴重性 — 問題的隱含層級。嚴重性層級為「高」、「中」、「低」和「資訊」。
 - 「檢測」 — 受影響資源的數量和問題的簡短描述。選擇此連結可檢視建議和分析詳細資訊。
 - 建議 — 要套用之建議動作的簡短描述。
 - 影響 — 未套用建議時可能造成的影響的簡短說明。
 - 類別 — 建議的類型。這些類別包括效能效率、安全性、可靠性、成本最佳化、卓越營運和永續性。
 - 狀態 — 建議的目前狀態。可能的狀態為「全部」、「作用中」、「已解除」、「已解決」和「擱置中」
 - 開始時間 — 問題開始的時間。例如，18 小時前。
 - 上次修改時間 — 由於嚴重性發生變更，或您回應建議的時間，系統上次更新建議的時間。例如，10 小時前。
 - 結束時間 — 問題結束的時間。時間不會顯示任何持續的問題。
 - 資源識別碼 — 一或多個資源的名稱。
3. (選擇性) 在欄位中選擇「嚴重性」或「類別」運算子，以篩選建議清單。

Recommendations (6) [Info](#)

The list of recommendations which include best practices for resource configuration, threshold based insights when Per load detection when DevOps Guru for RDS is turned on.

Use: "Severity"

Operators

- Severity** =
Equals
- Severity** !=
Does not equal
- Severity** >=
Greater than or equal
- Severity** <=
Less than or equal
- Severity** <
Less than
- Severity** >
Greater than

	Recommendation
...sql-instance is creating tempora	Review memory para
...d on drg-temp-tables-on-disk-	<ul style="list-style-type: none"> Investigate 1 wait Tune application

將會顯示所選作業的建議。

4. (選擇性) 選擇下列任一建議狀態：

- 使用中 (預設) — 顯示您可以套用的目前建議、排定下一個維護時段或關閉的建議。
- 全部 — 顯示具有目前狀態的所有建議。
- 已解除 — 顯示已解除的建議。
- 已解決 — 顯示已解決的建議。
- 擱置中 — 顯示建議的建議動作正在進行中或排定於下一個維護時段。

Recommendations (13) [Info](#) View details

The list of recommendations which include best practices for resource configuration, threshold based insights when Performance Insights is using the paid tier, and anomalous DB load detection when DevOps Guru for RDS is turned on.

Search: Severity Resolved Last modified: Last 1 month

<input type="checkbox"/>	Severity	Detection	Recommendation	Impact	Category	Status
<input type="checkbox"/>	Informational	2 parameter groups have optimizer statistic	Set the innodb_stats_persistent parameter v	Reduced database pi	Performance e...	Resolved
<input type="checkbox"/>	Informational	1 parameter group has an unsafe setting of	Set the innodb_default_row_format parame	Reduced database pi	Reliability	Resolved
<input type="checkbox"/>	Informational	3 resources are not Multi-AZ instances	Set up Multi-AZ for the impacted DB instanc	Data availability at ri	Reliability	Resolved
<input type="checkbox"/>	Informational	1 resource doesn't have storage autoscaling	Turn on Amazon RDS storage autoscaling wi	Data availability at ri	Reliability	Resolved
<input type="checkbox"/>	Informational	5 resources are not running the latest minor	Upgrade to latest engine version	Reduced database pi	Security	Resolved

5. (選擇性) 在上次修改時間中選擇相對模式或絕對模式來修改時間週期。「建議」頁面會顯示期間內產生的建議。預設時間週期為上個月。在「絕對」模式下，您可以選擇時間週期，或在「開始日期」和「結束日期」欄位中輸入時間。

Last modified

Recommendation

November 2023 December 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4						1	2
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30			24	25	26	27	28	29	30
							31						

Start date Start time End date End time

For date, use YYYY/MM/DD. For time, use 24 hr format.

此時會顯示設定期間的建議。

請注意，您可以將範圍設定為「全部」，查看帳戶中資源的所有建議。

6. (選擇性) 選擇右側的「偏好設定」以自訂要顯示的詳細資訊。您可以選擇頁面大小、換行文字，以及允許或隱藏欄。
7. (選擇性) 選擇建議，然後選擇 [檢視詳細資料]。

RDS > Recommendations

Recommendations (16) [Info](#)

The list of recommendations which include best practices for resource configuration, threshold based insights when Performance Insights is using the paid tier, and anomalous DB load detection when DevOps Guru for RDS is turned on.

Filter by text or property (example: Severity) Active Last modified Last 1 month < 1 > ⚙️

Severity	Detection	Recommendation	Impact	Category	Start time
<input checked="" type="checkbox"/> Medium	The InnoDB history list length increased sigr	<ul style="list-style-type: none"> Identify and address long-running transa Don't shut down the database 	<ul style="list-style-type: none"> Queries may run : Shut-down may t 	Performance e...	3 days ago
<input type="checkbox"/> Medium	High DB Load on dgr-reactive-test-final-ins	<ul style="list-style-type: none"> Investigate 1 wait event Tune application workload 	Reduced database pi	Performance e...	21 days ago

便會顯示建議詳細資訊頁面。標題會提供偵測到問題的資源總數和嚴重性。

如需異常型反應式建議詳細資料頁面上元件的相關資訊，請參閱 Amazon DevOps Guru 使用者指南中的[檢視反應異常](#)。

如需以臨界值為基礎之主動式建議之詳細資料頁面上元件的相關資訊，請參閱[檢視 Performance Insights 主動建議](#)。

其他自動建議會在「建議詳細資訊」頁面上顯示下列元件：

- 建議 — 建議摘要，以及套用建議是否需要停機時間。

RDS > Recommendations > 18 resources don't have Enhanced Monitoring enabled

18 resources don't have Enhanced Monitoring enabled ■ Informational severity [Provide feedback](#) [Dismiss](#) [Apply](#)

Recommendation [Info](#)

Summary
Your database resources don't have Enhanced Monitoring turned on. Enhanced Monitoring provides real-time operating system metrics for monitoring and troubleshooting.

Downtime
Downtime isn't required to apply this recommendation.

- 受影響的資源 — 受影響資源的詳細資訊。

Resources affected (18)					
<input type="text" value="Filter by resource identifier or role"/>					
<input checked="" type="checkbox"/>	<input type="checkbox"/> Resource identifier	Role	Engine	Next maintenance window	Recommended value (seconds)
<input type="checkbox"/>	aurora-mysql-cluster	Regional cluster	Aurora MySQL		
<input checked="" type="checkbox"/>	aurora-mysql-cluster-instance-1	Writer instance	Aurora MySQL	December 14, 2023 01:22 - 01:52 UTC-6	60
<input type="checkbox"/>	aurora-mysql-cluster-instance-clone2-cluster	Regional cluster	Aurora MySQL		
<input checked="" type="checkbox"/>	aurora-mysql-cluster-instance-clone2	Writer instance	Aurora MySQL	December 10, 2023 02:23 - 02:53 UTC-6	60
<input type="checkbox"/>	database-1	Regional cluster	Aurora MySQL		
<input checked="" type="checkbox"/>	database-1-instance-1	Writer instance	Aurora MySQL	December 14, 2023 01:53 - 02:23 UTC-6	60
<input checked="" type="checkbox"/>	delayed-instance	Instance	MySQL Community	December 10, 2023 07:19 - 07:49 UTC-6	60

- 建議詳細資料 — 支援的引擎資訊、套用建議所需的相關成本，以及可進一步瞭解的文件連結。

Recommendation details	
Supported engines MySQL Community, MariaDB, PostgreSQL, Oracle, SQL Server, Aurora MySQL, Aurora PostgreSQL	Learn more Turning Enhanced Monitoring on and off
Associated cost Yes	

CLI

若要檢視資料庫執行個體的 Amazon RDS 建議，請在中使用以下命令 AWS CLI。

```
aws rds describe-db-recommendations
```

RDS API

若要使用 Amazon RDS API 檢視 Amazon RDS 建議，請使用[說明的建議操作](#)。

回應 Amazon RDS 建議

從 RDS 建議清單中，您可以：

- 立即套用以組態為基礎的建議，或延遲到下一個維護時段。
- 關閉一或多個建議。
- 將一或多個已解除的建議移至作用中的建議。

應用 Amazon RDS Amazon 推薦

使用 Amazon RDS 主控台，在詳細資料頁面中選取以組態為基礎的建議或受影響的資源，然後立即套用建議或排定下一個維護時段。資源可能需要重新啟動，變更才會生效。對於一些資料庫參數群組建議，您可能需要重新啟動資源。

以閾值為基礎的主動式或異常型被動式建議不會有套用選項，而且可能需要進行其他審核。

主控台

套用以組態為基礎的建議

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，執行下列任一項作業：
 - 選擇「建議」。
 - 便會顯示「建議」頁面，其中包含所有建議的清單。
 - 選擇資料庫，然後在資料庫頁面中選擇資源的建議。
 - 詳細資料會顯示在所選建議的「建議」索引標籤中。
 - 在「建議」頁面或「資料庫」頁面的「建議」頁籤中，針對作用中建議選擇偵測。
 - 便會顯示建議詳細資訊頁面。
3. 在建議詳細資訊頁面中選擇建議或一或多個受影響的資源，然後執行下列任一項作業：
 - 選擇「套用」，然後選擇「立即套用」，立即套用建議。
 - 選擇 [套用]，然後選擇 [在下一個維護時段中套用]，以在下一個維護時段中排程
 - 選取的建議狀態會更新為擱置，直到下一個維護時段為止。

Recommendations (16) Info

The list of recommendations which include best practices for resource configuration, threshold based insights when Performance Insights is using RDS is turned on.

View details Apply ▲ Dismiss

Apply immediately

Apply in next maintenance window

Filter by text or property (example: Severity) Active Last modified Last 1 month

Severity	Detection	Recommendation	Impact	Category	Status
Medium	The InnoDB history list length increased sig	<ul style="list-style-type: none"> Identify and address long-running transa Don't shut down the database 	<ul style="list-style-type: none"> Queries may run : Shut-down may t 	Performance e...	Active
Medium	High DB Load on dgr-reactive-test-final-ins	<ul style="list-style-type: none"> Investigate 1 wait event Tune application workload 	Reduced database pr	Performance e...	Active
Informational	5 resources have only one DB instance	Add a reader DB instance to your DB cluster	Data availability at ri	Reliability	Active
Informational	4 resources are not Multi-AZ instances	Set up Multi-AZ for the impacted DB instanc	Data availability at ri	Reliability	Active
Informational	3 resources don't have storage autoscaling t	Turn on Amazon RDS storage autoscaling wi	Data availability at ri	Reliability	Active

確認視窗隨即出現。

- 選擇 [確認應用程式] 以套用建議。此視窗會確認資源是否需要自動重新啟動或手動重新啟動，變更才會生效。

下列範例顯示可立即套用建議的確認視窗。

Apply immediately ✕

Recommendation will be immediately applied on:
3 DB Instances ([database-1](#), [database-2](#), [database-3](#))

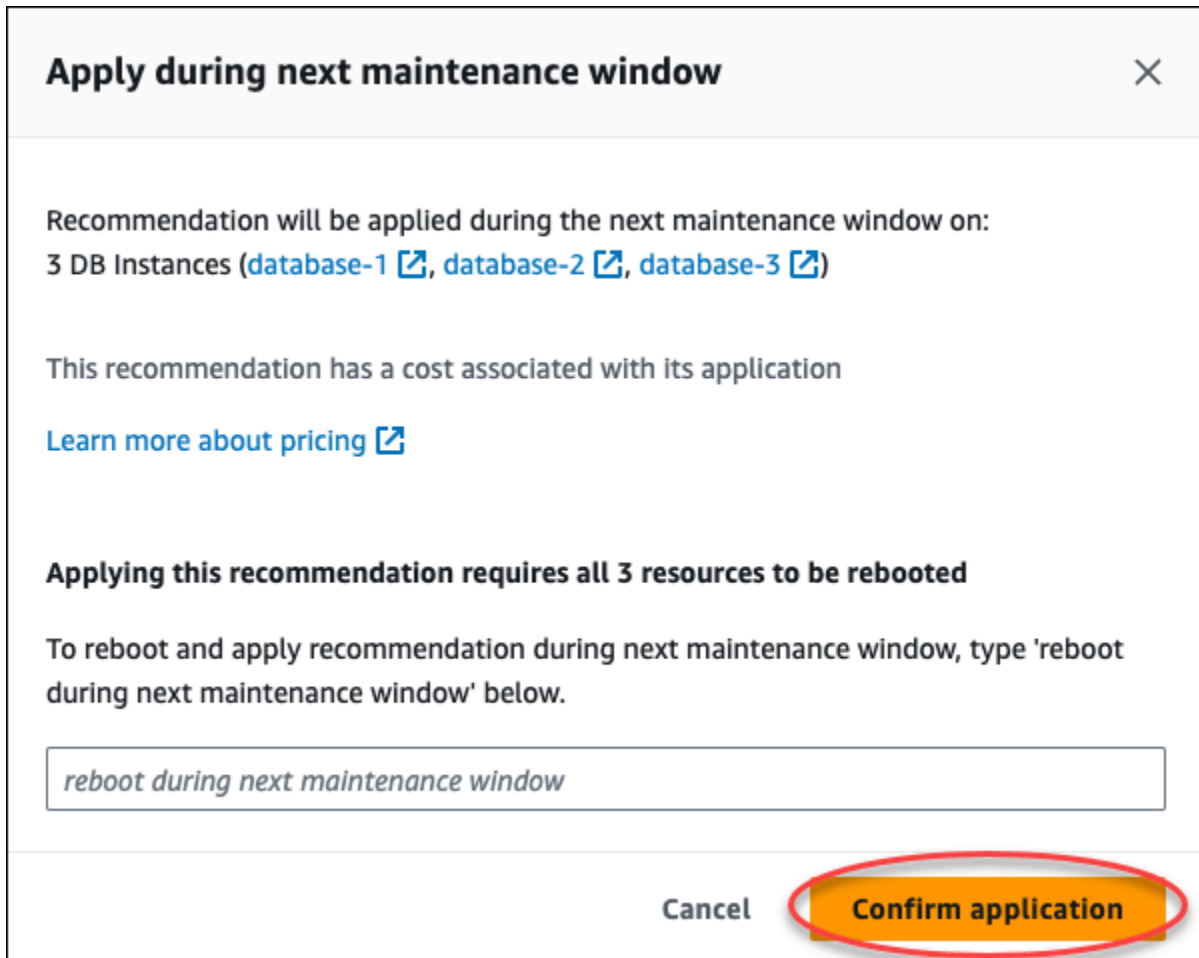
This recommendation has a cost associated with its application
[Learn more about pricing](#)

Applying this recommendation requires all 3 resources to be rebooted

To reboot and apply recommendation immediately, type 'reboot immediately' below.

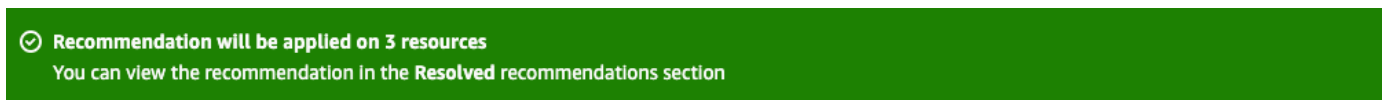
Cancel **Confirm application**

下列範例顯示排程在下一個維護時段中套用建議的確認視窗。



當套用的建議成功或失敗時，橫幅會顯示訊息。

下列範例顯示含有成功訊息的橫幅。



下列範例顯示含失敗訊息的橫幅。



RDS API

使用 Amazon RDS API 套用以組態為基礎的 RDS 建議

1. 使用[描述建議作業](#)。輸出RecommendedActions中的可以有一或多個建議的動作。

- 針 [RecommendedAction](#) 對步驟 1 中的每個建議動作使用物件。輸出包含 `Operation` 和 `Parameters`。

下列範例顯示具有一個建議動作的輸出。

```
"RecommendedActions": [
  {
    "ActionId": "0b19ed15-840f-463c-a200-b10af1b552e3",
    "Title": "Turn on auto backup", // localized
    "Description": "Turn on auto backup for my-mysql-instance-1", //
localized
    "Operation": "ModifyDbInstance",
    "Parameters": [
      {
        "Key": "DbInstanceIdentifier",
        "Value": "my-mysql-instance-1"
      },
      {
        "Key": "BackupRetentionPeriod",
        "Value": "7"
      }
    ],
    "ApplyModes": ["immediately", "next-maintenance-window"],
    "Status": "applied"
  },
  ... // several others
],
```

- `operation` 對於步驟 2 中輸出的每個建議動作使用，然後輸入 `Parameters` 值。
- 步驟 2 中的作業成功之後，請使用「[修改建議建議](#)」作業修改建議狀態。

取消 Amazon RDS Amazon 建議

您可以關閉一或多個建議。

主控台

若要關閉一或多個建議

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，執行下列任一項作業：

- 選擇「建議」。

便會顯示「建議」頁面，其中包含所有建議的清單。

- 選擇資料庫，然後在資料庫頁面中選擇資源的建議。

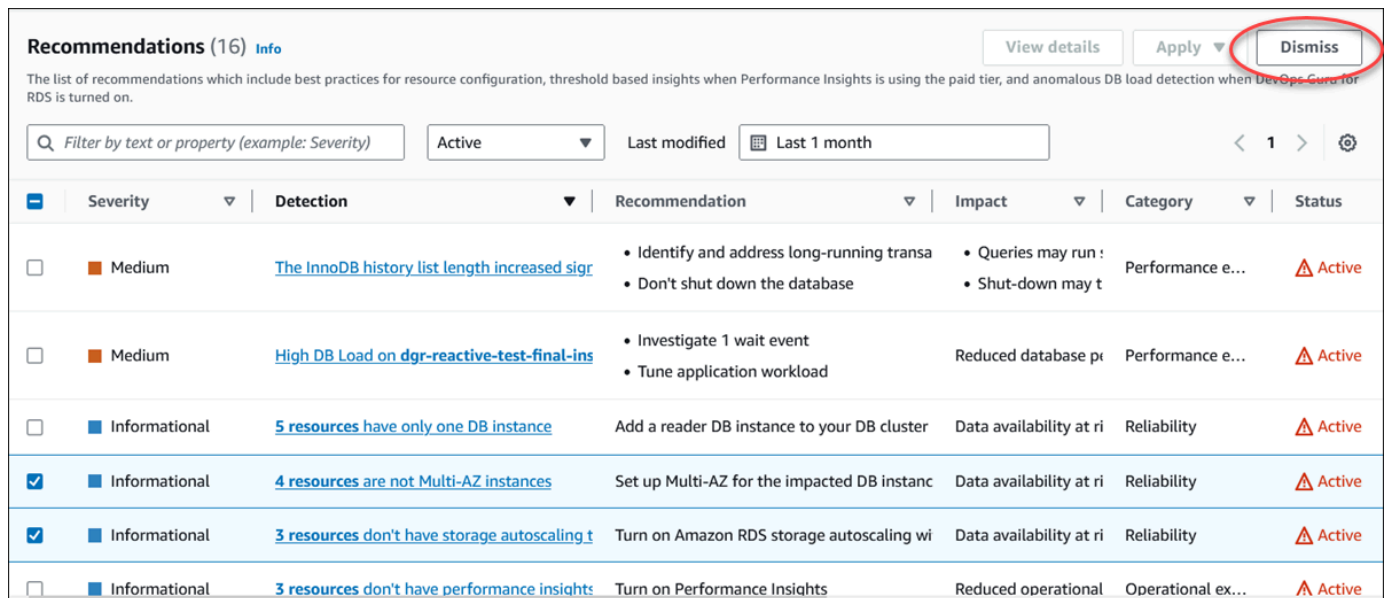
詳細資料會顯示在所選建議的「建議」索引標籤中。

- 在「建議」頁面或「資料庫」頁面的「建議」頁籤中，針對作用中建議選擇偵測。

[建議詳細資料] 頁面會顯示受影響資源的清單。

3. 在建議詳細資料頁面中選擇一或多個建議，或一或多個受影響的資源，然後選擇 [關閉]。

下列範例顯示「建議」頁面，其中包含多個選取要關閉的作用中建議。



The screenshot shows the 'Recommendations (16)' page in the AWS Management Console. At the top right, there are buttons for 'View details', 'Apply', and 'Dismiss' (circled in red). Below these are search and filter options. The main table lists recommendations with columns for Severity, Detection, Recommendation, Impact, Category, and Status. Several rows are selected with checkboxes.

Severity	Detection	Recommendation	Impact	Category	Status
Medium	The InnoDB history list length increased sigr	Identify and address long-running transa Don't shut down the database	Queries may run : Shut-down may t	Performance e...	Active
Medium	High DB Load on dgr-reactive-test-final-ins	Investigate 1 wait event Tune application workload	Reduced database pe	Performance e...	Active
Informational	5 resources have only one DB instance	Add a reader DB instance to your DB cluster	Data availability at ri	Reliability	Active
Informational	4 resources are not Multi-AZ instances	Set up Multi-AZ for the impacted DB instanc	Data availability at ri	Reliability	Active
Informational	3 resources don't have storage autoscaling t	Turn on Amazon RDS storage autoscaling wi	Data availability at ri	Reliability	Active
Informational	3 resources don't have performance insights	Turn on Performance Insights	Reduced operational	Operational ex...	Active

關閉選取的一或多個建議時，橫幅會顯示訊息。

下列範例顯示含有成功訊息的橫幅。

✔ Recommendation is dismissed on 3 resources
You can view the recommendation in the Dismissed recommendations section.

下列範例顯示含失敗訊息的橫幅。

⊗ Failed to dismiss recommendation on database-6
The status of the recommendation with ID 88a73eeb-2e32-4b27-86fb-35ddc7db5abe can't be changed from PENDING to DISMISSED.

CLI

若要使用關閉 RDS 建議 AWS CLI

1. 執行 `aws rds describe-db-recommendations --filters "Name=status,Values=active"` 命令。

輸出會提供 active 狀態中的建議清單。

2. 尋找您要從步驟 1 關閉的建議。 `recommendationId`
3. `>aws rds modify-db-recommendation --status dismissed --recommendationId <ID>` 使用 `recommendationId` 從步驟 2 執行命令以關閉建議。

RDS API

若要使用 Amazon RDS API 關閉 建議，請使用 [修改](#) 建議作業。

將已解除的 Amazon RDS Amazon 建議修改為主動建議

您可以將一或多個已解除的建議移至作用中的建議。

主控台

若要將一或多個已解除的建議移至作用中的建議

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，執行下列任一項作業：
 - 選擇「建議」。

[建議] 頁面會顯示依您帳戶中所有資源嚴重性排序的建議清單。

- 選擇 [資料庫]，然後在 [資料庫] 頁面中選擇資源的建議。

「建議」標籤會顯示所選資源的建議及其詳細資訊。

3. 從清單中選擇一或多個已關閉的建議，然後選擇「移至作用中」。

The screenshot shows the 'Recommendations (6) Info' section in the AWS console. At the top right, there are two buttons: 'View details' and 'Move to active', with the latter circled in red. Below the buttons is a search bar and filters for 'Dismissed' status and 'Last modified' date (Last 1 month). The main table has columns for Severity, Detection, Recommendation, Impact, Category, and Status. Three rows are visible, all with 'Dismissed' status. The first row is for a High severity recommendation about a MySQL instance, the second for a Medium severity recommendation about a MariaDB instance, and the third for another Medium severity recommendation about a MariaDB instance.

將選取的建議從「已關閉」狀態移至作用中狀態時，橫幅會顯示成功或失敗訊息。

下列範例顯示含有成功訊息的橫幅。

✔ Recommendation is moved to active on 3 resources
You can view the recommendation in the Active recommendations section.

下列範例顯示含失敗訊息的橫幅。

✘ Failed to move recommendation to active on database-3
The status of the recommendation with ID 31e23128-6755-4cd8-9ae3-df982656872b can't be changed from PENDING to ACTIVE.

CLI

若要使用將已關閉的 RDS 建議變更為使用中建議 AWS CLI

1. 執行 `aws rds describe-db-recommendations --filters "Name=status,Values=dismissed"` 命令。

輸出會提供 dismissed 狀態中的建議清單。

2. 尋找您要從步驟 1 變更狀態的建議。recommendationId
3. `>aws rds modify-db-recommendation --status active --recommendationId <ID>` 使用 recommendationId 從步驟 2 執行命令，以變更為使用中建議。

RDS API

若要使用 Amazon RDS API 將已關閉的 RDS 建議變更為使用中建議，請使用[修改](#)建議作業。

在 Amazon RDS 主控台中檢視指標

Amazon RDS 會與 Amazon CloudWatch 整合，以在 RDS 主控台中顯示各種 RDS 資料庫執行個體指標。如需有關這些指標的說明，請參閱 [Amazon RDS 的指標參考](#)。

對於資料庫執行個體，會監控以下類別的指標：

- CloudWatch - 顯示您可以於 RDS 主控台中存取的 RDS 的 Amazon CloudWatch 指標。您也可以 CloudWatch 主控台中存取這些指標。每一個指標都包括一個圖形，其中顯示在特定時間範圍內所監控的指標。如需 CloudWatch 指標的清單，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。
- Enhanced monitoring (增強型監控)：在 RDS 資料庫執行個體開啟增強型監控後，顯示作業系統指標的摘要。RDS 可將增強型監控的指標傳送至您的 Amazon CloudWatch Logs 帳戶中。每一個作業系統指標都包括一個圖形，其中顯示在特定時間範圍內所監控的指標。如需概觀，請參閱 [使用增強型監控來監控作業系統指標](#)。如需增強型監控指標的清單，請參閱 [增強型監控中的作業系統指標](#)。
- OS Process list (作業系統程序清單)：顯示您資料庫執行個體中執行的每個程序的詳細資訊。
- Performance Insights：開啟中資料庫執行個體的 Amazon RDS Performance Insights 儀表板。如需 Performance Insights 概觀，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。如需 Performance Insights 指標的清單，請參閱 [Amazon CloudWatch 指標的 Performance Insights](#)。

Amazon RDS 現在會在績效詳情儀表板中提供績效詳情和 CloudWatch 指標的合併檢視。必須為您的資料庫執行個體開啟績效詳情，才能使用此檢視。您可以在監控索引標籤或在導覽窗格的績效詳情中選擇新的監控檢視。若要檢視選擇此檢視的指示，請參閱 [在 Amazon RDS 主控台中檢視組合指標](#)。

如果您想要繼續使用舊式監控檢視，請繼續執行此程序。

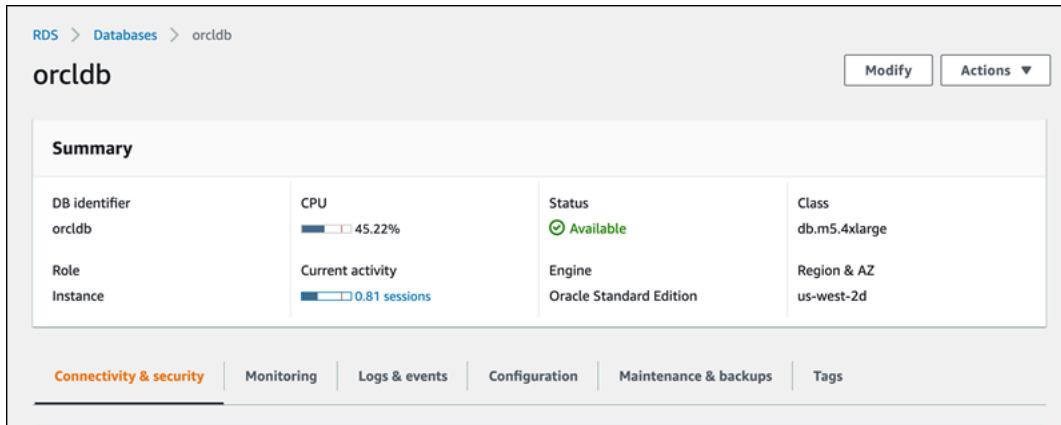
Note

舊版監控檢視將於 2023 年 12 月 15 日停止使用。

若要在舊版監控檢視中檢視資料庫執行個體的指標：

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想要監控的 資料庫執行個體的名稱。

資料庫頁面隨即出現。下列範例顯示名為 orclb 的 Oracle 資料庫。



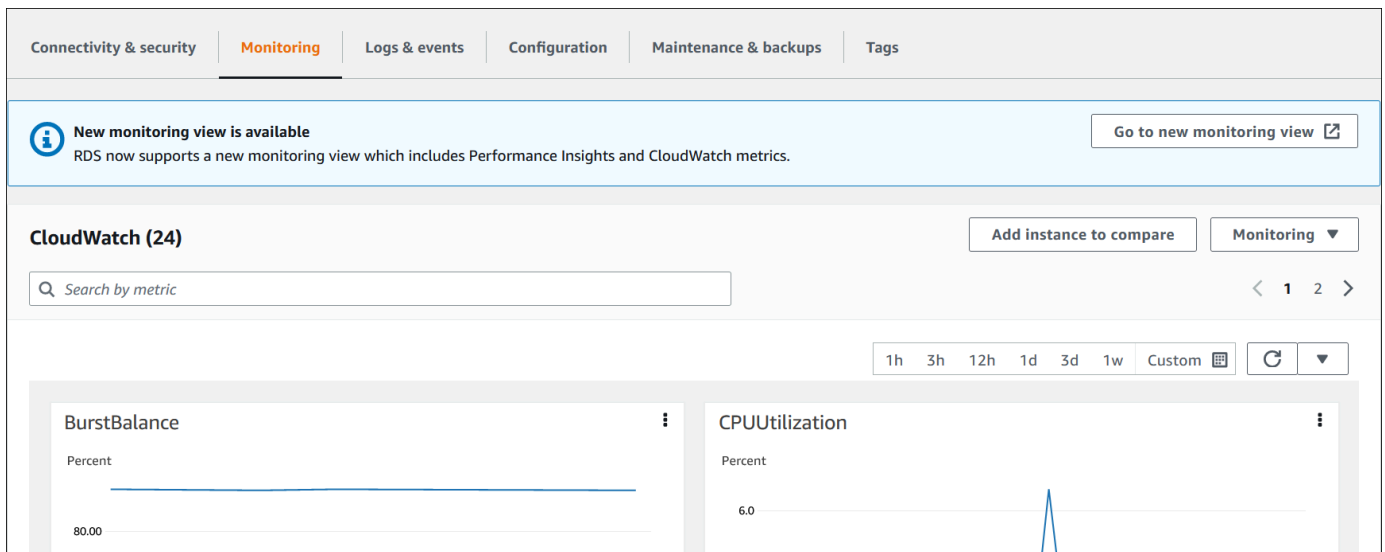
The screenshot shows the Amazon RDS console interface for a database instance named 'orclb'. The breadcrumb navigation is 'RDS > Databases > orclb'. The instance name 'orclb' is displayed at the top left, with 'Modify' and 'Actions' buttons to its right. Below this is a 'Summary' section with a table of key metrics:

Metric	Value
DB identifier	orclb
CPU	45.22%
Status	Available
Class	db.m5.4xlarge
Role	
Current activity	0.81 sessions
Engine	Oracle Standard Edition
Region & AZ	us-west-2d

At the bottom of the summary section, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Monitoring' tab is currently selected.

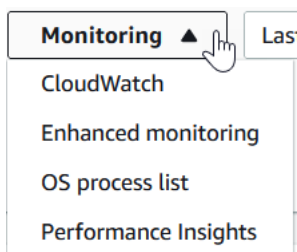
4. 向下捲動並選擇 Monitoring (監控)。

系統會顯示監控區段。依據預設，系統會顯示 CloudWatch 指標。如需這些指標的說明，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。



The screenshot shows the 'Monitoring' page in the Amazon RDS console. The breadcrumb navigation is 'Connectivity & security > Monitoring > Logs & events > Configuration > Maintenance & backups > Tags'. A notification banner at the top states: 'New monitoring view is available. RDS now supports a new monitoring view which includes Performance Insights and CloudWatch metrics. Go to new monitoring view'. Below the notification is a 'CloudWatch (24)' section with a search bar 'Search by metric' and a 'Monitoring' dropdown menu. The main area displays two metrics: 'BurstBalance' (Percent) and 'CPUUtilization' (Percent). The 'BurstBalance' chart shows a constant value of 80.00. The 'CPUUtilization' chart shows a peak of 6.0. The time range is set to '1h'.

5. 選擇 Monitoring (監控) 以查看指標類別。



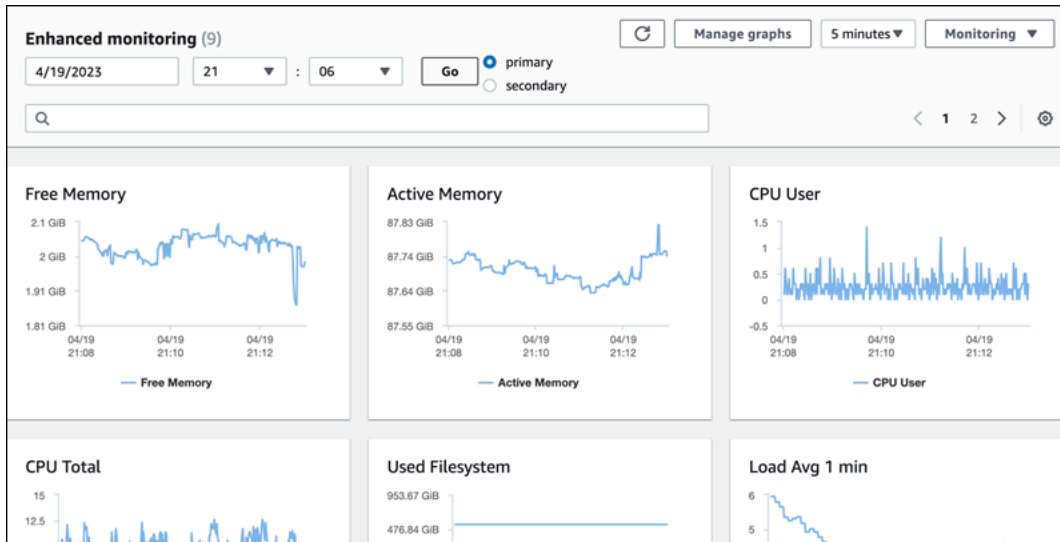
The screenshot shows the 'Monitoring' dropdown menu in the Amazon RDS console. The menu is open, showing the following options: 'Monitoring', 'CloudWatch', 'Enhanced monitoring', 'OS process list', and 'Performance Insights'. A mouse cursor is pointing at the 'Monitoring' option.

6. 選擇您要查看的指標分類。

以下範例會顯示增強型監控指標。如需這些指標的說明，請參閱 [增強型監控中的作業系統指標](#)。

Note

MariaDB 資料庫執行個體目前並不支援檢視多可用區域備用複本的作業系統指標。

**Tip**

若要選擇圖形所代表之指標的時間範圍，您可以使用時間範圍清單。您可以選擇任何圖形來啟動更詳細的檢視。您也可以將指標特定篩選條件套用至資料。

在 Amazon RDS 主控台中檢視組合指標

Amazon RDS 現在會在績效詳情儀表中為您的資料庫執行個體提供績效詳情和 CloudWatch 指標的合併檢視。您可以使用預先設定的儀表板或建立自訂儀表板。預先設定的儀表板提供最常用的指標，協助診斷資料庫引擎的效能問題。或者，您可以為資料庫引擎建立自訂儀表板，其中包含符合您分析需求的指標。然後，將此儀表板用於 AWS 帳戶中該資料庫引擎類型的所有資料庫執行個體。

您可以在監控索引標籤或在導覽窗格的績效詳情中選擇新的監控檢視。當您導覽至「績效詳情」頁面時，您會看到可在新監控檢視與舊版檢視之間進行選擇的選項。您選擇的選項會儲存為預設檢視。

您必須為資料庫執行個體開啟績效詳情，才能在「績效詳情」儀表中檢視組合指標。如需開啟績效詳情的詳細資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)。

Note

我們建議您選擇新的監控檢視。您可以繼續使用舊版監控檢視，直到它於 2023 年 12 月 15 日停止使用為止。

在監控索引標籤中選擇新的監控檢視

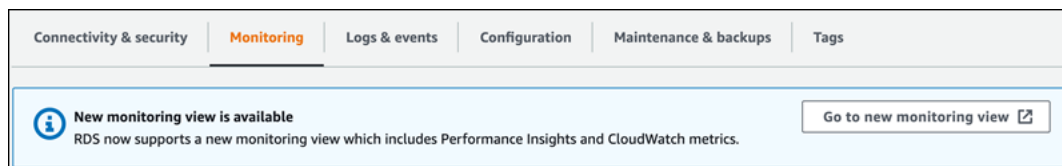
若要在監控索引標籤中選擇新的監控檢視：

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在左側的導覽窗格中，選擇資料庫。
3. 選擇您想要監控的資料庫執行個體。

資料庫頁面隨即出現。

4. 向下捲動並選擇監控。

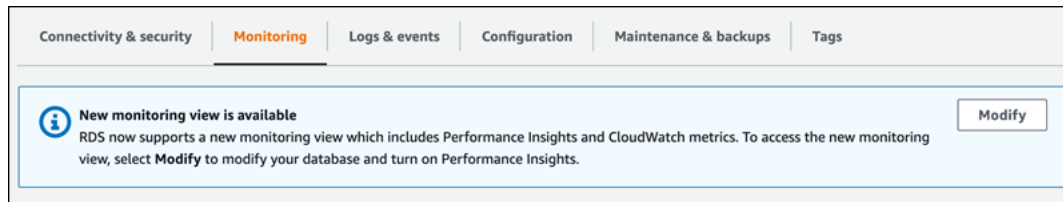
橫幅隨即出現，其中包含了選擇新監控檢視的選項。以下範例顯示了選擇新監控檢視的橫幅。



5. 選擇移至新的監控檢視以開啟績效詳情儀表板，其中包含資料庫執行個體的績效詳情和 CloudWatch 指標。

6. (選用) 如果資料庫執行個體的績效詳情已關閉，則會出現橫幅，其中包含可修改資料庫叢集並開啟績效詳情的選項。

下列範例顯示橫幅，以在監控索引標籤中修改資料庫叢集。



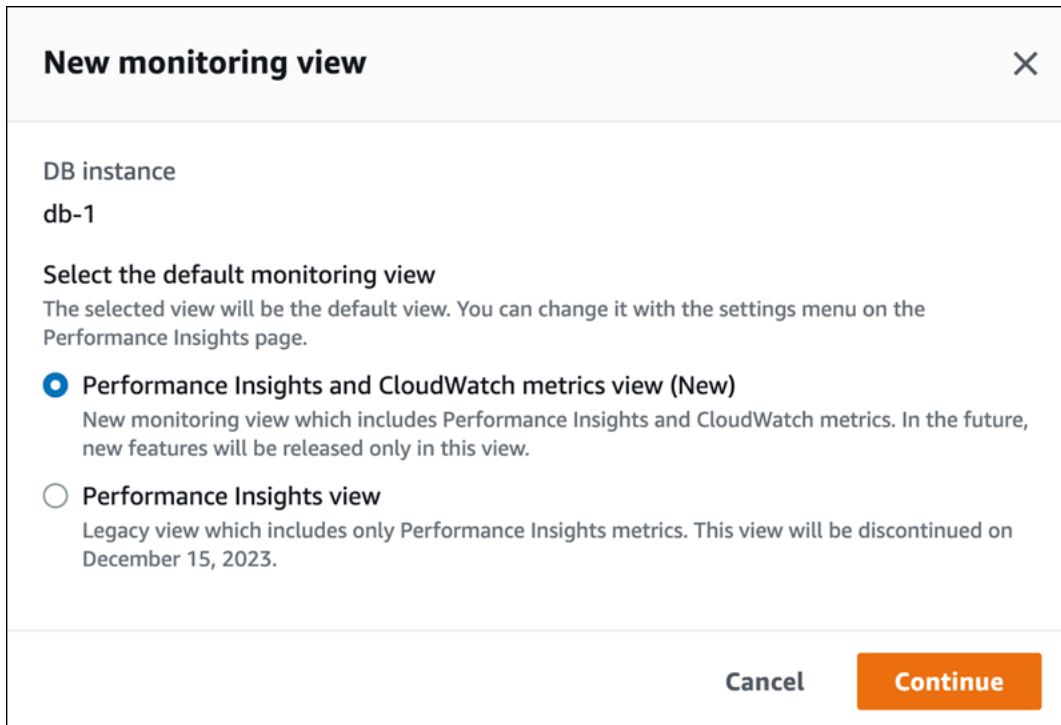
選擇修改以修改您的資料庫叢集，並開啟績效詳情。如需開啟績效詳情的詳細資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)

在導覽窗格中選擇具有績效詳情的新監控檢視

若要在導覽窗格中選擇具有績效詳情的新監控檢視：

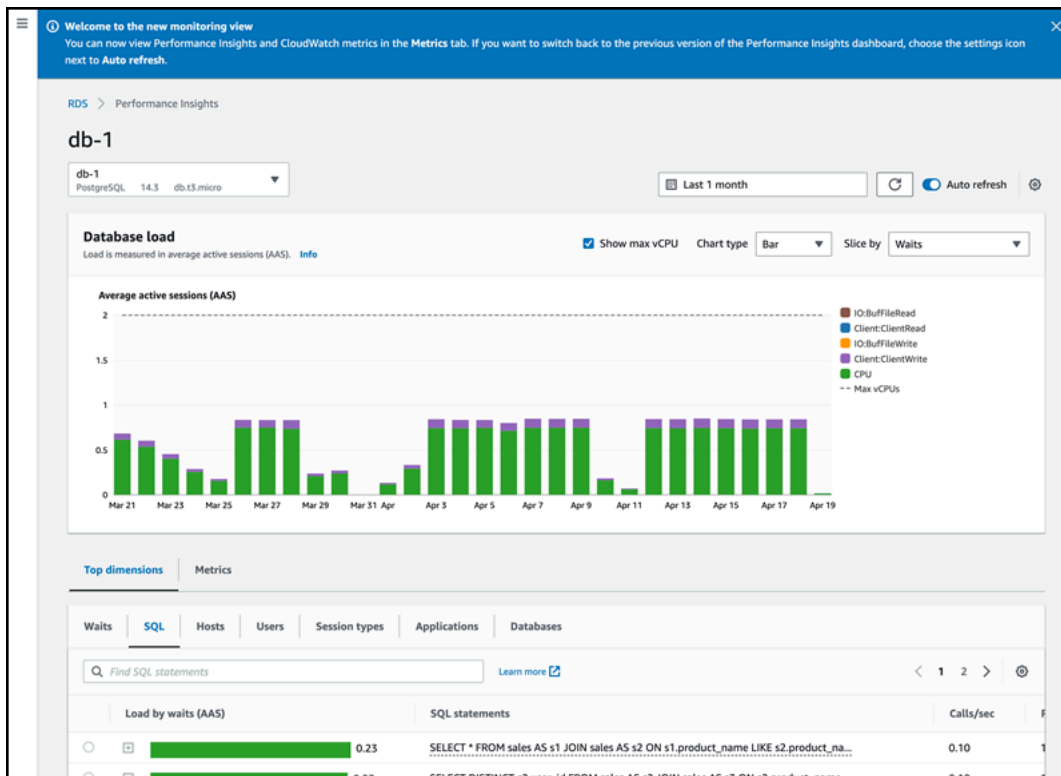
1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體以開啟具有監控檢視選項的視窗。

下列範例顯示了具有監控檢視選項的視窗。



4. 選擇績效詳情和 CloudWatch 指標檢視 (新增) 選項，然後選擇繼續。

您現在可以檢視績效詳情儀表板，其中同時顯示資料庫執行個體的績效詳情和 CloudWatch 指標。下列範例會顯示儀表板中的效能詳情和 CloudWatch 指標。



在導覽窗格中選擇具有績效詳情的舊版檢視

您可以選擇舊版監控檢視，僅檢視資料庫執行個體的績效詳情指標。

Note

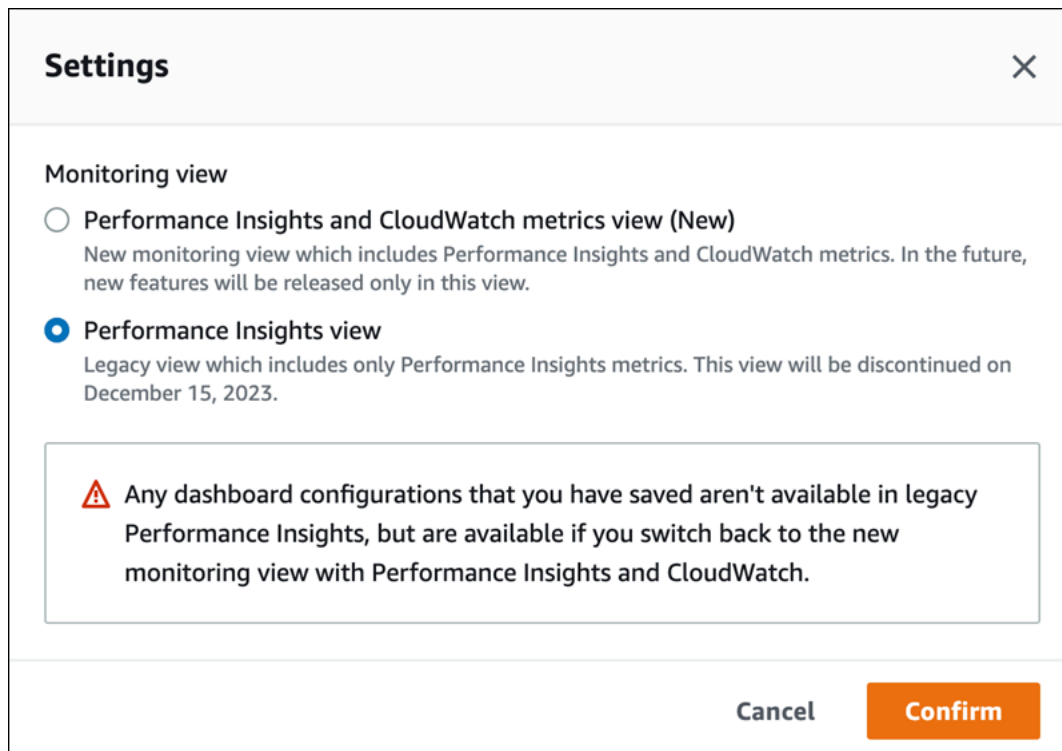
此檢視將於 2023 年 12 月 15 日停止使用。

若要在導覽窗格中選擇具有績效詳情的舊版監控檢視：

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。
4. 選擇績效詳情儀表板上的設定圖示。

現在，您可以看到設定視窗，其中顯示了選擇舊版績效詳情檢視的選項。

下列範例顯示了具有舊版監控檢視選項的視窗。



The screenshot shows a 'Settings' dialog box with a close button (X) in the top right corner. Under the heading 'Monitoring view', there are two radio button options:

- Performance Insights and CloudWatch metrics view (New)
New monitoring view which includes Performance Insights and CloudWatch metrics. In the future, new features will be released only in this view.
- Performance Insights view
Legacy view which includes only Performance Insights metrics. This view will be discontinued on December 15, 2023.

Below the options is a warning box with a red triangle icon and the text: 'Any dashboard configurations that you have saved aren't available in legacy Performance Insights, but are available if you switch back to the new monitoring view with Performance Insights and CloudWatch.'

At the bottom right of the dialog are two buttons: 'Cancel' and 'Confirm'.

5. 選取績效詳情檢視選項，並選擇繼續。

警告訊息即會出現。您儲存的任何儀表板組態都無法在此檢視中使用。

6. 選擇確認以繼續前往舊版績效詳情檢視。

您現在可以檢視績效詳情儀表板，其中僅顯示資料庫執行個體的績效詳情指標。

在導覽窗格中建立具有績效詳情的自訂儀表板

在新的監控檢視中，您可以建立自訂儀表板，其中具有您符合分析需求所需的指標。

您可以選取資料庫執行個體的績效詳情和 CloudWatch 指標，以建立自訂儀表板。您可以將此自訂儀表板用於 AWS 帳戶中相同資料庫引擎類型的其他資料庫執行個體。

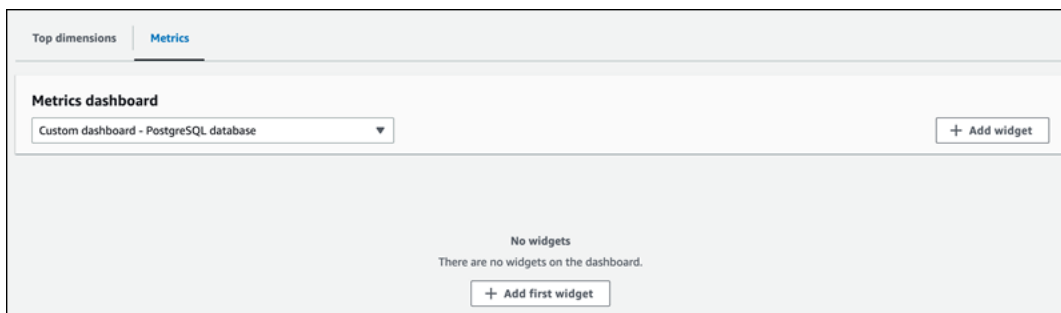
Note

自訂的儀表板最多支援 50 個指標。

使用 Widget 設定功能表來編輯或刪除儀表板，以及移動 Widget 視窗或調整其大小。

若要在導覽窗格中建立具有績效詳情的自訂儀表板：

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。
4. 向下捲動至視窗中的指標索引標籤。
5. 從下拉式清單中選取自訂儀表板。下列範例說明如何建立自訂儀表板。



6. 選擇新增 Widget 以開啟新增 Widget 視窗。您可以在視窗中開啟並檢視可用的作業系統 (OS) 指標、資料庫指標和 CloudWatch 指標。

下列範例顯示了具有指標的新增 Widget 視窗。

Add widget ✕

All metrics (152)
You can add up to 50 metrics to your custom dashboard.

<input type="checkbox"/>	Metric	Unit
<input checked="" type="checkbox"/>	OS metrics	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> General	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> CPU Utilization	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Disk IO	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> File Sys	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Load Average Minute	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Memory	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Network	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Swap	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Tasks	-
<input checked="" type="checkbox"/>	Database metrics	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Cache	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Checkpoint	-
<input type="checkbox"/>	<input checked="" type="checkbox"/> Concurrency	-

50 more metrics can be added to your dashboard. Cancel Add widget

7. 選取要在儀表板中檢視的指標，然後選擇新增 Widget。您可以使用搜尋欄位來尋找特定指標。

選取的指標會出現在您的儀表板上。

8. (選用) 如果您想要修改或刪除儀表板，請選擇 Widget 右上角的設定圖示，然後在功能表中選取下列其中一個動作。
 - 編輯 - 修改視窗中的指標清單。在您選取儀表板的指標之後，選擇更新 Widget。
 - 刪除 - 刪除 Widget。在確認視窗中選擇刪除。

在導覽窗格中選擇具有績效詳情的預先設定儀表板

您可以使用預先設定的儀表板檢視最常用的指標。此儀表板可協助診斷資料庫引擎的效能問題，並將平均復原時間從數小時縮短為數分鐘。

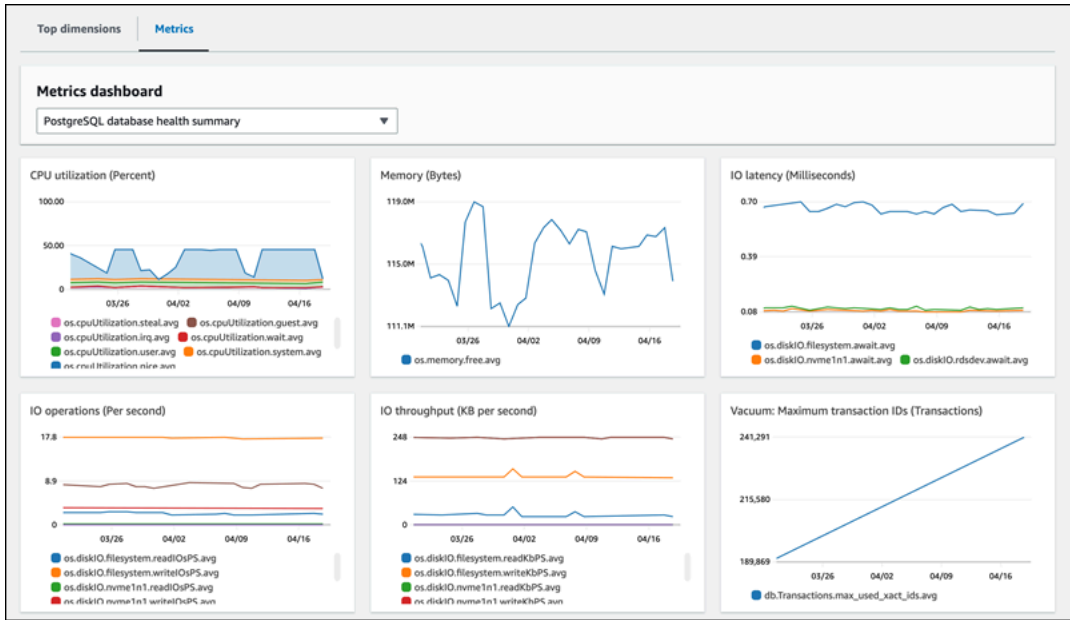
Note

無法編輯此儀表板。

若要在導覽窗格中選擇具有績效詳情的預先設定儀表板：

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。
4. 向下捲動至視窗中的指標索引標籤
5. 從下拉式清單中選取預先設定的儀表板。

您可以在儀表板中檢視資料庫執行個體的指標。下列範例顯示預先設定的指標儀表板。



使用 Amazon CloudWatch 監控 Amazon RDS 指標

Amazon CloudWatch 是指標儲存體。儲存體會收集並處理來自 Amazon RDS 的原始資料，進而將這些資料轉換為便於讀取且幾近即時的指標。如需傳送至 CloudWatch 的 Amazon RDS 指標完整清單，請參閱 [Amazon RDS 的指標參考](#)。

主題

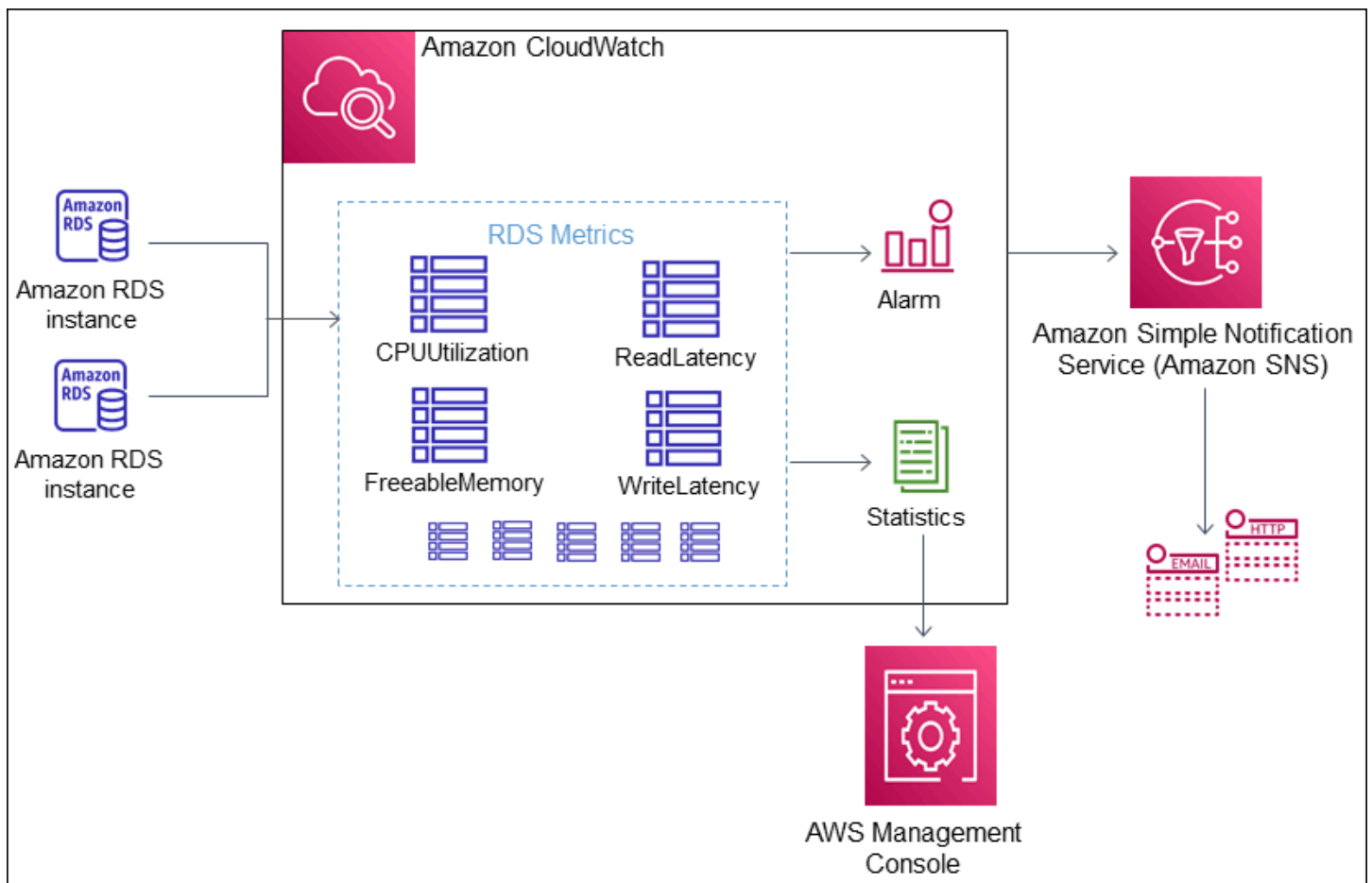
- [Amazon RDS 和 Amazon CloudWatch 的概觀](#)
- [在 CloudWatch 主控台中檢視資料庫執行個體指標 AWS CLI](#)
- [將 Performance Insights 指標匯出至 CloudWatch](#)
- [建立 CloudWatch 警示來監控 Amazon RDS](#)
- [教學課程：為多可用區域資料庫叢集複本延遲建立 Amazon CloudWatch 警示](#)

Amazon RDS 和 Amazon CloudWatch 的概觀

根據預設，Amazon RDS 每隔 1 分鐘會自動將指標資料傳送至 CloudWatch。例如，CPUUtilization 指標記錄一段時間內資料庫執行個體的 CPU 利用率百分比。含少於 60 秒期間 (1 分鐘) 的資料點可供使用 15 天。這表示您可以存取歷史資訊，並查看 Web 應用程式或服務的執行效能。

您現在可以將 Performance Insights 指標儀表板從 Amazon RDS 匯出到 Amazon CloudWatch。您可以將預先設定或自訂的指標儀表板匯出為新的儀表板，或將其新增至現有的 CloudWatch 儀表板。匯出的儀表板可在 CloudWatch 主控台中檢視。如需如何將 Performance Insights 指標儀表板匯出至 CloudWatch 的詳細資訊，請參閱[將 Performance Insights 指標匯出至 CloudWatch](#)。

如下圖所示，您可為 CloudWatch 指標設定警示。例如，您可能會建立一個警示，當執行個體的 CPU 利用率超過 70% 時發出訊號。您可設定 Amazon Simple Notification Service 以便在閾值超過時向您傳送電子郵件。



Amazon RDS 會向 Amazon CloudWatch 發佈下列類型的指標：

- RDS 資料庫執行個體的指標

如需關於這些指標的表格，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。

- 績效詳情指標

如需關於這些指標的表格，請參閱 [Amazon CloudWatch 指標的 Performance Insights](#) 和 [Performance Insights 計數器指標](#)。

- 增強型監控指標 (發佈至 Amazon CloudWatch Logs)

如需關於這些指標的表格，請參閱 [增強型監控中的作業系統指標](#)。

- 您 AWS 帳戶中 Amazon RDS Service Quotas 的用量指標

如需關於這些指標的表格，請參閱 [Amazon RDS Amazon 的亞馬遜 CloudWatch 使用指標](#)。如需 Amazon RDS 配額的詳細資訊，請參閱 [Amazon RDS 的配額和條件限制](#)。

如需 CloudWatch 的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 [什麼是 Amazon CloudWatch ?](#)。如需有關 CloudWatch 指標保留的詳細資訊，請參閱 [指標保留](#)。

在 CloudWatch 主控台中檢視資料庫執行個體指標 AWS CLI

接下來，您可以找到有關如何使用檢視資料庫執行個體指標的詳細資訊 CloudWatch。如需使用 CloudWatch Logs 即時監控資料庫執行個體作業系統指標的相關資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

當您使用 Amazon RDS Amazon 資源時，Amazon RDS Amazon CloudWatch 每分鐘都會向 Amazon 發送指標和維度。

您現在可以將 Performance Insights 指標儀表板從 Amazon RDS 匯出到 Amazon，CloudWatch 並在 CloudWatch 主控台中檢視這些指標。如需有關如何將 Performance Insights 指標儀表板匯出至的詳細資訊 CloudWatch，請參閱 [將 Performance Insights 指標匯出至 CloudWatch](#)。

使用下列程序在 CloudWatch 主控台和 CLI 中檢視 Amazon RDS 的指標。

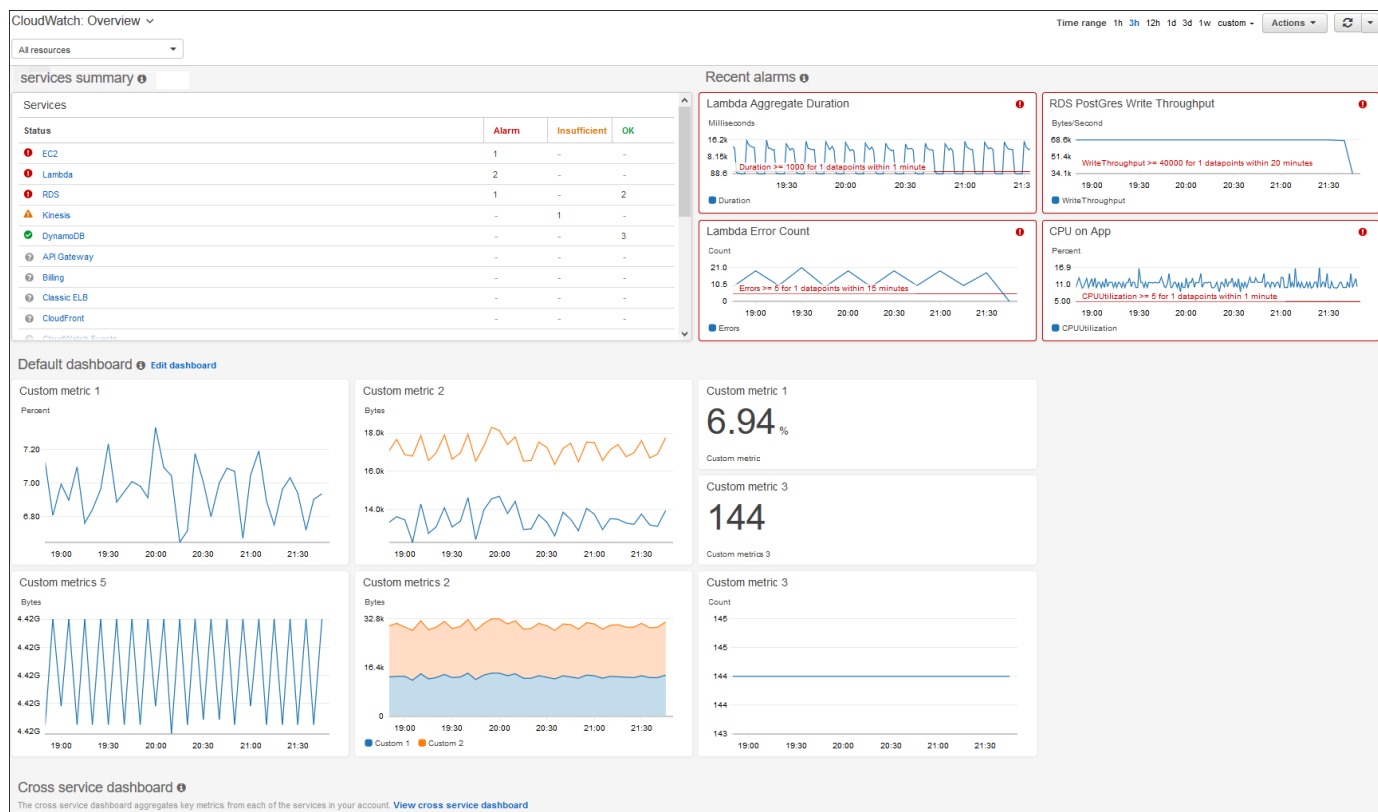
主控台

若要使用 Amazon CloudWatch 主控台檢視指標

指標會先依服務命名空間分組，再依各命名空間內不同的維度組合分類。

1. [請在以下位置開啟 CloudWatch 主控台](https://console.aws.amazon.com/cloudwatch/)。 <https://console.aws.amazon.com/cloudwatch/>

這時系統顯示 CloudWatch 概述首頁。



2. 如有需要，請變更 AWS 區域。從導覽列中，選擇您的 AWS 資源所在的 AWS 區域。如需詳細資訊，請參閱 [區域與端點](#)。
3. 在導覽窗格中，選擇 Metrics (指標)，然後選擇 All metrics (所有指標)。

The screenshot shows the Amazon CloudWatch Metrics console interface. At the top, there are tabs for 'Browse', 'Query', 'Graphed metrics', 'Options', and 'Source'. On the right, there are buttons for 'Add math' and 'Add query'. Below the tabs, the page title is 'Metrics (1301)' with an 'Info' link. There are buttons for 'Graph with SQL' and 'Graph search'. A dropdown menu shows 'N. Virginia' and a search bar with the placeholder text 'Search for any metric, dimension or resource id'. The main content is a grid of metric namespaces:

EBS	9	EC2	17	Events	5
Lambda	26	Logs	35	RDS	1152
S3	8	SSM Run Command	3	Usage	46

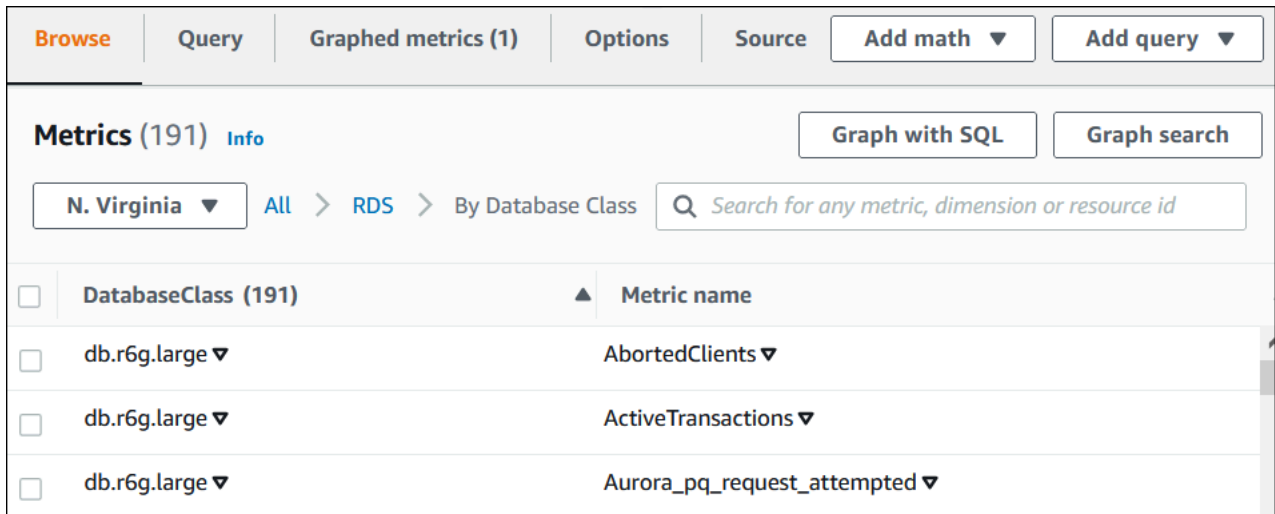
4. 向下捲動並選擇 RDS 指標命名空間。

此頁面顯示 Amazon RDS 維度。如需這些維度的說明，請參閱 [Amazon RDS 的 Amazon CloudWatch 維度](#)。

The screenshot shows the Amazon CloudWatch Metrics console interface with the RDS namespace selected. The page title is 'Metrics (1152)' with an 'Info' link. There are buttons for 'Graph with SQL' and 'Graph search'. A dropdown menu shows 'N. Virginia' and a breadcrumb path 'All > RDS'. A search bar with the placeholder text 'Search for any metric, dimension or resource id' is present. The main content is a grid of metric dimensions:

DBClusterIdentifier, Role	153	DbClusterIdentifier, EngineName	6	DBClusterIdentifier	133
Per-Database Metrics	332	By Database Class	191	By Database Engine	223
Across All Databases	114				

5. 選擇指標維度，例如 By Database Class (依照資料庫類別)。



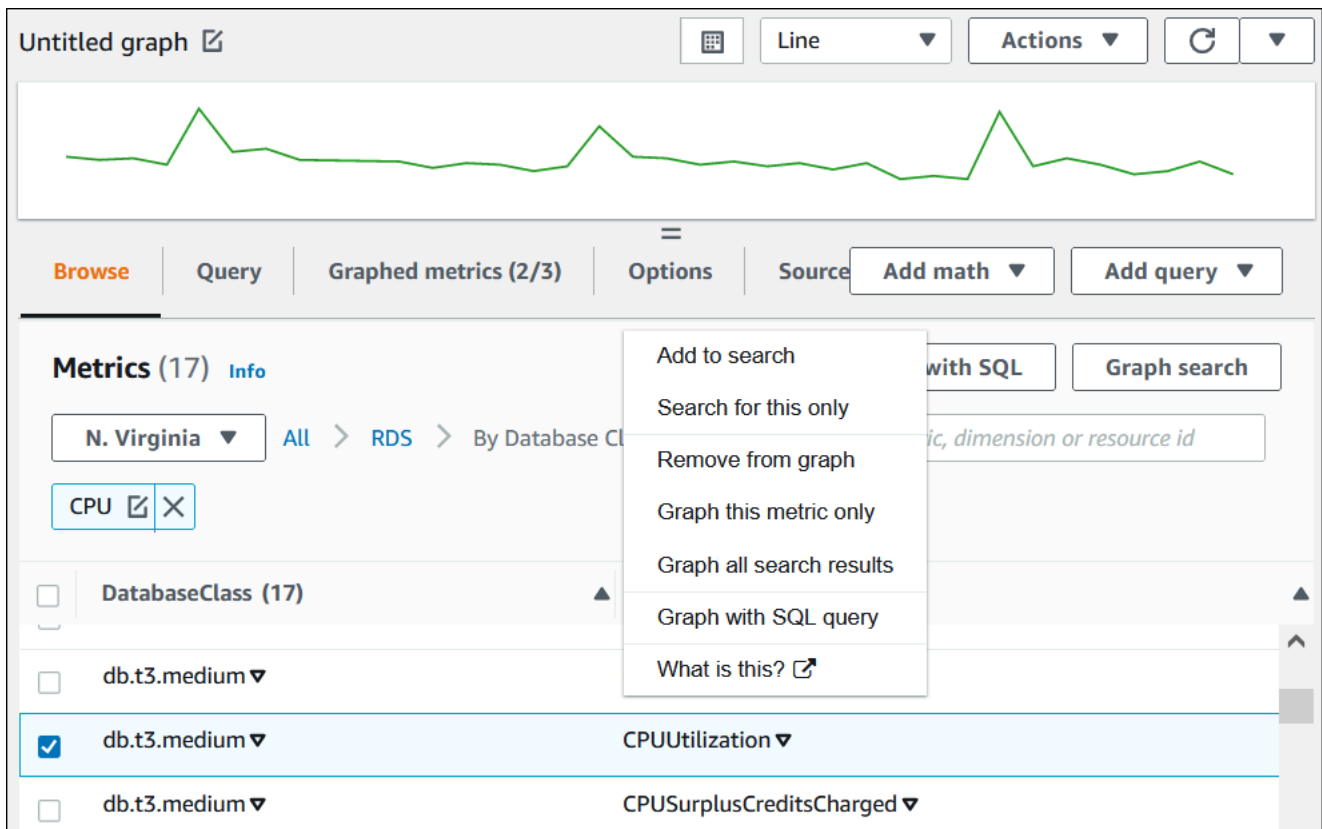
The screenshot shows the Amazon CloudWatch Metrics console interface. At the top, there are tabs for 'Browse', 'Query', 'Graphed metrics (1)', 'Options', and 'Source'. Below these are buttons for 'Add math' and 'Add query'. The main content area is titled 'Metrics (191) Info' and includes a 'Graph with SQL' and 'Graph search' button. A breadcrumb trail shows 'N. Virginia' > 'All' > 'RDS' > 'By Database Class'. A search bar is present with the placeholder text 'Search for any metric, dimension or resource id'. Below this is a table with two columns: 'DatabaseClass (191)' and 'Metric name'. The table lists three metrics for the 'db.r6g.large' database class: 'AbortedClients', 'ActiveTransactions', and 'Aurora_pq_request_attempted'. Each row has a checkbox on the left.

DatabaseClass (191)	Metric name
<input type="checkbox"/> db.r6g.large ▼	AbortedClients ▼
<input type="checkbox"/> db.r6g.large ▼	ActiveTransactions ▼
<input type="checkbox"/> db.r6g.large ▼	Aurora_pq_request_attempted ▼

6. 執行下列其中一個動作：

- 若要排序指標，請使用直欄標題。
- 若要將指標圖形化，請選取指標旁的核取方塊。
- 若要依資源篩選，請選擇資源 ID，然後選擇 Add to search (新增至搜尋)。
- 若要依指標篩選，請選擇指標名稱，然後選擇 Add to search (新增至搜尋)。

下面的範例會篩選 db.t3.medium 類別並繪製 CPUUtilization 指標。



AWS CLI

若要使用取得測量結果資訊AWS CLI，請使用 CloudWatch 指令 [list-metrics](#)。以下範例會列出 AWS/RDS 命名空間中的所有指標。

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

若要取得測量結果資料，請使用指令 [get-metric-data](#)。

下列範例會取得特定 24 小時期間內執行 my-instance 個體的 CPUUtilization 統計資料，其粒度為 5 分鐘。

使用下列內容建立 JSON 檔案 CPU_metric.json。

```
{
  "StartTime" : "2023-12-25T00:00:00Z",
  "EndTime" : "2023-12-26T00:00:00Z",
  "MetricDataQueries" : [{
    "Id" : "cpu",
    "MetricStat" : {
```

```
"Metric" : {
  "Namespace" : "AWS/RDS",
  "MetricName" : "CPUUtilization",
  "Dimensions" : [{ "Name" : "DBInstanceIdentifier" , "Value" : my-instance}]
},
"Period" : 360,
"Stat" : "Minimum"
}
}]
}
```

Example

對於LinuxmacOS、或Unix：

```
aws cloudwatch get-metric-data \
  --cli-input-json file://CPU_metric.json
```

在Windows中：

```
aws cloudwatch get-metric-data ^
  --cli-input-json file://CPU_metric.json
```

範例輸出如下所示：

```
{
  "MetricDataResults": [
    {
      "Id": "cpu",
      "Label": "CPUUtilization",
      "Timestamps": [
        "2023-12-15T23:48:00+00:00",
        "2023-12-15T23:42:00+00:00",
        "2023-12-15T23:30:00+00:00",
        "2023-12-15T23:24:00+00:00",
        ...
      ],
      "Values": [
        13.299778337027714,
        13.677507543049558,
        14.24976250395827,
        13.02521708695145,
        ...
      ]
    }
  ]
}
```

```
    ],
    "StatusCode": "Complete"
  }
],
"Messages": []
}
```

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的取得指[標統計](#)資料。

將 Performance Insights 指標匯出至 CloudWatch

Performance Insights 可讓您將資料庫執行個體的預先設定或自訂指標儀表板匯出到 Amazon CloudWatch。您可以將指標儀表板匯出為新儀表板，或將其新增至現有 CloudWatch 儀表板。當您選擇將儀表板新增至現有 CloudWatch 儀表板時，您可以建立標題標籤，以便量度顯示在 CloudWatch 儀表板的單獨區段中。

您可以在 CloudWatch 主控台中檢視匯出的指標儀表板。如果您在匯出後將新指標新增至 Performance Insights 指標儀表板，則必須再次匯出此儀表板，才能在 CloudWatch 主控台中檢視新的指標。

您也可以在此「Performance Insights」儀表板中選取指標 Widget，並在 CloudWatch 主控台中檢視指標資料。

如需有關在主控台中檢視測量結果的詳細資訊，請參閱在 [CloudWatch 主控台中檢視資料庫執行個體指標 AWS CLI](#)。

將 Performance Insights 指標匯出為新儀表板至 CloudWatch

從「績效見解」儀表板選擇預先設定或自訂指標 Performance Insights 板，並將其匯出為新儀表板至 CloudWatch。您可以在 CloudWatch 主控台中檢視匯出的儀表板。

若要將「Performance Insights」量度儀表板匯出為新儀表板，CloudWatch

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動並選擇指標。

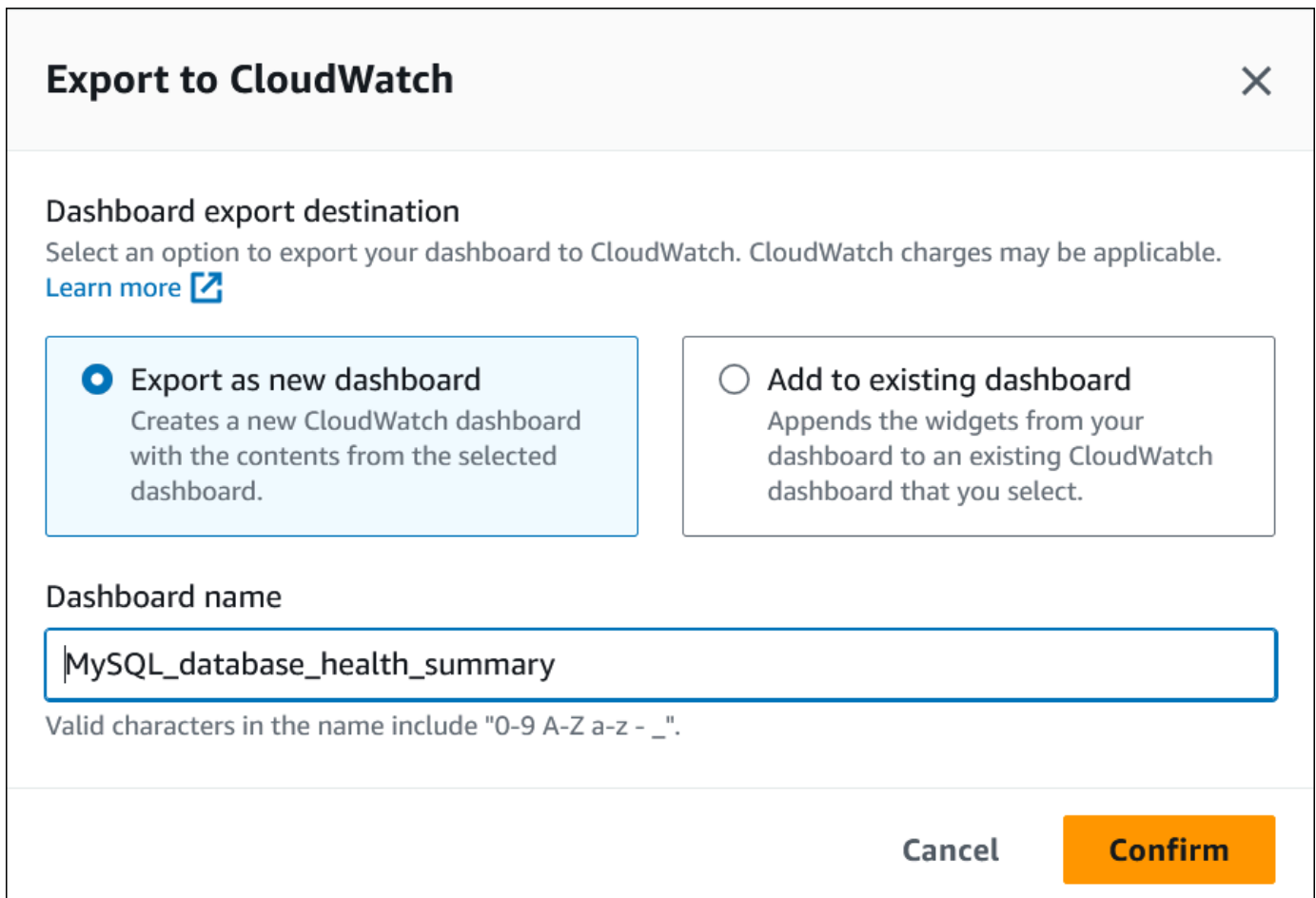
根據預設，具有 Performance Insights 指標的預先設定儀表板即會出現。

5. 選擇預先設定或自訂儀表板，然後選擇 [匯出至 CloudWatch]。

「匯出至」 CloudWatch 視窗隨即出現。

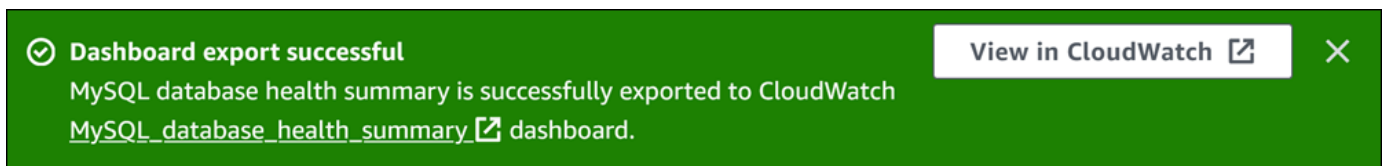


6. 選擇匯出為新儀表板。



7. 在儀表板名稱欄位中輸入新儀表板的名稱，然後選擇確認。

在成功匯出儀表板之後，橫幅會顯示一則訊息。



8. 選擇橫幅中的連結或「檢視」，以 CloudWatch 在 CloudWatch 主控台中檢視指標儀表板。

將 Performance Insights 指標新增至現有 CloudWatch 儀表板

將預先設定或自訂指標儀表板新增至現有 CloudWatch 儀表板。您可以將標籤新增至量度控制面板，以顯示在儀 CloudWatch 表板的個別區段中。

若要將量度匯出至現有的 CloudWatch 儀表板

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動並選擇指標。


根據預設，具有 Performance Insights 指標的預先設定儀表板即會出現。

5. 選擇預先設定或自訂儀表板，然後選擇 [匯出至 CloudWatch]。

「匯出至」 CloudWatch 視窗隨即出現。

6. 選擇新增至現有儀表板。

Export to CloudWatch ✕

Dashboard export destination
Select an option to export your dashboard to CloudWatch. CloudWatch charges may be applicable.
[Learn more](#) 

Export as new dashboard
Creates a new CloudWatch dashboard with the contents from the selected dashboard.

Add to existing dashboard
Appends the widgets from your dashboard to an existing CloudWatch dashboard that you select.

CloudWatch dashboard destination
MySQL_database_health_summary ▼

CloudWatch dashboard section label - *optional*
Additional graphs will appear in this section.

PI export - MySQL database health summary

Cancel **Confirm**

7. 指定儀表板目的地和標籤，然後選擇確認。

- CloudWatch 儀表板目標-選擇現有 CloudWatch 儀表板。
- CloudWatch 儀表板區段標籤-選用-輸入要顯示在儀 CloudWatch 表板中此區段中的 Performance Insights 量度的名稱。

在成功匯出儀表板之後，橫幅會顯示一則訊息。

8. 選擇橫幅中的連結或「檢視」，以 CloudWatch 在 CloudWatch 主控台中檢視指標儀表板。

檢視「Performance Insights」量度小器具 CloudWatch

在 Amazon RDS Performance Insights 儀表板中選取 Performance Insights 指標 Widget，然後在 CloudWatch 主控台中檢視指標資料。

匯出量度 Widget 並在 CloudWatch 主控台中檢視量度資料

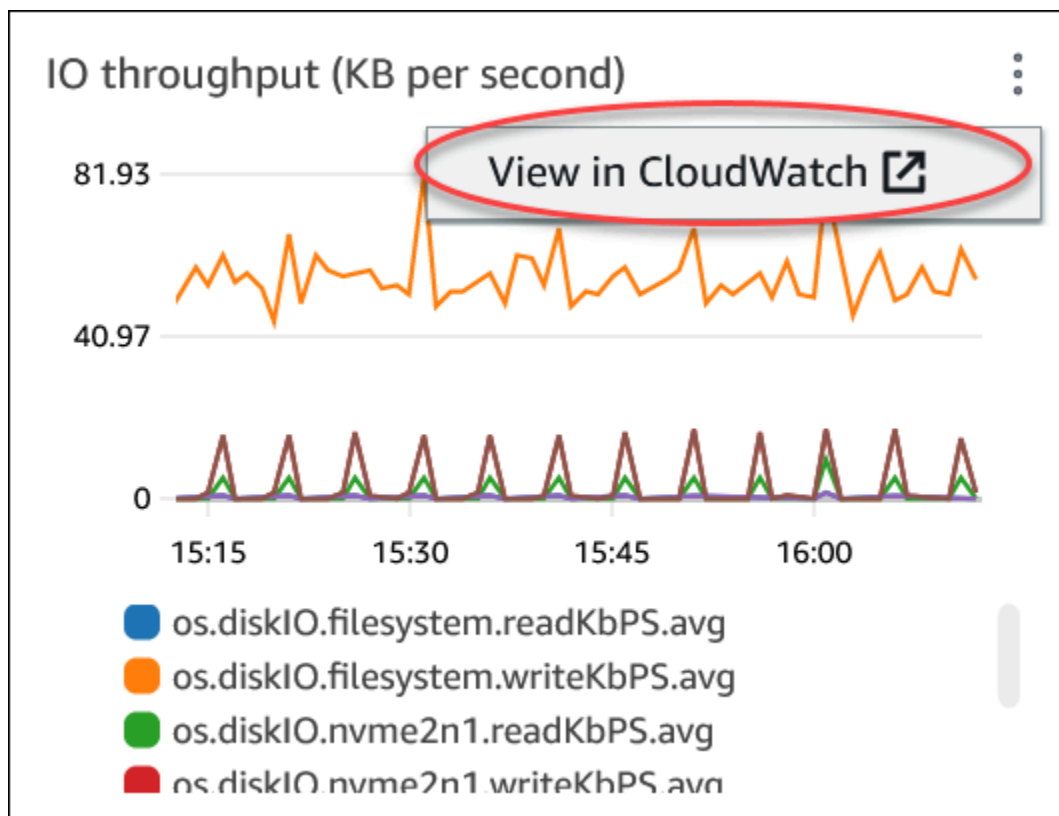
1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動至指標。

依預設，具有 Performance Insights 指標的預先設定儀表板即會出現。

5. 選擇量度 Widget，然後在選單 CloudWatch 中選擇「檢視」。



測量結果資料會顯示在主 CloudWatch 控台中。

建立 CloudWatch 警示來監控 Amazon RDS

您可以建立 CloudWatch 警報，在警示變更狀態時傳送 Amazon SNS 訊息。警示會在您指定的期間監看單一指標。警示會根據在數段期間內與指定閾值相關的指標值，來執行一個或多個動作。此動作是傳送到 Amazon SNS 主題或 Amazon EC2 Auto Scaling 政策的通知。

警示僅會針對持續狀態變更呼叫動作。CloudWatch 警示不會僅因為它們處於特定狀態而叫用動作。狀態必須發生變更並維持一段指定的時間。

您可以使用 CloudWatch 主控台中的 DB_PERF_INSIGHTS 指標數學函數查詢 Amazon RDS for Performance Insights 計數器指標。DB_PERF_INSIGHTS 函數也包含次分鐘間隔的 DBLoad 指標。您可以根據這些指標設定 CloudWatch 警示。

如需如何建立警示的詳細資訊，請參閱[從 AWS 資料庫建立 Performance Insights 計數器指標的警示](#)。

使用 AWS CLI 設定警示

- 呼叫 [put-metric-alarm](#)。如需更多詳細資訊，請參閱 [AWS CLI 命令參考](#)。

使用 CloudWatch API 設定警示

- 呼叫 [PutMetricAlarm](#)。如需更多詳細資訊，請參閱 [Amazon CloudWatch API 參考](#)。

如需有關設定 Amazon SNS 主題和建立警示的詳細資訊，請參閱[使用 Amazon CloudWatch 警示](#)。

教學課程：為多可用區域資料庫叢集複本延遲建立 Amazon CloudWatch 警示

您可以建立 Amazon CloudWatch 警示，在多可用區域資料庫叢集的複本延遲超過閾值時傳送 Amazon SNS 訊息。警示會在您指定的期間監看 ReplicaLag 指標。此動作是傳送到 Amazon SNS 主題或 Amazon EC2 Auto Scaling 政策的通知。

為多可用區域資料庫叢集複本延遲設定 CloudWatch 警示

1. 登入 AWS Management Console 並開啟位於 <https://console.aws.amazon.com/cloudwatch/> 的 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Alarms (警示)、All alarms (所有警示)。
3. 選擇建立警示。

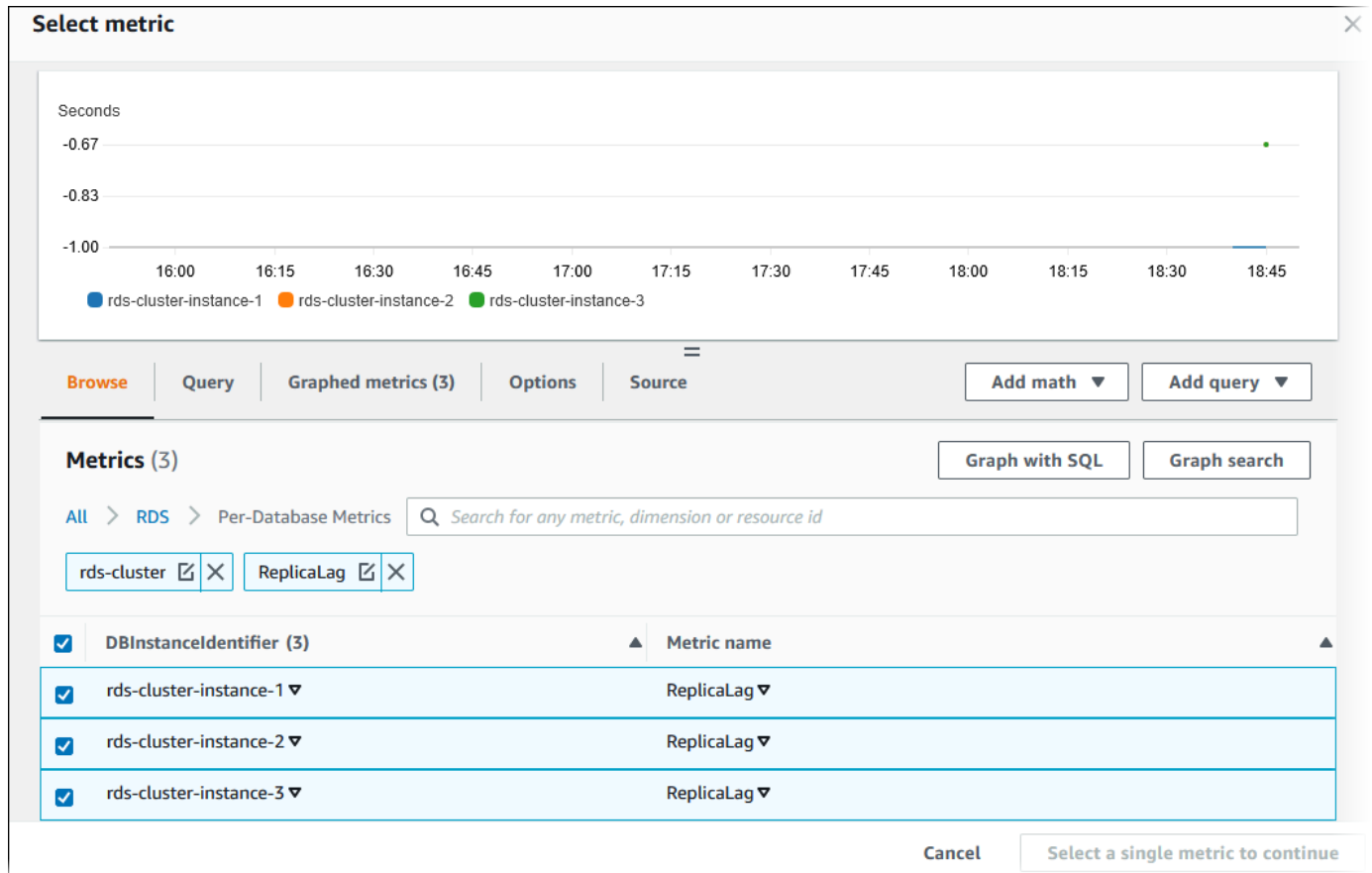
- 在指定指標和條件頁面上，選擇選取指標。
- 在搜尋方塊中輸入多可用區域資料庫叢集的名稱，然後按 Enter 鍵。

下圖顯示 Select metric (選取指標) 頁面，其中輸入了名為 `rds-cluster` 的多可用區域資料庫叢集。

The screenshot shows the 'Select metric' interface in AWS CloudWatch. The main graph area is empty, displaying the message 'Your CloudWatch graph is empty. Select some metrics to appear here.' The search bar at the bottom contains the text 'rds-cluster' and shows a search result for 'RDS > Per-Database Metrics 78'.

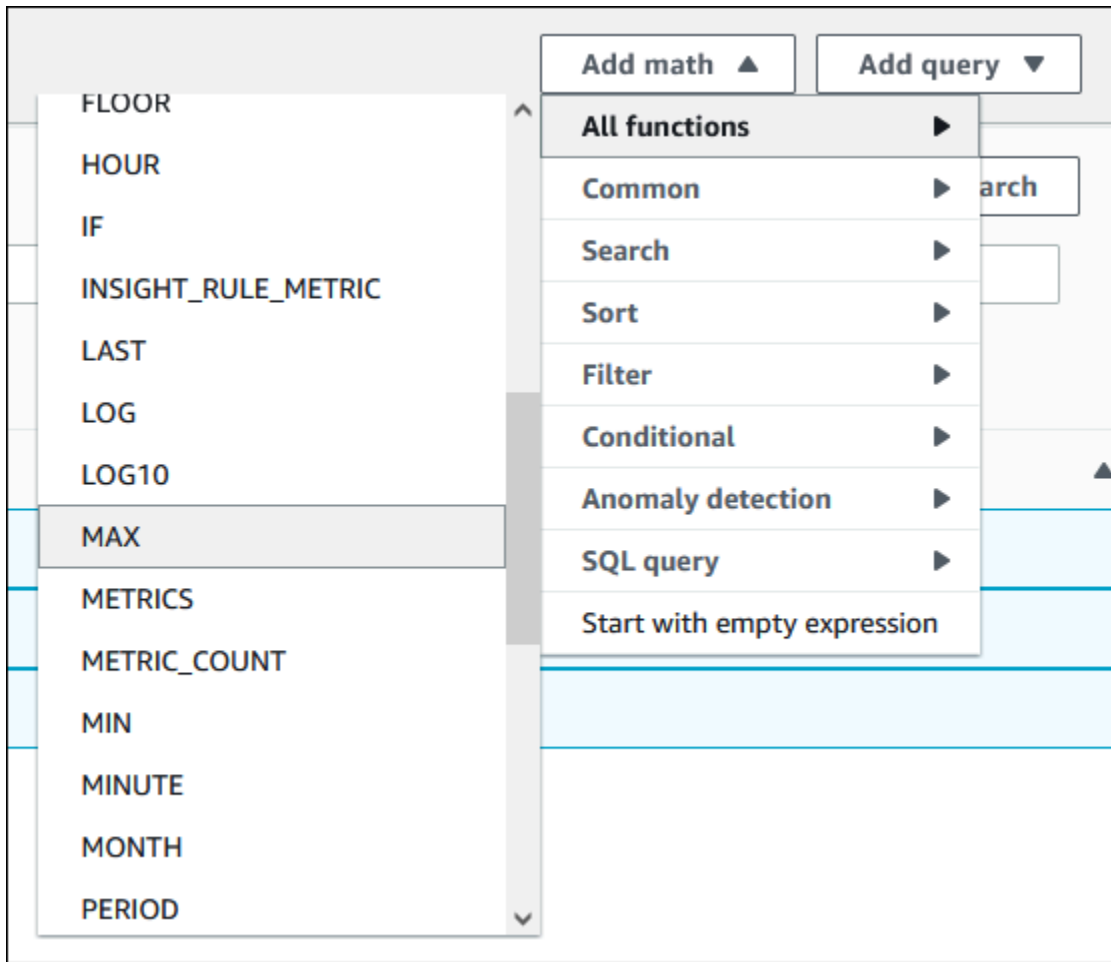
- 選擇 RDS、Per-Database Metrics (每個資料庫指標)。
- 在搜尋方塊中輸入 **ReplicaLag** 並按 Enter 鍵，然後選取資料庫叢集中的各個資料庫執行個體。

下圖顯示 Select metric (選取指標) 頁面，其中針對 ReplicaLag 指標選取了資料庫執行個體。



此警示考慮多可用區域資料庫叢集中所有三個資料庫執行個體的複本延遲。當任何資料庫執行個體超過閾值時，警示就會回應。其使用的數學表達式會傳回三個指標的最大值。首先按指標名稱排序，然後選擇所有三個 ReplicaLag 指標。

8. 從 Add math (新增數學) 中，選擇 All functions (所有函數)、MAX (最大值)。



9. 選擇 Graphed metrics (圖表化指標) 索引標籤，然後將 Expression1 的詳細資料編輯為 **MAX([m1,m2,m3])**。
10. 對於所有三個 ReplicaLag 指標，請將 Period (期間) 變更為 1 minute (1 分鐘)。
11. 清除所有指標中的選擇，Expression1 除外。

Select metric (選取指標) 頁面看起來應該會如下圖所示。

The screenshot shows the 'Select metric' dialog box. At the top, there's a title 'Select metric' and a close button. Below that, there's a section for the graph with a title 'Untitled graph' and a time range selector (1h, 3h, 12h, 1d, 3d, 1w, Custom) and a line chart. The chart shows a single data series 'Expression1' at a value of 0. Below the graph, there are tabs for 'Browse', 'Query', 'Graphed metrics (1/4)', 'Options', and 'Source'. There are also buttons for 'Add math' and 'Add query'. Below the tabs, there's a section for 'Add dynamic label' and 'Info', with a 'Statistic' dropdown set to 'Average' and a 'Period' dropdown set to '1 Minute'. Below this is a table of metrics:

<input type="checkbox"/>	Id	Label	Details	Statistic	Period	Y Axis	Actions
<input checked="" type="checkbox"/>	e1	Expression1	MAX([m1,m2,m3])				
<input type="checkbox"/>	m1	rds-cluster-ins...	RDS • ReplicaLag • DBInstanceIde...	Average	1 Minute		
<input type="checkbox"/>	m2	rds-cluster-ins...	RDS • ReplicaLag • DBInstanceIde...	Average	1 Minute		
<input type="checkbox"/>	m3	rds-cluster-ins...	RDS • ReplicaLag • DBInstanceIde...	Average	1 Minute		

At the bottom right, there are 'Cancel' and 'Select metric' buttons.

12. 選擇 Select metric (選取指標)。

13. 在 Specify metric and conditions (指定指標和條件) 頁面上，請將標籤變更為有意義的名稱，例如 **ClusterReplicaLag**，然後在 Define the threshold value (定義閾值) 中輸入秒數。在此教學課程中，輸入 **1200** 秒 (20 分鐘)。您可以根據工作負載要求調整此值。

Specify metric and conditions (指定指標和條件) 頁面看起來應該會如下圖所示。

Specify metric and conditions

Metric

Edit

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 1 minute.

No unit

1,000

500

0

17:00 18:00 19:00

ClusterReplicaLag

Label
ClusterReplicaLag

Math expression
MAX([m1,m2,m3])

Metrics
m1 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...
m2 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...
m3 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...

Period
1 minute

Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Whenever ClusterReplicaLag is...
Define the alarm condition.

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

than...
Define the threshold value.

1200

Must be a number

► **Additional configuration**

Cancel **Next**

14. 選擇 Next (下一頁) 後，Configure actions (設定動作) 頁面隨即出現。

15. 保持選取 In alarm (處於警示狀態)，選擇 Create new topic (建立新主題)，然後輸入主題名稱和有效的電子郵件地址。

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action. Remove

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Select an SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN

Create a new topic...
The topic name must be unique.

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification...
Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

user1@example.com, user2@example.com

16. 選擇 Create topic (建立主題)，然後選擇 Next (下一步)。
17. 在 Add name and description (新增名稱和描述) 頁面上，輸入 Alarm name (警示名稱) 和 Alarm description (警示描述)，然後選擇 Next (下一步)。

Add name and description

Name and description

Alarm name

Alarm description - *optional*

Up to 1024 characters (59/1024)

Cancel Previous **Next**

18. 在 Preview and create (預覽和建立) 頁面預覽您即將建立的警示，然後選擇 Create Alarm (建立警示)。

在 Amazon RDS 上使用績效詳情監控資料庫負載

績效詳情會擴展現有 Amazon RDS 監控功能，以說明並協助您分析資料庫效能。利用績效詳情儀表板，您可將 Amazon RDS 資料庫執行個體負載上的資料庫負載視覺化，並依等候、SQL 陳述式、主機或使用者篩選負載。如需有關搭配 Amazon DocumentDB 使用績效詳情的資訊，請參閱《[Amazon DocumentDB 開發人員指南](#)》。

主題

- [Amazon RDS 上的績效詳情概觀](#)
- [開啟和關閉 Amazon RDS 的 Performance Insights](#)
- [在 Amazon RDS for MariaDB 或 MySQL 上開啟績效詳情的效能結構描述](#)
- [設定績效詳情的存取政策](#)
- [使用績效詳情儀表板來分析指標](#)
- [檢視 Performance Insights 主動建議](#)
- [使用適用於 Amazon RDS 的 Performance Insights API 擷取指標](#)
- [使用 AWS CloudTrail 記錄績效詳情呼叫](#)

Amazon RDS 上的績效詳情概觀

根據預設，RDS 會在主控台中為所有 Amazon RDS 引擎啟用 Performance Insights 建立精靈。如果您在一個資料庫執行個體上有多個資料庫，績效詳情會彙整效能資料。

您可於下方影片中找到 Amazon RDS 的績效詳情概觀。

[使用績效詳情來分析 Amazon Aurora PostgreSQL 效能](#)

Important

下列主題說明搭配非 Aurora 資料庫引擎來使用 Amazon RDS 績效詳情。如需搭配使用 Amazon RDS Performance Insights 與 Amazon Aurora 的詳細資訊，請參閱《Amazon Aurora 使用者指南》中的[使用 Amazon RDS Performance Insights](#)。

主題

- [資料庫負載](#)
- [最大 CPU](#)

- [Performance Insights 支援 Amazon RDS 資料庫引擎和執行個體類別](#)
- [Performance Insights 的定價和資料保留](#)

資料庫負載

資料庫負載 (DB 載入) 會測量資料庫中工作階段活動的層級。DBLoad 是「Performance Insights 見」中的關鍵指標，而 Performance Insights 每秒都會收集資料庫負載。

主題

- [作用中的工作階段](#)
- [平均作用中工作階段](#)
- [平均作用中執行數](#)
- [維度](#)

作用中的工作階段

資料庫工作階段代表應用程式與關聯式資料庫的對話。作用中工作階段是已提交工作給資料庫引擎且正在等待回應的連線。

工作階段處於作用中是指工作階段正在 CPU 上執行，或等待資源變成可用以繼續執行。例如，作用中工作階段可能等待分頁 (或區塊) 讀入記憶體中，然後從分頁讀取資料時耗用 CPU。

平均作用中工作階段

平均作用中工作階段 (AAS) 是 DBLoad 績效詳情的單位。它會測量資料庫上同時處於作用中狀態的工作階段數目。

每一秒，績效詳情都會取樣同時執行查詢的工作階段數目。針對每個作用中的工作階段，績效詳情會收集下列資料：

- SQL 陳述式
- 工作階段狀態 (正在 CPU 上執行或等待中)
- 主機
- 執行 SQL 的使用者

績效詳情會計算 AAS，方法是將工作階段總數除以特定時段的樣本數。例如，下表顯示執行查詢的 5 個連續範例，間隔至少為 1 秒。

樣本	執行查詢的工作階段數目	AAS	算式
1	2	2	工作階段總數 2 / 1 個樣本
2	0	1	工作階段總數 2 / 2 個樣本
3	4	2	工作階段總數 6 / 3 個樣本
4	0	1.5	工作階段總數 6 / 4 個樣本
5	4	2	工作階段總數 10 / 5 個樣本

在上述範例中，時間間隔的資料庫負載為 2 AAS。此測量表示在採集 5 個樣本的時間間隔內，於任何特定時間平均有 2 個工作階段處於作用中。

平均作用中執行數

每秒平均作用中執行數 (AAE) 與 AAS 相關。若要計算 AAE，績效詳情會將查詢的總執行時間除以時間間隔。下表顯示上表中同一個查詢的 AAE 計算。

經過時間 (秒)	總執行時間 (秒)	AAE	算式
60	120	2	120 執行秒 / 60 秒經過
120	120	1	120 執行秒 / 120 秒經過
180	380	2.11	380 執行秒 / 180 秒經過
240	380	1.58	380 執行秒 / 240 秒經過
300	600	2	600 執行秒 / 300 秒經過

在大多數情況下，查詢的 AAS 和 AAE 大致相同。不過，由於計算的輸入是不同的資料來源，所以計算通常會略有不同。

維度

db.load 指標與其他時間序列指標不同，因為您可以將它分為名為維度的子元件。您可以將維度視為 DBLoad 指標不同特性的「配量依據」類別。

診斷效能問題時，下列維度通常最實用：

主題

- [等待事件](#)
- [最高 SQL](#)
- [計畫](#)

如需 Amazon RDS 引擎的維度完整清單，請參閱 [資料庫負載依維度配量](#)。

等待事件

等待事件會導致 SQL 陳述式等待特定事件發生後，才能繼續執行。等待事件指出工作在何處受阻，是資料庫負載的重要維度或類別。

每個使用中的工作階段都在 CPU 上執行或等待中。例如，工作階段在記憶體中搜尋緩衝區、執行計算或執行程序程式碼時會耗用 CPU。當工作階段不耗用 CPU 時，可能是在等待記憶體緩衝區變成可用、要讀取的資料檔或要寫入的記錄檔。工作階段等待資源越久，在 CPU 上執行的時間就越短。

調校資料庫時，您通常會嘗試查明工作階段正在等待的資源。例如，兩個或三個等待事件可能佔資料庫負載的 90%。此量值表示作用中工作階段平均花最多時間等待少量資源。如果您可以找出這些等待的原因，就可以嘗試解決方案。

等待事件依據資料庫引擎而有所差異：

- 如需關於所有 MariaDB 和 MySQL 等待事件的資訊，請參閱 MySQL 文件中的 [等待事件摘要表格](#)。
- 如需關於所有 PostgreSQL 等待事件的資訊，請參閱 PostgreSQL 文件中的 [統計數字收集器 > 等待事件表](#)。
- 如需關於所有 Oracle 等待事件的資訊，請參閱 Oracle 文件中的 [等待事件說明](#)。
- 如需所有 SQL Server 等待事件的相關資訊，請參閱 SQL 文件中的 [等待類別](#)。

Note

對於 Oracle，背景程序有時不需要相關的 SQL 陳述式即可發揮作用。在這些情況下，「績效詳情」會報告背景程序的類型，後面加上冒號，還有與該背景程序相關聯的等待類別。背景程序的類型包括 LGWR、ARC0、PMON 等。

例如，封存工具正在執行輸入/輸出時，績效詳情報告與 ARC1:System I/O 相當類似。有時候也會遺失背景程序類型，所以績效詳情只會報告等待類別，例如 :System I/O。

最高 SQL

等待事件會顯示瓶頸，而最高 SQL 則顯示哪些查詢對資料庫負載造成最大影響。例如，許多查詢目前可能正在資料庫上執行，但單一查詢可能會耗用 99% 的資料庫負載。在此情況下，高負載可能表示查詢發生問題。

根據預設，績效詳情主控台會顯示造成資料庫負載的常用 SQL 查詢。主控台也會顯示每個陳述式的相關統計資料。若要診斷特定陳述式的效能問題，您可以檢查其執行計劃。

計畫

執行計畫簡稱為計畫，是存取資料的一連串步驟。例如，聯結資料表 t1 和 t2 的計畫，可能會循環 t1 中的所有資料列，並將每一列與 t2 中的某列做比較。在關聯式資料庫中，內建程式碼最佳化工具可為 SQL 查詢最決定最有效的計畫。

對於資料庫執行個體，Performance Insights 會自動收集執行計劃。若要診斷 SQL 效能問題，請檢查擷取的計劃是否有高資源 SQL 查詢。這些計劃顯示了數據庫如何解析和運行查詢。

若要瞭解如何使用計劃分析資料庫負載，請參閱：

- Oracle: [使用績效詳情儀表板來分析 Oracle 執行計畫](#)
- SQL 伺服器： [使用 \[Performance Insights\] 儀表板分析 SQL Server 執行計畫](#)

計畫擷取

Performance Insights 每五分鐘就會識別最耗用資源的查詢，並擷取其計劃。因此您不需要手動收集和管理大量的計畫。而是可以使用 Top SQL (最高 SQL) 索引標籤，將重點放在問題最大的查詢的計畫上。

Note

若查詢的文字超過可收集的查詢文字上限，績效詳情就不會擷取其計畫。如需詳細資訊，請參閱 [在績效詳情儀表板中存取更多 SQL 文字](#)。

執行計畫的保留期間與所有 Performance Insights 資料的保留期間相同。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 [Performance Insights 的定價和資料保留](#)。

摘要查詢

Top SQL (最高 SQL) 索引標籤預設顯示摘要查詢。摘要查詢本身沒有計畫，但使用文字值的所有查詢都具有計畫。例如，摘要查詢可能包含文字 WHERE `email`=?。摘要可能包含兩個查詢，其中一個具有文字 WHERE email=user1@example.com，另一個含有 WHERE email=user2@example.com。這些文字查詢全都可能包含多個計畫。

當您選取摘要查詢時，主控台會顯示所選摘要之子陳述式的所有計畫。因此，您不需要查看所有的子陳述式來尋找計畫。您可能會看到前 10 個子陳述式清單中沒有顯示的計畫。主控台顯示已收集計畫的所有子查之計畫，無論這些查詢是否排入前 10 名。

最大 CPU

在儀表板中，資料庫負載圖表會收集、彙整並顯示工作階段資訊。若要查看作用中的工作階段是否超過最大 CPU，請查看它與最大 vCPU 數線的關係。Performance Insights 會根據資料庫執行個體的 vCPU (虛擬 CPU) 核心數量來決定最大 vCPU 值。

一次只能在 vCPU 上執行一個程序。如果程序數目超過 vCPUs 數目，則程序會開始排入佇列。佇列增加時，效能會受到影響。若資料庫負載通常高於最大 vCPU 數線，而主要等待狀態為 CPU，則 CPU 會超過負載。在此狀況下，您可能會想要節制與執行個體間的連線、以高 CPU 負載來微調任何 SQL 查詢、或者考慮使用較大的執行個體類別。處於任何等待狀態的密集且穩定的執行個體表示可能有您應解決的瓶頸或資源爭用問題。即使資料庫負載未超過最大 vCPU 數線，仍可能會有上述的問題。

Performance Insights 支援 Amazon RDS 資料庫引擎和執行個體類別

下列資料表提供您可找到支援 Performance Insights 的 Amazon RDS 資料庫引擎。

Note

如需 Amazon Aurora，請參閱《Amazon Aurora 使用者指南》中的[支援績效詳情的 Amazon Aurora 資料庫引擎](#)。

Amazon RDS 資料庫引擎	支援的引擎版本和區域	執行個體類別限制
Amazon RDS for MariaDB	如需 Performance Insights 搭配 RDS for MariaDB 的版本和區域可用性的詳細資訊，請參閱 支援的	下列執行個體類別不支援績效詳情： <ul style="list-style-type: none"> db.t2.micro

Amazon RDS 資料庫引擎	支援的引擎版本和區域	執行個體類別限制
	區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights。	<ul style="list-style-type: none"> • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
RDS for MySQL	如需 Performance Insights 搭配 RDS for MySQL 的版本和區域可用性的詳細資訊，請參閱 支援的區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights。	下列執行個體類別不支援績效詳情： <ul style="list-style-type: none"> • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
Amazon RDS for Microsoft SQL Server	如需 Performance Insights 搭配 RDS for SQL Server 的版本和區域可用性的詳細資訊，請參閱 支援的區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights。	N/A
Amazon RDS for PostgreSQL	如需 Performance Insights 搭配 RDS for PostgreSQL 的版本和區域可用性的詳細資訊，請參閱 支援的區域和資料庫引擎，提供 Amazon RDS 中的 Performance Insights。	N/A

Amazon RDS 資料庫引擎	支援的引擎版本和區域	執行個體類別限制
Amazon RDS for Oracle	如需 Performance Insights 搭配 RDS for Oracle 的版本和區域可用性的詳細資訊，請參閱 支援的區域和資料庫引擎 ，提供 Amazon RDS 中的 Performance Insights 。	N/A

Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別

下列資料表提供支援 Performance Insights 功能的 Amazon RDS 資料庫引擎。

功能	定價方案	支援的區域	支援的資料庫引擎	支援的執行個體類別
績效詳情的 SQL 統計數字	全部	全部	全部	全部
使用績效詳情儀表板來分析 Oracle 執行計畫	全部	全部	RDS for Oracle	全部
分析一段時間區間的資料庫效能	僅限付費方案	<ul style="list-style-type: none"> 美國東部 (俄亥俄) 美國東部 (維吉尼亞北部) 美國西部 (加利佛尼亞北部) 美國西部 (奧勒岡) 亞太區域 (孟買) 亞太區域 (首爾) 	RDS for PostgreSQL	全部

功能	<u>定價方案</u>	<u>支援的區域</u>	<u>支援的資料庫引擎</u>	<u>支援的執行個體類別</u>
		<ul style="list-style-type: none">• 亞太區域 (新加坡)• 亞太區域 (雪梨)• 亞太區域 (東京)• 加拿大 (中部)• 歐洲 (法蘭克福)• 歐洲 (愛爾蘭)• 歐洲 (倫敦)• 歐洲 (巴黎)• 歐洲 (斯德哥爾摩)		

功能	<u>定價方案</u>	<u>支援的區域</u>	<u>支援的資料庫引擎</u>	<u>支援的執行個體類別</u>
檢視 Performance Insights 主動建議	僅限付費方案	<ul style="list-style-type: none"> • 美國東部 (俄亥俄) • 美國東部 (維吉尼亞北部) • 美國西部 (加利佛尼亞北部) • 美國西部 (奧勒岡) • 亞太區域 (孟買) • 亞太區域 (首爾) • 亞太區域 (新加坡) • 亞太區域 (雪梨) • 亞太區域 (東京) • 加拿大 (中部) • 歐洲 (法蘭克福) • 歐洲 (愛爾蘭) • 歐洲 (倫敦) • 歐洲 (巴黎) • 歐洲 (斯德哥爾摩) • 南美洲 (聖保羅) 	全部	全部

Performance Insights 的定價和資料保留

根據預設，Performance Insights 提供免費方案，其中包括每月 7 天的效能資料歷史記錄和 1 百萬次 API 請求。您也可以購買較長的保留期間。如需有關費用的資訊，請參閱 [Performance Insights 定價](#)。

在 RDS 主控台中，您可以為 Performance Insights 資料選擇下列任一保留期間：

- 預設值 (7 天)
- n 月，其中 n 是介於 1 - 24 之間的數字

Performance Insights [Info](#)

Turn on Performance Insights [Info](#)

Retention period [Info](#)

7 days (free tier)	▲
7 days (free tier)	
1 month	
2 months	
3 months	
4 months	
5 months	
6 months	
7 months	
8 months	
9 months	
10 months	
11 months	
12 months	
13 months	
14 months	

若要了解如何使用 AWS CLI 設定保留期間，請參閱 [AWS CLI](#)。

開啟和關閉 Amazon RDS 的 Performance Insights

您可在建立資料庫執行個體或多可用區域資料庫叢集時為其開啟績效詳情。如有需要，您可於稍後將其關閉。開啟和關閉績效詳情不會造成停機、重新開機或容錯移轉。

Note

效能結構描述為 Amazon RDS for MariaDB 或 MySQL 使用的選用效能工具。若您開啟或關閉「效能架構」，則需要重新開機。然而，如果您開啟或關閉績效詳情，則無需重新開機。如需詳細資訊，請參閱 [在 Amazon RDS for MariaDB 或 MySQL 上開啟績效詳情的效能結構描述](#)。

績效詳情代理程式會耗用資料庫主機上有限的 CPU 和記憶體。當資料庫負載過高時，代理程式會減少收集資料的頻率來限制效能影響。

主控台

在主控台中，您可以在建立或修改資料庫執行個體時或多可用區域資料庫叢集時啟用或關閉 Performance Insights。

在建立資料庫執行個體或多可用區域資料庫叢集時，開啟或關閉績效詳情

當您建立新的資料庫執行個體或多可用區域資料庫叢集時，請在 Performance Insights (績效詳情) 區段中選擇 Enable Performance Insights (啟用績效詳情)，便可開啟績效詳情。或是選擇 Disable Performance Insights (停用績效詳情)。如需詳細資訊，請參閱下列主題：

- 若要建立資料庫執行個體，請遵循 [建立 Amazon RDS 資料庫執行個體](#) 中關於資料庫引擎的指示。
- 若要建立多可用區域資料庫叢集，請遵循 [建立多可用區域資料庫叢集](#) 中的資料庫引擎指示。

下列螢幕擷取畫面顯示 Performance Insights (績效詳情) 區段。



Turn on Performance Insights [Info](#)

Retention period [Info](#)

Default (7 days) ▼

AWS KMS Key [Info](#)

(default) aws/rds ▼

若您選擇 Enable Performance Insights (啟用績效詳情)，會有下列選項：

- Retention (保留) – 保留績效詳情資料的時間。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 [Performance Insights 的定價和資料保留](#)。
- AWS KMS key— 指定您的 AWS KMS key。績效詳情使用您的 KMS 金鑰來加密所有可能的敏感資料。將會對傳輸中與靜態資料進行加密。如需詳細資訊，請參閱 [設定績效詳情的 AWS KMS 政策](#)。

在修改或多可用區域資料庫叢集中的資料庫執行個體時，開啟或關閉績效詳情

在主控台中，您可於或多可用區域資料庫叢集中修改資料庫執行個體，以開啟或關閉績效詳情。

使用主控台為資料庫叢集中的資料庫執行個體開啟或關閉績效詳情

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 Databases (資料庫)。
3. 選擇一個資料庫執行個體或多可用區域資料庫叢集，然後選擇 Modify (修改)。
4. 在 Performance Insights (績效詳情) 區段中，選擇 Enable Performance Insights (啟用績效詳情) 或 Disable Performance Insights (停用績效詳情)。

若您選擇 Enable Performance Insights (啟用績效詳情)，會有下列選項：

- Retention (保留) – 保留績效詳情資料的時間。免費方案中的保留設定為 Default (7 days) (預設值 (7 天))。若要更長時間保留績效資料，請指定 1 - 24 個月。如需保留期間的詳細資訊，請參閱 [Performance Insights 的定價和資料保留](#)。
 - AWS KMS key – 指定您的 KMS 金鑰。績效詳情使用您的 KMS 金鑰來加密所有可能的敏感資料。將會對傳輸中與靜態資料進行加密。如需更多詳細資訊，請參閱 [加密 Amazon RDS 資源](#)。
5. 選擇 Continue (繼續)。
 6. 在 Scheduling of Modifications (修改排程) 中，選擇 Apply immediately (立即套用)。如果您選擇 Apply during the next scheduled maintenance window (在下一個排定的維護時段套用)，則執行個體會忽略此設定，並立即開啟 Performance Insights。
 7. 選擇 Modify instance (修改執行個體)。

AWS CLI

當您使用 [Create-db-instance](#) AWS CLI 命令時，請指定以開啟 Performance Insights。--enable-performance-insights 或者透過指定 --no-enable-performance-insights 來關閉 Performance Insights。

您也可以使用下列指 AWS CLI 令來指定這些值：

- [create-db-instance-read-replica](#)
- [modify-db-instance](#)
- [restore-db-instance-from-s3](#)
- [create-db-cluster](#) (多可用區域資料庫叢集)
- [modify-db-cluster](#) (多可用區域資料庫叢集)

下列程序說明如何使用 AWS CLI，為的現有資料庫執行個體開啟或關閉績效詳情。

若要AWS CLI

- 呼叫 [修改-db-執行個體](#) AWS CLI 命令，並提供下列值：
 - --db-instance-identifier – 資料庫執行個體名稱。
 - --enable-performance-insights 用來開啟，--no-enable-performance-insights 用來關閉

以下範例會開啟 sample-db-instance 的 Performance Insights。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier sample-db-instance \  
  --enable-performance-insights
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier sample-db-instance ^  
  --enable-performance-insights
```

當您開啟績效詳情時，可選擇使用 `--performance-insights-retention-period` 選項指定保留績效詳情資料的天數。您可指定為 7、`# * 31` (其中`#`是指 1 - 23) 的數字，或 731。例如，如果您想要保留效能資料 3 個月，請指定 93，也就是 3 個 * 31。預設值是 7 天。如需保留期間的詳細資訊，請參閱 [Performance Insights 的定價和資料保留](#)。

下列範例會開啟 `sample-db-instance` 的 Performance Insights，並指定 Performance Insights 資料保留 93 天 (3 個月)。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier sample-db-instance \  
  --enable-performance-insights \  
  --performance-insights-retention-period 93
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier sample-db-instance ^  
  --enable-performance-insights ^  
  --performance-insights-retention-period 93
```

如果您指定保留期間 (例如 94 天)，這不是有效值，RDS 就會發出錯誤。

```
An error occurred (InvalidParameterValue) when calling the CreateDBInstance operation:  
Invalid Performance Insights retention period. Valid values are: [7, 31, 62, 93, 124,  
 155, 186, 217,  
248, 279, 310, 341, 372, 403, 434, 465, 496, 527, 558, 589, 620, 651, 682, 713, 731]
```

RDS API

當您使用 [CreateDBInstance](#) 作業 Amazon RDS API 作業來建立新的資料庫執行個體時，請將 `EnablePerformanceInsights` 設為 `True` 以開啟績效詳情。若要關閉績效詳情，請將 `EnablePerformanceInsights` 設定為 `False`。

您也可以使用下列 API 操作來指定 `EnablePerformanceInsights` 值：

- [ModifyDBInstance](#)
- [創建InstanceRead數據庫副本](#)
- [恢復 B S3 InstanceFrom](#)

- [CreateDBCluster](#) (多可用區域資料庫叢集)
- [ModifyDBCluster](#) (多可用區域資料庫叢集)

當您開啟 Performance Insights 時，可以使用 `PerformanceInsightsRetentionPeriod` 參數來選擇保留 Performance Insights 資料的時間 (以天數為單位)。您可指定為 7、`# * 31` (其中 `#` 是指 1 - 23) 的數字，或 731。例如，如果您想要保留效能資料 3 個月，請指定 93，也就是 3 個 * 31。預設值是 7 天。如需保留期間的詳細資訊，請參閱 [Performance Insights 的定價和資料保留](#)。

在 Amazon RDS for MariaDB 或 MySQL 上開啟績效詳情的效能結構描述

效能結構描述為選用功能，用來在詳細資料低層級監控 Amazon RDS for MariaDB 或 MySQL 執行時間效能。效能結構描述專為盡量降低對資料庫效能的影響所設計。您在有或無效能結構描述的狀況下皆可使用績效詳情。

主題

- [效能結構描述概觀](#)
- [Performance Insights 和 Performance Schema](#)
- [透過績效詳情自動管理效能結構描述](#)
- [重新開機對效能結構描述的影響](#)
- [決定績效詳情是否管理效能結構描述](#)
- [設定效能結構描述為自動管理](#)

效能結構描述概觀

效能結構描述會監控 MariaDB 及 MySQL 資料庫中的事件。事件是個佔用時間的資料庫伺服器動作，並經過分析，可收集計時資訊。事件的範例如下：

- 函數呼叫
- 等待作業系統
- SQL 執行的階段
- SQL 陳述式的群組

PERFORMANCE_SCHEMA 儲存引擎是一種用於實作效能結構描述功能的機制。此引擎會使用資料庫來源碼中的檢測來收集事件資料。引擎會將事件儲存於 `performance_schema` 資料庫的僅限記憶體表

格中。您可查詢 `performance_schema`，就像您可以查詢任何其他表格一樣。如需 MySQL 效能結構描述的詳細資訊，請參閱 [MySQL 參考手冊](#) 中的 MySQL 效能結構描述。

Performance Insights 和 Performance Schema

績效詳情和效能結構描述是獨立的功能，但彼此互相連接。Performance Insights 對於 Amazon RDS for MariaDB 或 MySQL 的行為取決於 Performance Schema 是否開啟，如果是的話，Performance Insights 是否自動管理 Performance Schema。下表說明行為。

Performance Schema 已開啟	Performance Insights 管理模式	Performance Insights 行為
是	自動	<ul style="list-style-type: none"> • 收集詳細的低階監控資訊 • 收集每秒作用中工作階段指標 • 顯示依詳細等待事件分類的資料庫負載，您可以用來識別瓶頸
是	手動	<ul style="list-style-type: none"> • 收集等待事件和每個 SQL 指標 • 收集每五秒 (而不是每秒) 作用中工作階段指標 • 報告使用者狀態，例如插入和傳送，無法協助您識別瓶頸
否	N/A	<ul style="list-style-type: none"> • 不收集等待事件、每個 SQL 指標或其他詳細的低階監控資訊 • 收集每五秒 (而不是每秒) 作用中工作階段指標 • 報告使用者狀態，例如插入和傳送，無法協助您識別瓶頸

透過績效詳情自動管理效能結構描述

當您在開啟績效詳情的狀況下建立 Amazon RDS for MariaDB 或 MySQL 資料庫執行個體時，也會開啟效能結構描述。在此狀況下，績效詳情會自動管理您的效能結構描述參數。這是建議的組態。

當 Performance Insights 自動管理效能結構描述時，來源 `performance_schema` 為 `system`。

Note

`t4g.medium` 執行個體類別不支援效能結構描述的自動管理。

若您手動變更 `performance_schema` 參數，但稍後想變更為自動管理，請參閱 [設定效能結構描述為自動管理](#)。

Important

當績效詳情開啟效能結構描述時，其不會變更參數群組值。不過，這些值會在執行的資料庫執行個體上進行變更。查看變更值的唯一方法是執行 `SHOW GLOBAL VARIABLES` 命令。

重新開機對效能結構描述的影響

績效詳情和效能結構描述在資料庫執行個體重新開機的要求方面有所不同：

效能結構描述

若要開啟或關閉此功能，您必須重新開機資料庫執行個體。

Performance Insights

若要開啟或關閉此功能，不需要重新開機資料庫執行個體。

如果效能結構描述目前未開啟，並且您在未重新開機資料庫執行個體的情況下開啟績效詳情，則不會開啟效能結構描述。

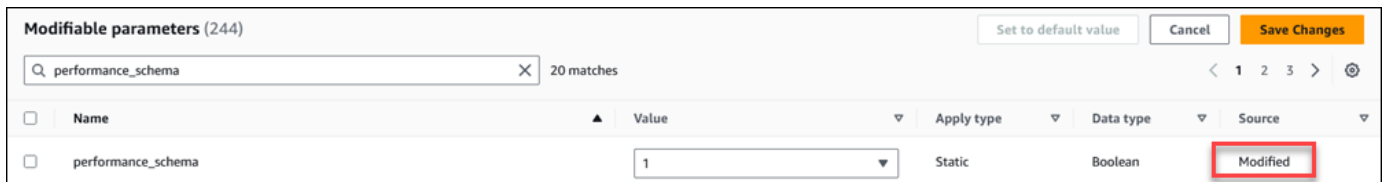
決定績效詳情是否管理效能結構描述

若要了解績效詳情目前是否正在管理主要引擎 5.6、5.7 和 8.0 版的績效詳情，請查看下表。

performance_schema 參數的設定	來源欄的設定	績效詳情管理效能結構描述
0	system	是
0 或 1	user	否

若要判斷績效詳情是否自動管理效能結構描述

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 Parameter groups (參數群組)。
3. 選擇資料庫執行個體的參數群組名稱。
4. 在搜尋列中，輸入 **performance_schema**。
5. 檢查 Source (來源) 是否為系統預設值，而 Values (值) 是否為 0。若是如此，績效詳情正在自動管理效能結構描述。若否，則績效詳情並未自動管理效能結構描述。



設定效能結構描述為自動管理

假設您的資料庫執行個體或多可用區域資料庫叢集的績效詳情已啟用，但目前並未管理效能結構描述。如果要允許績效詳情自動管理效能結構描述，請完成以下步驟。

設定效能結構描述為自動管理

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 Parameter groups (參數群組)。
3. 選擇資料庫執行個體或多可用區域資料庫叢集的參數群組名稱。
4. 在搜尋列中，輸入 **performance_schema**。
5. 選取 performance_schema 參數。
6. 選擇 Edit parameters (編輯參數)。

7. 選取 `performance_schema` 參數。
8. 於 Values (數值) 中，選擇 0。
9. 選擇儲存變更。
10. 重新啟動資料庫執行個體或多可用區域資料庫叢集。

Important

每當您開啟或關閉效能結構描述時，請務必重新開機資料庫執行個體或多可用區域資料庫叢集。

如需修改執行個體參數的相關資訊，請參閱 [修改資料庫參數群組中的參數](#)。如需儀表板中的詳細資訊，請參閱 [使用績效詳情儀表板來分析指標](#)。如需 MySQL 效能結構描述的詳細資訊，請參閱 [MySQL 8.0 參考手冊](#)。

設定績效詳情的存取政策

若要存取 Performance Insights，主體必須具有 AWS Identity and Access Management (IAM) 的適當許可。您可透過以下方式授予存取權：

- 將 `AmazonRDSPerformanceInsightsReadOnly` 受管政策附加至許可設定或角色，以存取 Performance Insights API 的僅供讀取操作。
- 將 `AmazonRDSPerformanceInsightsFullAccess` 受管政策附加至許可設定或角色，以存取 Performance Insights API 的所有操作。
- 建立自訂 IAM 政策，並將其連接至許可集或角色。

如果您在開啟 Performance Insights 時指定了客戶管理金鑰，請確定您帳戶中的使用者擁有 `kms:Decrypt` 和 `kms:GenerateDataKey` 權限 AWS KMS key

將 `AmazonRDSPerformanceInsightsReadOnly` 策附加到 IAM 主體

`AmazonRDSPerformanceInsightsReadOnly` 這是一項 AWS 受管政策，可授予對 Amazon RDS Performance Insights API 所有唯讀操作的存取權。

若您將 `AmazonRDSPerformanceInsightsReadOnly` 連接至許可集或角色，收件人可使用績效詳情搭配其他主控台功能。

如需詳細資訊，請參閱 [AWS 管理策略：亞馬遜 PerformanceInsightsReadOnly](#)。

將 AmazonRDSPerformanceInsightsFullAccess 策略附加到 IAM 主體

AmazonRDSPerformanceInsightsFullAccess 這是一項 AWS 受管政策，可授予對 Amazon RDS Performance Insights API 所有操作的存取權。

若您將 AmazonRDSPerformanceInsightsFullAccess 連接至許可集或角色，收件人可使用績效詳情搭配其他主控台功能。

如需詳細資訊，請參閱 [AWS 管理策略：亞馬遜 PerformanceInsightsFullAccess](#)。

建立績效詳情的自訂 IAM 政策

對於沒有或 AmazonRDSPerformanceInsightsFullAccess 政策的使用者，您可以透過建立 AmazonRDSPerformanceInsightsReadOnly 或修改使用者管理的 IAM 政策來授與 Performance Insights 的存取權。在將該政策連接至許可集或角色時，收件人可以使用績效詳情。

建立自訂政策

1. 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在導覽窗格中，選擇政策。
3. 選擇 Create policy (建立政策)。
4. 在 [建立原則] 頁面上，選擇 [JSON] 選項。
5. 複製並貼上「AWS 受管理策略參考指南」[AmazonRDSPerformanceInsightsReadOnly](#) 或 [AmazonRDSPerformanceInsightsFullAccess](#) 政策中「JSON 政策文件」區段中提供的文字。
6. 選擇 Review policy (檢閱政策)。
7. 為政策提供名稱並選擇性輸入描述，然後選擇 Create policy (建立政策)。

現在您可以將政策連接到許可集或角色。以下程序假設您已有基於此用途使用的使用者。

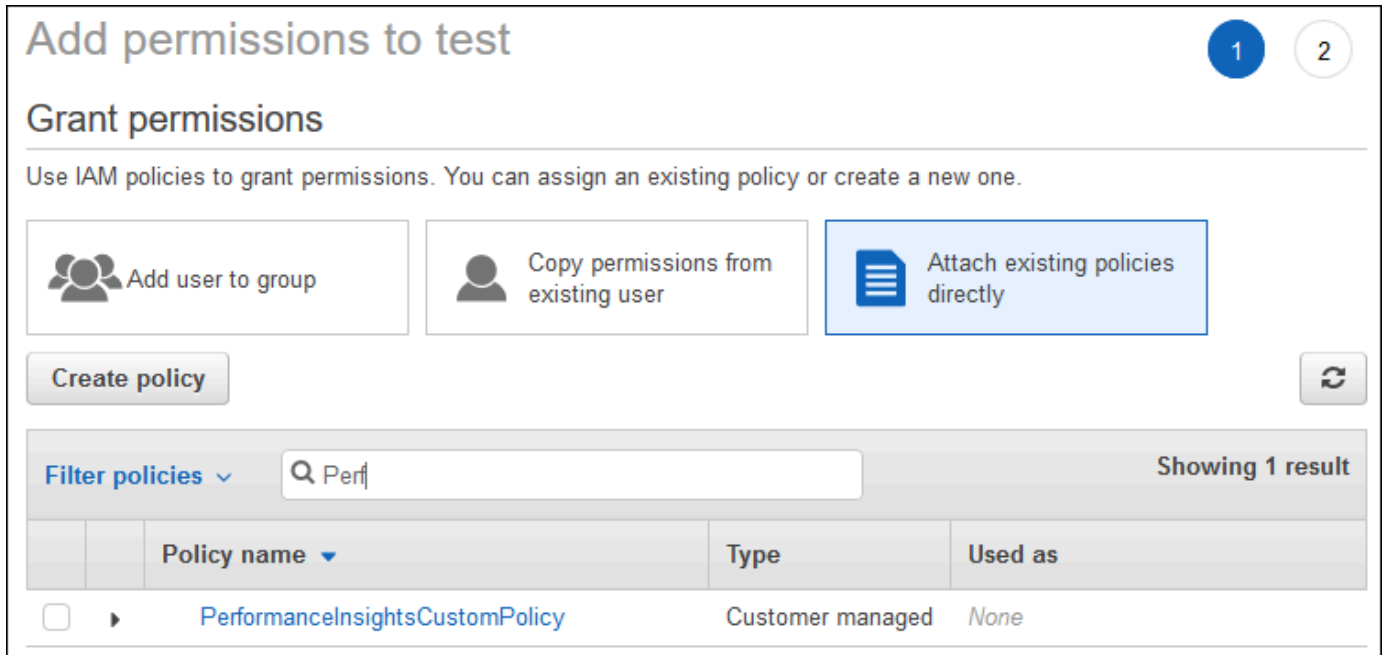
將政策連接至使用者

1. 在以下網址開啟 IAM 主控台：<https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇 Users (使用者)。
3. 從清單中選擇現有的使用者。

⚠ Important

若要使用績效詳情，除了自訂政策以外，請務必確認您還擁有存取 Amazon RDS 的權限。例如，AmazonRDSPerformanceInsightsReadOnly 預先定義政策提供對 Amazon RDS 的唯讀存取。如需更多詳細資訊，請參閱 [使用政策管理存取權](#)。

- 在 Summary (摘要) 頁面上，選擇 Add permissions (新增許可)。
- 選擇 Attach existing policies directly (直接連接現有政策)。對於搜尋，請輸入原則名稱的前幾個字元，如下圖所示。



The screenshot shows the 'Add permissions to test' interface in the AWS IAM console. It includes a 'Grant permissions' section with three main options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. Below these are buttons for 'Create policy' and a refresh icon. A search bar is used to filter policies, with the text 'Perf' entered. The search results show one policy: 'PerformanceInsightsCustomPolicy' with a type of 'Customer managed' and 'Used as' set to 'None'.

Policy name	Type	Used as
PerformanceInsightsCustomPolicy	Customer managed	None

- 選擇您的政策，然後選擇 Next: Review (下一步：檢閱)。
- 選擇 Add permissions (新增許可)。

設定績效詳情的 AWS KMS 政策

Performance Insights 使用 AWS KMS key 加密機密資料。當您透過 API 或主控台啟用績效詳情時，可以執行下列任一操作：

- 選擇預設值 AWS 受管金鑰。

Amazon RDS 會 AWS 受管金鑰 針對您的新資料庫執行個體使用。Amazon RDS 會為您的 AWS 帳戶建立 AWS 受管金鑰。你 AWS 帳戶 有一個不同 AWS 受管金鑰 的 Amazon RDS 為每個 AWS 區域。

- 選擇客戶受管金鑰。

如果您指定客戶受管金鑰，則您帳戶中呼叫績效詳情 API 的使用者需要 KMS 金鑰的 `kms:Decrypt` 和 `kms:GenerateDataKey` 許可。您可以透過 IAM 政策設定這些許可。不過，我們建議您透過 KMS 金鑰政策來管理這些許可。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [在 AWS KMS 中使用金鑰政策](#)。

Example

下列範例說明如何將陳述式新增至 KMS 金鑰政策。這些陳述式允許存取績效詳情。視您使用 KMS 金鑰的方式而定，您可能想要變更某些限制。在將陳述式新增至您的政策之前，請先移除所有註解。

```
{
  "Version" : "2012-10-17",
  "Id" : "your-policy",
  "Statement" : [ {
    //This represents a statement that currently exists in your policy.
  }
  ....,
  //Starting here, add new statement to your policy for Performance Insights.
  //We recommend that you add one new statement for every RDS instance
  {
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        //One or more principals allowed to access Performance Insights
        "arn:aws:iam::444455556666:role/Role1"
      ]
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" : {
      "StringEquals" : {
        //Restrict access to only RDS APIs (including Performance Insights).
        //Replace region with your AWS Region.
        //For example, specify us-west-2.
        "kms:ViaService" : "rds.region.amazonaws.com"
      }
    }
  },
}
```

```
"ForAnyValue:StringEquals": {
  //Restrict access to only data encrypted by Performance Insights.
  "kms:EncryptionContext:aws:pi:service": "rds",
  "kms:EncryptionContext:service": "pi",

  //Restrict access to a specific RDS instance.
  //The value is a DbResourceId.
  "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
}
}
```

Performance Insights 如何使用 AWS KMS 客戶管理的金鑰

Performance Insights 使用客戶壽館金鑰來加密敏感資料。當您開啟 Performance Insights 時，可以透過 API 提供 AWS KMS 金鑰。Performance Insights 會在此金鑰上建立 KMS 許可。它會使用金鑰並執行必要的操作來處理敏感資料。敏感資料包括使用者、資料庫、應用程式及 SQL 查詢文字等欄位。Performance Insights 可確保靜態資料和傳輸中資料保持加密狀態。

Performance Insights IAM 如何搭配使用 AWS KMS

IAM 會提供許可給特定 API。Performance Insights 具有下列公有 API，您可以使用 IAM 政策來限制這些 API：

- DescribeDimensionKeys
- GetDimensionKeyDetails
- GetResourceMetadata
- GetResourceMetrics
- ListAvailableResourceDimensions
- ListAvailableResourceMetrics

您可以使用下列 API 請求來取得敏感資料。

- DescribeDimensionKeys
- GetDimensionKeyDetails
- GetResourceMetrics

當您使用 API 取得敏感資料時，Performance Insights 會利用呼叫者的憑證。此檢查可確保敏感資料的存取權限於可存取 KMS 金鑰的人。

呼叫這些 API 時，您需要許可才能透過 IAM 政策和許可呼叫 API，以透過 AWS KMS 金鑰政策叫用 `kms:decrypt` 動作。

`GetResourceMetrics` API 可能會同時傳回敏感和非敏感資料。請求參數會決定回應是否應包含敏感資料。當請求的 `filter` 或 `group-by` 參數中包含敏感維度時，API 會傳回敏感資料。

如需可搭配 `GetResourceMetrics` API 使用之維度的詳細資訊，請參閱 [〈〉 DimensionGroup](#)。

Example 範例

以下範例會請求 `db.user` 群組的敏感資料：

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZ",
  "MetricQueries": [
    {
      "Metric": "db.load.avg",
      "GroupBy": {
        "Group": "db.user",
        "Limit": 2
      }
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

Example

以下範例會請求 db.load.avg 指標的非敏感資料：

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZ",
  "MetricQueries": [
    {
      "Metric": "db.load.avg"
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

授予 Performance Insights 的細微存取權

精細的存取控制提供了控制 Performance Insights 存取的其他方式。此存取控制可允許或拒絕存取 GetResourceMetrics、DescribeDimensionKeys 和「Perfor GetDimensionKeyDetails mance Insights」動作的個別維度。若要使用精細的存取權，請使用條件金鑰在 IAM 政策中指定維度。存取權的評估遵循 IAM 政策評估邏輯。如需詳細資訊，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。如果 IAM 政策陳述式未指定任何維度，則陳述式會控制指定動作所有維度的存取權。如需可用維度的清單，請參閱[DimensionGroup](#)。

若要找出您的認證有權存取的維度，請使用中的 AuthorizedActions 參數 ListAvailableResourceDimensions 並指定動作。允許的值 AuthorizedActions 如下所示：

- GetResourceMetrics

- DescribeDimensionKeys
- GetDimensionKeyDetails

例如，如果您指定 `GetResourceMetrics` 為 `AuthorizedActions` 參數，則會 `ListAvailableResourceDimensions` 傳回 `GetResourceMetrics` 動作授權存取的維度清單。如果您在 `AuthorizedActions` 參數中指定多個動作，則會 `ListAvailableResourceDimensions` 傳回授權存取這些動作的維度交集。

Example

下列範例提供對於和 `DescribeDimensionKeys` 動作之指定維度 `GetResourceMetrics` 的存取權。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowToDiscoverDimensions",
      "Effect": "Allow",
      "Action": [
        "pi:ListAvailableResourceDimensions"
      ],
      "Resource": [
        "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW"
      ]
    },
    {
      "Sid": "SingleAllow",
      "Effect": "Allow",
      "Action": [
        "pi:GetResourceMetrics",
        "pi:DescribeDimensionKeys"
      ],
      "Resource": [
        "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          // only these dimensions are allowed. Dimensions not included in
          // a policy with "Allow" effect will be denied
          "pi:Dimensions": [
```

```

        "db.sql_tokenized.id",
        "db.sql_tokenized.statement"
    ]
  }
}
]
}

```

以下是要求維度的回應：

```

// ListAvailableResourceDimensions API
// Request
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW",
  "Metrics": [ "db.load" ],
  "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
  "MetricDimensions": [ {
    "Metric": "db.load",
    "Groups": [
      {
        "Group": "db.sql_tokenized",
        "Dimensions": [
          { "Identifier": "db.sql_tokenized.id" },
          // { "Identifier": "db.sql_tokenized.db_id" }, // not included
because not allows in the IAM Policy
          { "Identifier": "db.sql_tokenized.statement" }
        ]
      }
    ]
  }
]
}

```

下列範例會為維度指定一個允許和兩個拒絕存取權。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowToDiscoverDimensions",
      "Effect": "Allow",
      "Action": [
        "pi:ListAvailableResourceDimensions"
      ],
      "Resource": [
        "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW"
      ]
    },
    {
      "Sid": "001AllowAllWithoutSpecifyingDimensions",
      "Effect": "Allow",
      "Action": [
        "pi:GetResourceMetrics",
        "pi:DescribeDimensionKeys"
      ],
      "Resource": [
        "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW"
      ]
    },
    {
      "Sid": "001DenyAppDimensionForAll",
      "Effect": "Deny",
      "Action": [
        "pi:GetResourceMetrics",
        "pi:DescribeDimensionKeys"
      ],
      "Resource": [
        "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "pi:Dimensions": [

```

```

        "db.application.name"
    ]
  }
},
{
  "Sid": "001DenySQLForGetResourceMetrics",
  "Effect": "Deny",
  "Action": [
    "pi:GetResourceMetrics"
  ],
  "Resource": [
    "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUW3W"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "pi:Dimensions": [
        "db.sql_tokenized.statement"
      ]
    }
  }
}
]
}

```

以下是要求維度的回應：

```

// ListAvailableResourceDimensions API
// Request
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUW3W",
  "Metrics": [ "db.load" ],
  "AuthorizedActions": ["GetResourceMetrics"]
}

// Response
{
  "MetricDimensions": [ {

```

```

    "Metric": "db.load",
    "Groups": [
      {
        "Group": "db.application",
        "Dimensions": [
          // removed from response because denied by the IAM Policy
          // { "Identifier": "db.application.name" }
        ]
      },
      {
        "Group": "db.sql_tokenized",
        "Dimensions": [
          { "Identifier": "db.sql_tokenized.id" },
          { "Identifier": "db.sql_tokenized.db_id" },
          // removed from response because denied by the IAM Policy
          // { "Identifier": "db.sql_tokenized.statement" }
        ]
      },
      ...
    ] }
  ] }
}

```

```

// ListAvailableResourceDimensions API
// Request
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUVWXYZW",
  "Metrics": [ "db.load" ],
  "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
  "MetricDimensions": [ {
    "Metric": "db.load",
    "Groups": [
      {
        "Group": "db.application",
        "Dimensions": [

```

```
        // removed from response because denied by the IAM Policy
        // { "Identifier": "db.application.name" }
    ]
},
{
    "Group": "db.sql_tokenized",
    "Dimensions": [
        { "Identifier": "db.sql_tokenized.id" },
        { "Identifier": "db.sql_tokenized.db_id" },

        // allowed for DescribeDimensionKeys because our IAM Policy
        // denies it only for GetResourceMetrics
        { "Identifier": "db.sql_tokenized.statement" }
    ]
},
...
] }
]
```

使用績效詳情儀表板來分析指標

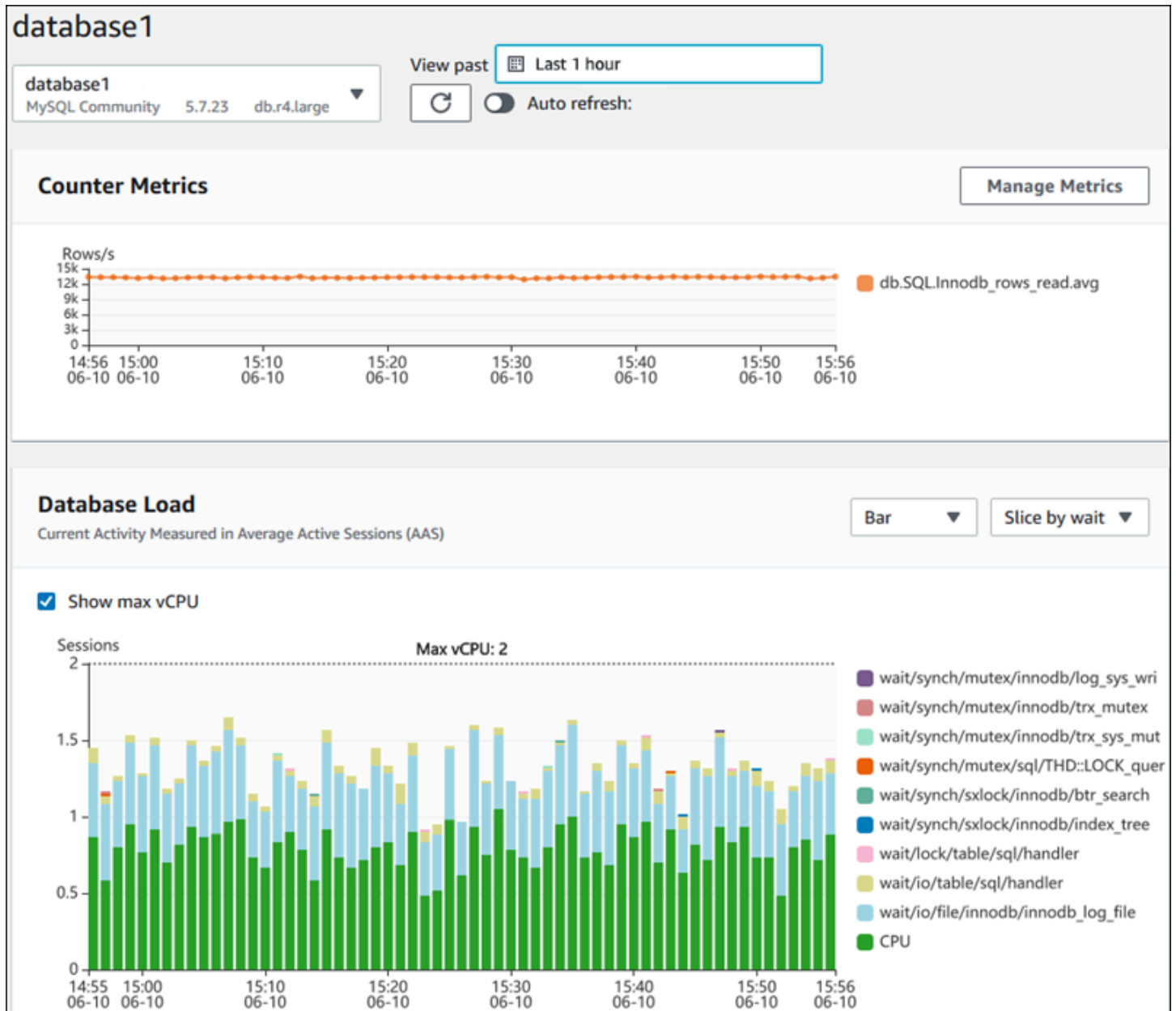
績效詳情儀表板包含資料庫效能資訊，可協助您分析效能問題並對其進行故障排除。在主要儀表板頁面上，您可以檢視關於資料庫負載的資訊。您可以依據等待事件或 SQL 等維度「配量」資料庫負載。

績效詳情儀表板

- [績效詳情儀表板概觀](#)
- [存取績效詳情儀表板](#)
- [按等待事件分析資料庫負載](#)
- [分析一段時間區間的資料庫效能](#)
- [在績效詳情儀表板中分析查詢](#)
- [分析最上層 Oracle PDB 負載](#)
- [使用 Performance Insights 儀表板分析執行計劃](#)

績效詳情儀表板概觀

儀表板是與績效詳情進行互動的最簡單方式。下列範例顯示 MySQL 資料庫執行個體的儀表板。

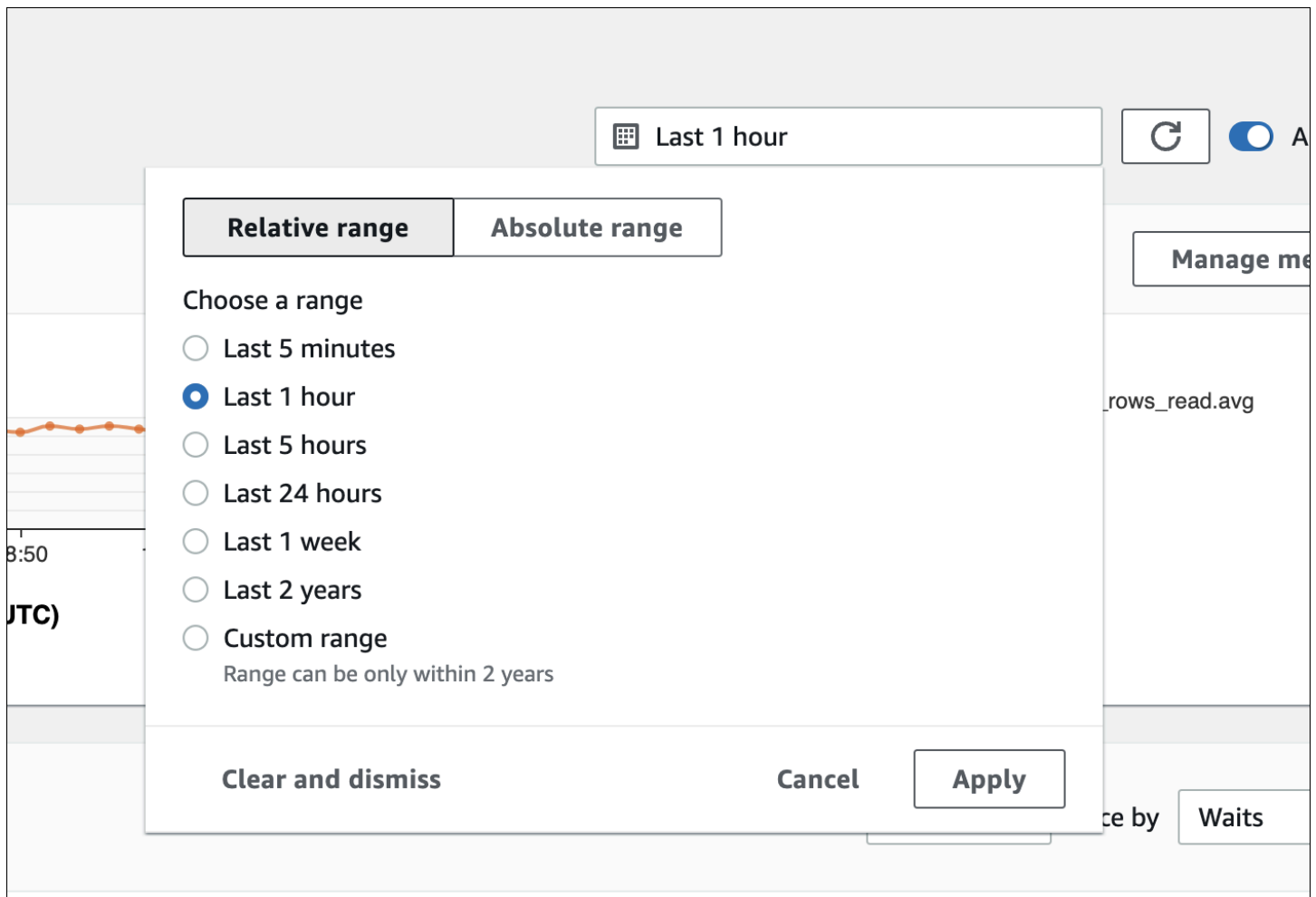


主題

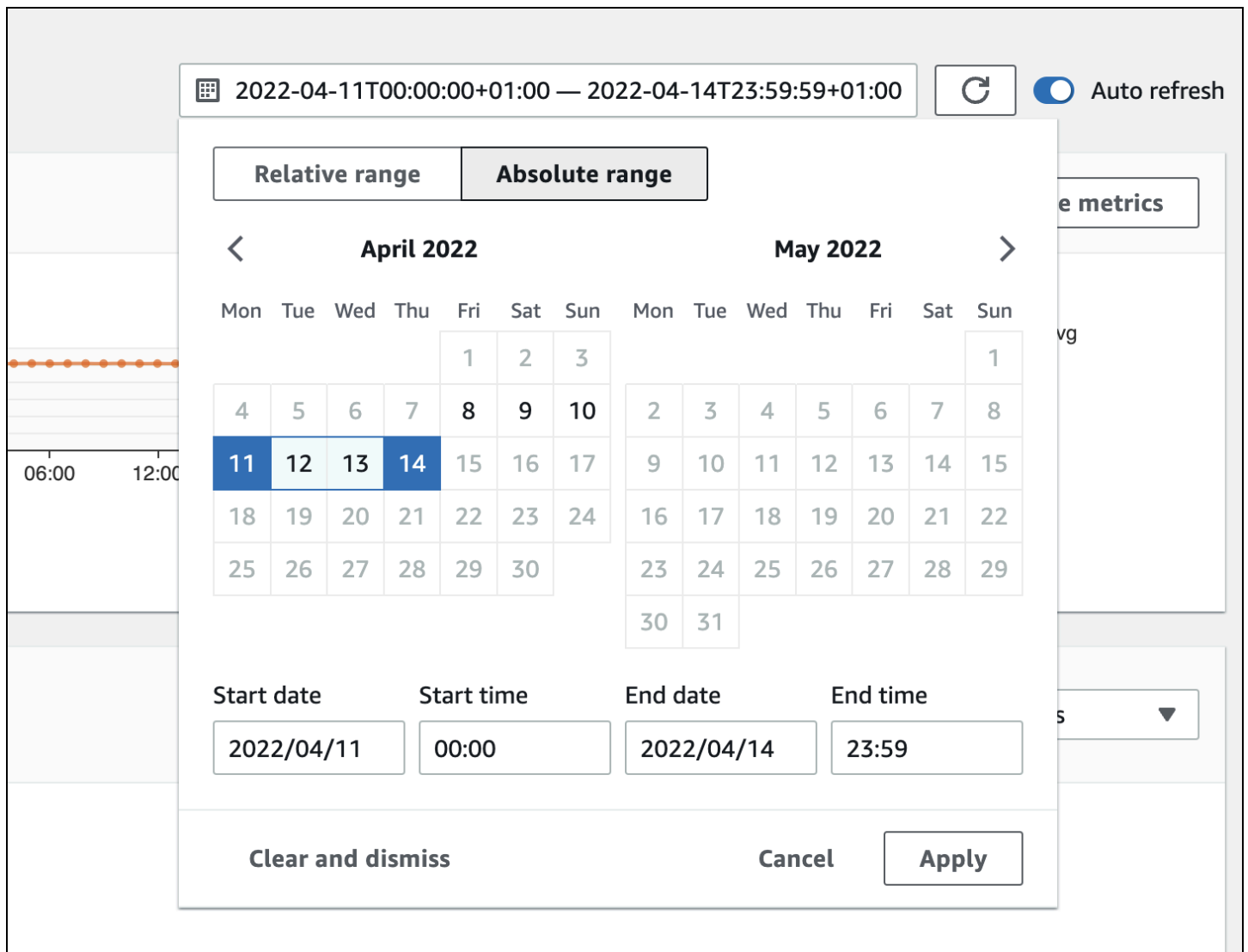
- [時間範圍篩選](#)
- [計數器指標圖表](#)
- [資料庫負載圖表](#)
- [最高維度表格](#)

時間範圍篩選

依預設，績效詳情儀表板會顯示過去一小時的資料庫負載。您可以將此範圍調整為短至 5 分鐘或長達 2 年。您也可以選取自訂相對範圍。



您可以選擇具有開始和結束日期時間的絕對範圍。以下範例顯示從 22/4/11 午夜開始到 22/4/14 晚上 11:59 結束的時間範圍。

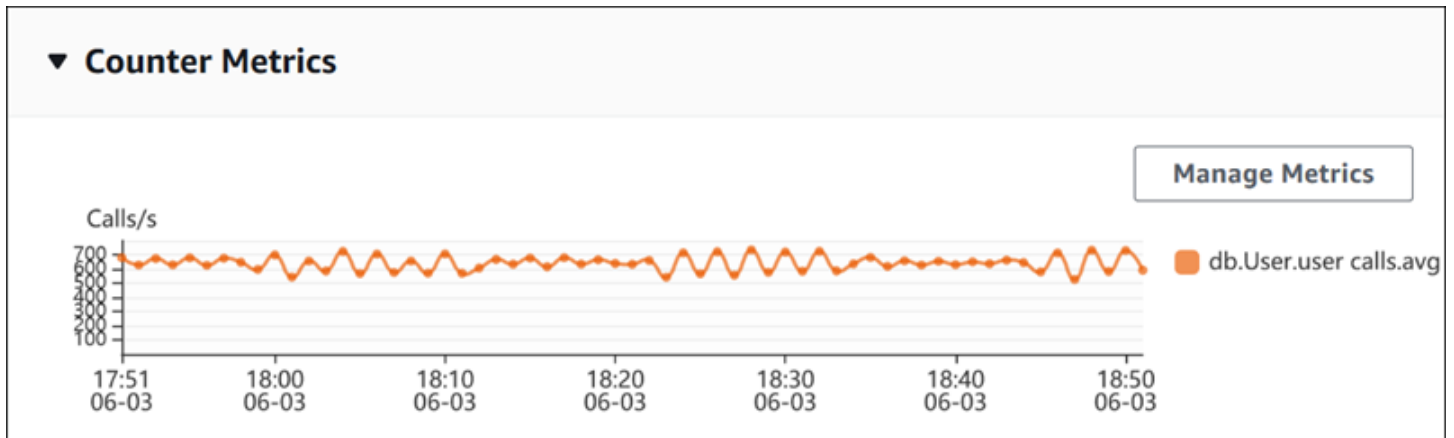


計數器指標圖表

您可以使用計數器指標來自訂績效詳情儀表板，以包含高達 10 個其它圖表。這些圖表顯示很多作業系統和資料庫效能指標。您可以將此資訊與資料庫負載相互關聯，以協助識別和分析效能問題。

Counter Metrics (計數器指標) 圖表顯示效能計數器的資料。預設指標取決於資料庫引擎：

- MySQL 和 MariaDB – `db.SQL.Innodb_rows_read.avg`
- Oracle – `db.User.user_calls.avg`
- Microsoft SQL Server – `db.Databases.Active Transactions(_Total).avg`
- PostgreSQL – `db.Transactions.xact_commit.avg`



若要變更效能計數器，請選擇 Manage Metrics (管理指標)。您可以選取多個作業系統指標或資料庫指標，如下列螢幕擷取畫面所示。若要查看任何指標的詳細資訊，請將游標移到指標名稱上。

Select metrics shown on the graph ✕

Check the metrics that you want to see on the Performance Insights dashboard.

OS metrics (0)
Database metrics (1)
Clear all selections

▼ User

<input type="checkbox"/> CPU used by this session	<input type="checkbox"/> SQL*Net roundtrips to/from client	<input type="checkbox"/> bytes received via SQL*Net from client
<input type="checkbox"/> user commits	<input type="checkbox"/> logons cumulative	<input checked="" type="checkbox"/> user calls
<input type="checkbox"/> bytes sent via SQL*Net to client	<input type="checkbox"/> user rollbacks	

▼ Redo

redo size

▼ Cache

<input type="checkbox"/> physical read bytes	<input type="checkbox"/> db block gets	<input type="checkbox"/> DBWR checkpoints
<input type="checkbox"/> physical reads	<input type="checkbox"/> consistent gets from cache	<input type="checkbox"/> db block gets from cache
<input type="checkbox"/> consistent gets		

▼ SQL

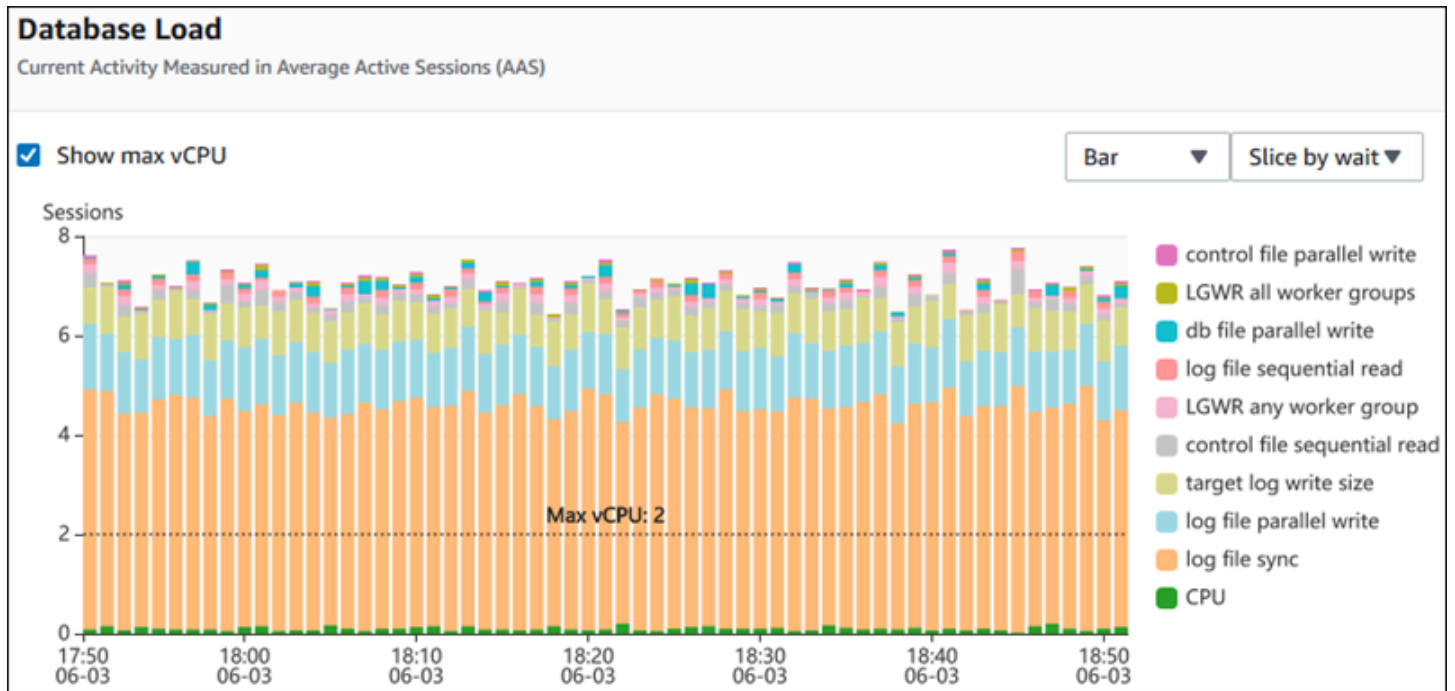
<input type="checkbox"/> parse count (total)	<input type="checkbox"/> parse count (hard)	<input type="checkbox"/> table scan rows gotten
<input type="checkbox"/> sorts (memory)	<input type="checkbox"/> sorts (disk)	<input type="checkbox"/> sorts (rows)

Cancel
Update graph

如需有關可為每個資料庫引擎新增之計數器指標的說明，請參閱 [Performance Insights 計數器指標](#)。

資料庫負載圖表

資料庫負載圖表顯示資料庫活動與資料庫執行個體容量間的比較值，以最大 vCPU 數線表示。依預設，堆疊折線圖以每個時間單位的平均作用中工作階段數來表示資料庫負載。資料庫負載依等待狀態切割 (分組)。

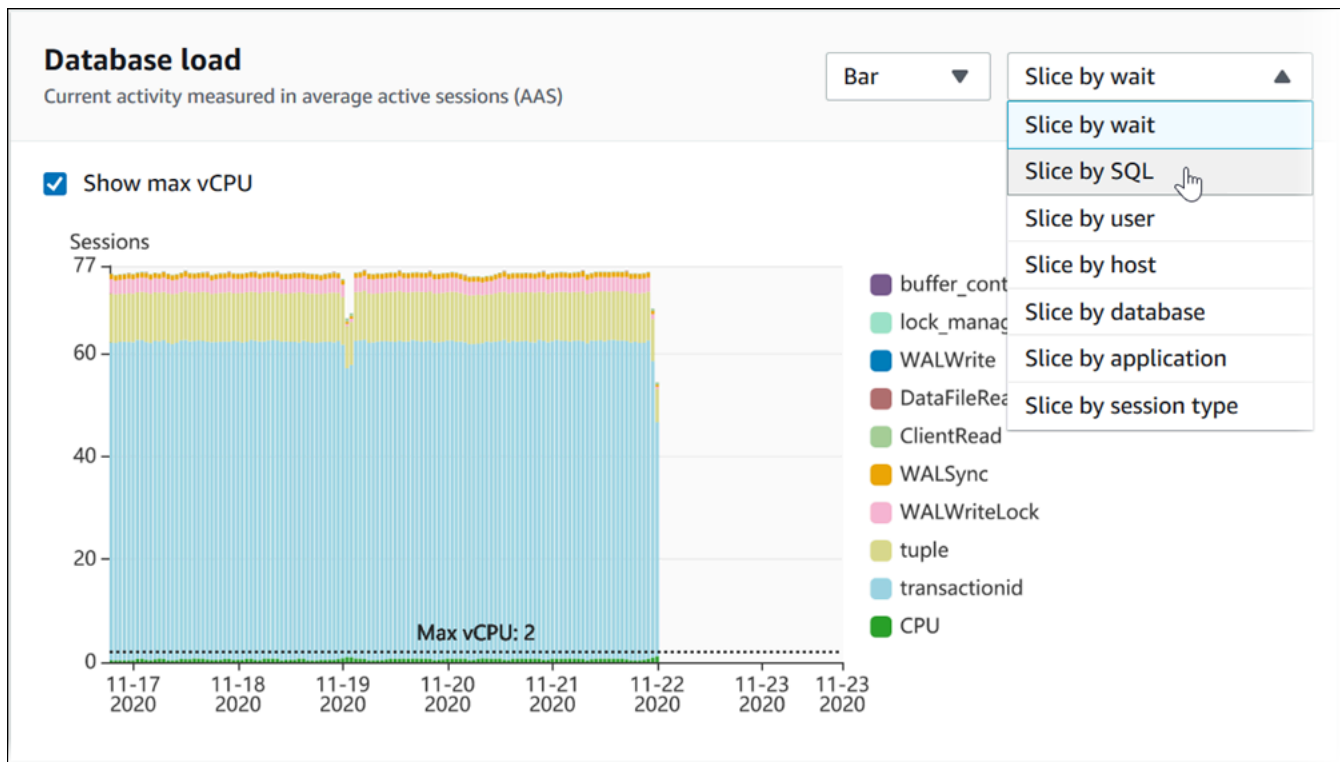


資料庫負載依維度配量

您可以選擇將負載顯示為作用中工作階段 (依任何支援維度分組)。下表顯示不同引擎各自支援的維度。

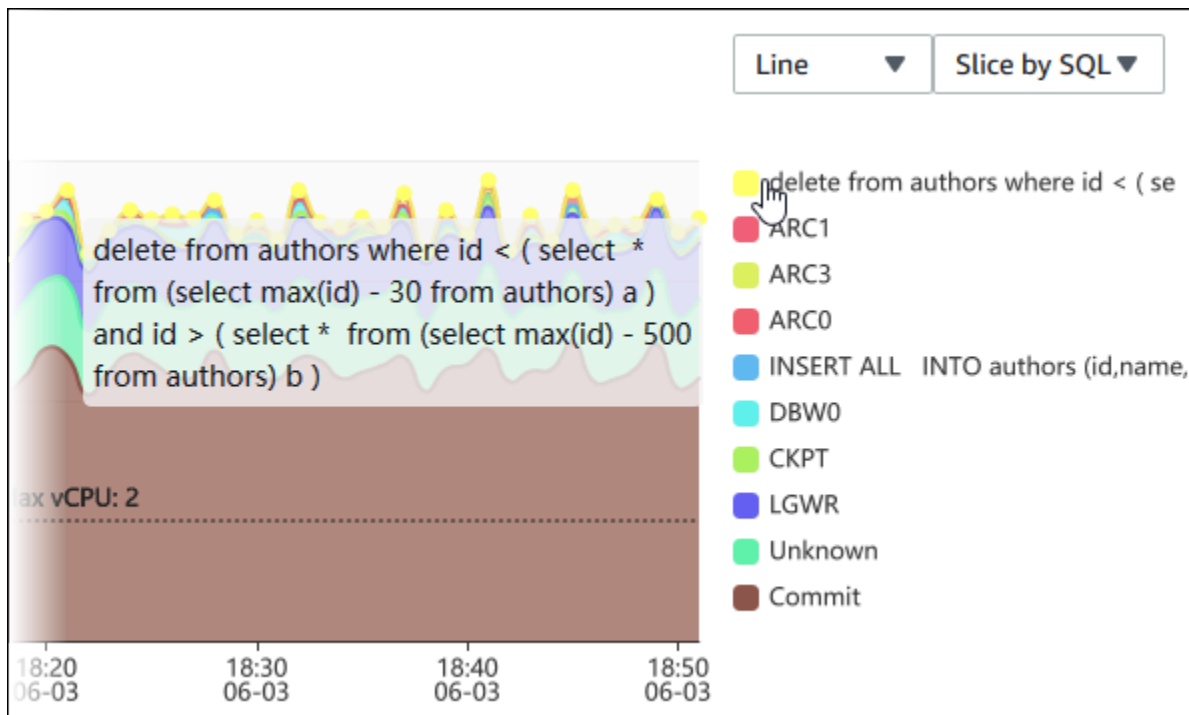
維度	Oracle	SQL Server	PostgreSQL	MySQL
主機	是	是	是	是
SQL	是	是	是	是
使用者	是	是	是	是
等待	是	是	是	是
計畫	是	否	否	否
應用程式	否	否	是	否
資料庫	否	否	是	是
工作階段類型	否	否	是	否

下圖顯示 PostgreSQL 資料庫執行個體的維度。

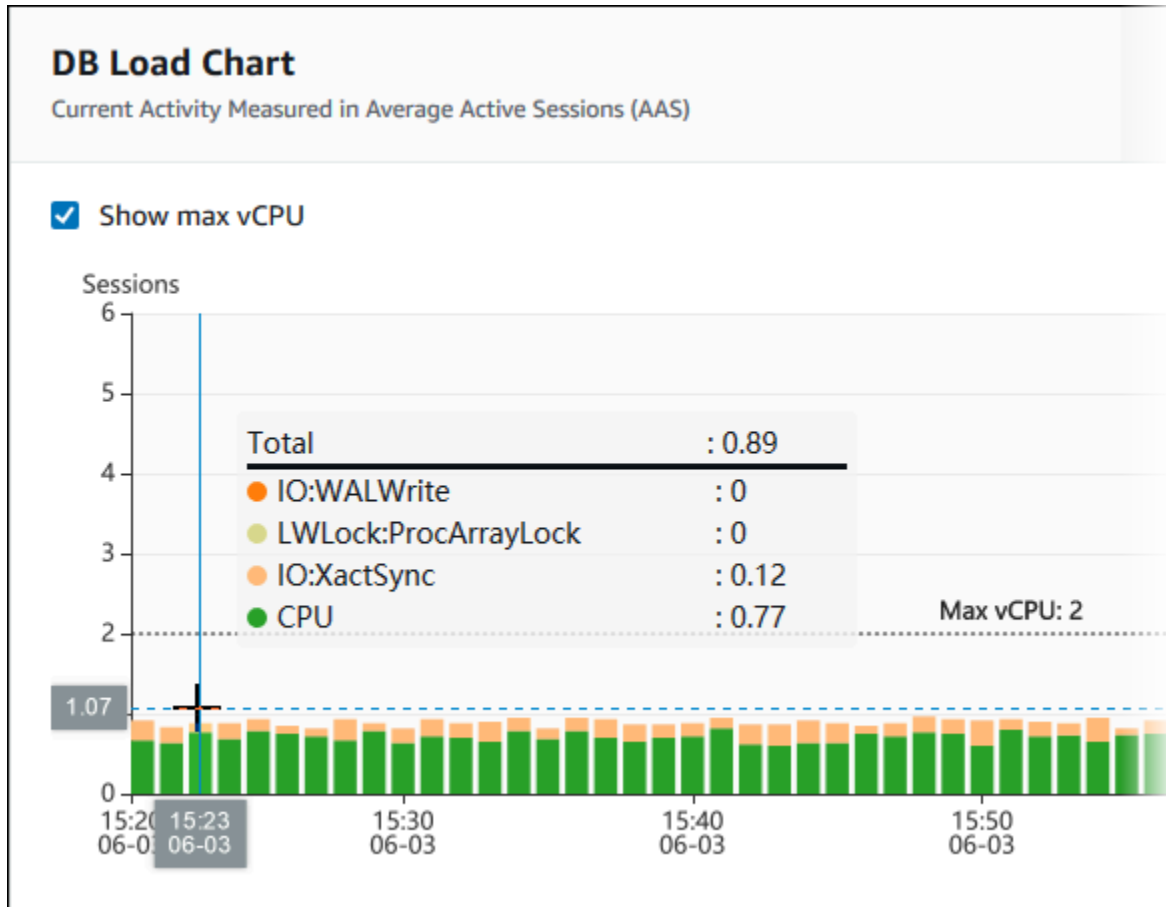


維度項目的資料庫負載詳細資訊

若要查看有關維度內資料庫負載項目的詳細資訊，請將游標移到項目名稱上。下圖顯示 SQL 陳述式的詳細資訊。

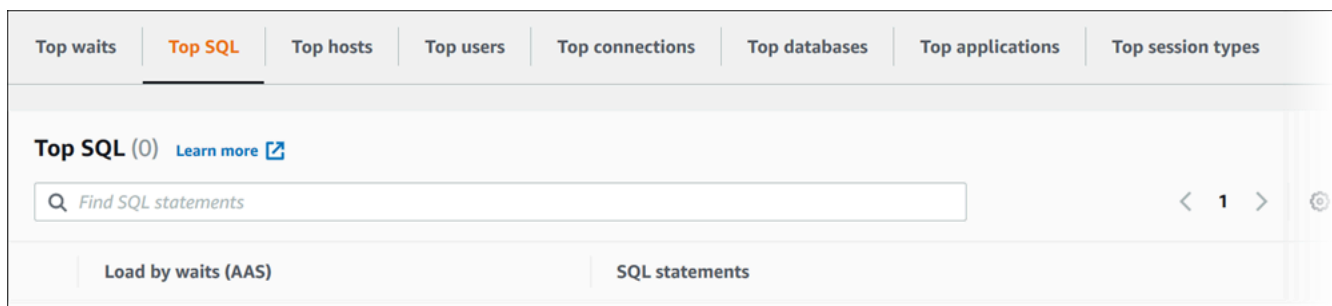


若要以圖例查看在所選時段內任何項目的詳細資訊，請將游標移到該項目上。



最高維度表格

最高維度表格依不同維度切割資料庫負載。維度是資料庫負載各種特性的類別或「配量依據」。如果維度是 SQL，Top SQL (最高 SQL) 會顯示在資料庫負載中佔最大比例的 SQL 陳述式。



選擇下列任一維度索引標籤：

Tab	描述	支援的引擎
最高 SQL	目前正在執行的 SQL 陳述式	全部
最高等待	資料庫後端正在等待的事件	全部
最高主機	連線用戶端的主機名稱	全部
最高使用者	登入資料庫的使用者	全部
最高資料庫	用戶端連線的資料庫名稱	PostgreSQL、MySQL、MariaDB 和 SQL 伺服器
最高應用程式	連線至資料庫的應用程式名稱	僅限 A@@ 和 SQL 伺服器
最高工作階段類型	目前工作階段的類型	僅限 PostgreSQL

若要了解如何使用 Top SQL (最高 SQL) 索引標籤來分析查詢，請參閱 [最高 SQL 索引標籤概觀](#)。

存取績效詳情儀表板

Amazon RDS 會在績效詳情儀表板中提供績效詳情和 CloudWatch 指標的合併檢視。

若要存取績效詳情儀表板，請使用下列程序。

檢閱 AWS 管理主控台中的績效詳情儀表板

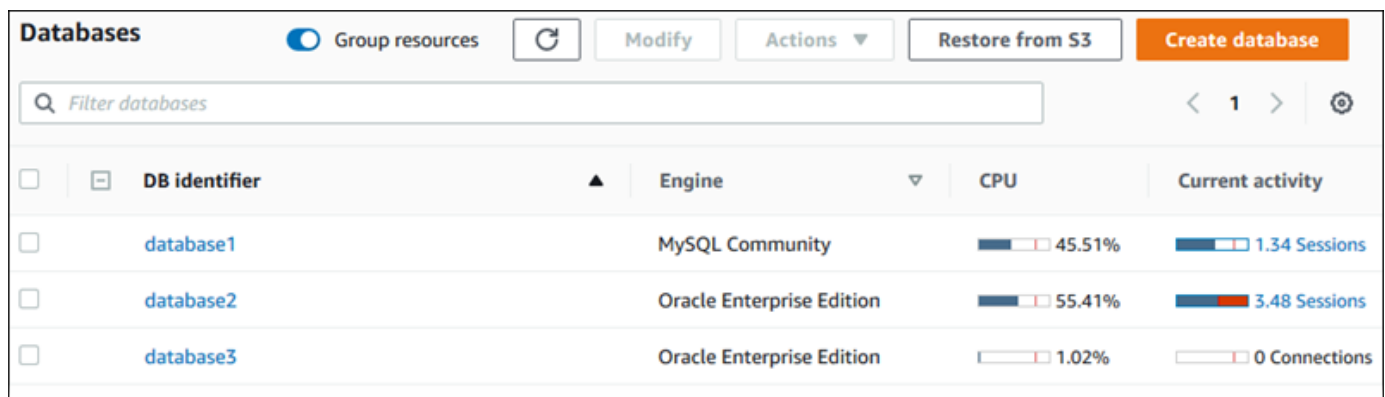
1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。
4. 在顯示的視窗中選擇預設監控檢視。
 - 選取績效詳情和 CloudWatch 指標檢視 (新增) 選項，然後選擇繼續以檢視績效詳情和 CloudWatch 指標。
 - 選取績效詳情檢視選項，並針對舊版監控檢視選擇繼續。然後，繼續執行此程序。

Note

此檢視將於 2023 年 12 月 15 日停止使用。

資料庫執行個體的績效詳情儀表板即會出現。

對於已開啟績效詳情的資料庫執行個體，您也可以在此資料庫執行個體清單中選擇工作階段項目，來搜尋儀表板。在 Current activity (目前活動) 中，Sessions (工作階段) 項目顯示了過去五分鐘內平均作用中工作階段的資料庫負載。負載以進度條圖形的方式顯示。當進度條為空時，代表資料庫執行個體閒置中。隨著負載增加，進度條會填入藍色。當負載超過資料庫執行個體類別上的虛擬 CPU (vCPU) 數量時，進度條會轉紅，指出可能遇到瓶頸。



<input type="checkbox"/>	<input type="checkbox"/> DB identifier	<input type="checkbox"/> Engine	<input type="checkbox"/> CPU	<input type="checkbox"/> Current activity
<input type="checkbox"/>	database1	MySQL Community	45.51%	1.34 Sessions
<input type="checkbox"/>	database2	Oracle Enterprise Edition	55.41%	3.48 Sessions
<input type="checkbox"/>	database3	Oracle Enterprise Edition	1.02%	0 Connections

- (選用) 選擇右上角的日期和時間範圍，並指定不同的相對或絕對時間間隔。您現在可以指定時間區間，並產生資料庫績效分析報告。該報告提供識別出的洞見和建議。如需更多詳細資訊，請參閱 [建立績效分析報告](#)。

📅 2023-04-27T10:01:02-07:00 — 2023-04-27T10:19:09-07:00
🔄 🔍

Relative range

Absolute range

Choose a range

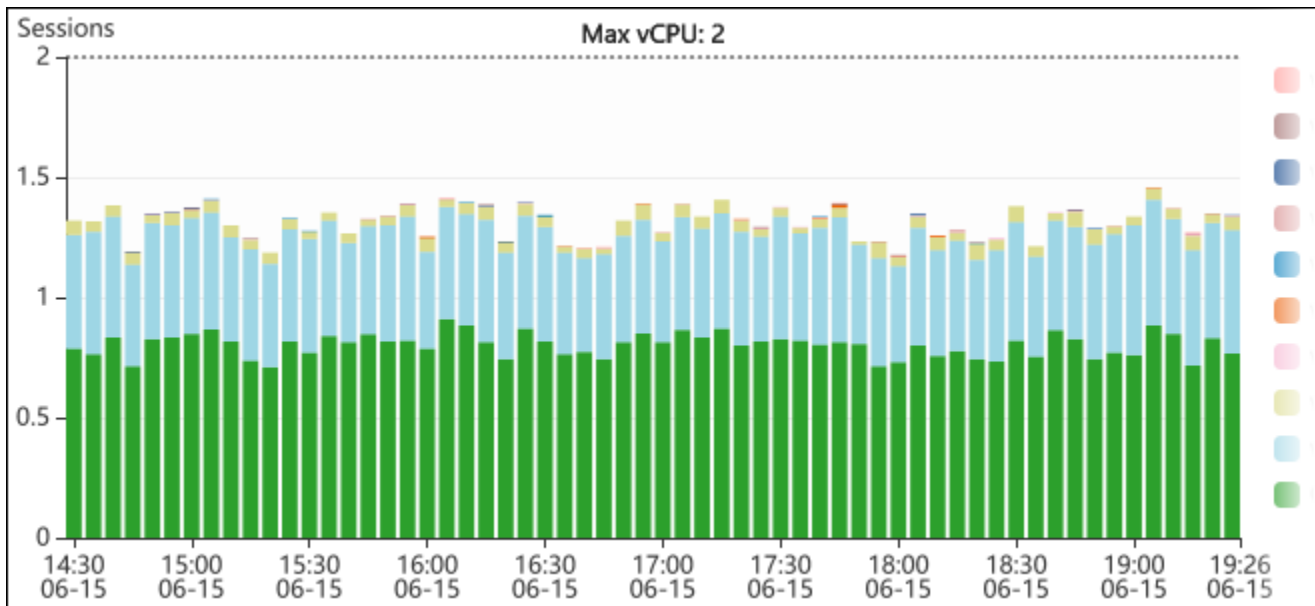
- Last 5 minutes
- Last 1 hour
- Last 5 hours
- Last 24 hours
- Last 1 week
- Custom range

Based on your current retention period, the maximum range is 1 week.
 You can increase the retention period by [modifying your database](#).

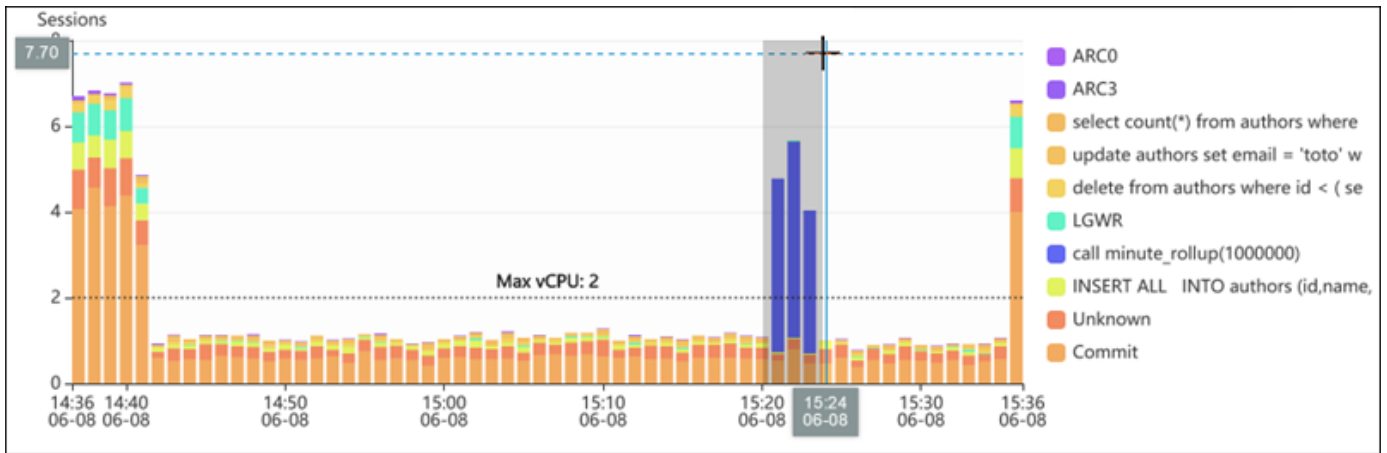
Clear and dismiss
Cancel

Apply

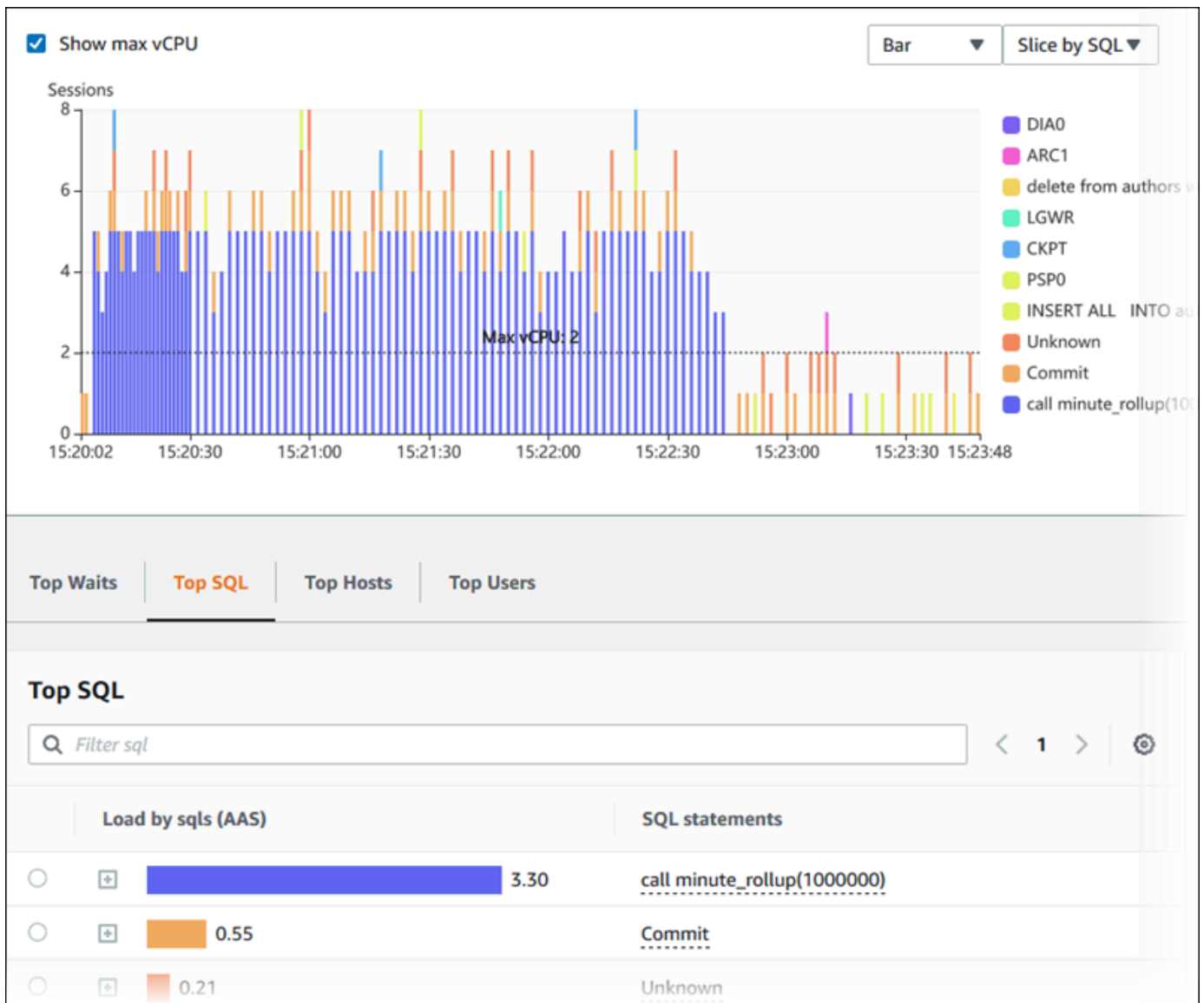
在下列螢幕擷取畫面中，資料庫負載為間隔為 5 小時。



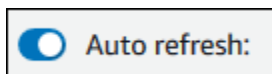
6. (選用) 若要放大一部分的資料庫負載圖表，請選擇開始時間並拖曳到您想要的時間範圍結束時間。
所選區域將在資料庫負載圖表中突出顯示。



放開滑鼠時，資料庫負載圖表會在所選 AWS 區域上放大，而 Top dimensions (最高維度) 資料表則會重新計算。



7. (選用) 若要自動重新整理資料，請選取自動重新整理。



績效詳情儀表板會自動以新資料進行重新整理。重新整理速度取決於顯示的資料量：

- 5 分鐘的資料每 10 秒重新整理一次。
- 1 小時的資料每 5 分鐘重新整理一次。
- 5 小時的資料每 5 分鐘重新整理一次。
- 24 小時的資料每 30 分鐘重新整理一次。
- 1 週的資料每天重新整理一次。

- 1 個月的資料每天重新整理一次。

按等待事件分析資料庫負載

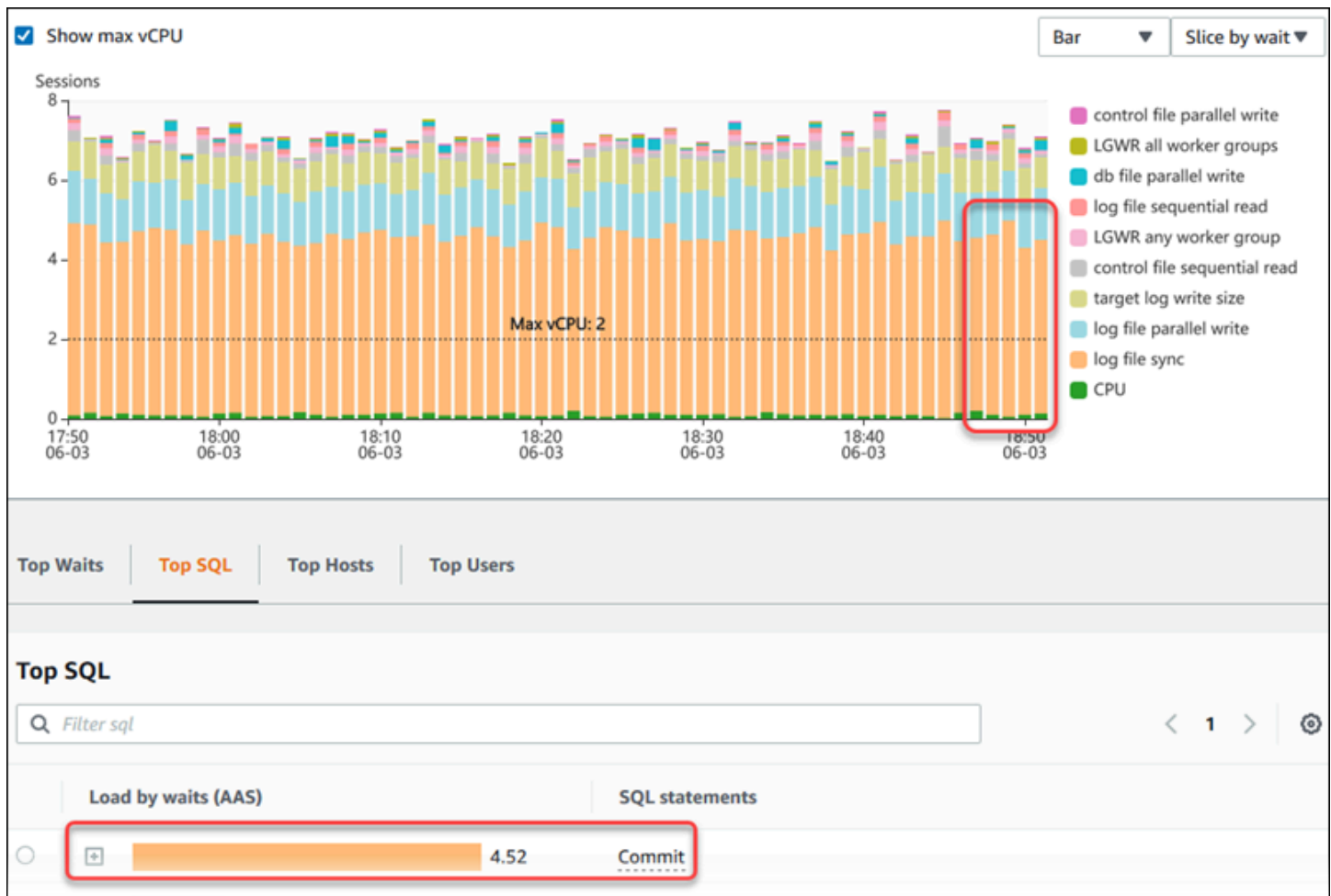
如果 Database load (資料庫負載) 圖表指出有瓶頸，您可以查明負載的來源。若要這麼做，請查看資料庫負載圖表下的最高負載項目表格。選擇特定項目，例如 SQL 查詢或使用者，來深入探討該項目並查看關於該項目的詳細資訊。

依等待和最高 SQL 查詢分組的資料庫負載是預設的績效詳情儀表板檢視。此組合通常可提供效能問題的最多見解。根據等待分組的資料庫負載顯示該資料庫中是否有任何資源或正在發生的瓶頸。在此情況下，最高負載項目表格的 SQL 標籤會顯示哪些查詢帶來這些負載量。

診斷效能問題的典型工作流程如下：

1. 檢閱資料庫負載圖表並查看是否有任何資料庫負載超越最高 CPU 線的情況。
2. 若有，請查看資料庫負載圖表，並找出哪一個或那幾個等待狀態是主因。
3. 利用檢視最高負載項目資料表上的 SQL 標籤之查詢對於那些等待狀態影響較大，藉此找出造成負載的摘要查詢。您可以根據等待列出資料庫負載欄來找出這些。
4. 選擇 SQL 標籤中的其中一個摘要查詢，展開並查看其中組成的子查詢。

例如，在下列儀表板中，日誌檔案同步等待佔了大部分的資料庫負載。LGWR 所有工作者群組等待也很高。最高 SQL 圖表顯示導致日誌檔案同步等待的原因：頻繁的 COMMIT 陳述式。在這種情況下，減少遞交頻率將能降低資料庫負載。



分析一段時間區間的資料庫效能

透過建立一段時間的效能分析報告，透過隨選分析來分析資料庫效能。檢視效能分析報告以找出效能問題，例如資源瓶頸或資料庫執行個體中查詢的變更。Performance Insights 儀表板可讓您選取時間段並建立績效分析報告。您也可以將一或多個標籤新增到報告。

若要使用此功能，您必須使用付費方案的保留期。如需更多資訊，請參閱[Performance Insights 的定價和資料保留](#)

該報告可在績效分析報告 - 新標籤中選取和檢視。該報告包含洞見、相關指標和解決效能問題的建議。您可以在 Performance Insights 保留期間內檢視報告。

如果報告分析期間的開始時間超過保留期，則會刪除報告。您也可以保留期結束之前刪除報告。

若要偵測效能問題並產生資料庫執行個體的分析報告，您必須開啟 Performance Insights。如需開啟績效詳情的詳細資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)。

如需此功能的區域、資料庫引擎和執行個體類別支援的資訊，請參閱 [Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別](#)

建立績效分析報告

您可以在 Performance Insights 儀表板中針對特定區間建立績效分析報告。您可以選取時間區間，並將一或多個標籤新增至分析報告。

分析期間的範圍可以從 5 分鐘到 6 天不等。在分析開始時間之前，必須至少有 24 小時的效能資料。

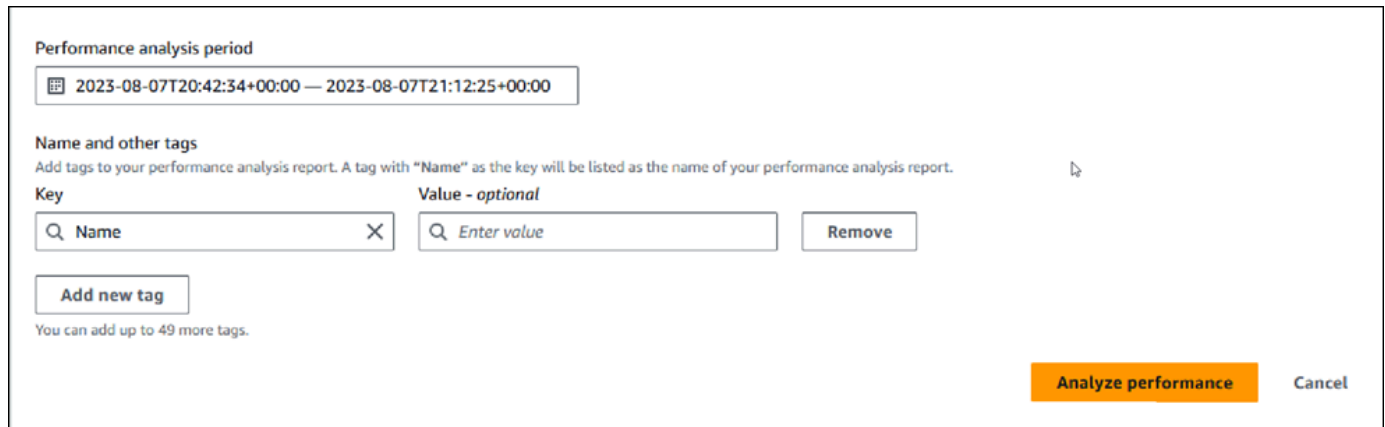
若要建立一段時間區間的績效分析報告

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 在儀表板上的資料庫負載區段選擇分析效能。

設定要顯示的時間區間並新增一或多個標籤至績效分析報告的欄位。



Performance analysis period

2023-08-07T20:42:54+00:00 — 2023-08-07T21:12:25+00:00

Name and other tags

Add tags to your performance analysis report. A tag with "Name" as the key will be listed as the name of your performance analysis report.

Key	Value - optional	
Name	Enter value	Remove

Add new tag

You can add up to 49 more tags.

Analyze performance Cancel

5. 選擇時間區間。如果您在右上角的相對範圍或者絕對範圍內設定時間區間，您只能輸入或選取此時間區間內的報告日期和時間。如果您選取此時間區間以外的分析區間，則會顯示錯誤訊息。

若要設定時間區間，您可以執行下列任一項操作：

- 按下並拖曳資料庫負載圖表上的任何滑桿。

該績效分析區間方塊會顯示所選的時段，而資料庫負載圖表會反白顯示所選取的時間區間。

- 在績效分析區間方塊中選擇開始日期、開始時間、結束日期和結束時間。

Performance analysis period

📅 2023-08-07T21:34:28+00:00 — 2023-08-07T21:36:58+00:00

< August 2023
September 2023 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5						1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23
27	28	29	30	31			24	25	26	27	28	29	30

Start date

Start time

End date

End time

For date, use YYYY/MM/DD. For time, use 24 hr format.

Clear and dismiss
Cancel
Apply

6. (選用) 輸入索引鍵和值 - 選用，以新增報告的標籤。

Name and other tags

Add tags to your performance analysis report. A tag with "Name" as the key will be listed as the name of your performance analysis report.

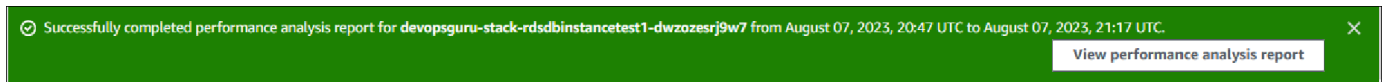
Key	Value - optional	
🔍 Name ×	🔍 Enter value	Remove
<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;">Add new tag</div>		

You can add up to 49 more tags.

7. 選擇分析效能。

橫幅會顯示報告產生成功或失敗的訊息。該訊息還會提供檢視報告的連結。

下列範例顯示成功建立報告的訊息橫幅。



該報告可以於績效分析報告 - 新標籤中檢視。

您可以使用 AWS CLI 建立效能分析報告。如需如何使用建立報表的範例 AWS CLI，請參閱 [建立一段時間區間的績效分析報告](#)。

檢視績效分析報告

該績效分析報告 - 新標籤會列出為資料庫執行個體建立的所有報告。為每個報告顯示以下內容：

- ID：報告的唯一識別碼。
- 名稱：新增至報告的標籤索引鍵。
- 報告建立時間：您建立報告的時間。
- 分析開始時間：報告中分析的開始時間。
- 分析結束時間：報告中分析的結束時間。

檢視績效分析報告

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇您要檢視其分析報告的資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動並選擇績效分析報告 - 新標籤。

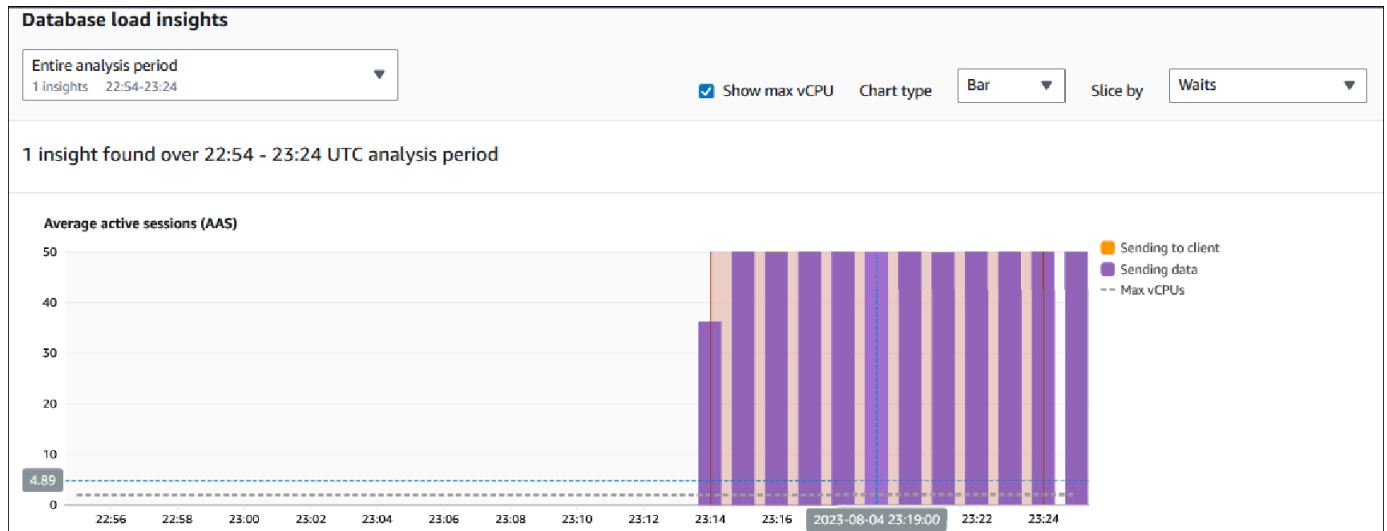
會顯示不同時間區間的所有分析報告。

5. 選擇您要檢視報告的 ID。

如果識別出多個洞見，資料庫負載圖表預設會顯示整個分析區間。如果報告識別出一個洞見，則資料庫負載圖表會依預設顯示洞見。

儀表板也會列出報告標籤區段的標籤。

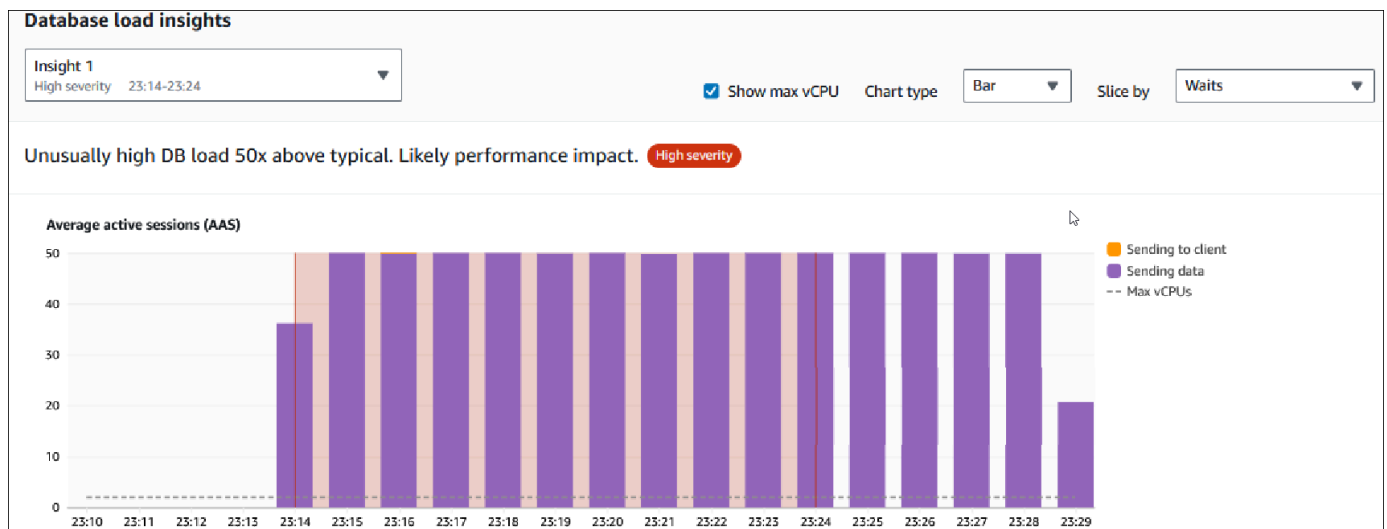
下列範例顯示報告的整個分析區間。



6. 如果報告中識別出一個以上的洞見，請選擇資料庫負載洞見清單中您要檢視的洞見。

儀表板會顯示洞見訊息、資料庫負載圖表會反白顯示洞見的時間區間、分析和建議，以及報告標籤的清單。

下列範例會顯示報告中的資料庫負載洞見。



▼ Analysis and Recommendations			
Detection	Analysis	Recommendations	Related Metrics
Performance schema	<p>The average active sessions (AAS) exceeded 50. The MySQL Performance Schema isn't enabled.</p> <p>Why is this a problem? -----</p>	<p>Investigate the following SQL digest IDs: 30217D9B9D80A045D9405DA58ED139DDBDC03AB -----</p> <p>View Top SQL in Performance Insights </p> <p>Also, consider enabling the MySQL Performance Schema.</p> <p>Why do we recommend this? -----</p>	Load (db.load.avg)

將標籤新增至績效分析報告

您可以在建立或檢視報告時新增標籤。您最多可以為報告新增 50 個標籤。

您需要許可以新增標籤。如需 Performance Insights 存取政策的詳細資訊，請參閱 [設定績效詳情的存取政策](#)。

若要在建立報告時新增一個或多個標籤，請參閱程序 [建立績效分析報告](#) 中的步驟 6。

在檢視報告時新增一個或多個標籤

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動並選擇績效分析報告 - 新標籤。
5. 選擇您要新增標籤的報告。

儀表板會顯示報告。

6. 向下捲動至標籤並選擇管理標籤。
7. 選擇 Add new tag (新增標籤)。
8. 輸入索引鍵和值 - 選用，然後選擇新增標籤。

下列範例提供為所選報告新增標籤的選項。

Manage tags

Tags

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="test"/> <input type="button" value="Remove"/>
<input type="text" value="Enter key"/> Custom tag key	<input type="text" value="Enter value"/> <input type="button" value="Remove"/>

You can add up to 48 more tags.

將為報告建立一個新標籤。

報告的標籤清單會顯示在儀表板上的標籤區段。如果您要從報告中移除標籤，請選擇標籤旁邊的移除。

刪除績效分析報告

您可以從績效分析報告標籤所顯示的報告清單中刪除報告，或是檢視報告時刪除報告。

刪除報告

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇資料庫執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

4. 向下捲動並選擇績效分析報告 - 新標籤。
5. 選取您要刪除的報告，然後選擇右上角的刪除。

Dimensions		Metrics - new		Performance analysis reports - new			
Performance analysis reports (7)						Feedback	Delete
<input type="text" value="Search"/>						< 1 >	
ID	Name	Status	Report creation time	Analysis start time	Analysis end time		
<input checked="" type="radio"/>	report-0d70bd664b712a171	Completed	August 07, 2023, 21:33 UTC	August 07, 2023, 20:47 UTC	August 07, 2023, 21:17 UTC		
<input type="radio"/>	report-06849e77acb402302	Completed	August 04, 2023, 23:32 UTC	August 04, 2023, 22:54 UTC	August 04, 2023, 23:24 UTC		

此時會顯示確認視窗。選擇確認後，即會刪除報告。

6. (選用) 選擇您要刪除報告的 ID。

在報告頁面，選擇右上角的刪除。

此時會顯示確認視窗。選擇確認後，即會刪除報告。

在績效詳情儀表中分析查詢

在 Amazon RDS 績效詳情儀表中，您可以在 Top dimensions (最高維度) 表格的 Top SQL (最高 SQL) 索引標籤中找到執行中和近期查詢的相關資訊。您可以使用此資訊來調校查詢。

主題

- [最高 SQL 索引標籤概觀](#)
- [在績效詳情儀表中存取更多 SQL 文字](#)
- [在績效詳情儀表中檢視 SQL 統計數字](#)

最高 SQL 索引標籤概觀

依預設，Top SQL (最高 SQL) 索引標籤會顯示在資料庫負載中佔最大比例的 25 個查詢。為了協助調校查詢，您可以分析查詢文字和 SQL 統計資料等資訊。您還可以選擇要出現在 Top SQL (最高 SQL) 索引標籤中的統計數字。

主題

- [SQL 文字](#)
- [SQL Statistics](#)
- [依等待分組的負載 \(AAS\)](#)
- [SQL 資訊](#)

- [Preferences \(偏好設定\)](#)

SQL 文字

根據預設，Top SQL (最高 SQL) 資料表中的每個資料列都會顯示每個 SQL 陳述式的 500 位元組文字。




Load by waits (AAS)	SQL statements
2.00	SELECT SEAT_LEVEL, SEAT_SECTION, SEAT_ROW FROM (SELECT SEAT_LEVEL, SEAT_SECTION, S...
1.71	select p.full_name, SUM(t.id) from ticket_purchase_hist h, person p, sporting_e...
1.17	SELECT MIN(SPORTING_EVENT_TICKET_ID), MAX(SPORTING_EVENT_TICKET_ID) FROM TICKET_...
0.54	SELECT MAX(SPORTING_EVENT_TICKET_ID) FROM TICKET_PURCHASE_HIST WHERE SPORTING_EV...
0.15	DECLARE SqlDevBind1Z_1 VARCHAR2(32767):=:SqlDevBind1ZInit1; SqlDevBind1Z_2 VARCH...
0.11	SELECT SUM(PURCHASE_PRICE) FROM TICKET_PURCHASE_HIST
0.08	UPDATE SPORTING_EVENT_TICKET SET TICKETHOLDER_ID = :B2 WHERE ID = :B1
0.04	SELECT * FROM SPORTING_EVENT_TICKET WHERE SPORTING_EVENT_ID = :B4 AND SEAT_LEVEL...

若要了解如何查看超過預設 500 位元組的 SQL 文字，請參閱 [在績效詳情儀表板中存取更多 SQL 文字](#)。

SQL 摘要綜合顯示結構相似但可能有不同常值的多個實際查詢。摘要中，問號會取代硬式編碼值。例如，摘要可能是 `SELECT * FROM emp WHERE lname = ?`。此摘要可能包含下列子查詢：

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

若要逐句查看摘要中的 SQL 陳述式，請選取查詢，然後選擇加號 (+)。在下列範例中，選取的查詢是摘要。

Load by waits (AAS)		SQL statements
<input checked="" type="radio"/>	 0.88	<u>select minute_rollups(?)</u>
<input type="radio"/>	 0.50	<u>select minute_rollups(1000000)</u>
<input type="radio"/>	 0.53	<u>select count(*) from authors where ic</u>

Note

SQL 摘要會將類似的 SQL 陳述式分組，但不會修訂敏感資訊。







績效詳情可以將 Oracle SQL 文字顯示為 Unknown (不明)。在以下情況中，文字具有此狀態：

- SYS 以外的 Oracle 資料庫使用者處於作用中狀態，但目前未執行 SQL。例如，當平行查詢完成時，查詢協調器會等待協助程式程序傳送工作階段統計數字。在等待期間，查詢文字顯示會顯示 Unknown (不明)。
- 對於標準版 2 上的 RDS for Oracle 執行個體，Oracle Resource Manager 會限制平行執行緒的數量。執行此工作的背景處理程序會導致查詢文字顯示為 Unknown (不明)。

SQL Statistics

SQL 統計數字是 SQL 查詢的效能相關指標。例如，績效詳情可能顯示每秒的執行次數或每秒處理的資料列數。績效詳情只收集最常用查詢的統計數字。這些通常會符合績效詳情儀表中依負載顯示的熱門查詢。

Top SQL (最高 SQL) 表格的每一行顯示 SQL 陳述式或摘要的相關統計數字，如下列範例所示。

Top SQL				
Filter sql				
	Load by waits (AAS)	SQL statements	calls/sec	rows/sec
<input type="radio"/>	 0.88	<code>select minute_rollups(?)</code>	0.06	0.06
<input type="radio"/>	 0.53	<code>select count(*) from authors where id < (select max(id) - 31 from authors) and...</code>	33.68	101.04
<input type="radio"/>	 0.17	<code>WITH cte AS (SELECT id FROM authors LIMIT ?) UPDATE ...</code>	33.68	33.68
<input type="radio"/>	 0.08	<code>delete from authors where id < (select * from (select max(id) - ? from authors...</code>	33.68	303.13
<input type="radio"/>	 0.07	<code>INSERT INTO authors (id,name,email) VALUES (nextval(?) ,?), (nextval(?) ,?...</code>	33.68	303.13
<input type="radio"/>	 0.06	<code>select count(*) from authors where id < (select max(id) - 31 from authors) and...</code>	0.00	0.00

績效詳情可將 SQL 統計數字報告為 0.00 和 - (不明)。在下列情況下會發生此情況：

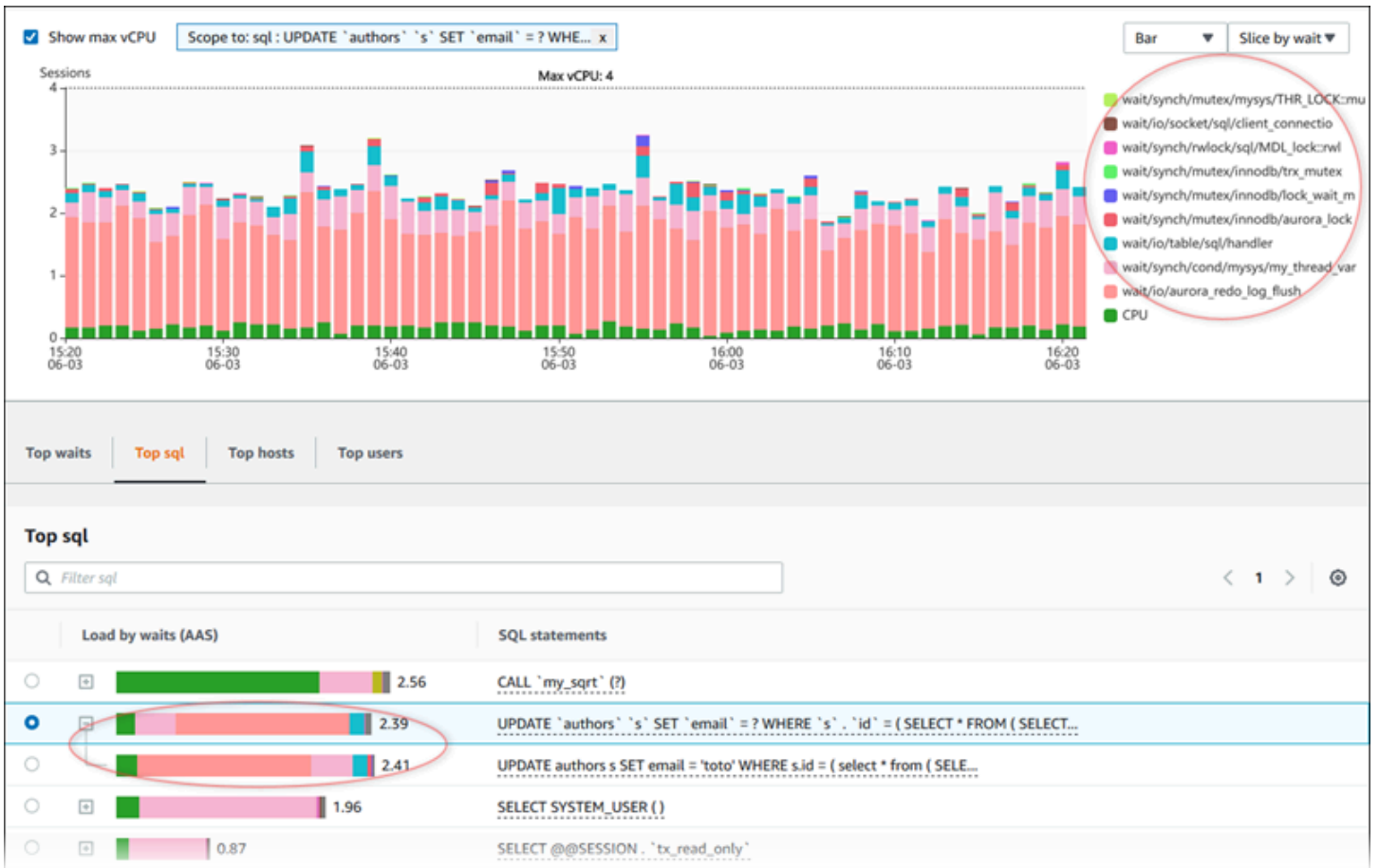
- 只有一個樣本存在。例如，績效詳情會根據 `pg_stat_statements` 檢視中的多個樣本來計算 RDS PostgreSQL 查詢。當工作負載執行時間很短時，績效詳情可能只收集到一個樣本，這代表它無法計算變化速率。不明值用破折號 (-) 表示。
- 兩個樣本具有相同的值。績效詳情無法計算變化速率，因為沒有發生變化，因此它將速率報告為 0.00。
- RDS PostgreSQL 陳述式缺少有效識別符。只有在解析和分析之後，PostgreSQL 才會為陳述式建立識別符。因此，陳述式可以存在於 PostgreSQL 內部記憶體中結構，而沒有識別符。由於績效詳情每秒對內部記憶體中結構採樣一次，所以低延遲查詢可能只顯示一個樣本。如果查詢識別符無法用於此樣本，則績效詳情無法將此陳述式與其統計數字產生關聯。不明值用破折號 (-) 表示。

如需 Amazon RDS 引擎之 SQL 統計數字的說明，請參閱[績效詳情的 SQL 統計數字](#)。

依等待分組的負載 (AAS)

在最高 SQL 中，根據等待的負載 (AAS) 欄說明了與每個最高負載項目相關聯的資料庫負載百分比。此欄根據目前在資料庫負載圖表中所選的群組依據來反映出該項目的負載。如需平均作用中工作階段 (AAS) 的詳細資訊，請參閱[平均作用中工作階段](#)。

例如，您可以依等待狀態將 DB load (資料庫負載) 圖表分組。您可檢查最高負載項目資料表中的 SQL 查詢。在此情況下，根據等待列出資料庫負載列較大且分段，並以顏色為代碼來顯示該查詢所帶來的指定等待狀態程度。這也會顯示哪些等待狀態會影響選取的查詢。



SQL 資訊

在最高 SQL 資料表中，您可以開啟陳述式以檢視其資訊。資訊會顯示在底部窗格中。

Load by waits (AAS)		SQL statements
<input type="radio"/>	0.88	<code>select minute_rollups(?)</code>
<input type="radio"/>	0.55	<code>select count(*) from authors where id < (select max(id) - 31 from al</code>
<input checked="" type="radio"/>	0.45	<code>select count(*) from authors where id < (select max(id) - 31 from al</code>
<input type="radio"/>	0.37	<code>INSERT INTO authors (id,name,email) VALUES (nextval(?,?),?)</code>
<input type="radio"/>	0.16	<code>WITH cte AS (SELECT id FROM authors LIMIT ?) UPDATE ...</code>
<input type="radio"/>	0.09	<code>delete from authors where id < (select * from (select max(id) - ? fro</code>
<input type="radio"/>	0.07	<code>INSERT INTO authors (id,name,email) VALUES (nextval(?,?), (ne</code>
<input type="radio"/>	0.06	<code>select count(*) from authors where id < (select max(id) - 31 from al</code>
<input type="radio"/>	0.02	<code>select minute_rollups(?)</code>
<input type="radio"/>	< 0.01	<code>autovacuum: ANALYZE public.authors</code>
<input type="radio"/>	< 0.01	<code>autovacuum: VACUUM public.authors</code>

SQL information

This SQL statement is truncated to the first 500 characters. To view the full SQL statement, choose **Download**.

```
select count(*) from authors where id < ( select max(id) - 31 from authors) and id > ( select max(id) - 2500 from authors) union
select count(*) from authors where id < ( select max(id) - 31 from authors) and id > ( select max(id) - 1500 from authors) union
select count(*) from authors where id < ( select max(id) - 31 from authors) and id > ( select max(id) - 1500 from authors) union
select count(*) from authors where id < ( select max(id) - 31 from authors) and id > ( select max(id) - 1
```

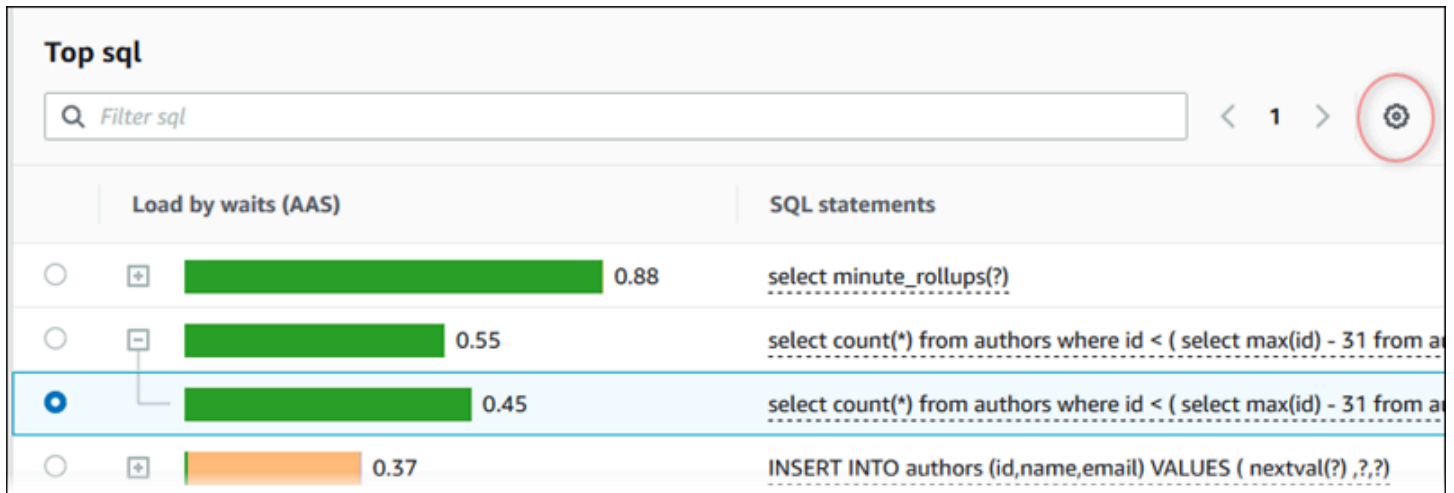
SQL ID: pi-135048318 ([Support SQL ID](#)) Digest ID: 1325689244 ([Support Digest ID](#))

以下幾種與 SQL 陳述式相關聯的識別符 (ID)：

- 支援 SQL ID – SQL ID 的雜湊值。當您使用 Sup AWS port 時，此值僅用於參考 SQL ID。AWS Support 無法存取您實際的 SQL ID 和 SQL 文字。
- 支援摘要 ID – 摘要 ID 的雜湊值。當您使用「Sup AWS port」時，此值僅用於參考摘要 ID。AWS Support 無法存取您的實際摘要 ID 和 SQL 文字。

Preferences (偏好設定)

您可以選擇 Preferences (偏好設定) 圖示，以控制 Top SQL (最高 SQL) 索引標籤中顯示的統計資料。



The screenshot shows the 'Top sql' interface. At the top, there is a search bar labeled 'Filter sql' and a navigation area with '< 1 >' and a settings gear icon circled in red. Below this is a table with two columns: 'Load by waits (AAS)' and 'SQL statements'. The table contains four rows of data, with the third row highlighted in blue. The first row has a value of 0.88 and a SQL statement 'select minute_rollups(?)'. The second row has a value of 0.55 and a SQL statement 'select count(*) from authors where id < (select max(id) - 31 from a'. The third row has a value of 0.45 and a SQL statement 'select count(*) from authors where id < (select max(id) - 31 from a'. The fourth row has a value of 0.37 and a SQL statement 'INSERT INTO authors (id,name,email) VALUES (nextval(?) ,?,?)'.

	Load by waits (AAS)	SQL statements
<input type="radio"/>	0.88	<code>select minute_rollups(?)</code>
<input type="radio"/>	0.55	<code>select count(*) from authors where id < (select max(id) - 31 from a</code>
<input checked="" type="radio"/>	0.45	<code>select count(*) from authors where id < (select max(id) - 31 from a</code>
<input type="radio"/>	0.37	<code>INSERT INTO authors (id,name,email) VALUES (nextval(?) ,?,?)</code>

當您選擇 Preferences (偏好設定) 圖示時，Preferences (偏好設定) 視窗會開啟。下列螢幕擷取畫面是 Preferences (偏好設定) 視窗的範例。

Preferences ✕

Page size

All resources

Wrap lines
Check to see all the text and wrap the lines

Columns

Load by waits (AAS)	<input checked="" type="checkbox"/>
SQL statements	<input checked="" type="checkbox"/>
calls/sec (calls_per_sec)	<input checked="" type="checkbox"/>
rows/sec (rows_per_sec)	<input checked="" type="checkbox"/>
AAE (total_time_per_sec)	<input type="checkbox"/>
blk hits/sec (shared_blks_hit_per_sec)	<input type="checkbox"/>
blk reads/sec (shared_blks_read_per_sec)	<input type="checkbox"/>
blk dirty/sec (shared_blks_dirtied_per_sec)	<input type="checkbox"/>
blk writes/sec (shared_blks_written_per_sec)	<input type="checkbox"/>
local blk hits/sec (local_blks_hit_per_sec)	<input type="checkbox"/>
local blk reads/sec (local_blks_read_per_sec)	<input type="checkbox"/>
local blk dirty/sec (local_blks_dirtied_per_sec)	<input type="checkbox"/>

若要讓您要的統計數字出現在 Top SQL (最高 SQL) 索引標籤中，請使用滑鼠捲動到視窗底部，然後選擇 Continue (繼續)。

如需 Amazon RDS 引擎每秒或每次呼叫統計資料的詳細資訊，請參閱 [績效詳情的 SQL 統計數字](#) 中的引擎特定 SQL 統計資料一節

在績效詳情儀表中存取更多 SQL 文字

根據預設，最高 SQL 資料表中的每個列會顯示每個 SQL 陳述式的 500 位元組的 SQL 文字。



SQL 陳述式超過 500 位元組時，您可以在 Top SQL 表格下方的 SQL text (SQL 文字) 區段檢視更多文字。在此情況下，SQL text (SQL 文字) 所顯示查詢的最大長度為 4 KB。此限制是由主控台引入，並受限於資料庫引擎設定的限制。若要儲存 SQL text (SQL 文字) 中顯示的文字，請選擇 Download (下載)。

主題

- [Amazon RDS 引擎 的文字大小限制](#)
- [為 Amazon RDS for PostgreSQL 資料庫執行個體設定 SQL 文字限制](#)
- [在 Performance Insights 儀表板中檢視和下載更多 SQL 文字](#)

Amazon RDS 引擎 的文字大小限制

當您下載 SQL 陳述式時，資料庫引擎會決定文字的最大長度。您可以下載的文字最大限制為以下每個引擎的限制。

資料庫引擎	下載文字長度上限
Amazon RDS for MySQL 和 MariaDB	1,024 位元組
Amazon RDS for Microsoft SQL Server	4,096 個字元
Amazon RDS for Oracle	1,000 位元組

Performance Insights 主控台的 SQL text (SQL 文字) 區段顯示上限為引擎傳回的最大值。例如，如果 MySQL 最多傳回 1 KB 內容到 Performance Insights，則其僅能收集並顯示 1 KB 內容，即使原始查詢較大也如此。因此，當您查看 SQL text (SQL 文字) 或下載查詢時，Performance Insights 會傳回相同的位元組數量。

如果您使用 AWS CLI 或 API，Performance Insights 就沒有主控台強制執行的 4 KB 限制。DescribeDimensionKeys 並 GetResourceMetrics 返回最多 500 個字節。

Note

GetDimensionKeyDetails 返回完整的查詢，但大小受引擎限制。

為 Amazon RDS for PostgreSQL 資料庫執行個體設定 SQL 文字限制

Amazon RDS for PostgreSQL 處理文字的方式不同。您可以使用資料庫執行個體參數 `track_activity_query_size` 設定文字大小限制。此參數具有下列特性：

預設文字大小

在 Amazon RDS for PostgreSQL 9.6 版中，`track_activity_query_size` 參數的預設設定為 1,024 位元組。在 Amazon RDS for PostgreSQL 10 版中，其預設設定為 4,096 位元組。

文字大小上限

Amazon RDS for PostgreSQL 第 12 版及較低版本中，`track_activity_query_size` 的限制為 102,400 位元組。第 13 版及更高版本適用的最大值為 1 MB。

如果引擎傳回 1 MB 內容到 Performance Insights，則主控台只會顯示前 4 KB 的內容。如果您下載查詢，則會取得完整的 1 MB 內容。在此情況下，檢視和下載會傳回不同的位元組數。如需 `track_activity_query_size` 資料庫執行個體參數的更多相關資訊，請參閱 PostgreSQL 文件中的 [執行時間統計資料](#)。

若要增加 SQL 文字大小，請提高 `track_activity_query_size` 限制。若要修改參數，請在與 Amazon RDS for PostgreSQL 資料庫執行個體關聯的參數群組中變更參數設定。

在執行個體使用預設參數群組時變更設定

1. 為適當的資料庫引擎和資料庫引擎版本建立新的資料庫執行個體參數群組。
2. 在新的參數群組中設定參數。
3. 將新的參數群組與資料庫執行個體建立關聯。

如需設定資料庫執行個體參數的相關資訊，請參閱 [修改資料庫參數群組中的參數](#)。

在 Performance Insights 儀表板中檢視和下載更多 SQL 文字

在 Performance Insights 儀表板中，您可以檢視和下載更多 SQL 文字。

在績效詳情儀表板內檢視更多 SQL 文字

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Performance Insights (績效詳情)。

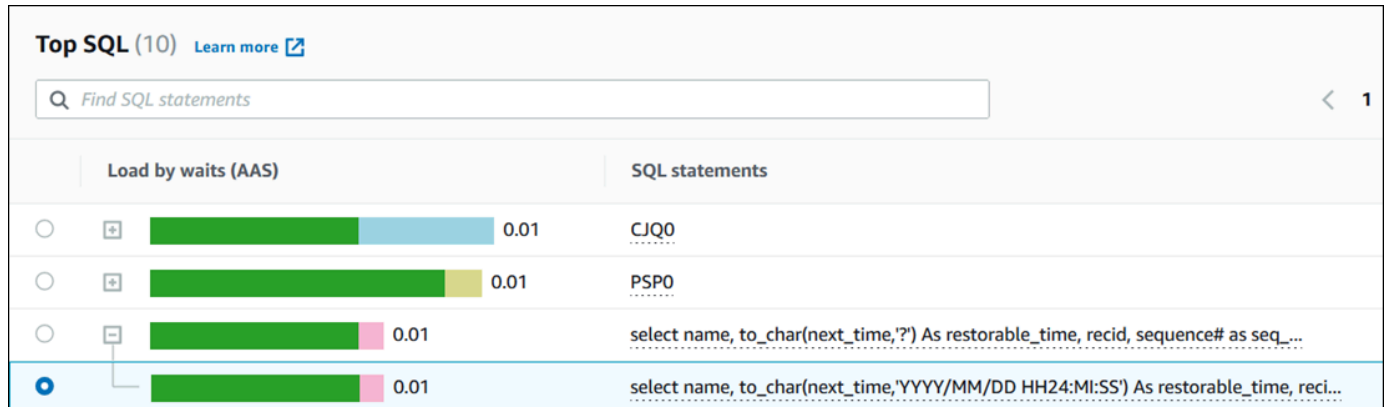
3. 選擇資料庫執行個體。

此時會顯示資料庫執行個體的 Performance Insights 儀表板。

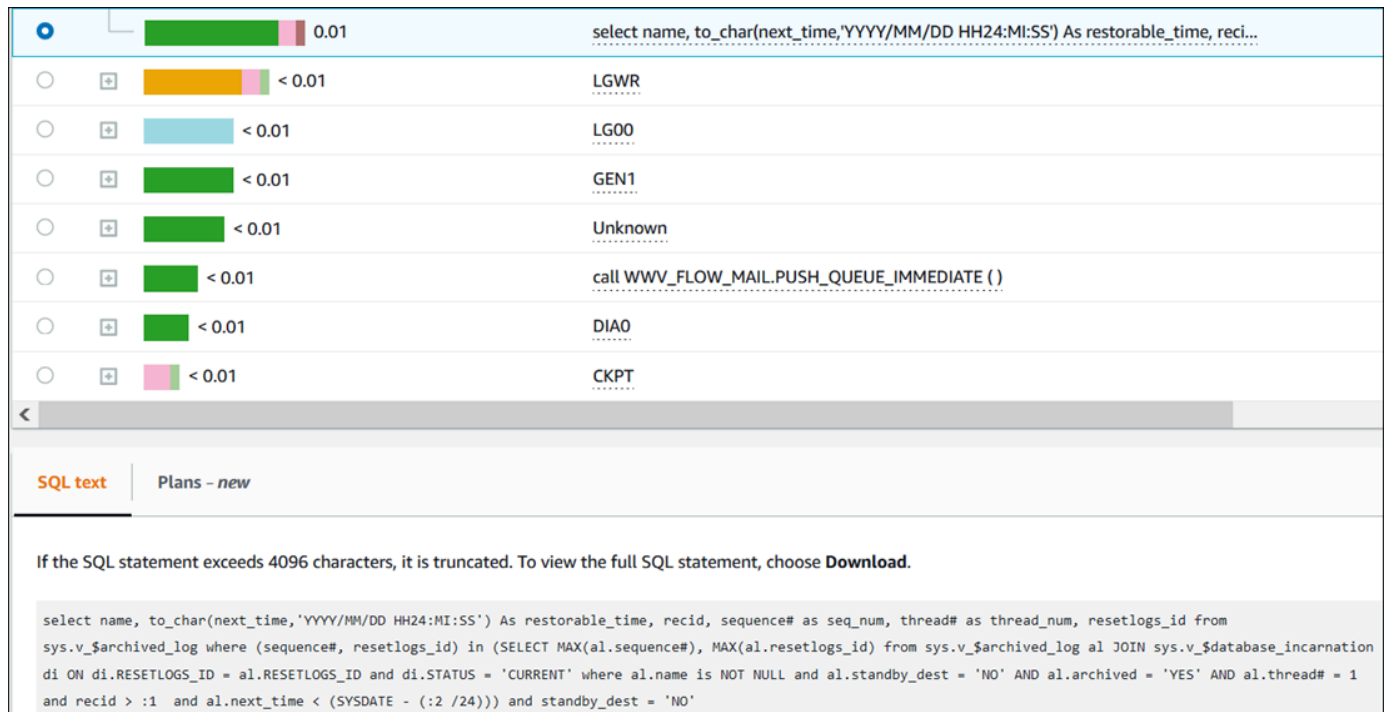
4. 向下捲動至 Top SQL (最高 SQL) 索引標籤。

5. 選擇加號以展開 SQL 摘要，然後選擇摘要的其中一個子查詢。

內含大於 500 位元組文字的 SQL 陳述式看起來與以下影像類似。



6. 向下捲動至 SQL text (SQL 文字) 索引標籤。



績效詳情儀表板可以為每個 SQL 陳述式顯示高達 4,096 位元組。

7. (選用) 選擇複製來複製顯示的 SQL 陳述式，或選擇下載來下載 SQL 陳述式，檢視達到資料庫引擎限制的 SQL 文字。

Note

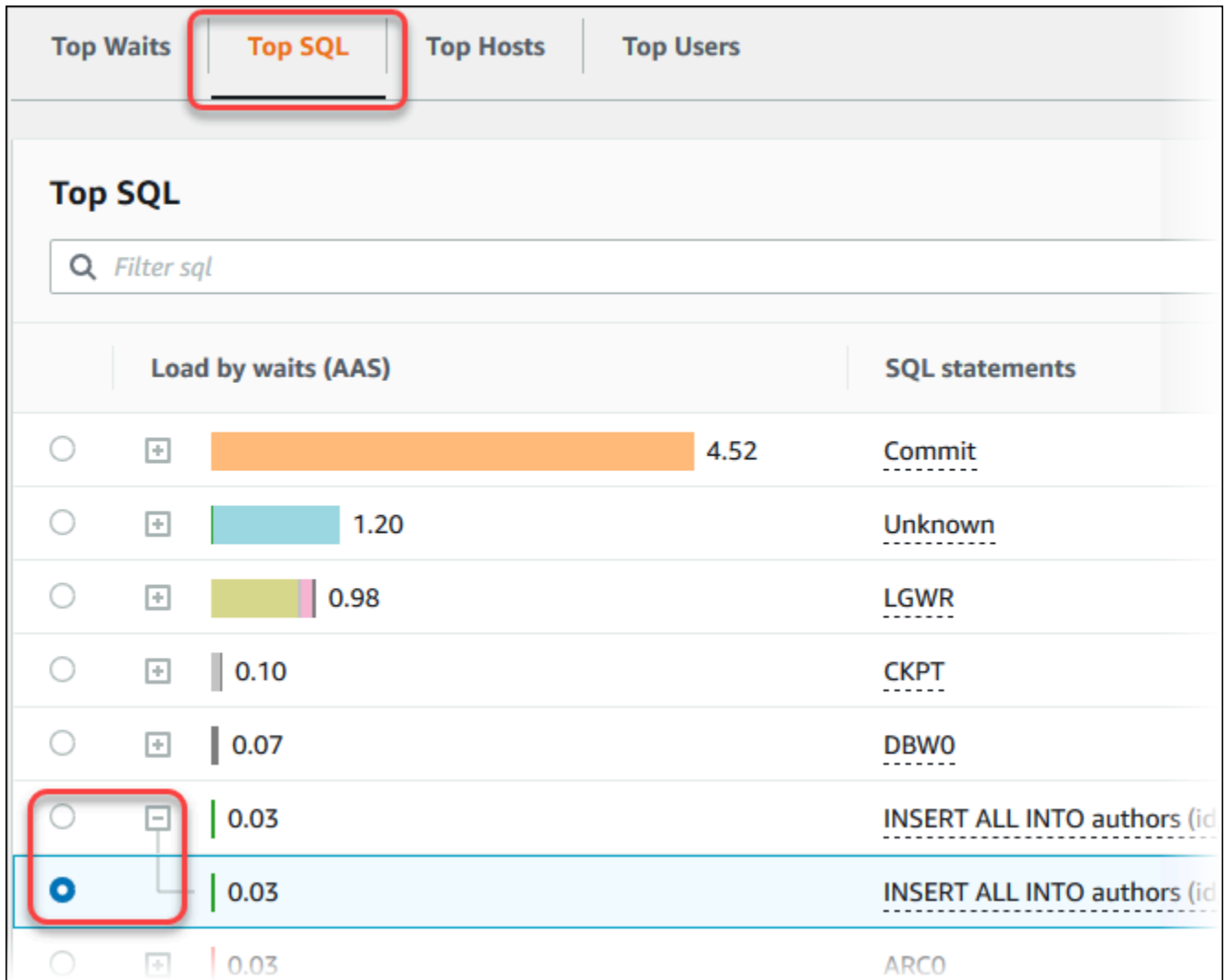
若要複製或下載 SQL 陳述式，停用彈出式封鎖程式。

在績效詳情儀表中檢視 SQL 統計數字

在績效詳情儀表中，Database load (資料庫負載) 圖表的 Top SQL (最高 SQL) 索引標籤會提供 SQL 統計數字。

檢視 SQL 統計數字

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在左側導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 在頁面頂端，選擇您想要查看其 SQL 統計數字的資料庫。
4. 捲動到頁面底部並選擇 Top SQL (最高 SQL)。
5. 選擇個別陳述 或摘要查詢。



6. 選擇圖表右上角的齒輪圖示，以選擇要顯示的統計資料。如需 Amazon RDS 引擎之 SQL 統計數字的說明，請參閱 [績效詳情的 SQL 統計數字](#)。

下列範例顯示 Oracle 資料庫執行個體的統計數字偏好設定。

Preferences ✕

Page size

All resources

Wrap lines
Check to see all the text and wrap the lines

Columns

Load by waits (AAS)	<input checked="" type="checkbox"/>
SQL statements	<input checked="" type="checkbox"/>
Support ID	<input type="checkbox"/>
ID	<input type="checkbox"/>
executions/sec (executions_per_sec)	<input checked="" type="checkbox"/>
AAE (elapsed_time_per_sec)	<input type="checkbox"/>
rows processed/sec (rows_processed_per_sec)	<input type="checkbox"/>
buffer gets/sec (buffer_gets_per_sec)	<input type="checkbox"/>
physical reads/sec (physical_read_requests_per_sec)	<input type="checkbox"/>
physical writes/sec (physical_write_requests_per_sec)	<input type="checkbox"/>
total shareable memory (bytes)/sec (total_sharable_mem_per_sec)	<input type="checkbox"/>

下列範例顯示 MariaDB 與 MySQL 資料庫執行個體的偏好設定。

Preferences ✕

Page size

All resources

Wrap lines
Check to see all the text and wrap the lines

Columns

Load by waits (AAS)	<input checked="" type="checkbox"/>
SQL statements	<input checked="" type="checkbox"/>
Support ID	<input type="checkbox"/>
ID	<input type="checkbox"/>
calls/sec (count_star_per_sec)	<input type="checkbox"/>
AAE (sum_timer_wait_per_sec)	<input type="checkbox"/>
select full join/sec (sum_select_full_join_per_sec)	<input type="checkbox"/>
select range check/sec (sum_select_range_check_per_sec)	<input type="checkbox"/>

7. 選擇 Save (儲存) 儲存偏好設定。

Top SQL (最高 SQL) 表格會重新整理。

以下範例顯示 Oracle SQL 查詢的統計數字。

SQL statements	executions/sec	elapsed time (ms)
Commit	-	-
Unknown	-	-
LGWR	-	-
CKPT	-	-
DBWO	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval , 'Priya','p@g...	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval , 'Priya','p@g...	73.38	0.56
ARCO	-	-

分析最上層 Oracle PDB 負載

分析 Oracle 容器資料庫 (CDB) 上的負載時，您可能想要確定哪些可插拔資料庫 (PDB) 對資料庫載入的貢獻最大。您可能還想要比較執行類似查詢的個別 PDB 的效能，以微調效能。如需 Oracle CDB 的詳細資訊，請參閱[RDS for Oracle 資料庫架構](#)。

在 Amazon RDS Performance Insights 儀表板中，您可以在「維度」索引標籤的「常用 PDB」索引標籤下找到有關可插拔資料庫 (PDB) 的資訊。

如需此功能的區域、資料庫引擎和執行個體類別支援資訊，請參閱[Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別](#)。

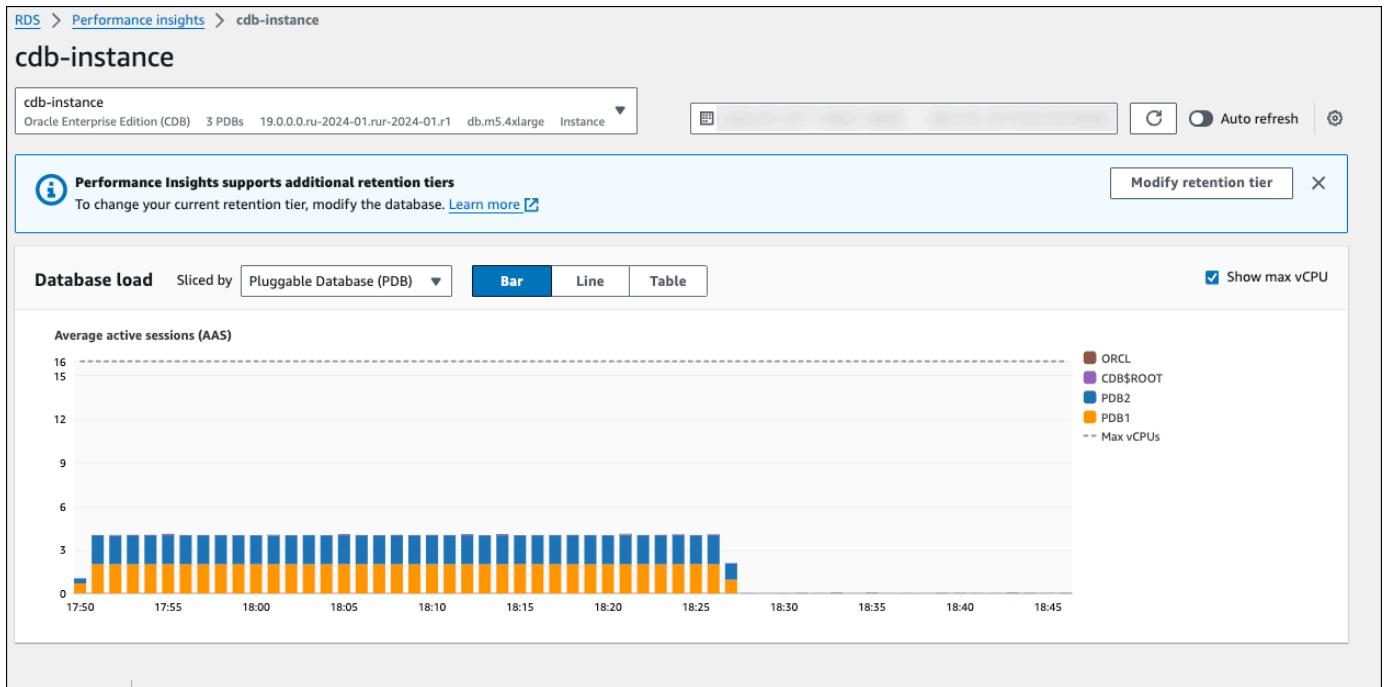
若要分析 Oracle CDB 中的「最上層 PDB」負載，請執行下

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在左側導覽窗格中，選取 [Performance Insights]。
3. 選擇甲骨文 CDB 執行個體。

資料庫執行個體的績效詳情儀表板即會出現。

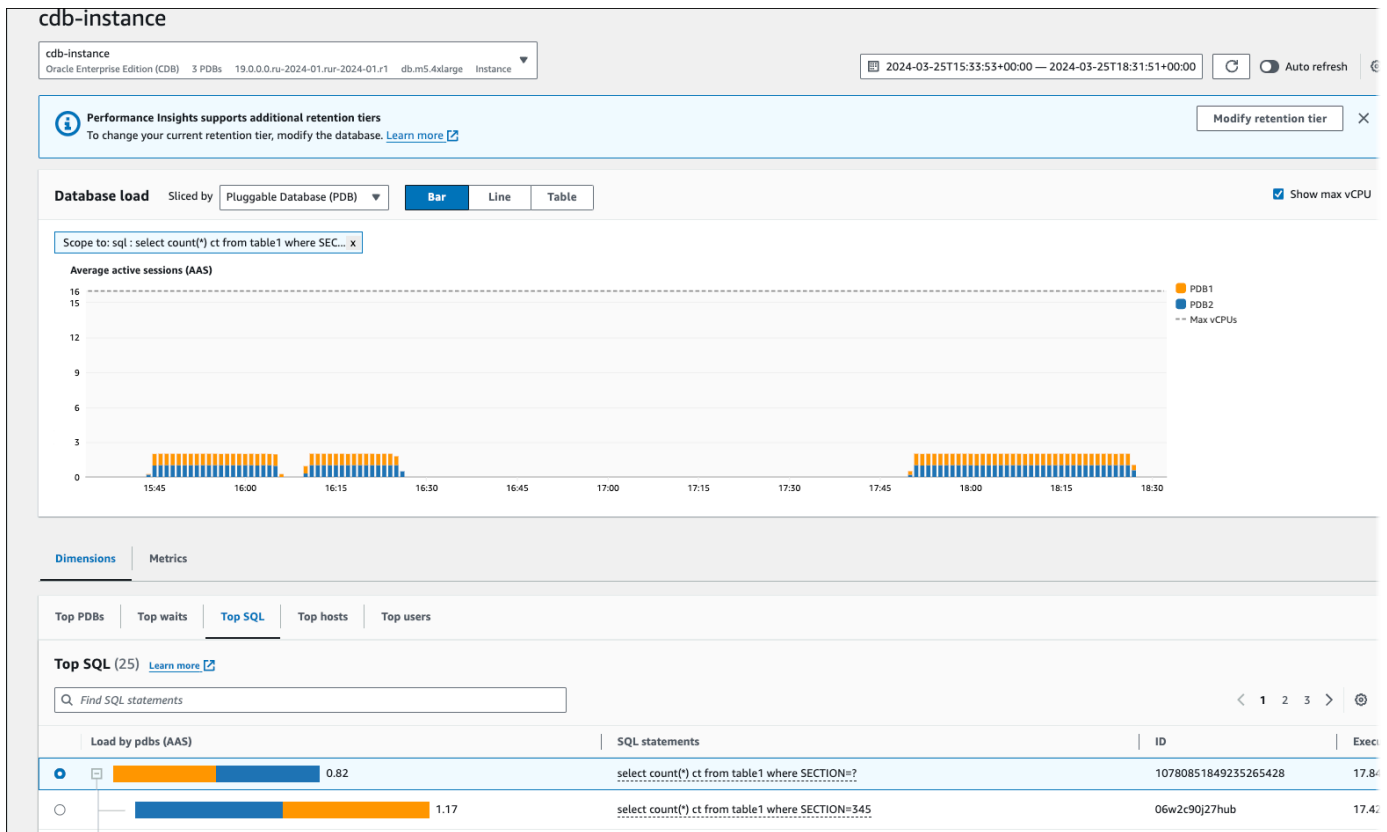
4. 在「資料庫載入 (DB 載入)」區段中，選擇「切片依據」旁邊的可插拔資料庫 (PDB)。

「平均作用中階段作業」圖表顯示負載最高的 PDB。PDB 識別碼會顯示在顏色編碼方塊的右側。每個識別碼可唯一識別 PDB。

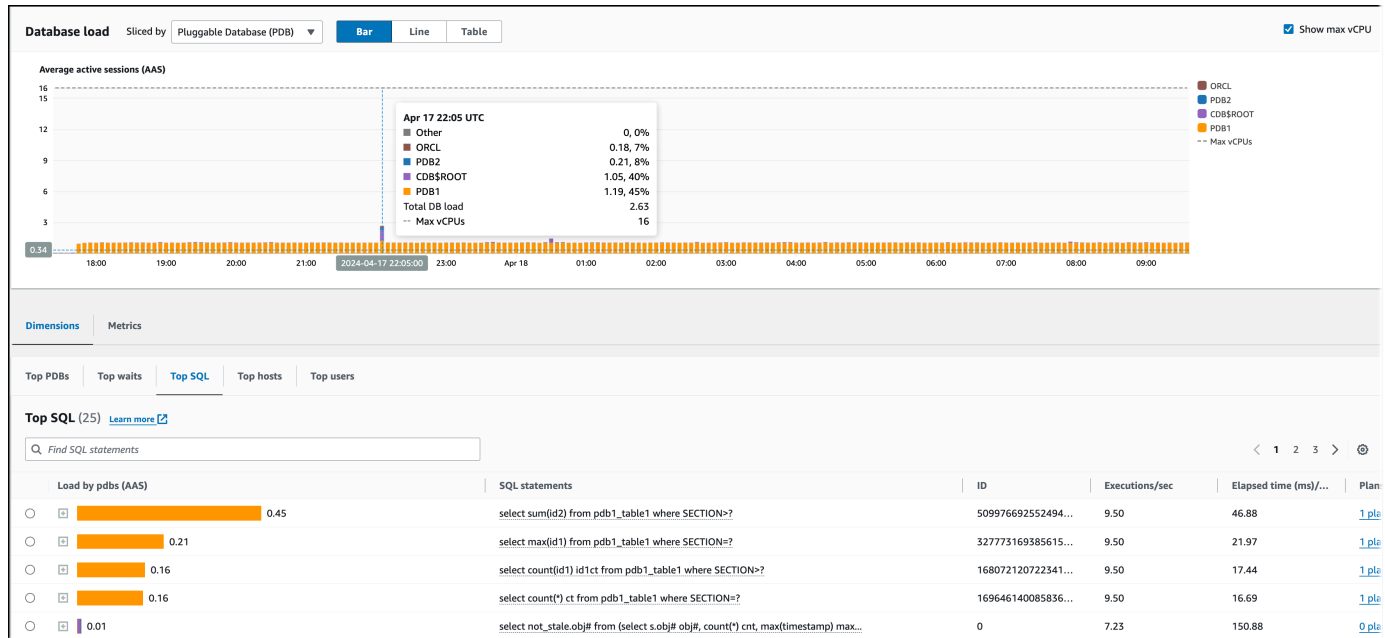


5. 向下捲動至 Top SQL (最高 SQL) 索引標籤。

在下面的例子中，你可以看到相同的 SQL 查詢和載入它驅動到多個 PDB。



在下列範例中，單一 PDB 處理的負載高於 CDB 中的其他 PDB。



如需有關 Oracle CDB 的詳細資訊，請參閱 [CDB 和 PDB](#)。

使用 Performance Insights 儀表板分析執行計劃

在 Amazon RDS Performance Insights 儀表板中，您可以找到 Oracle 和 SQL Server 資料庫執行個體執行計劃的相關資訊。您可以使用此信息來了解哪些計劃對資料庫負載的貢獻最大。

分析執行計畫

- [分析執行計畫的概觀](#)
- [使用績效詳情儀表板來分析 Oracle 執行計畫](#)
- [使用 \[Performance Insights\] 儀表板分析 SQL Server 執行計畫](#)

分析執行計畫的概觀

您可以使用 Amazon RDS Performance Insights 儀表板來瞭解哪些計劃對 Oracle 和 SQL Server 資料庫執行個體的資料庫負載貢獻最大。

例如，指定時間的最高 SQL 陳述式可能正使用下表中顯示的計畫。

最高 SQL	計畫
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 10	計畫 A
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 521	計畫 B
SELECT SUM(s_total) FROM sales WHERE region = 10	計畫 A
SELECT * FROM emp WHERE emp_id = 1000	計畫 C
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 72	計畫 A

可使用績效詳情的計畫功能執行以下操作：

- 找出最高 SQL 查詢使用的計畫。

例如，您可能會發現大多數資料庫負載是由使用計畫 A 和計畫 B 的查詢所產生，只有一小部分使用計畫 C。

- 比較同一查詢的不同計畫。

在上面的範例中，除了產品 ID 之外，三個查詢完全相同。兩個查詢使用計畫 A，但有一個查詢使用計畫 B。若要查看兩個計畫的差異，可以使用績效詳情。

- 了解查詢何時切換到新計畫。

您可能會看到有查詢原本使用計畫 A，然後在某個時間切換到計畫 B。此時資料庫是否有變化？例如，如果資料表空白，最佳化程式可能會選擇完整資料表掃描。如果資料表載入了一百萬列，則最佳化程式可能會切換到索引範圍掃描。

- 向下切入至成本最高的計畫之特定步驟。

例如，對於長時間執行的查詢，可能會在等值連接中顯示遺失聯結條件。這個缺少的條件強制使用笛卡兒聯結，聯結兩個資料表的所有列。

您可以使用績效詳情的計畫擷取功能執行上述任務。就像您可以透過等待事件和常用 SQL 來切割查詢一樣，也可以依計畫維度切割查詢。

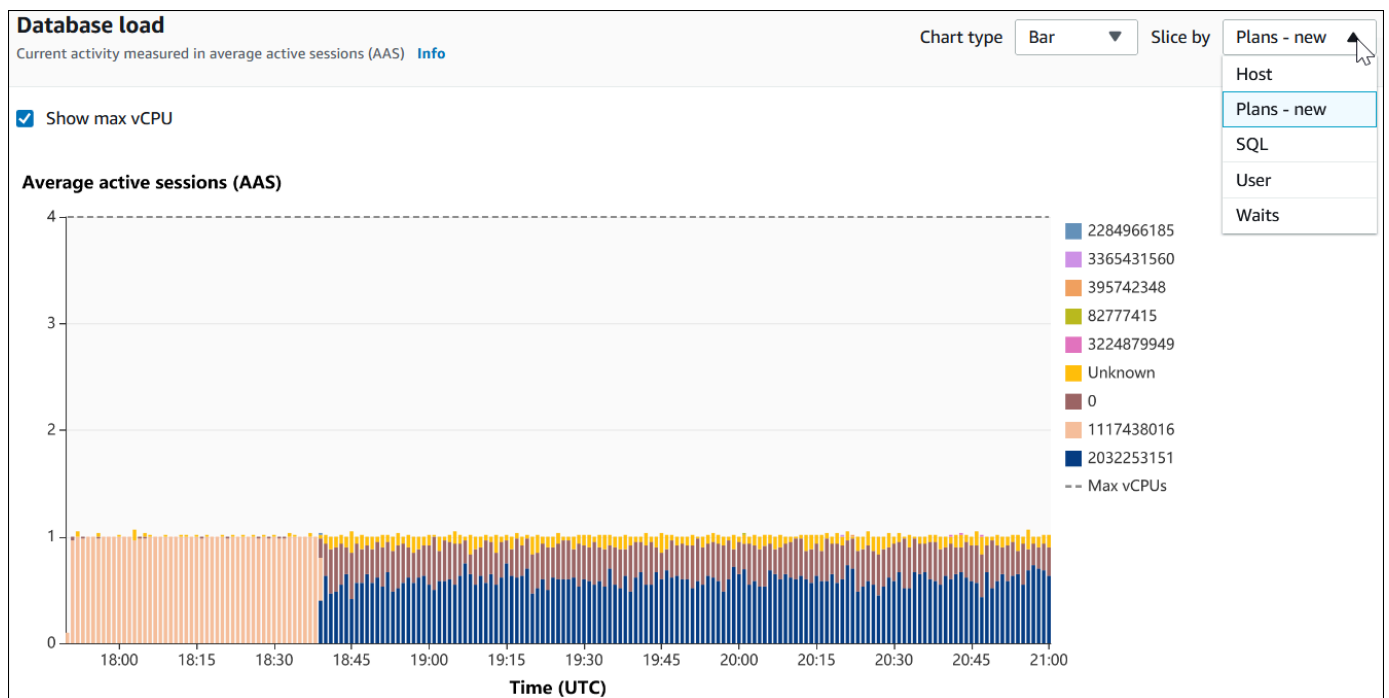
使用績效詳情儀表板來分析 Oracle 執行計畫

分析 Oracle 資料庫上的資料庫負載時，您可能想知道哪些計畫造成最多資料庫負載。您可以使用 Performance Insights 見的計劃擷取功能，判斷哪些計劃對資料庫負載的貢獻最大。

使用主控台分析 Oracle 執行計畫的方式

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Performance Insights (績效詳情)。
3. 選擇 Oracle 資料庫執行個體。顯示該資料庫執行個體的績效詳情儀表板。
4. 在 Database load (資料庫負載) 區段中，選擇 Slice Faves (配量依據) 旁邊的 Plans (計畫)。

平均作用中工作階段數圖顯示最高 SQL 陳述式使用的計畫。計畫雜湊值顯示在顏色編碼正形的右側。每個雜湊值唯一識別一個計畫。













5. 向下捲動至 Top SQL (最高 SQL) 索引標籤。

在下列範例中，最高 SQL 摘要包含兩個計畫。您可以透過陳述式中的問號來判斷這是摘要。






Top SQL (10) [Learn more](#)

Find SQL statements

	Load by plans (AAS)	SQL statements	Execution...	Plans cou...
<input type="radio"/>	 0.36	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=?	1611.28	2 plans
<input type="radio"/>	 0.24	DECLARE L_output NUMBER; BEGIN while true loop FOR i IN 1..2000 LOOP ...	0.00	0 plans
<input type="radio"/>	 0.02	SELECT	0.00	0 plans
<input type="radio"/>	 0.02	Unknown	0.00	0 plans
<input type="radio"/>	 0.01	PL/SQL EXECUTE	0.00	0 plans
<input type="radio"/>	 < 0.01	PSP0	0.00	0 plans
<input type="radio"/>	 < 0.01	DIA0	0.00	0 plans
<input type="radio"/>	 < 0.01	CKPT	0.00	0 plans
<input type="radio"/>	 < 0.01	LGWR	0.00	0 plans
<input type="radio"/>	 < 0.01	SELECT /* diffdigest1469 */ count(col1) FROM tab1 WHERE col1=?	7.74	1 plans

6. 選擇摘要，展開到其元件陳述式中。

在下列範例中，SELECT 陳述式是摘要查詢。摘要中的元件查詢使用兩個不同的計畫。計畫的顏色對應至資料庫負載圖。摘要中的計畫總數顯示在第二個資料欄中。

	Load by plans (AAS)	SQL statements	Execution...	Plans cou...
<input checked="" type="radio"/>	 0.36	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=?	1611.28	2 plans
<input type="radio"/>	 < 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996827	7.43	1 plans
<input type="radio"/>	 < 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=9961296	6.81	0 plans
<input type="radio"/>	 < 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996889	8.34	0 plans
<input type="radio"/>	 < 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996503	8.67	0 plans

7. 向下捲動，從 Plans for digest query (摘要查詢計畫) 清單中選擇兩個計畫進行比較。

一次可以查看查詢的一個或兩個計畫。下面的螢幕擷取畫面將摘要中的兩個計畫與雜湊 2032253151 和雜湊 1117438016 進行比較。在以下範例中，執行此摘要查詢的平均作用中工作階段中，62% 使用左側的計畫，而 38% 使用右側的計畫。

Plans for digest query **Info**
DB load caused by each plan is represented in average active session (AAS). In the DB load chart, you can slice the load by plans.

Choose plans

2032253151 Load by plan: 0.22 AAS
1117438016 Load by plan: 0.14 AAS

Choose up to 2 plans to examine at one time

2032253151 0.22 of 0.36 AAS (62%) total for this query

SQL_ID a2tm2f66sg3g2, child number 0
SELECT /* diffdigest1799 */ count(col1) FROM tab1 WHERE col1=53351799
Plan hash value: 2032253151

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				2 (100)	
1	SORT AGGREGATE		1	13		
* 2	INDEX RANGE SCAN	IND1	1	13	2 (0)	00:00:01

Query Block Name / Object Alias (identified by operation id):

```

1 - SEL$1
2 - SEL$1 / TAB1@SEL$1

```

Outline Data

1117438016 0.14 of 0.36 AAS (38%) total for this query

SQL_ID 50t2pcyygqf5s, child number 0
SELECT /* diffdigest1161 */ count(col1) FROM tab1 WHERE col1=53351161
Plan hash value: 1117438016

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				583 (100)	
1	SORT AGGREGATE		1	13		
* 2	TABLE ACCESS FULL	TAB1	23	299	583 (1)	00:00:01

Query Block Name / Object Alias (identified by operation id):

```

1 - SEL$1
2 - SEL$1 / TAB1@SEL$1

```

Outline Data

在此範例中，兩個計畫有很重要的差異。計畫 2032253151 中的步驟 2 使用索引掃描，而計畫 1117438016 使用完整資料表掃描。對於具有大量資料列的資料表來說，使用索引掃描進行單列查詢幾乎總是比較快。

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				2 (100)	
1	SORT AGGREGATE		1	13		
* 2	INDEX RANGE SCAN	IND1	1	13	2 (0)	00:00:01

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				583 (100)	
1	SORT AGGREGATE		1	13		
* 2	TABLE ACCESS FULL	TAB1	23	299	583 (1)	00:00:01

8. (選用) 選擇 Copy (複製) 將計畫複製到剪貼簿，或選 Download (下載) 將計畫儲存到硬碟中。

使用 [Performance Insights] 儀表板分析 SQL Server 執行計畫

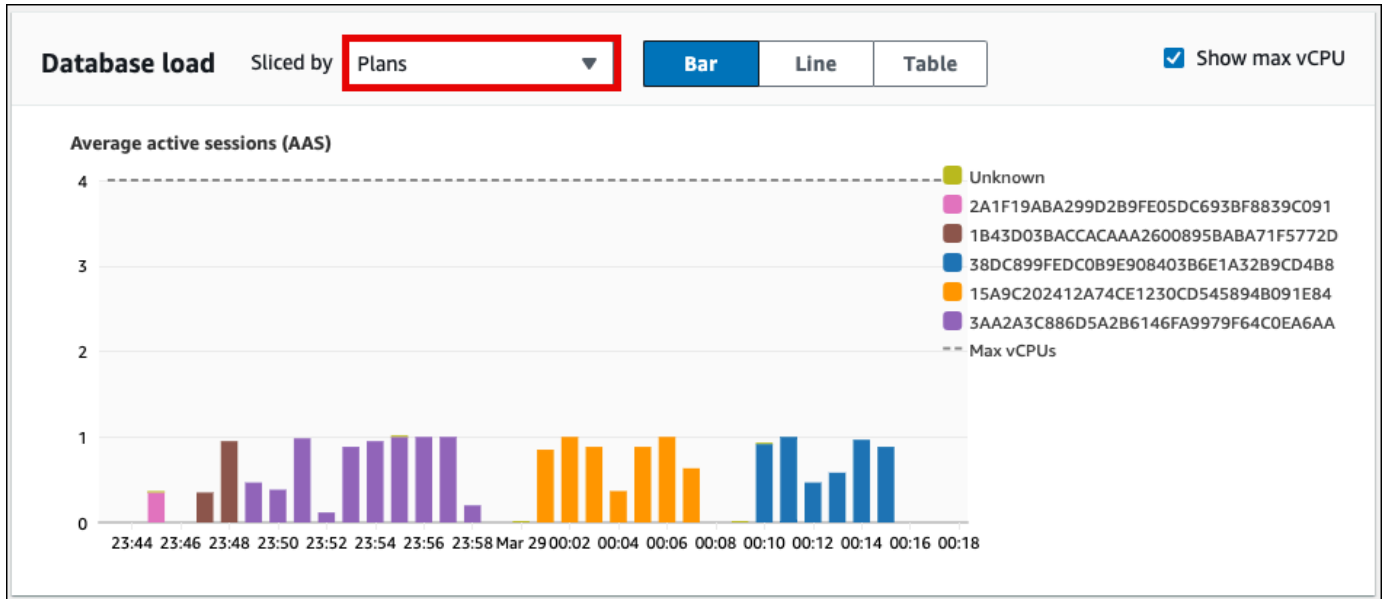
在 SQL Server 資料庫上分析資料庫負載時，您可能想知道哪些計畫對資料庫負載的貢獻最大。您可以使用 Performance Insights 見的計畫擷取功能，判斷哪些計畫對資料庫負載的貢獻最大。

使用主控台分析 SQL 伺服器執行計畫

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Performance Insights (績效詳情)。

- 選擇 SQL 伺服器資料庫執行個體。顯示該資料庫執行個體的績效詳情儀表板。
- 在 Database load (資料庫負載) 區段中，選擇 Slice Faves (配量依據) 旁邊的 Plans (計畫)。

平均作用中工作階段數圖顯示最高 SQL 陳述式使用的計畫。計畫雜湊值顯示在顏色編碼正形的右側。每個雜湊值唯一識別一個計畫。



- 向下捲動至 Top SQL (最高 SQL) 索引標籤。









在下列範例中，前幾個 SQL 摘要有三個計畫。SQL 陳述式中存在問號表示陳述式為摘要。若要檢視完整的 SQL 敘述句，請在 SQL 敘述句資料欄中選擇一個值。

The screenshot shows the 'Top SQL' dashboard with the 'Top SQL (6)' tab selected. The table below lists the top SQL statements, their load by plans (AAS), and the number of plans used.

Load by plans (AAS)	SQL statements	Plans count
0.48	SELECT CustOrders.OrderID FROM CustOrders WHERE CustOrders.OrderDate BETWEEN '?...	3 plans
0.04	INSERT INTO CustOrders (OrderID, CustomerID, OrderDate) VALUES (? (ABS(CHEC...	0 plans
< 0.01	SELECT [Orders].[OrderID] FROM [Orders] WHERE [Orders].[OrderDate]>=? AND [Order...	0 plans
< 0.01	BACKUP LOG ? TO VIRTUAL_DEVICE = ? WITH buffercount = ?, maxtransfersize = ?, IN...	0 plans
< 0.01	ALTER INDEX [PK__Orders__C3905BAF6D1AC47E] ON [dbo].[Orders] REBUILD PARTITION =...	0 plans
< 0.01	(? varchar(?),? varchar(?))SELECT [CustOrders].[OrderID] FROM [CustOrders] WHERE...	0 plans

- 選擇摘要，展開到其元件陳述式中。

在下列範例中，SELECT 陳述式是摘要查詢。摘要中的元件查詢會使用三種不同的執行計畫。指定給平面圖的顏色對應於資料庫負載圖表。

Top waits	Top SQL	Top hosts	Top users	Top applications	Top databases
Top SQL (6) Learn more					
<input type="text" value="Find SQL statements"/>					
Load by plans (AAS)		SQL statements			Plans count
<input checked="" type="radio"/>		SELECT CustOrders.OrderID FROM CustOrders WHERE CustOrders.OrderDate BETWEEN '?...'			3 plans
<input type="radio"/>		SELECT [CustOrders].[OrderID] FROM [CustOrders] WHERE [CustOrders].[OrderDate]>=...			2 plans
<input type="radio"/>		SELECT CustOrders.OrderID FROM CustOrders WHERE CustOrders.OrderDate BETWEEN '20...			1 plans
<input type="radio"/>		INSERT INTO CustOrders (OrderID, CustomerID, OrderDate) VALUES (? (ABS(CHEC...			0 plans
<input type="radio"/>		SELECT [Orders].[OrderID] FROM [Orders] WHERE [Orders].[OrderDate]>=? AND [Order...			0 plans
<input type="radio"/>		BACKUP LOG ? TO VIRTUAL_DEVICE = ? WITH buffercount = ?, maxtransfersize = ?, IN...			0 plans
<input type="radio"/>		ALTER INDEX [PK_Orders_C3905BAF6D1AC47E] ON [dbo].[Orders] REBUILD PARTITION =...			0 plans
<input type="radio"/>		(? varchar(?),? varchar(?))SELECT [CustOrders].[OrderID] FROM [CustOrders] WHERE...			0 plans

7. 向下捲動，從 Plans for digest query (摘要查詢計畫) 清單中選擇兩個計畫進行比較。

一次可以查看查詢的一個或兩個計畫。下列螢幕擷取畫面會比較摘要中的兩個計畫。在下列範例中，執行此摘要查詢的平均作用中工作階段中有 40% 正在使用左側的計畫，而 28% 則使用右側的計畫。

SQL text **Plans**

Plans for digest query [Info](#)
DB load caused by each plan is represented in average active session (AAS). In the DB load chart, you can slice the load by plans.

Choose plans

- 3AA2A3C886D5A2B6146FA9979F64C0EA6AAC8F25A0FDF36F61D1DF0863C89B79 X
Load by plan: 0.19 AAS
- 38DC899FEDC0B9E908403B6E1A32B9CD4B884E68F3CEBF8495FE1FA76EA82306 X
Load by plan: 0.13 AAS

Choose up to 2 plans to examine at one time

3AA2A3C886D5A2B6146FA9979F64C0EA6AAC8F25A0FDF36F61D1DF0863C89B79
0.19 of 0.48 AAS (40%) total for this query

38DC899FEDC0B9E908403B6E1A32B9CD4B884E68F3CEBF8495FE1FA76EA82306
0.13 of 0.48 AAS (28%) total for this query

Plan Details
(3AA2A3C886D5A2B6146FA9979F64C0EA6AAC8F25A0FDF36F61D1DF0863C89B79)

Filter plans by statement

Statement text	Rows estimate	Io estimate
Batch 0	-	-
@@1 varchar(8000),@2 varchar(8000))SELECT [CustOrders].[OrderID] FROM [CustOrder... Table Scan	75889	0.329129

Copy Download

Plan Details
(38DC899FEDC0B9E908403B6E1A32B9CD4B884E68F3CEBF8495FE1FA76EA82306)

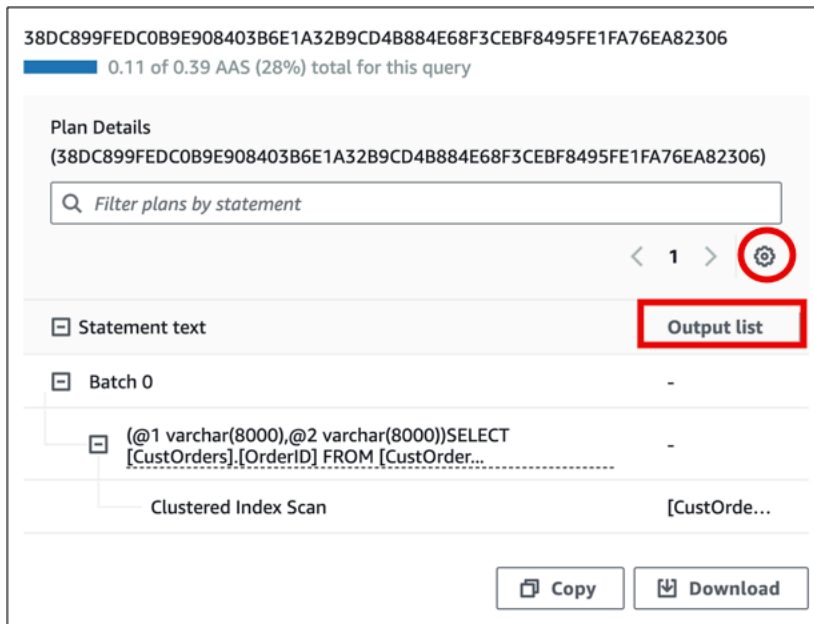
Filter plans by statement

Statement text	Rows estimate	Io estimate
Batch 0	-	-
@@1 varchar(8000),@2 varchar(8000))SELECT [CustOrders].[OrderID] FROM [CustOrder... Clustered Index Scan	75889	0.186088

Copy Download

在前面的例子中，計劃在一個重要的方式不同。左側計劃中的步驟 2 使用資料表掃描，而右側的計畫則使用叢集索引掃描。對於具有大量資料列的資料表，使用叢集索引掃描時，擷取單一資料列的查詢幾乎總是比較快速。


- (選擇性) 選擇「計劃詳細資訊」表上的「設定」圖示，以自訂欄的可見度和順序。下列螢幕擷取畫面顯示「計劃詳細資料」表格，其中「輸出」清單欄為第二欄。



38DC899FEDC0B9E908403B6E1A32B9CD4B884E68F3CEBF8495FE1FA76EA82306
0.11 of 0.39 AAS (28%) total for this query

Plan Details
(38DC899FEDC0B9E908403B6E1A32B9CD4B884E68F3CEBF8495FE1FA76EA82306)

Filter plans by statement

< 1 > 

Statement text	Output list
Batch 0	-
(@1 varchar(8000),@2 varchar(8000))SELECT [CustOrders],[OrderID] FROM [CustOrder...]	-
Clustered Index Scan	[CustOrde...]

Copy Download

9. (選用) 選擇 Copy (複製) 將計畫複製到剪貼簿，或選 Download (下載) 將計畫儲存到硬碟中。

Note

Performance Insights 會使用階層式樹狀結構資料表顯示預估的執行計。此表格包含每個陳述式的部分執行資訊。如需有關「計劃詳細資訊」表格中資料行的詳細資訊，請參閱 SQL Server 說明文件中的 [設定顯示 PLAN_ALL](#)。若要顯示預估執行計畫的完整執行資訊，請選擇 [下載] 下載計劃，然後將該計劃上傳至 SQL Server 管理工作室。如需有關使用 SQL Server 管理工作室顯示預估執行計畫的詳細資訊，請參閱 SQL Server 說明文件中的 [顯示預估的執行計畫](#)。

檢視 Performance Insights 主動建議

Amazon RDS Performance Insights 可能會監控特定指標，並透過分析指定資源可能發生問題的層級來自動建立閾值。當新的測量結果值在指定期間內跨越預先定義的臨界值時，Performance Insights 會產生主動式建議。此建議有助於防止 future 的資料庫效能影響。若要接收這些主動式建議，您必須開啟具有付費方案保留期的 Performance Insights。

如需開啟績效詳情的詳細資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)。如需效能洞見的定價和資料保留的相關 Performance Insights，請參閱 [Performance Insights 的定價和資料保留](#)。

若要瞭解主動式建議支援的區域、資料庫引擎和執行個體類別，請參閱 [Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別](#)。

您可以在建議詳細資訊頁面中檢視主動式建議的詳細分析和建議調查。

如需建議的詳細資訊，請參閱[檢視和回應 Amazon RDS 建議](#)。

若要檢視主動式建議的詳細分析

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，執行下列任一項作業：

- 選擇「建議」。

[建議] 頁面會顯示依您帳戶中所有資源嚴重性排序的建議清單。

- 選擇資料庫，然後在資料庫頁面中選擇資源的建議。

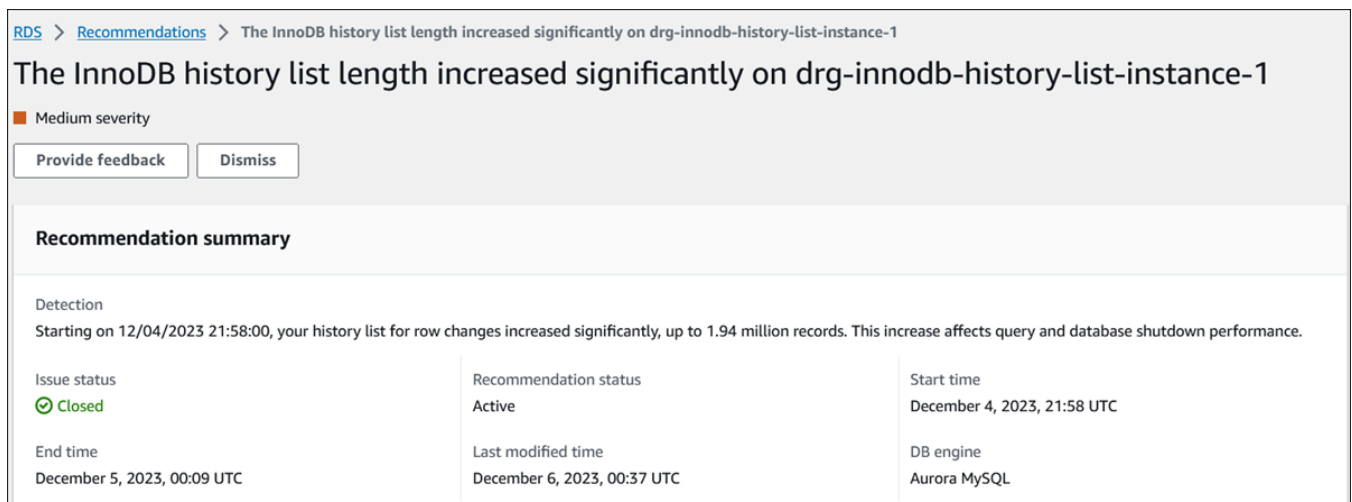
[建議] 索引標籤會顯示所選資源的建議及其詳細資訊。

3. 尋找主動式建議並選擇 [檢視詳細資料]。

便會顯示建議詳細資訊頁面。標題提供受影響資源的名稱，其中包含偵測到的問題和嚴重性。

下列是「建議詳細資訊」頁面上的元件：

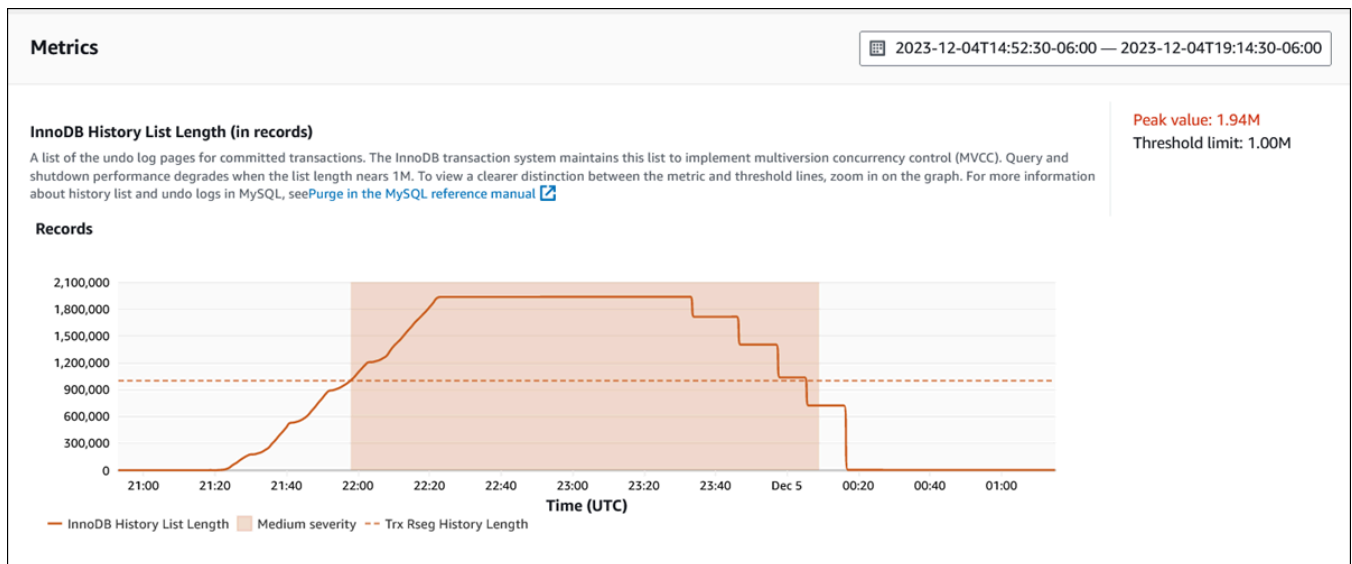
- 建議摘要 — 偵測到的問題、建議和問題狀態、問題開始和結束時間、建議修改時間，以及引擎類型。



The screenshot displays a recommendation card in the AWS Management Console. The card title is "The InnoDB history list length increased significantly on drg-innodb-history-list-instance-1" with a "Medium severity" indicator. It includes buttons for "Provide feedback" and "Dismiss". The "Recommendation summary" section contains the following details:

Detection	Starting on 12/04/2023 21:58:00, your history list for row changes increased significantly, up to 1.94 million records. This increase affects query and database shutdown performance.	
Issue status	Recommendation status	Start time
🟢 Closed	Active	December 4, 2023, 21:58 UTC
End time	Last modified time	DB engine
December 5, 2023, 00:09 UTC	December 6, 2023, 00:37 UTC	Aurora MySQL

- 指標 — 偵測到的問題的圖形。每個圖形都會顯示一個臨界值，由資源的基準行為和問題開始時間後報告的指標資料決定。



- 分析和建議 — 建議和建議建議的原因。

Analysis and recommendations

Recommendation	Why is this recommended?
<p>Do the following:</p> <ul style="list-style-type: none"> • Check for long-running transactions and end them with a commit or rollback. • Check the top hosts and top users in Performance Insights. Apply tuning to transactions that need to store a large number of row versions. • Don't shut down the database until the InnoDB history list decreases. <p>View troubleshooting doc</p>	<p>The InnoDB history list increased significantly because of long transactions or a heavy write load. Address this event to avoid degraded query and database shutdown performance.</p>

您可以檢閱問題的原因，然後執行建議的建議動作來修正問題，或選擇右上角的 [關閉] 以關閉建議。

使用適用於 Amazon RDS 的 Performance Insights API 擷取指標

在啟用 Performance Insights 時，API 會提供對執行個體效能的可見性。Amazon CloudWatch Logs 為服務的付費監控指標提供授權來源。AWS

績效詳情提供以平均作用中工作階段 (AAS) 評估的資料庫負載特定網域檢視。此指標在 API 消費者看來是二維時間序列資料集。資料的時間維度提供查詢的時間範圍內各時間點的資料庫負載資料。每個時間點會根據請求的維度來分解整體負載，例如 SQL、Wait-event、User、或者 Host，在該時間點所測得。

Amazon RDS 績效詳情會監控您的 Amazon RDS 資料庫執行個體，讓您可分析資料庫效能並對其進行故障診斷。檢視績效詳情資料的一個方法就是使用 AWS Management Console。績效詳情也提供公有 API，讓您可以查詢自己的資料。您可以使用 API 執行下列動作：

- 將資料卸載至資料庫
- 將績效詳情資料新增至現有監控儀表板
- 建置監控工具

若要使用績效詳情 API，請在其中一個 Amazon RDS 資料庫執行個體上啟用績效詳情。如需啟用績效詳情的相關資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)。如需績效詳情 API 的相關詳細資訊，請參閱 [Amazon RDS 績效詳情 API 參考](#)。

績效詳情 API 提供下列操作。

績效詳情動作	AWS CLI 命令	描述
CreatePerformanceAnalysisReport	aws pi create-performance-analysis-report	針對資料庫執行個體的特定時間區間建立績效分析報告。結果是 AnalysisReportId，也是報告的唯一識別符。
DeletePerformanceAnalysisReport	aws pi delete-performance-analysis-report	刪除績效分析報告。
DescribeDimensionKeys	aws pi describe-dimension-keys	針對特定時段，擷取其指標的前 N 個維度金鑰。
GetDimensionKeyDetails	aws pi get-dimension-key-details	擷取資料庫執行個體或資料來源之指定維度群組的屬性。比方說，如果指定 SQL ID，且有維度詳細資訊可用，則 GetDimensionKeyDetails 會擷取與此 ID db.sql.statement 相關聯之維度的完整文字。這項操作很有用，因為 GetResourceMetrics

績效詳情動作	AWS CLI 命令	描述
<u>GetPerformanceAnalysisReport</u>	<u>aws pi get-performance-analysis-report</u>	和 DescribeDimensionKeys 不支援擷取大量的 SQL 陳述式文字。 擷取報告，包括報告洞見。結果包括報告狀態、報告 ID、報告時間詳細資訊、洞見和建議。
<u>GetResourceMetadata</u>	<u>aws pi get-resource-metadata</u>	檢索不同功能的中繼資料。例如，中繼資料可能指出特定資料庫執行個體上某項功能已開啟或關閉。
<u>GetResourceMetrics</u>	<u>aws pi get-resource-metrics</u>	擷取一組資料來源某個時段的績效詳情指標。您可以提供特定維度群組和維度，以及為每個群組提供彙總和篩選條件。
<u>ListAvailableResourceDimensions</u>	<u>aws pi list-available-resource-dimensions</u>	檢索指定執行個體上每個指定指標類型可查詢的維度。
<u>ListAvailableResourceMetrics</u>	<u>aws pi list-available-resource-metrics</u>	檢索可為指定資料庫執行個體查詢的指定指標類型中所有可用的指標。
<u>ListPerformanceAnalysisReports</u>	<u>aws pi list-performance-analysis-reports</u>	擷取資料庫執行個體可用的所有分析報告。報告會根據每個報告的開始時間列出。
<u>ListTagsForResource</u>	<u>aws pi list-tags-for-resource</u>	列出所有新增至資源的中繼資料標籤。清單包括標籤的名稱和值。

績效詳情動作	AWS CLI 命令	描述
TagResource	aws pi tag-resource	將中繼資料標籤新增到 Amazon RDS 資源。標籤包括一個名稱和一個值。
UntagResource	aws pi untag-resource	從資源移除中繼資料標籤。

主題

- [AWS CLI Performance Insights](#)
- [擷取時間序列指標](#)
- [AWS CLI Performance Insights 的範例](#)

AWS CLI Performance Insights

您可以使用 AWS CLI 檢視績效詳情資料。您可以在命令列上輸入下列命 AWS CLI 令，以檢視 Performance Insights 指令的說明。

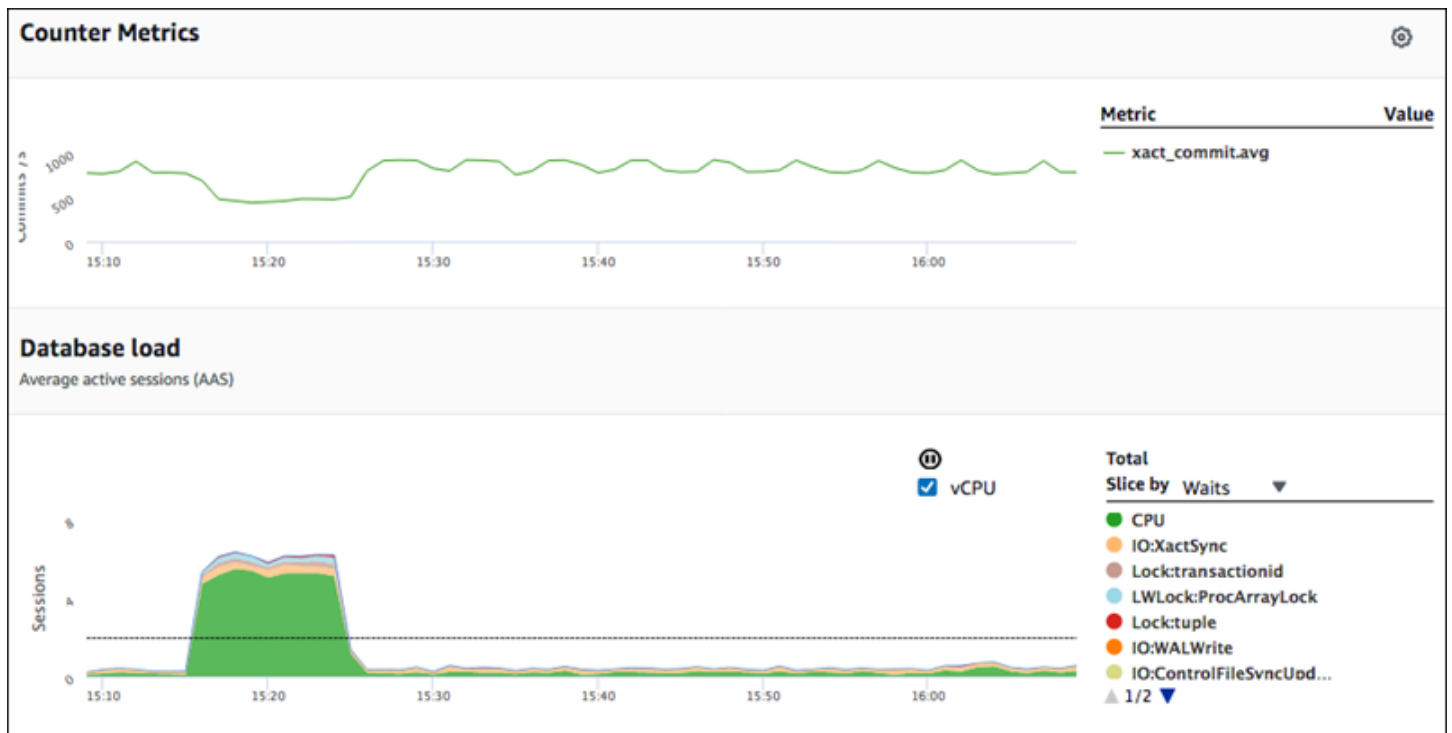
```
aws pi help
```

如果您尚未安 AWS CLI 裝，請參閱《AWS CLI 使用指南》AWS CLI 中的 [〈安裝〉](#)，以取得有關安裝它的資訊。

擷取時間序列指標

GetResourceMetrics 操作會從績效詳情資料中擷取一或多個時間時間序列指標。GetResourceMetrics 需要指標和時間間隔，並傳回含資料點清單的回應。

例如，AWS Management Console 用 GetResourceMetrics 來填入「計數器測量結果」圖表和「資料庫負載」圖表，如下圖所示。



GetResourceMetrics 傳回的所有指標，除 db.load 之外，皆為標準的時間序列指標。此指標會顯示在 Database Load (資料庫負載) 圖表中。db.load 指標與其他時間序列指標不同，因為您可以將它分為名為維度的子元件。在先前的影像中，db.load 已被細分，分組依據為組成 db.load 的等待狀態。

Note

GetResourceMetrics 也可以傳回 db.sampleload 指標，但 db.load 指標適用於大部分情況。

如需 GetResourceMetrics 所傳回指標的相關資訊，請參閱 [Performance Insights 計數器指標](#)。

這些指標支援下列計算：

- 平均值 – 指標在一段時間內的平均值。將 .avg 附加至指標名稱。
- 最小值 – 指標在一段時間內的最小值。將 .min 附加至指標名稱。
- 最大值 – 指標在一段時間內的最大值。將 .max 附加至指標名稱。
- 總和 – 指標值在一段時間內的總和。將 .sum 附加至指標名稱。
- 取樣計數 – 在一段時間內收集指標的次數。將 .sample_count 附加至指標名稱。

例如，假設收集指標的時間為 300 秒 (5 分鐘)，且每分鐘收集一次指標。每分鐘的值為 1、2、3、4 和 5。在此情況下，會傳回下列計算：

- 平均值 – 3
- 最小值 – 1
- 最大值 – 5
- 總和 – 15
- 取樣計數 – 5

若要取得有關使用 `get-resource-metrics` AWS CLI 指令的資訊，請參閱 [get-resource-metrics](#)。

對於 `--metric-queries` 選項，請指定您要取得結果的一或多個查詢。每個查詢的組成為必要的 `Metric` 和選用的 `GroupBy` 及 `Filter` 參數。以下是 `--metric-queries` 選項規格的範例。

```
{
  "Metric": "string",
  "GroupBy": {
    "Group": "string",
    "Dimensions": ["string", ...],
    "Limit": integer
  },
  "Filter": {"string": "string"
  ...}
```

AWS CLI Performance Insights 的範例

下列範例顯示如何使用「AWS CLI Performance Insights」。

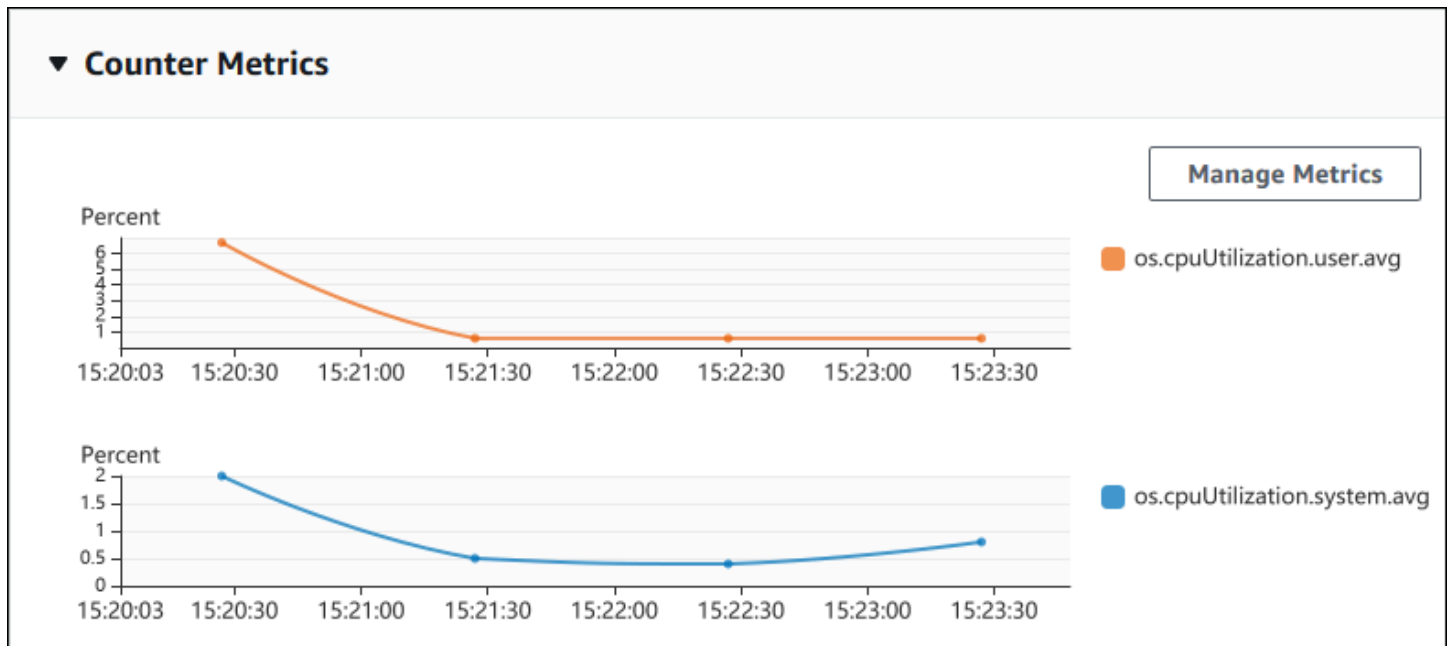
主題

- [擷取計數器指標](#)
- [擷取最久等待事件的資料庫負載平均值](#)
- [擷取最高 SQL 的資料庫負載平均值](#)
- [擷取依據 SQL 篩選的資料庫負載平均值](#)
- [擷取 SQL 陳述式的完整文字](#)
- [建立一段時間區間的績效分析報告](#)
- [擷取績效分析報告](#)

- [列出資料庫執行個體的所有績效分析報告](#)
- [刪除績效分析報告](#)
- [將標籤新增至績效分析報告](#)
- [列出績效分析報告的所有標籤](#)
- [從績效分析報告中刪除標籤](#)

擷取計數器指標

下列螢幕擷取畫面顯示 AWS Management Console 中的兩個計數器指標圖表。



下列範例顯示如何收集 AWS Management Console 用來產生兩個計數器測量結果圖表的相同資料。

對於LinuxmacOS、或Unix：

```
aws pi get-resource-metrics \
  --service-type RDS \
  --identifier db-ID \
  --start-time 2018-10-30T00:00:00Z \
  --end-time 2018-10-30T01:00:00Z \
  --period-in-seconds 60 \
  --metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
                    {"Metric": "os.cpuUtilization.idle.avg"}]'
```

在 Windows 中：

```
aws pi get-resource-metrics ^
  --service-type RDS ^
  --identifier db-ID ^
  --start-time 2018-10-30T00:00:00Z ^
  --end-time 2018-10-30T01:00:00Z ^
  --period-in-seconds 60 ^
  --metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
                    {"Metric": "os.cpuUtilization.idle.avg"}]'
```

您也可以透過指定 `--metrics-query` 選項的檔案來提高命令的可讀性。以下範例會將名為 `query.json` 的檔案用於此選項。此檔案的內容如下。

```
[
  {
    "Metric": "os.cpuUtilization.user.avg"
  },
  {
    "Metric": "os.cpuUtilization.idle.avg"
  }
]
```

執行下列命令來使用檔案。

對於LinuxmacOS、或Unix：

```
aws pi get-resource-metrics \
  --service-type RDS \
  --identifier db-ID \
  --start-time 2018-10-30T00:00:00Z \
  --end-time 2018-10-30T01:00:00Z \
  --period-in-seconds 60 \
  --metric-queries file://query.json
```

在 Windows 中：

```
aws pi get-resource-metrics ^
  --service-type RDS ^
  --identifier db-ID ^
  --start-time 2018-10-30T00:00:00Z ^
  --end-time 2018-10-30T01:00:00Z ^
  --period-in-seconds 60 ^
```



```
--metric-queries file://query.json
```

先前的範例會為選項指定下列值：

- `--service-type` – RDS for Amazon RDS
- `--identifier`– 資料執行個體的資源 ID
- `--start-time` 和 `--end-time` – 要查詢期間的 ISO 8601 DateTime 值，支援多種格式

它會查詢一小時的時間範圍：

- `--period-in-seconds=60` 適用於每分鐘的查詢
- `--metric-queries`– 兩個查詢的陣列，一個指標剛好一個查詢。

此指標名稱會使用點將指標分類在實用的類別，其中最後一個元素則做為函數。在此範例中，此函數是每個查詢的 avg。與 Amazon 一樣 CloudWatch，支援的功能是 minmax，total，和 avg。

回應看起來類似以下的內容。

```
{
  "Identifier": "db-XXX",
  "AlignedStartTime": 1540857600.0,
  "AlignedEndTime": 1540861200.0,
  "MetricList": [
    { //A list of key/datapoints
      "Key": {
        "Metric": "os.cpuUtilization.user.avg" //Metric1
      },
      "DataPoints": [
        //Each list of datapoints has the same timestamps and same number of
items
        {
          "Timestamp": 1540857660.0, //Minute1
          "Value": 4.0
        },
        {
          "Timestamp": 1540857720.0, //Minute2
          "Value": 4.0
        },
        {
          "Timestamp": 1540857780.0, //Minute 3
```

```

        "Value": 10.0
      }
      //... 60 datapoints for the os.cpuUtilization.user.avg metric
    ]
  },
  {
    "Key": {
      "Metric": "os.cpuUtilization.idle.avg" //Metric2
    },
    "DataPoints": [
      {
        "Timestamp": 1540857660.0, //Minute1
        "Value": 12.0
      },
      {
        "Timestamp": 1540857720.0, //Minute2
        "Value": 13.5
      },
      //... 60 datapoints for the os.cpuUtilization.idle.avg metric
    ]
  }
] //end of MetricList
} //end of response

```

回應具有 Identifier、AlignedStartTime 和 AlignedEndTime。--period-in-seconds 值為 60，開始和結束時間皆一致使用分鐘。如果 --period-in-seconds 是 3600，開始和結束時間則會一致使用小時。

回應中的 MetricList 擁有許多項目，每個都包含 Key 和 DataPoints 項目。每個 DataPoint 都有 Timestamp 和 Value。每個 Datapoints 清單有 60 個資料點，因為查詢是適用於一小時中的每分鐘資料，內含 Timestamp1/Minute1、Timestamp2/Minute2 等，最多可達 Timestamp60/Minute60。

因為此查詢是適用於兩個不同的計數器指標，回應 MetricList 中會有兩個元素。

擷取最久等待事件的資料庫負載平均值

下列範例與 AWS Management Console 用來產生堆疊區域線圖的查詢相同。此範例會使用根據前七個最久的等待事件而區分的負載來擷取前一小時的 db.load.avg。此命令與 [擷取計數器指標](#) 中的命令相同。然而，查詢 query.json 檔案有以下內容。

```
[
  {
```

```

    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
  }
]

```

執行下列命令。

對於Linux/macOS、或Unix：

```

aws pi get-resource-metrics \
  --service-type RDS \
  --identifier db-ID \
  --start-time 2018-10-30T00:00:00Z \
  --end-time 2018-10-30T01:00:00Z \
  --period-in-seconds 60 \
  --metric-queries file://query.json

```

在 Windows 中：

```

aws pi get-resource-metrics ^
  --service-type RDS ^
  --identifier db-ID ^
  --start-time 2018-10-30T00:00:00Z ^
  --end-time 2018-10-30T01:00:00Z ^
  --period-in-seconds 60 ^
  --metric-queries file://query.json

```

此範例會指定 db.load.avg 指標與前七個最久等待事件的 GroupBy。如需此範例有效值的詳細資訊，請參閱 Performance Insights API 參考 [DimensionGroup](#) 中的。

回應看起來類似以下的內容。

```

{
  "Identifier": "db-XXX",
  "AlignedStartTime": 1540857600.0,
  "AlignedEndTime": 1540861200.0,
  "MetricList": [
    { //A list of key/datapoints
      "Key": {
        //A Metric with no dimensions. This is the total db.load.avg
        "Metric": "db.load.avg"
      },
      "DataPoints": [

```

```

        //Each list of datapoints has the same timestamps and same number of
items
        {
            "Timestamp": 1540857660.0, //Minute1
            "Value": 0.5166666666666667
        },
        {
            "Timestamp": 1540857720.0, //Minute2
            "Value": 0.38333333333333336
        },
        {
            "Timestamp": 1540857780.0, //Minute 3
            "Value": 0.26666666666666666
        }
        //... 60 datapoints for the total db.load.avg key
    ]
},
{
    "Key": {
        //Another key. This is db.load.avg broken down by CPU
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.name": "CPU",
            "db.wait_event.type": "CPU"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1540857660.0, //Minute1
            "Value": 0.35
        },
        {
            "Timestamp": 1540857720.0, //Minute2
            "Value": 0.15
        },
        //... 60 datapoints for the CPU key
    ]
},
    //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
] //end of MetricList
} //end of response

```

在此回應中，MetricList 中有八個項目。有一個項目適用於總計 db.load.avg，有七個項目，分別適用於根據前七個最久等待事件區份的 db.load.avg。與第一個範例不同，因為其中有分組維度，每個指標分組都必須有一個索引鍵。每個指標不能只有一個索引鍵，如同基本計數器指標使用案例。

擷取最高 SQL 的資料庫負載平均值

以下範例會根據前 10 個 SQL 陳述式來分組 db.wait_events。SQL 陳述式有兩個不同的分組：

- db.sql- 完整的 SQL 陳述式，例如 `select * from customers where customer_id = 123`
- db.sql_tokenized- 字符化的 SQL 陳述式，例如 `select * from customers where customer_id = ?`

分析資料庫效能時，將僅參數不同的 SQL 陳述式視為單一邏輯項目可能會很有幫助。因此，您可以在查詢時使用 db.sql_tokenized。然而，特別是在您對說明計畫感興趣時，使用參數來檢查完整 SQL 陳述式並依據 db.sql 來查詢分組有時候會更有幫助。這是字符化與完整 SQL 之間的父子關係，內含使用相同字符化 SQL (父項) 分組的多個完整 SQL (子項)。

此範例中的命令與 [擷取最久等待事件的資料庫負載平均值](#) 中的命令類似。然而，查詢 query.json 檔案有以下內容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }
  }
]
```

以下範例使用 db.sql_tokenized。

對於LinuxmacOS、或Unix：

```
aws pi get-resource-metrics \
  --service-type RDS \
  --identifier db-ID \
  --start-time 2018-10-29T00:00:00Z \
  --end-time 2018-10-30T00:00:00Z \
  --period-in-seconds 3600 \
  --metric-queries file://query.json
```

在 Windows 中：

```
aws pi get-resource-metrics ^
--service-type RDS ^
--identifier db-ID ^
--start-time 2018-10-29T00:00:00Z ^
--end-time 2018-10-30T00:00:00Z ^
--period-in-seconds 3600 ^
--metric-queries file://query.json
```

這個範例會查詢超過 24 小時，包含一小時的時間 period-in-seconds。

此範例會指定 db.load.avg 指標與前七個最久等待事件的 GroupBy。如需此範例有效值的詳細資訊，請參閱 Performance Insights API 參考 [DimensionGroup](#) 中的。

回應看起來類似以下的內容。

```
{
  "AlignedStartTime": 1540771200.0,
  "AlignedEndTime": 1540857600.0,
  "Identifier": "db-XXX",

  "MetricList": [ //11 entries in the MetricList
    {
      "Key": { //First key is total
        "Metric": "db.load.avg"
      }
      "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a
value
        {
          "Value": 1.6964980544747081,
          "Timestamp": 1540774800.0
        },
        //... 24 datapoints
      ]
    },
    {
      "Key": { //Next key is the top tokenized SQL
        "Dimensions": {
          "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
VALUES\n( nextval(?) ,?,?)",
          "db.sql_tokenized.db_id": "pi-2372568224",
          "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
```

```

    },
    "Metric": "db.load.avg"
  },
  "DataPoints": [ //... 24 datapoints
  ]
},
// In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
] //End of MetricList
} //End of response

```

此回應在 MetricList 中有 11 個項目 (1 個總計，前 10 個字符化的 SQL)，每個項目擁有 24 個每小時 DataPoints。

對於字符化的 SQL，每個維度清單中有三個項目：

- db.sql_tokenized.statement– 字符化的 SQL 陳述式。
- db.sql_tokenized.db_id – 參考 SQL 所用的原生資料庫 ID，或是無法使用原生資料庫 ID 時，績效詳情為您產生的合成 ID。此範例會傳回 pi-2372568224 合成 ID。
- db.sql_tokenized.id– 績效詳情中查詢的 ID。

在中 AWS Management Console，此識別碼稱為 Support 識別碼。之所以之所以命名，是因為 ID 是 Sup AWS port 人員可以檢查的資料，以協助您疑難排解資料庫問題的資料。AWS 非常重視數據的安全性和隱私性，幾乎所有數據都使用您的密 AWS KMS 鑰進行加密存儲。因此，裡面沒有人 AWS 可以查看這些數據。在先前的範例中，tokenized.statement 和 tokenized.db_id 都同時會以加密的形式存放。如果您的資料庫有問題，Sup AWS port 部門可以參考支 Support ID 來協助您。

進行查詢時，在 Group 中指定 GroupBy 可能會讓您省下不少心力。然而，如需對已傳回的資料進行更精細的控制，請指定維度的清單。例如，如果所需的是 db.sql_tokenized.statement，則可將 Dimensions 屬性新增至 query.json 檔案。

```

[
  {
    "Metric": "db.load.avg",
    "GroupBy": {
      "Group": "db.sql_tokenized",
      "Dimensions": ["db.sql_tokenized.statement"],
      "Limit": 10
    }
  }
]

```

]

擷取依據 SQL 篩選的資料庫負載平均值



先前的影像顯示已選取特定的查詢，最高平均作用中工作階段堆疊區域折線圖的範圍仍涵蓋至該查詢。雖然此查詢仍適用於前七個整體等待事件，系統仍會將回應值篩選出來。此篩選條件會在工作階段符合特定篩選條件時，才進行篩選。

此範例中的對應 API 查詢與 [擷取最高 SQL 的資料庫負載平均值](#) 中的命令類似。然而，查詢 query.json 檔案有以下內容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_event", "Limit": 5 },
    "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
  }
]
```

對於LinuxmacOS、或Unix：

```
aws pi get-resource-metrics \
```



```
--service-type RDS \  
--identifier db-ID \  
--start-time 2018-10-30T00:00:00Z \  
--end-time 2018-10-30T01:00:00Z \  
--period-in-seconds 60 \  
--metric-queries file://query.json
```

在 Windows 中：

```
aws pi get-resource-metrics ^  
--service-type RDS ^  
--identifier db-ID ^  
--start-time 2018-10-30T00:00:00Z ^  
--end-time 2018-10-30T01:00:00Z ^  
--period-in-seconds 60 ^  
--metric-queries file://query.json
```

回應看起來類似以下的內容。

```
{  
  "Identifier": "db-XXX",  
  "AlignedStartTime": 1556215200.0,  
  "MetricList": [  
    {  
      "Key": {  
        "Metric": "db.load.avg"  
      },  
      "DataPoints": [  
        {  
          "Timestamp": 1556218800.0,  
          "Value": 1.4878117913832196  
        },  
        {  
          "Timestamp": 1556222400.0,  
          "Value": 1.192823803967328  
        }  
      ]  
    },  
    {  
      "Key": {  
        "Metric": "db.load.avg",  
        "Dimensions": {  
          "db.wait_event.type": "io",
```

```

        "db.wait_event.name": "wait/io/aurora_redo_log_flush"
    }
},
"DataPoints": [
    {
        "Timestamp": 1556218800.0,
        "Value": 1.1360544217687074
    },
    {
        "Timestamp": 1556222400.0,
        "Value": 1.058051341890315
    }
]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "io",
            "db.wait_event.name": "wait/io/table/sql/handler"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.16241496598639457
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.05163360560093349
        }
    ]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "synch",
            "db.wait_event.name": "wait/synch/mutex/innodb/aurora_lock_thread_slot_futex"
        }
    },
    "DataPoints": [
        {

```

```
        "Timestamp": 1556218800.0,
        "Value": 0.11479591836734694
    },
    {
        "Timestamp": 1556222400.0,
        "Value": 0.013127187864644107
    }
]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "CPU",
            "db.wait_event.name": "CPU"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.05215419501133787
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.05805134189031505
        }
    ]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "synch",
            "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.017573696145124718
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.002333722287047841
        }
    ]
}
```

```

    }
  ]
}
],
  "AlignedEndTime": 1556222400.0
} //end of response

```

在此回應中，系統會根據 query.json file 檔案中指定的字符化 SQL AKIAIOSFODNN7EXAMPLE 的影響程度來篩選所有值。此索引鍵遵循的順序可能會與不含篩選條件的查詢不同，因為這是影響篩選 SQL 的前五個等待事件。

擷取 SQL 陳述式的完整文字

下列範例會擷取資料庫執行個體 db-10BCD2EFGHIJ3KL4M5N06PQRS5 之 SQL 陳述式的完整文字。--group 即為 db.sql，而 --group-identifier 即為 db.sql.id。在此範例中，*my-sql-id* 代表藉由呼叫 pi get-resource-metrics 或 pi describe-dimension-keys 擷取的 SQL ID。

執行下列命令。

對於LinuxmacOS、或Unix：

```

aws pi get-dimension-key-details \
  --service-type RDS \
  --identifier db-10BCD2EFGHIJ3KL4M5N06PQRS5 \
  --group db.sql \
  --group-identifier my-sql-id \
  --requested-dimensions statement

```

在 Windows 中：

```

aws pi get-dimension-key-details ^
  --service-type RDS ^
  --identifier db-10BCD2EFGHIJ3KL4M5N06PQRS5 ^
  --group db.sql ^
  --group-identifier my-sql-id ^
  --requested-dimensions statement

```

在此範例中，維度詳細資訊可供使用。因此，績效詳情會擷取 SQL 陳述式的完整文字，而不會將其截斷。

```
{
  "Dimensions": [
    {
      "Value": "SELECT e.last_name, d.department_name FROM employees e, departments d
WHERE e.department_id=d.department_id",
      "Dimension": "db.sql.statement",
      "Status": "AVAILABLE"
    },
    ...
  ]
}
```

建立一段時間區間的績效分析報告

下列範例會以 db-loadtest-0 資料庫的 1682969503 開始時間和 1682979503 結束時間建立績效分析報告。

```
aws pi create-performance-analysis-report \
  --service-type RDS \
  --identifier db-loadtest-0 \
  --start-time 1682969503 \
  --end-time 1682979503 \
  --region us-west-2
```

回應為 report-0234d3ed98e28fb17，是報告的唯一識別碼。

```
{
  "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

擷取績效分析報告

下列範例擷取 report-0d99cc91c4422ee61 報告的分析報告詳細資訊。

```
aws pi get-performance-analysis-report \
  --service-type RDS \
  --identifier db-loadtest-0 \
  --analysis-report-id report-0d99cc91c4422ee61 \
  --region us-west-2
```

回應會提供報告狀態、ID、時間詳細資料和見解。

```

{
  "AnalysisReport": {
    "Status": "Succeeded",
    "ServiceType": "RDS",
    "Identifier": "db-loadtest-0",
    "StartTime": 1680583486.584,
    "AnalysisReportId": "report-0d99cc91c4422ee61",
    "EndTime": 1680587086.584,
    "CreateTime": 1680587087.139,
    "Insights": [
      ... (Condensed for space)
    ]
  }
}

```

列出資料庫執行個體的所有績效分析報告

下列範例會列出 db-loadtest-0 資料庫所有可用的績效分析報告。

```

aws pi list-performance-analysis-reports \
--service-type RDS \
--identifier db-loadtest-0 \
--region us-west-2

```

回應將列出所有報告，其中包含報告 ID，狀態和時間區間的詳細資訊。

```

{
  "AnalysisReports": [
    {
      "Status": "Succeeded",
      "EndTime": 1680587086.584,
      "CreationTime": 1680587087.139,
      "StartTime": 1680583486.584,
      "AnalysisReportId": "report-0d99cc91c4422ee61"
    },
    {
      "Status": "Succeeded",
      "EndTime": 1681491137.914,
      "CreationTime": 1681491145.973,
      "StartTime": 1681487537.914,
      "AnalysisReportId": "report-002633115cc002233"
    }
  ]
}

```

```
    },
    {
      "Status": "Succeeded",
      "EndTime": 1681493499.849,
      "CreationTime": 1681493507.762,
      "StartTime": 1681489899.849,
      "AnalysisReportId": "report-043b1e006b47246f9"
    },
    {
      "Status": "InProgress",
      "EndTime": 1682979503.0,
      "CreationTime": 1682979618.994,
      "StartTime": 1682969503.0,
      "AnalysisReportId": "report-01ad15f9b88bcbd56"
    }
  ]
}
```

刪除績效分析報告

下列範例會刪除 db-loadtest-0 資料庫的績效分析報告。

```
aws pi delete-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--region us-west-2
```

將標籤新增至績效分析報告

下列範例會將帶有金鑰 name 和值 test-tag 的標籤新增至 report-01ad15f9b88bcbd56 報告。

```
aws pi tag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tags Key=name,Value=test-tag \
--region us-west-2
```

列出績效分析報告的所有標籤

下列範例會列出 report-01ad15f9b88bcbd56 報告的所有標籤。

```
aws pi list-tags-for-resource \  
--service-type RDS \  
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/  
report-01ad15f9b88bcbd56 \  
--region us-west-2
```

回應會列出新增至報告的所有標籤值和金鑰：

```
{  
  "Tags": [  
    {  
      "Value": "test-tag",  
      "Key": "name"  
    }  
  ]  
}
```

從績效分析報告中刪除標籤

下列範例會從 report-01ad15f9b88bcbd56 報告刪除 name 標籤。

```
aws pi untag-resource \  
--service-type RDS \  
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/  
report-01ad15f9b88bcbd56 \  
--tag-keys name \  
--region us-west-2
```

刪除標籤之後，呼叫 list-tags-for-resource API 不會列出此標籤。

使用 AWS CloudTrail 記錄績效詳情呼叫

績效詳情與 AWS CloudTrail 服務搭配運作，此服務會記錄使用者、角色或績效詳情中 AWS 服務所採取的動作。CloudTrail 會將績效詳情的所有 API 呼叫擷取為事件。此擷取包含來自 Amazon RDS 主控台的呼叫，以及從程式碼呼叫到績效詳情 API 操作的呼叫。

如果您建立線索，就可以讓 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括績效詳情的案件。即使您未設定權仗，依然可以透過 CloudTrail 主控台內的 Event history (事件歷史記錄) 檢視最新事件。使用由 CloudTrail 收集的資訊，您就可以判斷特定詳細資訊。此資訊包括對績效詳情提出的請求、提出請求的 IP 地址、何人提出請求，以及提出請求的時間。此外也包括其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [《AWS CloudTrail 使用者指南》](#)。

在 CloudTrail 中使用績效詳情資訊

當您建立帳戶時，系統即會在 AWS 帳戶中啟用 CloudTrail。當績效詳情中發生活動時，該活動會記錄在 CloudTrail 事件中，其他 AWS 服務事件則記錄於 CloudTrail 主控台的事件歷程記錄中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需更多詳細資訊，請參閱 AWS CloudTrail 使用者指南中的 [使用 CloudTrail 事件歷史記錄檢視事件](#)。

若要不持續記錄 AWS 帳戶的事件 (包括績效詳情事件)，請建立線索。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立線索時，線索會套用到所有 AWS 區域。權杖會記錄來自 AWS 分割區中所有 AWS 區域的事件，然後將記錄檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需更多詳細資訊，請參閱 AWS CloudTrail 使用者指南中的以下主題：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案](#)

所有績效詳情操作都是由 CloudTrail 所記錄並記載在 [績效詳情 API 參考](#) 中。例如，對 DescribeDimensionKeys 和 GetResourceMetrics 操作的呼叫都會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

績效詳情日誌檔案項目

權杖是一種組態，能讓事件以記錄檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。「事件」代表來自任何來源的單一請求。每個事件皆包含請求操作、操作日期和時間、請求參數等相關資訊。CloudTrail 日誌檔案並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範 GetResourceMetrics 操作：

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2019-12-18T19:28:46Z",
  "eventSource": "pi.amazonaws.com",
  "eventName": "GetResourceMetrics",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.198.67",
  "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",
  "requestParameters": {
    "identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
    "metricQueries": [
      {
        "metric": "os.cpuUtilization.user.avg"
      },
      {
        "metric": "os.cpuUtilization.idle.avg"
      }
    ]
  },
  "startTime": "Dec 18, 2019 5:28:46 PM",
  "periodInSeconds": 60,
  "endTime": "Dec 18, 2019 7:28:46 PM",
  "serviceType": "RDS"
},
"responseElements": null,
"requestID": "9ffbe15c-96b5-4fe6-bed9-9fccff1a0525",
"eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

使用 Amazon RDS 專用的 Amazon DevOps 大師分析性能異常

Amazon DevOps Guru 是全受管的營運服務，可協助開發人員和操作員改善其應用程式的效能和可用性。DevOpsGuru 卸載了與識別操作問題相關的任務，以便您可以快速實施建議以改善應用程序。有關更多信息，請參閱[什麼是 Amazon DevOps 大師？](#) 在 Amazon 大 DevOps 師用戶指南。

DevOpsGuru 可偵測、分析所有 Amazon RDS 資料庫引擎的現有操作問題，並提出建議。DevOps 這些監控功能可讓 DevOps Guru for RDS 偵測和診斷效能瓶頸，並建議特定的修正動作。DevOps 適用於 RDS 的大師也可以在 庫 RDS 版 PostgreSQL 資料庫中偵測有問題的狀況，然後再發生這些問題。

您現在可以在 RDS 主控台中檢視這些建議。如需詳細資訊，請參閱[檢視和回應 Amazon RDS 建議](#)。

以下視頻是 RDS 大 DevOps 師的概述。

有關此主題的深入探討，請參閱[引擎蓋下的 RDS Amazon DevOps 大師](#)。

主題

- [RDS 大 DevOps 師的好處](#)
- [RDS 的 DevOps 大師如何工作](#)
- [為 RDS 設定 DevOps 大師](#)

RDS 大 DevOps 師的好處

如果您負責 RDS for PostgreSQL 資料庫，您可能不知道已發生事件或退化而正在影響該資料庫。得知問題時，您可能不知道為何發生或如何處理。您可以遵循 DevOps Guru for RDS 的建議，而不是向資料庫管理員 (DBA) 尋求協助或依賴第三方工具。

您可以從 RDS 的 DevOps 大師的詳細分析中獲得以下優勢：

快速診斷

DevOpsRDS 大師持續監視和分析資料庫遙測。Performance Insights、增強型監控和 Amazon 會 CloudWatch 收集資料庫執行個體的遙測資料。DevOpsGuru for RDS 使用統計和機器學習技術來挖掘這些數據並檢測異常。若要進一步了解遙測資料，請參閱，以及《Amazon RDS 使用者指南》中的[在 Amazon RDS 以績效詳情監控資料庫負載](#)和[以增強型監控來監控 OS 指標](#)。

快速解決

每個異常都指出效能問題，並建議調查途徑或更正行動。例如，RDS 版 DevOps Guru 可能會建議您調查特定的等待事件。或者，可能建議您調整應用程式集區設定，以限制資料庫連線的數目。採用這些建議，解決效能問題會比手動疑難排解更快。

主動式洞察

DevOpsGuru for RDS 使用資源中的指標來檢測潛在問題的行為，然後再成為更大的問題。例如，它可以偵測資料庫何時使用越來越多磁碟上暫存資料表，因為這可能會影響效能。DevOps 然後，Guru 提供建議以幫助您在問題變得更大的問題之前解決問題。

Amazon 工程師和機器學習的深厚知識

為了偵測效能問題並協助您解決瓶頸，DevOpsGuru for RDS 仰賴機器學習 (ML) 和進階數學公式。Amazon 資料庫工程師為開發 RDS 的 DevOps Guru 做出了貢獻，其中包含了管理數十萬個資料庫的多年。通過利用這個集體知識，DevOpsGuru for RDS 可以教你最佳實踐。

RDS 的 DevOps 大師如何工作

DevOps 適用於 RDS 的專家會從 Amazon RDS Performance Insights 收集關於您的 RDS 資料庫的資料。最重要的指標是 DBLoad。DevOpsGuru for RDS 會使用 Performance Insights 指標，使用機器學習進行分析，並將見解發佈到儀表板。

洞察力是 DevOps Guru 檢測到的相關異常的集合。

在適用於 RDS 的 DevOps 大師中，異常是一種模式，該模式會偏離您的 RDS 版 PostgreSQL 資料庫的正常效能。

主動式洞察

主動洞察可讓您在異常行為發生前了解該行為。它包含具有建議和相關指標的異常情況，可幫助您在問題擴大之前解決 RDS for PostgreSQL 資料庫的問題。這些見解發佈在 DevOps Guru 儀表板中。

例如，DevOps 大師可能會偵測到您的 RDS for PostgreSQL 資料庫正在建立許多磁碟上暫存資料表。若未解決問題，之後可能導致效能問題。每個主動洞察都包含修正行為的建議，以及 [使用 Amazon DevOps Guru 主動洞察，調校 RDS for PostgreSQL](#) 中相關主題的連結。如需詳細資訊，請參閱 Amazon DevOps Guru 使用者指南中的 DevOps Guru [中的深入解析](#)。

反應式洞察

反應式洞察會在發生異常行為時有效識別。如果 RDS DevOps 專家在您的 RDS 版 PostgreSQL 資料庫執行個體中發現效能問題，它會在 DevOps 大師儀表中發佈反應式洞察。如需詳細資訊，請參閱 Amazon DevOps Guru 使用者指南中的 DevOps Guru [中的深入解析](#)。

因果異常

因果異常是反應式洞察中最高等級的異常。資料庫負載 (資料庫負載) 是 RDS DevOps 大師的因果異常。

異常指派高、中、低的嚴重性層級來測量效能影響。若要進一步了解，請參閱 Amazon DevOps 大師 DevOps 師使用者指南 [中適用於 RDS 的重要概念](#)。

如果 DevOps Guru 偵測到資料庫執行個體目前發生異常，則會在 RDS 主控台的 [資料庫] 頁面中收到警示。主控台還會提醒您過去 24 小時內發生的異常。若要從 RDS 主控台移至異常頁面，請選擇提醒訊息中的連結。RDS 主控台也會在頁面中提醒您注意 RDS for PostgreSQL 資料庫執行個體。

情境異常

情境異常是資料庫負載內的研究結果，與反應式洞察相關。每個情境異常描述一個需要調查的特定 RDS for PostgreSQL 效能問題。例如，RDS 的 DevOps Guru 可能會建議您考慮增加 CPU 容量，或調查導致資料庫負載的等待事件。

Important

建議您先在測試執行個體上測試任何變更，然後再修改生產執行個體。如此就可以了解變更的影響。

若要進一步了解，請參閱 Amazon DevOps 大師使用者指南 [中的 Amazon RDS 中的分析異常情況](#)。

為 RDS 設定 DevOps 大師

若要允許 Amazon RDS DevOps 大師針對版 PostgreSQL 資料庫發佈見解，請完成以下任務。

主題

- [設定 RDS DevOps 專用的 IAM 存取政策](#)
- [對 RDS for PostgreSQL 資料庫執行個體開啟績效詳情](#)
- [開啟 DevOps Guru 並指定資源涵蓋範圍](#)

設定 RDS DevOps 專用的 IAM 存取政策

若要在 RDS 主控台中檢視 DevOps Guru 的警示，您的 AWS Identity and Access Management (IAM) 使用者或角色必須具有下列其中一項政策：

- AWS 受管理的策略 AmazonDevOpsGuruConsoleFullAccess
- 受 AWS 管理的策略 AmazonDevOpsGuruConsoleReadOnlyAccess 及下列其中一項策略：
 - AWS 受管理的策略 AmazonRDSFullAccess
 - 包含 `pi:GetResourceMetrics` 和 `pi:DescribeDimensionKeys` 的客戶受管政策

如需詳細資訊，請參閱 [設定績效詳情的存取政策](#)。

對 RDS for PostgreSQL 資料庫執行個體開啟績效詳情

DevOpsRDS 的大師依賴於 Performance Insights 其數據。如果沒有 Performance Insights，DevOpsGuru 會發佈異常情況，但不包括詳細的分析和建議。

在建立或修改 RDS for PostgreSQL 資料庫執行個體時，您可以啟用績效詳情。如需詳細資訊，請參閱 [開啟和關閉 Amazon RDS 的 Performance Insights](#)。

開啟 DevOps Guru 並指定資源涵蓋範圍

您可以開啟 DevOps 大師，讓它以下列其中一種方式監控您的 RDS 資料庫。

主題

- [在 RDS 主控台中開啟 DevOps 大師](#)
- [在大師主控台中新增適用於 PostgreSQL 的 RDS 資源](#)
- [使用新增適用於 PostgreSQL 的 RDS 資源 AWS CloudFormation](#)

在 RDS 主控台中開啟 DevOps 大師

您可以在 Amazon RDS 主控台中採取多個路徑來開啟 DevOps 大師。

主題

- [當您建立 RDS 資料庫時開啟 DevOps 大師](#)
- [從通知橫幅打開 DevOps Guru](#)
- [開啟 DevOps Guru 時回應權限錯誤](#)

當您建立 RDS 資料庫時開啟 DevOps 大師

建立工作流程包括開啟資料庫的 DevOps Guru 涵蓋範圍的設定。當您選擇 Production (生產) 範本時，此設定預設為開啟。

若要在建立 RDS 資料庫時 PostgreSQL DevOps 大師

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 請按照 [建立資料庫執行個體](#) 中的步驟進行，一直執行到 (但不包括) 您選擇監控設定的步驟。
3. 在 Monitoring (監控) 中，選擇 Turn on Performance Insights (開啟績效詳情)。若要讓 DevOps Guru for RDS 提供效能異常的詳細分析，必須開啟 Performance Insights。
4. 選擇開啟 DevOps 大師。

Monitoring

Turn on Performance Insights [Info](#)

Retention period for Performance Insights [Info](#)


7 days (free tier) ▼

AWS KMS key [Info](#)

(default) aws/rds ▼

Account
159066061753


KMS key ID
f08a73b3-0cad-44ee-96de-d4bc21629583

 You can't change the KMS key after enabling Performance Insights.

Turn on DevOps Guru [Info](#)

DevOps Guru for RDS automatically detects performance anomalies for DB instances and provides recommendations.

Tag key	Tag value
devops-guru-default	database-29

Cost per resource per hour
\$0.0042 [Amazon DevOps Guru pricing](#) 

5. 為您的數據庫創建一個標籤，以便 DevOps Guru 可以對其進行監視。請執行下列操作：

- 在 Tag key (標籤鍵) 的文字欄位中，輸入開頭為 **Devops-Guru-** 的名稱。
- 在 Tag value (標籤值) 的文字欄位中，輸入任何一個值。例如，如果您輸入 **rds-database-1** 作為 RDS for PostgreSQL 資料庫的名稱，您也可以輸入 **rds-database-1** 作為標籤值。

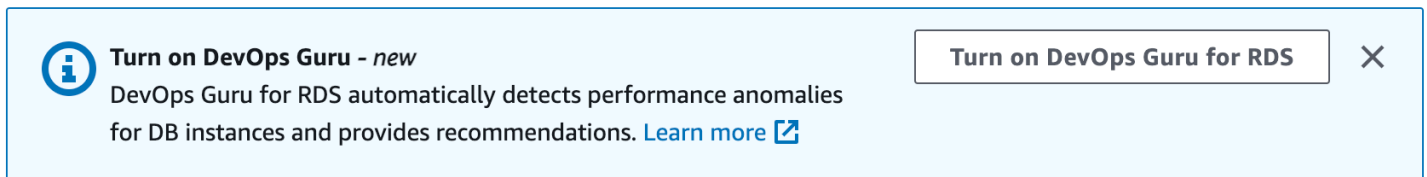
如需有關標籤的詳細資訊，請參閱 Amazon Guru 使用者指南中的「使用標籤來識別 DevOps G DevOps uru [應用程式中的資源](#)」。

6. 完成[建立資料庫執行個體](#)中剩餘的步驟。

從通知橫幅打開 DevOps Guru

如果 DevOps Guru 不涵蓋您的資源，Amazon RDS 會在下列位置使用橫幅通知您：

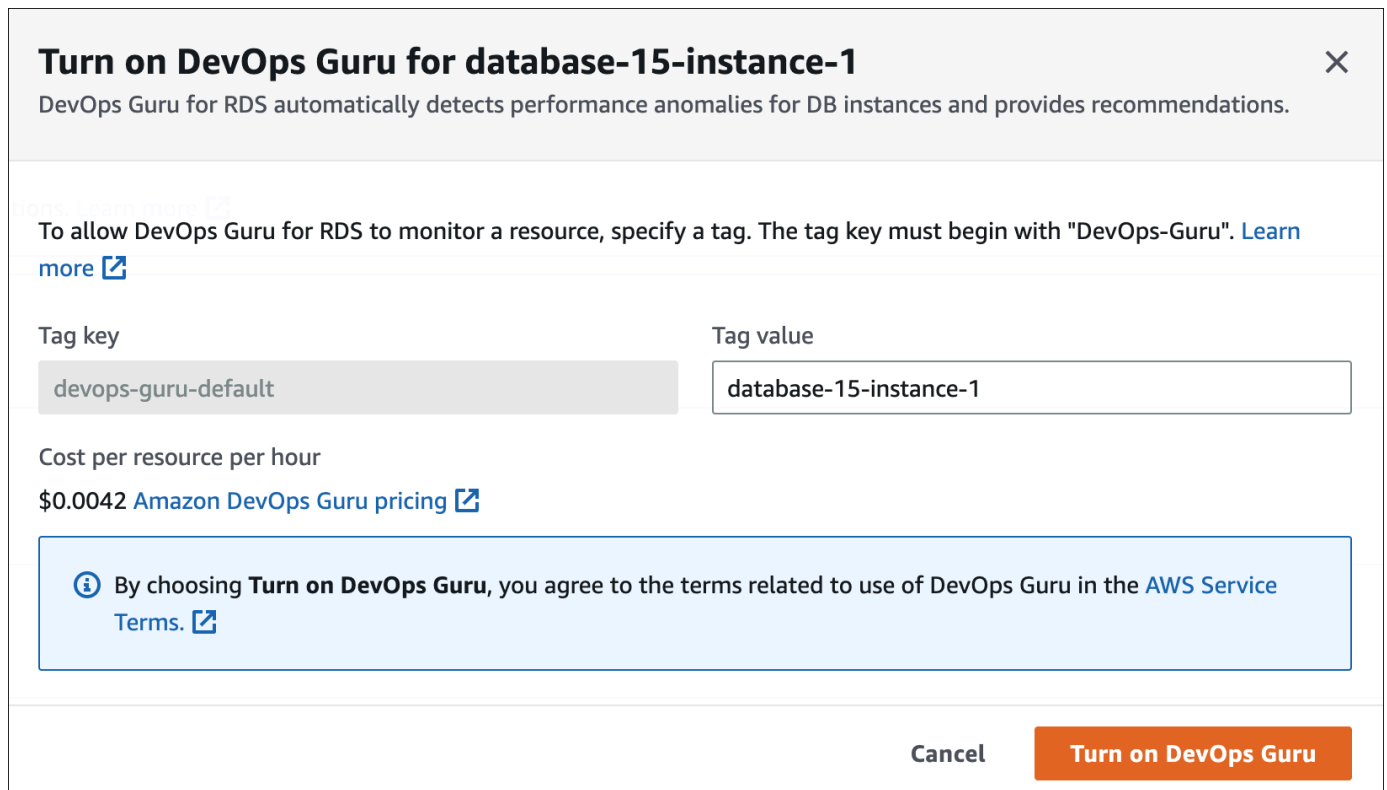
- 資料庫叢集執行個體的 Monitoring (監控) 索引標籤
- 績效詳情儀表板



The screenshot shows a notification banner with a blue border. On the left is an information icon (i) in a blue circle. To its right is the text: "Turn on DevOps Guru - new" followed by "DevOps Guru for RDS automatically detects performance anomalies for DB instances and provides recommendations. [Learn more](#)". On the right side of the banner is a button that says "Turn on DevOps Guru for RDS" and a close icon (X).

為您的 資料庫開 PostgreSQL DevOps 大師

1. 在橫幅中，選擇 [開啟 RDS DevOps 大師]。
2. 輸入標籤金鑰名稱與值。如需有關標籤的詳細資訊，請參閱 Amazon Guru 使用者指南中的「使用標籤來識別 DevOps G DevOps uru [應用程式中的資源](#)」。



The screenshot shows a dialog box titled "Turn on DevOps Guru for database-15-instance-1" with a close icon (X) in the top right. Below the title is the text: "DevOps Guru for RDS automatically detects performance anomalies for DB instances and provides recommendations." Below this is a section with a blue header "To allow DevOps Guru for RDS to monitor a resource, specify a tag. The tag key must begin with "DevOps-Guru". [Learn more](#)". There are two input fields: "Tag key" with the value "devops-guru-default" and "Tag value" with the value "database-15-instance-1". Below the fields is the text "Cost per resource per hour" followed by "\$0.0042 [Amazon DevOps Guru pricing](#)". At the bottom left is a blue box with an information icon (i) and the text: "By choosing **Turn on DevOps Guru**, you agree to the terms related to use of DevOps Guru in the [AWS Service Terms](#)". At the bottom right are two buttons: "Cancel" and "Turn on DevOps Guru".

3. 選擇開啟 DevOps大師。

開啟 DevOps Guru 時回應權限錯誤

如果您在建立資料庫時從 RDS 主控台開啟 DevOps Guru，RDS 可能會顯示下列關於遺失權限的標題。



⊗ Failed to turn on DevOps Guru for database-9-instance-1 because of missing permissions ×
Add permissions to your IAM policy to enable DevOps Guru. [Learn more](#)

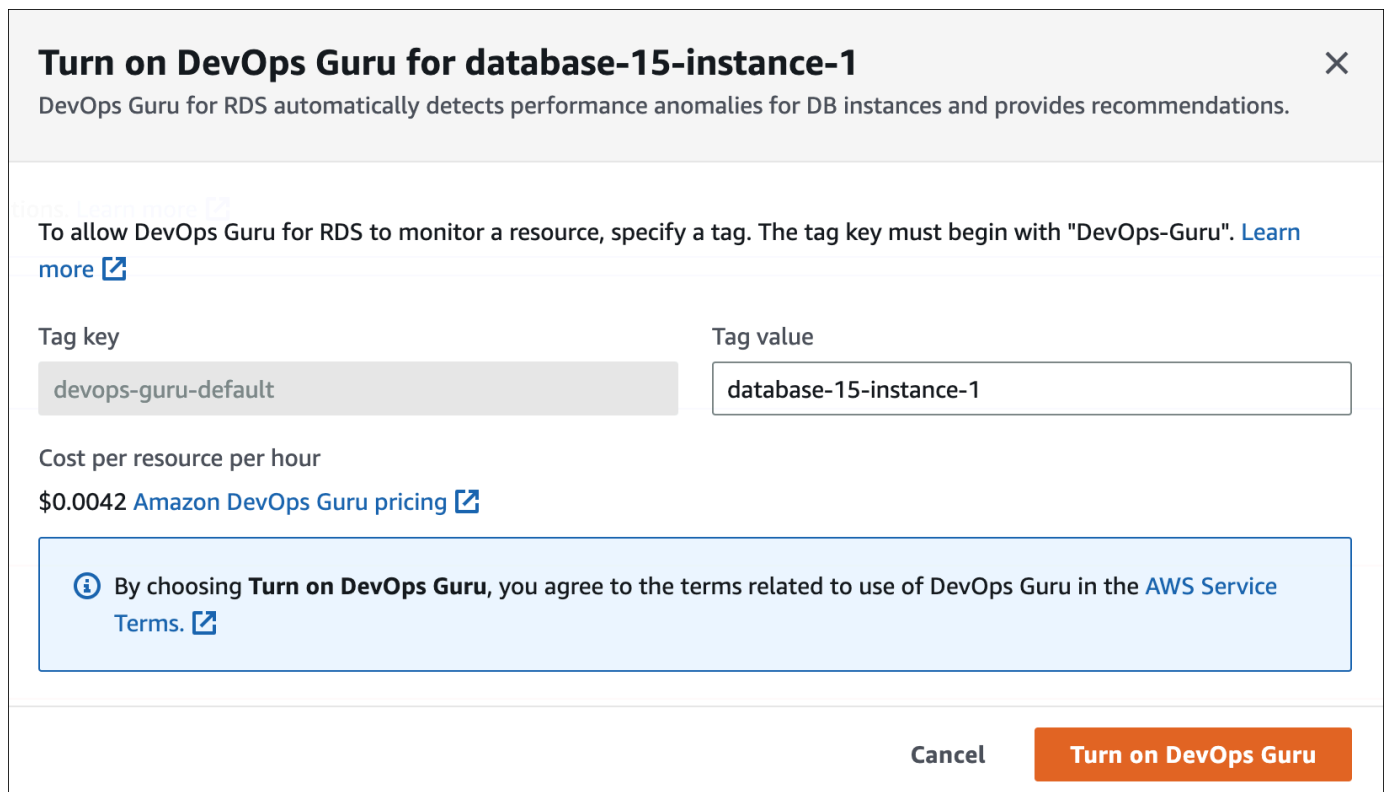
回應許可錯誤

1. 將使用者管理的角色 AmazonDevOpsGuruConsoleFullAccess 授予您的 IAM 使用者或角色。如需詳細資訊，請參閱 [設定 RDS DevOps 專用的 IAM 存取政策](#)。
2. 開啟 RDS 主控台。
3. 在導覽窗格中，選擇 Performance Insights (績效詳情)。
4. 在叢集中選擇您剛剛建立的資料庫執行個體。
5. 選擇開關以開啟 RDS DevOps 專用大師。



DevOps Guru for RDS

6. 選擇一個標籤值。如需詳細資訊，請參閱 Amazon Guru 使用者指南中的「使用標籤來識別 DevOps G DevOps uru [應用程式中的資源](#)」。



Turn on DevOps Guru for database-15-instance-1 ×
DevOps Guru for RDS automatically detects performance anomalies for DB instances and provides recommendations.

To allow DevOps Guru for RDS to monitor a resource, specify a tag. The tag key must begin with "DevOps-Guru". [Learn more](#)

Tag key	Tag value
<input type="text" value="devops-guru-default"/>	<input type="text" value="database-15-instance-1"/>

Cost per resource per hour
\$0.0042 [Amazon DevOps Guru pricing](#)

i By choosing **Turn on DevOps Guru**, you agree to the terms related to use of DevOps Guru in the [AWS Service Terms](#).

Cancel **Turn on DevOps Guru**

7. 選擇開啟 DevOps大師。

在大師主控台中新增適用於 PostgreSQL 的 RDS 資源

您可以在 DevOps Guru 主控台上指定您的 DevOps Guru 資源涵蓋範圍。請依照 Amazon DevOps Guru 使用者指南 [中指定您的大師資源涵蓋範圍](#) 中所述的步驟進行。在編輯分析的資源時，請選擇下列其中一個選項：

- 選擇 [所有帳戶資源] 以分析您 AWS 帳戶 和區域中所有支援的資源，包括適用於 PostgreSQL 的 RDS 資料庫。
- 選擇 CloudFormation 堆疊以分析您選擇的堆疊中的 RDS 版 PostgreSQL 資料庫。如需詳細資訊，請參閱 Amazon Guru 使用者指南中的使用 AWS CloudFormation 堆疊來識別 DevOps G DevOps uru [應用程式中的資源](#)。
- 選擇標籤分析已標記的 RDS for PostgreSQL 資料庫。如需詳細資訊，請參閱 Amazon Guru 使用者指南中的使用標籤識別 DevOps G DevOps uru [應用程式中的資源](#)。

如需詳細資訊，請參閱 Amazon DevOps G DevOps uru 使用者指南 [中的啟用大師](#)。

使用新增適用於 PostgreSQL 的 RDS 資源 AWS CloudFormation

您可以使用標籤，將 RDS 資源的涵蓋 PostgreSQL 圍新增至範本。 CloudFormation

若要使用標籤 A 定適用於 PostgreSQL 的資料庫執行個體 CloudFormation

1. 在資料庫執行個體的 CloudFormation 範本中，使用索引鍵/值組定義標籤。

下列範例會將值 my-db-instance1 指派給 RDS for PostgreSQL 資料庫執行個體的 Devops-guru-cfn-default。

```
MyDBInstance1:
  Type: "AWS::RDS::DBInstance"
  Properties:
    DBInstanceIdentifier: my-db-instance1
    Tags:
      - Key: Devops-guru-cfn-default
        Value: devopsguru-my-db-instance1
```

2. 在 DevOps Guru 堆疊的 CloudFormation 範本中，在資源集合篩選器中指定相同的標籤。

下列範例會將 DevOps Guru 設定為具有標籤值的資源提供涵蓋範圍。my-db-instance1

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
            - "devopsguru-my-db-instance1"
```

下列範例提供應用程式邊界 Devops-guru-cfn-default 內所有資源的涵蓋範圍。

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
            - "*"

```

如需詳細資訊，請參閱使用者指南中的[AWS::DevOpsGuru::ResourceCollection](#)和 [AWS::RDS::資料庫執行個體](#) AWS CloudFormation。

使用增強型監控來監控作業系統指標

使用增強型監控可即時監控資料庫執行個體的作業系統。當您想查看不同的程序或執行緒如何使用 CPU 時，增強型監控指標很有用。

主題

- [增強型監視概觀](#)
- [設定並啟用增強型監控](#)
- [在 RDS 主控台中檢視作業系統指標](#)
- [使用 CloudWatch Logs 檢視作業系統指標](#)

增強型監視概觀

Amazon RDS 針對資料庫執行個體執行所在的作業系統 (OS) 即時提供指標。您可以在主控台上檢視 RDS 資料庫執行個體的所有系統指標和處理程序資訊。您可以管理要為每個執行個體監控的指標，並根據需求自訂儀表板。如需增強型監控指標的說明，請參閱 [增強型監控中的作業系統指標](#)。

RDS 將增強型監控中的指標交付到您的 Amazon CloudWatch 日誌帳戶中。您可以 CloudWatch 從 CloudWatch 日誌中創建指標過濾器，並在 CloudWatch 儀表板上顯示圖形。您可以在您選擇的監控系統中使用 CloudWatch 記錄中的增強型監控 JSON 輸出。如需詳細資訊，請參閱「Amazon RDS 常見問答集」中的 [增強型監控](#)。

主題

- [增強型監控可用性](#)
- [CloudWatch 和增強型監控指標之間的差異](#)
- [保留增強型監控指標](#)
- [增強型監控的成本](#)

增強型監控可用性

Enhanced Monitoring 可用於下列資料庫引擎：

- Db2
- MariaDB
- Microsoft SQL Server

- MySQL
- Oracle
- PostgreSQL

增強型監控可用於所有資料庫執行個體類別 (db.m1.small 執行個體類別除外)。

CloudWatch 和增強型監控指標之間的差異

hypervisor 會建立並執行虛擬機器 (VM)。使用虛擬化管理程序，執行個體可藉由虛擬化共用記憶體和 CPU 來支援多個客體 VM。CloudWatch 從資料庫執行個體的虛擬化管理程序收集 CPU 使用率的相關指標。相反地，增強型監控會從資料庫執行個體上的代理程式中收集其指標。

您可能會發現 Hypervisor 層執行少量工作，因此您可能會發現「增強型監控」測量值 CloudWatch 和「增強型監視」測量之間 如果您的資料庫執行個體使用較小的執行個體類別，差異可能會更大。在此案例中，單一實體執行個體上的 Hypervisor 層可能會管理更多虛擬機器 (VM)。

如需增強型監控指標的說明，請參閱 [增強型監控中的作業系統指標](#)。如需有關指 CloudWatch 標的詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

保留增強型監控指標

根據預設，「增強型監控」指標會在 CloudWatch 記錄檔中儲存 30 天。此保留期間與典型 CloudWatch 度量不同。

若要修改指標儲存在 CloudWatch 記錄檔中的時間長度，請在 CloudWatch 主控台中變更 `RDSOSMetrics` 錄群組的保留。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的變更日誌中的 CloudWatch 日誌 [資料保留](#)。

增強型監控的成本

增強型監控指標會儲存在 CloudWatch 記錄檔中，而非 CloudWatch 指標中。增強型監控的成本取決於下列因素：

- 只有在您超過 Amazon CloudWatch Logs 提供的免費方案時，才會向您收取增強型監控費用。費用是根據 CloudWatch 記錄資料傳輸和儲存費率計算。
- 針對 RDS 執行個體傳輸的資訊量與針對「增強型監控」功能定義的精密度成正比。較短的監控時間間隔會導致較頻繁的作業系統指標報告，並增加您的監控成本。若要管理成本，請針對帳戶中的不同執行個體設定不同的精密度。

- 增強型監控的使用成本將套用至已啟用增強型監控的每個資料庫執行個體。監控大量資料庫執行個體的成本比監控少量資料庫執行個體昂貴得多。
- 支援運算更為密集之工作負載的資料庫執行個體，有更多的作業系統處理程序活動可以報告，其增強型監控的成本也比較高。

如需有關定價的詳細資訊，請參閱 [Amazon CloudWatch 定價](#)。

設定並啟用增強型監控

若要使用增強型監控，您必須建立 IAM 角色，然後啟用增強型監控。

主題

- [為增強型監控建立 IAM 角色](#)
- [開啟和關閉增強型監控](#)
- [防範混淆代理人問題](#)

為增強型監控建立 IAM 角色

增強型監控需要代表您採取行動的權限，才能將作業系統指標資訊傳送至 CloudWatch 記錄。您可以使用 AWS Identity and Access Management (IAM) 角色授予增強型監控權限。您可以在啟用增強型監控時建立此角色，也可以事先建立。

主題

- [啟用增強型監控時建立 IAM 角色](#)
- [在啟用增強型監控之前建立 IAM 角色](#)

啟用增強型監控時建立 IAM 角色

當您在 RDS 主控台中啟用增強型監控時，Amazon RDS 可以為您建立所需的 IAM 角色。角色已命名 `rds-monitoring-role`。RDS 會將此角色用於指定的資料庫執行個體、僅供讀取複本或多可用區域資料庫叢集。

啟用增強型監控時建立 IAM 角色

1. 請遵循 [開啟和關閉增強型監控](#) 中的步驟。
2. 在您選擇角色的步驟中，將「Monitoring Role (監控角色)」設定為「Default (預設)」。

在啟用增強型監控之前建立 IAM 角色

您可以在啟用增強型監控之前建立必要的角色。啟用增強型監控時，請指定新角色的名稱。如果您使用 AWS CLI 或 RDS API 啟用增強型監控，您將必須建立此必要的角色。

必須將 PassRole 許可授予會啟用增強型監控的使用者。如需詳細資訊，請參閱 [IAM 使用者指南中的授與使用者將角色傳遞給 AWS 服務](#) 的權限中的範例 2。

為 Amazon RDS 增強型監控建立 IAM 角色

1. 前往 <https://console.aws.amazon.com> 開啟 [IAM 主控台](#)
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 選擇 AWS service (AWS 服務) 索引標籤，然後從服務清單中選擇 RDS。
5. 選擇 RDS - Enhanced Monitoring (RDS - 增強型監控)，然後選擇 Next (下一步)。
6. 確定權限原則顯示 AmazonRDS EnhancedMonitoringRole，然後選擇 [下一步]。
7. 針對 Role name (角色名稱)，輸入您的角色名稱。例如，輸入 **emaccess**。

您角色的受信任實體是 AWS 服務監控。

8. 選擇建立角色。

開啟和關閉增強型監控

您可以使用 AWS Management Console、AWS CLI 或 RDS API 開啟和關閉增強型監控。您可以選擇您要開啟增強型監控所在的 RDS 資料庫執行個體。您可以針對每個資料庫執行個體上的指標收集設定不同的精密度。

主控台

您可以在建立資料庫執行個體、多可用區域資料庫叢集或僅供讀取複本時，或在您修改資料庫執行個體或多可用區域資料庫叢集時，開啟增強型監控。如果您修改資料庫執行個體以開啟增強型監控，您的資料庫執行個體無需重新啟動，改變即可生效。

您在 Databases (資料庫) 頁面上執行以下任一動作時，可在 RDS 主控台中開啟增強型監控：

- 建立資料庫執行個體或多可用區域資料庫叢集：選擇 Create database (建立資料庫)。
- 建立僅供讀取複本 – 選擇 Actions (動作)，然後選 Create read replica (建立僅供讀取複本)。
- 修改資料庫執行個體或多可用區域資料庫叢集：選擇 Modify (修改)。

在 RDS 主控台中開啟或關閉增強型監控

1. 捲動至 Additional configuration (其他組態)。
2. 在 Monitoring (監控) 中，為資料庫執行個體或僅供讀取複本選擇 Enable Enhanced Monitoring (啟用增強型監控)。若要關閉增強型監控，請選擇 Disable enhanced monitoring (停用增強型監控)。
3. 將監控角色屬性設定為您建立的 IAM 角色，以允許 Amazon RDS 為您與 Amazon CloudWatch 日誌通訊，或選擇預設讓 RDS 為您指定的角色建立一個角色 `rds-monitoring-role`。
4. 系統會收集資料庫執行個體或僅供讀取複本的指標，請將 Granularity (精細程度) 屬性設定為該資料點的時間間隔 (以秒為單位)。Granularity (精細程度) 屬性可設定為以下其中一個值：1、5、10、15、30 或 60。

RDS 主控台重新整理的最快時間為每 5 秒一次。如果您在 RDS 主控台中，將精細程度設定為 1 秒，您還是只能每 5 秒鐘才能看到更新後的指標。您可以使用 CloudWatch 記錄擷取 1 秒的指標更新。

AWS CLI

若要使用開啟增強型監控 AWS CLI，請在下列命令中將 `--monitoring-interval` 選項設定為值以外的值，0 並將 `--monitoring-role-arn` 選項設定為您在中建立的角色 [為增強型監控建立 IAM 角色](#)。

- [create-db-instance](#)
- [create-db-instance-read-複製品](#)
- [modify-db-instance](#)
- [create-db-cluster](#)(異地同步備份資料庫叢集)
- [modify-db-cluster](#)(異地同步備份資料庫叢集)

`--monitoring-interval` 為資料庫執行個體收集增強型監控指標點之間的時間 (秒)。選項的有效值為 0、1、5、10、15、30 和 60。

若要使用關閉增強型監視 AWS CLI，請 0 在這些指令中將 `--monitoring-interval` 選項設定為。

Example

下方範例會為資料庫執行個體開啟增強型監控：

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \
```

```
--db-instance-identifier mydbinstance \  
--monitoring-interval 30 \  
--monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

在 Windows 中：

```
aws rds modify-db-instance ^  
--db-instance-identifier mydbinstance ^  
--monitoring-interval 30 ^  
--monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Example

下方範例會為多可用區域資料庫叢集開啟增強型監控：

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
--db-cluster-identifier mydbcluster \  
--monitoring-interval 30 \  
--monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

在 Windows 中：

```
aws rds modify-db-cluster ^  
--db-cluster-identifier mydbcluster ^  
--monitoring-interval 30 ^  
--monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

RDS API

若要使用 RDS API 開啟增強型監控，請將 `MonitoringInterval` 參數設定為 0 以外的值，然後將 `MonitoringRoleArn` 參數設定為您為 [為增強型監控建立 IAM 角色](#) 中建立的角色。在下列動作中設定這些參數：

- [CreateDBInstance](#)
- [創建數據庫 InstanceReadReplica](#)
- [ModifyDBInstance](#)
- [CreateDBCluster](#) (多可用區域資料庫叢集)
- [ModifyDBCluster](#) (多可用區域資料庫叢集)

`MonitoringInterval` 參數會指定資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。有效值為：0、1、5、10、15、30 和 60。

若要使用 RDS API 關閉增強型監控，請將 `MonitoringInterval` 設定為 0。

防範混淆代理人問題

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆的副問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況，AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。如需詳細資訊，請參閱[混淆代理人問題](#)。

若要限制 Amazon RDS 可將資源提供給另一項服務的許可，建議針對增強型監控角色，在信任政策中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵。如果您同時使用兩個全域條件內容索引鍵，兩者必須使用相同的帳戶 ID。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，以及資源的完整 ARN。為 Amazon RDS 將 `aws:SourceArn` 設為 `arn:aws:rds:Region:my-account-id:db:dbname`。

下列範例展示如何在信任政策中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵，來預防混淆代理人問題。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitoring.rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:rds:Region:my-account-id:db:dbname"
        },
        "StringEquals": {
          "aws:SourceAccount": "my-account-id"
        }
      }
    }
  ]
}
```

```

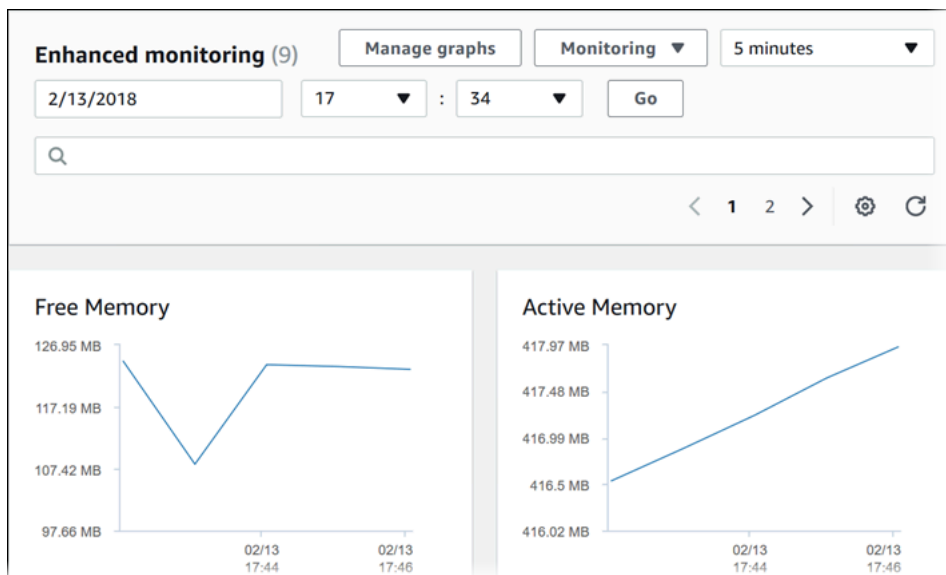
}
]
}

```

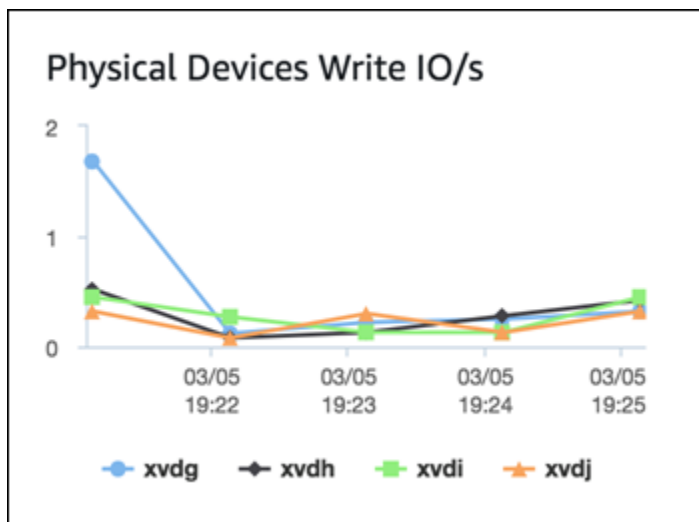
在 RDS 主控台中檢視作業系統指標

您可在 RDS 主控台的 Monitoring (監控) 選擇 Enhanced monitoring (增強型監控)，以檢視增強型監控回報的作業系統指標。

下列範例會顯示「增強型監控」頁面。如需增強型監控指標的說明，請參閱 [增強型監控中的作業系統指標](#)。



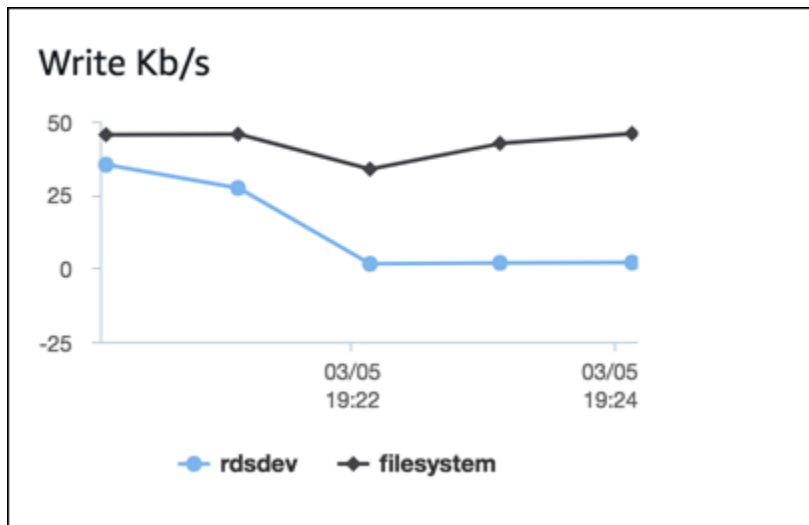
部分資料庫執行個體使用一個以上磁碟，作為資料庫執行個體的資料儲存磁碟區。Physical Devices (實體裝置) 圖表會針對這類資料庫執行個體，顯示每個磁碟的指標。例如下圖顯示四個磁碟的指標。



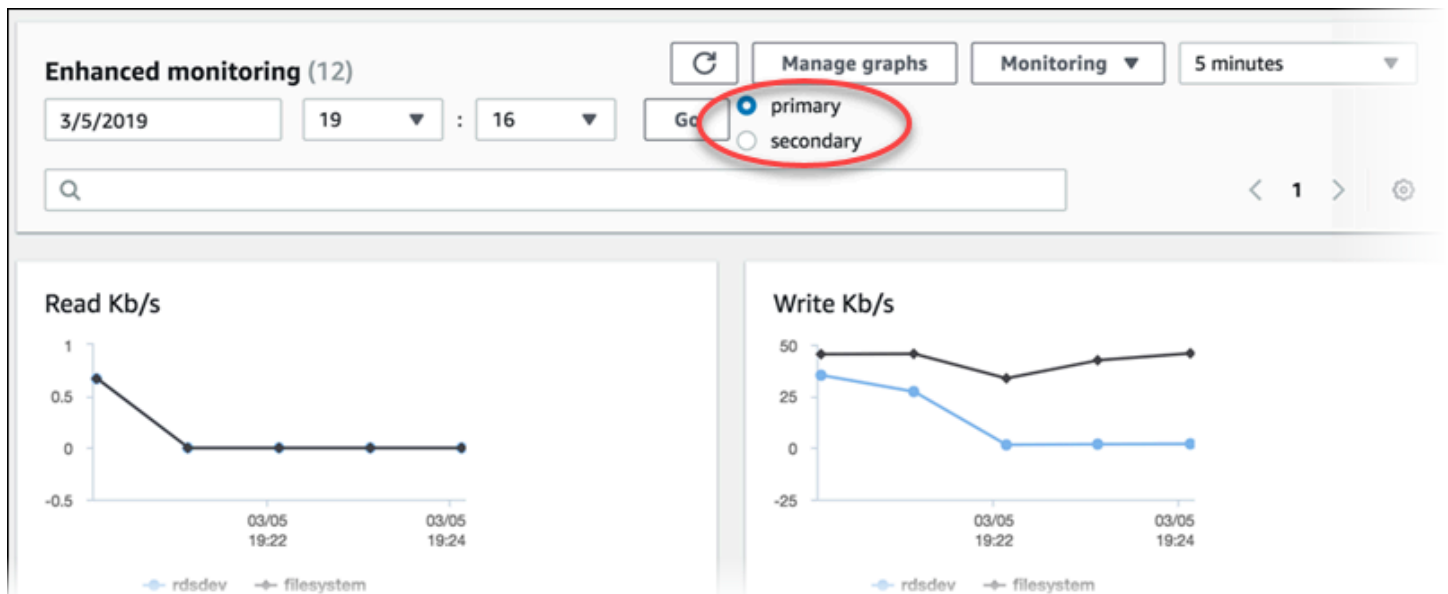
Note

目前 Physical Devices (實體裝置) 圖表不適用於 Microsoft SQL Server 資料庫執行個體。

您在檢視彙總的 Disk I/O (磁碟輸入/輸出) 及 File system (檔案系統) 圖表時，rdsdev 裝置與 / rdsdbdata 檔案系統有關，所有資料庫檔案和記錄皆儲存於其中。filesystem 裝置與 / 檔案系統 (也稱為根目錄) 有關，其中儲存作業系統相關資料。



如果資料庫執行個體為異地同步備份部署，您可以檢視主要資料庫執行個體的作業系統指標，及其異地同步備份備用複本。在 Enhanced monitoring (增強型監控) 檢視畫面中，選擇 primary (主要) 可檢視主要資料庫執行個體的作業系統指標，選擇 secondary (次要) 可檢視備用複本的作業系統指標。



如需異地同步備份部署的詳細資訊，請參閱[設定及管理多可用區部署](#)。

Note

MariaDB 資料庫執行個體目前並不支援檢視多可用區域備用複本的作業系統指標。

如果您要檢視資料庫執行個體上執行的處理程序，則請在 Monitoring (監控) 中選擇 OS process list (作業系統程序清單)，即可查看詳細資訊。

Process List (程序清單) 檢視畫面顯示如下。

NAME	VIRT	RES	CPU%	MEM%	VMLIMIT
postgres [3181]!	283.55 MB	17.11 MB	0.02	1.72	
postgres: rdsadmin	384.7 MB	9.51 MB	0.02	0.95	
postgres: rdsadmin localhost(40156)					
idle [2953]!					

顯示於 Process list (程序清單) 檢視畫面中的增強型監控指標整理如下：

- RDS child processes (RDS 子程序) – 顯示支援資料庫執行個體的 RDS 程序摘要，例如，以及 MySQL 資料庫執行個體的 mysqld。程序執行緒以巢狀顯示在父程序之下。程序執行緒僅顯示 CPU 使用率，因為其他指標與該程序的所有執行緒相同。主控台最多顯示 100 個程序與執行緒。其結果為消耗最多 CPU 與記憶體的程序與執行緒的組合。如果有 50 個以上的程序及 50 個以上的執行緒，主控台將顯示各類別中消耗量最高的 50 個程序與執行緒。此顯示有助於您識別哪些程序對於效能的影響最大。
- RDS 程序 - 顯示 RDS 管理代理程式使用的資源摘要、診斷監控程序，以及其他支援 RDS 資料庫執行個體時所需的 AWS 程序。
- OS processes (作業系統程序) – 顯示核心與系統程序的摘要，這些程序對效能的影響通常很小。

各程序所列出的項目為：

- VIRT (VIRT) – 顯示程序的虛擬記憶體大小。

- RES (RES) – 顯示程序實際使用的實體記憶體。
- CPU% – 會顯示程序正在使用的 CPU 頻寬總量百分比。
- MEM% – 會顯示程序正在使用的記憶體總量百分比。

RDS 主控台顯示的監控資料擷取自 Amazon CloudWatch Logs。您也可以從 CloudWatch Logs 擷取資料庫執行個體的指標做為日誌串流。如需更多詳細資訊，請參閱 [使用 CloudWatch Logs 檢視作業系統指標](#)。

在以下期間不會傳回增強型監控指標：

- 資料庫執行個體的容錯移轉。
- 變更資料庫執行個體的執行個體類別 (擴充運算)。

在資料庫執行個體重新開機過程中會傳回增強型監控指標，因為只有資料庫引擎會重新開機。作業系統指標仍會繼續回報。

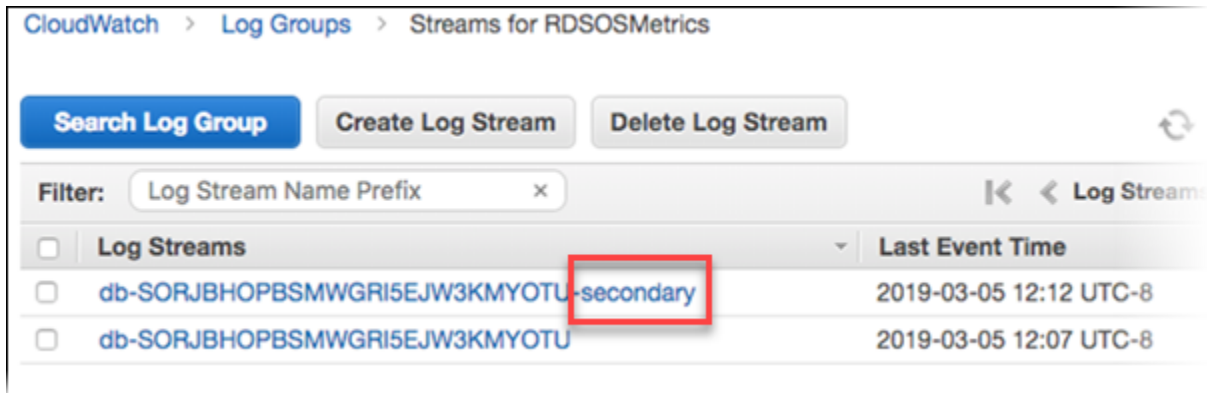
使用 CloudWatch Logs 檢視作業系統指標

在您為資料庫執行個體或多可用區域資料庫叢集啟用增強型監控之後，即可使用 CloudWatch Logs 與代表受監控之單一資料庫執行個體或資料庫叢集の日誌串流來檢視資料庫執行個體。日誌串流識別符為該資料庫執行個體或資料庫叢集的資源識別符 (DbiResourceId)。

檢視增強型監控日誌資料

1. 透過 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 如有必要，請選擇您資料庫執行個體或多可用區域資料庫叢集所在的 AWS 區域。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的 [區域和端點](#)。
3. 在導覽窗格中，選擇 Logs (日誌)。
4. 在日誌群組清單中，選擇 RDSOSMetrics。

在多可用區域資料庫執行個體部署中，名稱中附加 -secondary 的日誌檔案用於多可用區域備用複本。



The screenshot shows the Amazon CloudWatch console interface for 'Streams for RDSOSMetrics'. At the top, there are navigation breadcrumbs: 'CloudWatch > Log Groups > Streams for RDSOSMetrics'. Below this, there are three buttons: 'Search Log Group' (blue), 'Create Log Stream', and 'Delete Log Stream'. A filter box is set to 'Log Stream Name Prefix'. The main content area displays a table of log streams. The first row is a header with a checkbox, 'Log Streams', a dropdown arrow, and 'Last Event Time'. The second row has a checkbox, the log stream name 'db-SORJBHOPBSMWGRI5EJW3KMYOTU-secondary' (highlighted with a red box), and the timestamp '2019-03-05 12:12 UTC-8'. The third row has a checkbox, the log stream name 'db-SORJBHOPBSMWGRI5EJW3KMYOTU', and the timestamp '2019-03-05 12:07 UTC-8'.

<input type="checkbox"/>	Log Streams	Last Event Time
<input type="checkbox"/>	db-SORJBHOPBSMWGRI5EJW3KMYOTU-secondary	2019-03-05 12:12 UTC-8
<input type="checkbox"/>	db-SORJBHOPBSMWGRI5EJW3KMYOTU	2019-03-05 12:07 UTC-8

5. 在日誌串流清單中，選擇您要檢視的日誌串流。

Amazon RDS 的指標參考

在此參考中，您可以找到 Amazon CloudWatch、Performance Insights 和增強型監控的 Amazon RDS 指標描述。

主題

- [Amazon RDS 的 Amazon CloudWatch 指標](#)
- [Amazon RDS 的 Amazon CloudWatch 維度](#)
- [Amazon CloudWatch 指標的 Performance Insights](#)
- [Performance Insights 計數器指標](#)
- [績效詳情的 SQL 統計數字](#)
- [增強型監控中的作業系統指標](#)

Amazon RDS 的 Amazon CloudWatch 指標

Amazon RDS 在 AWS/RDS 和 AWS/Usage 命名空間 CloudWatch 中向 Amazon 發布指標。

主題

- [Amazon RDS 的 CloudWatch Amazon 實例級指標](#)
- [Amazon RDS Amazon 的亞馬遜 CloudWatch 使用指標](#)

Amazon RDS 的 CloudWatch Amazon 實例級指標

Amazon 中的 AWS/RDS 命名空間 CloudWatch 包含下列執行個體層級指標。


Note

Amazon RDS 主控台可能會以不同於傳送至 Amazon 的單位顯示指標 CloudWatch。例如，Amazon RDS 主控台可能會以兆位元組 (MB) 為單位顯示指標，而指標則以位元組為單位傳送至 Amazon CloudWatch。

指標	描述	適用對象	單位
BinLogDiskUsage	二進位日誌佔用的磁碟空間量。如果已啟用 MySQL 和 MariaDB 執行個體的自	MariaDB	位元組

指標	描述	適用對象	單位
	動備份，包括僅供讀取複本，則會建立二進位日誌。	MySQL	
BurstBalance	一般用途 SSD (gp2) 叢發儲存貯體可用輸入/輸出額度的百分比。	全部	百分比
CheckpointLag	自最近檢查點以來的時間量。		秒鐘
ConnectionAttempts	嘗試連線至執行個體的次數 (無論是否成功)。	MySQL	計數
CPUUtilization	CPU 使用率。	全部	百分比

指標	描述	適用對象	單位
CPUCredit Usage	<p>執行個體為 CPU 使用率花費的 CPU 額度數量。一個 CPU 額度等於在 100 % 使用率的情形下執行一分鐘的一個 vCPU，或是相等組合的 vCPU、利用率和時間。例如，您可能會有在 50% 利用率的情形下執行兩分鐘的一個 vCPU，或在 25% 利用率的情形下執行兩分鐘的兩個 vCPU。</p> <p>此量度僅適用於db.t2db.t3、和db.t4g執行個體。</p> <div data-bbox="391 737 956 1094" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>建議您在開發、測試伺服器或其他非生產伺服器時，僅使用 T 資料庫執行個體類別。如需 T 執行個體類別的詳細資訊，請參閱 資料庫執行個體類別的類型</p></div> <p>CPU 額度指標僅提供 5 分鐘頻率。如果您要指定大於 5 分鐘的期間，請使用 Sum 統計資訊代替 Average 統計資訊。</p>		額度 (vCPU-分鐘)

指標	描述	適用對象	單位
CPUCreditBalance	<p>自執行個體啟動或開始後，累積獲得的 CPU 額度數量。如果是 T2 Standard，CPUCreditBalance 也包含已產生的啟動額度。</p> <p>獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。額度餘額有最大值限制，它取決於執行個體大小。到達限制之後，任何獲得的新點數都會遭到捨棄。如果是 T2 Standard，啟動額度不會計入此限制。</p> <p>CPUCreditBalance 中的額度可供執行個體支付以大幅提升並超越基準 CPU 使用率。</p> <p>執行個體執行時，CPUCreditBalance 中的額度不會過期。當執行個體停止時，CPUCreditBalance 不會保持，所有累積的額度都將流失。</p> <p>CPU 額度指標僅提供 5 分鐘頻率。此量度僅適用於 db.t2db.t3、和 db.t4g 執行個體。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>建議您在開發、測試伺服器或其他非生產伺服器時，僅使用 T 資料庫執行個體類別。如需 T 執行個體類別的詳細資訊，請參閱 資料庫執行個體類別的類型</p> </div> <p>啟動積分在 Amazon RDS 中的運作方式與 Amazon EC2 相同。如需詳細資</p>		額度 (vCPU-分鐘)

指標	描述	適用對象	單位
	<p>訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的啟動積分。</p>		
CPUSurplusCreditBalance	<p>無限制執行個體已在其 CPUCreditBalance 值為 0 時支出的剩餘額度數量。</p> <p>CPUSurplusCreditBalance 值由獲得的 CPU 額度支付。如果剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量，超過最大值的支出剩餘額度將必須負擔額外的費用。</p> <p>CPU 額度指標僅提供 5 分鐘頻率。</p>	全部	額度 (vCPU-分鐘)
CPUSurplusCreditsCharged	<p>若支出剩餘額度數量未由獲得的 CPU 額度付清，會產生額外的費用。</p> <p>發生以下任何情況時，將收取支出剩餘額度的費用。</p> <ul style="list-style-type: none"> 支出剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量。在小時結束時，將收取超過最大值的支出剩餘額度的費用。 執行個體已停止或終止。 執行個體從 unlimited 切換至 standard。 <p>CPU 額度指標僅提供 5 分鐘頻率。</p>	全部	額度 (vCPU-分鐘)

指標	描述	適用對象	單位
DatabaseConnections	<p>連線至資料庫執行個體的用戶端網路連線數。</p> <p>資料庫工作階段數可能高於指標值，因為指標值不包括以下項目：</p> <ul style="list-style-type: none"> • 不再有網路連線但尚未清理資料庫的工作階段 • 資料庫引擎為供自己使用建立的工作階段 • 資料庫引擎的平行執行功能建立的工作階段 • 資料庫引擎任務排程器建立的工作階段 • Amazon RDS 連線 	全部	計數
DiskQueueDepth	等待存取磁碟的未完成 IO (讀/寫請求) 數量。	全部	計數
DiskQueueDepthLogVolume	等待存取日誌磁碟區磁碟的未完成 IO (讀/寫請求) 數量。	全部	計數
EBSByteBalance%	<p>RDS 資料庫高載儲存貯體中剩餘輸送量額度的百分比。只有基本監控才提供此指標。</p> <p>指標值是以所有磁碟區 (包括根磁碟區) 的輸送量為基礎，而非僅以包含資料庫檔案的那些磁碟區為基礎。</p> <p>若要尋找支援此指標的執行個體大小，請參閱 Amazon EC2 使用者指南中 EBS 預設優化 表格中帶有星號 (*) 的執行個體大小。Sum 統計資料不適用於此指標。</p>	全部	百分比

指標	描述	適用對象	單位
EBSIOBalance%	<p>RDS 資料庫高載儲存貯體中剩餘輸入/輸出額度的百分比。只有基本監控才提供此指標。</p> <p>指標值是以所有磁碟區 (包括根磁碟區) 的 IOPS 為基礎，而非僅以包含資料庫檔案的那些磁碟區為基礎。</p> <p>若要尋找支援此指標的執行個體大小，請參閱 Amazon EC2 使用者指南中 EBS 預設優化 表格中帶有星號 (*) 的執行個體大小。Sum 統計資料不適用於此指標。</p> <p>此指標不同於 BurstBalance。若要了解如何使用此指標，請參閱 使用 Amazon EBS 優化執行個體高載功能來改善應用程式效能並降低成本。</p>	全部	百分比
FailedSQLServerAgentJobsCount	最後 1 分鐘內失敗的 Microsoft SQL Server Agent 任務數量。	Microsoft SQL Server	每分鐘計數
FreeableMemory	<p>可用的隨機存取記憶體的数量。</p> <p>若為 MariaDB、MySQL、Oracle 和 PostgreSQL 資料庫執行個體，此指標報告 MemAvailable 的 /proc/meminfo 欄位值。</p>	全部	位元組

指標	描述	適用對象	單位
FreeLocalStorage	<p>可用的本機儲存空間量。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe 固態硬碟執行個體儲存體磁碟區。</p>		位元組
FreeStorageSpace	可用的儲存空間的數量。	全部	位元組
FreeStorageSpaceLoggingVolume	日誌磁碟區上可用的儲存空間數量。	全部	位元組
MaximumUsedTransactionIDs	已使用的交易 ID 上限。	PostgreSQL	計數
NetworkReceiveThroughput	資料庫執行個體的外來 (接收) 網路流量，包括客戶資料庫流量及用於監控與複寫的 Amazon RDS 流量。	全部	每秒位元組數
NetworkTransmitThroughput	資料庫執行個體的外送 (傳輸) 網路流量，包括客戶資料庫流量及用於監控與複寫的 Amazon RDS 流量。	全部	每秒位元組數
OldestReplicationSlotLag	以接收到 WAL 資料而言，複本遲延程度最大的遲延大小。	PostgreSQL	位元組

指標	描述	適用對象	單位
ReadIOPS	磁碟讀取輸入/輸出操作的每秒平均次數。	全部	每秒計數
ReadIOPSLocalStorage	<p>磁碟讀取輸入/輸出操作到本機儲存體的每秒平均次數。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe 固態硬碟執行個體儲存體磁碟區。</p>		每秒計數
ReadIOPSLogVolume	日誌磁碟區每秒磁碟讀取 I/O 操作的平均次數。	全部	每秒計數
ReadLatency	平均每次磁碟輸入/輸出操作耗用的時間量。	全部	秒鐘
ReadLatencyLocalStorage	<p>平均每次本機儲存體磁碟輸入/輸出操作耗用的時間量。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe SSD 執行個體儲存體磁碟區。</p>		秒鐘

指標	描述	適用對象	單位
ReadLatencyLogVolume	日誌磁碟區每個磁碟 I/O 操作所花費的平均時間量。	全部	秒鐘
ReadThroughput	平均每秒從磁碟讀取的位元組數目。	全部	每秒位元組數
ReadThroughputLocalStorage	<p>平均每秒從磁碟讀取的本機儲存體位元組數目。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe 固態硬碟執行個體儲存體磁碟區。</p>		每秒位元組數
ReadThroughputLogVolume	日誌磁碟區每秒從磁碟讀取的平均位元組數目。	全部	每秒位元組數
ReplicaLag	<p>針對僅供讀取複本的組態，讀取複本資料庫執行個體落後於來源資料庫執行個體的時間量。適用於 MariaDB、Microsoft SQL Server、MySQL、Oracle 和 PostgreSQL 僅供讀取複本。</p> <p>針對多可用區域資料庫叢集，寫入器資料庫執行個體上的最新交易與讀取器資料庫執行個體上最新套用交易之間的時間差異。</p>		秒鐘

指標	描述	適用對象	單位
ReplicationChannelLag	對於多來源複本組態，多來源複本上特定通道的時間長度落後於來源資料庫執行個體。如需詳細資訊，請參閱 the section called “監視多重來源複製通道” 。	MySQL	秒鐘
ReplicationSlotDiskUsage	複本插槽檔案使用的磁碟空間。	PostgreSQL	位元組
SwapUsage	資料庫執行個體上使用的交換空間的量。	MariaDB MySQL Oracle PostgreSQL	位元組
TransactionLogsDiskUsage	交易日誌使用的磁碟空間。	PostgreSQL	位元組
TransactionLogsGeneration	每秒產生的交易日誌的大小。	PostgreSQL	每秒位元組數
WriteIOPS	磁碟寫入輸入/輸出操作的每秒平均次數。	全部	每秒計數

指標	描述	適用對象	單位
WriteIOPS LocalStorage	<p>本機儲存體上磁碟寫入輸入/輸出操作的每秒平均次數。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe 固態硬碟執行個體儲存體磁碟區。</p>		每秒計數
WriteIOPS LogVolume	日誌磁碟區每秒磁碟寫入 I/O 操作的平均次數。	全部	每秒計數
WriteLatency	平均每次磁碟輸入/輸出操作耗用的時間量。	全部	秒鐘
WriteLatencyLocalStorage	<p>平均每次本機儲存體磁碟輸入/輸出操作耗用的時間量。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe SSD 執行個體儲存體磁碟區。</p>		秒鐘


指標	描述	適用對象	單位
WriteLatenessLogVolume	日誌磁碟區每個磁碟 I/O 操作所花費的平均時間量。	全部	秒鐘
WriteThroughput	平均每秒寫入磁碟的位元組數目。	全部	每秒位元組數
WriteThroughputLogVolume	日誌磁碟區每秒寫入至磁碟的平均位元組數目。	全部	每秒位元組數
WriteThroughputLocalStorage	<p>每秒寫入磁碟的本機儲存體平均位元組數目。</p> <p>此指標僅適用於具有 NVMe SSD 執行個體儲存體磁碟區的資料庫執行個體類別。如需使用 NVMe SSD 執行個體儲存體磁碟區的 Amazon EC2 執行個體的相關資訊，請參閱執行個體儲存體區磁碟區。對等的 RDS 資料庫執行個體類別具有相同的執行個體儲存體磁碟區。例如，db.m6gd 和 db.r6gd 資料庫執行個體類別具有 NVMe 固態硬碟執行個體儲存體磁碟區。</p>		每秒位元組數

Amazon RDS Amazon 的亞馬遜 CloudWatch 使用指標

Amazon 中的 AWS/Usage 命名空間 CloudWatch 包含 Amazon RDS 服務配額的帳戶層級使用量指標。CloudWatch 自動收集所有人的使用狀況指標 AWS 區域。

如需詳細資訊，請參閱 Amazon [CloudWatch 使用 CloudWatch 者指南中的使用量](#) 指標。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的 [請求提高配額](#)。

指標	描述	單位*
AllocatedStorage	所有資料庫執行個體的儲存總量 總和不包括臨時遷移執行個體。	GB
DBClusterParameterGroups	AWS 帳戶中資料庫叢集參數群組的數量上限 計數不包括預設參數群組。	計數
DBClusters	AWS 帳戶中 Amazon Aurora 資料庫叢集的數量。	計數
DBInstances	AWS 帳戶中的資料庫執行個體數量。	計數
DBParameterGroups	AWS 帳戶中資料庫參數群組的數量。計數不包括預設的資料庫參數群組。	計數
DBSecurityGroups	AWS 帳戶中的安全群組數量。該計數不包括預設安全群組和預設 VPC 安全群組。	計數
DBSubnetGroups	AWS 帳戶中的資料庫子網路群組數量。計數不包括預設子網路組。	計數
ManualClusterSnapshots	AWS 帳戶中手動建立的資料庫叢集快照數量。計數不包括無效的快照。	計數
ManualSnapshots	AWS 帳戶中手動建立的資料庫快照數量。計數不包括無效的快照。	計數
OptionGroups	AWS 帳戶中的選項群組數量。計數不包括預設選項群組。	計數
ReservedDBInstances	AWS 帳戶中的保留資料庫執行個體數量。計數不包括淘汰或拒絕的執行個體。	計數

 Note

Amazon RDS 不會將使用量指標的單位發佈到 CloudWatch。這些單位僅顯示於文件中。

Amazon RDS 的 Amazon CloudWatch 維度

您可以使用下表中的任何維度來篩選 Amazon RDS 指標資料。

維度	篩選請求的資料...
DBInstanceIdentifier	特定的資料庫執行個體。
DatabaseClass	資料庫類別中的所有執行個體。例如，您可以為屬於資料庫類別 <code>db.r5.large</code> 的所有執行個體彙總指標。
EngineName	僅已識別的引擎名稱。例如，您可以為具有引擎名稱 <code>postgres</code> 的所有執行個體彙總指標。
SourceRegion	僅指定的區域。例如，您可以針對 <code>us-east-1</code> 區域中的所有資料庫執行個體來彙總指標。

Amazon CloudWatch 指標的 Performance Insights

Performance Insights 會自動將某些指標發佈到 Amazon CloudWatch。您可以從「Performance Insights」中查詢相同的資料，但使用指標可 CloudWatch 讓您輕鬆新增 CloudWatch 警示。它還可以輕鬆地將指標添加到現有 CloudWatch 儀表板。

指標	描述
DBLoad	資料庫引擎的作用中工作階段數量。您通常會需要平均作用中工作階段數量的資料。在績效詳情中，系統會以 <code>db.load.avg</code> 的形式來查詢此資料。
DBLoadCPU	當等待事件類型為 CPU 時，作用中工作階段的數量。在績效詳情中，系統會以 <code>db.load.avg</code> 的形式來查詢此資料，篩選依據為等待事件類型 CPU。
資料庫 LoadNon CPU	當等待事件類型不是 CPU 時，作用中工作階段的數量。

Note

只有在資料庫執行個體有負載時，CloudWatch 才會將這些指標發佈到。

您可以使用 CloudWatch 主控台、或 CloudWatch API 來檢查這些指標。AWS CLI 您也可以使用特殊的量度數學函數來檢查其他「Performance Insights」計數器量度。如需詳細資訊，請參閱 [查詢其他 Performance Insights 計數器量度 CloudWatch](#)。

例如，您可以執行 [get-metric-statistics](#) 命令來取得 DBLoad 測量結果的統計資料。

```
aws cloudwatch get-metric-statistics \  
  --region us-west-2 \  
  --namespace AWS/RDS \  
  --metric-name DBLoad \  
  --period 60 \  
  --statistics Average \  
  --start-time 1532035185 \  
  --end-time 1532036185 \  
  --dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

此範例會產生類似下列範例的輸出結果。

```
{  
  "Datapoints": [  
    {  
      "Timestamp": "2021-07-19T21:30:00Z",  
      "Unit": "None",  
      "Average": 2.1  
    },  
    {  
      "Timestamp": "2021-07-19T21:34:00Z",  
      "Unit": "None",  
      "Average": 1.7  
    },  
    {  
      "Timestamp": "2021-07-19T21:35:00Z",  
      "Unit": "None",  
      "Average": 2.8  
    },  
    {  
      "Timestamp": "2021-07-19T21:31:00Z",
```



```
"Unit": "None",
"Average": 1.5
},
{
"Timestamp": "2021-07-19T21:32:00Z",
"Unit": "None",
"Average": 1.8
},
{
"Timestamp": "2021-07-19T21:29:00Z",
"Unit": "None",
"Average": 3.0
},
{
"Timestamp": "2021-07-19T21:33:00Z",
"Unit": "None",
"Average": 2.4
}
],
"Label": "DBLoad"
}
```

如需詳細資訊 CloudWatch，請參閱[什麼是 Amazon CloudWatch？](#) 在 Amazon 用 CloudWatch 戶指南。

查詢其他 Performance Insights 計數器量度 CloudWatch

您可以從「RDS Performance Insights」指標上查詢、警示和圖形 CloudWatch。您可以使用 DB_PERF_INSIGHTS 指標數學函數來存取有關資料資料庫執行個體的資訊 CloudWatch。此功能可讓您使用未直接報告的「Performance Insights」度量 CloudWatch 來建立新的時間序列。

您可以按一下 CloudWatch 主控台中「選取量度」畫面中的「新增數學」下拉式功能表，以使用新的「公制數學」函數。您可以使用它來建立「效能洞見」指標或「Performance Insights」指標的組合上的警示 CloudWatch 和圖表，包括低分鐘指標的高解析度警示。您也可以在[get-metric-data](#)要求中包含 Metric Math 運算式，以程式設計方式使用函數。如需詳細資訊，請參閱[指標數學語法和函數和從 AWS 料庫建立 Performance Insights 計數器指標的警示](#)。

Performance Insights 計數器指標

計數器指標是 Performance Insights 儀表板中的作業系統和資料庫效能指標。若要協助識別並分析效能問題，您可以將計數器指標與資料庫負載相互關聯。您可以將統計函數新增至指標，以取得指標值。例如，`os.memory.active` 指標支援的函數為 `.avg`、`.min`、`.max`、`.sum` 和 `.sample_count`。

每分鐘收集一次計數器指標。作業系統指標收集方式取決於增強型監控是開啟或關閉。如果關閉增強型監控，系統每分鐘收集一次作業系統指標。如果開啟增強型監控，系統會在選定期間內收集作業系統指標。如需增強型監控的詳細資訊，請參閱 [開啟和關閉增強型監控](#)。

主題

- [績效詳情作業系統計數器](#)
- [Amazon RDS for MariaDB 和 MySQL 的績效詳情計數器](#)
- [Amazon RDS for Microsoft SQL Server 的績效詳情計數器](#)
- [Amazon RDS for Oracle 的績效詳情計數器](#)
- [Amazon RDS for PostgreSQL 的績效詳情計數器](#)

績效詳情作業系統計數器

對於 RDS for SQL Server 以外的 RDS 引擎，下列作業系統計數器 (其字首為 `os`) 可與 Performance Insights 搭配使用。

您可以將 `ListAvailableResourceMetrics` API 用於資料庫執行個體的可用計數器指標清單。如需詳細資訊，請參閱 Amazon RDS Performance Insights API 參考指南 [ListAvailableResourceMetrics](#) 中的。

計數器	類型	指標	描述
作用中	記憶體	<code>os.memory.active</code>	已指派的記憶體數量，以 KB 為單位。
緩衝區域	記憶體	<code>os.memory.buffers</code>	在寫入至儲存裝置之前，用於緩衝 I/O 請求的記憶體數量，以 KB 為單位。

計數器	類型	指標	描述
已快取	記憶體	os.memory.cached	用於快取檔案系統型 I/O 的記憶體數量，以 KB 為單位。
資料庫快取	記憶體	os.memory.db.cache	資料庫程序 (包括 tmpfs (shmem)) 用於頁面快取的記憶體數量，以位元組為單位。
資料庫常駐集大小	記憶體	存儲. 數據庫居民 SetSize	資料庫程序 (不包括 tmpfs (shmem)) 用於匿名和交換快取的記憶體數量，以位元組為單位。
資料庫交換	記憶體	os.memory.db.swap	資料庫程序用於交換的記憶體數量，以位元組為單位。
髒	記憶體	os.memory.dirty	RAM 之中已修改但尚未寫入至儲存裝置中相關資料區塊的記憶體分頁數量，以 KB 為單位。
免費	記憶體	os.memory.free	未指派的記憶體數量，以 KB 為單位。
釋出的大內存頁	記憶體	記憶體. 巨大 PagesFree	自由巨型分頁的數量。巨型分頁為 Linux 核心的功能。
保留的大內存頁	記憶體	記憶體. 巨大 PagesRsvd	已遞交的巨型分頁的數量。

計數器	類型	指標	描述
大內存頁尺寸	記憶體	記憶體. 巨大 PageSize	每個巨型分頁的大小，以 KB 為單位。
抑制的大內存頁	記憶體	記憶體. 巨大 PagesSurp	超過總數的可用剩餘巨型分頁的數量。
大內存頁總數	記憶體	記憶體. 巨大 PagesTotal	大內存頁總數。
非作用中	記憶體	os.memory.inactive	使用頻率最低的記憶體分頁數量，以 KB 為單位。
已對應	記憶體	os.memory.mapped	在程序地址空間內對應的檔案系統內容總量，以 KB 為單位。
記憶體不足終止計數	記憶體	記憶體輸出 OfMemory KillCount	上次收集間隔內發生的 OOM 終止數目。
內存頁資料表	記憶體	os.memory.pageTables	分頁表使用的記憶體數量，以 KB 為單位。
板	記憶體	os.memory.slab	個重複使用的核心資料結構數量，以 KB 為單位。
總計	記憶體	os.memory.total	記憶體總量，以 KB 為單位。
回寫	記憶體	os.memory.writeback	RAM 之中仍被寫入至支援儲存裝置的中途分頁數量，以 KB 為單位。
訪客	CPU 使用率	os.cpuUtilization. guest	客體程式使用中的 CPU 百分比。

計數器	類型	指標	描述
閒置	CPU 使用率	os.cpuUtilization.idle	CPU 閒置的百分比。
Irq	CPU 使用率	os.cpuUtilization irq	軟體中斷使用中的 CPU 百分比。
良好	CPU 使用率	os.cpuUtilization.nice	以最低優先順序執行之程式使用中的 CPU 百分比。
挪用	CPU 使用率	os.cpuUtilization.steal	其他虛擬機器使用中的 CPU 百分比。
系統	CPU 使用率	os.cpuUtilization.system	核心使用中的 CPU 百分比。
總計	CPU 使用率	os.cpuUtilization.total	使用中的 CPU 總百分比。此值包含良好值。
使用者	CPU 使用率	os.cpuUtilization.user	使用者程式使用中的 CPU 百分比。
等候	CPU 使用率	os.cpuUtilization.wait	等待 I/O 存取時未使用的 CPU 百分比。
閱讀 IO PS	磁碟 IO	os.diskIO.<devicename>.readIOsPS	每秒讀取操作的次數。
寫入 IO PS	磁碟 IO	os.diskIO.<devicename>.writeIOsPS	每秒寫入操作的次數。
平均佇列長度	磁碟 IO	OS. 迪斯基厄.<devicename>.平均 QueueLen	在 I/O 裝置的佇列中等待的請求數量。

計數器	類型	指標	描述
平均請求大小	磁碟 IO	OS. 迪斯基厄. <devicename>. 平均 ReqSz	在 I/O 裝置的佇列中等待的請求數量。
等候	磁碟 IO	os.diskIO.<devicename>.await	回應請求時所需的毫秒數，包括佇列時間與服務時間。
讀取 IO PS	磁碟 IO	os.diskIO.<devicename>.readIOsPS	每秒讀取操作的次數。
讀取 KB	磁碟 IO	os.diskIO.<devicename>.readKb	讀取的 KB 總數。
讀取 KB PS	磁碟 IO	os.diskIO.<devicename>.readKbPS	每秒讀取的 KB 總數。
Rrqm PS	磁碟 IO	os.diskIO.<devicename>.rrqmPS	每秒佇列的合併讀取請求數量。
TPS	磁碟 IO	os.diskIO.<devicename>.tps	每秒的 I/O 交易數量。
使用率	磁碟 IO	os.diskIO.<devicename>.util	發出請求的 CPU 時間百分比。
寫入 KB	磁碟 IO	os.diskIO.<devicename>.writeKb	寫入的 KB 總數。
寫入 KB PS	磁碟 IO	os.diskIO.<devicename>.writeKbPS	每秒寫入的 KB 總數。

計數器	類型	指標	描述
Wrqm PS	磁碟 IO	os.diskIO.<deviceName>.wrqmPS	每秒佇列的合併寫入請求數量。
封鎖	任務	os.tasks.blocked	封鎖的任務數量。
執行中	任務	os.tasks.running	執行中的任務數量。
休眠中	任務	os.tasks.sleeping	睡眠中的任務數量。
已停止	任務	os.tasks.stopped	已停止的任務數量。
總計	任務	os.tasks.total	任務的總數。
廢止中	任務	os.tasks.zombie	與作用中父任務進行互動的子任務數量。
一	負載平均分鐘	加載. 一 AverageMinute	過去 1 分鐘內請求 CPU 時間的程序數量。
十五	負載平均分鐘	加載 .15 AverageMinute	過去 15 分鐘內請求 CPU 時間的程序數量。
五	負載平均分鐘	加載 .5 AverageMinute	過去 5 分鐘內請求 CPU 時間的程序數量。
已快取	交換	os.swap.cached	做為快取記憶體使用的 swap 記憶體數量，以 KB 為單位。
免費	交換	os.swap.free	可用的交換記憶體數量，以 KB 為單位。

計數器	類型	指標	描述
In (入)	交換	os.swap.in	從磁碟交換輸入的記憶體數量，以 KB 為單位。
Out (出)	交換	os.swap.out	交換輸出到磁碟的記憶體數量，以 KB 為單位。
總計	交換	os.swap.total	可用的交換記憶體總量，以 KB 為單位。
檔案數上限	檔案系統	os.fileSys.maxFiles	檔案相同可建立的最大檔案數量。
已使用的檔案	檔案系統	os.fileSys.usedFiles	檔案系統中的檔案數量。
已使用的檔案百分比	檔案系統	操作系統. 檔案. 使用 FilePercent	使用中的可用檔案百分比。
已使用百分比	檔案系統	os.fileSys.usedPercent	使用中檔案系統磁碟空間的百分比。
已使用	檔案系統	os.fileSys.used	檔案系統中的檔案已使用的磁碟空間容量，以 KB 為單位。
總計	檔案系統	os.fileSys.total	檔案系統可用的磁碟空間總容量，以 KB 為單位。
Rx	網路	os.network.rx	每秒接收的位元組數量。
Tx	網路	os.network.tx	每秒上傳的位元組數量。

計數器	類型	指標	描述
ACU 使用率	一般	os.general.acuUtilization	目前容量佔最大設定容量的百分比。
最大設定的 ACU	一般	一般. 最大 Configure dAcu	使用者設定的最大容量，以 ACU 為單位。
最小設定的 ACU	一般	通用. 分鐘 Configure dAcu	使用者設定的最小容量，以 ACU 為單位。
VCPU 數目	一般	os.general.numVCPU s	資料庫執行個體的虛擬 CPU 數量。
無伺服器資料庫容量	一般	OS. 一般. 無伺服器 DatabaseCapacity	執行個體的目前容量，以 ACU 為單位。

Amazon RDS for MariaDB 和 MySQL 的績效詳情計數器

以下資料庫計數器適用於 Amazon RDS for MariaDB 和 MySQL 的績效詳情。

主題

- [RDS for MariaDB 和 RDS for MySQL 的原生計數器](#)
- [Amazon RDS for MariaDB 和 MySQL 非原生計數器](#)

RDS for MariaDB 和 RDS for MySQL 的原生計數器

原生指標由資料庫引擎定義，而不是由 Amazon RDS。如需這些原生指標的定義，請參閱 MySQL 文件中的[伺服器狀態變數](#)。

計數器	類型	單位	指標
Com_analyze	SQL	每秒查詢數	db.SQL.Com_analyze
Com_optimize	SQL	每秒查詢數	db.SQL.Com_optimize
Com_select	SQL	每秒查詢數	db.SQL.Com_select

計數器	類型	單位	指標
連線	SQL	每分鐘 (成功與否) 對 MySQL 伺服器的連接嘗試次數	db.Users.Connections
Innodb_rows_deleted	SQL	每秒列數	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	每秒列數	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	每秒列數	db.SQL.Innodb_rows_read
Innodb_rows_updated	SQL	每秒列數	db.SQL.Innodb_rows_updated
Select_full_join	SQL	每秒查詢數	db.SQL.Select_full_join
Select_full_range_join	SQL	每秒查詢數	db.SQL.Select_full_range_join
Select_range	SQL	每秒查詢數	db.SQL.Select_range
Select_range_check	SQL	每秒查詢數	db.SQL.Select_range_check
Select_scan	SQL	每秒查詢數	db.SQL.Select_scan
Slow_queries	SQL	每秒查詢數	db.SQL.Slow_queries
Sort_merge_passes	SQL	每秒查詢數	db.SQL.Sort_merge_passes
Sort_range	SQL	每秒查詢數	db.SQL.Sort_range
Sort_rows	SQL	每秒查詢數	db.SQL.Sort_rows
Sort_scan	SQL	每秒查詢數	db.SQL.Sort_scan
問題	SQL	每秒查詢數	db.SQL.Questions
Innodb_row_lock_time	鎖定	毫秒 (平均)	db.Locks.Innodb_row_lock_time
Table_locks_immediate	鎖定	每秒請求數	db.Locks.Table_locks_immediate


計數器	類型	單位	指標
Table_locks_waited	鎖定	每秒請求數	db.Locks.Table_locks_waited
Aborted_clients	使用者	連線	db.Users.Aborted_clients
Aborted_connects	使用者	連線	db.Users.Aborted_connects
max_connections	使用者	連線	db.User.max_connections
Threads_created	使用者	連線	db.Users.Threads_created
Threads_running	使用者	連線	db.Users.Threads_running
Innodb_data_writes	輸入/輸出	每秒操作數	db.IO.Innodb_data_writes
Innodb_dblwr_writes	輸入/輸出	每秒操作數	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	輸入/輸出	每秒操作數	db.IO.Innodb_log_write_requests
Innodb_log_writes	輸入/輸出	每秒操作數	db.IO.Innodb_log_writes
Innodb_pages_written	輸入/輸出	每秒頁面數	db.IO.Innodb_pages_written
Created_tmp_disk_tables	暫存	每秒資料表數	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	暫存	每秒資料表數	db.Temp.Created_tmp_tables
Innodb_buffer_pool_pages_data	快取	頁面	db.Cache.Innodb_buffer_pool_pages_data
Innodb_buffer_pool_pages_total	快取	頁面	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	快取	每秒頁面數	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	快取	每秒頁面數	db.Cache.Innodb_buffer_pool_reads

計數器	類型	單位	指標
Opened_tables	快取	資料表	db.Cache.Opened_tables
Opened_table_definitions	快取	資料表	db.Cache.Opened_table_definitions
Qcache_hits	快取	查詢	db.Cache.Qcache_hits

Amazon RDS for MariaDB 和 MySQL 非原生計數器

非原生計數器指標是 Amazon RDS 定義的計數器。非原生指標可以是您使用特定查詢取得的指標。非原生指標也可以是衍生的指標，其中會將兩個以上的原生計數器用在計算中的比率、命中率或延遲。

計數器	類型	指標	描述	定義
innodb_buffer_pool_hits	快取	db.Cache.innoDB_buffer_pool_hits	InnoDB 可從緩衝集區獲得的讀取數。	$\text{innodb_buffer_pool_read_requests} - \text{innodb_buffer_pool_reads}$
innodb_buffer_pool_hit_rate	快取	db.Cache.innoDB_buffer_pool_hit_rate	InnoDB 可從緩衝集區獲得的讀取百分比。	$100 * \frac{\text{innodb_buffer_pool_read_requests}}{(\text{innodb_buffer_pool_read_requests} + \text{innodb_buffer_pool_reads})}$
innodb_buffer_pool_usage	快取	db.Cache.innoDB_buffer_pool_usage	包含資料 (頁面) 的 InnoDB 緩	$\frac{\text{Innodb_buffer_pool_pages_data}}{\text{Innodb_bu}}$

計數器	類型	指標	描述	定義
			衝集區百分比。	$\text{fffer_pool_pages_total} * 100.0$
			<p> Note</p> <p>使用壓縮的資料表時，此值可能會有所不同。如需詳細資訊，請參閱 MySQL 文</p>	

計數器	類型	指標	描述	定義
innodb_datafile_writes_to_disk	輸入/輸出	db.IO.innoDB_datafile_writes_to_disk	InnoDB 資料檔案寫入磁碟的數量，不包含重複寫入和重做記錄寫入操作。	Innodb_data_writes - Innodb_log_writes - Innodb_db_lwr_writes
innodb_rows_changed	SQL	db.SQL.innodb_rows_changed	InnoDB 列操作總計。	db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated
active_transactions	交易	db.Transactions.active_transactions	作用中交易總計。	SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA INNOODB_TRX

計數器	類型	指標	描述	定義
trx_rseg_history_len	交易	db.Transactions.trx_rseg_history_len	InnoDB 交易系統所維護之已確認交易的還原日誌頁面清單，用於實作多版本並行控制。如需還原日誌詳情的詳細資訊，請參閱 MySQL 文件中的 https://dev.mysql.com/doc/refman/8.0/en/innodb-multi-versioning.html 。	SELECT COUNT AS trx_rseg_history_len FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='trx_rseg_history_len'
innodb_deadlocks	鎖定	db.Locks.innodb_deadlocks	死鎖總數。	SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'

計數器	類型	指標	描述	定義
innodb_lock_timeouts	鎖定	db.Locks.innodb_lock_timeouts	逾時的鎖定總數。	SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'
innodb_row_lock_waits	鎖定	db.Locks.innodb_row_lock_waits	造成等待的列鎖定總數。	SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'

Amazon RDS for Microsoft SQL Server 的績效詳情計數器

RDS for Microsoft SQL Server 的績效詳情提供以下資料庫計數器。

RDS for Microsoft SQL Server 的原生計數器

原生指標由資料庫引擎定義，而不是由Amazon RDS。在 Microsoft SQL Server 文件的[使用 SQL Server 物件](#)中，您可以找到這些原生指標的定義。

計數器	類型	單位	指標
Forwarded Records	存取方法	每秒記錄數	db.Access Methods.Forwarded Records

計數器	類型	單位	指標
Page Splits	存取方法	每秒分割數	db.Access Methods.P age Splits
Buffer cache hit ratio	Buffer Manager	比率	db.Buffer Manager.B uffer cache hit ratio
Page life expectancy	Buffer Manager	預期壽命 (以秒為單位)	db.Buffer Manager.P age life expectancy
Page lookups	Buffer Manager	每秒查閱數	db.Buffer Manager.P age lookups
Page reads	Buffer Manager	每秒讀取數	db.Buffer Manager.P age reads
Page writes	Buffer Manager	每秒寫入數	db.Buffer Manager.P age writes
Active Transactions	資料庫	交易	db.Databases.Active Transactions (_Total)
Log Bytes Flushed	資料庫	每秒清空位組數	db.Databases.Log Bytes Flushed (_Total)
Log Flush Waits	資料庫	每秒等待數	db.Databases.Log Flush Waits (_Total)
Log Flushes	資料庫	每秒清空數	db.Databases.Log Flushes (_Total)
Write Transactions	資料庫	每秒交易數	db.Databases.Write Transactions (_Total)
Processes blocked	General Statistics	Processes blocked	db.General Statistic s.Processes blocked

計數器	類型	單位	指標
User Connections	General Statistics	連線	db.General Statistics.User Connections
Latch Waits	Latches	每秒等待數	db.Latches.Latch Waits
Number of Deadlocks	鎖定	每秒死鎖數	db.Locks.Number of Deadlocks (_Total)
Memory Grants Pending	Memory Manager	記憶體授與數	db.Memory Manager.Memory Grants Pending
Batch Requests	SQL Statistics	每秒請求數	db.SQL Statistics.Batch Requests
SQL Compilations	SQL Statistics	每秒編譯數	db.SQL Statistics.SQL Compilations
SQL Re-Compilations	SQL Statistics	每秒重新編譯數	db.SQL Statistics.SQL Re-Compilations

Amazon RDS for Oracle 的績效詳情計數器

RDS for Oracle 的績效詳情提供以下資料庫計數器。

RDS for Oracle 的原生計數器

原生指標由資料庫引擎定義，而不是由Amazon RDS。您可以在 Oracle 文件的[統計資料描述](#)中找到這些原生指標的定義。

Note

對於 CPU used by this session 計數器指標，系統已將單位從原生的釐秒轉換為作用中的工作階段來讓此值更容易使用。例如，在資料庫負載圖表中傳送的 CPU 代表對 CPU 的需求。計數器指標 CPU used by this session 表示 Oracle 工作階段使用的 CPU 數。您可

以將傳送的 CPU 與 CPU used by this session 計數器指標相比較。CPU 的需求高於使用 CPU 時，工作階段會等待 CPU 時間。

計數器	類型	單位	指標
此工作階段使用的 CPU	使用者	作用中的工作階段	db.User.CPU used by this session
多次來回用戶端的 SQL*Net	使用者	每秒來回數	db.User.SQL*Net roundtrips to/from client
從用戶端透過 SQL*Net 接收的位元組數	使用者	每秒位元組數	db.User.bytes received via SQL*Net from client
使用者遞交	使用者	每秒遞交數	db.User.user commits
登入累積	使用者	每秒登入數	db.User.logons cumulative
使用者呼叫	使用者	每秒呼叫數	db.User.user calls
透過 SQL*Net 傳送至用戶端的位元組數	使用者	每秒位元組數	db.User.bytes sent via SQL*Net to client
使用者轉返	使用者	每秒轉返數	db.User.user rollbacks
重做大小	重做	每秒位元組數	db.Redo.redo size
剖析計數 (總計)	SQL	每秒剖析數	db.SQL.parse count (total)
剖析計數 (硬性)	SQL	每秒剖析數	db.SQL.parse count (hard)
資料表經掃描的列數	SQL	每秒列數	db.SQL.table scan rows gotten

計數器	類型	單位	指標
排序 (記憶體)	SQL	每秒排序數	db.SQL.sorts (memory)
排序 (磁碟)	SQL	每秒排序數	db.SQL.sorts (disk)
每秒 (列)	SQL	每秒排序數	db.SQL.sorts (rows)
實體讀取位元組數	快取	每秒位元組數	db.Cache.physical read bytes
資料庫區塊取得數	快取	每秒區塊數	db.Cache.db block gets
DBWR 檢查點	快取	每分鐘檢查點	db.Cache.DBWR checkpoints
實體讀取數	快取	每秒讀取數	db.Cache.physical reads
從快取中一致取得數	快取	每秒取得數	db.Cache.consistent gets from cache
從快取中資料庫區塊取得數	快取	每秒取得數	db.Cache.db block gets from cache
一致取得	快取	每秒取得數	db.Cache.consistent gets

Amazon RDS for PostgreSQL 的績效詳情計數器

以下資料庫計數器適用於 Amazon RDS for PostgreSQL 的績效詳情。

主題

- [Amazon RDS for PostgreSQL 的原生計數器](#)
- [Amazon RDS for PostgreSQL 的非原生計數器](#)

Amazon RDS for PostgreSQL 的原生計數器

原生指標由資料庫引擎定義，而不是由Amazon RDS。您可以在 PostgreSQL 文件的[檢視統計資料](#)中找到這些原生指標的定義。

計數器	類型	單位	指標
blks_hit	快取	每秒區塊數	db.Cache.blks_hit
buffers_alloc	快取	每秒區塊數	db.Cache.buffers_alloc
buffers_checkpoint	檢查點	每秒區塊數	db.Checkpoint.buffers_checkpoint
checkpoint_sync_time	檢查點	每個檢查點的毫秒數	db.Checkpoint.checkpoint_sync_time
checkpoint_write_time	檢查點	每個檢查點的毫秒數	db.Checkpoint.checkpoint_write_time
checkpoints_req	檢查點	每分鐘檢查點	db.Checkpoint.checkpoints_req
checkpoints_timed	檢查點	每分鐘檢查點	db.Checkpoint.checkpoints_timed
maxwritten_clean	檢查點	每分鐘的 Bgwriter 清除停止數	db.Checkpoint.maxwritten_clean
deadlocks	並行數量	每分鐘的鎖死數	db.Concurrency.deadlocks
blk_read_time	輸入/輸出	毫秒	db.IO.blk_read_time
blks_read	輸入/輸出	每秒區塊數	db.IO.blks_read
buffers_backend	輸入/輸出	每秒區塊數	db.IO.buffers_backend
buffers_backend_fsync	輸入/輸出	每秒區塊數	db.IO.buffers_backend_fsync
buffers_clean	輸入/輸出	每秒區塊數	db.IO.buffers_clean
tup_deleted	SQL	每秒元組數	db.SQL.tup_deleted
tup_fetched	SQL	每秒元組數	db.SQL.tup_fetched

計數器	類型	單位	指標
tup_inserted	SQL	每秒元組數	db.SQL.tup_inserted
tup_returned	SQL	每秒元組數	db.SQL.tup_returned
tup_updated	SQL	每秒元組數	db.SQL.tup_updated
temp_bytes	暫存	每秒位元組數	db.Temp.temp_bytes
temp_files	暫存	每分鐘的檔案數	db.Temp.temp_files
xact_commit	交易	每秒遞交數	db.Transactions.xact_commit
xact_rollback	交易	每秒轉返數	db.Transactions.xact_rollback
numbackends	使用者	連線	db.User.numbackends
archived_count	預寫日誌 (WAL)	每分鐘的檔案數	db.WAL.archived_count

Amazon RDS for PostgreSQL 的非原生計數器

非原生計數器指標是 Amazon RDS 定義的計數器。非原生指標可以是您使用特定查詢取得的指標。非原生指標也可以是衍生的指標，其中會將兩個以上的原生計數器用在計算中的比率、命中率或延遲。

計數器	類型	指標	描述	定義
checkpoint_t_sync_latency	檢查點	db.Checkpoint.checkpoint_sync_latency	在檔案同步至磁碟的檢查點處理過程中所用的時間總計。	$\text{checkpoint_t_sync_time} / (\text{checkpoints_timed} + \text{checkpoints_req})$
checkpoint_t_write_latency	檢查點	db.Checkpoint.checkpoint_write_latency	在檔案寫入磁碟的檢查點處理過程中所用的時間總計。	$\text{checkpoint_t_write_time} / (\text{checkpoi}$

計數器	類型	指標	描述	定義
				nts_timed + checkpoint_req)
read_latency	輸入/ 輸出	db.IO.read_latency	在此執行個體中後端讀取資料檔案區塊所用的時間。	blk_read_time / blks_read
idle_in_transaction_aborted_count	State	資料庫狀態中止 _ 計數	idle in transaction (aborted) 狀態中的工作階段數目。	-
idle_in_transaction_count	State	數據庫狀態 .idle_in 交易計數	idle in transaction 狀態中的工作階段數目。	-
idle_in_transaction_max_time	State	資料庫狀態 .idle_ 最大交易時間	idle in transaction 狀態中最長執行之交易的持續時間，以秒為單位。	-
active_transactions	交易	db.Transactions.active_transactions	作用中交易的數目。	-
blocked_transactions	交易	db.Transactions.blocked_transactions	封鎖的交易數目。	-
max_used_xact_ids	交易	db.Transactions.max_used_xact_ids	尚未吸塵的交易數量。	-

計數器	類型	指標	描述	定義
max_connections	使用者	db.User.max_connections	資料庫執行個體允許的最大連線數 (如max_connections 參數中所設定)。	-
archive_failed_count	WAL	db.WAL.archive_failed_count	封存 WAL 檔案失敗嘗試的次數，以每分鐘檔案為單位。	-

績效詳情的 SQL 統計數字

SQL 統計數字是績效詳情所收集有關 SQL 查詢的效能相關指標。績效詳情會收集查詢執行的每一秒和每個 SQL 呼叫的統計資料。SQL 統計資料是所選時間範圍的平均值。

SQL 摘要是具有給定模式但不一定具有相同字面值的的所有查詢的複合。摘要會以問號來取代字面值。例如 `SELECT * FROM emp WHERE lname = ?`。此摘要可能包含下列子查詢：

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

所有引擎都支援摘要查詢的 SQL 統計資料。

如需此功能的區域、資料庫引擎和執行個體類別支援的資訊，請參閱 [Performance Insights 功能支援 Amazon RDS 資料庫引擎和執行個體類別](#)

主題

- [MariaDB 和 MySQL 的 SQL 統計資料](#)
- [Oracle 的 SQL 統計資料](#)
- [SQL Server 的 SQL 統計資料](#)
- [RDS PostgreSQL 的 SQL 統計資料](#)

MariaDB 和 MySQL 的 SQL 統計資料

MariaDB 和 MySQL 只收集摘要層級的 SQL 統計數字。不會顯示陳述式層級的統計數字。

主題

- [MariaDB 和 MySQL 的摘要統計資料](#)
- [MariaDB 和 MySQL 的每秒統計數字](#)
- [MariaDB 和 MySQL 的每次呼叫統計數字](#)

MariaDB 和 MySQL 的摘要統計資料

績效詳情會從 `events_statements_summary_by_digest` 資料表收集 SQL 摘要統計數字。`events_statements_summary_by_digest` 資料表由資料庫管理。

摘要資料表沒有移出政策。此資料表填滿時，AWS Management Console 會顯示下列訊息：

```
Performance Insights is unable to collect SQL Digest statistics on new queries because the table events_statements_summary_by_digest is full. Please truncate events_statements_summary_by_digest table to clear the issue. Check the User Guide for more details.
```

在這種情況下，MariaDB 和 MySQL 不會追蹤 SQL 查詢。如要解決此問題，績效詳情會在滿足下列兩個條件時自動截斷摘要資料表：

- 表格已滿。
- 績效詳情會自動管理效能結構描述。

若要進行自動管理，`performance_schema` 參數必須設定為 0，而 Source (來源) 不可設定為 user。如果績效詳情未自動管理效能結構描述，請參閱 [在 Amazon RDS for MariaDB 或 MySQL 上開啟績效詳情的效能結構描述](#)。

在 AWS CLI 中，請執行 [describe-db-parameters](#) 命令來檢查參數值的來源。

MariaDB 和 MySQL 的每秒統計數字

以下 SQL 統計資料適用於 MariaDB 和 MySQL 資料庫執行個體。

指標	單位
db.sql_tokenized.stats.count_star_per_sec	每秒呼叫數
db.sql_tokenized.stats.sum_timer_wait_per_sec	每秒平均作用中執行數 (AAE)
db.sql_tokenized.stats.sum_select_full_join_per_sec	選取每秒完整聯結
db.sql_tokenized.stats.sum_select_range_check_per_sec	選取每秒範圍檢查
db.sql_tokenized.stats.sum_select_scan_per_sec	選取每秒掃描
db.sql_tokenized.stats.sum_sort_merge_passes_per_sec	排序每秒合併路徑
db.sql_tokenized.stats.sum_sort_scan_per_sec	排序每秒掃描
db.sql_tokenized.stats.sum_sort_range_per_sec	排序每秒範圍
db.sql_tokenized.stats.sum_sort_rows_per_sec	排序每秒列數
db.sql_tokenized.stats.sum_rows_affected_per_sec	每秒影響的列數
db.sql_tokenized.stats.sum_rows_examined_per_sec	每秒檢查的列數
db.sql_tokenized.stats.sum_rows_sent_per_sec	每秒傳送的列數
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_sec	每秒建立的暫存磁碟資料表
db.sql_tokenized.stats.sum_created_tmp_tables_per_sec	每秒建立的暫存資料表
db.sql_tokenized.stats.sum_lock_time_per_sec	每秒鎖定時間 (毫秒)

MariaDB 和 MySQL 的每次呼叫統計數字

下列指標提供 SQL 陳述式的每次呼叫統計數字。

指標	單位
db.sql_tokenized.stats.sum_timer_wait_per_call	每次呼叫平均延遲 (毫秒)
db.sql_tokenized.stats.sum_select_full_join_per_call	選取每個呼叫的完整聯結
db.sql_tokenized.stats.sum_select_range_check_per_call	選取每個呼叫的範圍檢查
db.sql_tokenized.stats.sum_select_scan_per_call	選取每個呼叫的掃描
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	排序每個呼叫的合併路徑
db.sql_tokenized.stats.sum_sort_scan_per_call	排序每個呼叫的掃描
db.sql_tokenized.stats.sum_sort_range_per_call	排序每個呼叫的範圍
db.sql_tokenized.stats.sum_sort_rows_per_call	排序每個呼叫的列數
db.sql_tokenized.stats.sum_rows_affected_per_call	每個呼叫受影響的列數
db.sql_tokenized.stats.sum_rows_examined_per_call	每個呼叫所檢查的列數
db.sql_tokenized.stats.sum_rows_sent_per_call	每個呼叫傳送的列數
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_call	每個呼叫建立的暫存磁碟資料表
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	每個呼叫建立的暫存資料表
db.sql_tokenized.stats.sum_lock_time_per_call	每個呼叫鎖定時間 (毫秒)

Oracle 的 SQL 統計資料

Amazon RDS for Oracle 同時收集陳述式層級和摘要層級的 SQL 統計數字。在陳述式層級，ID 欄代表 V\$SQL.SQL_ID 的值。在摘要層級，ID 欄顯示 V\$SQL.FORCE_MATCHING_SIGNATURE 的值。

在摘要層級，如果 ID 為 0，表示 Oracle Database 已斷定此陳述句不適合重複使用。在這種情況下，SQL 子陳述式可能歸入別的摘要。不過，第一次收集的 SQL 陳述式會一起組合在 digest_text 下方。

主題

- [Oracle 的每秒統計數字](#)
- [Oracle 的每次呼叫統計數字](#)

Oracle 的每秒統計數字

下列指標提供 Oracle SQL 查詢的每秒統計數字。

指標	單位
db.sql.stats.executions_per_sec	每秒執行數
db.sql.stats.elapsed_time_per_sec	平均作用中執行數 (AAE)
db.sql.stats.rows_processed_per_sec	每秒處理列數
db.sql.stats.buffer_gets_per_sec	每秒緩衝區取得數
db.sql.stats.physical_read_requests_per_sec	每秒實體讀取數
db.sql.stats.physical_write_requests_per_sec	每秒實體寫入數
db.sql.stats.total_sharable_mem_per_sec	每秒可共用記憶體總數 (位元組)
db.sql.stats.cpu_time_per_sec	每秒 CPU 時間 (毫秒)

下列指標提供 Oracle SQL 摘要查詢的每次呼叫統計數字。

指標	單位
db.sql_tokenized.stats.executions_per_sec	每秒執行數
db.sql_tokenized.stats.elapsed_time_per_sec	平均作用中執行數 (AAE)
db.sql_tokenized.stats.rows_processed_per_sec	每秒處理列數
db.sql_tokenized.stats.buffer_gets_per_sec	每秒緩衝區取得數
db.sql_tokenized.stats.physical_read_requests_per_sec	每秒實體讀取數
db.sql_tokenized.stats.physical_write_requests_per_sec	每秒實體寫入數
db.sql_tokenized.stats.total_sharable_mem_per_sec	每秒可共用記憶體總數 (位元組)
db.sql_tokenized.stats.cpu_time_per_sec	每秒 CPU 時間 (毫秒)

Oracle 的每次呼叫統計數字

下列指標提供 Oracle SQL 陳述式的每次呼叫統計數字。

指標	單位
db.sql.stats.elapsed_time_per_exec	每次執行經歷時間 (毫秒)
db.sql.stats.rows_processed_per_exec	每秒執行處理列數
db.sql.stats.buffer_gets_per_exec	每次執行緩衝區取得數
db.sql.stats.physical_read_requests_per_exec	每次執行實體讀取數
db.sql.stats.physical_write_requests_per_exec	每次執行實體寫入數
db.sql.stats.total_sharable_mem_per_exec	每次執行可共用記憶體總數 (位元組)

指標	單位
db.sql.stats.cpu_time_per_exec	每次執行 CPU 時間 (毫秒)

下列指標提供 Oracle SQL 摘要查詢的每次呼叫統計數字。

指標	單位
db.sql_tokenized.stats.elapsed_time_per_exec	每次執行經歷時間 (毫秒)
db.sql_tokenized.stats.rows_processed_per_exec	每秒執行處理列數
db.sql_tokenized.stats.buffer_gets_per_exec	每次執行緩衝區取得數
db.sql_tokenized.stats.physical_read_requests_per_exec	每次執行實體讀取數
db.sql_tokenized.stats.physical_write_requests_per_exec	每次執行實體寫入數
db.sql_tokenized.stats.total_sharable_mem_per_exec	每次執行可共用記憶體總數 (位元組)
db.sql_tokenized.stats.cpu_time_per_exec	每次執行 CPU 時間 (毫秒)

SQL Server 的 SQL 統計資料

Amazon RDS for SQL Server 同時收集陳述式層級和摘要層級的 SQL 統計資料。在陳述式層級，ID 欄代表 sql_handle 的值。在摘要層級，ID 欄顯示 query_hash 的值。

SQL Server 會針對幾個陳述式的 query_hash 傳回 NULL 值。例如，ALTER INDEX、CHECKPOINT、UPDATE STATISTICS、COMMIT TRANSACTION、FETCH NEXT FROM Cursor，以及少數 INSERT 陳述式、SELECT @<variable>、條件陳述式和可執行預存程序。在這種情況下，sql_handle 值會顯示為該陳述式摘要層級的 ID。

主題

- [SQL Server 的每秒統計資料](#)

- [SQL Server 的每次呼叫統計資料](#)

SQL Server 的每秒統計資料

下列指標提供 SQL Server SQL 查詢的每秒統計資料。

指標	單位
db.sql.stats.execution_count_per_sec	每秒執行數
db.sql.stats.total_elapsed_time_per_sec	每秒經過時間總長
db.sql.stats.total_rows_per_sec	每秒處理總列數
db.sql.stats.total_logical_reads_per_sec	每秒邏輯讀取總數
db.sql.stats.total_logical_writes_per_sec	每秒邏輯寫入總數
db.sql.stats.total_physical_reads_per_sec	每秒實體讀取總數
db.sql.stats.total_worker_time_per_sec	CPU 時間總長 (毫秒)

下列指標提供 SQL Server SQL 摘要查詢的每秒統計資料。

指標	單位
db.sql_tokenized.stats.execution_count_per_sec	每秒執行數
db.sql_tokenized.stats.total_elapsed_time_per_sec	每秒經過時間總長
db.sql_tokenized.stats.total_rows_per_sec	每秒處理總列數
db.sql_tokenized.stats.total_logical_reads_per_sec	每秒邏輯讀取總數
db.sql_tokenized.stats.total_logical_writes_per_sec	每秒邏輯寫入總數

指標	單位
db.sql_tokenized.stats.total_physical_reads_per_sec	每秒實體讀取總數
db.sql_tokenized.stats.total_worker_time_per_sec	CPU 時間總長 (毫秒)

SQL Server 的每次呼叫統計資料

下列指標提供 SQL Server SQL 陳述式的每次呼叫統計資料。

指標	單位
db.sql.stats.total_elapsed_time_per_call	每次執行的經過時間總長
db.sql.stats.total_rows_per_call	每秒執行處理總列數
db.sql.stats.total_logical_reads_per_call	每次執行邏輯讀取總數
db.sql.stats.total_logical_writes_per_call	每次執行邏輯寫入總數
db.sql.stats.total_physical_reads_per_call	每次執行實體讀取總數
db.sql.stats.total_worker_time_per_call	每次執行 CPU 時間總長 (毫秒)

下列指標提供 SQL Server SQL 摘要查詢的每次呼叫統計資料。

指標	單位
db.sql_tokenized.stats.total_elapsed_time_per_call	每次執行的經過時間總長
db.sql_tokenized.stats.total_rows_per_call	每秒執行處理總列數
db.sql_tokenized.stats.total_logical_reads_per_call	每次執行邏輯讀取總數

指標	單位
db.sql_tokenized.stats.total_logical_writes_per_call	每次執行邏輯寫入總數
db.sql_tokenized.stats.total_physical_reads_per_call	每次執行實體讀取總數
db.sql_tokenized.stats.total_worker_time_per_call	每次執行 CPU 時間總長 (毫秒)

RDS PostgreSQL 的 SQL 統計資料

Performance Insights 會針對每一次 SQL 呼叫和執行查詢的每一秒收集 SQL 統計資料。RDS for PostgreSQL 只收集摘要層級的 SQL 統計數資料。不會顯示陳述式層級的統計資料。

您可以在下文中了解 RDS for PostgreSQL 摘要層級統計資料的相關資訊。

主題

- [RDS PostgreSQL 的摘要統計數字](#)
- [RDS PostgreSQL 的每秒摘要統計資料](#)
- [RDS PostgreSQL 的每次呼叫摘要統計資料](#)

RDS PostgreSQL 的摘要統計數字

如要檢視 SQL 摘要統計資料，RDS PostgreSQL 必須載入 `pg_stat_statements` 程式庫。若為與 PostgreSQL 11 或更新版本相容的 PostgreSQL 資料庫執行個體，依預設，資料庫會載入此程式庫。對於與 PostgreSQL 10 或更早版本相容的 PostgreSQL 資料庫執行個體，請手動啟用此程式庫。若要手動啟用，請在與資料庫執行個體相關聯的資料庫參數群組中，將 `pg_stat_statements` 新增至 `shared_preload_libraries`。然後，重新啟動您的資料庫執行個體。如需更多詳細資訊，請參閱 [使用參數群組](#)。

Note

績效詳情只能在 `pg_stat_activity` 中收集未截斷的查詢的統計資料。根據預設，PostgreSQL 資料庫會截斷超過 1,024 位元組的查詢。若要增加查詢大小，請變更與資料

庫執行個體相關聯的資料庫參數群組中的 `track_activity_query_size` 參數。當您變更此參數時，需要重新啟動資料庫執行個體。

RDS PostgreSQL 的每秒摘要統計資料

PostgreSQL 資料庫執行個體有下列 SQL 摘要統計資料。

指標	單位
<code>db.sql_tokenized.stats.calls_per_sec</code>	每秒呼叫數
<code>db.sql_tokenized.stats.rows_per_sec</code>	每秒列數
<code>db.sql_tokenized.stats.total_time_per_sec</code>	每秒平均作用中執行數 (AAE)
<code>db.sql_tokenized.stats.shared_blks_hit_per_sec</code>	每秒區塊命中數
<code>db.sql_tokenized.stats.shared_blks_read_per_sec</code>	每秒區塊讀取數
<code>db.sql_tokenized.stats.shared_blks_dirtied_per_sec</code>	每秒區塊變動數
<code>db.sql_tokenized.stats.shared_blks_written_per_sec</code>	每秒區塊寫入數
<code>db.sql_tokenized.stats.local_blks_hit_per_sec</code>	每秒本機區塊命中數
<code>db.sql_tokenized.stats.local_blks_read_per_sec</code>	每秒本機區塊讀取數
<code>db.sql_tokenized.stats.local_blks_dirtied_per_sec</code>	每秒本機區塊變動數
<code>db.sql_tokenized.stats.local_blks_written_per_sec</code>	每秒本機區塊寫入數
<code>db.sql_tokenized.stats.temp_blks_written_per_sec</code>	每秒暫時寫入數

指標	單位
db.sql_tokenized.stats.temp_blks_read_per_sec	每秒暫時讀取數
db.sql_tokenized.stats.blk_read_time_per_sec	每秒平均並行讀取數
db.sql_tokenized.stats.blk_write_time_per_sec	每秒平均並行寫入數

RDS PostgreSQL 的每次呼叫摘要統計資料

下列指標提供 SQL 陳述式的每次呼叫統計數字。

指標	單位
db.sql_tokenized.stats.rows_per_call	每次呼叫列數
db.sql_tokenized.stats.avg_latency_per_call	每次呼叫平均延遲 (毫秒)
db.sql_tokenized.stats.shared_blks_hit_per_call	每次呼叫區塊命中數
db.sql_tokenized.stats.shared_blks_read_per_call	每次呼叫區塊讀取數
db.sql_tokenized.stats.shared_blks_written_per_call	每次呼叫區塊寫入數
db.sql_tokenized.stats.shared_blks_dirtied_per_call	每次呼叫區塊變動數
db.sql_tokenized.stats.local_blks_hit_per_call	每次呼叫本機區塊命中數
db.sql_tokenized.stats.local_blks_read_per_call	每次呼叫本機區塊讀取數
db.sql_tokenized.stats.local_blks_dirtied_per_call	每次呼叫本機區塊變動數
db.sql_tokenized.stats.local_blks_written_per_call	每次呼叫本機區塊寫入數

指標	單位
db.sql_tokenized.stats.temp_blks_written_per_call	每次呼叫暫時區塊寫入數
db.sql_tokenized.stats.temp_blks_read_per_call	每次呼叫暫時區塊讀取數
db.sql_tokenized.stats.blk_read_time_per_call	每次呼叫讀取時間 (毫秒)
db.sql_tokenized.stats.blk_write_time_per_call	每次呼叫寫入時間 (毫秒)

如需這些指標的詳細資訊，請參閱 PostgreSQL 文件中的 [pg_stat_statements](#)。

增強型監控中的作業系統指標

Amazon RDS 可針對資料庫執行個體執行所在的作業系統 (OS) 即時提供指標。RDS 會將增強型監控的指標交付到您的 Amazon CloudWatch 日誌帳戶。下表列出可使用 Amazon CloudWatch 日誌使用的作業系統指標。

主題

- [Db2、MariaDB、MySQL、甲骨文和 PostgreSQL 的作業系統度量](#)
- [Microsoft SQL Server 的作業系統指標](#)

Db2、MariaDB、MySQL、甲骨文和 PostgreSQL 的作業系統度量

群組	指標	主控台名稱	描述
General	engine	不適用	資料庫執行個體的資料庫引擎。
	instanceID	不適用	資料庫執行個體識別符。
	instanceResourceID	不適用	資料庫執行個體的不可變識別碼，對 AWS 區域而言是唯一的，也會作為日誌串流識別碼使用。
	numVCPU	不適用	資料庫執行個體的虛擬 CPU 數量。

群組	指標	主控台名稱	描述
	timestamp	不適用	擷取指標的時間。
	uptime	不適用	資料庫執行個體作用中的時間長度。
	version	不適用	作業系統指標的串流 JSON 格式的版本。
cpuUtilization	guest	CPU 訪客	客體程式使用中的 CPU 百分比。
	idle	CPU 閒置	CPU 閒置的百分比。
	irq	CPU IRQ	軟體中斷使用中的 CPU 百分比。
	nice	CPU 不錯	以最低優先順序執行之程式使用中的 CPU 百分比。
	steal	CPU 竊取	其他虛擬機器使用中的 CPU 百分比。
	system	CPU 系統	核心使用中的 CPU 百分比。
	total	CPU 總計	使用中的 CPU 總百分比。此數值包含 nice 值。
	user	CPU 使用者	使用者程式使用中的 CPU 百分比。
	wait	CPU 等待	等待 I/O 存取時未使用的 CPU 百分比。
diskIO	avgQueueLen	平均佇列大小	在 I/O 裝置的佇列中等待的請求數量。
	avgReqSz	平均請求大小	平均的請求大小，以 KB 為單位。
	await	磁碟 I/O 等待	回應請求時所需的毫秒數，包括佇列時間與服務時間。
	device	不適用	使用中磁碟裝置的識別符。
	readIOPS	讀取 IO/秒	每秒讀取操作的次數。

群組	指標	主控台名稱	描述
	readKb	讀取總數	讀取的 KB 總數。
	readKbPS	讀取 KB/秒	每秒讀取的 KB 總數。
	readLatency	讀取延遲	提交讀取 I/O 要求與完成之間的經過時間，以毫秒為單位。 此指標僅適用於 Amazon Aurora。
	readThroughput	讀取輸送量	要求資料庫叢集所使用的網路輸送量，以每秒位元組為單位。 此指標僅適用於 Amazon Aurora。
	rrqmPS	Rrqms	每秒佇列的合併讀取請求數量。
	tps	TPS	每秒的 I/O 交易數量。
	util	磁碟 I/O 公用程式	發出請求的 CPU 時間百分比。
	writeIOsPS	寫入 IO/秒	每秒寫入操作的次數。
	writeKb	寫入總計	寫入的 KB 總數。
	writeKbPS	寫入 KB/秒	每秒寫入的 KB 總數。
	writeLatency	寫入延遲	送出寫入 I/O 要求與完成之間的平均經過時間 (毫秒)。 此指標僅適用於 Amazon Aurora。
	writeThroughput	寫入輸送量	資料庫叢集回應所使用的網路輸送量，以每秒位元組為單位。 此指標僅適用於 Amazon Aurora。
	wrqmPS	Wrqs	每秒佇列的合併寫入請求數量。

群組	指標	主控台名稱	描述
physicalDeviceIO	avgQueueLen	實體裝置平均佇列大小	在 I/O 裝置的佇列中等待的請求數量。
	avgReqSz	實體裝置平均請求大小	平均的請求大小，以 KB 為單位。
	await	實體裝置磁碟 I/O 等待	回應請求時所需的毫秒數，包括佇列時間與服務時間。
	device	不適用	使用中磁碟裝置的識別符。
	readIOsPS	每秒實體裝置讀取 IO	每秒讀取操作的次數。
	readKb	實體裝置讀取總數	讀取的 KB 總數。
	readKbPS	每秒實體裝置讀取 Kb	每秒讀取的 KB 總數。
	rrqmPS	實體裝置 Rrqms	每秒佇列的合併讀取請求數量。
	tps	實體裝置 TPS	每秒的 I/O 交易數量。
	util	實體裝置磁碟 I/O 公用程式	發出請求的 CPU 時間百分比。
	writeIOsPS	每秒實體裝置寫入 IO	每秒寫入操作的次數。
	writeKb	實體裝置寫入總計	寫入的 KB 總數。
	writeKbPS	每秒實體裝置寫入 Kb	每秒寫入的 KB 總數。

群組	指標	主控台名稱	描述
	wrqmPS	實體裝置 Wrqms	每秒佇列的合併寫入請求數量。
fileSys	maxFiles	最大節點	檔案相同可建立的最大檔案數量。
	mountPoint	不適用	指向檔案系統的路徑。
	name	不適用	檔案系統的名稱。
	total	檔案系統總數	檔案系統可用的磁碟空間總容量，以 KB 為單位。
	used	已使用的檔案系統	檔案系統中的檔案已使用的磁碟空間容量，以 KB 為單位。
	usedFilePercent	已使用的節點	使用中的可用檔案百分比。
	usedFiles	已使用比率	檔案系統中的檔案數量。
loadAverageMinute	usedPercent	已使用的檔案系統	使用中檔案系統磁碟空間的百分比。
	fifteen	平均載入時間 15 分鐘	過去 15 分鐘內請求 CPU 時間的程序數量。
	five	平均載入時間 5 分鐘	過去 5 分鐘內請求 CPU 時間的程序數量。
	one	平均載入 1 分鐘	過去 1 分鐘內請求 CPU 時間的程序數量。
memory	active	作用中記憶體	已指派的記憶體數量，以 KB 為單位。

群組	指標	主控台名稱	描述
	buffers	緩衝記憶體	在寫入至儲存裝置之前，用於緩衝 I/O 請求的記憶體數量，以 KB 為單位。
	cached	快取記憶體	用來快取檔案系統輸入/輸出的記憶體數量。
	dirty	未清理的記憶體	RAM 之中已修改但尚未寫入至儲存裝置中相關資料區塊的記憶體分頁數量，以 KB 為單位。
	free	可用記憶體	未指派的記憶體數量，以 KB 為單位。
	hugePages Free	釋出的大內存頁	自由巨型分頁的數量。巨型分頁為 Linux 核心的功能。
	hugePages Rsvd	保留的大內存頁	已遞交的巨型分頁的數量。
	hugePages Size	大內存頁尺寸	每個巨型分頁的大小，以 KB 為單位。
	hugePages Surp	抑制的大內存頁	超過總數的可用剩餘巨型分頁的數量。
	hugePages Total	大內存頁總數	大內存頁總數。
	inactive	停用的記憶體	使用頻率最低的記憶體分頁數量，以 KB 為單位。
	mapped	對應記憶體	在程序地址空間內映射的檔案系統內容的總量，以 KB 為單位。
	pageTables	內存頁資料表	分頁表使用的記憶體數量，以 KB 為單位。
	slab	Slab 記憶體	個重複使用的核心資料結構數量，以 KB 為單位。
	total	記憶體總計	記憶體總量，以 KB 為單位。

群組	指標	主控台名稱	描述
	writeback	回寫記憶體	RAM 之中仍被寫入至支援儲存裝置的中途分頁數量，以 KB 為單位。
network	interface	不適用	用於資料庫執行個體之網路介面的識別符。
	rx	RX	每秒接收的位元組數量。
	tx	TX	每秒上傳的位元組數量。
processList	cpuUsedPc	CPU %	程序所使用的 CPU 百分比。
	id	不適用	程序的識別符。
	memoryUsedPc	MEM%	程序使用的記憶體百分比。
	name	不適用	程序的名稱。
	parentID	不適用	程序之父程序的程序識別符。
	rss	RES	配置於程序的 RAM 數量，以 KB 為單位。
	tgid	不適用	執行緒群組識別符，它代表執行緒所屬之程序 ID 號碼。此識別符用於將來自相同程序的執行緒進行分組。
	vss	VIRT	配置於程序的虛擬記憶體數量，以 KB 為單位。
	swap	swap	交換
swap in		換入	從磁碟交換輸入的記憶體數量，以 KB 為單位。
swap out		換出	交換輸出到磁碟的記憶體數量，以 KB 為單位。
free		自由交換	可用的交換記憶體數量，以 KB 為單位。
committed		認可的交換	做為快取記憶體使用的 swap 記憶體數量，以 KB 為單位。

群組	指標	主控台名稱	描述
tasks	blocked	封鎖的任務	封鎖的任務數量。
	running	執行中的任務	執行中的任務數量。
	sleeping	睡眠的任務	睡眠中的任務數量。
	stopped	任務已停止	已停止的任務數量。
	total	任務總計	任務的總數。
	zombie	任務殭屍	與作用中父任務進行互動的子任務數量。

Microsoft SQL Server 的作業系統指標

群組	指標	主控台名稱	描述
General	engine	不適用	資料庫執行個體的資料庫引擎。
	instanceID	不適用	資料庫執行個體識別符。
	instanceResourceID	不適用	資料庫執行個體的不可變識別碼，對 AWS 區域而言是唯一的，也會作為日誌串流識別碼使用。
	numVCPU	不適用	資料庫執行個體的虛擬 CPU 數量。
	timestamp	不適用	擷取指標的時間。
	uptime	不適用	資料庫執行個體作用中的時間長度。
	version	不適用	作業系統指標的串流 JSON 格式的版本。
cpuUtilization	idle	CPU 閒置	CPU 閒置的百分比。
	kern	CPU 核心	核心使用中的 CPU 百分比。

群組	指標	主控台名稱	描述
	user	CPU 使用者	使用者程式使用中的 CPU 百分比。
disks	name	不適用	磁碟的識別符。
	totalKb	總磁碟空間	磁碟空間總量，以 KB 為單位。
	usedKb	已使用的磁碟空間	磁碟上已使用的空間數量，以 KB 為單位。
	usedPc	已使用的磁碟空間比率	磁碟上已使用的空間百分比。
	availKb	可用磁碟空間	磁碟上可用的空間，以 KB 為單位。
	availPc	可用磁碟空間比率	磁碟上可用的空間百分比。
	rdCountPS	讀取/秒	每秒讀取操作的次數
	rdBytesPS	讀取 KB/秒	每秒讀取的位元組數量。
	wrCountPS	寫入 IO/秒	每秒寫入操作的次數。
	wrBytesPS	寫入 KB/秒	每秒寫入的位元組數量。
memory	commitTotKb	提交總計	使用中的分頁檔案支援的虛擬地址數量，亦即目前的確認負載。此數值包含主記憶體 (RAM) 與磁碟 (分頁檔)。
	commitLimitKb	最大提交	commitTotKb 指標的最大可能值。此數值是目前的分頁檔大小加上分頁表可用的實體記憶體，但已指派至非可分頁區域的 RAM 除外。
	commitPeakKb	提交峰值	自從作業系統上一次啟動以來，commitTotKb 指標的最大值。
	kernTotKb	核心記憶體總數	已分頁與非分頁核心集區中的記憶體總和，以 KB 為單位。

群組	指標	主控台名稱	描述
	kernPagedKb	分頁核心記憶體	已分頁的核心集區中的記憶體數量，以 KB 為單位。
	kernNonpagedKb	未分頁的 Kernel 記憶體	非分頁的核心集區中的記憶體數量，以 KB 為單位。
	pageSize	頁面大小	分頁的大小，以位元組為單位。
	physTotKb	記憶體總計	實體記憶體數量，以 KB 為單位。
	physAvailKb	可用的記憶體量	可用的實體記憶體數量，以 KB 為單位。
	sqlServerTotKb	SQL Server 總記憶體	已提供給 Microsoft SQL Server 的記憶體數量，以 KB 為單位。
	sysCacheKb	系統快取	系統快取記憶體數量，以 KB 為單位。
network	interface	不適用	用於資料庫執行個體之網路介面的識別符。
	rdBytesPS	網路讀取 KB/秒	每秒接收的位元組數量。
	wrBytesPS	網路寫入 KB/秒	每秒傳送的位元組數量。
processList	cpuUsedPc	已使用比率	程序所使用的 CPU 百分比。
	memUsedPc	MEM%	程序使用的記憶體總量百分比。
	name	不適用	程序的名稱。
	pid	不適用	程序的識別符。對於 Amazon RDS 所擁有的程序而言，此數值不存在。

群組	指標	主控台名稱	描述
	ppid	不適用	此程序之父程序的程序識別符。只有子程序才有此數值。
	tid	不適用	執行緒識別符。只有執行緒會有此數值。使用 pid 數值可識別擁有的程序。
	workingSetKb	不適用	私有工作集的記憶體數量加上程序使用中並可與其他程序共享的記憶體數量，以 KB 為單位。
	workingSetPrivKb	不適用	程序使用中但不可與其他程序共享的記憶體數量，以 KB 為單位。
	workingSetShareableKb	不適用	程序使用中並可與其他程序共享的記憶體數量，以 KB 為單位。
	virtKb	不適用	程序使用中的虛擬地址空間數量，以 KB 為單位。使用虛擬地址空間不一定表示會相應地使用磁碟或主記憶體分頁。
system	handles	控制碼	系統使用中的控制代碼數量。
	processes	Processes	在系統上執行的程序數量。
	threads	執行緒	在系統上執行的執行緒數量。

在 Amazon RDS 資料庫執行個體中監控事件、日誌和串流

當您監控 Amazon RDS Amazon 資料庫和其他 AWS 解決方案時，您的目標是維護下列項目：

- 可靠性
- 可用性
- 效能
- 安全

[監控 Amazon RDS 執行個體中的指標](#) 解釋如何使用指標監控執行個體。完整的解決方案也必須監視資料庫事件、記錄檔和活動串流。AWS 為您提供下列監視工具：

- Amazon EventBridge 是一種無伺服器事件匯流排服務，可讓您輕鬆地將應用程式與各種來源的資料連接起來。EventBridge 從您自己的應用程式、Software-as-a 服務 (SaaS) 應用程式和 AWS 服務提供即時資料串流。EventBridge 將數據路由到目標，例如 AWS Lambda。如此一來，您可以監控在服務中發生的事件，並建置事件導向的架構。如需詳細資訊，請參閱 [Amazon EventBridge 使用者指南](#)。
- Amazon CloudWatch 日誌提供了一種方法，可以從 Amazon RDS Amazon 執行個體和其他來源監控 AWS CloudTrail、存放和存取日誌檔。Amazon CloudWatch Logs 可以監控日誌檔中的資訊，並在符合特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。
- AWS CloudTrail 擷取由您或代表您進行的 API 呼叫和相關事件 AWS 帳戶。CloudTrail 將日誌檔交付到您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 位址，以及呼叫發生的時間。如需詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。
- 資料庫活動串流是一項 Amazon RDS 功能，提供有關您資料庫執行個體的近乎即時活動串流。Amazon RDS 會將活動推送至 Amazon Kinesis 資料串流。系統會自動建立 Kinesis 串流。在 Kinesis 中，您可以設定 AWS 服務 (例如 Amazon 資料 Firehose)，AWS Lambda 以及使用串流和存放資料。

主題

- [在 Amazon RDS 主控台中檢視日誌、事件和串流](#)
- [監控 Amazon RDS 事件](#)
- [監控 Amazon RDS 日誌檔案](#)
- [在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)

- [使用資料庫活動串流來監控 Amazon RDS](#)

在 Amazon RDS 主控台中檢視日誌、事件和串流

Amazon RDS 與 AWS 服務 整合，可在 RDS 主控台中顯示有關日誌、事件和資料庫活動串流的資訊。

RDS 資料庫執行個體的日誌和事件索引標籤會顯示以下資訊：

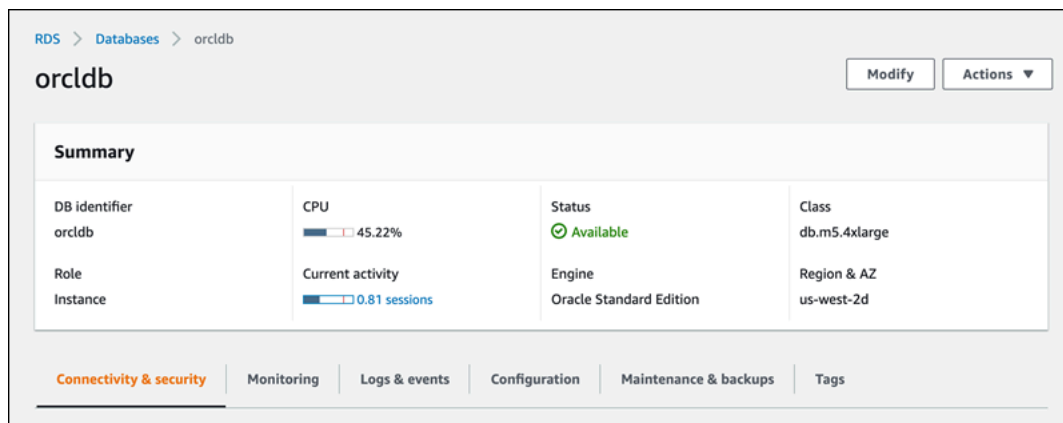
- Amazon CloudWatch alarms (Amazon CloudWatch 警示)：顯示您為 的資料庫執行個體設定的任何指標警示。如果您尚未設定警示，則可以在 RDS 主控台中加以建立。如需詳細資訊，請參閱 [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#)。
- Recent events (最近事件)：顯示 RDS 資料庫執行個體的事件摘要 (環境變更)。如需更多詳細資訊，請參閱 [檢視 Amazon RDS 事件](#)。
- Logs (日誌)：顯示 的資料庫執行個體產生的資料庫日誌檔案。如需更多詳細資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。

Configuration (組態) 索引標籤會顯示資料庫活動串流的資訊。

在 RDS 主控台中檢視 資料庫執行個體的日誌、事件和串流

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想要監控的 資料庫執行個體的名稱。

資料庫頁面隨即出現。下列範例顯示名為 orclb 的 Oracle 資料庫。



The screenshot shows the Amazon RDS console interface for a database instance named 'orclb'. The breadcrumb navigation at the top reads 'RDS > Databases > orclb'. Below the instance name, there are 'Modify' and 'Actions' buttons. A 'Summary' section displays the following details:

DB identifier	CPU	Status	Class
orclb	45.22%	Available	db.m5.4xlarge
Role	Current activity	Engine	Region & AZ
Instance	0.81 sessions	Oracle Standard Edition	us-west-2d

At the bottom, there is a navigation bar with tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Configuration' tab is currently selected.

4. 選擇 Logs & events (日誌和事件)。

此時將顯示「日誌和事件」區段。

The screenshot shows the Amazon RDS console interface. At the top, there are navigation tabs: Connectivity & security, Monitoring, **Logs & events**, Configuration, Maintenance & backups, and Tags. The 'Logs & events' tab is selected.

Under the 'Logs & events' tab, there are three main sections:



- CloudWatch alarms (0)**: Includes a refresh button, 'Edit alarm', and 'Create alarm' buttons. A search bar labeled 'Filter by alarms' is present.
- Recent events (2)**: Includes a refresh button and a search bar labeled 'Filter by db events'. It displays two events:

Time	System notes
February 04, 2022, 10:01:40 AM UTC	Backing up DB instance
February 04, 2022, 10:05:26 AM UTC	Finished DB Instance backup
- Logs (1478)**: Includes a refresh button, 'View', 'Watch', and 'Download' buttons. A search bar labeled 'Filter by db events' is present. Below the search bar is a table of log files:

Name	Last written	Logs
audit/ORCLB_j001_23080_20220202220030509284475170.aud	Wed Feb 02 2022 17:01:09 GMT-0500	649.6 kB
audit/ORCLB_j003_450_20220203220017482333361498.aud	Thu Feb 03 2022 17:00:32 GMT-0500	537.7 kB

5. 選擇 Configuration (組態)。

下列範例顯示資料庫執行個體的資料庫活動串流的狀態。

Configuration	Maintenance & backups	Tags
Storage		
Encryption		
Not enabled		
Storage type		
General Purpose SSD (gp2)		
Provisioned IOPS		
-		
Storage		
98 GiB		
Storage autoscaling		
Enabled		
Maximum storage threshold		
1000 GiB		
Performance Insights		
		Performance Insights enabled
		Yes
		AWS KMS key
		aws/rds 
		Retention period
		731 days
Published logs		
		CloudWatch Logs
		Alert
		Audit
		Listener
		Trace
Database activity stream		
		Status
		 Stopped

監控 Amazon RDS 事件

事件表示環境中有變更的事件。這可以是 AWS 環境、SaaS 合作夥伴服務或應用程式，或是自訂應用程式或服務。有關 RDS 事件的說明，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

主題

- [Amazon RDS 的事件概觀](#)
- [檢視 Amazon RDS 事件](#)
- [使用 Amazon RDS 事件通知](#)
- [建立由 Amazon RDS 事件觸發的規則](#)
- [Amazon RDS 事件類別和事件訊息](#)

Amazon RDS 的事件概觀

RDS 事件表示 Amazon RDS 環境中的變更。例如，當資料庫執行個體的狀態從待定變更為執行中時，Amazon RDS 會產生一個事件。Amazon RDS 以近乎即時 EventBridge 的方式交付活動。

Note

Amazon RDS 會全力發出事件。我們建議您避免編寫會根據通知事件的順序或存在的程式，因為這些事件可能會被移出序列或遺漏。

Amazon RDS 記錄與下列資源相關的事件：

- 資料庫執行個體

如需資料庫執行個體事件清單，請參閱 [資料庫執行個體事件](#)。

- 資料庫參數群組

如需有關資料庫參數群組事件的列表，請參閱 [資料庫參數群組事件](#)。

- 資料庫安全群組

如需有關資料庫安全群組事件的清單，請參閱 [資料庫安全群組事件](#)。

- 資料庫快照

如需資料庫快照事件的清單，請參閱 [資料庫快照事件](#)。

- RDS Proxy 事件

如需 RDS Proxy 事件的清單，請參閱 [RDS Proxy 事件](#)。

- 藍/綠部署事件

如需藍/綠部署事件的清單，請參閱 [藍/綠部署事件](#)。

此資訊包含下列項目：

- 事件的日期和時間
- 事件的來源名稱和來源類型
- 與事件相關聯的訊息
- 事件通知包括訊息傳送時的標籤，可能不會反映事件發生時的標籤

檢視 Amazon RDS 事件

您可以擷取資料庫 (主機台會顯示資訊) Amazon RDS 資源以下的版本資訊：

- 資源名稱
- 資源類型
- 事件的時間
- 活動的訊息摘要

透過存取事件 AWS Management Console，它會顯示過去 24 小時內的事件。您也可以使用[描述事件 AWS CLI 命令](#)或 [DescribeEventsRDS API 作業擷取事件](#)。如果您使用 AWS CLI 或 RDS API 來檢視事件，您可以擷取最多過去 14 天的事件。

Note

如果您需要更長的時間存放事件，可以將 Amazon RDS 事件傳送到 EventBridge。如需詳細資訊，請參閱 [建立由 Amazon RDS 事件觸發的規則](#)

如需有關 Amazon RDS 事件的說明，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

若要使 AWS CloudTrail 用 (包括要求參數) 存取有關事件的詳細資訊，請參閱 [CloudTrail 事件](#)。

主控台

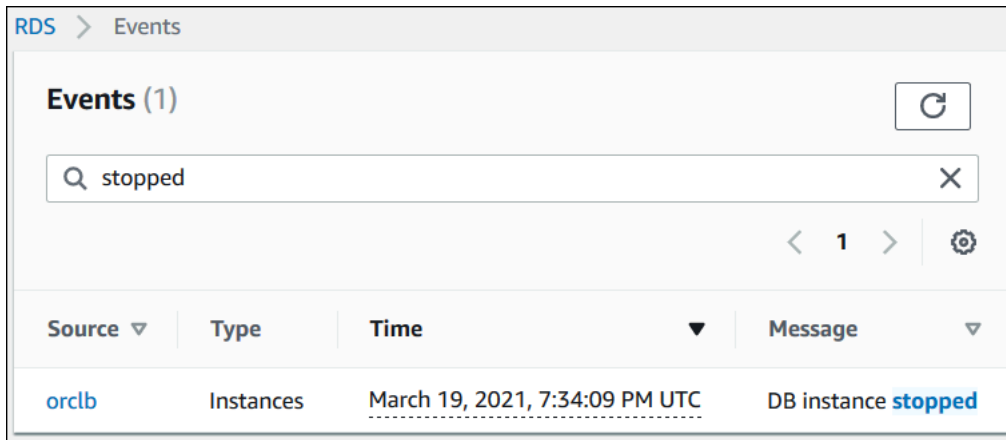
如要檢視過去 24 小時內的所有 Amazon RDS 事件

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Events (事件)。

可用的事件隨即會出現在清單中。

3. (選用) 輸入搜尋詞來篩選結果。

下列範例所顯示的事件清單，是依字元 **stopped** 篩選的結果。



Source ▾	Type	Time ▾	Message ▾
orclb	Instances	March 19, 2021, 7:34:09 PM UTC	DB instance stopped

AWS CLI

如要檢視過去一小時內產生的所有事件，請呼叫不含參數的 [describe-events](#)。

```
aws rds describe-events
```

下列範例輸出顯示資料庫執行個體已遭停止。

```
{
  "Events": [
    {
      "EventCategories": [
        "notification"
      ],
      "SourceType": "db-instance",
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:testinst",
      "Date": "2022-04-22T21:31:00.681Z",
      "Message": "DB instance stopped",
      "SourceIdentifier": "testinst"
    }
  ]
}
```

若要檢視過去 10080 分鐘 (7 天) 的所有 Amazon RDS 事件，請撥打[描述事件](#) AWS CLI 命令並將參數設定為。--duration 10080

```
aws rds describe-events --duration 10080
```

下列範例顯示資料庫執行個體 *test-instance* 在指定時間範圍內的事件。

```
aws rds describe-events \  
  --source-identifier test-instance \  
  --source-type db-instance \  
  --start-time 2022-03-13T22:00Z \  
  --end-time 2022-03-13T23:59Z
```

下列範例輸出顯示備份的狀態。

```
{  
  "Events": [  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Backing up DB instance",  
      "Date": "2022-03-13T23:09:23.983Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    },  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Finished DB Instance backup",  
      "Date": "2022-03-13T23:15:13.049Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    }  
  ]  
}
```

API

您可以呼叫 RDS API 操作並將 `Duration` 參數設定為 `14`，來檢視過去 14 天內的所有 Amazon [DescribeEvents](#) RDS 執行個體事件 20160。

使用 Amazon RDS 事件通知

Amazon RDS 使用 Amazon Simple Notification Service (Amazon SNS) 在 Amazon RDS 事件發生時提供通知。這些通知可以採用任何 Amazon SNS 在 AWS 區域中支援的通知形式，例如電子郵件、文字訊息或呼叫 HTTP 端點。

主題

- [Amazon RDS 事件通知概觀](#)
- [授予將通知發佈至 Amazon SNS 主題的許可](#)
- [訂閱 Amazon RDS 事件通知](#)
- [Amazon RDS 事件通知標籤與屬性](#)
- [列出 Amazon RDS 事件通知訂閱](#)
- [修改 Amazon RDS 事件通知訂閱](#)
- [將來源識別符新增至 Amazon RDS 事件通知訂閱](#)
- [將來源識別符從 Amazon RDS 事件通知訂閱中移除](#)
- [列出 Amazon RDS 事件通知類別](#)
- [刪除 Amazon RDS 事件通知訂閱](#)

Amazon RDS 事件通知概觀

Amazon RDS 將事件分成幾個類別供您訂閱，讓您在該類別的事件發生時收到通知。

主題

- [符合事件訂閱資格的 RDS 資源](#)
- [訂閱 Amazon RDS 事件通知的基本程序](#)
- [RDS 事件通知的傳遞](#)
- [Amazon RDS 事件通知的帳單](#)
- [使用 Amazon 的 Amazon RDS 活動示例 EventBridge](#)

符合事件訂閱資格的 RDS 資源

您可以針對下列資源訂閱事件類別：

- 資料庫執行個體

- 資料庫快照
- DB parameter group (資料庫參數群組)
- 資料庫安全群組
- RDS Proxy
- 自訂引擎版本

例如，如果您訂閱指定資料庫執行個體的備份類別，當發生會影響資料庫執行個體的備份相關事件時，您將會收到通知。如果您訂閱資料庫執行個體的組態變更類別，當資料庫執行個體變更時，您將會收到通知。當事件通知訂閱變更時，您也會收到通知。

您可能會想要建立數個不同的訂閱。例如，您可能會建立一個訂閱以接收所有資料庫執行個體的所有事件通知，並建立另一個訂閱以僅包含資料庫執行個體子集的重要事件。對於第二個訂閱，請在篩選條件中指定一或多個資料庫執行個體。

訂閱 Amazon RDS 事件通知的基本程序

訂閱 Amazon RDS 事件通知的程序如下：

1. 您可以使用 Amazon RDS 主控台或 API 建立 Amazon RDS 事件通知訂閱。AWS CLI

Amazon RDS 使用 Amazon SNS 主題的 ARN 來識別每個訂閱。Amazon RDS 主控台會在您建立訂閱時為您建立 ARN。使用 Amazon SNS 主控台、或 Amazon SNS API 來建立 ARN。AWS CLI

2. Amazon RDS 會將核准電子郵件或 SMS 訊息傳送至您在訂閱時提交的地址。
3. 您可以選擇收到的通知中的連結，以確認訂閱。
4. Amazon RDS 主控台會以您的訂閱狀態更新 My Event Subscriptions (我的事件訂閱) 區段。
5. Amazon RDS 會將通知傳送到您在建立訂閱時提供的地址。

若要了解使用 Amazon SNS 時的 Identity and Access Management，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 中的 Identity and Access Management](#)。

您可以用來處理 AWS Lambda 來自資料庫執行個體的事件通知。如需詳細資訊，請參閱 [AWS Lambda 開發人員指南中的與 Amazon RDS 搭配使用](#)。

RDS 事件通知的傳遞

Amazon RDS 會將通知傳送到您在建立訂閱時提供的地址。通知可以包含訊息屬性，其會提供有關訊息的結構化中繼資料。如需訊息屬性的詳細資訊，請參閱 [Amazon RDS 事件類別和事件訊息](#)

事件通知的傳送可能需要 5 分鐘。

Important

Amazon RDS 不保證事件串流中傳送事件的順序。事件順序可能會改變。

當 Amazon SNS 傳送通知至已訂閱之 HTTP 或 HTTPS 端點時，已傳送至端點的 POST 訊息具有包含 JSON 文件的訊息內文。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 訊息與 JSON 格式](#)。

您可以將 SNS 設定為以簡訊通知您。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [手機簡訊 \(SMS\)](#)。

若要關閉通知而不刪除訂閱項目，請在 Amazon RDS 主控台中，為 Enabled (啟用) 選擇 No (否)。或者，您可以將 Enabled 參數設置為 false 使用 AWS CLI 或 Amazon RDS API。

Amazon RDS 事件通知的帳單

Amazon RDS 事件通知的帳單是透過 Amazon SNS 傳送。使用事件通知需要支付 Amazon SNS 費用。如需 Amazon SNS 帳單的詳細資訊，請參閱 [Amazon Simple Notification Service 定價](#)。

使用 Amazon 的 Amazon RDS 活動示例 EventBridge

下列範例以 JSON 格式說明不同類型的 Amazon RDS 事件。如需演示如何擷取和檢視 JSON 格式事件的教學課程，請參閱 [教學：使用 Amazon 記錄資料庫執行個體狀態變更 EventBridge](#)。

主題

- [資料庫執行個體事件的範例](#)
- [資料庫參數群組事件的範例](#)
- [資料庫快照事件範例](#)

資料庫執行個體事件的範例

以下是 JSON 格式的資料庫執行個體事件範例。此事件顯示 RDS 對名為 my-db-instance 的執行個體執行了多可用區域容錯移轉。事件 ID 是 RDS-EVENT-0049。

```
{
  "version": "0",
```

```
"id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
"detail-type": "RDS DB Instance Event",
"source": "aws.rds",
"account": "123456789012",
"time": "2018-09-27T22:36:43Z",
"region": "us-east-1",
"resources": [
  "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
],
"detail": {
  "EventCategories": [
    "failover"
  ],
  "SourceType": "DB_INSTANCE",
  "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
  "Date": "2018-09-27T22:36:43.292Z",
  "Message": "A Multi-AZ failover has completed.",
  "SourceIdentifier": "my-db-instance",
  "EventID": "RDS-EVENT-0049"
}
}
```

資料庫參數群組事件的範例

以下是 JSON 格式的資料庫參數群組事件範例。事件顯示參數群組 `time_zone` 的參數 `my-db-param-group` 已更新。事件 ID 是 `RDS-EVENT-0037`。

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Parameter Group Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PARAM",
```

```
"SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
>Date": "2018-10-06T12:26:13.882Z",
>Message": "Updated parameter time_zone to UTC with apply method immediate",
>SourceIdentifier": "my-db-param-group",
>EventID": "RDS-EVENT-0037"
}
}
```

資料庫快照事件範例

以下是 JSON 格式的資料庫快照事件範例。此事件會顯示刪除名為 my-db-snapshot 的快照。事件 ID 是 RDS-EVENT-0041。

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
  ],
  "detail": {
    "EventCategories": [
      "deletion"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Deleted manual snapshot",
    "SourceIdentifier": "my-db-snapshot",
    "EventID": "RDS-EVENT-0041"
  }
}
```

授予將通知發佈至 Amazon SNS 主題的許可

若要授予 Amazon RDS 許可以將通知發佈至 Amazon Simple Notification Service (Amazon SNS) 主題，請將 AWS Identity and Access Management (IAM) 政策連接至目的地政策。有關許可的詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon Simple Notification Service 存取控制的範例案例](#)。

依預設，Amazon SNS 主題具有一項政策，可允許同一帳戶內的所有 Amazon RDS 資源向其發佈通知。您可以連接自訂政策以允許跨帳戶通知，或限制對特定資源的存取權。

以下是您連接至目的地 Amazon SNS 主題之 IAM 政策的範例。它將主題限制為名稱符合指定字首的資料庫執行個體。若要使用此政策，請指定下列值：

- Resource – Amazon SNS 主題的 Amazon Resource Name (ARN)
- SourceARN – 您的 RDS 資源 ARN
- SourceAccount – 您的 AWS 帳戶 ID

若要查看資源類型清單及其 ARN，請參閱服務授權參考中的 [Amazon RDS 定義的資源](#)。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.rds.amazonaws.com"
      },
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic_name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:prefix-*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

}

訂閱 Amazon RDS 事件通知

建立訂閱最簡單的方式是使用 RDS 主控台。如果您選擇使用 CLI 或 API 建立事件通知訂閱，您必須建立 Amazon Simple Notification Service 主題，並使用 Amazon SNS 主控台或 Amazon SNS API 訂閱該主題。您也將必須保留主題的 Amazon Resource Name (ARN)，因為在提交 CLI 命令或 API 操作時會用到它。如需建立和訂閱 SNS 主題的詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 入門](#)。

您可以指定想要收到通知的來源類型，以及觸發事件的 Amazon RDS 來源。

Source type (來源類型)

來源類型。例如：Source type (來源類型) 可能是 Instances (執行個體)。您必須選擇來源類型。

要包含的

正在產生事件的 Amazon RDS 資源。例如，您可以選擇 Select specific instances (選取特定執行個體) 然後選擇 myDBInstance1。

下表說明了指定或不指定要包含的 ## 時的結果。

要包含的資源	描述	範例
指定	RDS 只會通知您指定資源的所有資料庫執行個體事件。	如果您的 Source type (來源類型) 是 Instances (執行個體)，您的資源是 myDBInstance1，RDS 只會通知您有關 myDBInstance1。
未指定	您會收到所有 Amazon RDS 資源中指定來源類型事件的通知。	如果您的 Source type (來源類型) 是 Instances (執行個體)，RDS 會通知您帳戶中所有與執行個體相關的事件。

根據預設，Amazon SNS 主題訂閱者會接收發佈到主題的每個訊息。若要僅接收一部分的訊息，訂閱者必須將篩選政策指派給主題訂閱。如需有關 SNS 訊息篩選的詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 訊息篩選](#)

主控台

訂閱 RDS 事件通知

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。
3. 在 Event subscriptions (事件訂閱) 窗格中，選擇 Create event subscription (建立事件訂閱)。
4. 輸入您的訂閱詳細資訊，如下所示：
 - a. 在 Name (名稱) 中，輸入事件通知訂閱的名稱。
 - b. 對於 Send notification to: (傳送通知給：)，執行以下其中一項：
 - 選擇 New email topic (新的電子郵件主題)。輸入電子郵件主題的名稱和收件者清單。建議您將事件訂閱設定為與主要帳戶聯絡人相同的電子郵件地址。建議、服務事件和個人健康訊息會使用不同的通道傳送。訂閱相同電子郵件地址可確保所有郵件都合併在一個位置。
 - 選擇 Amazon Resource Name (ARN)。然後在 Amazon SNS 主題中選擇現有 Amazon SNS ARN。

如果您想要使用已為伺服器端加密 (SSE) 啟用的主題，請授予 Amazon RDS 存取 AWS KMS key 所需的許可。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的[啟用 AWS 服務的事件來源與加密主題之間的相容性](#)。
 - c. 在 Source type (來源類型) 中選擇來源類型。例如，選擇 Instances (執行個體) 或 Parameter groups (參數群組)。
 - d. 選擇您要接收通知的事件類型和資源。

下列範例設定名為 `testinst` 之資料庫執行個體的事件通知。

Source

Source type
Source type of resource this subscription will consume events from

Instances ▼

Instances to include
Instances that this subscription will consume events from

All instances

Select specific instances

Specific instances

Select instances ▼

testinst ✕

Event categories to include
Event categories that this subscription will consume events from

All event categories

Select specific event categories

e. 選擇 Create (建立)。

Amazon RDS 主控台顯示正在建立訂閱。

Event subscriptions (2)				
<input type="text" value="Filter event subscriptions"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Create event subscription"/> 				
<input type="checkbox"/>	Name	Status	Source Type	Enabled
<input type="checkbox"/>	Configchangerdspgres	active	Instances	Yes
<input type="checkbox"/>	Test	creating	Instances	Yes

AWS CLI

使用 AWS CLI [create-event-subscription](#) 命令來訂閱 RDS 事件通知。包含下列必要參數：

- `--subscription-name`
- `--sns-topic-arn`

Example

對於LinuxmacOS、或Unix：

```
aws rds create-event-subscription \
  --subscription-name myeventsubscription \
```

```
--sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS \  
--enabled
```

在Windows中：

```
aws rds create-event-subscription ^  
--subscription-name myeventsubscription ^  
--sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS ^  
--enabled
```

API

如要訂閱 Amazon RDS 事件通知，請呼叫 Amazon RDS API 函數 [CreateEventSubscription](#)。包含下列必要參數：

- SubscriptionName
- SnsTopicArn

Amazon RDS 事件通知標籤與屬性

當 Amazon RDS 將事件通知傳送至 Amazon Simple Notification Service (SNS) 或 Amazon EventBridge 時，通知會包含訊息屬性和事件標籤。RDS 會隨訊息單獨傳送訊息屬性，而事件標籤位於訊息內文中。使用訊息屬性和 Amazon RDS 標籤，將中繼資料新增至資源。您可以使用關於資料庫執行個體的表示法。如需標記 Amazon RDS Aurora 資源的詳細資訊，請參閱 [標記 Amazon RDS 資源](#)。

根據預設，Amazon SNS 和 Amazon EventBridge 會接收傳送給它們的每則訊息。SNS 和 EventBridge 可以篩選訊息並將通知傳送到偏好的通訊模式，例如電子郵件、簡訊或呼叫 HTTP 端點。

Note

透過電子郵件或簡訊傳送的通知不會有事件標籤。

下表顯示傳送到訂閱者的 RDS 事件的訊息屬性。

Amazon RDS 事件屬性	描述
eventID	RDS 事件訊息的識別符，例如 RDS-EVENT-0006。
資源	發出事件之資源的 ARN 識別符，例如 <code>arn:aws:rds:ap-southeast-2:123456789012:db:database-1</code> 。

RDS 標籤提供受服務事件影響之資源的相關資料。當通知傳送至 SNS 或 EventBridge 時，RDS 會在訊息內文中新增標籤的目前狀態。

如需有關 SNS 訊息篩選屬性的詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 訊息篩選](#)

如需為 EventBridge 篩選事件標籤的詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 [Amazon EventBridge 事件模式的內容篩選](#)。

如需篩選 SNS 的以承載為基礎標籤的詳細資訊，請參閱 <https://aws.amazon.com/blogs/compute/introducing-payload-based-message-filtering-for-amazon-sns/>

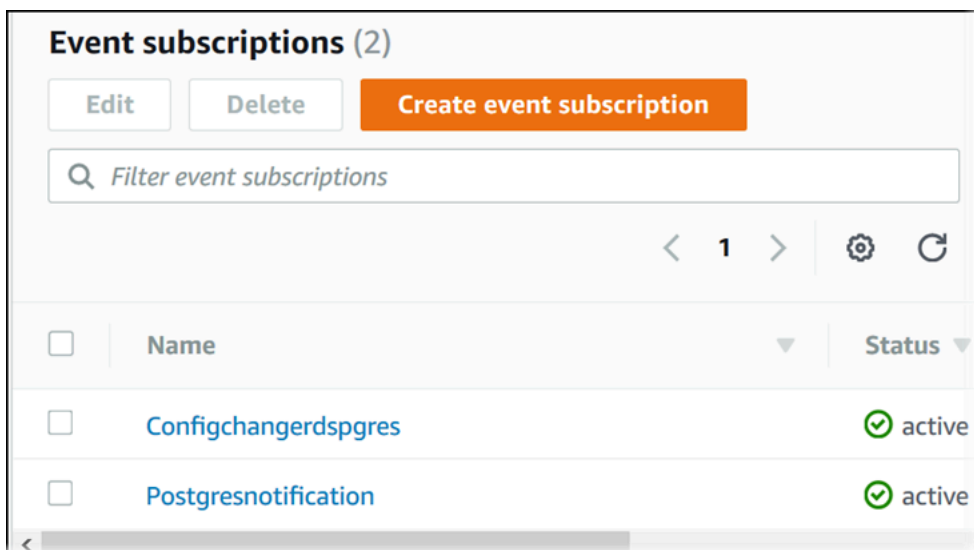
列出 Amazon RDS 事件通知訂閱

您可以列出您目前的 Amazon RDS 事件通知訂閱。

主控台

列出您目前的 Amazon RDS 事件通知訂閱

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。Event subscriptions (事件訂閱) 窗格顯示所有事件通知訂閱。



AWS CLI

使用 AWS CLI [describe-event-subscriptions](#) 命令來列出您目前的 Amazon RDS 事件通知訂閱。

Example

以下範例說明所有事件訂閱。

```
aws rds describe-event-subscriptions
```

以下範例說明 myfirsteventssubscription。

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

呼叫 Amazon RDS API [DescribeEventSubscriptions](#) 動作來列出您目前的 Amazon RDS 事件通知訂閱。

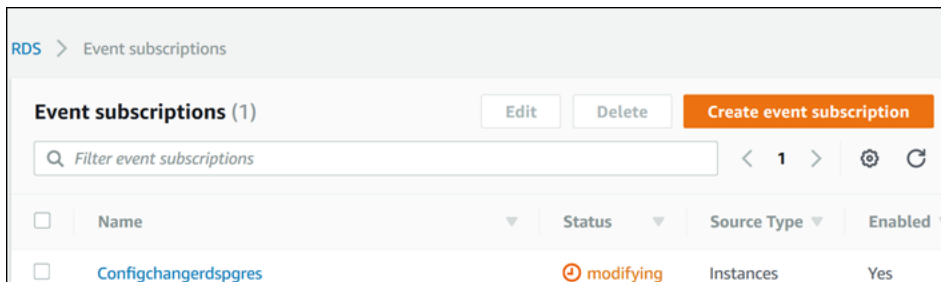
修改 Amazon RDS 事件通知訂閱

在您建立訂閱之後，您可以變更訂閱名稱、來源識別碼、類別或主題 ARN。

主控台

修改 Amazon RDS 事件通知訂閱

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。
3. 在 Event subscriptions (事件訂閱) 窗格中，選擇您要修改的訂閱，然後選擇 Edit (編輯)。
4. 在 Target (目標) 或 Source (來源) 區段中，對訂閱進行變更。
5. 選擇編輯。Amazon RDS 主控台顯示正在修改訂閱。



AWS CLI

使用 AWS CLI [modify-event-subscription](#) 命令來修改 Amazon RDS 事件通知訂閱。包含下列必要參數：

- `--subscription-name`

Example

下列範例程式碼啟動 `myeventsubscription`。

對於 Linux、macOS、或 Unix：

```
aws rds modify-event-subscription \  
  --subscription-name myeventsubscription \  
  --enabled
```

在Windows中：

```
aws rds modify-event-subscription ^  
  --subscription-name myeventsubscription ^  
  --enabled
```

API

若要修改 Amazon RDS 事件，請呼叫 Amazon RDS API 操作 [ModifyEventSubscription](#)。包含下列必要參數：

- SubscriptionName

將來源識別符新增至 Amazon RDS 事件通知訂閱

您可以將來源識別碼 (產生事件的 Amazon RDS 來源) 新增至現有的訂閱。

主控台

您可以在修改訂閱時，使用 Amazon RDS 主控台選取或取消選取來源識別碼，以輕鬆地新增或移除來源識別碼。如需更多詳細資訊，請參閱 [修改 Amazon RDS 事件通知訂閱](#)。

AWS CLI

使用 AWS CLI [add-source-identifier-to-subscription](#) 命令將來源識別碼新增至 Amazon RDS 事件通知訂閱。包含下列必要參數：

- `--subscription-name`
- `--source-identifier`

Example

下列範例將來源識別碼 `mysqldb` 新增至 `myrdseventsubscription` 訂閱。

對於LinuxmacOS、或Unix：

```
aws rds add-source-identifier-to-subscription \  
  --subscription-name myrdseventsubscription \  
  --source-identifier mysqldb
```

在Windows中：

```
aws rds add-source-identifier-to-subscription ^  
  --subscription-name myrdseventsubscription ^  
  --source-identifier mysqldb
```

API

呼叫 Amazon RDS API [AddSourceIdentifierToSubscription](#) 將來源識別碼新增至 Amazon RDS 事件通知訂閱。包含下列必要參數：

- `SubscriptionName`
- `SourceIdentifier`

將來源識別符從 Amazon RDS 事件通知訂閱中移除

如果您不再希望收到某個來源的事件通知，您可以從訂閱中移除該來源識別碼 (產生事件的 Amazon RDS 來源)。

主控台

您可以在修改訂閱時，使用 Amazon RDS 主控台選取或取消選取來源識別碼，以輕鬆地新增或移除來源識別碼。如需更多詳細資訊，請參閱 [修改 Amazon RDS 事件通知訂閱](#)。

AWS CLI

使用 AWS CLI [remove-source-identifier-from-subscription](#) 命令將來源識別碼從 Amazon RDS 事件通知訂閱中移除。包含下列必要參數：

- `--subscription-name`
- `--source-identifier`

Example

下列範例將來源識別碼 `mysqldb` 從 `myrdseventsubscription` 訂閱中移除。

對於LinuxmacOS、或Unix：

```
aws rds remove-source-identifier-from-subscription \  
  --subscription-name myrdseventsubscription \  
  --source-identifier mysqldb
```

在Windows中：

```
aws rds remove-source-identifier-from-subscription ^  
  --subscription-name myrdseventsubscription ^  
  --source-identifier mysqldb
```

API

使用 Amazon RDS API [RemoveSourceIdentifierFromSubscription](#) 命令將來源識別碼從 Amazon RDS 事件通知訂閱中移除。包含下列必要參數：

- `SubscriptionName`

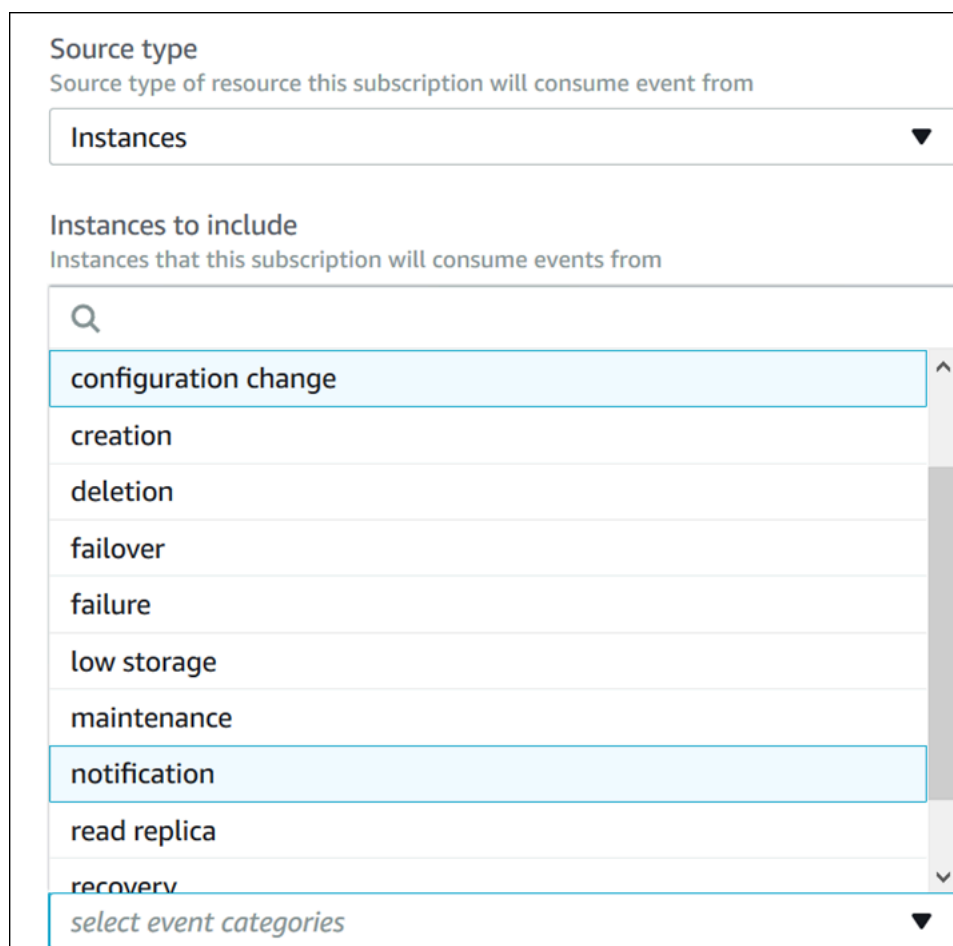
- **SourceIdentifier**

列出 Amazon RDS 事件通知類別

一種資源類型的所有事件會群組為各種類別。若要查看可用的類別清單，請使用下列程序。

主控台

當您建立或修改事件通知訂閱時，事件類別會顯示在 Amazon RDS 主控台。如需更多詳細資訊，請參閱 [修改 Amazon RDS 事件通知訂閱](#)。



The screenshot shows the Amazon RDS console interface for configuring event subscriptions. It features two main sections: 'Source type' and 'Instances to include'. The 'Source type' section has a dropdown menu currently set to 'Instances'. The 'Instances to include' section contains a search bar and a list of event categories. The categories listed are: configuration change, creation, deletion, failover, failure, low storage, maintenance, notification, read replica, and recovery. The 'notification' category is currently selected and highlighted in light blue. At the bottom of the list, there is a link that says 'select event categories'.

AWS CLI

使用 AWS CLI [describe-event-categories](#) 命令來列出 Amazon RDS 事件通知訂閱類別。此命令沒有要求的參數。

Example

```
aws rds describe-event-categories
```

API

使用 Amazon RDS API [DescribeEventCategories](#) 命令來列出 Amazon RDS 事件通知訂閱類別。此命令沒有要求的參數。

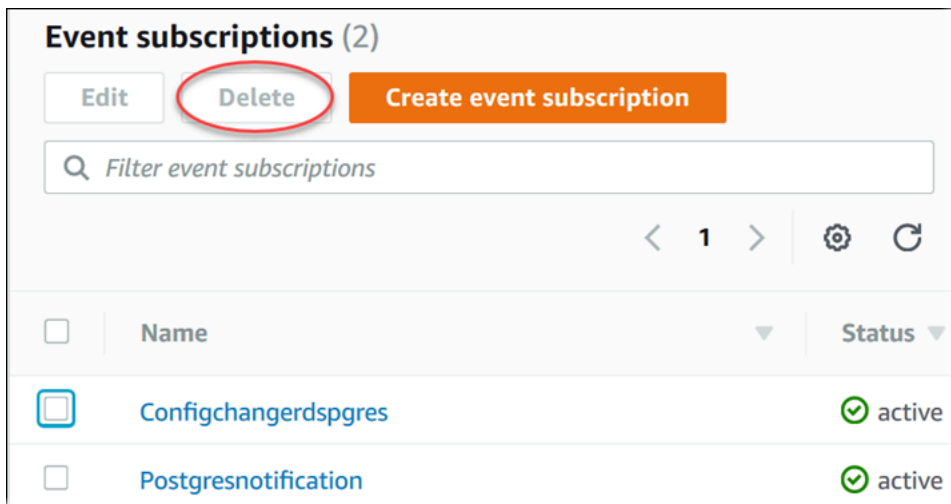
刪除 Amazon RDS 事件通知訂閱

您可以刪除不再需要的訂閱。該主題的所有訂閱者將不會再收到該訂閱指定的事件通知。

主控台

刪除 Amazon RDS 事件通知訂閱

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 DB Event Subscriptions (資料庫事件訂閱)。
3. 在 My DB Event Subscriptions (我的資料庫事件訂閱) 窗格中，選擇您要刪除的訂閱。
4. 選擇 Delete (刪除)。
5. Amazon RDS 主控台顯示正在刪除訂閱。



AWS CLI

使用 AWS CLI [delete-event-subscription](#) 命令來刪除 Amazon RDS 事件通知訂閱。包含下列必要參數：

- `--subscription-name`

Example

以下範例刪除訂閱 myrdssubscription。

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

使用 RDS API [DeleteEventSubscription](#) 命令來刪除 Amazon RDS 事件通知訂閱。包含下列必要參數：

- SubscriptionName

建立由 Amazon RDS 事件觸發的規則

使用 Amazon EventBridge，您可以自動化 AWS 服務並回應系統事件，例如應用程式可用性問題或資源變更。

主題

- [創建規則以將 Amazon RDS 事件發送到 Amazon EventBridge](#)
- [教學：使用 Amazon 記錄資料庫執行個體狀態變更 EventBridge](#)

創建規則以將 Amazon RDS 事件發送到 Amazon EventBridge

您可編寫簡單的規則，來指示您在意的 Amazon RDS 事件，以及事件符合規則時所要自動執行的動作。您可以設定各種目標，例如以 JSON 格式接收事件的 AWS Lambda 函數或 Amazon SNS 主題。例如，您可以將 Amazon RDS Amazon 設定為在建立或刪除資料庫執行個體 EventBridge 時，將事件傳送到 Amazon。如需詳細資訊，請參閱 [Amazon CloudWatch 活動使用者指南](#) 和 [Amazon EventBridge 使用者指南](#)。

建立由 RDS 事件觸發的規則：

1. 開啟主 CloudWatch 控制台，網址為 <https://console.aws.amazon.com/cloudwatch/>。
2. 在導覽窗格的 Events (事件) 中，選擇 Rules (規則)。
3. 選擇建立規則。
4. 針對 Event Source (事件來源)，請執行下列動作：
 - a. 選擇 Event Pattern (事件模式)。
 - b. 對於 Service Name (服務名稱)，選擇 Relational Database Service (RDS) (關聯式資料庫服務)。
 - c. 若為 Event Type (事件類型)，選擇觸發事件的 Amazon RDS 資源類型。例如，若的資料庫執行個體觸發事件，請選擇 RDS 資料庫執行個體事件。
5. 對於「目標」，請選擇「新增目標」，然後選擇偵測到所選類型的事件時要採取的 AWS 服務。
6. 在此區段的其他欄位中，輸入此目標類型的特定資訊 (如有需要)。
7. 對於許多目標類型，EventBridge 需要將事件傳送至目標的權限。在這些情況下，EventBridge 可以建立事件執行所需的 IAM 角色：
 - 若要自動建立 IAM 角色，請選擇 Create a new role for this specific resource (為此特定資源建立新角色)。

- 若要使用您之前建立的 IAM 角色，請選擇 Use existing role (使用現有角色)。
8. 您也可以選擇重複操作步驟 5 到 7，為此規則新增其他目標。
 9. 選擇 Configure details (設定詳細資訊)。針對 Rule definition (規則定義)，請輸入規則的名稱和描述。

規則名稱在此區域內必須是唯一的。

10. 選擇 Create rule (建立規則)。

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的[建立對事件觸發的 EventBridge 規則](#)。

教學：使用 Amazon 記錄資料庫執行個體狀態變更 EventBridge

在本教學中，您會建立記錄 Amazon RDS 執行個體狀態變更的 AWS Lambda 函數。然後，您建立一個規則，在現有 RDS 資料庫執行個體的狀態變更時執行該函數。本教學假設您擁有可以暫時關閉的小型執行中測試執行個體。

Important

請勿在執行生產資料庫執行個體上執行本教學課程。

主題

- [步驟 1：建立 AWS Lambda 函數](#)
- [步驟 2：建立規則](#)
- [步驟 3：測試規則](#)

步驟 1：建立 AWS Lambda 函數

建立 Lambda 函數以記錄狀態變更事件。當您在建立規則時指定此函數。

建立 Lambda 函數

1. 開啟主 AWS Lambda 控制台，[網址為 https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/)。
2. 如果您是第一次使用 Lambda，將會看到歡迎頁面。選擇 Get Started Now (立即開始)。否則，請選擇 Create function (建立函數)。
3. 選擇 Author from scratch (從頭開始撰寫)。

4. 在 Create function (建立函數) 頁面上，執行下列動作：
 - a. 輸入 Lambda 函數的名稱和描述。例如，將函數命名為 **RDSInstanceStateChange**。
 - b. 在 Runtime (執行時間) 中，選取 Node.js 16x。
 - c. 對於 Architecture (架構)，選擇 x86_64。
 - d. 對於 Execution role (執行角色)，執行下列任何一項：
 - 選擇 Create a new role with basic Lambda permissions (建立具備基本 Lambda 許可的新角色)。
 - 針對 Existing role (現有角色)，選擇 Use an existing role (使用現有的角色)。選擇您想使用的角色。
 - e. 選擇建立函數。
5. 在 RDS InstanceStateChange 頁面上，執行下列動作：
 - a. 在 Code source (程式碼來源) 中，選取 index.js。
 - b. 在 index.js 窗格中，刪除現有的程式碼。
 - c. 輸入下列程式碼：

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

- d. 選擇 Deploy (部署)。

步驟 2：建立規則

建立規則，在您啟動 Amazon RDS 執行個體時，執行您的 Lambda 函數。

若要建立 EventBridge 規則

1. 在 <https://console.aws.amazon.com/events/> 打開 Amazon EventBridge 控制台。
2. 在導覽窗格中，選擇規則。
3. 選擇建立規則。
4. 輸入規則的名稱和描述。例如，輸入 **RDSInstanceStateChangeRule**。
5. 選擇 Rule with an event pattern (具有事件模式的規則)，然後選擇 Next (下一步)。

6. 對於事件來源，請選擇AWS 事件或 EventBridge 合作夥伴事件。
7. 向下捲動到 Event pattern (事件模式) 區段中。
8. 在 Event source (事件來源)，選擇 AWS 服務。
9. 對於 AWS Service (服務)，選擇 Relational Database Service (RDS) (關聯式資料庫服務)。
10. 對於 Event type (事件類型)，選擇 RDS DB Instance Event (RDS 資料庫執行個體事件)。
11. 保留預設事件模式。然後選擇下一步。
12. 在目標類型欄位中，選擇 AWS 服務。
13. 對於 Select a target (選取目標)，選擇 Lambda function (Lambda 函數)。
14. 針對 Function (函數)，選擇您建立的 Lambda 函數。然後選擇下一步。
15. 在 Configure tags (設定標籤) 中，選擇 Next (下一步)。
16. 檢閱規則中的步驟。然後，選擇 Create role (建立角色)。

步驟 3：測試規則

若要測試您的規則，請關閉 RDS 資料庫執行個體。等待幾分鐘讓執行個體關閉，驗證您的 Lambda 函數是否被叫用。

停用資料庫執行個體以測試您的規則

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 停止 RDS 資料庫執行個體。
3. 在 <https://console.aws.amazon.com/events/> 打開 Amazon EventBridge 控制台。
4. 在導覽窗格中，選擇 Rules (規則)，然後選擇您建立的規則名稱。
5. 在規則詳細資料中，選擇監控。

您將被重定向到 Amazon CloudWatch 控制台。如果您未重新導向，請按一下檢視中的測量結果 CloudWatch。

6. 在 All metrics (所有指標) 中，選擇您建立的規則名稱。

該圖表應指示已叫用的規則。

7. 在導覽窗格中，選擇 Log groups (日誌群組)。
8. 選擇您的 Lambda 函數的日誌群組名稱 (/aws/lambda/**function-name**)。
9. 選擇日誌串流名稱以檢視函數為您啟動的執行個體所提供的資料。您應該會看到類似以下接收的事件：

```
{
  "version": "0",
  "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",
  "detail-type": "RDS DB Instance Event",
  "source": "aws.rds",
  "account": "111111111111",
  "time": "2021-03-19T19:34:09Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:111111111111:db:testdb"
  ],
  "detail": {
    "EventCategories": [
      "notification"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",
    "Date": "2021-03-19T19:34:09.293Z",
    "Message": "DB instance stopped",
    "SourceIdentifier": "testdb",
    "EventID": "RDS-EVENT-0087"
  }
}
```

如需更多 JSON 格式的 RDS 事件範例，請參閱 [Amazon RDS 的事件概觀](#)。

10. (選用) 完成後，您可以開啟 Amazon RDS 主控台並啟用您停止的執行個體。

Amazon RDS 事件類別和事件訊息

Amazon RDS 會在您可以使用 Amazon RDS 主控台或 API 訂閱的類別中產生大量事件。AWS CLI

主題

- [資料庫叢集事件](#)
- [資料庫執行個體事件](#)
- [資料庫參數群組事件](#)
- [資料庫安全群組事件](#)
- [資料庫快照事件](#)
- [資料庫叢集快照事件](#)
- [RDS Proxy 事件](#)
- [藍/綠部署事件](#)
- [自訂引擎版本事件](#)

資料庫叢集事件

下表顯示當資料庫叢集為來源類型時的事件類別和事件清單。

如需異地同步備份資料庫叢集部署的詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0016	重設主要憑證。	
建立	RDS-EVENT-0170	已建立資料庫叢集。	
容錯移轉	RDS-EVENT-0069	叢集容錯移轉失敗，請檢查叢集執行個體的運作狀態，然後再試一次。	
容錯移轉	RDS-EVENT-0070	再次提升上一個主要項目： <i>name</i> 。	
容錯移轉	RDS-EVENT-0071	已完成容錯移轉至資料庫執行個體： <i>name</i> 。	

類別	RDS 事件 ID	訊息	備註
容錯移轉	RDS-EVENT-0072	已啟動相同的 AZ 容錯移轉至資料庫執行個體： <i>name</i> 。	
容錯移轉	RDS-EVENT-0073	已啟動跨 AZ 容錯移轉至資料庫執行個體： <i>name</i> 。	
失敗	RDS-EVENT-0354	由於資源不相容，您無法建立資料庫叢集。 <i>##</i> 。	該 <i>##</i> 包含有關故障的詳細資料。
失敗	RDS-EVENT-0355	由於資源限制不足，因此無法創建數據庫集群。 <i>##</i> 。	該 <i>##</i> 包含有關故障的詳細資料。
全域容錯移轉	RDS-EVENT-0181	已啟動區域 <i>name</i> 中的全球轉換至資料庫叢集 <i>name</i> 。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。因為其他作業正在資料庫叢集上執行，所以此程序可能延遲。
全域容錯移轉	RDS-EVENT-0182	區域 <i>name</i> 中的舊主要資料庫叢集 <i>name</i> 成功關閉。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。全域資料庫中的舊主要執行個體不接受寫入。所有磁碟區都會同步化。
全域容錯移轉	RDS-EVENT-0183	等待跨全域叢集成員的資料同步。目前落後於主要資料庫叢集： <i>reason</i> 。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。複寫延遲發生在全域資料庫容錯移轉的同步階段期間。

類別	RDS 事件 ID	訊息	備註
全域容錯移轉	RDS-EVENT-0184	區域 <i>name</i> 中的新主要資料庫叢集 <i>name</i> 成功提升。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。 全域資料庫的磁碟區拓撲會以新的主要磁碟區重新建立。
全域容錯移轉	RDS-EVENT-0185	已完成區域 <i>name</i> 中的全球轉換至資料庫叢集 <i>name</i> 。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。 已在主要資料庫叢集上完成全球資料庫轉換。容錯移轉完成後，複本可能需要很長時間才能上線。
全域容錯移轉	RDS-EVENT-0186	已取消區域 <i>name</i> 中的全球轉換至資料庫叢集 <i>name</i> 。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。
全域容錯移轉	RDS-EVENT-0187	區域 <i>name</i> 中的全球轉換至資料庫叢集 <i>name</i> 失敗。	此為轉換操作的事件 (舊稱為「受管計畫容錯移轉」)。
全域容錯移轉	RDS-EVENT-0238	已完成區域 <i>name</i> 中的全球容錯移轉至資料庫叢集 <i>name</i> 。	
全域容錯移轉	RDS-EVENT-0239	區域 <i>name</i> 中的全域容錯移轉至資料庫叢集 <i>name</i> 失敗。	
全域容錯移轉	RDS-EVENT-0240	全球容錯移轉後，已開始重新同步處理區域 <i>name</i> 中資料庫叢集 <i>name</i> 的成員。	
全域容錯移轉	RDS-EVENT-0241	全球容錯移轉後，已完成重新同步處理區域 <i>name</i> 中資料庫叢集 <i>name</i> 的成員。	

類別	RDS 事件 ID	訊息	備註
維護	RDS-EVENT-0156	資料庫叢集有可用的資料庫引擎次要版本升級。	
維護	RDS-EVENT-0176	資料庫叢集引擎主要版本已升級。	
維護	RDS-EVENT-0286	資料庫叢集引擎版本升級已開始。	
維護	RDS-EVENT-0287	偵測到作業系統升級需求。	
維護	RDS-EVENT-0288	叢集作業系統升級開始中。	
維護	RDS-EVENT-0289	叢集作業系統升級已完成。	
notification	RDS-EVENT-0172	叢集已從 <i>name</i> 重新命名為 <i>name</i> 。	

資料庫執行個體事件

下表顯示當資料庫執行個體為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
可用性	RDS-EVENT-0004	資料庫執行個體關機。	
可用性	RDS-EVENT-0006	資料庫執行個體已重新啟動。	
可用性	RDS-EVENT-0022	重新啟動 mysql 時發生錯誤： <i>message</i> 。	重新啟動 MySQL 時發生錯誤。
可用性	RDS-EVENT-0221	資料庫執行個體已達到儲存已滿閾值，且資料庫已關閉。您可以增加配置的儲存來解決這個問題。	

類別	RDS 事件 ID	訊息	備註
可用性	RDS-EVENT-0222	資料庫執行個體 <i>name</i> 的可用儲存容量低至所配置儲存的 <i>percentage</i> [配置的儲存： <i>amount</i> ，可用的儲存： <i>amount</i>]。如果可用儲存低於 <i>amount</i> ，則資料庫會關閉以防止損毀。您可以增加配置的儲存來解決這個問題。	如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。
可用性	RDS-EVENT-0330	專用交易記錄磁碟區的可用儲存容量對資料庫執行個體##而言太低。記錄磁碟區可用儲存空間是已配置儲存體的###。[分配的存儲：數#，可用存儲：數#] 如果可用存儲量低於數#，數據庫將被關閉以防止損壞。您可以停用專用的交易記錄檔磁碟區來解決此問題。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。
可用性	RDS-EVENT-0331	專用交易記錄磁碟區的可用儲存容量對資料庫執行個體##而言太低。記錄磁碟區可用儲存體是已佈建儲存區的###。[佈建的儲存體：##，可用儲存空間：##] 您可以停用專用交易記錄檔磁碟區來解決此問題。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。
備份	RDS-EVENT-0001	備份資料庫執行個體。	
備份	RDS-EVENT-0002	已完成資料庫執行個體備份。	

類別	RDS 事件 ID	訊息	備註
備份	RDS-EVENT-0086	我們無法將選項群組 <i>name</i> 與資料庫執行個體 <i>name</i> 建立關聯。確認您的資料庫執行個體類別和組態支援選項群組 <i>name</i> 。若是如此，請驗證所有選項群組設定並重試。	如需更多詳細資訊，請參閱 使用選項群組 。
設定更改	RDS-EVENT-0011	已更新為使用資料庫 ParameterGroup ##。	
組態變更	RDS-EVENT-0012	套用修改至資料庫執行個體類別。	
組態變更	RDS-EVENT-0014	將修改套用至資料庫執行個體類別已完成。	
組態變更	RDS-EVENT-0016	重設主要憑證。	
組態變更	RDS-EVENT-0017	將修改套用至配置的儲存已完成。	
組態變更	RDS-EVENT-0018	正在將修改套用至配置的儲存。	
組態變更	RDS-EVENT-0024	正在套用修改以轉換為多可用區域資料庫執行個體。	
組態變更	RDS-EVENT-0025	套用修改以轉換為多可用區域資料庫執行個體已完成。	
組態變更	RDS-EVENT-0028	已停用自動備份。	
組態變更	RDS-EVENT-0029	套用修改以轉換為標準 (單一可用區域) 資料庫執行個體已完成。	

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0030	正在套用修改以轉換為標準 (單一可用區域) 資料庫執行個體。	
組態變更	RDS-EVENT-0032	已啟用自動備份。	
組態變更	RDS-EVENT-0033	有 <i>number</i> 個使用者符合主要使用者名稱；僅重設未繫結至特定主機的使用者。	
組態變更	RDS-EVENT-0067	無法重設您的密碼。錯誤資訊： <i>message</i> 。	
組態變更	RDS-EVENT-0078	監控間隔已變更為 <i>number</i> 。	增強型監控組態已變更。
組態變更	RDS-EVENT-0092	已完成資料庫參數群組的更新。	
組態變更	RDS-EVENT-0217	將自動擴展啟動的修改套用至已配置的儲存。	
組態變更	RDS-EVENT-0218	已完成將自動擴展啟動的修改套用至已配置的儲存。	
組態變更	RDS-EVENT-0295	儲存體組態升級已開始。	
組態變更	RDS-EVENT-0296	儲存體組態升級已完成。	
組態變更	RDS-EVENT-0332	專用記錄磁碟區已停用。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。
組態變更	RDS-EVENT-0333	停用專用記錄磁碟區已開始。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。
組態變更	RDS-EVENT-0334	啟用專用記錄磁碟區已經開始。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0335	專用記錄磁碟區已啟用。	如需詳細資訊，請參閱 專用記錄檔磁碟區 (DLV) 。
建立	RDS-EVENT-0005	已建立資料庫執行個體。	
刪除	RDS-EVENT-0003	已刪除資料庫執行個體。	
容錯移轉	RDS-EVENT-0013	多可用區域執行個體容錯移轉已啟動。	導致備用資料庫執行個體提升的多可用區域容錯移轉已啟動。
容錯移轉	RDS-EVENT-0015	多可用區域容錯移轉至備用完成 - DNS 傳播可能需要幾分鐘的時間。	導致備用資料庫執行個體提升的多可用區域容錯移轉已完成。DNS 傳輸到新的主要資料庫執行個體可能需要幾分鐘的時間。
容錯移轉	RDS-EVENT-0034	自資料庫執行個體最近發生的容錯移轉後，放棄使用者請求的容錯移轉。	Amazon RDS 不會因為資料庫執行個體最近發生的容錯移轉，而嘗試請求的容錯移轉。
容錯移轉	RDS-EVENT-0049	多可用區域執行個體容錯移轉已完成。	
容錯移轉	RDS-EVENT-0050	多可用區域執行個體啟用已啟動。	成功復原資料庫執行個體後，異地同步備份啟動已開始。
容錯移轉	RDS-EVENT-0051	多可用區域執行個體啟用已完成。	多可用區域啟用已完成。現在應可存取您的資料庫。
容錯移轉	RDS-EVENT-0065	已從部分容錯移轉復原。	

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0031	資料庫執行個體進入 <i>name</i> 狀態。RDS 建議您啟動 point-in-time-restore。	因為不相容的組態或基礎儲存問題，資料庫執行個體已失敗。開始資料 point-in-time-restore 料庫執行個體。
失敗	RDS-EVENT-0035	資料庫執行個體進入 <i>state</i> 。 <i>message</i> 。	資料庫執行個體有無效的參數。例如，如果此執行個體類別與記憶體相關的參數設定過高造成資料庫叢集無法啟動，則您的動作將為修改記憶體參數並重新啟動資料庫執行個體。
失敗	RDS-EVENT-0036	資料庫執行個體處於 <i>state</i> 。 <i>message</i> 。	資料庫執行個體位於不相容的網路中。部分指定的子網路 ID 無效或不存在。
失敗	RDS-EVENT-0058	Statspack 安裝失敗。 <i>message</i> 。	建立 Oracle Statspack 使用者帳戶 PERFSTAT 時發生錯誤。新增 STATSPACK 選項之前，請捨棄帳戶。
失敗	RDS-EVENT-0079	Amazon RDS 無法建立用於增強型監控的憑證，且此功能已停用。這可能是因為您的帳戶中 rds-monitoring-role 不存在且設定正確。如需進一步的詳細資訊，請參閱 Amazon RDS 文件中的疑難排解一節。	若沒有增強型監控 IAM 角色，無法啟用增強型監控。如需建立 IAM 角色的相關資訊，請參閱 為 Amazon RDS 增強型監控建立 IAM 角色 。

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0080	Amazon RDS 無法在您的執行個體上設定增強型監控： <i>name</i> ，且此功能已停用。這可能是因為您的帳戶中 rds-monitoring-role 不存在且設定正確。如需進一步的詳細資訊，請參閱 Amazon RDS 文件中的疑難排解一節。	因為組態變更期間發生錯誤，所以增強型監控已停用。可能是未正確設定增強型監控 IAM 角色。如需建立增強型監控 IAM 角色的相關資訊，請參閱 為 Amazon RDS 增強型監控建立 IAM 角色 。
失敗	RDS-EVENT-0081	Amazon RDS 無法為 <i>name</i> 選項建立憑證。這是因為未在您的帳戶中正確設定 <i>name</i> IAM 角色所致。如需進一步的詳細資訊，請參閱 Amazon RDS 文件中的疑難排解一節。	您用來為 SQL Server 原生備份與還原存取 Amazon S3 儲存貯體的 IAM 角色設定不正確。如需更多詳細資訊，請參閱 設定原生備份與還原 。
失敗	RDS-EVENT-0165	RDS Custom 資料庫執行個體在支援周邊外面。	您必須負責修正讓 RDS Custom 資料庫執行個體進入 unsupported-configuration 狀態的組態問題。如果問題出在 AWS 基礎結構上，您可以使用主控台或修正此問題。AWS CLI 如果問題與作業系統或資料庫組態相關，您可以登入主機進行修正。 如需詳細資訊，請參閱 RDS Custom 支援周邊 。

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0188	資料庫執行個體處於無法升級的狀態。 <i>message</i>	由於與資料字典相關的不相容性，Amazon RDS 無法將 MySQL 資料庫執行個體從 5.7 版升級至 8.0 版。資料庫執行個體已復原至 MySQL 5.7 版。如需詳細資訊，請參閱 從 MySQL 5.7 升級至 8.0 失敗後轉返 。
失敗	RDS-EVENT-0219	資料庫執行個體處於無效狀態。無需採取任何動作。稍後將重試自動擴展。	
失敗	RDS-EVENT-0220	資料庫執行個體處於先前擴展儲存操作的冷卻期。我們正在最佳化您的資料庫執行個體。這至少需要 6 個小時。無需採取任何動作。自動擴展將在冷靜期後重試。	
失敗	RDS-EVENT-0223	儲存自動擴展無法擴展儲存，原因如下： <i>reason</i> 。	
失敗	RDS-EVENT-0224	儲存自動擴展已觸發將達到或超出最大儲存閾值的擱置擴展儲存任務。增加最大儲存閾值。	
失敗	RDS-EVENT-0237	資料庫執行個體具有目前在可用區域中無法使用的儲存類型。稍後將重試自動擴展。	

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0254	此客戶帳戶的基礎儲存配額已超出限制。請增加允許的儲存配額，讓擴展可在執行個體上進行。	
失敗	RDS-EVENT-0278	資料庫執行個體建立失敗。##	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0279	RDS Custom 僅供讀取複本提升失敗。##	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0280	RDS Custom 無法升級資料庫執行個體，因為預先檢查失敗。##	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0281	RDS Custom 無法修改資料庫執行個體，因為預先檢查失敗。##	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0282	RDS Custom 無法修改資料庫執行個體，因為彈性 IP 權限不正確。請確認彈性 IP 地址已標記為 AWSRDSCustom。	
失敗	RDS-EVENT-0283	RDS Custom 無法修改資料庫執行個體，因為您的帳戶已達到彈性 IP 限制。釋放未使用的彈性 IP 或要求提高彈性 IP 地址限制的配額。	
失敗	RDS-EVENT-0284	RDS Custom 無法將執行個體轉換至高可用性，因為預先檢查失敗。##	該##包含有關故障的詳細資料。

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0285	RDS Custom 無法建立資料庫執行個體的最終快照，因為##。	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0306	儲存體組態升級失敗。請重試升級。	
失敗	RDS-EVENT-0315	無法將不相容的網路資料庫 <i>name</i> 移至可用狀態： <i>message</i>	資料庫網路組態無效。無法將資料庫從不相容的網路移至可用的網路。
失敗	RDS-EVENT-0328	無法將主機加入網域。執行個體#####的網域成員資格狀態已設定為「失敗」。	
失敗	RDS-EVENT-0329	無法將主機加入您的網域。在域加入過程中，Microsoft Windows 返回錯誤代碼##。驗證您的網路和權限設定，並發出modify-db-instance 要求以重新嘗試加入網域。	使用自我管理的作用中目錄時，請參閱 對自我管理 Active Directory 進行疑難排解 。
失敗	RDS-EVENT-0353	由於資源限制不足，因此無法建立資料庫執行個體。##。	該##包含有關故障的詳細資料。
失敗	RDS-EVENT-0356	RDS 無法在您的網域中設定 Kerberos 端點。這可能會阻止您的資料庫執行個體進行 Kerberos 驗證。驗證資料庫執行個體和網域控制站之間的網路組態。	

類別	RDS 事件 ID	訊息	備註
低儲存	RDS-EVENT-0007	配置的儲存已用盡。請配置額外的儲存來解決此問題。	已經使用為資料庫執行個體分配的儲存。若要解決此問題，為此資料庫執行個體分配額外儲存空間。如需詳細資訊，請參閱 RDS 常見問答集 。您可以使用 Free Storage Space (可用儲存空間) 指標來監控資料庫執行個體的儲存空間。
低儲存	RDS-EVENT-0089	資料庫執行個體 <i>name</i> 的可用儲存容量低至所佈建儲存體的 <i>percentage</i> [佈建的儲存體： <i>size</i> ，可用的儲存體： <i>size</i>]。您可能想要增加已佈建的儲存體來解決此問題。	資料庫執行個體已消耗超過其分配儲存容量的 90%。您可以使用 Free Storage Space (可用儲存空間) 指標來監控資料庫執行個體的儲存空間。
低儲存	RDS-EVENT-0227	Aurora 叢集的儲存體極低，只剩下 <i>amount</i> TB。請採取措施來減少叢集上的儲存負載。	Aurora 儲存子系統的空間不足。
維護	RDS-EVENT-0026	正在將離線修補程式套用至資料庫執行個體。	資料庫執行個體的離線維護正在進行。資料庫執行個體目前無法使用。
維護	RDS-EVENT-0027	已完成將離線修補程式套用至資料庫執行個體。	資料庫執行個體的離線維護已完成。資料庫執行個體現在可用。
維護	RDS-EVENT-0047	已修補資料庫執行個體。	
維護	RDS-EVENT-0155	資料庫執行個體有可用的資料庫引擎次要版本升級。	

類別	RDS 事件 ID	訊息	備註
維護	RDS-EVENT-0264	資料庫引擎版本升級已開始預先檢查。	
維護	RDS-EVENT-0265	資料庫引擎版本升級已完成預先檢查。	
維護	RDS-EVENT-0266	資料庫執行個體的停機時間已開始。	
維護	RDS-EVENT-0267	引擎版本升級已開始。	
維護	RDS-EVENT-0268	引擎版本升級已完成。	
維護	RDS-EVENT-0269	升級後任務正在進行中。	
維護	RDS-EVENT-0270	資料庫引擎版本升級失敗。 引擎版本升級復原成功。	
維護, 故障	RDS-EVENT-0195	<i>message</i>	更新 Oracle 時區檔案失敗。如需詳細資訊, 請參閱 Oracle 時區檔案自動升級 。
維護, 通知	RDS-EVENT-0191	時區檔案的新版本可用於更新。	當您更新 RDS for Oracle 資料庫引擎時, 如果您尚未選擇時區檔案升級, 且資料庫並未使用執行個體上可用的最新 DST 時區檔案, 則 Amazon RDS 會產生此事件。如需詳細資訊, 請參閱 Oracle 時區檔案自動升級 。
維護, 通知	RDS-EVENT-0192	已開始更新您的時區檔案。	您的 Oracle 時區檔案已開始升級。如需詳細資訊, 請參閱 Oracle 時區檔案自動升級 。

類別	RDS 事件 ID	訊息	備註
維護，通知	RDS-EVENT-0193	沒有更新可用於目前的時區檔案版本。	<p>您的 Oracle 資料庫執行個體正在使用最新的時區檔案版本，且下列任一陳述式為 true：</p> <ul style="list-style-type: none"> 您最近新增了 TIMEZONE_FILE_AUTOUPGRADE 選項。 您的 Oracle 資料庫引擎正在進行升級。 <p>如需詳細資訊，請參閱 Oracle 時區檔案自動升級。</p>
維護，通知	RDS-EVENT-0194	更新您的時區檔案已完成。	更新您的 Oracle 時區檔案已完成。如需詳細資訊，請參閱 Oracle 時區檔案自動升級 。
notification	RDS-EVENT-0044	<i>message</i>	這是操作員發出的通知。如需更多詳細資訊，請參閱事件訊息。
則通知	RDS-EVENT-0048	延遲資料庫引擎升級，因為此執行個體具有需要先升級的僅供讀取複本。	資料庫執行個體的修補作業已延遲。

類別	RDS 事件 ID	訊息	備註
則通知	RDS-EVENT-0054	<i>message</i>	您正在使用的 MySQL 儲存空間引擎並非 InnoDB，此為 Amazon RDS 建議使用的 MySQL 儲存空間引擎。如需 MySQL 儲存引擎的相關資訊，請參閱 支援的 RDS for MySQL 儲存引擎 。
notification	RDS-EVENT-0055	<i>message</i>	您擁有的資料庫執行個體表格數量已超過建議用於 Amazon RDS 的最佳實務。減少資料庫執行個體上的表格數量。如需建議最佳實務的詳細資訊，請參閱 Amazon RDS 基本操作準則 。
則通知	RDS-EVENT-0056	<i>message</i>	您的資料庫執行個體上的資料庫數量已超過建議用於 Amazon RDS 的最佳實務。減少資料庫執行個體上的資料庫數量。如需建議最佳實務的詳細資訊，請參閱 Amazon RDS 基本操作準則 。
則通知	RDS-EVENT-0064	TDE 加密金鑰已成功輪換。	如需建議最佳實務的詳細資訊，請參閱 Amazon RDS 基本操作準則 。

類別	RDS 事件 ID	訊息	備註
則通知	RDS-EVENT-0084	無法將資料庫執行個體轉換為多可用區域： <i>message</i> 。	您嘗試將資料庫執行個體轉換為多可用區域，但其包含多可用區域不支援的記憶體內檔案群組。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 的異地同步備份部署 。
則通知	RDS-EVENT-0087	資料庫執行個體已停止。	
notification	RDS-EVENT-0088	資料庫執行個體已啟動。	
notification	RDS-EVENT-0154	資料庫執行個體由於超過允許停止的時間上限而正在啟動。	
notification	RDS-EVENT-0157	無法修改資料庫執行個體類別。 <i>message</i> 。	RDS 無法修改資料庫執行個體類別，因為目標執行個體類別不支援來源資料庫執行個體上存在的資料庫數量。錯誤訊息顯示為：「執行個體有 N 個資料庫，但轉換後只支援 N。如需更多詳細資訊，請參閱 Microsoft SQL Server 資料庫執行個體的限制 。
則通知	RDS-EVENT-0158	資料庫執行個體處於無法升級的狀態： <i>message</i> 。	
則通知	RDS-EVENT-0167	<i>message</i>	RDS Custom 支援周邊組態已變更。

類別	RDS 事件 ID	訊息	備註
則通知	RDS-EVENT-0189	RDS 資料庫執行個體的 gp2 爆量餘額點數不足。若要解決此問題，請降低 IOPS 使用量，或修改儲存設定以啟用更高的效能。	RDS 資料庫執行個體的 gp2 爆量餘額點數不足。若要解決此問題，請降低 IOPS 使用量，或修改儲存設定以啟用更高的效能。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的輸入/輸出 額度和爆量效能 。
notification	RDS-EVENT-0225	配置的儲存體大小 <i>amount</i> GB 正在接近最大儲存體閾值 <i>amount</i> GB。增加最大儲存閾值。	當配置的儲存體達到最大儲存體閾值的 80% 時，系統會叫用此事件。若要避免此事件，請增加最大儲存閾值。
notification	RDS-EVENT-0231	資料庫執行個體的儲存修改時發生內部錯誤。修改請求處於待定狀態，將於稍後重試。	<p>僅供讀取複寫程序發生錯誤。如需更多詳細資訊，請參閱事件訊息。</p> <p>此外，請參閱資料庫引擎僅供讀取複本的疑難排解一節。</p> <ul style="list-style-type: none"> • 對 MariaDB 僅供讀取複本的問題進行故障診斷 • 對 SQL Server 僅供讀取複本問題進行故障診斷 • 對 MySQL 僅供讀取複本問題進行故障診斷 • 對 RDS for Oracle 複本進行故障診斷

類別	RDS 事件 ID	訊息	備註
notification	RDS-EVENT-0253	資料庫正在使用雙寫緩衝區。 <i>message</i> 。如需詳細資訊，請參閱 <i>name</i> 文件的 RDS Optimized Writes。	RDS 最佳化寫入與執行個體儲存組態不相容。如需詳細資訊，請參閱 使用 RDS Optimized Writes 改善 MySQL 的寫入效能 及 使用 Amazon RDS Optimized Writes 改善 MariaDB 的寫入效能 。 您可以執行儲存體組態升級，藉由 建立藍/綠部署 來啟用 Optimized Writes。
notification	RDS-EVENT-0297	資料庫執行個體 <i>name</i> 的儲存體組態支援 16384 GiB 的大小上限。執行儲存體組態升級，以支援大於 16384 GiB 的儲存體大小。	您無法將資料庫執行個體的配置儲存體大小增加至 16384 GiB 以上。若要克服此限制，請執行儲存體組態升級。如需詳細資訊，請參閱 升級資料庫執行個體的儲存檔案系統 。
notification	RDS-EVENT-0298	資料庫執行個體 <i>name</i> 的儲存體組態支援 2048 GiB 的資料表大小上限。執行儲存體組態升級，以支援大於 2048 GiB 的資料表大小。	具有此限制的 RDS MySQL 和 MariaDB 執行個體不能具有超過 2048 GiB 的資料表大小。若要克服此限制，請執行儲存體組態升級。如需詳細資訊，請參閱 升級資料庫執行個體的儲存檔案系統 。
notification	RDS-EVENT-0327	Amazon RDS 找不到秘密 <i>ARN ###</i> 。 <i>###</i>	

類別	RDS 事件 ID	訊息	備註
僅供讀取複本	RDS-EVENT-0045	複寫已停止。	資料庫執行個體的複寫已由於儲存體不足而停止。擴展儲存體或減少重做日誌的大小上限，讓複寫繼續進行。 ##### MiB ##### ##### MiB #####
僅供讀取複本	RDS-EVENT-0046	僅供讀取複本已恢復複寫。	此訊息會在您首次建立僅供讀取複本時出現，或做為確認複寫正常運作的監控訊息顯示。如果此訊息出現在 RDS-EVENT-0045 通知之後，則在錯誤或複寫停止之後已恢復複寫。
僅供讀取複本	RDS-EVENT-0057	已終止複寫串流。	
僅供讀取複本	RDS-EVENT-0062	已手動停止僅供讀取複本的複寫。	
僅供讀取複本	RDS-EVENT-0063	已重設從非 RDS 執行個體的複寫。	
僅供讀取複本	RDS-EVENT-0202	僅供讀取複本建立失敗。	
僅供讀取複本	RDS-EVENT-0357	複製通道##已啟動。	如需複製通道的相關資訊，請參閱 the section called “設定多重來源複製” 。
僅供讀取複本	RDS-EVENT-0358	複寫通道##已停止。	如需複製通道的相關資訊，請參閱 the section called “設定多重來源複製” 。
僅供讀取複本	RDS-EVENT-0359	已手動停止複製通道##。	如需複製通道的相關資訊，請參閱 the section called “設定多重來源複製” 。

類別	RDS 事件 ID	訊息	備註
僅供讀取複本	RDS-EVENT-0360	複製通道##已重設。	如需複製通道的相關資訊，請參閱 the section called “設定多重來源複製” 。
復原	RDS-EVENT-0020	資料庫執行個體的復原已啟動。復原時間依據恢復的資料量而有不同。	
復原	RDS-EVENT-0021	資料庫執行個體的復原已完成。	
復原	RDS-EVENT-0023	緊急快照請求： <i>message</i> 。	已請求手動備份，但 Amazon RDS 目前正在建立資料庫快照。請在 Amazon RDS 完成資料庫快照後，再次提交請求。
復原	RDS-EVENT-0052	多可用區域執行個體復原已啟動。	復原時間依據恢復的資料量而有不同。
復原	RDS-EVENT-0053	多可用區域執行個體復原已完成。擱置容錯移轉或啟動。	
復原	RDS-EVENT-0066	重新建立鏡像時，執行個體將降級： <i>message</i> 。	SQL Server 資料庫執行個體正在重新建立其鏡像。將會降低效能，直到鏡像重新建立完成。找到含有非 FULL 還原模型的資料庫。還原模型已變回 FULL，且已啟動鏡像還原。(<db name>: <recovery model found>[,...])”
復原	RDS-EVENT-0166	<i>message</i>	RDS Custom 資料庫執行個體在支援周邊之內。

類別	RDS 事件 ID	訊息	備註
復原	RDS-EVENT-0361	備用資料庫執行個體的復原已開始。	待命資料庫執行個體會在復原程序期間重建。在復原程序期間，資料庫效能會受到影響。
復原	RDS-EVENT-0362	備用資料庫執行個體的復原已完成。	待命資料庫執行個體會在復原程序期間重建。在復原程序期間，資料庫效能會受到影響。
還原	RDS-EVENT-0019	已從資料庫執行個體 <i>name</i> 還原至 <i>name</i> 。	資料庫執行個體已從 point-in-time 備份還原。
安全	RDS-EVENT-0068	解密 hsm 分割區密碼以更新執行個體。	RDS 正在解密 AWS CloudHSM 分割區密碼以對資料庫執行個體進行更新。如需詳細資訊，請參閱《AWS CloudHSM 使用者指南》中的 使用 AWS CloudHSM 進行 Oracle 資料庫透明資料加密 (TDE) 。
安全性修補程式	RDS-EVENT-0230	系統更新適用於您的資料庫執行個體。如需套用更新的相關資訊，請參閱《RDS 使用者指南》中的「維護資料庫執行個體」。	有新的作業系統更新可用。 新的次要版本作業系統更新適用於您的資料庫執行個體。如需套用更新的相關資訊，請參閱 使用強制作業系統更新 。

資料庫參數群組事件

下表顯示當資料庫參數群組為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0037	已將參數 <i>name</i> 更新為 <i>value</i> ，套用方法為 <i>method</i> 。	

資料庫安全群組事件

下表顯示當資料庫安全群組為來源類型時的事件類別和事件清單。

Note

資料庫安全群組是 EC2-Classic 的資源。EC2-Classic 在 2022 年 8 月 15 日淘汰。如果您還沒有從 EC2-Classic 遷移至 VPC，建議您盡快這麼做。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[從 EC2-Classic 遷移至 VPC](#)，以及部落格文章[EC2-Classic Networking is Retiring – Here's How to Prepare](#)。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0038	已將變更套用至安全群組。	
失敗	RDS-EVENT-0039	正以 <i>user</i> 身分撤銷授權。	由 <i>user</i> 擁有的安全群組不存在。已撤銷安全群組的授權，因為它無效。

資料庫快照事件

下表顯示當資料庫快照為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
建立	RDS-EVENT-0040	正在建立手動快照。	
建立	RDS-EVENT-0042	已建立手動快照	

類別	RDS 事件 ID	訊息	備註
建立	RDS-EVENT-0090	正在建立自動快照。	
建立	RDS-EVENT-0091	已建立自動快照。	
刪除	RDS-EVENT-0041	已刪除使用者快照。	
notification	RDS-EVENT-0059	已從區域 <i>name</i> 開始複製快照 <i>name</i> 。	這是跨區域快照複本。
則通知	RDS-EVENT-0060	已在 <i>number</i> 分鐘內，完成從區域 <i>name</i> 複製快照 <i>name</i> 。	這是跨區域快照複本。
則通知	RDS-EVENT-0061	已從區域 <i>name</i> 取消 <i>name</i> 的快照複製請求。	這是跨區域快照複本。
則通知	RDS-EVENT-0159	快照匯出任務失敗。	
notification	RDS-EVENT-0160	已取消快照匯出任務。	
notification	RDS-EVENT-0161	已完成快照匯出任務。	
notification	RDS-EVENT-0196	已開始複製區域 <i>name</i> 中的快照 <i>name</i> 。	這是本機快照複本。
則通知	RDS-EVENT-0197	已完成複製區域 <i>name</i> 中的快照 <i>name</i> 。	這是本機快照複本。
則通知	RDS-EVENT-0190	已取消區域 <i>name</i> 中 <i>name</i> 的快照複製請求。	這是本機快照複本。
還原	RDS-EVENT-0043	已從快照 <i>name</i> 還原。	正在從資料庫快照還原資料庫執行個體。

資料庫叢集快照事件

下表顯示當資料庫叢集快照為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
備份	RDS-EVENT-0074	正在建立手動叢集快照。	
備份	RDS-EVENT-0075	已建立手動叢集快照。	
備份	RDS-EVENT-0168	建立自動叢集快照。	
備份	RDS-EVENT-0169	已建立自動叢集快照。	

RDS Proxy 事件

下表顯示 RDS Proxy 為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0204	RDS 已修改資料庫代理 <i>name</i> 。	
組態變更	RDS-EVENT-0207	RDS 已修改資料庫代理 <i>name</i> 的端點。	
組態變更	RDS-EVENT-0213	RDS 偵測到新增了資料庫執行個體，並自動將其新增到資料庫代理 <i>name</i> 的目標群組。	
組態變更	RDS-EVENT-0213	RDS 偵測到建立了資料庫執行個體 <i>name</i> ，並自動將其新增至資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中。	
組態變更	RDS-EVENT-0214	RDS 偵測到刪除了資料庫執行個體 <i>name</i> ，並自動將其從資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中移除。	

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0215	RDS 偵測到刪除了資料庫叢集 <i>name</i> ，並自動將其從資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中移除。	
建立	RDS-EVENT-0203	RDS 已建立資料庫代理 <i>name</i> 。	
建立	RDS-EVENT-0206	RDS 已建立資料庫代理 <i>name</i> 的端點 <i>name</i> 。	
刪除	RDS-EVENT-0205	RDS 已刪除資料庫代理 <i>name</i> 。	
刪除	RDS-EVENT-0208	RDS 已刪除資料庫代理 <i>name</i> 的端點 <i>name</i> 。	
失敗	RDS-EVENT-0243	RDS 無法佈建代理 <i>name</i> 的容量，因為您的子網路中沒有足夠的可用 IP 地址： <i>name</i> 。若要修正此問題，請確定您的子網路具有最低數量的未使用 IP 地址，如 RDS Proxy 文件所建議。	若要確定執行個體類別的建議數量，請參閱 規劃 IP 地址容量 。
失敗	RDS-EVENT-0275	RDS ##### ##從用戶端到 Proxy 的同時連線要求數目已超過上限。	

藍/綠部署事件

下表顯示當藍/綠部署為來源類型時的事件類別和事件清單。

如需藍/綠部署的詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

類別	Amazon RDS 事件 ID	訊息	備註
建立	RDS-EVENT-0244	藍/綠部署任務已完成。您可以在部署之後，於綠色環境中對資料庫進行其他修改。	
失敗	RDS-EVENT-0245	建立藍/綠部署失敗，因為找不到 (來源/目標) 資料庫 (執行個體/叢集)。	
刪除	RDS-EVENT-0246	已刪除藍/綠部署。	
notification	RDS-EVENT-0247	從 <i>blue</i> 至 <i>green</i> 的轉換已開始。	
notification	RDS-EVENT-0248	已完成藍/綠部署上的切換。	
失敗	RDS-EVENT-0249	已取消藍/綠部署上的切換。	
notification	RDS-EVENT-0250	從主要/僅供讀取複本 <i>blue</i> 至 <i>green</i> 的轉換已開始。	
notification	RDS-EVENT-0251	從主要/僅供讀取複本 <i>blue</i> 至 <i>green</i> 的轉換已完成。已將 <i>blue</i> 重新命名為 <i>blue-old</i> ，並將 <i>green</i> 重新命名為 <i>blue</i> 。	
失敗	RDS-EVENT-0252	由於 <i>reason</i> ，已取消從主要/僅供讀取複本 <i>blue</i> 至 <i>green</i> 的轉換。	
notification	RDS-EVENT-0307	將 <i>blue</i> 轉換至 <i>green</i> 的序列同步已啟動。使用序列時轉換可能會導致停機時間延長。	

類別	Amazon RDS 事件 ID	訊息	備註
notification	RDS-EVENT-0308	將 <i>blue</i> 轉換至 <i>green</i> 的序列同步已完成。	
失敗	RDS-EVENT-0310	由於序列同步失敗，已取消將 <i>blue</i> 轉換至 <i>green</i> 的序列同步。	

自訂引擎版本事件

下表顯示自訂引擎版本為來源類型時的事件類別和事件清單。

類別	Amazon RDS 事件 ID	訊息	備註
建立	RDS-EVENT-0316	正在準備建立自訂引擎版本 <i>name</i> 。整個建立程序最多可能需要四個小時來完成。	
建立	RDS-EVENT-0317	正在建立自訂引擎版本 <i>name</i> 。	
建立	RDS-EVENT-0318	正在驗證自訂引擎版本 <i>name</i> 。	
建立	RDS-EVENT-0319	已成功建立自訂引擎版本 <i>name</i> 。	
建立	RDS-EVENT-0320	由於內部問題，RDS 無法建立自訂引擎版本 <i>name</i> 。我們正在解決問題，並在必要時與您聯絡。如需進一步協助，請聯絡 AWS 付費支援 。	

類別	Amazon RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0198	無法建立自訂引擎版本 <i>name</i> 。 <i>message</i>	該 ## 包含有關故障的詳細資料，例如遺失的檔案。
失敗	RDS-EVENT-0277	刪除自訂引擎版本 ## 的過程失敗。 ##	該 ## 包含有關故障的詳細資料。
還原	RDS-EVENT-0352	point-in-time 還原支援的最大資料庫計數已變更。	## 包含有關事件的詳細資料。

監控 Amazon RDS 日誌檔案

每個 RDS 資料庫引擎都會產生日誌，可供您存取進行稽核和疑難排解。日誌類型視您的資料庫引擎而定。

您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 Amazon RDS API 來存取資料庫日誌。您無法檢視、查看或下載交易記錄。

主題

- [檢視並列出資料庫日誌檔案](#)
- [下載資料庫日誌檔案](#)
- [查看資料庫日誌檔案](#)
- [將資料庫日誌發佈至 Amazon CloudWatch Logs](#)
- [使用 REST 讀取日誌檔案內容](#)
- [MariaDB 資料庫日誌檔案](#)
- [Microsoft SQL Server 資料庫日誌檔案](#)
- [MySQL 資料庫日誌檔案](#)
- [Oracle 資料庫日誌檔案](#)
- [RDS for PostgreSQL 資料庫日誌檔](#)

檢視並列出資料庫日誌檔案

您可使用 AWS Management Console 來檢視 Amazon RDS DB 引擎的資料庫日誌檔案。您可以使用 AWS CLI 或 Amazon RDS API 來列出可用於下載或監控的記錄檔案。

Note

若您無法檢視現有 RDS for Oracle 資料庫執行個體的日誌檔案清單，請重新開啟執行個體以檢視清單。

主控台

檢視資料庫日誌檔案

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。

2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想檢視的日誌檔案所在的資料庫執行個體的名稱。
4. 選擇 Logs & events (日誌與事件) 標籤。
5. 向下捲動至 Logs (日誌) 區段。
6. (選用) 輸入搜尋詞來篩選結果。
7. 選擇您想要檢視的日誌，然後選擇 View (檢視)。

AWS CLI

若要檢視資料庫執行個體的可用執行個體日誌檔案，請使用 AWS CLI [describe-db-log-files](#) 命令。

下列範例會回傳名為 my-db-instance 的資料庫執行個體之日誌檔案列表。

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

RDS API

若要列出資料庫執行個體的可用資料庫日誌檔案，請使用 Amazon RDS API [DescribeDBLogFiles](#) 動作。

下載資料庫日誌檔案

您可以使用 AWS Management Console、AWS CLI 或 API 來下載資料庫日誌檔案。

主控台

下載資料庫日誌檔案

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想檢視的日誌檔案所在的資料庫執行個體的名稱。
4. 選擇 Logs & events (日誌與事件) 標籤。
5. 向下捲動至 Logs (日誌) 區段。
6. 在 Logs (日誌) 區段中，選擇您想要下載的日誌旁的按鈕，然後選擇 Download (下載)。

7. 開啟提供之連結的內容 (右鍵) 功能表，然後選擇 Save Link As (另存連結為)。輸入您要儲存日誌檔案的位置，然後選擇 Save (儲存)。



AWS CLI

若要下載資料庫日誌檔案，請使用 AWS CLI 命令 [download-db-log-file-portion](#)。在預設情況下，這個命令只會下載日誌檔案的最新部分。不過，您可以指定參數 `--starting-token 0` 來下載整個檔案。

下列範例顯示下載名為 `log/ERROR.4` 之日誌檔案的完整內容方法，並將該檔案儲存於名為 `errorlog.txt` 的本機檔案中。

Example

對於LinuxmacOS、或Unix：

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier myexampledb \  
  --starting-token 0 --output text \  
  --log-file-name log/ERROR.4 > errorlog.txt
```

在Windows中：

```
aws rds download-db-log-file-portion ^  
  --db-instance-identifier myexampledb ^  
  --starting-token 0 --output text ^  
  --log-file-name log/ERROR.4 > errorlog.txt
```

RDS API

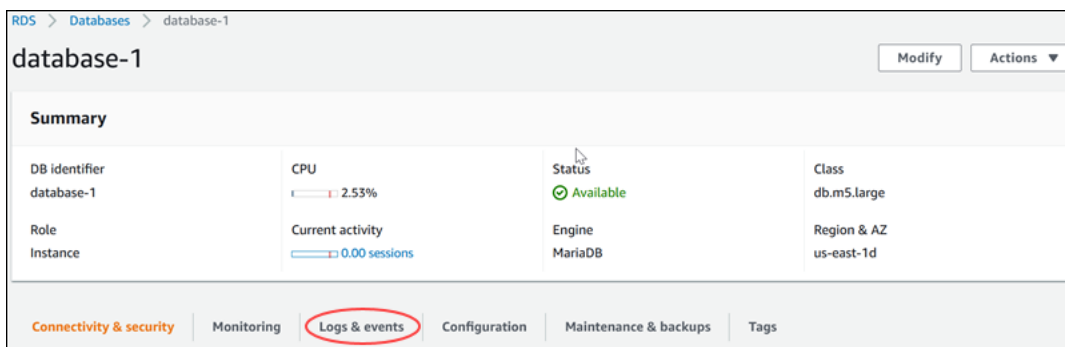
若要下載資料庫日誌檔案，請使用 Amazon RDS API [DownloadDBLogFilePortion](#) 動作。

查看資料庫日誌檔案

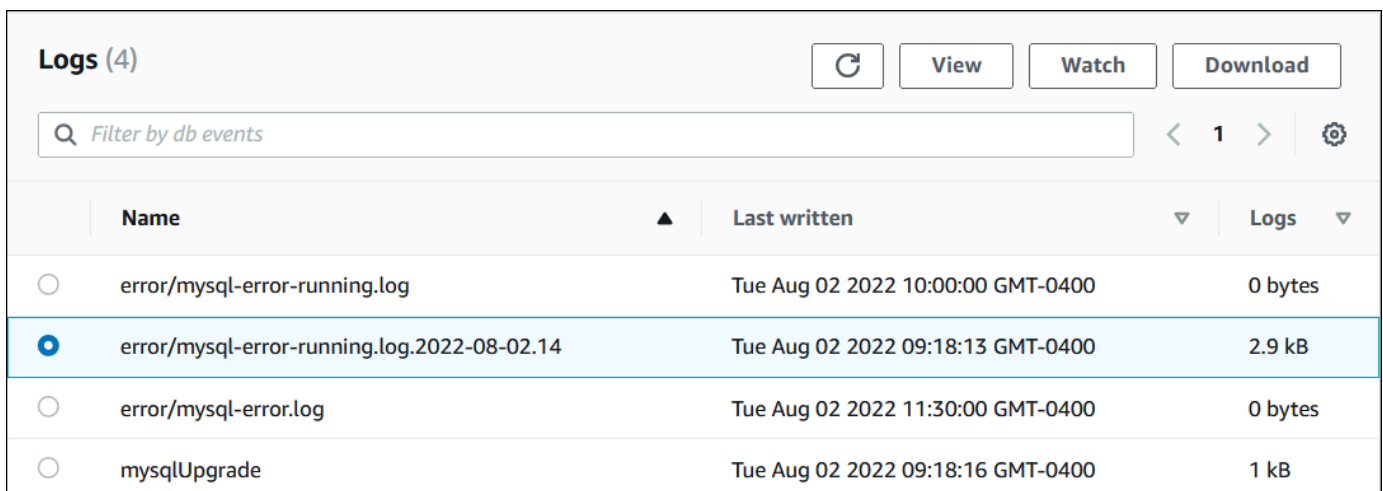
監視資料庫日誌檔等同於在 UNIX 或 Linux 系統上追蹤檔案。您可以使用 AWS Management Console 來觀看日誌檔。RDS 每 5 秒會重新整理日誌的結尾。

查看資料庫日誌檔案

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想檢視的日誌檔案所在的資料庫執行個體的名稱。
4. 選擇 Logs & events (日誌與事件) 標籤。



5. 在 Logs (日誌) 區段中，選擇日誌檔案，然後選擇 Watch (監看)。



RDS 會顯示日誌檔的尾端，如下列 MySQL 範例所示。

Watching Log: error/mysql-error-running.log.2022-08-02.14 (2.9 kB)

text: background:

```
2022-08-02T13:18:12.483484Z 0 [Warning] [MY-011068] [Server] The syntax 'skip_slave_start' is deprecated and
will be removed in a future release. Please use skip_replica_start instead.
2022-08-02T13:18:12.483491Z 0 [Warning] [MY-011068] [Server] The syntax 'slave_exec_mode' is deprecated and
will be removed in a future release. Please use replica_exec_mode instead.
2022-08-02T13:18:12.483498Z 0 [Warning] [MY-011068] [Server] The syntax 'slave_load_tmpdir' is deprecated and
will be removed in a future release. Please use replica_load_tmpdir instead.
2022-08-02T13:18:12.485031Z 0 [Warning] [MY-010101] [Server] Insecure configuration for --secure-file-priv:
Location is accessible to all OS users. Consider choosing a different directory.
2022-08-02T13:18:12.485063Z 0 [Warning] [MY-010918] [Server] 'default_authentication_plugin' is deprecated and
will be removed in a future release. Please use authentication_policy instead.
2022-08-02T13:18:12.485811Z 0 [System] [MY-010116] [Server] /rdsdbbin/mysql/bin/mysqld (mysqld 8.0.28)
starting as process 722
2022-08-02T13:18:12.559455Z 0 [Warning] [MY-010075] [Server] No existing UUID has been found, so we assume
that this is the first time that this server has been started. Generating a new UUID: 8f6bd551-1265-11ed-
840d-0251cdc2d067.
2022-08-02T13:18:12.580292Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-08-02T13:18:12.592437Z 1 [Warning] [MY-012191] [InnoDB] Scan path '/rdsdbdata/db/innodb' is ignored
because it is a sub-directory of '/rdsdbdata/db/'
2022-08-02T13:18:12.856761Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-08-02T13:18:13.126041Z 0 [Warning] [MY-013414] [Server] Server SSL certificate doesn't verify: unable to
get issuer certificate
2022-08-02T13:18:13.126139Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS.
Encrypted connections are now supported for this channel.
2022-08-02T13:18:13.158424Z 0 [System] [MY-010931] [Server] /rdsdbbin/mysql/bin/mysqld: ready for connections.
Version: '8.0.28' socket: '/tmp/mysql.sock' port: 3306 Source distribution.
----- END OF LOG -----
```

Watching error/mysql-error-running.log.2022-08-02.14, updates every 5 seconds.

將資料庫日誌發佈至 Amazon CloudWatch Logs

在內部部署資料庫中，資料庫日誌位於檔案系統上。Amazon RDS 不提供對資料庫執行個體的檔案系統上資料庫日誌的主機存取。因此，Amazon RDS 可讓您將資料庫日誌匯出至 [Amazon CloudWatch Logs](#)。您可以使用 CloudWatch Logs，執行日誌資料的即時分析。您也可以將資料存放在高耐久的儲存體中，並使用 CloudWatch Logs 代理程式來管理資料。

主題

- [RDS 與 CloudWatch Logs 整合概觀](#)
- [決定要發佈到 CloudWatch Logs 的日誌](#)
- [指定要發佈到 CloudWatch Logs 的日誌](#)
- [在 CloudWatch Logs 中搜尋和篩選您的日誌](#)

RDS 與 CloudWatch Logs 整合概觀

在 CloudWatch Logs 日誌串流是一系列共用相同來源的日誌事件。CloudWatch Logs 中的每個單獨日誌來源組成單獨的日誌串流。日誌群組是共用相同保留、監控和存取控制設定的日誌串流群組。

Amazon RDS 持續將您的資料庫執行個體日誌串流至日誌群組。例如，您所發佈每種類型的日誌都有一個日誌群組 `/aws/rds/instance/instance_name/log_type`。此日誌群組與產生日誌的資料庫執行個體位於相同的 AWS 區域中。

AWS 會長期保留發佈至 CloudWatch Logs 的日誌資料，除非您指定保留期間。如需更多資訊，請參閱[更改在 CloudWatch Logs 中的日誌資料保留期](#)。

決定要發佈到 CloudWatch Logs 的日誌

每個 RDS 資料庫引擎都支援自己的日誌集。若要瞭解資料庫引擎適用的選項，請檢閱下列主題：

- [the section called “將 MariaDB 日誌發佈到 Amazon 日誌 CloudWatch ”](#)
- [the section called “將 MySQL 日誌發佈到 Amazon CloudWatch 日誌”](#)
- [the section called “將 Oracle 日誌發佈到 Amazon CloudWatch 日誌”](#)
- [the section called “將 PostgreSQL 日誌發佈到 Amazon 日誌 CloudWatch ”](#)
- [the section called “將 SQL 伺服器日誌發佈到 Amazon CloudWatch 日誌”](#)

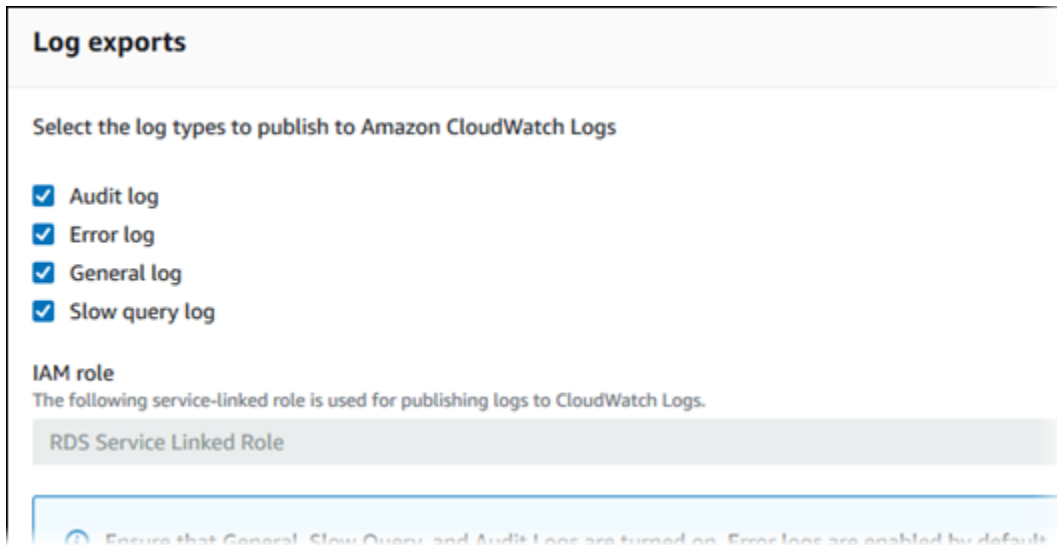
指定要發佈到 CloudWatch Logs 的日誌

您可以指定要在主控台中發佈的日誌。確定您在 AWS Identity and Access Management (IAM) 中有服務連結角色。如需服務連結角色的詳細資訊，請參閱[使用 Amazon RDS 的服務連結角色](#)。

若要指定要發佈的日誌

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 執行下列任何一項：
 - 選擇 Create database (建立資料庫)。
 - 從清單中，選擇您的資料庫，然後選擇 Modify (修改)。
4. 在 Logs exports (日誌匯出) 中，選擇要發佈的日誌。

下列範例會指定稽核日誌、錯誤日誌、一般日誌和慢查詢日誌。



在 CloudWatch Logs 中搜尋和篩選您的日誌

您可以使用 CloudWatch Logs 主控台搜尋與指定條件相符的日誌項目。您可以透過導向 CloudWatch Logs 主控台的 RDS 主控台存取日誌，或從 CloudWatch Logs 主控台直接存取。

若要使用 RDS 主控台搜尋 RDS 日誌

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇一個資料庫執行個體。
4. 選擇 Configuration (組態)。
5. 在 Published logs (發佈日誌) 下方，選擇您要檢視的資料庫日誌。

使用 CloudWatch Logs 主控台搜尋 RDS 日誌

1. 在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 在篩選方塊中，輸入 `/aws/rds`。
4. 針對 Log Groups (日誌群組)，輸入包含要搜尋之日誌串流的日誌群組名稱。
5. 對於 Log Streams (日誌串流)，選擇要搜尋的日誌串流名稱。
6. 在 Log events (日誌事件) 下方，輸入要使用的篩選條件語法。

如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的[搜尋和篩選日誌資料](#)。如需說明如何監控 RDS 日誌的教學課程，請參閱[使用 Amazon CloudWatch Logs、AWS Lambda 和 Amazon SNS 為 Amazon RDS 建置主動資料庫監控](#)。

使用 REST 讀取日誌檔案內容

Amazon RDS 提供允許存取資料庫執行個體日誌檔案的 REST 端點。若您需要編寫應用程式來串流 Amazon RDS 日誌檔案內容，此操作非常有幫助。

語法是：

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

下列是必要參數：

- *DBInstanceIdentifier* — 含有您想要下載之日誌檔案的資料庫執行個體名稱。
- *LogFileName* — 要下載的日誌檔案名稱。

回應包括要求之日誌檔案的內容，做為串流形式。

下列範例下載名為 log/ERROR.6 的日誌檔案供名為 sample-sql 的資料庫執行個體使用，位於 us-west-2 區域中。

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////
wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYa1FSn6UyJuEFTft9n0bg1x4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afb4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

若您指定不存在的資料庫執行個體，回應會包含下列錯誤：

- DBInstanceNotFound — *DBInstanceIdentifier* 不代表現有的資料庫執行個體。(HTTP 狀態碼：404)

MariaDB 資料庫日誌檔案

您可以監控 MariaDB 錯誤日誌、慢查詢日誌以及一般日誌。預設會產生 MariaDB 錯誤記錄，您可以透過在資料庫參數群組中設定參數，來產生慢查詢記錄和一般記錄。Amazon RDS 會輪換所有 MariaDB 記錄檔案；每種類型的間隔如下。

您可以透過 Amazon RDS 主控台、Amazon RDS API、Amazon RDS CLI 或 AWS 開發套件直接監控 MariaDB 日誌。您可以將日誌指向主要資料庫中的資料庫資料表並查詢該資料表，以存取 MariaDB 日誌。您可以使用 `mysqlbinlog` 公用程式來下載二進位日誌。

如需關於檢視、下載與查看資料庫日誌檔案的資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。

主題

- [存取 MariaDB 錯誤日誌](#)
- [存取 MariaDB 慢查詢日誌及一般日誌](#)
- [將 MariaDB 日誌發佈到 Amazon 日誌 CloudWatch](#)
- [日誌檔案大小](#)
- [管理表格式 MariaDB 日誌](#)
- [二進位日誌格式](#)
- [存取 MariaDB 二進位日誌](#)
- [二進位日誌註釋](#)

存取 MariaDB 錯誤日誌

MariaDB 錯誤日誌寫入於 `<host-name>.err` 檔案中。您可以使用 Amazon RDS 控制台查看此文件，您也可以使用 Amazon RDS API，Amazon RDS AWS CLI 或開發套件檢索日誌。`<host-name>.err` 檔案每 5 分鐘會清空一次，而其內容會附加於 `mysql-error-running.log`。`mysql-error-running.log` 檔案接著會於每個小時輪換，而過去 24 小時間產生的每小時檔案將會保留。每個日誌檔案的產生時間 (UTC 時區) 皆會附加於檔案名稱中。日誌檔案也有時間戳記，可協助您判定日誌項目寫入的時間。

MariaDB 只會在啟動、關機和發生錯誤時寫入錯誤日誌。資料庫執行個體可在未寫入新項目到錯誤日誌的情況下持續執行數小時或數日。若您沒有看到最近的項目，是因為伺服器未遇到需寫入日誌項目的錯誤。

存取 MariaDB 慢查詢日誌及一般日誌

您可在資料庫參數群組中設定參數，將 MariaDB 慢查詢日誌與一般日誌寫入至檔案或資料庫表格。如需建立和修改資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。您必須先設定這些參數，AWS CLI 才能在 Amazon RDS 主控台或使用 Amazon RDS API 或 AWS 開發套件檢視慢速查詢日誌或一般日誌。

您可以使用清單中的參數來控制 MariaDB 日誌：

- `slow_query_log`或`log_slow_query`：若要建立慢速查詢記錄，請設定為 1。預設值為 0。
- `general_log`：若要建立一般日誌，請設為 1。預設值為 0。
- `long_query_time`或`log_slow_query_time`：若要避免在慢速查詢記錄檔中記錄快速執行的查詢，請指定要記錄的最短查詢執行時間值 (以秒為單位)。預設為 10 秒，最短時間為 0。若 `log_output = FILE`，您可以指定以毫秒解析度為單位的浮點值。若 `log_output = TABLE`，您必須指定以秒為單位的整數值。只會記錄執行時間超過`long_query_time`或`log_slow_query_time`值的查詢。例如，`log_slow_query_time`將`long_query_time`或設定為 0.1 可防止記錄執行少於 100 毫秒的任何查詢。
- `log_queries_not_using_indexes`：若要將所有不使用索引的查詢記錄於慢查詢日誌中，請將此參數設為 1。預設為 0。將會記錄不使用索引的查詢，即使其執行時間低於 `long_query_time` 參數的值。
- `log_output` *option*：您可為 `log_output` 參數指定下列其中一個選項：
 - TABLE (預設) – 將一般查詢寫入 `mysql.general_log` 表格，而慢查詢則寫入 `mysql.slow_log` 表格。
 - FILE – 同時將一般與慢查詢日誌寫入檔案系統中。日誌檔案每小時輪換。
 - NONE – 停用日誌記錄。

記錄啟用時，Amazon RDS 輪換表格日誌或以規律間隔來刪除日誌檔案。此方法為預防措施，可降低大型日誌檔封鎖資料庫使用或影響效能的可能性。FILE 與 TABLE 日誌記錄運用以下方式執行輪換與刪除：

- 當 FILE 日誌記錄啟用時，每個小時將檢視日誌檔，而早於 24 小時前的日誌檔將會刪除。在部分情況下，刪除後剩餘的總日誌檔大小可能超過資料庫執行個體配得之空間的 2% 閾值。在這些情況中，最大的日誌檔將被刪除，直到日誌檔大小不再超過閾值。
- 當 TABLE 日誌記錄啟用時，在某些情況下，日誌資料表每 24 小時會輪換一次。在表格日誌使用的空間超過所配置儲存空間的 20% 時，就會發生輪換。如果所有日誌合併後的大小超過 10 GB 時，也會發生這種情況。若資料庫執行個體使用的空間總量高於資料庫執行個體配得之儲存空間的 90%

時，日誌輪換的閾值將會降低。在表格日誌使用的空間超過所配置儲存空間的 10% 時，就會輪換日誌表格。如果所有日誌合併後的大小超過 5 GB 時，也會轉換它們。

輪換日誌資料表時，目前日誌資料表會複製到備份日誌資料表，並移除目前日誌資料表中的項目。如果備份日誌資料表已存在，則其會在目前日誌資料表複製到備份之前刪除。如有需要，您可以查詢備份日誌資料表。mysql.general_log 資料表的備份日誌資料表名為 mysql.general_log_backup。mysql.slow_log 資料表的備份日誌資料表名為 mysql.slow_log_backup。

您可以呼叫 mysql.general_log 程序來輪換 mysql.rds_rotate_general_log 表格。您可以呼叫 mysql.slow_log 程序來輪換 mysql.rds_rotate_slow_log 表格。

表格日誌會在資料庫版本升級時輪換。

Amazon RDS 會在 Amazon RDS 中同時記錄 TABLE 與 FILE 日誌輪換，並傳送通知給您。

若要使用來自 Amazon RDS 主控台、Amazon RDS API、Amazon RDS AWS CLI 或開發套件的日誌，請將 log_output 參數設定為 FILE。如同 MariaDB 錯誤日誌，這些日誌檔案也會每小時輪換一次。前 24 小時之間產生的日誌檔案將會保留。

如需有關慢查詢與一般日誌的詳細資訊，請參閱 MariaDB 文件中的下列主題：

- [慢查詢日誌](#)
- [一般查詢日誌](#)

將 MariaDB 日誌發佈到 Amazon 日誌 CloudWatch

您可以將 MariaDB 資料庫執行個體設定為將日誌資料發佈到 Amazon CloudWatch 日誌中的日誌群組。使用 CloudWatch Logs，您可以對記錄資料執行即時分析，並用 CloudWatch 來建立警示和檢視指標。您可以使用 CloudWatch 日誌將日誌記錄存儲在高度耐用的存儲中。

Amazon RDS 將每個 MariaDB 資料庫日誌發佈為日誌群組中獨立的資料庫流。例如，假設您設定匯出函數來包含慢查詢日誌。然後，慢查詢資料會存放在 `/aws/rds/instance/my_instance/slowquery` 日誌群組的慢查詢日誌串流中。

錯誤日誌預設為啟用。下表摘要說明其他 MariaDB 日誌的要求。

日誌	要求
稽核日誌	資料庫執行個體必須使用含有選項 <code>MARIADB_AUDIT_PLUGIN</code> 的自訂選項群組。
一般日誌	資料庫執行個體必須使用含有參數設定 <code>general_log = 1</code> 的自訂參數群組，以啟用一般日誌。
慢查詢日誌	資料庫執行個體必須使用具有參數設定的自訂參數群組， <code>slow_query_log = 1</code> 或 <code>log_slow_query = 1</code> 啟用慢速查詢記錄。
記錄輸出	資料庫執行個體必須使用具有參數設定的自訂參數群組， <code>log_output = FILE</code> 才能將記錄寫入檔案系統，並將其發佈至 CloudWatch 記錄。

主控台

從控制台發佈 MariaDB 日 CloudWatch 誌到日誌

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。
4. 在 [記錄匯出] 區段中，選擇您要開始發佈至 CloudWatch 記錄的記錄檔。
5. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Instance (修改資料庫執行個體)。

AWS CLI

您可以發佈一個 MariaDB 的日誌與 AWS CLI 您可以使用下列參數來呼叫 [modify-db-instance](#) 命令：

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 選項的變更一律立即套用到資料庫執行個體。因此，`--apply-immediately` 和 `--no-apply-immediately` 選項沒有效果。

您也可以通過調用以下命令發布 MariaDB 日誌：AWS CLI

- [create-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

使用下列選項執行這些 AWS CLI 命令之一：

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

視您執行的 AWS CLI 命令而定，可能需要其他選項。

Example

下列範例會修改現有 MariaDB 資料庫執行個體，將記錄檔發佈至記錄檔。CloudWatch `--cloudwatch-logs-export-configuration` 值為 JSON 物件。此物件的金鑰為 `EnableLogTypes`，而其值為字串陣列，其中包含 `audit`、`error`、`general` 以及 `slowquery` 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":  
["audit","error","general","slowquery"]}'
```

在Windows中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

下列命令會建立 MariaDB 資料庫執行個體，並將記錄檔發佈至CloudWatch 記錄檔。--enable-cloudwatch-logs-exports 值為 JSON 字串陣列。字串可以是 audit、error、general 以及 slowquery 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \
  --db-instance-identifier mydbinstance \
  --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
  --db-instance-class db.m4.large \
  --engine mariadb
```

在Windows中：

```
aws rds create-db-instance ^
  --db-instance-identifier mydbinstance ^
  --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
  --db-instance-class db.m4.large ^
  --engine mariadb
```

RDS API

您可以使用 RDS API 來發佈 MariaDB 日誌。您可以搭配下列參數呼叫 [ModifyDBInstance](#) 操作：

- DBInstanceIdentifier
- CloudwatchLogsExportConfiguration

Note

CloudwatchLogsExportConfiguration 參數的變更一律立即套用到資料庫執行個體。因此，ApplyImmediately 參數沒有效果。

您也可以呼叫下列 RDS API 操作來發佈 MariaDB 日誌：

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

以下列參數來執行其中一個 RDS API 操作：

- DBInstanceIdentifier
- EnableCloudwatchLogsExports
- Engine
- DBInstanceClass

視您執行的 AWS CLI 命令而定，可能需要其他參數。

日誌檔案大小

MariaDB 慢查詢日誌、錯誤日誌以及一般日誌檔案大小限制為不超過資料庫執行個體配得之儲存空間的 2%。為保持此閾值，日誌會每小時輪換，而超過 24 小時的日誌檔案將會移除。若在移除舊日誌檔案後總日誌檔案大小仍超過閾值，將會從最大的日誌檔案開始刪除，直到日誌檔案大小不再超過閾值為止。

管理表格式 MariaDB 日誌

您可以將一般與慢查詢日誌導向至資料庫執行個體上的表格。若要這樣做，請建立資料庫參數群組，並將 log_output 伺服器參數設定為 TABLE。一般查詢會記錄於 mysql.general_log 表格，而慢查詢會記錄至 mysql.slow_log 表格。您可以查詢表格來存取日誌資訊。啟用此日誌記錄會增加寫入至資料庫的資料總數量，可能造成效能降低。

一般日誌和慢查詢日誌根據預設將會停用。若要啟用資料表的記錄功能，您也必須將下列伺服器參數設定為1：

- `general_log`
- `slow_query_log` 或 `log_slow_query`

日誌表格將持續擴增，直到重設適用參數為 0 時，各日誌記錄活動才會關閉。大量資料通常會隨時間累積，可能會佔用配得之儲存空間的大量比例。Amazon RDS 不允許您截斷記錄資料表，但是您可以移動其中的內容。輪換表格可將其內容儲存至備份表格，然後建立新的空白日誌表格。您可以透過以下命令列程序來手動輪換日誌表格，命令提示由 `PROMPT>` 表示：

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

為完全移除舊資料並重新取得磁碟空間，請連續呼叫兩次適用的程序。

二進位日誌格式

Amazon RDS 上的 MariaDB 支援基於列、基於陳述式和混合式二進位日誌記錄格式。預設二進位日誌格式為混合式。如需不同 MariaDB 二進位日誌格式的詳細資訊，請參閱 MariaDB 文件中的 [二進位日誌格式](#)。

如果您打算使用複寫，二進位記錄格式很重要。這是因為其決定資料變更的記錄，而此記錄會記錄在來源中並傳送到複寫目標。如需複寫時各種二進位日誌記錄格式的優缺點的相關資訊，請參閱 MySQL 文件中的 [基於陳述式和基於列的複寫的優缺點](#)。

Important

將二進位日誌格式設為行形式可能導致非常大的二進位日誌檔案。大型二進位日誌檔會減少資料庫執行個體可用的儲存空間數量。它們還會增加執行資料庫執行個體還原操作的時間量。基於陳述式的複寫可能會造成來源資料庫執行個體與僅供讀取複本不一致。如需詳細資訊，請參閱 MariaDB 文件中的 [在基於陳述式的複寫中不安全的陳述式](#)。

設定 MariaDB 二進位日誌格式

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。

2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇您要修改的資料庫執行個體所使用的參數群組。

您無法修改預設參數群組。如果資料庫執行個體使用預設參數群組，請建立新的參數群組，並將它與資料庫執行個體建立關聯。

如需資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。

4. 針對 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。
5. 將 binlog_format 參數設為您選擇的二進位日誌記錄格式 (ROW、STATEMENT 或 MIXED)。
6. 選擇 Save changes (儲存變更) 來儲存對資料庫參數群組的更新。

存取 MariaDB 二進位日誌

您可以使用 mysqlbinlog 公用程式，自 MariaDB 資料庫執行個體下載文字格式的二進位日誌。二進位日誌已下載到您的本機電腦。如需使用 mysqlbinlog 公用程式的詳細資訊，請前往 MariaDB 文件中的[使用 mysqlbinlog](#)。

若要在 Amazon RDS 執行個體上執行 mysqlbinlog 公用程式，請使用下列選項：

- 指定 --read-from-remote-server 選項。
- --host：指定來自執行個體端點的 DNS 名稱。
- --port：指定執行個體使用的連接埠。
- --user：指定已授予複寫從屬許可的 MariaDB 使用者。
- --password：指定使用者的密碼，或者省略密碼值，讓公用程式提示您密碼。
- --result-file：指定接收輸出的本機檔案。
- 指定一個或一個以上的二進位日誌檔案名稱。若要取得可用日誌清單，請使用 SQL 命令 SHOW BINARY LOGS。

如需 mysqlbinlog 選項的詳細資訊，請前往 MariaDB 文件中的[mysqlbinlog 選項](#)。

以下是範例：

對於LinuxmacOS、或Unix：

```
mysqlbinlog \  
  --read-from-remote-server \  
  --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \  
  --result-file=/tmp/binlog.txt
```

```
--port=3306 \  
--user ReplUser \  
--password <password> \  
--result-file=/tmp/binlog.txt
```

在Windows中：

```
mysqlbinlog ^  
--read-from-remote-server ^  
--host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^  
--port=3306 ^  
--user ReplUser ^  
--password <password> ^  
--result-file=/tmp/binlog.txt
```

Amazon RDS 通常會儘快清除二進位日誌。不過，執行個體上仍須有可用的二進位日誌，以供 `mysqlbinlog` 存取。若要指定 RDS 保留二進位日誌的時數，請使用 `mysql.rds_set_configuration` 預存程序。指定具有足夠時間供您下載日誌的期間。設定保留期間之後，請監控資料庫執行個體的儲存體用量，確定保留的二進位日誌沒有佔用太多儲存空間。

下列範例將保留期間設定為 1 天。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

若要顯示目前設定，請使用 `mysql.rds_show_configuration` 預存程序。

```
call mysql.rds_show_configuration;
```

二進位日誌註釋

在 MariaDB 資料庫執行個體中，您可以使用會導致行事件之含有 SQL 查詢副本的 `Annotate_rows` 事件來對行事件做註釋。此方法提供類似的功能，可啟用 RDS for MySQL 資料庫執行個體上的 `binlog_rows_query_log_events` 參數。

您可以建立自訂參數群組，並將 `binlog_annotate_row_events` 參數設為 **1**，以全域啟用二進位日誌註釋。您也可以透過呼叫 `SET SESSION binlog_annotate_row_events = 1`，於工作階段層級啟用註釋。若已啟用二進位日誌，請使用 `replicate_annotate_row_events` 將二進位日誌註釋複寫至複本執行個體。不需要特殊權限即可使用這些設定。

下列為 MariaDB 中行形式交易的範例。將交易隔離等級設定為專供讀取時，將觸發使用行形式日誌。

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
```

若沒有註釋，交易的二進位日誌項目將如下所示：

```
BEGIN
/*!*/;
# at 1163
# at 1209
#150922 7:55:57 server id 1855786460 end_log_pos 1209          Table_map:
`test`.`square` mapped to number 76
#150922 7:55:57 server id 1855786460 end_log_pos 1247          Write_rows: table id 76
flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 1247
#150922 7:56:01 server id 1855786460 end_log_pos 1274          Xid = 62
COMMIT/*!*/;
```

下列陳述式為此相同交易啟用工作階段等級的註釋，並在完成交易後停用：

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotate_row_events = 1;
BEGIN;
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotate_row_events = 0;
```

若有註釋，交易的二進位日誌項目將如下所示：

```
BEGIN
/*!*/;
```

```
# at 423
# at 483
# at 529
#150922 8:04:24 server id 1855786460 end_log_pos 483 Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
#150922 8:04:24 server id 1855786460 end_log_pos 529 Table_map: `test`.`square`
mapped to number 76
#150922 8:04:24 server id 1855786460 end_log_pos 567 Write_rows: table id 76 flags:
STMT_END_F
### INSERT INTO `test`.`square`
### SET
### @1=5
### @2=25
# at 567
#150922 8:04:26 server id 1855786460 end_log_pos 594 Xid = 88
COMMIT/*!*/;
```


Microsoft SQL Server 資料庫日誌檔案

您可使用 Amazon RDS 主控台、AWS CLI 或 RDS API 存取 Microsoft SQL Server 錯誤記錄、代理程式記錄、追蹤檔案和傾印檔案。如需關於檢視、下載與查看資料庫日誌檔案的資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。

主題

- [保留排程](#)
- [使用 rds_read_error_log 程序檢視 SQL Server 錯誤](#)
- [將 SQL 伺服器日誌發佈到 Amazon CloudWatch 日誌](#)

保留排程

不論您的資料庫執行個體何時重新啟動，日誌檔案都會每日輪換。下列為 Amazon RDS 上的 Microsoft SQL Server 日誌保留排程。

日誌類型	保留排程
錯誤日誌	最多會保留 30 個錯誤記錄。Amazon RDS 可能會刪除超過 7 日的錯誤記錄。
代理程式日誌	最多會保留 10 個代理程式記錄。Amazon RDS 可能會刪除超過 7 日的代理程式記錄。
追蹤檔案	追蹤檔案會依據您資料庫執行個體的追蹤檔案保留期間進行保存。預設的追蹤檔案保留期間為 7 天。若要修改您資料庫執行個體的追蹤檔案保留期間，請參閱 設定追蹤和傾印檔案的保留期間 。
傾印檔案	傾印檔案會依據您資料庫執行個體的傾印檔案保留期間進行保存。預設的傾印檔案保留期間為 7 天。若要修改您資料庫執行個體的傾印檔案保留期間，請參閱 設定追蹤和傾印檔案的保留期間 。

使用 rds_read_error_log 程序檢視 SQL Server 錯誤

您可以使用 Amazon RDS 預存程序 rds_read_error_log 來檢視錯誤日誌與代理程式日誌。如需詳細資訊，請參閱[檢視錯誤和代理程式日誌](#)。

將 SQL 伺服器日誌發佈到 Amazon CloudWatch 日誌

使用適用 Amazon RDS for SQL Server，您可以將錯誤和代理程式記錄事件直接發佈到 Amazon CloudWatch 日誌。使用記錄檔分析記 CloudWatch 錄資料，然後用 CloudWatch 來建立警示和檢視指標。

使用 CloudWatch 日誌，您可以執行以下操作：

- 將日誌存放在高耐用性儲存空間，並由您定義保留期間。
- 搜尋和篩選日誌資料。
- 在帳戶之間共享日誌資料。
- 將日誌匯出至 Amazon S3。
- 將數據流傳輸到 Amazon OpenSearch 服務。
- 使用 Amazon Kinesis Data Streams 即時處理日誌資料。如需詳細資訊，請參閱[適用於 SQL 應用程式的 Apache Flink 受管服務中使用 Amazon CloudWatch 日誌](#)開發人員指南。

Amazon RDS 將每個 SQL Server 資料庫日誌發佈為日誌群組中獨立的資料庫流。例如，如果您發佈代理程式記錄檔和錯誤記錄檔，錯誤資料會儲存在記錄群組的錯誤記aws/rds/instance/my_instance/error錄資料流中，而代理程式記錄檔資料會儲存在記aws/rds/instance/my_instance/agent錄群組中。

對於多可用區域資料庫執行個體，Amazon RDS 會在日誌群組中以兩個獨立的串流形式發佈資料庫日誌。例如，如果您發佈錯誤日誌，錯誤資料會分別存放在錯誤日誌串流aws/rds/instance/my_instance.node1/error和aws/rds/instance/my_instance.node2/error中。日誌串流在容錯移轉期間不會變更，而且每個節點的錯誤日誌串流可能包含來自主要或次要執行個體的錯誤日誌。使用異地同步備份時，系統會自動建立日誌串流，aws/rds/instance/my_instance/rds-events以儲存事件資料，例如資料庫執行個體容錯移轉。

Note

根據預設，不會啟用將 SQL Server CloudWatch 記錄檔發佈到記錄檔。不支援發佈追蹤檔案和傾印檔案。除了亞太區域 (香港) 以外，所有區域都支援將 SQL Server CloudWatch 記錄檔發佈至記錄檔。

主控台

若要將 SQL 伺服器資料庫記錄發佈至 CloudWatch 記錄，請從 AWS Management Console

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。
4. 在 [記錄匯出] 區段中，選擇您要開始發佈至 CloudWatch 記錄檔的記錄檔。

您可以選擇 Agent log (代理程式日誌)、Error log (錯誤日誌)，或兩者皆選。

5. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Instance (修改資料庫執行個體)。

AWS CLI

若要發佈 SQL Server 日誌，您可搭配下列參數使用 [modify-db-instance](#) 命令：

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 選項的變更一律立即套用到資料庫執行個體。因此，`--apply-immediately` 和 `--no-apply-immediately` 選項沒有效果。

您亦可使用下列命令來發佈 SQL Server 日誌：

- [create-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-to-point-in-time](#)

Example

下列範例會建立啟用 CloudWatch 記錄發佈的 SQL Server 資料庫執行個體。`--enable-cloudwatch-logs-exports` 值是 JSON 字串陣列，可包含 `error`、`agent` 或兩者。

對於 Linux/macOS、或 Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --enable-cloudwatch-logs-exports ["error","agent"] \  
  --db-instance-class db.m4.large \  
  --engine sqlserver-se
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --enable-cloudwatch-logs-exports ["error","agent"] ^  
  --db-instance-class db.m4.large ^  
  --engine sqlserver-se
```

Note

使用 Windows 命令提示字元時，您必須在 JSON 程式碼中的雙引號 (") 開頭加上反斜線 (\)，以逸出雙引號。

Example

下列範例會修改現有的 SQL Server 資料庫執行個體，以將記錄檔發佈至 CloudWatch 記錄檔。--cloudwatch-logs-export-configuration 值為 JSON 物件。此物件的金鑰為 EnableLogTypes，而值為字串陣列，可包含 error、agent 或兩者。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

Note

使用 Windows 命令提示字元時，您必須在 JSON 程式碼中的雙引號 (") 開頭加上反斜線 (\)，以逸出雙引號。

Example

下列範例會修改現有的 SQL Server 資料庫執行個體，以停用將代理程式記錄檔發佈至 CloudWatch 記錄檔。--cloudwatch-logs-export-configuration 值為 JSON 物件。此物件的金鑰為 DisableLogTypes，而值為字串陣列，可包含 error、agent 或兩者。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["agent"]}'
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"agent\"]}"
```

Note

使用 Windows 命令提示字元時，您必須在 JSON 程式碼中的雙引號 (") 開頭加上反斜線 (\)，以逸出雙引號。

MySQL 資料庫日誌檔案

您可以直接透過 Amazon RDS 主控台、Amazon RDS API、AWS CLI 或 AWS SDK 來監控 MySQL 日誌。您可以將日誌指向主要資料庫中的資料庫表並查詢該表格，藉此存取 MySQL 日誌。您可以使用 mysqlbinlog 公用程式來下載二進位日誌。

如需關於檢視、下載與查看資料庫日誌檔案的資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。

主題

- [RDS for MySQL 資料庫日誌概觀](#)
- [將 MySQL 日誌發佈到 Amazon CloudWatch 日誌](#)
- [管理以資料表為基礎的 MySQL 日誌](#)
- [設定適用於 MySQL RDS](#)
- [存取 MySQL 二進位日誌](#)

RDS for MySQL 資料庫日誌概觀

您可以監控下列類型的 RDS for MySQL 日誌檔案：

- 錯誤日誌
- 慢查詢日誌
- 一般日誌
- 稽核日誌

預設情況下會產生 RDS for MySQL 錯誤日誌。透過在資料庫參數群組中設定參數，產生慢查詢日誌和一般日誌。

主題

- [RDS for MySQL 錯誤日誌](#)
- [RDS for MySQL 慢查詢與一般查詢](#)
- [MySQL 稽核日誌](#)
- [MySQL RDS 的日誌輪換和保留](#)
- [重做日誌的大小限制](#)

RDS for MySQL 錯誤日誌

RDS for MySQL 會將錯誤寫入 `mysql-error.log` 檔案中。每個日誌檔案的產生時間 (UTC 時區) 皆會附加於檔案名稱中。日誌檔案也有時間戳記，可協助您判定日誌項目寫入的時間。

只有在開機、當機以及發生錯誤時，RDS for MySQL 才會寫入錯誤日誌。資料庫執行個體可在未寫入新項目到錯誤日誌的情況下持續執行數小時或數日。若您沒有看到最近的項目，這是因為伺服器未遇到需寫入日誌項目的錯誤。

根據設計，系統會篩選錯誤日誌，以僅顯示未預期的事件，例如錯誤。不過，錯誤日誌還包含一些其他未顯示的資料庫資訊，例如查詢進度。因此即使沒有任何實際錯誤，但錯誤日誌的大小可能會因為在進行的資料庫活動而增加。雖然您可能會在 AWS Management Console 看到特定大小 (以位元組或 KB 為單位) 的錯誤日誌，但下載時它們可能是 0 個位元組。

RDS for MySQL 每 5 分鐘將 `mysql-error.log` 寫入磁碟一次。這會將日誌的內容追加到 `mysql-error-running.log`。

RDS for MySQL 每小時會輪換 `mysql-error-running.log` 檔案。它會保留最近兩週產生的日誌。

Note

Amazon RDS 和 Aurora 的日誌保留期間不同。

RDS for MySQL 慢查詢與一般查詢

您可以將 RDS for MySQL 慢查詢日誌與一般日誌寫入至檔案或資料庫表格。若要這樣做，請在您的資料庫參數群組中設定參數。如需建立和修改資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。您必須先設定這些參數，才可在 Amazon RDS 主控台中檢視慢查詢記錄或一般記錄，或者使用 Amazon RDS API、Amazon RDS CLI 或 AWS SDK 檢視。

您可以使用清單中的參數來控制 RDS for MySQL 日誌記錄：

- `slow_query_log`：若要建立慢查詢，請設為 1。預設為 0。
- `general_log`：若要建立一般日誌，請設為 1。預設值為 0。
- `long_query_time`：若要避免快速執行查詢記錄於慢查詢日誌中，請為需記錄之最短查詢執行時間指定一個值，以秒為單位。預設為 10 秒，最短時間為 0。若 `log_output = FILE`，您可以指定以毫秒解析度為單位的浮點值。若 `log_output = TABLE`，您必須指定以秒為單位的整數值。只會記錄執行時間超過 `long_query_time` 值的查詢。例如，將 `long_query_time` 設為 0.1 可避免記錄任何在 100 毫秒內執行之查詢。

- `log_queries_not_using_indexes`：若要將所有不使用索引的查詢記錄於慢查詢日誌中，請設為 1。系統會記錄不使用索引的查詢，即使其執行時間低於 `long_query_time` 參數的值。預設值為 0。
- `log_output` *option*：您可為 `log_output` 參數指定下列其中一個選項。
 - TABLE (預設) – 將一般查詢寫入 `mysql.general_log` 表格，而慢查詢則寫入 `mysql.slow_log` 表格。
 - FILE – 同時將一般與慢查詢日誌寫入檔案系統中。
 - NONE – 停用日誌記錄。

如需慢查詢與一般日誌的詳細資訊，請參閱 MySQL 文件中的下列主題：

- [慢查詢日誌](#)
- [一般查詢日誌](#)

MySQL 稽核日誌

若要存取稽核記錄，資料庫執行個體必須使用含有選項 `MARIADB_AUDIT_PLUGIN` 的自訂選項群組。如需詳細資訊，請參閱[MariaDB 稽核外掛程式支援 MySQL](#)。

MySQL RDS 的日誌輪換和保留

記錄啟用時，Amazon RDS 輪換表格日誌或以規律間隔來刪除日誌檔案。此方法為預防措施，可降低大型日誌檔封鎖資料庫使用或影響效能的可能性。RDS for MySQL 會處理輪流與刪除，如下所示：

- MySQL 慢查詢日誌、錯誤日誌以及一般日誌檔案大小限制為不超過資料庫執行個體所配置之儲存空間的 2%。為保持此閾值，日誌會每小時自動輪換。MySQL 會移除兩週以上的日誌檔。若在移除舊日誌檔案後總日誌檔案大小仍超過閾值，將會從最舊的日誌檔案開始刪除，直到日誌檔案大小不再超過閾值為止。
- 啟用 FILE 日誌記錄時，系統會每小時檢查日誌檔案，並刪除超過兩週的日誌檔案。在部分情況下，刪除後剩餘的總日誌檔大小可能超過資料庫執行個體配得之空間的 2% 閾值。於這些狀況中，最舊的日誌檔將遭刪除，直到日誌檔大小不再超過閾值。
- 當 TABLE 日誌記錄啟用時，在某些情況下，日誌資料表每 24 小時會輪換一次。在表格日誌使用的空間超過所配置儲存空間的 20% 時，就會發生輪換。如果所有日誌合併後的大小超過 10 GB 時，也會發生這種情況。若資料庫執行個體使用的空間總量高於資料庫執行個體配得之儲存空間的 90% 時，日誌輪換的閾值將會降低。在表格日誌使用的空間超過所配置儲存空間的 10% 時，就會輪換日誌表格。如果所有日誌合併後的大小超過 5 GB 時，也會轉換它們。您可訂閱 `low_free_storage`

事件，日誌表格輪換以釋放空間時就會通知您。如需詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。

輪換日誌表格時，目前日誌表格首先會複製到備份日誌表格。然後移除目前日誌表格中的項目。如果備份日誌資料表已存在，則其會在目前日誌資料表複製到備份之前刪除。如有需要，您可以查詢備份日誌資料表。mysql.general_log 資料表的備份日誌資料表名為 mysql.general_log_backup。mysql.slow_log 資料表的備份日誌資料表名為 mysql.slow_log_backup。

您可以呼叫 mysql.general_log 程序來輪換 mysql.rds_rotate_general_log 表格。您可以呼叫 mysql.slow_log 程序來輪換 mysql.rds_rotate_slow_log 表格。

表格日誌會在資料庫版本升級時輪換。

若要從 Amazon RDS 主控台、Amazon RDS API、Amazon RDS CLI 或 AWS SDK 使用記錄，請將 log_output 參數設為 FILE。如同 MySQL 錯誤日誌，這些日誌檔案也會每小時輪換。系統會保留在過去兩週內產生的日誌檔案。請注意，Amazon RDS 和 Aurora 的保留期間不同。

重做日誌的大小限制

對於 MySQL 版本 8.0.32 及更低版本的 RDS，此參數的預設值為 256 MB。此數量是藉由參數的預設值 (128 MB) 乘以 innodb_log_file_size 參數的預設值 (2) 而得出。innodb_log_files_in_group 如需詳細資訊，請參閱[設定適用於 Amazon RDS for MySQL 參數的最佳實務，第 1 部分：與效能相關的參數](#)。

從適用於 MySQL 的 RDS 版本 8.0.33 開始，Amazon RDS 使用該 innodb_redo_log_capacity 參數而不是參數。innodb_log_file_size 該參數的 Amazon RDS 默認 innodb_redo_log_capacity 值是 2 GB。如需詳細資訊，請參閱 MySQL 文件中的 [MySQL 8.0.30 中的變更](#)。

將 MySQL 日誌發佈到 Amazon CloudWatch 日誌

您可以將 MySQL 資料庫執行個體設定為將日誌資料發佈到 Amazon CloudWatch 日誌中的日誌群組。使用 CloudWatch Logs，您可以對記錄資料執行即時分析，並用 CloudWatch 來建立警示和檢視指標。您可以使用 CloudWatch 日誌將日誌記錄存儲在高度耐用的存儲中。

Amazon RDS 將每個 MySQL 資料庫日誌發佈為日誌群組中獨立的資料庫流。例如，若您設定匯出功能以包含慢查詢日誌，則慢查詢資料會存放在 /aws/rds/instance/*my_instance*/slowquery 日誌群組內的慢查詢日誌串流中。

錯誤日誌預設為啟用。下表摘要說明其他 MySQL 日誌的要求。

日誌	要求
稽核日誌	資料庫執行個體必須使用含有選項 MARIADB_AUDIT_PLUGIN 的自訂選項群組。
一般日誌	資料庫執行個體必須使用含有參數設定 <code>general_log = 1</code> 的自訂參數群組，以啟用一般日誌。
慢查詢日誌	資料庫執行個體必須使用含有參數設定 <code>slow_query_log = 1</code> 的自訂參數群組，以啟用慢查詢日誌。
記錄輸出	資料庫執行個體必須使用具有參數設定的自訂參數群組， <code>log_output = FILE</code> 才能將記錄寫入檔案系統，並將其發佈至 CloudWatch 記錄。

主控台

若要使用主控台將 MySQL 記 CloudWatch 錄檔發佈至記錄

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。
4. 在 [記錄匯出] 區段中，選擇您要開始發佈至 CloudWatch 記錄檔的記錄檔。
5. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Instance (修改資料庫執行個體)。

AWS CLI

您可以使用 AWS CLI 來發佈 MySQL 日誌。您可以使用下列參數來呼叫 [modify-db-instance](#) 命令：

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

--cloudwatch-logs-export-configuration 選項的變更一律立即套用到資料庫執行個體。因此，--apply-immediately 和 --no-apply-immediately 選項沒有效果。

您也可以呼叫下列 AWS CLI 命令來發佈 MySQL 日誌：

- [create-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

執行上述其中一個 AWS CLI 命令並指定下列選項：

- --db-instance-identifier
- --enable-cloudwatch-logs-exports
- --db-instance-class
- --engine

視您執行的 AWS CLI 命令而定，可能需要其他選項。

Example

下列範例會修改現有的 MySQL 資料庫執行個體，以將記錄檔發佈至 CloudWatch 記錄檔。--cloudwatch-logs-export-configuration 值為 JSON 物件。此物件的金鑰為 EnableLogTypes，而其值為字串陣列，其中包含 audit、error、general 以及 slowquery 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":  
["audit","error","general","slowquery"]}'
```

在Windows中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

下列範例會建立 MySQL 資料庫執行個體，並將記錄檔發佈至 CloudWatch 記錄檔。--enable-cloudwatch-logs-exports 值為 JSON 字串陣列。字串可以是 audit、error、general 以及 slowquery 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \
  --db-instance-identifier mydbinstance \
  --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
  --db-instance-class db.m4.large \
  --engine MySQL
```

在Windows中：

```
aws rds create-db-instance ^
  --db-instance-identifier mydbinstance ^
  --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
  --db-instance-class db.m4.large ^
  --engine MySQL
```

RDS API

您可以使用 RDS API 來發佈 MySQL 日誌。您可以使用下列參數來呼叫 [ModifyDBInstance](#) 動作：

- DBInstanceIdentifier
- CloudwatchLogsExportConfiguration

Note

CloudwatchLogsExportConfiguration 參數的變更一律立即套用到資料庫執行個體。因此，ApplyImmediately 參數沒有效果。

您也可以呼叫下列 RDS API 操作來發佈 MySQL 日誌：

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

以下列參數來執行其中一個 RDS API 操作：

- DBInstanceIdentifier
- EnableCloudwatchLogsExports
- Engine
- DBInstanceClass

視您執行的 AWS CLI 命令而定，可能需要其他參數。

管理以資料表為基礎的 MySQL 日誌

您可以建立資料庫參數群組，並將 log_output 伺服器參數設為 TABLE，將一般與慢查詢日誌導向資料庫執行個體上的表格。一般查詢會記錄於 mysql.general_log 表格，而慢查詢會記錄至 mysql.slow_log 表格。您可以查詢表格來存取日誌資訊。啟用此日誌記錄會增加寫入至資料庫的資料總數量，可能造成效能降低。

一般日誌和慢查詢日誌根據預設將會停用。為啟用日誌記錄至表格，您也必須將 general_log 與 slow_query_log 伺服器參數設為 1。

日誌表格將持續擴增，直到重設適用參數為 0 時，各日誌記錄活動才會關閉。大量資料通常會隨時間累積，可能會佔用配得之儲存空間的大量比例。Amazon RDS 不允許截斷記錄資料表，但是您可以移動其中的內容。輪換表格可將其內容儲存至備份表格，然後建立新的空白日誌表格。您可以透過以下命令列程序來手動輪換日誌表格，命令提示由 PROMPT> 表示：

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

為完全移除舊資料並重新取得磁碟空間，請連續呼叫兩次適用的程序。

設定適用於 MySQL RDS

二進位日誌是一組日誌檔案，其中包含對 MySQL 伺服器執行個體所做資料修正的相關資訊。二進位日誌包含的資訊如下：

- 描述資料庫變更的事件 (如建立資料表或修改資料列)
- 有關更新了資料的每個陳述式持續時間的資訊
- 本應更新資料但未更新的陳述式事件

二進位日誌會記錄複寫過程中傳送的陳述式。某些復原操作也需要這些日誌。如需詳細資訊，請參閱 MySQL 說明文件中的 [二進位日誌](#) 和 [二進位日誌概觀](#)。

自動備份功能會決定 MySQL 開啟還是關閉二進位日誌記錄功能。您有下列選項：

開啟二進位日誌記錄功能

將備份保留期設定為正的非零值。

關閉二進位日誌記錄功能

將備份保留期間設定為 0。

如需詳細資訊，請參閱 [啟用自動備份](#)。

Amazon RDS 上的 MySQL 支援 row-based (列型)、statement-based (陳述式型)，和 mixed (混和式) 二進位記錄格式。除非需要特定的 binlog 格式，否則我們建議混合使用。如需各種 MySQL 二進位日誌格式的詳細資訊，請參閱 MySQL 說明文件中的 [二進位日誌記錄格式](#)。

如果您打算使用複寫，二進位日誌記錄格式很重要，因為這決定資料變更的記錄，而此記錄會記錄在來源中並傳送到複寫目標。如需複寫時各種二進位日誌記錄格式的優缺點的相關資訊，請參閱 MySQL 文件中的 [基於陳述式和基於列的複寫的優缺點](#)。

⚠ Important

將二進位日誌格式設為行形式可能導致非常大的二進位日誌檔案。大型二進位日誌檔案會減少資料庫執行個體可用的儲存空間數量，並會增加執行資料庫執行個體還原操作的總時間。基於陳述式的複寫可能會造成來源資料庫執行個體與僅供讀取複本不一致。如需詳細資訊，請參閱 MySQL 文件中的 [二進位日誌記錄中安全和不安全陳述式的判定](#)。

啟用二進位記錄會增加資料庫執行個體的寫入磁碟 I/O 操作次數。您可以使用 WriteIOPS CloudWatch 量度監視 IOPS 使用情況。

設定 MySQL 二進位記錄格式

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 您無法修改預設參數群組。如果資料庫執行個體使用預設參數群組，請建立新的參數群組，並將其與資料庫執行個體建立關聯。

如需參數群組的詳細資訊，請參閱 [使用參數群組](#)。

4. 從「動作」中選擇「編輯」。
5. 將 `binlog_format` 參數設為您選擇的二進位日誌記錄格式 (ROW、STATEMENT 或 MIXED)。

您可以將資料庫執行個體的備份保留期間設定為零來關閉二進位記錄，但這會停用每日自動備份。停用自動備份會關閉或停用 `log_bin` 工作階段變數。這會停用 RDS for MySQL 資料庫執行個體上的二進位記錄，進而將 `binlog_format` 工作階段變數重設為資料庫 ROW 中的預設值。我們建議您不要停用備份。如需備份保留期間設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

6. 選擇 Save changes (儲存變更) 來儲存對資料庫參數群組的更新。

由於 `binlog_format` 參數在 RDS for MySQL 中是動態的，因此您不需要重新啟動資料庫執行個體即可套用變更。(請注意，在 Aurora MySQL 中，此參數是靜態的。如需詳細資訊，請參閱 [設定 Aurora MySQL 二進位記錄](#)。)

⚠ Important

變更資料庫參數群組會影響使用該參數群組的所有資料庫執行個體。如果您想要為 AWS 區域中的不同 MySQL 資料庫執行個體指定不同的二進位記錄格式，資料庫執行個體必須使用不同

的資料庫參數群組。這些參數群組會識別不同的記錄格式。將適當的資料庫參數群組指派給每個資料庫執行個體。

存取 MySQL 二進位日誌

您可使用 `mysqlbinlog` 公用程式，從 RDS for MySQL 資料庫執行個體下載或串流二進位日誌。二進位日誌會下載至您的本機電腦，您可於此執行動作，例如使用 `mysql` 公用程式來重新執行日誌。如需使用 `mysqlbinlog` 公用程式的詳細資訊，請參閱 MySQL 文件中的[使用 `mysqlbinlog` 備份二進位日誌檔案](#)。

若要在 Amazon RDS 執行個體上執行 `mysqlbinlog` 公用程式，請使用下列選項：

- `--read-from-remote-server` - 必要項目。
- `--host` - 來自執行個體端點的 DNS 名稱。
- `--port` - 執行個體使用的連接埠。
- `--user` - 已授予 REPLICATION SLAVE 許可的 MySQL 使用者。
- `--password` - MySQL 使用者的密碼，或者省略密碼值，讓公用程式提示您密碼。
- `--raw` - 以二進位格式下載檔案。
- `--result-file` - 接收列輸出的本機檔案。
- `--stop-never` - 串流二進位日誌檔。
- `--verbose` - 當您使用 ROW binlog 格式時，請加入此選項，將資料列事件視為虛擬 SQL 陳述式。如需 `--verbose` 選項的詳細資訊，請參閱 MySQL 文件中的[mysqlbinlog row event display](#)。
- 指定一個或一個以上的二進位日誌檔案名稱。若要取得可用日誌清單，請使用 SQL 命令 `SHOW BINARY LOGS`。

如需 `mysqlbinlog` 選項的詳細資訊，請參閱 MySQL 文件中的[mysqlbinlog — 處理二進位日誌檔案的公用程式](#)。

下列範例顯示如何使用 `mysqlbinlog` 公用程式。

對於LinuxmacOS、或Unix：

```
mysqlbinlog \  
  --read-from-remote-server \  
  --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com \  
  --port=3306 \  
  --result-file=binlog.txt
```



```
--user ReplUser \  
--password \  
--raw \  
--verbose \  
--result-file=/tmp/ \  
binlog.00098
```

在Windows中：

```
mysqlbinlog ^  
--read-from-remote-server ^  
--host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com ^  
--port=3306 ^  
--user ReplUser ^  
--password ^  
--raw ^  
--verbose ^  
--result-file=/tmp/ ^  
binlog.00098
```

Amazon RDS 通常會儘快清除二進位日誌，但執行個體上必須有可由 `mysqlbinlog` 存取的二進位日誌。若要指定保留二進位日誌的 RDS 時數，請使用 [mysql.rds_set_configuration](#) 預存程序，並指定讓您有足夠時間下載日誌的期間，如下列範例所示。設定保留期間之後，請監控資料庫執行個體的儲存體用量，確定保留的二進位日誌沒有佔用太多儲存空間。

下列範例將保留期間設定為 1 天。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

若要顯示目前設定，請使用 [mysql.rds_show_configuration](#) 預存程序。

```
call mysql.rds_show_configuration;
```

Oracle 資料庫日誌檔案

您可使用 Amazon RDS 主控台或 API 存取 Oracle 提醒日誌、稽核檔案及追蹤檔案。如需關於檢視、下載與查看資料庫日誌檔案的資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。

所提供的 Oracle 稽核檔案為標準 Oracle 稽核檔案。Amazon RDS 可支援 Oracle 精細稽核 (FGA) 功能。然而，存放於 SYS.FGA_LOG\$ 資料表且可由 DBA_FGA_AUDIT_TRAIL 檢視的 FGA 事件，將無法由日誌存取。

[DescribeDBLogFiles](#) API 操作會列出可供資料庫執行個體使用的 Oracle 日誌檔案，並忽略 MaxRecords 參數，且最多回傳 1,000 筆記錄。呼叫傳回 LastWritten 作為 POSIX 日期，以毫秒為單位。

主題

- [保留排程](#)
- [使用 Oracle 追蹤檔案](#)
- [將 Oracle 日誌發佈到 Amazon CloudWatch 日誌](#)
- [存取警示日誌和監聽器日誌](#)

保留排程

若日誌檔案變得過大，Oracle 資料庫引擎可能會輪換這些檔案。欲保留稽核或追蹤檔案，請下載這些檔案。如果您將檔案儲存在本機，可降低 Amazon RDS 的儲存成本，並為資料騰出更多可用空間。

下列資料表顯示 Amazon RDS 上的 Oracle 提醒日誌、稽核檔案及追蹤檔案的保留排程。

日誌類型	保留排程
提醒日誌	文字提醒記錄每日輪換，其 30 天的保留則由 Amazon RDS 管理。XML 提醒日誌至少會保留七天。您可使用 ALERTLOG 畫面來存取此記錄。
稽核檔案	預設稽核檔案的保留時間為 7 日。Amazon RDS 可能會刪除超過 7 日的稽核檔案。
追蹤檔案	預設的追蹤檔案保留時間為 7 日。Amazon RDS 可能會刪除超過 7 日的追蹤檔案。

日誌類型	保留排程
接聽程式日誌	預設接聽程式記錄的保留時間為 7 日。Amazon RDS 可能會刪除超過 7 日的接聽程式記錄。

Note

稽核檔案和追蹤檔案共享相同的保留組態。

使用 Oracle 追蹤檔案

您可於下方取得 Amazon RDS 建立、重新整理、存取和刪除追蹤檔案的說明。

主題

- [列出檔案](#)
- [產生追蹤檔案並追蹤工作階段](#)
- [擷取追蹤檔案](#)
- [清除追蹤檔案](#)

列出檔案

您可使用這兩個程序之一來存取 background_dump_dest 路徑上的檔案。第一個程序會重新整理畫面，其中內含 background_dump_dest 中現有的所有檔案。

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

重新整理畫面後，請查詢下列畫面來存取結果。

```
SELECT * FROM rdsadmin.tracefile_listing;
```

上述程序的替代做法是使用 FROM table，採用類似資料表的格式來串流非關聯式資料，以列出資料庫目錄內容。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

下列查詢顯示日誌檔案的文字。

```
SELECT text FROM
TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_<dbname>.log.<date>'));
```

在僅供讀取複本上，查詢 V\$DATABASE.DB_UNIQUE_NAME 以取得 BDUMP 目錄名稱。如果唯一名稱為 DATABASE_B，則 BDUMP 目錄為 BDUMP_B。下列範例會在複本上查詢 BDUMP 名稱，然後使用此名稱查詢 alert_DATABASE.log.2020-06-23 的內容。

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*(_[A-Z])', '\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
-----
BDUMP_B

SELECT TEXT FROM
table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

產生追蹤檔案並追蹤工作階段

由於 ALTER SESSION 不具任何限制，許多在 Oracle 中產生追蹤檔案的標準方法仍可供 Amazon RDS 資料庫執行個體使用。需要更多存取權限的追蹤檔案，可參考下列程序。

Oracle 方法	Amazon RDS 方法
oradebug hanganalyze 3	EXEC rdsadmin.manage_tracefiles.hanganalyze;
oradebug dump systemstate 266	EXEC rdsadmin.manage_tracefiles.dump_systemstate;

您可使用許多標準方法來追蹤連接至 Amazon RDS 中 Oracle 資料庫執行個體的個別工作階段。若要啟用工作階段的追蹤功能，您可以在 Oracle 提供的 PL/SQL 套件中執行子程式，例如 DBMS_SESSION 和 DBMS_MONITOR。如需詳細資訊，請參閱 Oracle 文件中的 [為工作階段啟用追蹤](#)。

擷取追蹤檔案

您可使用 Amazon RDS 管理之外部資料表上的標準 SQL 查詢，擷取 background_dump_dest 中的所有追蹤檔案。欲使用此方法，務必執行此程序，為此資料表設定特定追蹤檔案的位置。

例如，您可以使用上述的 `rdsadmin.tracefile_listing` 畫面，之後即可列出系統上的所有追蹤檔案。接著，請使用下列程序，將 `tracefile_table` 畫面設定為指向特定追蹤檔案。

```
EXEC
rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

下列範例會在目前的結構描述內建立外部資料表，其位置則設定為所提供的文件。您可使用 SQL 查詢將內容擷取至本機檔案。

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
SPOOL OFF;
```

清除追蹤檔案

追蹤檔案可能會累積和耗用磁碟空間。依預設，Amazon RDS 會清除超過 7 日的追蹤檔案和記錄檔案。您可使用 `show_configuration` 程序來檢視並設定追蹤檔案保留期間。您應執行命令 `SET SERVEROUTPUT ON` 以檢視組態結果。

下列範例顯示目前的追蹤檔案保留期間，然後設定為新的追蹤檔案保留期間。

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
  bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);
SQL> commit;

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
  bdump are automatically deleted.
```

除了定期的清除程序，您也可透過 `background_dump_dest` 手動移除檔案。下列範例說明如何清除超過五分鐘的所有檔案。

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

您亦可清除符合特定模式的所有檔案 (如果這樣做，請勿包含副檔名，例如 .trc)。下列範例說明如何清除以 SCHPOC1_ora_5935 為開頭的所有檔案。

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

將 Oracle 日誌發佈到 Amazon CloudWatch 日誌

您可以將 RDS for Oracle 資料庫執行個體設定為將日誌資料發佈到 Amazon CloudWatch 日誌中的日誌群組。使用 CloudWatch 日誌，您可以分析日誌數據，並用 CloudWatch 於創建警報和查看指標。您可以使用 CloudWatch 日誌將日誌記錄存儲在高度耐用的存儲中。

Amazon RDS 將每個 Oracle 資料庫日誌發佈為日誌群組中獨立的資料庫流。例如，若您將匯出功能設定為包含稽核日誌，則稽核資料會存放在 /aws/rds/instance/my_instance/audit 日誌群組內的稽核日誌流中。下表總結了適用於 Oracle 的 RDS 將日誌發佈到 Amazon CloudWatch 日誌的需求。

日誌名稱	需求	預設
警示日誌	無。您無法停用此記錄檔。	已啟用
追蹤日誌	將 <code>trace_enabled</code> 參數設定為 TRUE 或保持其設定為預設值。	TRUE
稽核日誌	將 <code>audit_trail</code> 參數設定為下列任一允許的值： <pre>{ none os db [, extended] xml [, extended] }</pre>	none
接聽程式日誌	無。您無法停用此記錄檔。	已啟用
Oracle Management Agent 日誌	無。您無法停用此記錄檔。	已啟用

此 Oracle Management Agent 日誌包含下表所示的日誌群組。

日誌名稱	CloudWatch 記錄群組
emctl.log	oemagent-emctl
emdctlj.log	oemagent-emdctlj
gcagent.log	oemagent-gcagent
gcagent_errors.log	oemagent-gcagent-errors
emagent.nohup	oemagent-emagent-nohup
secure.log	oemagent-secure

如需詳細資訊，請參閱 Oracle 文件中的[尋找 Management Agent 日誌和追蹤檔案](#)。

主控台

若要將 Oracle 資料庫 CloudWatch 記錄從 AWS Management Console

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Modify (修改)。
4. 在 [記錄匯出] 區段中，選擇您要開始發佈至 CloudWatch 記錄檔的記錄檔。
5. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Instance (修改資料庫執行個體)。

AWS CLI

欲發佈 Oracle 日誌，您可搭配下列參數使用 [modify-db-instance](#) 命令：

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 選項的變更一律立即套用到資料庫執行個體。因此，`--apply-immediately` 和 `--no-apply-immediately` 選項沒有效果。

您亦可使用下列命令來發佈 Oracle 日誌：

- [create-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

Example

下列範例會建立啟用 CloudWatch 記錄發佈的 Oracle 資料庫執行個體。--cloudwatch-logs-export-configuration 值為 JSON 字串陣列。字串可以是 alert、audit、listener 以及 trace 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration  
  '["trace","audit","alert","listener","oemagent"]' \  
  --db-instance-class db.m5.large \  
  --allocated-storage 20 \  
  --engine oracle-ee \  
  --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 \  
  --license-model bring-your-own-license \  
  --master-username myadmin \  
  --manage-master-user-password
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^  
  --db-instance-class db.m5.large ^  
  --allocated-storage 20 ^  
  --engine oracle-ee ^  
  --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 ^  
  --license-model bring-your-own-license ^  
  --master-username myadmin ^  
  --manage-master-user-password
```


Example

下列範例會修改現有的 Oracle 資料庫執行個體，以將記錄檔發佈至 CloudWatch 記錄檔。--cloudwatch-logs-export-configuration 值為 JSON 物件。此物件的金鑰為 EnableLogTypes，而其值為字串陣列，其中包含 alert、audit、listener 以及 trace 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":  
["trace","alert","audit","listener","oemagent"]}'
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\", \"alert\", \"audit  
\", \"listener\", \"oemagent\"
```

Example

下列範例會修改現有的 Oracle 資料庫執行個體，以停用將稽核和接聽程式記錄檔發佈至 CloudWatch 記錄檔。--cloudwatch-logs-export-configuration 值為 JSON 物件。此物件的金鑰為 DisableLogTypes，而其值為字串陣列，其中包含 alert、audit、listener 以及 trace 的任意組合。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\", \"listener\"
```

RDS API

您可以使用 RDS API 來發佈 Oracle 資料庫日誌。您可以使用下列參數來呼叫 [ModifyDBInstance](#) 動作：

- DBInstanceIdentifier
- CloudwatchLogsExportConfiguration

Note

CloudwatchLogsExportConfiguration 參數的變更一律立即套用到資料庫執行個體。因此，ApplyImmediately 參數沒有效果。

您也可以呼叫下列 RDS API 操作來發佈 Oracle 日誌：

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

以下列參數來執行其中一個 RDS API 操作：

- DBInstanceIdentifier
- EnableCloudwatchLogsExports
- Engine
- DBInstanceClass


視您執行的 RDS 操作而定，可能需要其他參數。

存取警示日誌和監聽器日誌

您可以使用 Amazon RDS 主控台檢視提醒日誌。您也可以使用下面的 SQL 語句。

```
SELECT message_text FROM alertlog;
```

使用 Amazon CloudWatch 日誌存取接聽程式日誌。

 Note

Oracle 會在提醒與接聽程式日誌超過 10 MB 時進行輪換，此時這些日誌將不會顯示於 Amazon RDS 的畫面。

RDS for PostgreSQL 資料庫日誌檔

RDS for PostgreSQL 會將資料庫活動記錄到預設的 PostgreSQL 日誌檔。對於內部部署 PostgreSQL 資料庫執行個體，這些訊息會在本機存放於 `log/postgresql.log` 中。對於 RDS for PostgreSQL 資料庫執行個體，日誌檔可在 Amazon RDS 執行個體上取得。此外，您必須使用 Amazon RDS 主控台來檢視或下載其內容。預設的記錄層級會擷取登入失敗、嚴重的伺服器錯誤、死鎖和查詢失敗。

如需有關如何檢視、下載和監看檔案型資料庫日誌的詳細資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。若要進一步了解 PostgreSQL 日誌，請參閱 [Working with Amazon RDS and Aurora PostgreSQL logs: Part 1](#) (使用 Amazon RDS 和 Aurora PostgreSQL 日誌：第 1 部分) 以及 [Working with Amazon RDS and Aurora PostgreSQL logs: Part 2](#) (使用 Amazon RDS 和 Aurora PostgreSQL 日誌：第 2 部分)。

除了本主題中討論的標準 PostgreSQL 日誌之外，RDS for PostgreSQL 也支援 PostgreSQL 稽核擴充功能 (pgAudit)。大多數受管制的產業和政府機構都需要維護對資料所做變更的稽核日誌或稽核線索，以符合法律要求。如需安裝與使用 pgAudit 的資訊，請參閱 [使用 PgAudit 記錄資料庫活動](#)。

主題

- [影響日誌記錄行為的參數](#)
- [針對您的 RDS for PostgreSQL 資料庫執行個體開啟查詢記錄](#)
- [將 PostgreSQL 日誌發佈到 Amazon 日誌 CloudWatch](#)

影響日誌記錄行為的參數

您可以修改各種參數，為 RDS for PostgreSQL 資料庫執行個體自訂記錄行為。在下表中，您可以找到影響日誌檔存放時間、何時輪換日誌，以及是否以 CSV (逗號分隔值) 格式輸出日誌。您也可以找到已傳送至 STDERR 的文字輸出，以及其他設定。若要變更可修改之參數的設定，請將自訂資料庫參數群組用於 RDS for PostgreSQL 執行個體。如需詳細資訊，請參閱 [在資料庫執行個體中使用資料庫參數群組](#)。如表中所述，無法變更 `log_line_prefix`。

參數	預設	描述
<code>log_destination</code>	<code>stderr</code>	設定日誌的輸出格式。預設值是 <code>stderr</code> ，您也可以將 <code>csvlog</code> 新增至設定來指定逗號分隔值 (CSV)。如需詳細資訊，請參閱 設定日誌目標 (stderr、csvlog) 。

參數	預設	描述
log_filename	postgresql.log.%Y-%m-%d-%H	指定日誌檔名稱的模式。除了預設值之外，此參數還支援檔案名稱模式的 postgresql.log.%Y-%m-%d。
log_line_prefix	%t:%r:%u@%d:[%p]:	定義寫入至 stderr 的每個日誌行的字首，以記錄時間 (%t)、遠端主機 (%r)、使用者 (%u)、資料庫 (%d) 和程序 ID (%p)。您無法修改此參數。
log_rotation_age	60	日誌檔會多少分鐘後自動轉換。您可以在 1 和 1440 分鐘的範圍內變更此值。如需詳細資訊，請參閱 設定日誌檔案輪換 。
log_rotation_size	–	日誌檔自動轉換的大小 (kB)。根據預設，不會使用此參數，因為記錄檔會根據 log_rotation_age 參數進行輪替。如需進一步了解，請參閱 設定日誌檔案輪換 。
rds.log_retention_period	4320	早於指定分鐘數的 PostgreSQL 日誌將遭到刪除。預設值 4320 分鐘將在 3 天後刪除日誌檔案。如需詳細資訊，請參閱 設定日誌保留期間 。

如要識別應用程式問題，您可在日誌中尋找查詢失敗、登入失敗、鎖死和致命的伺服器錯誤。例如，假設您已將舊版應用程式從 Oracle 轉換為 Amazon RDS PostgreSQL，但並非所有查詢都已正確轉換。這些格式不正確的查詢會產生您可在日誌中尋找的錯誤訊息，以協助識別問題。如需記錄查詢的詳細資訊，請參閱 [針對您的 RDS for PostgreSQL 資料庫執行個體開啟查詢記錄](#)。

在下列主題中，您可以找到如何設定各種參數的相關資訊，這些參數控制 PostgreSQL 日誌的基本詳細資訊。

主題

- [設定日誌保留期間](#)
- [設定日誌檔案輪換](#)
- [設定日誌目標 \(stderr、csvlog\)](#)

- [了解 log_line_prefix 參數](#)

設定日誌保留期間

`rds.log_retention_period` 參數指定 RDS for PostgreSQL 資料庫執行個體保留其日誌檔的時間長度。預設設定為 3 天 (4,320 分鐘)，但您可以將此值設為 1 天 (1,440 分鐘) 至 7 天 (10,080 分鐘)。請確定您的 RDS for PostgreSQL 資料庫執行個體具有足夠的儲存空間來保留日誌檔一段時間。

我們建議您定期將日誌發佈到 Amazon 日誌 CloudWatch 誌，以便在從 資料庫叢集中移除日誌後，可以很長時間檢視和分析系統資料。RDS for PostgreSQL 資料庫執行個體。如需詳細資訊，請參閱 [將 PostgreSQL 日誌發佈到 Amazon 日誌 CloudWatch](#)。

設定日誌檔案輪換

依預設，Amazon RDS 每小時都會建立新的日誌檔。時間由 `log_rotation_age` 參數控制。此參數的預設值為 60 (分鐘)，但您可以將其設為從 1 分鐘至 24 小時 (1,440 分鐘) 的任何時間。在輪換時，會建立一個新的不同日誌檔案。該檔案的命名是依據 `log_filename` 參數所指定的模式。

日誌檔案也可依其大小進行旋轉，如 `log_rotation_size` 參數中所指定。此參數指定當日誌達到指定大小 (以 KB 為單位) 時應輪換日誌。若為 RDS for PostgreSQL 資料庫執行個體，`log_rotation_size` 未設定，即並未指定任何值。不過，您可以設定從 0-2097151 KB 的參數。

日誌檔案名稱會以 `log_filename` 參數中指定的檔案名稱模式為基礎。此參數的可用設定如下所示：

- `postgresql.log.%Y-%m-%d` – 日誌檔名稱的預設格式。在日誌檔的名稱中包含年、月和日期。
- `postgresql.log.%Y-%m-%d-%H` – 在日誌檔名稱格式中包括小時。

如需詳細資訊，請參閱 PostgreSQL 文件中的 [log_rotation_age](#) 和 [log_rotation_size](#)。

設定日誌目標 (stderr、csvlog)

預設情況下，Amazon RDS PostgreSQL 生成標準錯誤 (stderr) 格式の日誌。此格式為 `log_destination` 參數的預設設定。每則訊息都會使用 `log_line_prefix` 參數中指定的模式作為字首。如需詳細資訊，請參閱 [了解 log_line_prefix 參數](#)。

RDS for PostgreSQL 也會以 `csvlog` 格式產生日誌檔。將日誌資料當作逗號分隔值 (CSV) 進行分析時，`csvlog` 很有用。例如，假設您使用 `log_fdw` 延伸模組，將日誌作為外部資料表處理。在 `stderr` 日誌檔案上建立的外部資料表包含一個具日誌事件資料的單一欄。透過將 `csvlog` 新增至

`log_destination` 參數，您可以取得 CSV 格式的日誌檔，其中包含外部資料表的多個資料欄的分界。您現在可以更輕鬆地排序和分析日誌。如要進一步了解如何使用 `log_fdw` 和 `csvlog`，請參閱 [使用 `log_fdw` 擴充功能存取使用 SQL 的資料庫日誌](#)。

如果您為此參數指定 `csvlog`，請注意會同時產生 `stderr` 和 `csvlog` 檔案。請務必監控日誌所使用的儲存體，同時考慮 `rds.log_retention_period` 及影響日誌儲存體和更換的其他設定。使用 `stderr` 和 `csvlog` 會使日誌所使用的儲存體空間增加一倍以上。

如果您將 `csvlog` 新增至 `log_destination`，並且想要單獨還原為 `stderr`，則需要重設參數。若要這麼做，請開啟 Amazon RDS 主控台，然後您的執行個體開啟自訂資料庫參數群組。選擇 `log_destination` 參數、選擇 Edit parameter (編輯參數)，然後選擇 Reset (重設)。

如需有關設定日誌記錄的詳細資訊，請參閱 [使用 Amazon RDS 和 Aurora PostgreSQL 日誌：第 1 部分](#)。

了解 `log_line_prefix` 參數

`stderr` 日誌格式會將 `log_line_prefix` 參數指定的詳細資訊作為每個日誌訊息的字首，如下所示。

```
%t:%r:%u@d:[%p]:t
```

您無法變更此設定。傳送至 `stderr` 的每個日誌項目都包括下列資訊。

- `%t` – 日誌項目的時間
- `%r` – 遠端主機地址
- `%u@d` – 使用者名稱 @ 資料庫名稱
- `[%p]` – 程序 ID (若可用)

針對您的 RDS for PostgreSQL 資料庫執行個體開啟查詢記錄

您可以設定下表中列出的一些參數，來收集有關資料庫活動的詳細資訊，包括查詢、等待鎖定的查詢、檢查點，以及許多其他詳細資訊。本主題著重於記錄查詢。

參數	預設	描述
<code>log_connections</code>	–	記錄每個成功連線。

參數	預設	描述
log_disconnections	–	記錄每個工作階段的結束及其持續時間。
log_checkpoints	1	記錄每個檢查點。
log_lock_waits	–	記錄長鎖定等待。根據預設，不會設定此參數。
log_min_duration_sample	–	(毫秒) 設定執行時間下限，超出此時間就會記錄陳述式樣本。使用 log_statement_sample_rate 參數設定範例大小。
log_min_duration_statement	–	至少執行指定時間或更長時間的任何 SQL 陳述式都會被記錄下來。根據預設，不會設定此參數。開啟此參數可以協助您尋找未最佳化的查詢。
log_statement	–	設定已記錄的陳述式類型。依預設，不會設定此參數，但您可以將其變更為 all、ddl 或 mod，以指定您要記錄的 SQL 陳述式類型。如果您針對這個參數指定了 none 以外的任何值，您也應該採取額外的步驟，以防止在日誌檔中暴露密碼。如需詳細資訊，請參閱 降低使用查詢記錄時密碼暴露的風險 。
log_statement_sample_rate	–	超過要記錄之 log_min_duration_sample 中所指定時間的陳述式百分比，以介於 0.0 與 1.0 之間的浮點值表示。
log_statement_stats	–	將累積效能統計資訊寫入至伺服器日誌。

使用記錄來尋找執行緩慢的查詢

您可以記錄 SQL 陳述式和查詢，以協助尋找執行緩慢的查詢。您可以依照本節所述修改 log_statement 和 log_min_duration 參數中的設定來開啟此功能。在針對您的 RDS for PostgreSQL 資料庫執行個體 開啟查詢記錄之前，您應該注意到日誌檔中可能暴露密碼，以及如何降低風險。如需詳細資訊，請參閱 [降低使用查詢記錄時密碼暴露的風險](#)。

接下來，您可以尋找有關 log_statement 和 log_min_duration 參數的參考資訊。

log_statement

此參數指定應該傳送至日誌的 SQL 陳述式類型。預設值為 none。如果您將此參數變更為 all、ddl 或 mod，請務必套用建議的動作，以降低在日誌檔中暴露密碼的風險。如需詳細資訊，請參閱 [降低使用查詢記錄時密碼暴露的風險](#)。

全部

記錄所有陳述式。此設定是基於偵錯用途而建議的。

DDL

記錄所有資料定義語言 (DDL) 陳述式，例如 CREATE、ALTER、DROP 等。

MOD

記錄所有 DDL 陳述式和資料操作語言 (DML) 陳述式，例如 INSERT、UPDATE 和 DELETE)，這些陳述式會修改資料。

無

不會記錄任何 SQL 陳述式。建議您使用此設定，以避免在日誌中暴露密碼的風險。

log_min_duration_statement

至少執行指定時間或更長時間的任何 SQL 陳述式都會被記錄下來。根據預設，不會設定此參數。開啟此參數可以協助您尋找未最佳化的查詢。

-1-2147483647

記錄陳述式之執行時間的毫秒數。

設定查詢記錄

這些步驟假設您的 RDS for PostgreSQL 資料庫執行個體使用自訂資料庫參數群組。

1. 將 log_statement 參數設為 all。下列範例顯示使用此參數設定寫入至 postgresql.log 檔案的資訊。

```
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: statement:
SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
```

```

2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: QUERY
STATISTICS
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:DETAIL: ! system
usage stats:
! 0.017355 s user, 0.000000 s system, 0.168593 s elapsed
! [0.025146 s user, 0.000000 s system total]
! 36644 kB max resident size
! 0/8 [0/8] filesystem blocks in/out
! 0/733 [0/1364] page faults/reclaims, 0 [0] swaps
! 0 [0] signals rcvd, 0/0 [0/0] messages rcvd/sent
! 19/0 [27/0] voluntary/involuntary context switches
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: SELECT
feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:ERROR: syntax error
at or near "ORDER" at character 1
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: ORDER BY
s.confidence DESC;
----- END OF LOG -----

```

- 設定 `log_min_duration_statement` 參數。下列範例說明此參數設定為 `postgresql.log` 時寫入至 1 檔案的資訊：

系統會記錄超過 `log_min_duration_statement` 參數中所指定持續時間的查詢。下列顯示一個範例。您可以在 Amazon RDS 主控台中檢視 RDS for PostgreSQL 資料庫執行個體的日誌檔。

```

2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: statement: DROP
table comments;
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: duration:
167.754 ms
2022-10-05 19:08:07 UTC::@[355]:LOG: checkpoint starting: time
2022-10-05 19:08:08 UTC::@[355]:LOG: checkpoint complete: wrote 11 buffers
(0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.013 s, sync=0.006 s,
total=1.033 s; sync files=8, longest=0.004 s, average=0.001 s; distance=131028 kB,
estimate=131028 kB
----- END OF LOG -----

```

降低使用查詢記錄時密碼暴露的風險

建議您保持 `log_statement` 設定為 `none` 以避免密碼暴露。如果您將 `log_statement` 設定為 `all`、`ddl` 或 `mod`，建議您採取下列一或多個步驟。

- 對於用戶端，請加密敏感資訊。如需的詳細資訊，請參閱 PostgreSQL 文件中的[加密選項](#)。使用 CREATE 和 ALTER 陳述式的 ENCRYPTED (和 UNENCRYPTED) 選項。如需詳細資訊，請參閱 PostgreSQL 文件中的 [CREATE USER](#)。
- 對於您的 RDS for PostgreSQL 資料庫執行個體，請設定並使用 PostgreSQL 稽核 (pgAudit) 擴充功能。此擴充功能會刪減傳送至日誌的 CREATE 和 ALTER 陳述式中的敏感資訊。如需詳細資訊，請參閱 [使用 PgAudit 記錄資料庫活動](#)。
- 限制對 CloudWatch 日誌的訪問。
- 使用更強大的身分驗證機制，例如 IAM。

將 PostgreSQL 日誌發佈到 Amazon 日誌 CloudWatch

若要將 PostgreSQL 日誌記錄存放在高耐用性儲存體中，您可以使用 Amazon CloudWatch 日誌。使用 CloudWatch Logs，您還可以對日誌數據進行實時分析，並用 CloudWatch 於查看指標和創建警報。例如，如果您將 `log_statement` 設定為 `ddl`，您可以設定警示，以便在執行 DDL 陳述式時發出提醒。在建立適用於 PostgreSQL 資料庫執行個體的過程中，您可以選擇將 PostgreSQL CloudWatch 記錄檔上傳至記錄檔。如果您當時選擇不上傳日誌，您可以稍後修改執行個體，從該點開始上傳日誌。換言之，不會上傳現有日誌。只會上傳新的日誌，因為它們是在修改後的 RDS for PostgreSQL 資料庫執行個體上建立的。

所有目前可用的 RDS 版本都支援將記錄檔發佈至 CloudWatch 記錄檔。如需詳細資訊，請參閱《Amazon RDS for PostgreSQL 版本備註》中的 [Amazon RDS for PostgreSQL 更新](#)。

若要使用 CloudWatch 記錄，請將 RDS for PostgreSQL 資料庫執行個體設定為將記錄資料發佈到記錄群組。

您可以將下列記錄類型發佈到 RDS 的 CloudWatch 日誌 PostgreSQL：

- Postgresql 日誌
- 升級日誌

完成組態後，Amazon RDS 會發佈日誌事件以記錄日誌群組內的串流。例如，PostgreSQL 日誌資料存放在日誌群組 `/aws/rds/instance/my_instance/postgresql` 內。若要檢視您的記錄檔，請開啟主 CloudWatch 控制台，網址為 <https://console.aws.amazon.com/cloudwatch/>。

主控台

若要使用主控台將 PostgreSQL 記錄檔發佈到 CloudWatch 記錄檔

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要修改的資料庫執行個體，然後選擇 Modify (修改)。
4. 在 [記錄匯出] 區段中，選擇您要開始發佈至 CloudWatch 記錄檔的記錄檔。

「記錄匯出」區段僅適用於支援發佈至 CloudWatch 記錄的 PostgreSQL 版本。

5. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Instance (修改資料庫執行個體)。

AWS CLI

您 PostgreSQL 用 AWS CLI 您可以使用下列參數來呼叫 [modify-db-instance](#) 命令。

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 選項的變更一律立即套用到資料庫執行個體。因此，`--apply-immediately` 和 `--no-apply-immediately` 選項沒有效果。

您也可以呼叫下列 CLI 命令來發布 PostgreSQL 日誌：

- [create-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-to-point-in-time](#)

執行上述其中一個 CLI 命令並指定下列選項：

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`

- `--engine`

視您執行的 CLI 命令而定，可能需要其他選項。

Example 修改執行個體以將記錄檔發佈至 CloudWatch 記錄

下列範例會修改現有 PostgreSQL 資料庫執行個體，以將記錄檔發佈至 CloudWatch 記錄檔。`--cloudwatch-logs-export-configuration` 值為 JSON 物件。此物件的索引鍵為 `EnableLogTypes`，而其值為包含 `postgresql` 和 `upgrade` 任意組合的字串陣列。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql",  
"upgrade"]}'
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":  
["postgresql","upgrade"]}'
```

Example 建立執行個體以將記錄檔發佈至 CloudWatch 記錄

下列範例會建立 PostgreSQL 資料庫執行個體，並將記錄檔發佈至 CloudWatch 記錄檔。`--enable-cloudwatch-logs-exports` 值為 JSON 字串陣列。字串可以是 `postgresql` 和 `upgrade` 的任意組合。

對於 Linux/macOS、或 Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' \  
  --db-instance-class db.m4.large \  
  --engine postgres
```

在 Windows 中：

```
aws rds create-db-instance ^
  --db-instance-identifier mydbinstance ^
  --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' ^
  --db-instance-class db.m4.large ^
  --engine postgres
```

RDS API

您可以使用 RDS API 來發佈 PostgreSQL 日誌。您可以使用下列參數來呼叫 [ModifyDBInstance](#) 動作：

- DBInstanceIdentifier
- CloudwatchLogsExportConfiguration

Note

CloudwatchLogsExportConfiguration 參數的變更一律立即套用到資料庫執行個體。因此，ApplyImmediately 參數沒有效果。

您也可以呼叫下列 RDS API 操作來發佈 PostgreSQL 日誌：

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceToPointInTime](#)

以下列參數來執行其中一個 RDS API 操作：

- DBInstanceIdentifier
- EnableCloudwatchLogsExports
- Engine
- DBInstanceClass

視您執行的操作而定，可能需要其他參數。

在 AWS CloudTrail 中監控 Amazon RDS API 呼叫

AWS CloudTrail 是一項可協助您稽核 AWS 帳戶的 AWS 服務。AWS CloudTrail 會在您建立 AWS 帳戶時開啟。如需有關 CloudTrail 的詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。

主題

- [CloudTrail 與 Amazon RDS 整合](#)
- [Amazon RDS 日誌檔案項目](#)

CloudTrail 與 Amazon RDS 整合

CloudTrail 會記錄所有 Amazon RDS 動作。對於使用者、角色或 AWS 服務在 Amazon RDS 中採取的動作，CloudTrail 會提供記錄。

CloudTrail 事件

CloudTrail 會擷取 Amazon RDS 的 API 呼叫當作事件。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。事件包括從 Amazon RDS 主控台的呼叫，以及對 Amazon RDS API 操作的程式碼呼叫。

Amazon RDS 活動會記錄在 Event history (事件歷史記錄) 的 CloudTrail 事件中。您可以使用 CloudTrail 主控台檢視 AWS 區域中過去 90 天所記錄的 API 活動和事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷程記錄檢視事件](#)。

CloudTrail 線索

如需 AWS 帳戶中正在記錄事件 (包括 Amazon RDS 的事件)，請建立 trail (追蹤)。權杖是一種組態，能讓事件交付到指定的 Amazon S3 儲存貯體。CloudTrail 通常會在帳戶活動的 15 分鐘內交付日誌檔案。

Note

即使您未設定權杖，依然可以透過 CloudTrail 主控台內的 Event history (事件歷史記錄) 檢視最新事件。

您可以為 AWS 帳戶建立兩種類型的追蹤：套用至所有區域的追蹤，或套用至一個區域的追蹤。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有區域。

此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案](#)

Amazon RDS 日誌檔案項目

CloudTrail 日誌檔案包含一或多個日誌項目。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 CreateDBInstance 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2018-07-30T22:14:06Z",
  "eventSource": "rds.amazonaws.com",
  "eventName": "CreateDBInstance",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
  "requestParameters": {
    "enableCloudwatchLogsExports": [
      "audit",
      "error",
      "general",
      "slowquery"
    ],
    "dbInstanceIdentifier": "test-instance",
    "engine": "mysql",
```

```
    "masterUsername": "myawsuser",
    "allocatedStorage": 20,
    "dbInstanceClass": "db.m1.small",
    "masterUserPassword": "*****"
  },
  "responseElements": {
    "dbInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
    "storageEncrypted": false,
    "preferredBackupWindow": "10:27-10:57",
    "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
    "backupRetentionPeriod": 1,
    "allocatedStorage": 20,
    "storageType": "standard",
    "engineVersion": "8.0.28",
    "dbInstancePort": 0,
    "optionGroupMemberships": [
      {
        "status": "in-sync",
        "optionGroupName": "default:mysql-8-0"
      }
    ],
    "dbParameterGroups": [
      {
        "dbParameterGroupName": "default.mysql8.0",
        "parameterApplyStatus": "in-sync"
      }
    ],
    "monitoringInterval": 0,
    "dbInstanceClass": "db.m1.small",
    "readReplicaDBInstanceIdentifiers": [],
    "dbSubnetGroup": {
      "dbSubnetGroupName": "default",
      "dbSubnetGroupDescription": "default",
      "subnets": [
        {
          "subnetAvailabilityZone": {"name": "us-east-1b"},
          "subnetIdentifier": "subnet-cbfff283",
          "subnetStatus": "Active"
        },
        {
          "subnetAvailabilityZone": {"name": "us-east-1e"},
          "subnetIdentifier": "subnet-d7c825e8",
          "subnetStatus": "Active"
        }
      ]
    }
  },
```

```
    {
      "subnetAvailabilityZone": {"name": "us-east-1f"},
      "subnetIdentifier": "subnet-6746046b",
      "subnetStatus": "Active"
    },
    {
      "subnetAvailabilityZone": {"name": "us-east-1c"},
      "subnetIdentifier": "subnet-bac383e0",
      "subnetStatus": "Active"
    },
    {
      "subnetAvailabilityZone": {"name": "us-east-1d"},
      "subnetIdentifier": "subnet-42599426",
      "subnetStatus": "Active"
    },
    {
      "subnetAvailabilityZone": {"name": "us-east-1a"},
      "subnetIdentifier": "subnet-da327bf6",
      "subnetStatus": "Active"
    }
  ],
  "vpcId": "vpc-136a4c6a",
  "subnetGroupStatus": "Complete"
},
"masterUsername": "myawsuser",
"multiAZ": false,
"autoMinorVersionUpgrade": true,
"engine": "mysql",
"caCertificateIdentifier": "rds-ca-2015",
"dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",
"dbSecurityGroups": [],
"pendingModifiedValues": {
  "masterUserPassword": "*****",
  "pendingCloudwatchLogsExports": {
    "logTypesToEnable": [
      "audit",
      "error",
      "general",
      "slowquery"
    ]
  }
}
},
"dbInstanceStatus": "creating",
"publiclyAccessible": true,
```

```
    "domainMemberships": [],
    "copyTagsToSnapshot": false,
    "dbInstanceIdentifier": "test-instance",
    "licenseModel": "general-public-license",
    "iAMDatabaseAuthenticationEnabled": false,
    "performanceInsightsEnabled": false,
    "vpcSecurityGroups": [
      {
        "status": "active",
        "vpcSecurityGroupId": "sg-f839b688"
      }
    ]
  },
  "requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",
  "eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

如上述範例中的 `userIdentity` 元素所示，每個事件或記錄項目都包含產生要求者的相關資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需 `userIdentity` 的詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。如需有關 `CreateDBInstance` 和其他 Amazon RDS 動作的詳細資訊，請參閱 [《Amazon RDS API 參考》](#)。

使用資料庫活動串流來監控 Amazon RDS

透過使用資料庫活動串流，您就可以監控資料庫活動的近乎即時的串流。

主題

- [資料庫活動串流概觀](#)
- [設定 Oracle 資料庫的統一稽核](#)
- [正在配置 Microsoft SQL Server 稽核政策](#)
- [開始資料庫活動串流](#)
- [修改資料庫活動串流](#)
- [取得資料庫活動串流的狀態](#)
- [停用資料庫活動串流](#)
- [監控資料庫活動串流](#)
- [管理資料庫活動串流的存取](#)

資料庫活動串流概觀

作為 Amazon RDS 資料庫管理員，您需要保護資料庫並遵守合規與法規要求。其中一項策略是整合資料庫活動串流與監控工具。透過此方式，您可以監控 資料庫中的稽核活動，並設定警示。

外部和內部都有安全威脅。若要防範內部威脅，您可以設定資料庫活動串流功能來控制管理員對資料串流的存取。Amazon RDS 資料庫管理員沒有存取收集、傳輸、儲存和處理串流的權限。

主題

- [資料庫活動串流運作方式](#)
- [在 Oracle 資料庫和 Microsoft SQL Server 資料庫中進行稽核](#)
- [資料庫活動串流的非同步模式](#)
- [資料庫活動串流的要求與限制](#)
- [區域和版本可用性](#)
- [支援資料庫活動串流的資料庫執行個體類別](#)

資料庫活動串流運作方式

Amazon RDS 會近乎即時地將活動推送至 Amazon Kinesis 資料串流。系統會自動建立 Kinesis 串流。在 Kinesis 中，您可以設定 AWS 服務 (例如 Amazon 資料 Firehose)，AWS Lambda 以及使用串流和存放資料。

Important

在 Amazon RDS 中使用資料庫活動串流功能是一項免費功能，但 Amazon Kinesis 會收取資料串流費用。如需詳細資訊，請參閱 [Amazon Kinesis Data Streams 定價](#)。

您可為合規管理設定應用程式以使用資料庫活動串流。這些應用程式可以使用串流來產生警示，並稽核您的 資料庫。

Amazon RDS 支援異地同步備份部署中的資料庫活動串流。於此狀況下，資料庫活動串流會同時稽核主執行個體和備用執行個體。

在 Oracle 資料庫和 Microsoft SQL Server 資料庫中進行稽核

稽核是針對已設定的資料庫執行的監控和記錄動作。Amazon RDS 根據預設不會擷取資料庫活動。您可以自行在資料庫中建立和管理稽核政策。

主題

- [Oracle 資料庫中的統一稽核](#)
- [在 Microsoft SQL Server 中進行稽核](#)
- [Oracle 資料庫和 SQL Server 的非原生稽核欄位](#)
- [資料庫參數群組覆寫](#)

Oracle 資料庫中的統一稽核

在 Oracle 資料庫中，統一稽核政策是稽核設定的具名群組，您可以使用此群組來稽核使用者行為。政策可以簡單地稽核單個使用者的活動。您也可以建立使用條件的複雜稽核政策。

Oracle 資料庫會寫入稽核記錄，包括從 SYS 稽核記錄至統一稽核權杖，皆會記錄。例如，如果在 INSERT 陳述式期間發生錯誤，標準稽核會指出錯誤編號和所執行的 SQL。稽核權杖位於 AUDSYS 結構描述中的唯讀資料表中。若要存取這些記錄，可以查詢 UNIFIED_AUDIT_TRAIL 資料字典檢視。

一般而言，您可如下所示，設定資料庫活動串流：

1. 透過使用 CREATE AUDIT POLICY 命令，建立 Oracle 資料庫稽核政策。

Oracle 資料庫會產生稽核記錄。

2. 透過使用 AUDIT POLICY 命令，啟用稽核政策。
3. 設定資料庫活動串流。

只有符合 Oracle 資料庫稽核政策的活動才會被擷取，並傳送至 Amazon Kinesis Data Streams。在啟用資料庫活動串流時，Oracle 資料庫管理員無法變更稽核政策或移除稽核記錄。

若要進一步了解統一稽核政策，請參閱 Oracle 資料庫安全指南中的[關於使用統一稽核政策和稽核來稽核活動](#)。

在 Microsoft SQL Server 中進行稽核

資料庫活動串流使用 SQLAudit 功能來稽核 SQL Server 資料庫。

RDS for SQL Server 執行個體包含下列項目：

- 伺服器稽核 — SQL Server 稽核會收集伺服器或資料庫層級動作的單一執行個體，以及要監控的動作群組。伺服器層級稽核 RDS_DAS_AUDIT 和 RDS_DAS_AUDIT_CHANGES 由 RDS 管理。
- 伺服器稽核規格 — 伺服器稽核規格會記錄伺服器層級事件。您可以修改 RDS_DAS_SERVER_AUDIT_SPEC 規格。此規格已連結至伺服器稽核 RDS_DAS_AUDIT。該 RDS_DAS_CHANGES_AUDIT_SPEC 規格由 RDS 管理。
- 資料庫稽核規格 — 資料庫稽核規格會記錄資料庫層級事件。您可以建立資料庫稽核規格 RDS_DAS_DB_<name>，並將其連結至 RDS_DAS_AUDIT 伺服器稽核。

您可以使用主控台或 CLI 來設定資料庫活動串流。一般而言，您可如下所示，設定資料庫活動串流：

1. (選用) 使用 CREATE DATABASE AUDIT SPECIFICATION 命令建立資料庫稽核規格，並將其連結至 RDS_DAS_AUDIT 伺服器稽核。
2. (選用) 使用 ALTER SERVER AUDIT SPECIFICATION 命令修改伺服器稽核規格，並定義政策。
3. 啟用資料庫和伺服器稽核政策。例如：

```
ALTER DATABASE AUDIT SPECIFICATION [<Your database specification>] WITH  
(STATE=ON)
```

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC] WITH  
(STATE=ON)
```

4. 設定資料庫活動串流。

只有符合該伺服器與資料庫稽核政策的活動才會被擷取，並傳送至 Amazon Kinesis Data Streams。在啟用資料庫活動串流且政策鎖定時，資料庫管理員無法變更稽核政策或移除稽核記錄。

Important

若已啟用特定資料庫的資料庫稽核規格，且政策處於鎖定狀態，則無法捨棄該資料庫。

如需有關 SQL Server 稽核的詳細資訊，請參閱 Microsoft SQL Server 文件中的 [SQL Server 稽核元件](#)。

Oracle 資料庫和 SQL Server 的非原生稽核欄位

在您啟動資料庫活動串流時，每個資料庫事件都會產生對應的活動串流事件。例如，資料庫使用者可能會執行 SELECT 和 INSERT 陳述式。資料庫會稽核這些事件，並將其傳送至 Amazon Kinesis Data Streams。

串流中的事件會以 JSON 物件表示。JSON 物件包含 DatabaseActivityMonitoringRecord，其中包含 databaseActivityEventList 陣列。陣列中的預先定義欄位包含 class、clientApplication 以及 command。

根據預設，活動串流不包含引擎原生稽核欄位。您可以設定 Amazon RDS for Oracle 和 Amazon RDS for Oracle，使其能在 engineNativeAuditFields JSON 物件中包含這些額外欄位。

在 Oracle 資料庫中，整合稽核權杖中的大部分事件都可映射至 RDS 資料活動串流中的欄位。例如，統一稽核中的 UNIFIED_AUDIT_TRAIL.SQL_TEXT 欄位會映射至資料庫活動串流中的 commandText 欄位。但是，如 OS_USERNAME 等 Oracle 資料庫稽核欄位，則不會映射到資料庫活動串流中的預先定義欄位。

在 SQL Server 中，SQLAudit 所記錄的大部分事件欄位都會對應至 RDS 資料庫活動串流的欄位。例如，稽核 sys.fn_get_audit_file 的 code 欄位會對應至資料庫活動串流的 commandText 欄位。但是，如 permission_bitmask 等 SQL Server 資料庫稽核欄位，則不會對應到資料庫活動串流中的預先定義欄位。

如需 databaseActivityEvent 「清單」的更多資訊，請參閱[databaseActivityEvent列出數組](#)。

資料庫參數群組覆寫

一般而言，您可以透過連接參數群組，在 RDS 中開啟統一稽核功能。但是，資料庫活動串流需要其他配置。為了改善您的客戶體驗，Amazon RDS 會執行下列內容：

- 如果您啟用活動串流，則 RDS for Oracle 會忽略參數群組中的稽核參數。
- 如果您停用活動串流，則 RDS for Oracle 會不再忽略稽核參數。

SQL Server 的資料庫活動串流與您在 SQL 稽核選項中設定的任何參數皆無關。

資料庫活動串流的非同步模式

Amazon RDS 中的活動串流始終為非同步。在資料庫工作階段產生活動串流事件時，工作階段會立即傳回正常的活動。Amazon RDS 會在背景中將活動串流事件變成耐久的記錄。

如果在背景任務中發生錯誤，Amazon RDS 就會產生事件。此事件會指出活動串流事件記錄可能遺失的任何時段的開頭和結尾。非同步模式對資料庫效能的幫助較大，對活動串流精準度的幫助較小。

資料庫活動串流的要求與限制

在 RDS 中，資料庫活動串流具有以下要求和限制：

- 資料庫活動串流需要使用 Amazon Kinesis。
- AWS Key Management Service 資料庫活動串流需要 (AWS KMS)，因為它們一律會加密。
- 對 Amazon Kinesis 資料串流套用額外加密與已使用金 AWS KMS 鑰加密的資料庫活動串流不相容。
- 您可自行建立和管理稽核政策。與 Amazon Aurora 不同，根據預設，RDS for Oracle 不會擷取資料庫活動。
- 您可自行建立和管理稽核政策或規格。與 Amazon Aurora 不同，根據預設，Amazon RDS 不會擷取資料庫活動。
- 於異地同步備份部署中，僅於主要資料庫執行個體上啟動資料庫活動串流。活動串流會自動稽核主資料庫執行個體和備用資料庫執行個體。容錯移轉期間不需要執行其他步驟。
- 重新命名資料庫執行個體並不會建立新的 Kinesis 串流。
- RDS for Oracle 不支援 CDB。
- 不支援僅供讀取複本。

區域和版本可用性

每個資料庫引擎的特定版本以及 AWS 區域 的功能可用性和支援各自不同。如需有關資料庫活動串流版本和區域可用性的詳細資訊，請參閱 [Amazon RDS 中資料庫活動串流支援的區域和資料庫引擎](#)。

支援資料庫活動串流的資料庫執行個體類別

針對 RDS for Oracle,您可以搭配下列資料庫執行個體類別來使用資料庫活動串流：

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5.*large.tpc*.mem*x
- db.r5b.*large
- db.r5b.*large.tpc*.mem*x
- db.r5d.*large
- db.r5.large*
- db.x2idn.*large
- db.x2iedn.*large
- db.x2iezn.*large
- db.z1d.*large

針對 RDS for SQL Server,您可以搭配下列資料庫執行個體類別來使用資料庫活動串流：

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large

- db.r5.*large
- db.r5b.*large
- db.r5d.*large
- db.r5.large*
- db.x1e.*large
- db.z1d.*large

如需執行個體類別類型的詳細資訊，請參閱 [資料庫執行個體類別](#)。

設定 Oracle 資料庫的統一稽核

在您設定統一稽核以搭配資料庫活動串流使用時，可能會出現以下情況：

- 未針對您的 Oracle 資料庫設定統一稽核。

在此情況下，請使用 CREATE AUDIT POLICY 命令建立新政策，然後使用 AUDIT POLICY 命令啟用這些政策。以下範例會建立並啟用政策，以監控具有特定權限和角色的使用者。

```
CREATE AUDIT POLICY table_pol
PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
ROLES emp_admin, sales_admin;

AUDIT POLICY table_pol;
```

如需完整指示，請參閱 Oracle 資料庫文件中的 [設定稽核政策](#)。

- 已針對您的 Oracle 資料庫設定統一稽核。

在您啟用資料庫活動串流時，RDS for Oracle 會自動清除現有的稽核資料。它也會撤銷稽核權杖權限。RDS for Oracle 無法再執行下列作業：

- 清除統一稽核追蹤記錄。
- 新增、刪除或修改統一稽核政策。
- 更新上次封存的時間戳記。

Important

我們強烈建議您在啟用資料庫活動串流前，先行備份稽核資料。

如需 UNIFIED_AUDIT_TRAIL 檢視的說明，請參閱 [UNIFIED_AUDIT_TRAIL](#)。如果您有 Oracle 支援的帳戶，請參閱[如何清除統一稽核權杖](#)。

正在配置 Microsoft SQL Server 稽核政策

SQL Server 資料庫執行個體具有由 Amazon RDS 管理的伺服器稽核 RDS_DAS_AUDIT。您可以定義政策，以便在伺服器稽核規格 RDS_DAS_SERVER_AUDIT_SPEC 中記錄伺服器事件。您可以建立資料庫稽核規格 (例如 RDS_DAS_DB_<name>)，並定義記錄資料庫事件的政策。如需伺服器和資料庫層級稽核動作群組的清單，請參閱《Microsoft SQL Server 文件》中的 [SQL Server 稽核動作群組和動作](#)。

預設伺服器政策只會監控失敗的登入，以及資料庫活動串流中，資料庫或伺服器稽核規格的變更。

稽核與稽核規格的限制包括下列各項：

- 當資料庫活動串流處於鎖定狀態時，您無法修改伺服器或資料庫稽核規格。
- 您無法修改伺服器稽核 RDS_DAS_AUDIT 規格。
- 您無法修改 SQL Server 稽核 RDS_DAS_CHANGES，或其相關伺服器稽核規格 RDS_DAS_CHANGES_AUDIT_SPEC。
- 建立資料庫稽核規格時，您必須使用格式 RDS_DAS_DB_<name>，例如 RDS_DAS_DB_databaseActions。

Important

對於較小的執行個體類別，建議您只稽核所需資料，而非所有內容。這能降低資料庫活動串流對這些執行個體類別的效能影響。

下列程式碼範例會修改伺服器稽核規格 RDS_DAS_SERVER_AUDIT_SPEC，並稽核任何登出和成功登入動作：

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
    WITH (STATE=OFF);
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
    ADD (LOGOUT_GROUP),
    ADD (SUCCESSFUL_LOGIN_GROUP)
    WITH (STATE = ON );
```

下列程式碼範例會建立資料庫稽核規格 RDS_DAS_DB_database_spec，並將其連接至伺服器稽核 RDS_DAS_AUDIT：

```
USE testDB;
CREATE DATABASE AUDIT SPECIFICATION [RDS_DAS_DB_database_spec]
  FOR SERVER AUDIT [RDS_DAS_AUDIT]
  ADD ( INSERT, UPDATE, DELETE
        ON testTable BY testUser )
  WITH (STATE = ON);
```

設定稽核規格後，請確定規格 RDS_DAS_SERVER_AUDIT_SPEC 和 RDS_DAS_DB_<name> 皆設為 ON 狀態。現在，他們可以將稽核資料傳送至您的資料庫活動串流。

開始資料庫活動串流

在開啟資料庫執行個體的活動串流時，您在稽核政策中設定的每個資料庫活動事件都會產生活動串流事件。CONNECT 和 SELECT 之類的 SQL 命令會產生存取事件。CREATE 和 INSERT 之類的 SQL 命令會產生變更事件。

Important

啟用 Oracle 資料庫執行個體的活動串流會清除現有的稽核資料。它也會撤銷稽核權杖權限。啟用串流時，RDS for Oracle 便無法再執行下列動作：

- 清除統一稽核追蹤記錄。
- 新增、刪除或修改統一稽核政策。
- 更新上次封存的時間戳記。

主控台

若要開始資料庫活動串流

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要在其上啟動活動串流的 Amazon RDS 資料庫執行個體。於異地同步備份部署中，僅於主要資料庫執行個體上啟動串流。活動串流會稽核主資料庫執行個體和備用資料庫執行個體兩者。
4. 針對 Actions (動作)，選擇 Start activity stream (啟動活動串流)。

Start database activity stream: *name* (開始資料庫活動串流 : name) 視窗隨即出現，其中 *name* 是您的 RDS 執行個體。

5. 輸入以下設定：

- 對於 AWS KMS key，請從 AWS KMS keys 清單中選擇一個金鑰。


Amazon RDS 會使用 KMS 金鑰來加密金鑰，此金鑰會依序加密資料庫活動。請選擇預設金鑰以外的 KMS 金鑰。如需更多有關加密金鑰和 AWS KMS 的資訊，請參閱 AWS Key Management Service 開發人員指南中的 [什麼是 AWS Key Management Service ?](#)。

- 若為資料庫活動事件，請選擇包含引擎原生稽核欄位，以包含引擎特定稽核欄位。
- 選擇 Immediately (立即)。

在您選擇 Immediately (立即) 時，RDS 執行個體會立即重新啟動。如果您選擇 During the next maintenance window (下個維護時段期間)，則 RDS 執行個體不會立即重新啟動。在這種情況下直到下一個維護時段前，資料庫活動串流都不會啟動。

6. 選擇 Start database activity stream (啟動資料庫活動串流)。

資料庫的狀態會顯示活動串流正在開始。

 Note

如果收到錯誤訊息 You can't start a database activity stream in this configuration，請檢查 [支援資料庫活動串流的資料庫執行個體類別](#) 以查看 RDS 執行個體是否使用支援的執行個體類別。

AWS CLI

若要啟動資料庫的資料庫活動串流資料庫執行個體，請使用 [start-activity-stream](#) AWS CLI 指令為資料庫設定資料庫。

- `--resource-arn arn` – 指定資料庫的 Amazon 資源名稱 (ARN) 執行個體。
- `--kms-key-id key` – 指定用於加密資料庫活動串流中訊息的 KMS 金鑰識別碼。AWS KMS 金鑰識別碼是 AWS KMS key 的金鑰 ARN、金鑰 ID、別名 ARN 或別名。
- `--engine-native-audit-fields-included` – 在資料串流中包含特定於引擎的稽核欄位。若要排除這些欄位，請指定 `--no-engine-native-audit-fields-included` (預設)。

下列範例會在非同步模式下開始資料庫執行個體的活動串流。

對於LinuxmacOS、或Unix：

```
aws rds start-activity-stream \  
  --mode async \  
  --kms-key-id my-kms-key-arn \  
  --resource-arn my-instance-arn \  
  --engine-native-audit-fields-included \  
  --apply-immediately
```

在Windows中：

```
aws rds start-activity-stream ^  
  --mode async ^  
  --kms-key-id my-kms-key-arn ^  
  --resource-arn my-instance-arn ^  
  --engine-native-audit-fields-included ^  
  --apply-immediately
```

RDS API

若要啟動資料庫的資料庫活動串流資料庫執行個體，請使用。

使用以下參數呼叫動作：

- Region
- KmsKeyId
- ResourceArn
- Mode
- EngineNativeAuditFieldsIncluded

修改資料庫活動串流

您可能想要在活動串流開始後，自訂 Amazon RDS 稽核政策。如果您不想停止活動串流，浪費時間和資料，您可以將稽核政策狀態變更為下列設定之一：

已鎖定 (預設)

資料庫中的稽核政策皆為僅供讀取。

已解鎖

資料庫中的稽核政策為讀取/寫入。

基本步驟如下：

1. 將稽核政策狀態修改為已解鎖。
2. 自訂稽核政策。
3. 將稽核政策狀態修改為已鎖定。

主控台

若要修改活動串流的稽核政策狀態

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 對於 Action (動作)，選擇 Modify database activity stream (修改資料庫活動串流)。

會出現 Modify database activity stream: *name* (修改資料庫活動串流：名稱) 視窗，其中 *name* (名稱) 是您的 RDS 執行個體。

4. 請選擇下列其中一個選項：

Locked (已鎖定)

當您鎖定稽核政策時，它會變成僅供讀取。除非您解除鎖定政策或停止活動串流，否則無法編輯稽核政策。

Unlocked (已解鎖)

當您解除鎖定稽核政策時，它會變成讀取/寫入。您可以在活動串流已啟動時編輯稽核政策。

5. 選擇 Modify DB activity stream (修改資料庫活動串流)。

Amazon RDS 資料庫的狀態會顯示設定活動串流。

6. (選用) 選擇資料庫執行個體連結。然後選擇 Configuration (組態) 標籤。

所以 Audit policy status (稽核政策狀態) 欄位顯示下列其中一個值：

- Locked (已鎖定)
- Unlocked (已解鎖)

- Locking policy (鎖定政策)
- Unlocking policy (解除鎖定政策)

AWS CLI

若要修改資料庫執行個體的活動串流狀態，請使用[modify-activity-stream](#) AWS CLI 指令。

選項	是否為必要？	描述
<code>--resource-arn <i>my-instance-ARN</i></code>	是	RDS 資料庫執行個體的 Amazon Resource Name (ARN)。
<code>--audit-policy-state</code>	否	新的執行個體上資料庫活動串流的政策狀態：locked 或 unlocked。

下列範例會解除鎖定已啟動活動串流的稽核政策 *my-instance-ARN*。

對於Linux macOS、或Unix：

```
aws rds modify-activity-stream \
  --resource-arn my-instance-ARN \
  --audit-policy-state unlocked
```

在Windows中：

```
aws rds modify-activity-stream ^
  --resource-arn my-instance-ARN ^
  --audit-policy-state unlocked
```

下列範例說明 *my-instance* 執行個體。部分範例輸出結果顯示稽核政策已解鎖。

```
aws rds describe-db-instances --db-instance-identifier my-instance

{
  "DBInstances": [
    {
      ...
    }
  ]
}
```

```
    "Engine": "oracle-ee",
    ...
    "ActivityStreamStatus": "started",
    "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
    "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
    "ActivityStreamMode": "async",
    "ActivityStreamEngineNativeAuditFieldsIncluded": true,
    "ActivityStreamPolicyStatus": "unlocked",
    ...
  }
]
}
```

RDS API

若要修改資料庫活動串流的原則狀態，請使用 [ModifyActivityStream](#) 作業。

使用以下參數呼叫動作：

- AuditPolicyState
- ResourceArn

取得資料庫活動串流的狀態

您可以使用主控台或 AWS CLI 來取得 Amazon RDS 資料庫執行個體的活動串流狀態。

主控台

取得資料庫活動串流的狀態

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇 (資料庫叢集)DB instance (資料庫執行個體) 連結。
3. 選擇 Configuration (組態) 標籤，然後確認 Database activity stream (資料庫活動串流) 的狀態。

AWS CLI

您可以取得資料庫執行個體的活動串流組態，作為對 [describe-db-instances](#) CLI 請求的回應。

以下範例描述 *my-instance*。

```
aws rds --region my-region describe-db-instances --db-instance-identifier my-db
```

JSON 回應如以下範例所示。下列欄位會顯示：

- ActivityStreamKinesisStreamName
- ActivityStreamKmsKeyId
- ActivityStreamStatus
- ActivityStreamMode
- ActivityStreamPolicyStatus

```
{
  "DBInstances": [
    {
      ...
      "Engine": "oracle-ee",
      ...
      "ActivityStreamStatus": "starting",
      "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
      "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
      "ActivityStreamMode": "async",
      "ActivityStreamEngineNativeAuditFieldsIncluded": true,
      "ActivityStreamPolicyStatus": "locked",
      ...
    }
  ]
}
```

RDS API

您可以取得資料庫的活動串流組態，作為對 [DescribeDBInstances](#) 操作的回應。

停用資料庫活動串流

您可以使用主控台或 AWS CLI 來停止活動串流。

若您刪除 Amazon RDS 資料庫執行個體，則會停止活動串流，並自動刪除底層的 Amazon Kinesis 串流。

主控台

關閉活動串流

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要停止資料庫活動串流的資料庫。
4. 針對 Actions (動作)，選擇 Stop activity stream (停止活動串流)。Database Activity Stream (資料庫活動串流) 即會顯示。

- a. 選擇 Immediately (立即)。

在您選擇 Immediately (立即) 時，RDS 執行個體會立即重新啟動。如果您選擇 During the next maintenance window (下個維護時段期間)，則 RDS 執行個體不會立即重新啟動。在此情況下，直到下一個維護時段前，資料庫活動串流都不會被停用。

- b. 選擇 Continue (繼續)。

AWS CLI

若要停止資料庫活動串流，請使用 AWS CLI 指令設定資料庫執行個體 [stop-activity-stream](#)。使用 `--region` 參數來識別資料庫執行個體的 AWS 區域。`--apply-immediately` 為選用參數。

對於 Linux/macOS、或 Unix：

```
aws rds --region MY_REGION \  
  stop-activity-stream \  
  --resource-arn MY_DB_ARN \  
  --apply-immediately
```

在 Windows 中：

```
aws rds --region MY_REGION ^  
  stop-activity-stream ^  
  --resource-arn MY_DB_ARN ^  
  --apply-immediately
```

RDS API

若要停止資料庫活動串流，請使用該 [StopActivityStream](#) 作業設定資料庫執行個體。使用 `Region` 參數來識別資料庫執行個體的 AWS 區域。`ApplyImmediately` 為選用參數。

監控資料庫活動串流

資料庫活動串流會監控和報告活動。活動資料串流會收集並傳送至 Amazon Kinesis。從 Kinesis 中，您可以監視活動串流，或者其他服務和應用程式可以使用活動串流以供進一步分析。您可以使用 AWS CLI 命令或 RDS API 作業來尋找基礎 Kinesis 串流名稱。

Amazon RDS 會為您管理 Kinesis 串流，如下所示：

- Amazon RDS 會自動建立保留時間為 24 小時的 Kinesis 串流。
- 如有必要，Amazon RDS 可擴展 Kinesis 串流。
- 如果您停止資料庫活動串流或刪除資料庫執行個體，則 Amazon RDS 會刪除 Kinesis 串流。

系統會監控以下類別的活動並將其放在活動串流稽核日誌中：

- SQL 命令 – 所有 SQL 命令都經稽核，也是預備陳述式、內建函數和 PL/SQL 函數。對預存程序的呼叫會進行稽核。在儲存的程序或函數中發出的任何 SQL 語句也被稽核。
- 其他資料庫資訊 – 監控的活動包含完整的 SQL 陳述式、因 DML 命令而受影響的資料列數、經存取的物件和唯一的資料庫名稱。資料庫活動串流還會監控連結變數和已儲存的程序參數。

Important

活動資料串流稽核記錄中會顯示每個陳述式的完整 SQL 文字，包括任何敏感資料。但是，如果 Oracle 可以從內容 (例如下列 SQL 陳述式) 判斷資料庫使用者密碼，則該密碼將會被修訂。

```
ALTER ROLE role-name WITH password
```

- 連線資訊 – 監控的活動包含工作階段和網路資訊、伺服器程序 ID 和結束代碼。

如果活動串流在監控資料庫執行個體時失敗，系統會透過 RDS 事件來通知您。

主題

- [透過 Kinesis 存取活動串流](#)
- [稽核日誌內容和範例](#)
- [databaseActivityEvent 列出數組](#)
- [使用 AWS SDK 處理資料庫活動串流](#)

透過 Kinesis 存取活動串流

在您啟用資料庫的活動串流時，系統就會為您建立 Kinesis 串流。透過 Kinesis，您就可以即時監控資料庫活動。若要進一步分析資料庫活動，您可以將 Kinesis 串流連線至消費者應用程式。您也可以將串流連線至法規遵循管理應用程式，例如 IBM 安全衛士或 Imperva 的 SecureSphere 資料庫稽核與保護。

您可以從 RDS 主控台或 Kinesis 主控台存取 Kinesis 串流。

使用 RDS 主控台從 Kinesis 存取活動串流

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇已在其上啟動活動串流的 Amazon RDS 資料庫執行個體。
4. 選擇 Configuration (組態)。
5. 在 Database activity stream (資料庫活動串流) 下，選擇 Kinesis stream (Kinesis 串流) 下的連結。
6. 在 Kinesis 主控台中，選擇 Monitoring (監控) 來開始觀察資料庫活動。

使用 Kinesis 主控台從 Kinesis 存取活動串流

1. 在 <https://console.aws.amazon.com/kinesis> 上開啟 Kinesis 主控台。
2. 從 Kinesis 串流清單中選擇活動串流。

活動串流的名稱包含字首 `aws-rds-das-db-`，後接資料庫的資源 ID。以下是範例。

```
aws-rds-das-db-NHV0V4PCLWHGF52NP
```

若要使用 Amazon RDS 主控台來尋找資料庫的資源 ID，請從資料庫清單中選擇您的資料庫執行個體，然後選擇 Configuration (組態) 索引標籤。

若 AWS CLI 要使用尋找活動串流的完整 Kinesis 串流名稱，請使用 CLI 要求並記下回應 `ActivityStreamKinesisStreamName` 中的值。

3. 選擇 Monitoring (監控) 來開始觀察資料庫活動。

如需使用 Amazon Kinesis 的詳細資訊，請參閱 [什麼是 Amazon Kinesis Data Streams ?](#)。

稽核日誌內容和範例

受監控的事件會以 JSON 字串的形式在資料庫活動串流中顯示。此結構包含 JSON 物件，內含的 DatabaseActivityMonitoringRecord 會依序包含活動事件的 databaseActivityEventList 陣列。

主題

- [活動串流的稽核記錄範例](#)
- [DatabaseActivityMonitoringRecords物件](#)
- [databaseActivityEvents 對象](#)

活動串流的稽核記錄範例

以下是活動事件記錄的範例解密 JSON 稽核日誌。

Example a CONNECT SQL 陳述式 的活動事件記錄

以下活動事件記錄顯示 JDBC 精簡用戶端 (clientApplication) 使用 CONNECT SQL 陳述式 (command) 登入您的 Oracle 資料庫。

```
{
  "class": "Standard",
  "clientApplication": "JDBC Thin Client",
  "command": "LOGON",
  "commandText": null,
  "dbid": "0123456789",
  "databaseName": "ORCL",
  "dbProtocol": "oracle",
  "dbUserName": "TEST",
  "endTime": null,
  "errorMessage": null,
  "exitCode": 0,
  "logTime": "2021-01-15 00:15:36.233787",
  "netProtocol": "tcp",
  "objectName": null,
  "objectType": null,
  "paramList": [],
  "pid": 17904,
  "remoteHost": "123.456.789.012",
  "remotePort": "25440",
  "rowCount": null,
```

```
"serverHost": "987.654.321.098",
"serverType": "oracle",
"serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
"serviceName": "oracle-ee",
"sessionId": 987654321,
"startTime": null,
"statementId": 1,
"substatementId": null,
"transactionId": "0000000000000000",
"engineNativeAuditFields": {
  "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
  "FGA_POLICY_NAME": null,
  "DV_OBJECT_STATUS": null,
  "SYSTEM_PRIVILEGE_USED": "CREATE SESSION",
  "OLS_LABEL_COMPONENT_TYPE": null,
  "XS_SESSIONID": null,
  "ADDITIONAL_INFO": null,
  "INSTANCE_ID": 1,
  "DBID": 123456789
  "DV_COMMENT": null,
  "RMAN_SESSION_STAMP": null,
  "NEW_NAME": null,
  "DV_ACTION_NAME": null,
  "OLS_PROGRAM_UNIT_NAME": null,
  "OLS_STRING_LABEL": null,
  "RMAN_SESSION_RECID": null,
  "OBJECT_PRIVILEGES": null,
  "OLS_OLD_VALUE": null,
  "XS_TARGET_PRINCIPAL_NAME": null,
  "XS_NS_ATTRIBUTE": null,
  "XS_NS_NAME": null,
  "DBLINK_INFO": null,
  "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT_ADDRESS\u003d((ADDRESS
\u003d(PROTOCOL\u003dtcp)(HOST\u003d205.251.233.183)(PORT\u003d25440)))));",
  "OBJECT_EDITION": null,
  "OLS_PRIVILEGES_GRANTED": null,
  "EXCLUDED_USER": null,
  "DV_ACTION_OBJECT_NAME": null,
  "OLS_LABEL_COMPONENT_NAME": null,
  "EXCLUDED_SCHEMA": null,
  "DP_TEXT_PARAMETERS1": null,
  "XS_USER_NAME": null,
  "XS_ENABLED_ROLE": null,
  "XS_NS_ATTRIBUTE_NEW_VAL": null,
```



```
"DIRECT_PATH_NUM_COLUMNS_LOADED": null,  
"AUDIT_OPTION": null,  
"DV_EXTENDED_ACTION_CODE": null,  
"XS_PACKAGE_NAME": null,  
"OLS_NEW_VALUE": null,  
"DV_RETURN_CODE": null,  
"XS_CALLBACK_EVENT_TYPE": null,  
"USERHOST": "a1b2c3d4e5f6.amazon.com",  
"GLOBAL_USERID": null,  
"CLIENT_IDENTIFIER": null,  
"RMAN_OPERATION": null,  
"TERMINAL": "unknown",  
"OS_USERNAME": "sumepate",  
"OLS_MAX_READ_LABEL": null,  
"XS_PROXY_USER_NAME": null,  
"XS_DATASEC_POLICY_NAME": null,  
"DV_FACTOR_CONTEXT": null,  
"OLS_MAX_WRITE_LABEL": null,  
"OLS_PARENT_GROUP_NAME": null,  
"EXCLUDED_OBJECT": null,  
"DV_RULE_SET_NAME": null,  
"EXTERNAL_USERID": null,  
"EXECUTION_ID": null,  
"ROLE": null,  
"PROXY_SESSIONID": 0,  
"DP_BOOLEAN_PARAMETERS1": null,  
"OLS_POLICY_NAME": null,  
"OLS_GRANTEE": null,  
"OLS_MIN_WRITE_LABEL": null,  
"APPLICATION_CONTEXTS": null,  
"XS_SCHEMA_NAME": null,  
"DV_GRANTEE": null,  
"XS_COOKIE": null,  
"DBPROXY_USERNAME": null,  
"DV_ACTION_CODE": null,  
"OLS_PRIVILEGES_USED": null,  
"RMAN_DEVICE_TYPE": null,  
"XS_NS_ATTRIBUTE_OLD_VAL": null,  
"TARGET_USER": null,  
"XS_ENTITY_TYPE": null,  
"ENTRY_ID": 1,  
"XS_PROCEDURE_NAME": null,  
"XS_INACTIVITY_TIMEOUT": null,  
"RMAN_OBJECT_TYPE": null,
```

```
"SYSTEM_PRIVILEGE": null,
"NEW_SCHEMA": null,
"SCN": 5124715
}
}
```

下列活動事件記錄顯示 SQL Server 資料庫的登入失敗。

```
{
  "type": "DatabaseActivityMonitoringRecord",
  "clusterId": "",
  "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ7703Q",
  "databaseActivityEventList": [
    {
      "class": "LOGIN",
      "clientApplication": "Microsoft SQL Server Management Studio",
      "command": "LOGIN FAILED",
      "commandText": "Login failed for user 'test'. Reason: Password did not
match that for the login provided. [CLIENT: local-machine]",
      "databaseName": "",
      "dbProtocol": "SQLSERVER",
      "dbUserName": "test",
      "endTime": null,
      "errorMessage": null,
      "exitCode": 0,
      "logTime": "2022-10-06 21:34:42.7113072+00",
      "netProtocol": null,
      "objectName": "",
      "objectType": "LOGIN",
      "paramList": null,
      "pid": null,
      "remoteHost": "local machine",
      "remotePort": null,
      "rowCount": 0,
      "serverHost": "172.31.30.159",
      "serverType": "SQLSERVER",
      "serverVersion": "15.00.4073.23.v1.R1",
      "serviceName": "sqlserver-ee",
      "sessionId": 0,
      "startTime": null,
      "statementId": "0x1eb0d1808d34a94b9d3dcf5432750f02",
      "substatementId": 1,
      "transactionId": "0",
    }
  ]
}
```

```

    "type": "record",
    "engineNativeAuditFields": {
      "target_database_principal_id": 0,
      "target_server_principal_id": 0,
      "target_database_principal_name": "",
      "server_principal_id": 0,
      "user_defined_information": "",
      "response_rows": 0,
      "database_principal_name": "",
      "target_server_principal_name": "",
      "schema_name": "",
      "is_column_permission": false,
      "object_id": 0,
      "server_instance_name": "EC2AMAZ-NFUJJN0",
      "target_server_principal_sid": null,
      "additional_information": "<action_info xmlns=\"http://
schemas.microsoft.com/sqlserver/2008/sqlaudit_data\"><pooled_connection>0</
pooled_connection><error>0x00004818</error><state>8</state><address>local machine</
address><PasswordFirstNibbleHash>B</PasswordFirstNibbleHash></action_info\"-->,
      "duration_milliseconds": 0,
      "permission_bitmask": "0x00000000000000000000000000000000",
      "data_sensitivity_information": "",
      "session_server_principal_name": "",
      "connection_id": "98B4F537-0F82-49E3-AB08-B9D33B5893EF",
      "audit_schema_version": 1,
      "database_principal_id": 0,
      "server_principal_sid": null,
      "user_defined_event_id": 0,
      "host_name": "EC2AMAZ-NFUJJN0"
    }
  }
]
}

```

Note

若未啟用資料庫活動串流，則 JSON 文件中的最後一個欄位為 "engineNativeAuditFields": { }。

Example CREATE TABLE 陳述式的活動事件記錄

以下是 您 Oracle 資料庫的 CREATE TABLE 事件範例。

```
{
  "class": "Standard",
  "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
  "command": "CREATE TABLE",
  "commandText": "CREATE TABLE persons(\n  person_id NUMBER GENERATED BY DEFAULT AS
IDENTITY,\n  first_name VARCHAR2(50) NOT NULL,\n  last_name VARCHAR2(50) NOT NULL,\n\n  PRIMARY KEY(person_id)\n)",
  "dbid": "0123456789",
  "databaseName": "ORCL",
  "dbProtocol": "oracle",
  "dbUserName": "TEST",
  "endTime": null,
  "errorMessage": null,
  "exitCode": 0,
  "logTime": "2021-01-15 00:22:49.535239",
  "netProtocol": "beq",
  "objectName": "PERSONS",
  "objectType": "TEST",
  "paramList": [],
  "pid": 17687,
  "remoteHost": "123.456.789.0",
  "remotePort": null,
  "rowCount": null,
  "serverHost": "987.654.321.01",
  "serverType": "oracle",
  "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
  "serviceName": "oracle-ee",
  "sessionId": 1234567890,
  "startTime": null,
  "statementId": 43,
  "substatementId": null,
  "transactionId": "090011007F0D0000",
  "engineNativeAuditFields": {
    "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
    "FGA_POLICY_NAME": null,
    "DV_OBJECT_STATUS": null,
    "SYSTEM_PRIVILEGE_USED": "CREATE SEQUENCE, CREATE TABLE",
    "OLS_LABEL_COMPONENT_TYPE": null,
    "XS_SESSIONID": null,
    "ADDITIONAL_INFO": null,
    "INSTANCE_ID": 1,
    "DV_COMMENT": null,
    "RMAN_SESSION_STAMP": null,
  }
}
```

```
"NEW_NAME": null,
"DV_ACTION_NAME": null,
"OLS_PROGRAM_UNIT_NAME": null,
"OLS_STRING_LABEL": null,
"RMAN_SESSION_RECID": null,
"OBJECT_PRIVILEGES": null,
"OLS_OLD_VALUE": null,
"XS_TARGET_PRINCIPAL_NAME": null,
"XS_NS_ATTRIBUTE": null,
"XS_NS_NAME": null,
"DBLINK_INFO": null,
"AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT_ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)))";
"OBJECT_EDITION": null,
"OLS_PRIVILEGES_GRANTED": null,
"EXCLUDED_USER": null,
"DV_ACTION_OBJECT_NAME": null,
"OLS_LABEL_COMPONENT_NAME": null,
"EXCLUDED_SCHEMA": null,
"DP_TEXT_PARAMETERS1": null,
"XS_USER_NAME": null,
"XS_ENABLED_ROLE": null,
"XS_NS_ATTRIBUTE_NEW_VAL": null,
"DIRECT_PATH_NUM_COLUMNS_LOADED": null,
"AUDIT_OPTION": null,
"DV_EXTENDED_ACTION_CODE": null,
"XS_PACKAGE_NAME": null,
"OLS_NEW_VALUE": null,
"DV_RETURN_CODE": null,
"XS_CALLBACK_EVENT_TYPE": null,
"USERHOST": "ip-10-13-0-122",
"GLOBAL_USERID": null,
"CLIENT_IDENTIFIER": null,
"RMAN_OPERATION": null,
"TERMINAL": "pts/1",
"OS_USERNAME": "rdsdb",
"OLS_MAX_READ_LABEL": null,
"XS_PROXY_USER_NAME": null,
"XS_DATASEC_POLICY_NAME": null,
"DV_FACTOR_CONTEXT": null,
"OLS_MAX_WRITE_LABEL": null,
"OLS_PARENT_GROUP_NAME": null,
"EXCLUDED_OBJECT": null,
"DV_RULE_SET_NAME": null,
```



```
"dbProtocol": "SQLSERVER",
"dbUserName": "test",
"endTime": null,
"errorMessage": null,
"exitCode": 1,
"logTime": "2022-10-06 21:44:38.4120677+00",
"netProtocol": null,
"objectName": "dbo",
"objectType": "SCHEMA",
"paramList": null,
"pid": null,
"remoteHost": "local machine",
"remotePort": null,
"rowCount": 0,
"serverHost": "172.31.30.159",
"serverType": "SQLSERVER",
"serverVersion": "15.00.4073.23.v1.R1",
"serviceName": "sqlserver-ee",
"sessionId": 84,
"startTime": null,
"statementId": "0x5178d33d56e95e419558b9607158a5bd",
"substatementId": 1,
"transactionId": "4561864",
"type": "record",
"engineNativeAuditFields": {
  "target_database_principal_id": 0,
  "target_server_principal_id": 0,
  "target_database_principal_name": "",
  "server_principal_id": 2,
  "user_defined_information": "",
  "response_rows": 0,
  "database_principal_name": "dbo",
  "target_server_principal_name": "",
  "schema_name": "",
  "is_column_permission": false,
  "object_id": 1,
  "server_instance_name": "EC2AMAZ-NFUJJNO",
  "target_server_principal_sid": null,
  "additional_information": "",
  "duration_milliseconds": 0,
  "permission_bitmask": "0x00000000000000000000000000000000",
  "data_sensitivity_information": "",
  "session_server_principal_name": "test",
  "connection_id": "EE1FE3FD-EF2C-41FD-AF45-9051E0CD983A",
```

```

        "audit_schema_version": 1,
        "database_principal_id": 1,
        "server_principal_sid":
"0x01050000000000000515000000bdc2795e2d0717901ba6998cf4010000",
        "user_defined_event_id": 0,
        "host_name": "EC2AMAZ-NFUJJN0"
    }
}
]
}

```

Example SELECT 陳述式的活動事件記錄

下列範例顯示 Oracle 資料庫的 SELECT 事件。

```

{
  "class": "Standard",
  "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
  "command": "SELECT",
  "commandText": "select count(*) from persons",
  "databaseName": "1234567890",
  "dbProtocol": "oracle",
  "dbUserName": "TEST",
  "endTime": null,
  "errorMessage": null,
  "exitCode": 0,
  "logTime": "2021-01-15 00:25:18.850375",
  "netProtocol": "beq",
  "objectName": "PERSONS",
  "objectType": "TEST",
  "paramList": [],
  "pid": 17687,
  "remoteHost": "123.456.789.0",
  "remotePort": null,
  "rowCount": null,
  "serverHost": "987.654.321.09",
  "serverType": "oracle",
  "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
  "serviceName": "oracle-ee",
  "sessionId": 1080639707,
  "startTime": null,
  "statementId": 44,
  "substatementId": null,
  "transactionId": null,
}

```



```
"engineNativeAuditFields": {
  "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
  "FGA_POLICY_NAME": null,
  "DV_OBJECT_STATUS": null,
  "SYSTEM_PRIVILEGE_USED": null,
  "OLS_LABEL_COMPONENT_TYPE": null,
  "XS_SESSIONID": null,
  "ADDITIONAL_INFO": null,
  "INSTANCE_ID": 1,
  "DV_COMMENT": null,
  "RMAN_SESSION_STAMP": null,
  "NEW_NAME": null,
  "DV_ACTION_NAME": null,
  "OLS_PROGRAM_UNIT_NAME": null,
  "OLS_STRING_LABEL": null,
  "RMAN_SESSION_RECID": null,
  "OBJECT_PRIVILEGES": null,
  "OLS_OLD_VALUE": null,
  "XS_TARGET_PRINCIPAL_NAME": null,
  "XS_NS_ATTRIBUTE": null,
  "XS_NS_NAME": null,
  "DBLINK_INFO": null,
  "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)))";",
  "OBJECT_EDITION": null,
  "OLS_PRIVILEGES_GRANTED": null,
  "EXCLUDED_USER": null,
  "DV_ACTION_OBJECT_NAME": null,
  "OLS_LABEL_COMPONENT_NAME": null,
  "EXCLUDED_SCHEMA": null,
  "DP_TEXT_PARAMETERS1": null,
  "XS_USER_NAME": null,
  "XS_ENABLED_ROLE": null,
  "XS_NS_ATTRIBUTE_NEW_VAL": null,
  "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
  "AUDIT_OPTION": null,
  "DV_EXTENDED_ACTION_CODE": null,
  "XS_PACKAGE_NAME": null,
  "OLS_NEW_VALUE": null,
  "DV_RETURN_CODE": null,
  "XS_CALLBACK_EVENT_TYPE": null,
  "USERHOST": "ip-12-34-5-678",
  "GLOBAL_USERID": null,
  "CLIENT_IDENTIFIER": null,
```

```
"RMAN_OPERATION": null,
"TERMINAL": "pts/1",
"OS_USERNAME": "rdsdb",
"OLS_MAX_READ_LABEL": null,
"XS_PROXY_USER_NAME": null,
"XS_DATASEC_POLICY_NAME": null,
"DV_FACTOR_CONTEXT": null,
"OLS_MAX_WRITE_LABEL": null,
"OLS_PARENT_GROUP_NAME": null,
"EXCLUDED_OBJECT": null,
"DV_RULE_SET_NAME": null,
"EXTERNAL_USERID": null,
"EXECUTION_ID": null,
"ROLE": null,
"PROXY_SESSIONID": 0,
"DP_BOOLEAN_PARAMETERS1": null,
"OLS_POLICY_NAME": null,
"OLS_GRANTEE": null,
"OLS_MIN_WRITE_LABEL": null,
"APPLICATION_CONTEXTS": null,
"XS_SCHEMA_NAME": null,
"DV_GRANTEE": null,
"XS_COOKIE": null,
"DBPROXY_USERNAME": null,
"DV_ACTION_CODE": null,
"OLS_PRIVILEGES_USED": null,
"RMAN_DEVICE_TYPE": null,
"XS_NS_ATTRIBUTE_OLD_VAL": null,
"TARGET_USER": null,
"XS_ENTITY_TYPE": null,
"ENTRY_ID": 13,
"XS_PROCEDURE_NAME": null,
"XS_INACTIVITY_TIMEOUT": null,
"RMAN_OBJECT_TYPE": null,
"SYSTEM_PRIVILEGE": null,
"NEW_SCHEMA": null,
"SCN": 5136972
}
}
```

以下範例顯示 SQL Server 資料庫的 SELECT 事件。

```
{
```

```
"type": "DatabaseActivityMonitoringRecord",
"clusterId": "",
"instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ7703Q",
"databaseActivityEventList": [
  {
    "class": "TABLE",
    "clientApplication": "Microsoft SQL Server Management Studio - Query",
    "command": "SELECT",
    "commandText": "select * from [testDB].[dbo].[TestTable]",
    "databaseName": "testDB",
    "dbProtocol": "SQLSERVER",
    "dbUserName": "test",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 1,
    "logTime": "2022-10-06 21:24:59.9422268+00",
    "netProtocol": null,
    "objectName": "TestTable",
    "objectType": "TABLE",
    "paramList": null,
    "pid": null,
    "remoteHost": "local machine",
    "remotePort": null,
    "rowCount": 0,
    "serverHost": "172.31.30.159",
    "serverType": "SQLSERVER",
    "serverVersion": "15.00.4073.23.v1.R1",
    "serviceName": "sqlserver-ee",
    "sessionId": 62,
    "startTime": null,
    "statementId": "0x03baed90412f564fad640ebe51f89b99",
    "substatementId": 1,
    "transactionId": "4532935",
    "type": "record",
    "engineNativeAuditFields": {
      "target_database_principal_id": 0,
      "target_server_principal_id": 0,
      "target_database_principal_name": "",
      "server_principal_id": 2,
      "user_defined_information": "",
      "response_rows": 0,
      "database_principal_name": "dbo",
      "target_server_principal_name": "",
      "schema_name": "dbo",
```

```

        "is_column_permission": true,
        "object_id": 581577110,
        "server_instance_name": "EC2AMAZ-NFUJJN0",
        "target_server_principal_sid": null,
        "additional_information": "",
        "duration_milliseconds": 0,
        "permission_bitmask": "0x00000000000000000000000000000001",
        "data_sensitivity_information": "",
        "session_server_principal_name": "test",
        "connection_id": "AD3A5084-FB83-45C1-8334-E923459A8109",
        "audit_schema_version": 1,
        "database_principal_id": 1,
        "server_principal_sid":
"0x01050000000000000515000000bdc2795e2d0717901ba6998cf4010000",
        "user_defined_event_id": 0,
        "host_name": "EC2AMAZ-NFUJJN0"
    }
}
]
}

```

DatabaseActivityMonitoringRecords 物件

資料庫活動事件記錄位於 JSON 物件中，其中包含下列資訊。

JSON 欄位	資料類型	描述
type	string	JSON 記錄類型。值為 DatabaseActivityMonitoringRecords。
version	string	資料庫活動監控記錄的版本。Oracle 資料庫使用 1.3 版，SQL Server 則使用 1.4 版。這些引擎版本引入 engineNativeAuditFields JSON 物件。
databaseActivityEvents	string	包含活動事件的 JSON 物件。
金鑰	string	您用來解密 databaseActivityEvent清單 的加密金鑰

databaseActivityEvents 對象

databaseActivityEvents JSON 物件包含以下資訊。

JSON 記錄中的最上層欄位

稽核記錄檔中的每個事件都會包裝在 JSON 格式的記錄中。此記錄包含下列欄位。

type

此欄位永遠具有值 DatabaseActivityMonitoringRecords。

version

此欄位代表資料庫活動串流資料通訊協定或合約的版本。其會定義哪些欄位可用。

databaseActivityEvents

代表一或多個活動事件的加密字串。它被表示為一個 base64 位元組陣列。在您解密字串時，結果會是 JSON 格式的記錄，其中包含欄位，如本節範例所示。

金鑰

用來加密 databaseActivityEvents 字串的加密資料金鑰。這與 AWS KMS key 您啟動資料庫活動串流時所提供的相同。

下列範例顯示此記錄的格式。

```
{
  "type": "DatabaseActivityMonitoringRecords",
  "version": "1.3",
  "databaseActivityEvents": "encrypted audit records",
  "key": "encrypted key"
}
```

```
  "type": "DatabaseActivityMonitoringRecords",
  "version": "1.4",
  "databaseActivityEvents": "encrypted audit records",
  "key": "encrypted key"
```

採取下列步驟來解密 databaseActivityEvents 欄位的內容：

1. 使用您在啟動資料庫活動串流時提供的 KMS 金鑰，解密 key JSON 欄位中的值。這麼做會以純文字傳回資料加密金鑰。
2. Base64 解碼 databaseActivityEvents JSON 欄位中的值，以取得稽核承載的二進位格式的加密文字。
3. 使用您在第一個步驟中解碼的資料加密金鑰來解密二進位密文。
4. 解壓縮已解密的承載。
 - 加密的承載在 databaseActivityEvents 欄位。
 - 該 databaseActivityEventList 欄位包含稽核記錄的陣列。陣列中的 type 欄位可以是 record 或 heartbeat。

稽核日誌活動事件記錄是包含以下資訊的 JSON 物件。

JSON 欄位	資料類型	描述
type	string	JSON 記錄類型。值為 DatabaseActivityMonitoringRecord 。
instanceId	string	資料庫執行個體資源識別符。它對應於資料庫執行個體屬性 DbiResourceId 。
databaseActivityEvent清單	string	活動稽核記錄或活動訊號訊息的陣列。

databaseActivityEvent列出數組

稽核日誌承載是加密的 databaseActivityEventList JSON 陣列。以下資料表列出稽核記錄中已解密 DatabaseActivityEventList 陣列中，每個活動事件的欄位 (按英文字母順序列出)。

在 Oracle 資料庫中啟用統一稽核時，稽核記錄會填入此新稽核權杖中。此 UNIFIED_AUDIT_TRAIL 檢視會從稽核權杖擷取稽核記錄，以表格形式顯示稽核記錄。在您開始資料庫活動串流時，UNIFIED_AUDIT_TRAIL 中的欄位會映射至 databaseActivityEventList 陣列。

Important

事件結構可能會改變。Amazon RDS 可能會在未來將新的欄位新增至活動事件。在剖析 JSON 資料的應用程式中，請確定程式碼可以忽略，或針對未知欄位名稱採取適當的動作。

databaseActivityEvent 列出適用於甲骨文的亞馬遜 RDS 字段

欄位	資料類型	來源	描述
class	string	UNIFIED_AUDIT_TRAIL 中的 AUDIT_TYPE 欄位	<p>活動事件的類別。這對應於 AUDIT_TYPE 欄位中的 UNIFIED_AUDIT_TRAIL 檢視。Amazon RDS for Oracle 的有效值如下：</p> <ul style="list-style-type: none"> • Standard • FineGrainedAudit • XS • Database Vault • Label Security • RMAN_AUDIT • Datapump • Direct path API <p>如需更多資訊，請參閱 Oracle 文件中的 UNIFIED_AUDIT_TRAIL。</p>
clientApplication	string	CLIENT_PROGRAM_NAME 中的 UNIFIED_AUDIT_TRAIL	用戶端報告用來連接的應用程式。用戶端不需提供此資訊，因此此值可以是 Null。範例值為 JDBC Thin Client。
command	string	UNIFIED_AUDIT_TRAIL 中的 ACTION_NAME 欄位	由使用者執行的動作名稱。若要瞭解完整

欄位	資料類型	來源	描述
			動作，請讀取命令名稱和 AUDIT_TYPE 值。範例值為 ALTER DATABASE。
commandText	string	UNIFIED_AUDIT_TRAIL 中的 SQL_TEXT 欄位	與事件相關聯的 SQL 陳述式。範例值為 ALTER DATABASE BEGIN BACKUP。
databaseName	string	V\$DATABASE 中的 NAME 欄位	資料庫的名稱。
dbid	數字	UNIFIED_AUDIT_TRAIL 中的 DBID 欄位	資料庫的數字識別符。範例值為 1559204751 。
dbProtocol	string	N/A	資料庫通訊協定。在此測試版中，值為 oracle。
dbUserName	string	UNIFIED_AUDIT_TRAIL 中的 DBUSERNAME 欄位	動作已受稽核的資料庫使用者名稱。範例值為 RDSADMIN。
endTime	string	N/A	此欄位不會用於 RDS for Oracle 且一律為空。

欄位	資料類型	來源	描述
engineNativeAuditFields	物件	UNIFIED_AUDIT_TRAIL	<p>根據預設，此物件為空。在您使用 <code>--engine-native-audit-fields-included</code> 選項開始活動串流時，此物件會包含下列欄位及其值：</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre> ADDITIONAL_INFO APPLICATION _CONTEXTS AUDIT_OPTION AUTHENTICATIO N_TYPE CLIENT_IDENTIFIE R CURRENT_USER DBLINK_INFO DBPROXY_USERNAME DIRECT_PATH_NU M_COLUMNS_LOADED DP_BOOLEAN _PARAMETERS1 DP_TEXT_PARAME TERS1 DV_ACTION_CODE DV_ACTION_NAME DV_ACTION_OBJECT_N AME DV_COMMENT DV_EXTENDED_ ACTION_CODE DV_FACTOR_CONTEXT DV GRANTEE DV_OBJECT_STATUS DV_RETURN_CODE DV_RULE_SET_NAME </pre> </div>

欄位	資料 類型	來源	描述
			ENTRY_ID EXCLUDED_OBJECT EXCLUDED_SCHEMA EXCLUDED_USER EXECUTION_ID EXTERNAL_USERID FGA_POLICY_NAME GLOBAL_USERID INSTANCE_ID KSACL_SER VICE_NAME KSACL_SOURCE_LOCA TION KSACL_USER_NAME NEW_NAME NEW_SCHEMA OBJECT_EDITION OBJECT_PRIVILEGES OLS GRANTEE OLS_LABEL_COM PONENT_NAME OLS_LABEL_COMPO NENT_TYPE OLS_MAX_READ_LABEL OLS_MAX_WRITE_LA BEL OLS_MIN_WRITE_ LABEL OLS_NEW_VALUE OLS_OLD_VALUE OLS_PARENT_ GROUP_NAME OLS_POLICY_NAME OLS_PRIVILEGES_GRA NTED OLS_PRIVILEGE S_USED

欄位	資料 類型	來源	描述
			OLS_PROGRAM _UNIT_NAME OLS_STRING_LABEL OS_USERNAME PROTOCOL_ACTIO N_NAME PROTOCOL_MESSAGE PROTOCOL_RET URN_CODE PROTOCOL_SESSION_I D PROTOCOL_USERHOST PROXY_SESSIONID RLS_INFO RMAN_DEVICE_TYPE RMAN_OBJECT_TYPE RMAN_OPERATION RMAN_SESSION_RECID RMAN_SESSION_STAMP ROLE SCN SYSTEM_PRIVILEGE SYSTEM_PRIVIL EGE_USED TARGET_USER TERMINAL UNIFIED_AUDIT_P OLICIES USERHOST XS_CALLBAC K_EVENT_TYPE XS_COOKIE XS_DATASEC_PO LICY_NAME XS_ENABLED_ROLE XS_ENTITY_TYPE XS_INACTIVITY _TIMEOUT

欄位	資料類型	來源	描述
			<p>XS_NS_ATTRIBUTE XS_NS_ATTRI BUTE_NEW_VAL XS_NS_ATTRIBUT E_OLD_VAL XS_NS_NAME XS_PACKAGE_NAME XS_PROCEDURE_NAME XS_PROXY_USER_NAME XS_SCHEMA_NAME XS_SESSIONID XS_TARGET_PRINC IPAL_NAME XS_USER_NAME</p> <p>如需更多資訊，請參閱 Oracle 資料庫文件中的 UNIFIED_AUDIT_TRAI L。</p>
errorMessage	string	N/A	此欄位不會用於 RDS for Oracle 且一律為空。
exitCode	數字	UNIFIED_AUDIT_TRAIL 中的 RETURN_CODE 欄位	因動作而產生的 Oracle 資料庫錯誤代碼。若動作成功，則值為 0。
logTime	string	UNIFIED_AUDIT_TRAIL 中的 EVENT_TIMESTAMP_UTC 欄位	建立稽核權杖項目的時間戳記。範例值為 2020-11-27 06:56:14.981404 。
netProtocol	string	UNIFIED_AUDIT_TRAIL 中的 AUTHENTICATION_TYPE 欄位	網路通訊協定。範例值為 TCP。

欄位	資料 類型	來源	描述
objectName	string	UNIFIED_AUDIT_TRAIL 中的 OBJECT_NAME 欄位	受動作影響的物件名稱。 範例值為 employees 。
objectType	string	UNIFIED_AUDIT_TRAIL 中的 OBJECT_SCHEMA 欄位	受動作影響的物件結構描 述名稱。範例值為 hr。
paramList	清 單	UNIFIED_AUDIT_TRAIL 中的 SQL_BINDS 欄位	與 SQL_TEXT 相關聯的 綁定變數清單 (如有)。 範例值為 parameter _1,parameter_2 。
pid	數 字	UNIFIED_AUDIT_TRAIL 中的 OS_PROCESS 欄位	Oracle 資料庫程序的作業 系統程序識別符。範例值 為 22396。
remoteHost	string	UNIFIED_AUDIT_TRAI L 中的 AUTHENTIC ATION_TYPE 欄位	在產生工作階段後的用戶 端 IP 地址或主機名稱。 範例值為 123.456.7 89.123 。
remotePort	string	UNIFIED_AUDIT_TRAI L 中的 AUTHENTIC ATION_TYPE 欄位	用戶端連接埠號碼。 Oracle 資料庫環境中的典 型值為 1521。
rowCount	數 字	N/A	此欄位不會用於 RDS for Oracle 且一律為空。
serverHost	string	資料庫主機	資料庫伺服器主機的 IP 地址。範例值為 123.456.789.123 。
serverType	string	N/A	資料庫伺服器類型。值一 律為 ORACLE。

欄位	資料類型	來源	描述
serverVersion	string	資料庫主機	Amazon RDS for Oracle 版本，發布更新 (RU)，以及發布更新修訂版 (RUR)。範例值為 19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3 。
serviceName	string	資料庫主機	服務的名稱。範例值為 oracle-ee 。
sessionId	數字	UNIFIED_AUDIT_TRAIL 中的 SESSIONID 欄位	稽核的工作階段識別符。例如，1894327130 。
startTime	string	N/A	此欄位不會用於 RDS for Oracle 且一律為空。
statementId	數字	UNIFIED_AUDIT_TRAIL 中的 STATEMENT_ID 欄位	每個陳述式執行的數字 ID。陳述式可能會引起許多動作。範例值為 142197。
substatementId	N/A	N/A	此欄位不會用於 RDS for Oracle 且一律為空。
transactionId	string	UNIFIED_AUDIT_TRAIL 中的 TRANSACTION_ID 欄位	其中物件已修改的交易識別符。範例值為 02000800D5030000 。

databaseActivityEvent 列出 Amazon RDS for SQL Server 欄位

欄位	資料類型	來源	描述
class	string	sys.fn_get_audit_file.class_type 對應至 sys.dm_audit_class_type_map.class_type_desc	活動事件的類別。如需詳細資訊，請參閱 Microsoft 文件中的 SQL Server 稽核 (資料庫引擎) 。
clientApplication	string	sys.fn_get_audit_file.application_name	根據用戶端報告，用戶端連線的應用程式 (SQL Server 第 14 版及更高版本)。此欄位在 SQL Server 第 13 版為空值。
command	string	sys.fn_get_audit_file.action_id 對應至 sys.dm_audit_actions.name	SQL 陳述式的一般類別。此欄位的值取決於該類別的值。
commandText	string	sys.fn_get_audit_file.statement	此欄位會顯示該 SQL 陳述式。
databaseName	string	sys.fn_get_audit_file.database_name	資料庫的名稱。
dbProtocol	string	N/A	資料庫通訊協定。此值為 SQLSERVER 。
dbUserName	string	sys.fn_get_audit_file.server_principal_name	用戶端驗證所用的資料庫使用者。
endTime	string	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
engineNativeAuditFields	object	未列於此欄位的 sys.fn_get_audit_file 中各欄。	根據預設，此物件為空。使用 --engine-native-audit-fields-included 選項開啟活動串流時，此物件會包含此

欄位	資料 類型	來源	描述
			JSON 對應不會回傳的其他原生引擎稽核欄位。
errorMessage	string	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
exitCode	integer	sys.fn_get_audit_file.succeeded	<p>顯示是否成功啟動事件。此欄不能為空值。針對登入事件以外的所有事件，此欄會顯示許可檢查成功與否，但不會顯示操作是否成功。</p> <p>數值包含：</p> <ul style="list-style-type: none"> • 0 – Fail • 1 – Success
logTime	string	sys.fn_get_audit_file.event_time	SQL Server 記錄的事件時間戳記。
netProtocol	string	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
objectName	string	sys.fn_get_audit_file.object_name	SQL 陳述式在其中操作的資料庫物件名稱。
objectType	string	sys.fn_get_audit_file.class_type 對應至 sys.dm_audit_class_type_map.class_type_desc	SQL 陳述式在其中操作的資料庫物件類型。
paramList	string	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
pid	integer	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。

欄位	資料 類型	來源	描述
remoteHost	string	sys.fn_get_audit_file.client_ip	發出 SQL 陳述式 (SQL Server 第 14 版及更高版本) 之用戶端的 IP 地址或主機名稱。此欄位在 SQL Server 第 13 版為空值。
remotePort	integer	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
rowCount	integer	sys.fn_get_audit_file.affected_rows	SQL 陳述式影響的資料列數 (SQL Server 第 14 版及更高版本)。此欄位在 SQL Server 第 13 版中。
serverHost	string	資料庫主機	主機資料庫伺服器的 IP 地址。
serverType	string	N/A	資料庫伺服器類型。值為 SQLSERVER。
serverVersion	string	資料庫主機	資料庫伺服器版本，例如，SQL Server 2017 年的 15.00.4073.23.v1.R1。
serviceName	string	資料庫主機	服務的名稱。範例值為 sqlserver-ee。
sessionId	integer	sys.fn_get_audit_file.session_id	工作階段的唯一識別碼。
startTime	string	N/A	Amazon RDS for SQL Server 不使用此欄位，且其值為空值。
statementId	string	sys.fn_get_audit_file.sequence_group_id	唯一用戶端 SQL 陳述式的識別碼。每個產生出的事件的事件的識別碼都不同。範例值為 0x38eaf4156267184094bb82071aaab644。

欄位	資料 類型	來源	描述
substatementId	integer	sys.fn_get_audit_file.sequence_number	用來決定陳述式序號的識別碼。當大量記錄分成多個記錄時，此識別碼將有所助益。
transactionId	integer	sys.fn_get_audit_file.transaction_id	交易的識別碼。如果沒有任何作用中交易，則該值為零。
type	string	已產生的資料庫活動串流	事件的類型。值為 record 或 heartbeat 。

使用 AWS SDK 處理資料庫活動串流

您可以使用 AWS SDK 以程式設計方式處理活動串流。下列是功能完整的 Java 和 Python 範例，其示範使用資料庫活動串流記錄進行執行個體型啟用。

Java

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.Security;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.zip.GZIPInputStream;

import javax.crypto.Cipher;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
```

```
import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoInputStream;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;
import
    com.amazonaws.services.kinesis.clientlibrary.exceptions.InvalidStateException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ShutdownException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ThrottlingException;
import com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessor;
import
    com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorCheckpoint;
import
    com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorFactory;
import
    com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream;
import
    com.amazonaws.services.kinesis.clientlibrary.lib.worker.KinesisClientLibConfiguration;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.ShutdownReason;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker.Builder;
import com.amazonaws.services.kinesis.model.Record;
import com.amazonaws.services.kms.AWSKMS;
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.DecryptRequest;
import com.amazonaws.services.kms.model.DecryptResult;
import com.amazonaws.util.Base64;
import com.amazonaws.util.IOUtils;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.SerializedName;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class DemoConsumer {

    private static final String STREAM_NAME = "aws-rds-das-[instance-external-
resource-id]"; // aws-rds-das-db-ABCD123456
    private static final String APPLICATION_NAME = "AnyApplication"; //unique
application name for dynamo table generation that holds kinesis shard tracking
    private static final String AWS_ACCESS_KEY =
"[AWS_ACCESS_KEY_TO_ACCESS_KINESIS]";
    private static final String AWS_SECRET_KEY =
"[AWS_SECRET_KEY_TO_ACCESS_KINESIS]";
    private static final String RESOURCE_ID = "[external-resource-id]"; // db-
ABCD123456
```

```
private static final String REGION_NAME = "[region-name]"; //us-east-1, us-
east-2...
private static final BasicAWSCredentials CREDENTIALS = new
BasicAWSCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY);
private static final AWSStaticCredentialsProvider CREDENTIALS_PROVIDER = new
AWSStaticCredentialsProvider(CREDENTIALS);

private static final AwsCrypto CRYPTO = new AwsCrypto();
private static final AWSKMS KMS = AWSKMSClientBuilder.standard()
    .withRegion(REGION_NAME)
    .withCredentials(CREDENTIALS_PROVIDER).build();

class Activity {
    String type;
    String version;
    String databaseActivityEvents;
    String key;
}

class ActivityEvent {
    @SerializedName("class") String _class;
    String clientApplication;
    String command;
    String commandText;
    String databaseName;
    String dbProtocol;
    String dbUserName;
    String endTime;
    String errorMessage;
    String exitCode;
    String logTime;
    String netProtocol;
    String objectName;
    String objectType;
    List<String> paramList;
    String pid;
    String remoteHost;
    String remotePort;
    String rowCount;
    String serverHost;
    String serverType;
    String serverVersion;
    String serviceName;
    String sessionId;
```

```
String startTime;
String statementId;
String substatementId;
String transactionId;
String type;
}

class ActivityRecords {
    String type;
    String clusterId; // note that clusterId will contain an empty string on RDS
Oracle and RDS SQL Server
    String instanceId;
    List<ActivityEvent> databaseActivityEventList;
}

static class RecordProcessorFactory implements IRecordProcessorFactory {
    @Override
    public IRecordProcessor createProcessor() {
        return new RecordProcessor();
    }
}

static class RecordProcessor implements IRecordProcessor {

    private static final long BACKOFF_TIME_IN_MILLIS = 3000L;
    private static final int PROCESSING_RETRIES_MAX = 10;
    private static final long CHECKPOINT_INTERVAL_MILLIS = 60000L;
    private static final Gson GSON = new
GsonBuilder().serializeNulls().create();

    private static final Cipher CIPHER;
    static {
        Security.insertProviderAt(new BouncyCastleProvider(), 1);
        try {
            CIPHER = Cipher.getInstance("AES/GCM/NoPadding", "BC");
        } catch (NoSuchAlgorithmException | NoSuchPaddingException |
NoSuchProviderException e) {
            throw new ExceptionInInitializerError(e);
        }
    }

    private long nextCheckpointTimeInMillis;

    @Override
```

```
public void initialize(String shardId) {
}

@Override
public void processRecords(final List<Record> records, final
IRecordProcessorCheckpointier checkpointier) {
    for (final Record record : records) {
        processSingleBlob(record.getData());
    }

    if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
        checkpoint(checkpointer);
        nextCheckpointTimeInMillis = System.currentTimeMillis() +
CHECKPOINT_INTERVAL_MILLIS;
    }
}

@Override
public void shutdown(IRecordProcessorCheckpointier checkpointier,
ShutdownReason reason) {
    if (reason == ShutdownReason.TERMINATE) {
        checkpoint(checkpointer);
    }
}

private void processSingleBlob(final ByteBuffer bytes) {
    try {
        // JSON $Activity
        final Activity activity = GSON.fromJson(new String(bytes.array(),
StandardCharsets.UTF_8), Activity.class);

        // Base64.Decode
        final byte[] decoded =
Base64.decode(activity.databaseActivityEvents);
        final byte[] decodedDataKey = Base64.decode(activity.key);

        Map<String, String> context = new HashMap<>();
        context.put("aws:rds:db-id", RESOURCE_ID);

        // Decrypt
        final DecryptRequest decryptRequest = new DecryptRequest()
.withCiphertextBlob(ByteBuffer.wrap(decodedDataKey)).withEncryptionContext(context);
        final DecryptResult decryptResult = KMS.decrypt(decryptRequest);
    }
}
```

```

        final byte[] decrypted = decrypt(decoded,
getByteArray(decryptResult.getPlaintext()));

        // GZip Decompress
        final byte[] decompressed = decompress(decrypted);
        // JSON $ActivityRecords
        final ActivityRecords activityRecords = GSON.fromJson(new
String(decompressed, StandardCharsets.UTF_8), ActivityRecords.class);

        // Iterate through $ActivityEvents
        for (final ActivityEvent event :
activityRecords.databaseActivityEventList) {
            System.out.println(GSON.toJson(event));
        }
    } catch (Exception e) {
        // Handle error.
        e.printStackTrace();
    }
}

private static byte[] decompress(final byte[] src) throws IOException {
    ByteArrayInputStream byteArrayInputStream = new
ByteArrayInputStream(src);
    GZIPInputStream gzipInputStream = new
GZIPInputStream(byteArrayInputStream);
    return IOUtils.toByteArray(gzipInputStream);
}

private void checkpoint(IRecordProcessorCheckpointter checkpointer) {
    for (int i = 0; i < PROCESSING_RETRIES_MAX; i++) {
        try {
            checkpointer.checkpoint();
            break;
        } catch (ShutdownException se) {
            // Ignore checkpoint if the processor instance has been shutdown
(fail over).
            System.out.println("Caught shutdown exception, skipping
checkpoint." + se);
            break;
        } catch (ThrottlingException e) {
            // Backoff and re-attempt checkpoint upon transient failures
            if (i >= (PROCESSING_RETRIES_MAX - 1)) {
                System.out.println("Checkpoint failed after " + (i + 1) +
"attempts." + e);
            }
        }
    }
}

```

```
        break;
    } else {
        System.out.println("Transient issue when checkpointing -
attempt " + (i + 1) + " of " + PROCESSING_RETRIES_MAX + e);
    }
} catch (InvalidStateException e) {
    // This indicates an issue with the DynamoDB table (check for
table, provisioned IOPS).
    System.out.println("Cannot save checkpoint to the DynamoDB table
used by the Amazon Kinesis Client Library." + e);
    break;
}
}
try {
    Thread.sleep(BACKOFF_TIME_IN_MILLIS);
} catch (InterruptedException e) {
    System.out.println("Interrupted sleep" + e);
}
}
}

private static byte[] decrypt(final byte[] decoded, final byte[] decodedDataKey)
throws IOException {
    // Create a JCE master key provider using the random key and an AES-GCM
encryption algorithm
    final JceMasterKey masterKey = JceMasterKey.getInstance(new
SecretKeySpec(decodedDataKey, "AES"),
        "BC", "DataKey", "AES/GCM/NoPadding");
    try (final CryptoInputStream<JceMasterKey> decryptingStream =
CRYPTO.createDecryptingStream(masterKey, new ByteArrayInputStream(decoded));
        final ByteArrayOutputStream out = new ByteArrayOutputStream()) {
        IOUtils.copy(decryptingStream, out);
        return out.toByteArray();
    }
}

public static void main(String[] args) throws Exception {
    final String workerId = InetAddress.getLocalHost().getCanonicalHostName() +
":" + UUID.randomUUID();
    final KinesisClientLibConfiguration kinesisClientLibConfiguration =
        new KinesisClientLibConfiguration(APPLICATION_NAME, STREAM_NAME,
CREDENTIALS_PROVIDER, workerId);

    kinesisClientLibConfiguration.withInitialPositionInStream(InitialPositionInStream.LATEST);
}
```



```

kinesisClientLibConfiguration.withRegionName(REGION_NAME);
final Worker worker = new Builder()
    .recordProcessorFactory(new RecordProcessorFactory())
    .config(kinesisClientLibConfiguration)
    .build();

System.out.printf("Running %s to process stream %s as worker %s...\n",
APPLICATION_NAME, STREAM_NAME, workerId);

try {
    worker.run();
} catch (Throwable t) {
    System.err.println("Caught throwable while processing data.");
    t.printStackTrace();
    System.exit(1);
}
System.exit(0);
}

private static byte[] getByteArray(final ByteBuffer b) {
    byte[] byteArray = new byte[b.remaining()];
    b.get(byteArray);
    return byteArray;
}
}

```

Python

```

import base64
import json
import zlib
import aws_encryption_sdk
from aws_encryption_sdk import CommitmentPolicy
from aws_encryption_sdk.internal.crypto import WrappingKey
from aws_encryption_sdk.key_providers.raw import RawMasterKeyProvider
from aws_encryption_sdk.identifiers import WrappingAlgorithm, EncryptionKeyType
import boto3

REGION_NAME = '<region>' # us-east-1
RESOURCE_ID = '<external-resource-id>' # db-ABCD123456
STREAM_NAME = 'aws-rds-das-' + RESOURCE_ID # aws-rds-das-db-ABCD123456

```

```
enc_client =
    aws_encryption_sdk.EncryptionSDKClient(commitment_policy=CommitmentPolicy.FORBID_ENCRYPT_AL

class MyRawMasterKeyProvider(RawMasterKeyProvider):
    provider_id = "BC"

    def __new__(cls, *args, **kwargs):
        obj = super(RawMasterKeyProvider, cls).__new__(cls)
        return obj

    def __init__(self, plain_key):
        RawMasterKeyProvider.__init__(self)
        self.wrapping_key =
WrappingKey(wrapping_algorithm=WrappingAlgorithm.AES_256_GCM_IV12_TAG16_NO_PADDING,
            wrapping_key=plain_key,
wrapping_key_type=EncryptionKeyType.SYMMETRIC)

    def _get_raw_key(self, key_id):
        return self.wrapping_key

def decrypt_payload(payload, data_key):
    my_key_provider = MyRawMasterKeyProvider(data_key)
    my_key_provider.add_master_key("DataKey")
    decrypted_plaintext, header = enc_client.decrypt(
        source=payload,

materials_manager=aws_encryption_sdk.materials_managers.default.DefaultCryptoMaterialsManag
    return decrypted_plaintext

def decrypt_decompress(payload, key):
    decrypted = decrypt_payload(payload, key)
    return zlib.decompress(decrypted, zlib.MAX_WBITS + 16)

def main():
    session = boto3.session.Session()
    kms = session.client('kms', region_name=REGION_NAME)
    kinesis = session.client('kinesis', region_name=REGION_NAME)

    response = kinesis.describe_stream(StreamName=STREAM_NAME)
    shard_iters = []
    for shard in response['StreamDescription']['Shards']:
```

```
    shard_iter_response = kinesis.get_shard_iterator(StreamName=STREAM_NAME,
    ShardId=shard['ShardId'],

    ShardIteratorType='LATEST')
    shard_iters.append(shard_iter_response['ShardIterator'])

    while len(shard_iters) > 0:
        next_shard_iters = []
        for shard_iter in shard_iters:
            response = kinesis.get_records(ShardIterator=shard_iter, Limit=10000)
            for record in response['Records']:
                record_data = record['Data']
                record_data = json.loads(record_data)
                payload_decoded =
                base64.b64decode(record_data['databaseActivityEvents'])
                data_key_decoded = base64.b64decode(record_data['key'])
                data_key_decrypt_result =
                kms.decrypt(CiphertextBlob=data_key_decoded,

                EncryptionContext={'aws:rds:db-id': RESOURCE_ID})
                print (decrypt_decompress(payload_decoded,
                data_key_decrypt_result['Plaintext']))
                if 'NextShardIterator' in response:
                    next_shard_iters.append(response['NextShardIterator'])
            shard_iters = next_shard_iters

if __name__ == '__main__':
    main()
```

管理資料庫活動串流的存取

任何具備資料庫活動串流之適當 AWS Identity and Access Management (IAM) 角色權限的任何使用者，皆可建立、開始、停止和修改資料庫執行個體的活動串流設定。這些動作會包含在串流的稽核日誌中。為了達到最佳合規實務，我們建議您不要將這些權限提供給 DBA。

您可以使用 IAM 政策來設定資料庫活動串流的存取。如需更多有關 Amazon RDS 身分驗證的資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。如需建立 IAM 政策的詳細資訊，請參閱 [建立並使用 IAM 政策進行 IAM 資料庫存取](#)。

Example 允許資料庫活動串流設定的原則

若要提供使用者更精細的存取權來修改活動串流，請在 IAM 政策中使用服務特定操作內容金鑰 `rds:StartActivityStream` 與 `rds:StopActivityStream`。以下 IAM 政策範例會允許使用者或角色設定活動串流。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfigureActivityStreams",
      "Effect": "Allow",
      "Action": [
        "rds:StartActivityStream",
        "rds:StopActivityStream"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 允許資料庫活動串流開始的原則

以下 IAM 政策範例會允許使用者或角色開始活動串流。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowStartActivityStreams",
      "Effect": "Allow",
      "Action": "rds:StartActivityStream",
      "Resource": "*"
    }
  ]
}
```

Example 允許資料庫活動串流停用的原則

以下 IAM 政策範例會允許使用者或角色停止活動串流。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowStopActivityStreams",
    "Effect": "Allow",
    "Action": "rds:StopActivityStream",
    "Resource": "*"
  }
]
```

Example 拒絕資料庫活動串流開始的原則

以下 IAM 政策範例會防止使用者或角色開始活動串流。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyStartActivityStreams",
      "Effect": "Deny",
      "Action": "rds:StartActivityStream",
      "Resource": "*"
    }
  ]
}
```

Example 拒絕資料庫活動串流停用的原則

以下 IAM 政策範例會防止使用者或角色停止活動串流。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyStopActivityStreams",
      "Effect": "Deny",
      "Action": "rds:StopActivityStream",
      "Resource": "*"
    }
  ]
}
```


使用 Amazon RDS Custom

Amazon RDS Custom 會自動化資料庫管理任務和操作。RDS Custom 讓您以資料庫管理員身分存取和自訂資料庫環境和作業系統。使用 RDS Custom，您可以自訂以符合舊版、自訂和封裝應用程式的需求。

如需有關 RDS Custom 的最新網路研討會和部落格，請參閱 [Amazon RDS Custom 資源](#)。

主題

- [解決資料庫自訂的挑戰](#)
- [Amazon RDS Custom 的管理模型和優點](#)
- [Amazon RDS Custom 架構](#)
- [Amazon RDS Custom 的安全性](#)
- [使用 RDS Custom for Oracle](#)
- [使用 RDS Custom for SQL Server](#)

解決資料庫自訂的挑戰

Amazon RDS Custom 將 Amazon RDS 的優點帶到由於第三方應用程式需要的自訂而無法輕易移至全受管服務的市場。Amazon RDS Custom 可節省管理時間、耐用且可隨您的業務擴展。

如果您需要全面管理整個資料庫和作業系統 AWS，我們建議使用 Amazon RDS。如果您需要資料庫和基礎作業系統的管理權限讓相依應用程式可用，Amazon RDS Custom 是更好的選擇。如果您想要完整的管理責任，只需要受管運算服務，最好的選擇就是在 Amazon EC2 上自行管理您的商業資料庫。

為了提供受管服務體驗，Amazon RDS 並不會讓您存取基礎主機。Amazon RDS 也會將存取權限制在某些需要高階權限的系統程序和物件。不過，對於某些應用程式，您可能需要以具有特殊權限的作業系統 (OS) 使用者身分執行操作。

例如，您可能需要執行下列操作：

- 安裝自訂資料庫和作業系統修補程式和套件。
- 設定特定資料庫設定。
- 將檔案系統設定為直接與其應用程式共用檔案。

以前，如果您需要自訂應用程式，則必須在內部部署或 Amazon EC2 上部署資料庫。在此情況下，您必須承擔大部分或全部資料庫管理的責任，如下表所述。

功能	內部部署責任	Amazon EC2 責任	Amazon RDS 責任
應用程式最佳化	客戶	客戶	客戶
擴展	客戶	客戶	AWS
高可用性	客戶	客戶	AWS
資料庫備份	客戶	客戶	AWS
資料庫軟體修補	客戶	客戶	AWS
資料庫軟體安裝	客戶	客戶	AWS
作業系統修補	客戶	客戶	AWS
作業系統安裝	客戶	客戶	AWS
伺服器維護	客戶	AWS	AWS
硬體生命週期	客戶	AWS	AWS
電源、網路和冷卻	客戶	AWS	AWS

當您自行管理資料庫軟體時，您可以獲得更多的控制權，但也更容易出現使用者錯誤。例如，當您手動進行變更時，可能會意外造成應用程式停機。您可能會花費數小時檢查每項變更，以識別並修正問題。理想情況下，您想要一個受管資料庫服務，可自動化一般資料庫管理員任務，同時支援資料庫和基礎作業系統的特殊權限存取。

Amazon RDS Custom 的管理模型和優點

Amazon RDS Custom 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊版、自訂和封裝應用程式。RDS Custom 可自動化中資料庫的設定、操作和擴展，AWS 雲端同時授與您對資料庫和基礎作業系統的存取權。透過此存取權，您可以進行設定、安裝修補程式，以及啟用原生功能，以符合相依應用程式的需求。使用 RDS 自訂，您可以使用 AWS Management Console 或執行資料庫工作負載 AWS CLI。

RDS Custom 僅支持 Oracle 資料庫和 Microsoft SQL Server 資料庫引擎。

主題

- [RDS Custom 中共同的責任模型](#)
- [RDS Custom 中的支援周邊和不支援的組態](#)
- [RDS Custom 的主要優點](#)

RDS Custom 中共同的責任模型

在 RDS Custom 中，您可以使用 Amazon RDS 的受管功能，但管理主機和自訂作業系統的方式則與 Amazon EC2 中相同。除了在 Amazon RDS 中承擔的責任之外，您還承擔額外的資料庫管理責任。因此，與 Amazon RDS 相比，您對資料庫和資料庫執行個體的管理擁有更多的控制權，同時仍能受益於 RDS 自動化。

共同的責任表示：

1. 使用 RDS Custom 功能時，您擁有程序的一部分。

例如，在 RDS Custom for Oracle 中，您可以控制要使用哪些 Oracle 資料庫修補程式，以及何時將修補程式套用至資料庫執行個體。

2. 您必須負責確保對 RDS Custom 功能的任何自訂都能正常運作。

為了防止無效的自訂，RDS Custom 擁有在資料庫執行個體之外執行的自動化軟體。如果基礎 Amazon EC2 執行個體故障，RDS Custom 會自動嘗試透過重新開機或取代 EC2 執行個體來解決這些問題。使用者唯一可見的變更是新的 IP 地址。如需詳細資訊，請參閱 [Amazon RDS Custom 主機更換](#)。

下表詳細說明不同 RDS Custom 功能的共同責任模型。

功能	Amazon EC2 責任	Amazon RDS 責任	RDS Custom for Oracle 責任	RDS Custom for SQL Server 責任
應用程式最佳化	客戶	客戶	客戶	客戶
擴展	客戶	AWS	共同	共同
高可用性	客戶	AWS	客戶	AWS

功能	Amazon EC2 責任	Amazon RDS 責任	RDS Custom for Oracle 責任	RDS Custom for SQL Server 責任
資料庫備份	客戶	AWS	共同	AWS
資料庫軟體修補	客戶	AWS	共同	AWS 對於 RPEV, CEV 1 的客戶
資料庫軟體安裝	客戶	AWS	共同	AWS 對於 RPEV, CEV 1 的客戶
作業系統修補	客戶	AWS	客戶	AWS 對於 RPEV, CEV 1 的客戶
作業系統安裝	客戶	AWS	共同	AWS
伺服器維護	AWS	AWS	AWS	AWS
硬體生命週期	AWS	AWS	AWS	AWS
電源、網路和冷卻	AWS	AWS	AWS	AWS

¹ 自訂引擎版本 (CEV) 是資料庫版本和 Amazon 機器映像 (AMI) 的二進位磁碟區快照。RDS 提供的引擎版本 (RPEV) 是默認的 Amazon 機器映像 (AMI) 和 Microsoft SQL 服務器安裝。

您可以使用 Microsoft SQL Server 建立 RDS Custom 資料庫執行個體。在此案例中：

- 您有兩種授權模式可供選擇：已包含授權 (LI) 和自攜媒體 (BYOM)。
- 有了 LI，您就不需要另外購買 SQL 伺服器授權。AWS 持有 SQL 伺服器資料庫軟體的授權。
- 使用 BYOM，您可以提供並安裝自己的 Microsoft SQL Server 二進位檔案和授權。

您可以使用 Oracle 資料庫建立 RDS Custom 資料庫執行個體。在此情況下，請執行下列操作：

- 管理您自己的媒體。

使用 RDS Custom 時，您可以上傳自己的資料庫安裝檔案和修補程式。您可以從這些檔案建立自訂引擎版本 (CEV)。然後，您可以使用此 CEV 建立 RDS Custom 資料庫執行個體。

- 管理您自己的授權。

您可以自行攜帶 Oracle 資料庫授權，並自行管理授權。

RDS Custom 中的支援周邊和不支援的組態

RDS Custom 提供稱為支援周邊的監控功能。此功能可確保您的主機和資料庫環境設定正確。如果您進行的變更會導致資料庫執行個體超出支援周邊的範圍，RDS Custom 會將執行個體狀態變更為 `unsupported-configuration`，直到您手動修正組態問題。如需詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

RDS Custom 的主要優點

使用 RDS Custom，您可以執行下列操作：

- 將許多與 Amazon RDS 相同的管理任務自動化，包括下列各項：
 - 資料庫生命週期管理
 - 自動備份與 point-in-time 復原 (PITR)
 - 監視 RDS Custom DB 執行個體的健全狀況，並觀察基礎結構、作業系統和資料庫程序的變更。
 - 根據資料庫執行個體的中斷情況，通知或採取行動來修正問題
- 安裝第三方應用程式。

您可以安裝軟體來執行自訂應用程式和代理程式。由於您對主機具有特殊權限存取權，因此您可以修改檔案系統以支援舊版應用程式。

- 安裝自訂修補程式。

您可以在 RDS Custom 資料庫執行個體上套用自訂資料庫修補程式或修改作業系統套件。

- 先暫存內部部署資料庫，再將其移至全受管服務。

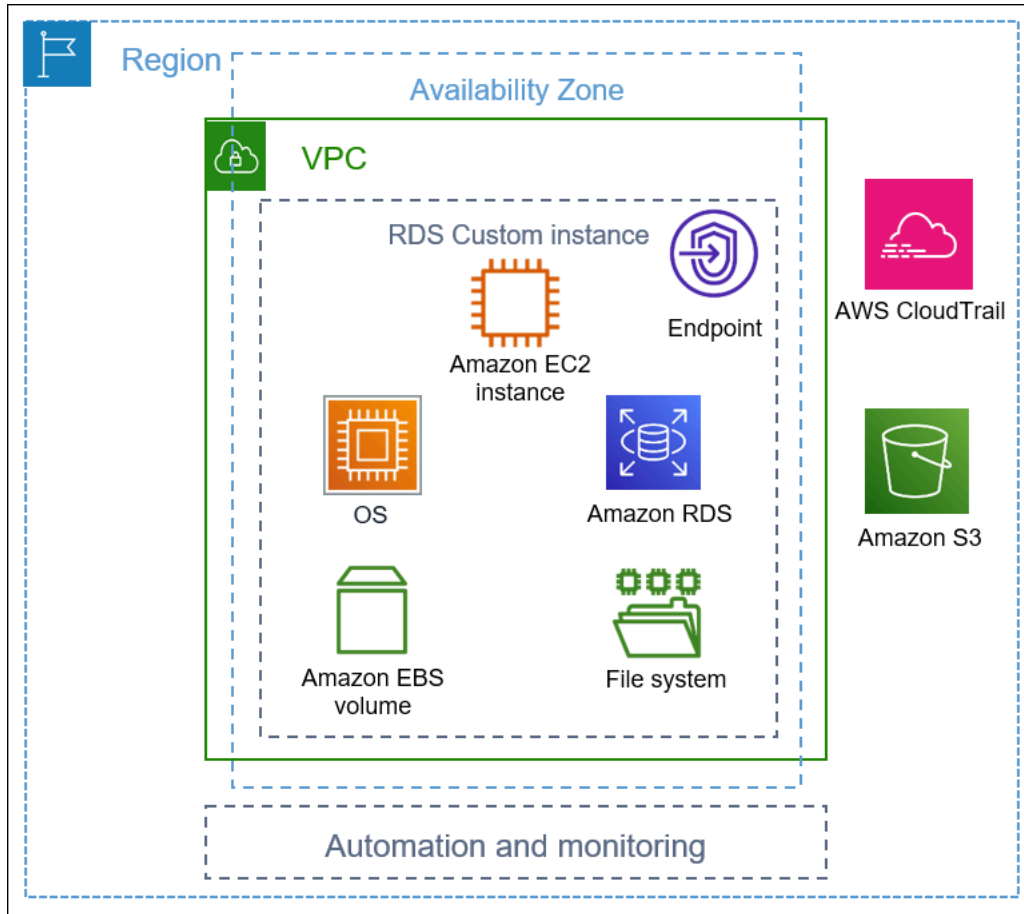
如果您管理自己的內部部署資料庫，則可以將資料庫依原樣暫存至 RDS Custom。熟悉雲端環境之後，您可以將資料庫遷移到全受管 Amazon RDS 資料庫執行個體。

- 建立您自己的自動化。

您可以建立、排程及執行報告、管理或診斷工具的自訂自動化指令碼。

Amazon RDS Custom 架構

Amazon RDS Custom 架構是根據 Amazon RDS，但是有重要差異。下列圖表顯示 RDS Custom 架構的關鍵元件。

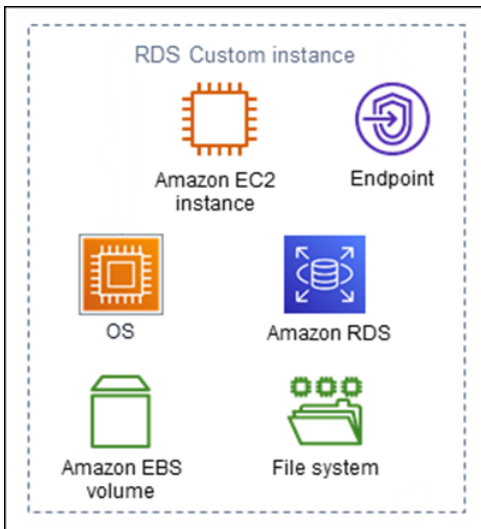


主題

- [VPC](#)
- [RDS Custom 自動化與監控](#)
- [Amazon S3](#)
- [AWS CloudTrail](#)

VPC

與 Amazon RDS 相同，您的 RDS Custom 資料庫執行個體位於 Virtual Private Cloud (VPC) 中。



RDS Custom 資料庫執行個體包含下列主要元件：

- Amazon EC2 執行個體
- 執行個體端點
- 安裝在 Amazon EC2 執行個體上的作業系統
- Amazon EBS 儲存體，其中包含任何其他檔案系統

RDS Custom 自動化與監控

RDS Custom 具有在資料庫執行個體外部執行的自動化軟體。此軟體會與資料庫執行個體上的代理程式以及整體 RDS Custom 環境中的其他元件進行通訊。

RDS Custom 監控和復原功能提供類似於 Amazon RDS 的功能。根據預設，RDS Custom 處於全自動化模式。自動化軟體具有下列主要責任：

- 收集指標及傳送通知
- 執行自動執行個體復原

RDS Custom 自動化的一個重要責任是回應 Amazon EC2 執行個體的問題。由於各種原因，主機可能會受損或無法連線。RDS Custom 透過重新開機或取代 Amazon EC2 執行個體來解決這些問題。

主題

- [Amazon RDS Custom 主機更換](#)
- [RDS Custom 支援周邊](#)

Amazon RDS Custom 主機更換

如果 Amazon EC2 主機受損，則 RDS Custom 會嘗試將其重新啟動。如果此工作失敗，RDS Custom 會使用 Amazon EC2 中包含的相同停止和啟動功能。取代主機時，客戶唯一可見的變更是新的公有 IP 地址。

主題

- [停止和啟動主機](#)
- [主機取代的影響](#)
- [Amazon EC2 主機的**最佳實務**](#)

停止和啟動主機

RDS Custom 會自動執行下列步驟，不需要使用者介入：

1. 停止 Amazon EC2 主機。

EC2 執行個體會執行正常的關機，然後停止執行。所有的 Amazon EBS 磁碟區仍會繼續連結至執行個體，其中的資料也會持續存在。所有存放在執行個體存放區磁碟區中的資料 (RDS Custom 不支援) 或主機電腦的 RAM 中的資料都會遺失。

[如需詳細資訊，請參閱 Amazon EC2 使用者指南中的停止和啟動執行個體。](#)

2. 啟動 Amazon EC2 主機。

EC2 執行個體會遷移至新的基礎主機硬體。在某些情況下，RDS Custom 資料庫執行個體會保留在原始主機上。

主機取代的影響

在 RDS Custom 中，您可以完全控制根裝置磁碟區和 Amazon EBS 儲存磁碟區。根磁碟區可能包含您不想遺失的重要資料和組態。

RDS Custom for Oracle 會在操作後保留所有資料庫和客戶資料，包括根磁碟區資料。不需要使用者介入。在 RDS Custom for SQL Server 上，會保留資料庫資料，但是 C: 磁碟機上的任何資料 (包括作業系統和客戶資料) 都會遺失。

在取代程序之後，Amazon EC2 主機會擁有新的公有 IP 地址。主機會保留下列項目：

- 執行個體 ID

- 私有 IP 地址
- 彈性 IP 地址
- 執行個體中繼資料
- 資料儲存磁碟區資料
- 根磁碟區資料 (在 RDS Custom for Oracle 上)

Amazon EC2 主機的最佳實務

Amazon EC2 主機取代功能涵蓋大多數 Amazon EC2 受損案例。建議您遵循下列最佳實務：

- 變更組態或作業系統之前，請先備份您的資料。如果根磁碟區或作業系統損毀，主機取代也無法修復它。您唯一的選擇是從數據庫快照還原或 point-in-time 恢復。
- 請勿手動停止或終止實體 Amazon EC2 主機。這兩個動作都會導致執行個體放在 RDS Custom 支援周邊外面。
- (RDS Custom for SQL Server) 如果您將其他磁碟區連接到 Amazon EC2 主機，請將其設定為在重新啟動時重新掛載。如果主機受損，RDS Custom 可能會自動停止和啟動主機。

RDS Custom 支援周邊

RDS Custom 提供額外的監控功能，稱為支援周邊。此額外監控可確保您的 RDS Custom 資料庫執行個體使用支援的 AWS 基礎結構、作業系統和資料庫。

支援周邊會檢查資料庫執行個體是否符合 [修正 RDS Custom for Oracle 中不支援的組態](#) 和 [修正 RDS Custom for SQL Server 中不支援的組態](#) 中所列的需求。如果有任何需求不符合，RDS Custom 會將您的資料庫執行個體視為不在支援周邊。

主題

- [RDS Custom 中不支援的組態](#)
- [對不支援的組態進行故障診斷](#)

RDS Custom 中不支援的組態

如果資料庫執行個體不在支援周邊，RDS Custom 會將資料庫執行個體的狀態變更為 unsupported-configuration，並傳送事件通知。修正組態問題之後，RDS Custom 會將資料庫執行個體狀態變更為 available。

當資料庫執行個體處於 `unsupported-configuration` 狀態時，以下為真：

- 您的資料庫可連線。但資料庫執行個體處於 `unsupported-configuration` 狀態時則例外，因為資料庫會意外關閉。
- 您無法修改資料庫執行個體。
- 您無法擷取資料庫快照。
- 不會建立自動備份。
- (僅適用 RDS Custom for SQL Server 資料庫執行個體) RDS Custom 不會更換故障的基礎 Amazon EC2 執行個體。如需主機更換的詳細資訊，請參閱 [Amazon RDS Custom 主機更換](#)。
- 您可以刪除資料庫執行個體，但大多數其他 RDS Custom API 操作無法使用。
- RDS 自訂會將重做日誌檔存檔並上傳到 Amazon S3，繼續支援 point-in-time 復原 (PITR)。處於 `unsupported-configuration` 狀態的 PITR 在以下方面有所不同：
 - PITR 可能需要很長的時間才能完全還原到新的 RDS Custom 資料庫執行個體。發生此情況是因為執行個體處於 `unsupported-configuration` 狀態時，您無法拍攝自動或手動快照。
 - PITR 必須從執行個體進入 `unsupported-configuration` 狀態之前擷取的最近快照開始，重新播放更多重做日誌。
 - 在某些情況下，資料庫執行個體處於 `unsupported-configuration` 狀態是因為您所做的變更使得封存的重做日誌檔案無法上傳。範例包括停止 EC2 執行個體、停止 RDS Custom 代理程式，以及卸離 EBS 磁碟區。在這類情況下，PITR 無法將資料庫執行個體還原至最新的可還原時間。

對不支援的組態進行故障診斷

RDS Custom 針對 `unsupported-configuration` 狀態提供了故障診斷指引。雖然部分指引同時適用於 RDS Custom for Oracle 和 RDS Custom for SQL Server，但其他指引則取決於您的資料庫引擎。如需引擎專屬的故障診斷資訊，請參閱下列主題：

- [修正 RDS Custom for Oracle 中不支援的組態](#)
- [修正 RDS Custom for SQL Server 中不支援的組態](#)

Amazon S3

若您使用 RDS Custom for Oracle，則會將安裝媒體上傳至使用者建立的 Amazon S3 儲存貯體。RDS Custom for Oracle 使用此儲存貯體中的媒體來建立自訂引擎版本 (CEV)。CEV 是資料庫版本和 Amazon Machine Image (AMI) 的二進位磁碟區快照。您可從 CEV 建立 RDS Custom 資料庫執行個體。如需詳細資訊，請參閱 [使用 Amazon RDS Custom for Oracle 的自訂引擎版本](#)。

若為 RDS Custom for Oracle 和 RDS Custom for SQL Server 兩者，RDS Custom 會自動建立以字串 `do-not-delete-rds-custom-` 為字首的 Amazon S3 儲存貯體。RDS Custom 使用 `do-not-delete-rds-custom-` S3 儲存貯體來儲存下列類型的檔案：

- AWS CloudTrail RDS 自訂所建立的追蹤記錄
- 支援周邊成品 (請參閱 [RDS Custom 支援周邊](#))
- 資料庫重做日誌檔案 (僅限 RDS Custom for Oracle)
- 交易日誌 (僅限 RDS Custom for SQL Server)
- 自訂引擎版本成品 (僅限 RDS Custom for Oracle)

當您建立下列任一資源時，RDS Custom 會建立 `do-not-delete-rds-custom-` S3 儲存貯體：

- RDS Custom for Oracle 的首個 CEV
- RDS Custom for SQL Server 的首個資料庫執行個體

RDS Custom 為下列各項組合建立一個儲存貯體：

- AWS 帳戶 身份證
- 引擎類型 (RDS Custom for Oracle 或 RDS Custom for SQL Server)
- AWS 區域

例如，如果您在單一儲存貯體中建立適用於 Oracle CEV 的 RDS 自訂項目 AWS 區域，則存在一個 `do-not-delete-rds-custom-` 值區。如果您為 SQL Server 執行個體建立多個 RDS 自訂，且它們位於不同的執行個體中 AWS 區域，則每個執行個體中都有一個 `do-not-delete-rds-custom-` 值區 AWS 區域。如果您在單一執行個體中建立一個適用於 Oracle 執行個體的 RDS 自訂和兩個適用於 SQL Server 的 RDS 自訂執行個體 AWS 區域，則存在兩個 `do-not-delete-rds-custom-` 值區

AWS CloudTrail

RDS 自訂會自動建立名稱開頭為 `do-not-delete-rds-custom-` 的 AWS CloudTrail 追蹤。RDS 自訂支援範圍會依賴事件 CloudTrail 來判斷您的動作是否會影響 RDS 自訂自動化。如需詳細資訊，請參閱 [對不支援的組態進行故障診斷](#)。

RDS Custom 在您建立第一個資料庫執行個體時建立追蹤。RDS Custom 為下列各項組合建立一個追蹤：

- AWS 帳戶 身份證
- 引擎類型 (RDS Custom for Oracle 或 RDS Custom for SQL Server)
- AWS 區域

刪除 RDS 自訂資料庫執行個體時，不會自動移除此執行個體的 CloudTrail。在這種情況下，系統會 AWS 帳戶 繼續向您收取未刪除 CloudTrail 的費用。RDS Custom 不負責刪除此資源。若要瞭解如何 CloudTrail 手動移除，請參閱《AWS CloudTrail 使用指南》中的 [〈刪除追蹤〉](#)。

Amazon RDS Custom 的安全性

熟悉 RDS Custom 的安全考量。

主題

- [RDS Custom 如何安全地代表您管理任務](#)
- [SSL 憑證](#)
- [保護您的 Amazon S3 儲存貯體，避免混淆代理人問題](#)
- [輪換 RDS Custom for Oracle 憑證以符合合規計畫](#)

RDS Custom 如何安全地代表您管理任務

RDS Custom 使用下列工具和技術代表您安全地執行操作：

AWSServiceRoleForRDSCustom 服務連結角色

服務連結角色是由服務預先定義，內含該服務代您呼叫其他 AWS 服務 所需的所有許可。對於 RDS Custom，AWSServiceRoleForRDSCustom 是根據最低權限原則定義的服務連結角色。RDS Custom 會使用 AmazonRDSCustomServiceRolePolicy 中的權限 (連接此角色的政策) 來執行大部分的佈建和所有非主機的管理任務。有關更多信息，請參閱[亞馬遜 RDS CustomServiceRolePolicy](#)。

當它的主機上執行工作時，RDS Custom Automation 會使用服務連結角色的認證來執行使用 AWS Systems Manager 的命令。您可以透過系統管理員指令歷程記錄和 AWS CloudTrail 來稽核命令歷史記錄。系統管理員會使用您的網路設定連線到 RDS Custom 資料庫執行個體。如需詳細資訊，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。

暫時性 IAM 憑證

佈建或刪除資源時，RDS Custom 有時會使用暫時性憑證，其是由呼叫 IAM 主體憑證衍生而來。這些 IAM 憑證受到該主體所連接的 IAM 政策限制，並在操作完成後隨即過期。如需了解使用 RDS Custom 的 IAM 主體所需許可，請參閱 [步驟 5：將必要許可授予 IAM 使用者或角色](#)。

Amazon EC2 執行個體設定檔

EC2 執行個體設定檔是適用於 IAM 角色的容器，可讓您將角色資訊傳遞至 EC2 執行個體。EC2 執行個體位於 RDS Custom 資料庫執行個體的底層。建立 RDS Custom 資料庫執行個體時，您可以提供執行個體設定檔。在執行以主機為基礎的管理任務 (例如備份) 時，RDS Custom 會使用 EC2 執行個體設定檔憑證。如需詳細資訊，請參閱 [手動建立 IAM 角色和執行個體設定檔](#)。

SSH 金鑰對

當 RDS Custom 建立資料庫執行個體底層的 EC2 執行個體時，它會代表您建立 SSH 金鑰對。該鍵使用命名前綴 `do-not-delete-rds-custom-ssh-privatekey-db-`。AWS Secrets Manager 存儲此 SSH 私鑰作為秘密在您的 AWS 帳戶。Amazon RDS 不會儲存、存取或使用這些憑證。如需詳細資訊，請參閱 [Amazon EC2 金鑰對與 Linux 執行個體](#)。

SSL 憑證

RDS Custom 資料庫執行個體不支援受管 SSL 憑證。如果您要部署 SSL，可以在自己的錢包中自行管理 SSL 憑證，並建立 SSL 接聽程式，以保護用戶端資料庫之間或資料庫複寫的連線。如需詳細資訊，請參閱 Oracle Database 文件中的 [設定 Transport Layer Security 身分驗證](#)。

保護您的 Amazon S3 儲存貯體，避免混淆代理人問題

建立 Amazon RDS Custom for Oracle 的自訂引擎版本 (CEV) 或 RDS Custom for Oracle 的資料庫執行個體時，RDS Custom 會建立 Amazon S3 儲存貯體。S3 儲存貯體儲存如 CEV 成品、重做 (交易) 日誌、支援範圍的組態項目等檔案，以及 AWS CloudTrail 日誌。

您可以使用全域條件內容金鑰來防止全混淆代理人問題，使 S3 儲存貯體貯體更加安全。如需詳細資訊，請參閱 [防止跨服務混淆代理人問題](#)。

下列 RDS Custom for Oracle 範例，顯示 S3 儲存貯體政策中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰。對於 RDS Custom for Oracle，請確保包含 CEV 的 Amazon Resource Name (ARN) 和資料庫執行個體。對於 RDS Custom for SQL Server，請確保包含資料庫執行個體的 ARN。

```
...
{
  "Sid": "AWSRDSCustomForOracleInstancesObjectLevelAccess",
  "Effect": "Allow",
  "Principal": {
    "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion",
    "s3:DeleteObject",
    "s3:DeleteObjectVersion",
    "s3:GetObjectRetention",
    "s3:BypassGovernanceRetention"
  ]
}
```

```

],
  "Resource": "arn:aws:s3::do-not-delete-rds-custom-123456789012-us-east-2-c8a6f7/
RDSCustomForOracle/Instances/*",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": [
        "arn:aws:rds:us-east-2:123456789012:db:*",
        "arn:aws:rds:us-east-2:123456789012:cev:*/*"
      ]
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
},
...

```

輪換 RDS Custom for Oracle 憑證以符合合規計畫

有些合規計畫需要定期變更資料庫使用者憑證，例如每 90 天變更一次。RDS Custom for Oracle 會自動輪換部分預先定義資料庫使用者的憑證。

主題

- [為預先定義使用者自動輪換憑證](#)
- [輪換使用者憑證的指南](#)
- [手動輪換使用者憑證](#)

為預先定義使用者自動輪換憑證

如果您的 RDS Custom for Oracle 資料庫執行個體是在 Amazon RDS 中託管，下列預先定義 Oracle 使用者的憑證會每 30 天自動輪換一次。上述使用者的認證位於中 AWS Secrets Manager。

預先定義的 Oracle 使用者

資料庫使用者	建立者	支援的引擎版本	備註
SYS	Oracle	custom-oracle-ee	
		custom-oracle-ee-cdb	
		custom-oracle-se2	

資料庫使用者	建立者	支援的引擎版本	備註
		custom-oracle-se2-國別分組	
SYSTEM	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2-國別分組	
RDSADMIN	RDS	custom-oracle-ee custom-oracle-se2	
C##RDSADMIN	RDS	custom-oracle-ee-cdb custom-oracle-se2-國別分組	帶有C##前綴的使用者名稱僅存在於 CDB 中。如需有關 CDB 的詳細資訊，請參閱 Amazon RDS Custom for Oracle 架構的概觀 。
RDS_DATAGUARD	RDS	custom-oracle-ee	此使用者僅存在於僅供讀取複本、僅供讀取複本的來源資料庫，以及您以 Oracle Data Guard 實際移轉至 RDS Custom 的資料庫。
C##RDS_DATAGUARD	RDS	custom-oracle-ee-cdb	此使用者僅存在於僅供讀取複本、僅供讀取複本的來源資料庫，以及您以 Oracle Data Guard 實際移轉至 RDS Custom 的資料庫。帶有C##前綴的使用者名稱僅存在於 CDB 中。如需有關 CDB 的詳細資訊，請參閱 Amazon RDS Custom for Oracle 架構的概觀 。

自動憑證輪換的例外狀況是您已手動設為待命資料庫的 Oracle 資料庫執行個體之 RDS Custom。RDS 只會輪換您使用 `create-db-instance-read-replica` CLI 命令或 `CreateDBInstanceReadReplica` API 建立的僅供讀取複本憑證。

輪換使用者憑證的指南

為了確保您的憑證會根據合規計畫進行輪換，請注意下列事項：

- 如果您的資料庫執行個體會自動輪換憑證，請勿手動變更或刪除[預先定義 Oracle 使用者](#)列出之使用者秘密、密碼檔案或密碼。否則，RDS Custom 可能會將您的資料庫執行個體置於支援範圍之外，從而暫停自動輪換。
- RDS 主要使用者並未預先定義，因此您必須手動變更密碼或在 Secrets Manager 中設定自動輪換。如需詳細資訊，請參閱[旋轉 AWS Secrets Manager 密碼](#)。

手動輪換使用者憑證

針對以下資料庫類別，RDS 不會自動輪換[預先定義 Oracle 使用者](#)列出的使用者憑證：

- 您手動設為待命資料庫的資料庫。
- 內部部署資料庫。
- 位於支援範圍以外的資料庫執行個體，或處於 RDS Custom 自動化無法執行的狀態。在此情況下，RDS Custom 也不會輪換金鑰。

若您的資料庫屬於上述任何類別，您必須手動輪換使用者憑證。

手動輪換資料庫執行個體的使用者憑證

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 請確認資料庫中，RDS 目前未備份您的資料庫執行個體，或執行高可用性設定等操作。
3. 在資料庫詳細資訊頁面中，選擇組態並記下資料庫執行個體的資源 ID。或者，您可以使用該 AWS CLI 命令 `describe-db-instances`。
4. 前往以下位置開啟機密管理員控制台：<https://console.aws.amazon.com/secretsmanager/>。
5. 在搜尋方塊中，輸入您的資料庫資源 ID，並以下列格式搜尋秘密：

```
do-not-delete-rds-custom-db-resource-id-numeric-string
```

此秘密會儲存 RDSADMIN、SYS 和 SYSTEM 的密碼。下列範例金鑰適用於具有資料庫資源 ID db-ABCDEFGHIJKLNMNOPQRS3TUVWX 的資料庫執行個體：

```
do-not-delete-rds-custom-db-ABCDEFGHIJKLNMNOPQRS3TUVWX-123456
```

⚠ Important

若您的資料庫執行個體是僅供讀取複本並使用 custom-oracle-ee-cdb 引擎，則會有兩個具字尾 *db-resource-id-numeric-string* 的秘密，一個用於主要使用者，另一個用於 RDSADMIN、SYS 和 SYSTEM。若要尋找正確的秘密，請在主機執行下列命令：

```
cat /opt/aws/rdscustomagent/config/database_metadata.json | python3 -c  
"import sys,json; print(json.load(sys.stdin)['dbMonitoringUserPassword'])"
```

dbMonitoringUserPassword 屬性顯示 RDSADMIN、SYS 和 SYSTEM 的秘密。

6. 如果您的資料庫執行個體存在於 Oracle Data Guard 組態中，請使用下列格式搜尋秘密：

```
do-not-delete-rds-custom-db-resource-id-numeric-string-dg
```

此秘密會儲存 RDS_DATAGUARD 的密碼。下列範例金鑰適用於具有資料庫資源 ID db-ABCDEFGHIJKLNMNOPQRS3TUVWX 的資料庫執行個體：

```
do-not-delete-rds-custom-db-ABCDEFGHIJKLNMNOPQRS3TUVWX-789012-dg
```

7. 對於「[預先定義的 Oracle 使用者](#)」中列出的所有資料庫使用者，請依照[修改密碼中的指示更新 AWS Secrets Manager 密碼](#)。
8. 如果您的資料庫是獨立資料庫或 Oracle Data Guard 組態中的來源資料庫：

- a. 啟動您的 Oracle SQL 用戶端並以 SYS 登入。
- b. 針對[預先定義 Oracle 使用者](#)列出的每個資料庫使用者，以下列格式執行 SQL 陳述式：

```
ALTER USER user-name IDENTIFIED BY pwd-from-secrets-manager ACCOUNT UNLOCK;
```

例如，如果 RDSADMIN 儲存在 Secrets Manager 的密碼是 pwd-123，請執行下列陳述式：


```
ALTER USER RDSADMIN IDENTIFIED BY pwd-123 ACCOUNT UNLOCK;
```

9. 若您的資料庫執行個體執行 Oracle 資料庫 12c 版本 1 (12.1)，且由 Oracle Data Guard 管理，請手動將密碼檔案 (orapw) 從主要資料庫執行個體複製到每個待命資料庫執行個體。

若您的資料庫執行個體託管在 Amazon RDS，則密碼檔案位置為 `/rdsdbdata/config/orapw`。對於不在 Amazon RDS 託管的資料庫，在 Linux 和 UNIX 的預設位置是 `$ORACLE_HOME/dbs/orapw$ORACLE_SID`，在 Windows 的預設位置是 `%ORACLE_HOME%\database\PWD%ORACLE_SID%.ora`。

使用 RDS Custom for Oracle

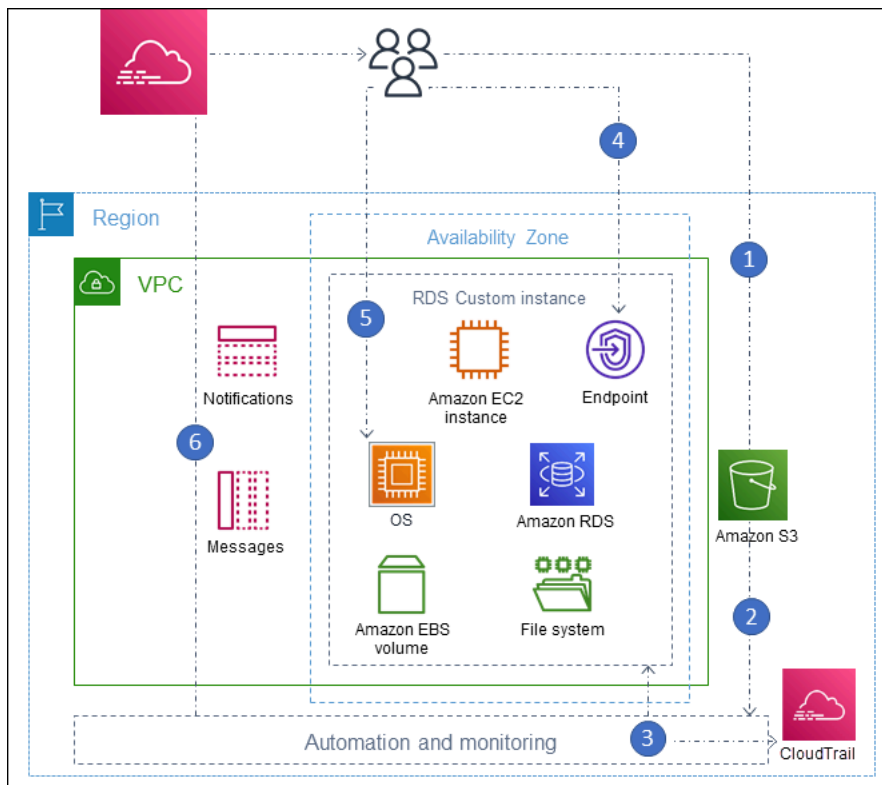
在下列內容，您可以找到建立、管理和維護 RDS Custom for Oracle 資料庫執行個體的指示。

主題

- [RDS Custom for Oracle 工作流程](#)
- [Amazon RDS Custom for Oracle 的資料庫架構](#)
- [適用於 Oracle 的 RDS 自訂功能的可用性和支援](#)
- [RDS Custom for Oracle 需求和限制](#)
- [設定您的 Amazon RDS Custom for Oracle 的環境](#)
- [使用 Amazon RDS Custom for Oracle 的自訂引擎版本](#)
- [設定 Amazon RDS Custom for Oracle 的資料庫執行個體](#)
- [管理 Amazon RDS Custom for Oracle 資料庫執行個體](#)
- [使用 RDS Custom for Oracle 的 Oracle 複本](#)
- [備份與還原 Amazon RDS Custom for Oracle DB 資料庫執行個體](#)
- [在適用於 Oracle 的 RDS 自訂中使用選項群組](#)
- [將內部部署資料庫遷移至 RDS Custom for Oracle](#)
- [升級 Amazon RDS Custom for Oracle 的資料庫執行個體](#)
- [針對 Amazon RDS Custom for Oracle 的資料庫問題進行故障診斷](#)

RDS Custom for Oracle 工作流程

下圖顯示 RDS Custom for Oracle 的典型工作流程。



步驟如下：

1. 將資料庫軟體上傳到您的 Amazon S3 儲存貯體。

如需詳細資訊，請參閱 [步驟 3：將您的安裝檔案上傳到 Amazon S3](#)。

2. 從您的媒體建立 RDS Custom for Oracle 自訂引擎版本 (CEV)。

選擇 CDB 架構或傳統的非 CDB 架構。如需詳細資訊，請參閱 [建立 CEV](#)。

3. 從 CEV 建立 RDS Custom for Oracle 資料庫執行個體。

如需詳細資訊，請參閱 [建立 RDS Custom for Oracle 資料庫執行個體](#)。

4. 將應用程式連線到資料庫執行個體端點。

如需詳細資訊，請參閱 [使用 SSH 連線到您的 RDS Custom 資料庫執行個體](#) 及 [使用工作階段管理員連線至 RDS Custom 資料庫執行個體](#)。

5. (選用) 存取主機以自訂您的軟體。

6. 監控 RDS Custom 自動化產生的通知和訊息。

資料庫安裝檔案

您對媒體的責任是 Amazon RDS 與和 RDS Custom 之間的關鍵差異。Amazon RDS 是一個全受管服務，提供 Amazon Machine Image (AMI)和資料庫軟體。Amazon RDS 資料庫軟體已預先安裝，因此您只需選擇資料庫引擎和版本，然後建立資料庫。

對於 RDS Custom，您可以提供自己的媒體。當您建立自訂引擎版本時，RDS Custom 會安裝您提供的媒體。RDS Custom 媒體包含您的資料庫安裝檔案和修補程式。此服務模型稱為使用自有媒體 (BYOM)。

RDS Custom for Oracle 的自訂引擎版本

RDS Custom for Oracle 自訂引擎版本 (CEV) 是資料庫版本和 AMI 的二進位磁碟區快照。依預設，RDS Custom for Oracle 會使用 Amazon EC2 提供的最新 AMI。您也可以選擇重複使用現有的 IAM。

CEV 清單檔案

從 Oracle 下載 Oracle 資料庫安裝檔案之後，您可以將其上傳到 Amazon S3 儲存貯體。建立 CEV 時，您可以在稱為 CEV 清單檔案的 JSON 文件中指定檔案名稱。RDS Custom for Oracle 會使用指定的檔案和 AMI 來建立您的 CEV。

RDS Custom for Oracle 會針對每個支援的 Oracle Database 版本，提供 JSON 清單檔案範本以及我們建議的 .zip 檔案。例如，下列範本適用於 19.17.0.0.0 RU。

```
{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V982063-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_190000_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [
    "p34419443_190000_Linux-x86-64.zip",
    "p34411846_190000_Linux-x86-64.zip"
  ],
  "otherPatchFileNames": [
    "p28852325_190000_Linux-x86-64.zip",
    "p29997937_190000_Linux-x86-64.zip",
    "p31335037_190000_Linux-x86-64.zip",
```

```
"p32327201_190000_Linux-x86-64.zip",
"p33613829_190000_Linux-x86-64.zip",
"p34006614_190000_Linux-x86-64.zip",
"p34533061_190000_Linux-x86-64.zip",
"p34533150_190000_Generic.zip",
"p28730253_190000_Linux-x86-64.zip",
"p29213893_1917000DBRU_Generic.zip",
"p33125873_1917000DBRU_Linux-x86-64.zip",
"p34446152_1917000DBRU_Linux-x86-64.zip"
]
}
```

您也可以在 JSON 清單檔案中指定安裝參數。例如，您可以為 Oracle 基本目錄、Oracle 主目錄，以及 UNIX/Linux 使用者和群組的 ID 和名稱設定非預設值。如需詳細資訊，請參閱 [CEV 資訊清單中的 JSON 欄位](#)。

CEV 命名格式

使用客戶指定的字串命名您的 RDS Custom for Oracle CEV。名稱格式如下，具體取決於 Oracle 資料庫版本：

- 19.*customized_string*
- 18.*customized_string*
- 12.2.*customized_string*
- 12.1.*customized_string*

您可以使用 1–50 個英數字元、字元、底線、破折號和句點。例如，您可以將 CEV 命名為 19.my_cev1。

RDS Custom for Oracle 中的 Oracle 多租戶架構

Oracle 多租戶架構可讓 Oracle 資料庫作為容器資料庫 (CDB) 運作。CDB 包含零個、一個或多個顧客建立的插入式資料庫 (PDB)。PDB 是結構描述和物件的可攜性集合，此集合會對應用程式顯示為傳統非 CDB。從 Oracle Database 21c 開始，所有 Oracle 資料庫皆為 CDB。

建立 RDS Custom for Oracle CEV 時，您可以指定 CDB 或非 CDB 架構。只有當您用來建立 RDS Custom for Oracle CDB 的 CEV 使用 Oracle 多租戶架構時，才能建立該 CDB。如需詳細資訊，請參閱 [使用 Amazon RDS Custom for Oracle 的自訂引擎版本](#)。

建立 RDS Custom for Oracle 資料庫執行個體

建立 CEV 後，您就可以使用它。您可以建立多個 CEV，也可以從任何 CEV 建立多個 RDS Custom for Oracle 資料庫執行個體。您也可以變更 CEV 的狀態，使其成為可用或非作用中。

您可以使用 Oracle 多租戶架構 (custom-oracle-ee-cdb 或 custom-oracle-se2-cdb 引擎類型) 或傳統的非 CDB 架構 (或引擎類型) 來建立適用於 Oracle 資料庫執行個體的 RDS 自訂。custom-oracle-ee 或 custom-oracle-se2 當您建立一個容器資料庫 (CDB) 時，它包含一個插入式資料庫 (PDB) 和一個 PDB 種子。您可以使用 Oracle SQL 手動建立額外的 PDB。

若要建立 RDS Custom for Oracle 資料庫執行個體，請使用 `create-db-instance` 命令。在此命令中，指定要使用的 CEV。該程序類似於建立 Amazon RDS 資料庫執行個體的程序。不過，某些參數不同。如需詳細資訊，請參閱 [設定 Amazon RDS Custom for Oracle 的資料庫執行個體](#)。

資料庫連線

與 Amazon RDS 資料庫執行個體相同，RDS Custom 資料庫執行個體位於 虛擬私有雲端 (VPC) 中。您的應用程式會使用 Oracle 接聽程式連線至 Oracle 資料庫。

如果您的資料庫是 CDB，您可以使用接聽程式 `L_RDSCDB_001` 連線至 CDB 根和 PDB。如果您將非 CDB 插入 CDB 中，請務必設定 `USE_SID_AS_SERVICE_LISTENER = ON`，以便遷移的應用程式保持相同的設定。

當您連線至非 CDB 時，主要使用者就是非 CDB 的使用者。當您連線至 CDB 時，主要使用者就是 PDB 的使用者。若要連線至 CDB 根，請登入主機、啟動 SQL 用戶端，然後使用 SQL 命令建立管理使用者。

RDS Custom 自訂

您可以存取 RDS Custom 主機以安裝或自訂軟體。若要避免變更與 RDS Custom 自動化之間發生衝突，您可以在指定期間暫停自動化。在此期間，RDS Custom 不會執行監控或執行個體復原。在期間結束時，RDS Custom 會繼續全自動化。如需更多詳細資訊，請參閱 [暫停和繼續 RDS Custom 資料庫執行個體](#)。

Amazon RDS Custom for Oracle 的資料庫架構

RDS Custom for Oracle 同時支援 Oracle 多租戶和非多租戶架構。

主題

- [支援的 Oracle 資料庫架構](#)
- [支援的引擎類型](#)
- [Oracle 多租戶架構中支援的功能](#)

支援的 Oracle 資料庫架構

Oracle 多租戶架構 (也稱為 CDB 架構) 可讓 Oracle 資料庫作為容器資料庫 (CDB) 運作。CDB 包含插入式資料庫 (PDB)。PDB 是結構描述和物件的集合，此集合會對應用程式顯示為傳統 Oracle 資料庫。如需詳細資訊，請參閱《Oracle 多租戶管理員指南》中的[多租戶架構簡介](#)。

CDB 和非 CDB 架構是互斥的。如果 Oracle 資料庫不是 CDB，則其是非 CDB，因此不能包含 PDB。在 RDS Custom for Oracle 中，僅 Oracle Database 19c 支援 CDB 多租戶架構。因此，如果您使用舊版 Oracle 資料庫建立資料庫執行個體，則只能建立非 CDB。如需詳細資訊，請參閱[多租戶架構考量](#)。

支援的引擎類型

當您為 Oracle CEV 或資料庫執行個體建立 Amazon RDS 自訂時，請選擇 CDB 引擎類型或非 CDB 引擎類型：

- `custom-oracle-ee-cdb` 和 `custom-oracle-se2-cdb`

這些引擎類型會指定 Oracle 多租用戶架構。此選項僅適用於 Oracle Database 19c。當您使用多租用戶架構建立 RDS for Oracle 資料庫執行個體時，您的 CDB 包含下列容器：

- CDB 根 (CDB\$ROOT)
- PDB 種子 (PDB\$SEED)
- 起始 PDB

您可以使用 Oracle SQL 命令 `CREATE PLUGGABLE DATABASE` 建立更多 PDB。您無法使用 RDS API 建立或刪除 PDB。

- `custom-oracle-ee` 和 `custom-oracle-se2`

這些引擎類型指定了傳統的非 CDB 架構。非 CDB 不能包含插入式資料庫 (PDB)。

如需詳細資訊，請參閱 [多租戶架構考量](#)。

Oracle 多租戶架構中支援的功能

RDS Custom for Oracle CDB 執行個體支援下列功能：

- 備份
- 從備份還原和 point-time-restore (PITR)
- 僅供讀取複本
- 次要版本升級

適用於 Oracle 的 RDS 自訂功能的可用性和支援

在本主題中，您可以找到適用於 Oracle 的 RDS 自訂功能的可用性和支援摘要，以供快速參考。

主題

- [AWS 區域和資料庫版本支援 RDS 自訂](#)
- [適用於 Oracle 的 RDS 自訂資料庫版本支援](#)
- [RDS Custom for Oracle 的版本和授權支援](#)
- [RDS Custom for Oracle 的資料庫執行個體類別支援](#)
- [適用於 Oracle 的 RDS 自訂選項群組支援](#)

AWS 區域和資料庫版本支援 RDS 自訂

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 RDS Custom for Oracle 的版本和區域可用性的詳細資訊，請參閱 [RDS 自訂支援的區域和資料庫引擎](#)。

適用於 Oracle 的 RDS 自訂資料庫版本支援

適用於 Oracle 的 RDS 自訂支援下列 Oracle 資料庫版本：

- Oracle 資料庫 19c
- Oracle 資料庫 18c
- Oracle 資料庫 12c 版本 2 (12.2)
- Oracle 資料庫 12c 版本 1 (12.1)

RDS Custom for Oracle 的版本和授權支援

適用於甲骨文的 RDS 自定義支持 BYOL 模型上的企業版 (EE) 和標準版 2 (SE2)。

請注意標準版 2 的下列限制：

- 不支援 Oracle 資料保全。因此，您無法建立 Oracle 僅供讀取複本。
- 您只能使用具有 16 個或更少 vCPUs (最多 4 個) 的資料庫執行個體類別。
- 標準版 2 上的 CDB 執行個體最多支援 3 個租用戶資料庫。
- 您無法在企業版和標準版 2 之間遷移資料。

RDS Custom for Oracle 的資料庫執行個體類別支援

RDS Custom for Oracle 支援下列資料庫執行個體類別。如果您在標準版 2 上建立資料庫執行個體，則只能使用 16 個或更少 vCPUs (最多 4 倍) 的執行個體類別。

Type	大小
db.r6i	db.r6i.large db.r6i.xlarge db.r6i.2xlarge db.r6i.4xlarge db.r6i.8xlarge db.r6i.12xlarge db.r6i.16xlarge db.r6i.24xlarge db.r6i.32xlarge
db.r5b	db.r5b.large db.r5b.xlarge db.r5b.2xlarge db.r5b.4xlarge db.r5b.8xlarge db.r5b.12xlarge db.r5b.16xlarge db.r5b.24xlarge
db.r5	db.r5.large db.r5.xlarge db.r5.2xlarge db.r5.4xlarge db.r5.8xlarge db.r5.12xlarge db.r5.16xlarge db.r5.24xlarge
db.x2iecd	db.x2iedn.xlarge db.x2iedn.2xlarge db.x2iedn.4xlarge db.x2iedn.8xlarge db.x2iedn.16xlarge db.x2iedn.24xlarge db.x2iedn.32xlarge
數據庫 .x2	db.x2iezn.2xlarge db.x2iezn.4xlarge db.x2iezn.6xlarge db.x2iezn.8xlarge db.x2iezn.12xlarge
db.m6i	db.m6i.large db.m6i.xlarge db.m6i.2xlarge db.m6i.4xlarge db.m6i.8xlarge db.m6i.12xlarge db.m6i.16xlarge db.m6i.24xlarge db.m6i.32xlarge
db.m5	db.m5.large db.m5.xlarge db.m5.2xlarge db.m5.4xlarge db.m5.8xlarge db.m5.12xlarge db.m5.16xlarge db.m5.24xlarge
db.t3	db.t3.medium db.t3.large db.t3.xlarge db.t3.2xlarge

適用於 Oracle 的 RDS 自訂選項群組支援

您可以在建立或修改適用於 Oracle 資料庫執行個體的 RDS 自訂時指定選項群組。如需更多詳細資訊，請參閱 [在適用於 Oracle 的 RDS 自訂中使用選項群組](#)。

RDS Custom for Oracle 需求和限制

在本主題中，您可以尋找 Amazon RDS Custom for Oracle 功能可用性和需求的摘要，以進行快速參考。

主題

- [RDS Custom for Oracle 的一般需求](#)
- [RDS Custom for Oracle 的一般限制](#)
- [適用於甲骨文的 RDS 自定義的 CEV 和 AMI 限制](#)
- [不支援的建立和修改工作流程設定](#)
- [您的資料庫執行個體配額 AWS 帳戶](#)

RDS Custom for Oracle 的一般需求

請務必符合 Amazon RDS 自訂適用於甲骨文的下列要求：

- 您可以存取 [「我的 Oracle 客戶服務部」](#) 和 [「Oracle 軟體交付雲端」](#)，以下載適用於 Oracle 的 RDS 自訂安裝檔案和修補程式的 Support 援清單。如果您使用未知的修補程式，則自訂引擎版本 (CEV) 建立就會失敗。在此情況下，請聯絡 RDS Custom 支援團隊，並要求新增遺失的修補程式。如需詳細資訊，請參閱 [步驟 2：從 Oracle Software Delivery Cloud 下載資料庫安裝檔案和修補程式](#)。
- 您具有 Amazon S3 的存取權。您需要這項服務的原因如下：
 - 您可以將 Oracle 安裝檔案上傳到 S3 儲存貯體。建立 RDS Custom CEV 時，您可以使用上傳的安裝檔案。
 - RDS Custom for Oracle 使用從內部定義 S3 儲存貯體下載的指令碼，對資料庫執行個體執行動作。加入動作和 RDS Custom 自動化需要這些指令碼。
 - RDS Custom for Oracle 會將特定檔案上傳到您客戶帳戶中的 S3 儲存貯體。這些儲存貯體使用以下命名格式：`do-not-delete-rds-custom-account_id-region-six_character_alphanumeric_string`。例如，您可能會有名為 `do-not-delete-rds-custom-123456789012-us-east-1-12a3b4` 的儲存貯體。

如需詳細資訊，請參閱 [步驟 3：將您的安裝檔案上傳到 Amazon S3](#) 及 [建立 CEV](#)。

- 您可以使用中列出的資料庫執行個體類別 [RDS Custom for Oracle 的資料庫執行個體類別支援](#) 來建立適用於 Oracle 資料庫執行個體的 RDS 自訂。
- Oracle 資料庫執行個體的 RDS 自訂執行個體會執行 Oracle Linux 7 更新 9 或更新版本。
- 您可以為 Amazon EBS 儲存指定 gp2、gp3 或 io1 固態磁碟機。最大儲存大小為 64 TiB。

- 您有一個 AWS KMS 金鑰可以建立適用於 Oracle 資料庫執行個體的 RDS 自訂。如需詳細資訊，請參閱 [步驟 1：建立或重複使用對稱加密 AWS KMS 金鑰](#)。
- 您具有建立適用於 Oracle 資料庫執行個體的 RDS 自訂所需的 AWS Identity and Access Management (IAM) 角色和執行個體設定檔。如需詳細資訊，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。
- 建立 CEV 或 RDS 自訂資料庫執行個體的 AWS Identity and Access Management (IAM) 使用者具有 IAM 和 Amazon S3 所需的許可。CloudTrail

如需詳細資訊，請參閱 [步驟 5：將必要許可授予 IAM 使用者或角色](#)。

- 您可以提供您自己的 Virtual Private Cloud (VPC) 和安全群組組態。如需詳細資訊，請參閱 [步驟 6：為 Oracle 設定適用於 RDS 的自訂 VPC 人雲端](#)。
- 您提供的網路組態，RDS 自訂適用於 Oracle 可以用來存取其他 AWS 服務。如需特定需求，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。

RDS Custom for Oracle 的一般限制

下列限制適用於 RDS Custom for Oracle：

- 您無法修改適用於 Oracle 資料庫執行個體的現有 RDS Custom 資料庫執行個體識別碼。
- 您只能為 Oracle 資料庫 19c 指定 Oracle 多租戶架構。
- 您無法在單一 RDS Custom for Oracle 資料庫執行個體上建立多個 Oracle 資料庫。
- 您無法停止 RDS Custom for Oracle 資料庫執行個體或其基礎 Amazon EC2 執行個體。無法停止 RDS Custom for Oracle 資料庫執行個體的計費。
- 您無法使用自動共用記憶體管理，因為適用於 Oracle 的 RDS 自訂僅支援自動記憶體管理。如需詳細資訊，請參閱《Oracle 資料庫管理員指南》中的 [自動記憶體管理](#)。
- 務必不要變更主要資料庫執行個體的 DB_UNIQUE_NAME。變更名稱會導致任何還原操作停滯。

如需修改 RDS Custom for Oracle 資料庫執行個體的特有限制，請參閱 [修改您的 RDS Custom for Oracle 資料庫執行個體](#)。如需複寫限制，請參閱 [RDS Custom for Oracle 複本的一般限制](#)。

適用於甲骨文的 RDS 自定義的 CEV 和 AMI 限制

下列限制適用於 Oracle CEV 和 AMI 的 RDS 自訂：

- 您無法提供您自己的 AMI，以便在 Oracle CEV 的 RDS 自訂中使用。您可以指定預設 AMI 或先前已由 Oracle CEV 的 RDS 自訂使用的 AMI。

Note

Oracle 版 RDS 自訂會在發現常見弱點和暴露時發佈新的預設 AMI。沒有固定的時間表可用或保證。適用於甲骨文的 RDS 自定義傾向於每 30 天發布一次新的默認 AMI。

- 您無法修改 CEV 以使用不同的 AMI。
- 您無法從使用 custom-oracle-ee 或 custom-oracle-se2 引擎類型的 CEV 建立 CDB 執行個體。CEV 必須使用 custom-oracle-ee-cdb 或 custom-oracle-se2-cdb。
- 適用於 Oracle 的 RDS 自訂目前不允許您使用 RDS API 呼叫升級適用於 Oracle 資料庫執行個體的 RDS 自訂作業系統。因應措施是，您可以使用下列命令手動更新作業系統：`sudo yum update --security`。

不支援的建立和修改工作流程設定

建立或修改適用於 Oracle 資料庫執行個體的 RDS 自訂時，您無法執行下列動作：

- 變更資料庫執行個體類別中每個核心的 CPU 核心數量和每個核心的執行緒數量。
- 開啟儲存體自動擴展。
- 建立多可用區域部署。

Note

如需替代的 HA 解決方案，請參閱部 AWS 部落格文章使用僅供讀取複本為 Oracle 建立適用於 [Oracle 的高可用性](#)。

- 將備份保留設定為 0。
- 設定 Kerberos 身分驗證。
- 指定您自己的資料庫參數群組或選項群組。
- 開啟績效詳情。
- 開啟自動次要版本升級。

您的資料庫執行個體配額 AWS 帳戶

確保 RDS Custom 和 Amazon RDS 資料庫執行個體的合併數量不超過您的配額限制。例如，如果您的 Amazon RDS 配額為 40 個資料庫執行個體，則可以擁有 20 個 RDS Custom for Oracle 資料庫執行個體和 20 個 Amazon RDS 資料庫執行個體。

設定您的 Amazon RDS Custom for Oracle 的環境

在建立 Amazon RDS Custom for Oracle 資料庫執行個體之前，請執行下列任務。

主題

- [步驟 1：建立或重複使用對稱加密 AWS KMS 金鑰](#)
- [步驟 2：下載並安裝 AWS CLI](#)
- [步驟 3：擷取適用於 Oracle 的 RDS 自訂 CloudFormation 範本](#)
- [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)
- [步驟 5：將必要許可授予 IAM 使用者或角色](#)
- [步驟 6：為 Oracle 設定適用於 RDS 的自訂 VPC 人雲端](#)

步驟 1：建立或重複使用對稱加密 AWS KMS 金鑰

客戶管理金鑰 AWS KMS keys 位於您建立、擁有及管理的 AWS 帳戶中。RDS Custom 需要客戶受管對稱加密 KMS 金鑰。建立 RDS Custom for Oracle 資料庫執行個體時，您要提供 KMS 金鑰識別符。如需詳細資訊，請參閱 [設定 Amazon RDS Custom for Oracle 的資料庫執行個體](#)。

您有下列選項：

- 如果您的中有現有的客戶受管 KMS 金鑰 AWS 帳戶，您可以將其與 RDS 自訂搭配使用。無需採取進一步動作。
- 若您已為不同的 RDS Custom 引擎建立客戶受管對稱加密 KMS 金鑰，您可以重複使用相同的 KMS 金鑰。無需採取進一步動作。
- 若您的帳戶中並無現有的客戶受管對稱加密 KMS 金鑰，請藉由遵循《AWS Key Management Service 開發人員指南》中的[建立金鑰](#)來建立 KMS 金鑰。
- 如果您要建立 CEV 或 RDS 自訂資料庫執行個體，而您的 KMS 金鑰位於不同的 AWS 帳戶，請務必使用 AWS CLI。您無法搭配跨帳戶 KMS 金鑰使用 AWS 主控台。

Important

RDS 自訂不支援 AWS 受管 KMS 金鑰。

確保您的對稱加密金鑰授 `kms:Decrypt` 與存取 IAM 執行個體設定檔中 AWS Identity and Access Management (IAM) 角色的存取權和 `kms:GenerateDataKey` 操作。若您的帳戶中有新的對稱加密金鑰，則不需要變更。否則，請確定對稱加密金鑰政策授予這些操作的存取權。

如需詳細資訊，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。

如需設定 RDS Custom for Oracle 的 IAM 的相關詳細資訊，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。

步驟 2：下載並安裝 AWS CLI

AWS 提供您使用 RDS 自訂功能的命令列介面。您可以使用 AWS CLI 的第 1 版或第 2 版。

如需有關下載和安裝的資訊 AWS CLI，請參閱 [安裝或更新最新版本的 AWS CLI](#)。

如果以下任一情況屬實，請略過此步驟：

- 您計劃僅從存取 RDS 自訂 AWS Management Console。
- 您已經下載了 AWS CLI 適用於 Amazon RDS 或不同的 RDS 自訂資料庫引擎。

步驟 3：擷取適用於 Oracle 的 RDS 自訂 CloudFormation 範本

為了簡化設定，我們強烈建議您使用 AWS CloudFormation 範本建立 CloudFormation 堆疊。如果您打算手動設定 IAM 和 VPC，請略過此步驟。

主題

- [步驟 3a：下載模 CloudFormation 板文件](#)
- [步驟 3：提取 custom-oracle-iam.json](#)
- [步驟 3：提取自定義](#)

步驟 3a：下載模 CloudFormation 板文件

CloudFormation 模板是構成堆棧的 AWS 資源的聲明。範本會儲存為 JSON 檔案。

下載 CloudFormation 範本檔案的步驟

1. 開啟 [custom-oracle-iam.zip](#) 連結的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「另存連結為」。

2. 將檔案儲存到您的電腦。
3. 針對連結 [custom-vpc.zip](#) 重複先前的步驟。

如果您已經為 RDS Custom 設定 VPC，請略過這個步驟。

步驟 3：提取 custom-oracle-iam.json

開啟您下載的 custom-oracle-iam.zip 檔案，然後解壓縮檔案 custom-oracle-iam.json。檔案的開頭如下所示。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EncryptionKey": {
      "Type": "String",
      "Default": "*",
      "Description": "KMS Key ARN for encryption of data managed by RDS Custom and by
DB Instances."
    }
  },
  "Resources": {
    "RDSCustomInstanceServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": { "Fn::Sub": "AWSRDSCustomInstanceRole-${AWS::Region}" },
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                "Service": "ec2.amazonaws.com"
              }
            }
          ]
        }
      }
    }
  }
},...
```

步驟 3：提取自定義

Note

如果您已經設定適用於 Oracle 的 RDS 自訂的現有 VPC 人雲端，請略過此步驟。如需詳細資訊，請參閱 [為 Oracle 的 RDS 自訂手動設定您的 VPC 人雲端](#)。

打開您下載的 custom-vpc.zip 圖片，然後解壓縮。custom-vpc.json 檔案的開頭如下所示。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "PrivateVpc": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "Private VPC Id to use for RDS Custom DB Instances"
    },
    "PrivateSubnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "Private Subnets to use for RDS Custom DB Instances"
    },
    "RouteTable": {
      "Type": "String",
      "Description": "Route Table that must be associated with the PrivateSubnets and used by S3 VPC Endpoint",
      "AllowedPattern": "rtb-[0-9a-z]+"
    }
  },
  "Resources": {
    "DBSubnetGroup": {
      "Type": "AWS::RDS::DBSubnetGroup",
      "Properties": {
        "DBSubnetGroupName": "rds-custom-private",
        "DBSubnetGroupDescription": "RDS Custom Private Network",
        "SubnetIds": {
          "Ref": "PrivateSubnets"
        }
      }
    }
  },
  ...
}
```

步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM

您透過主控台或 AWS CLI，使用 IAM 角色或 IAM 使用者 (也稱為 IAM 實體)，來建立 RDS Custom 資料庫執行個體。此 IAM 實體必須擁有執行個體建立的必要許可。

您可以使用 CloudFormation 或手動步驟設定 IAM。

Important

我們強烈建議您使用 AWS CloudFormation。這是最簡單且最不容易出錯的方法。

主題

- [使用以下方式設 CloudFormation](#)
- [手動建立 IAM 角色和執行個體設定檔](#)

使用以下方式設 CloudFormation

當您使用 IAM 的 CloudFormation 範本時，它會建立下列必要資源：

- 執行個體設定檔名為 `AWSRDSCustomInstanceProfile-region`
- 名為 `AWSRDSCustomInstanceRole-region` 的服務角色
- 附加至服務角色 `AWSRDSCustomIamRolePolicy` 的名稱存取原則

若要使用設定 IAM CloudFormation

1. [請在以下位置開啟 CloudFormation 主控台。](https://console.aws.amazon.com/cloudformation) <https://console.aws.amazon.com/cloudformation>
2. 啟動「建立堆疊」精靈，然後選擇 Create Stack (建立堆疊)。
3. 在 Create stack (建立堆疊) 頁面上，執行下列動作：
 - a. 針對 Prepare template (準備範本)，選擇 Template is ready (範本已就緒)。
 - b. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。
 - c. 在選擇檔案中，導覽並選擇 custom-oracle-iam.json。
 - d. 選擇下一步。
4. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，執行下列操作：

- a. 針對堆疊名稱輸入 **custom-oracle-iam**。
 - b. 選擇下一步。
5. 在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。
 6. 在「複查 custom-oracle-iam」頁面上，執行下列動作：
 - a. 選取 I acknowledge that AWS CloudFormation might create IAM resources with custom names (確認 AWS CloudFormation 可能使用自訂名稱建立 IAM 資源) 核取方塊。
 - b. 選擇提交。

CloudFormation 建立適用於 Oracle 的 RDS 自訂所需的 IAM 角色。在左側面板中，當 custom-oracle-iam 顯示 CREATE_COMPLETE 時，請繼續執行下一個步驟。

7. 在左側面板中，選擇 custom-oracle-iam。在右側面板中，執行下列作業：
 - a. 選擇堆疊資訊。您的堆疊具有格式為 `arn:aws:cloudformation:region:account-no:stack/custom-oracle-iam/identifier` 的 ID。
 - b. 選擇資源。請查看下列事項：
 - 名為 `AWSRDSCustomInstanceProfile-##` 的執行個體設定檔
 - 名為 `AWSRDSCustomInstanceRole-##` 的服務角色

建立 RDS Custom 資料庫執行個體時，您必須提供執行個體設定檔 ID。

手動建立 IAM 角色和執行個體設定檔

使用時配置是最簡單的 CloudFormation。或者，您也可以手動設定 IAM。對於手動設定，請執行下列操作：

- [步驟 1：建立 AWSRDSCustomInstanceRoleForRdsCustomInstance IAM 角色。](#)
- [步驟 2：將存取原則新增至 AWSRDSCustomInstanceRoleForRdsCustomInstance。](#)
- [步驟 2：將存取原則新增至 AWSRDSCustomInstanceRoleForRdsCustomInstance。](#)
- [步驟 4：加 AWSRDSCustomInstanceRoleForRdsCustomInstance 入 AWSRDSCustomInstanceProfile。](#)

步驟 1：建立 AWSRDSCustomInstanceRoleForRdsCustomInstance IAM 角色

在此步驟中，您可以使用命名格式 `AWSRDSCustomInstanceRole-region` 來建立角色。使用信任政策，Amazon EC2 可以擔任角色。下列範例假設您已將環境變數 `$REGION` 設為 AWS 區域，且您要在其中建立資料庫執行個體。

```
aws iam create-role \  
  --role-name AWSRDSCustomInstanceRole-$REGION \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Action": "sts:AssumeRole",  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "ec2.amazonaws.com"  
        }  
      }  
    ]  
  }'
```

步驟 2：將存取原則新增至 AWSRDSCustomInstanceRoleForRdsCustomInstance

當您在 IAM 角色中內嵌內遷政策時，內嵌政策會用來做為角色存取 (許可) 政策的一部分。您可以建立 `AWSRDSCustomIamRolePolicy` 政策，允許 Amazon EC2 傳送和接收訊息以及執行各種動作。

下列範例會建立名為 `AWSRDSCustomIamRolePolicy` 的存取政策，並將其新增至 IAM 角色 `AWSRDSCustomInstanceRole-region`。此範例假設您已設定下列環境變數：

`$REGION`

將此變數設定為您打算 AWS 區域 在其中建立資料庫執行個體的變數。

`$ACCOUNT_ID`

將此變數設定為您的 AWS 帳戶 號碼。

`$KMS_KEY`

將此變數設為 AWS KMS key 的 Amazon Resource Name (ARN)，您想要用於 RDS Custom 資料庫執行個體。若要指定一個以上的 KMS 金鑰，請將它新增至陳述式 ID (Sid) 11 的 Resources 區段。

```
aws iam put-role-policy \  
  --role-name AWSRDSCustomInstanceRole-$REGION \  
  --policy-name AWSRDSCustomIamRolePolicy \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "1",  
        "Effect": "Allow",  
        "Action": [  
          "ssm:DescribeAssociation",  
          "ssm:GetDeployablePatchSnapshotForInstance",  
          "ssm:GetDocument",  
          "ssm:DescribeDocument",  
          "ssm:GetManifest",  
          "ssm:GetParameter",  
          "ssm:GetParameters",  
          "ssm:ListAssociations",  
          "ssm:ListInstanceAssociations",  
          "ssm:PutInventory",  
          "ssm:PutComplianceItems",  
          "ssm:PutConfigurePackageResult",  
          "ssm:UpdateAssociationStatus",  
          "ssm:UpdateInstanceAssociationStatus",  
          "ssm:UpdateInstanceInformation",  
          "ssm:GetConnectionStatus",  
          "ssm:DescribeInstanceInformation",  
          "ssmmessages:CreateControlChannel",  
          "ssmmessages:CreateDataChannel",  
          "ssmmessages:OpenControlChannel",  
          "ssmmessages:OpenDataChannel"  
        ],  
        "Resource": [  
          "*"   
        ]  
      },  
      {  
        "Sid": "2",  
        "Effect": "Allow",  
        "Action": [  
          "ec2messages:AcknowledgeMessage",  
          "ec2messages:DeleteMessage",  
          "ec2messages:FailMessage",
```

```

        "ec2messages:GetEndpoint",
        "ec2messages:GetMessages",
        "ec2messages:SendReply"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "3",
    "Effect": "Allow",
    "Action": [
        "logs:PutRetentionPolicy",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups",
        "logs:CreateLogStream",
        "logs:CreateLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:$REGION:$ACCOUNT_ID:log-group:rds-custom-instance*"
    ]
},
{
    "Sid": "4",
    "Effect": "Allow",
    "Action": [
        "s3:putObject",
        "s3:getObject",
        "s3:getObjectVersion"
    ],
    "Resource": [
        "arn:aws:s3::do-not-delete-rds-custom-*/*"
    ]
},
{
    "Sid": "5",
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricData"
    ],
    "Resource": [
        "*"
    ]
},

```

```

    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": [
          "RDSCustomForOracle/Agent"
        ]
      }
    },
    {
      "Sid": "6",
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "7",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret"
      ],
      "Resource": [
        "arn:aws:secretsmanager:$REGION:$ACCOUNT_ID:secret:do-not-delete-
rds-custom-*"
      ]
    },
    {
      "Sid": "8",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucketVersions"
      ],
      "Resource": [
        "arn:aws:s3:::do-not-delete-rds-custom-*"
      ]
    },
    {
      "Sid": "9",
      "Effect": "Allow",
      "Action": "ec2:CreateSnapshots",

```



```

    "Resource": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:ec2:*:*:volume/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/AWSRDSCustom": "custom-oracle"
      }
    }
  },
  {
    "Sid": "10",
    "Effect": "Allow",
    "Action": "ec2:CreateSnapshots",
    "Resource": [
      "arn:aws:ec2:*:*:snapshot/*"
    ]
  },
  {
    "Sid": "11",
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$KMS_KEY'"
    ]
  },
  {
    "Sid": "12",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "ec2:CreateAction": [
          "CreateSnapshots"
        ]
      }
    }
  }
}
]

```

```
}'
```

步驟 3：建立 RDS 自訂執行個體設定檔 AWSRDSCustomInstanceProfile

執行個體設定檔是包含單一 IAM 角色的容器。RDS Custom 會使用執行個體設定檔，將角色傳遞到執行個體。

如果您使用 CLI 建立角色，則建立角色與執行個體設定檔為不同動作，且可以使用不同名稱。依照下列方式建立 IAM 執行個體設定檔，並以格式 `AWSRDSCustomInstanceProfile-region` 命名。下列範例假設您已 `$REGION` 將環境變數設定為要 AWS 區域 在其中建立資料庫執行個體的環境變數。

```
aws iam create-instance-profile \  
  --instance-profile-name AWSRDSCustomInstanceProfile-$REGION
```

步驟 4：加 AWSRDSCustomInstanceRoleForRdsCustomInstance 入 AWSRDSCustomInstanceProfile

將 IAM 角色新增至先前建立的執行個體設定檔。下列範例假設您已 `$REGION` 將環境變數設定為要 AWS 區域 在其中建立資料庫執行個體的環境變數。

```
aws iam add-role-to-instance-profile \  
  --instance-profile-name AWSRDSCustomInstanceProfile-$REGION \  
  --role-name AWSRDSCustomInstanceRole-$REGION
```

步驟 5：將必要許可授予 IAM 使用者或角色

確定建立 CEV 或 RDS 自訂資料庫執行個體的 IAM 主體 (使用者或角色) 具有下列其中一項政策：

- AdministratorAccess 政策
- 具有 Amazon S3 和建立 CEV 和 AWS KMS 資料庫執行個體所需許可的 AmazonRDSFullAccess 政策

主題

- [Amazon S3 和 AWS KMS 所需的 IAM 許可](#)
- [建立 CEV 所需的 IAM 許可](#)
- [從 CEV 建立資料庫執行個體所需的 IAM 許可](#)

Amazon S3 和 AWS KMS所需的 IAM 許可

若要為 Oracle 資料庫執行個體建立 CEV 或 RDS 自訂，您的 IAM 主體需要存取 Amazon S3 和 AWS KMS。以下範例 JSON 政策授予必要許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateS3Bucket",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutBucketPolicy",
        "s3:PutBucketObjectLockConfiguration",
        "s3:PutBucketVersioning"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
    },
    {
      "Sid": "CreateKmsGrant",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

如需 kms:CreateGrant 許可的詳細資訊，請參閱 [AWS KMS key管理](#)。

建立 CEV 所需的 IAM 許可

若要建立 CEV，您的 IAM 主體需要下列其他許可：

```
s3:GetObjectAcl
s3:GetObject
s3:GetObjectTagging
s3:ListBucket
mediaimport:CreateDatabaseBinarySnapshot
```

以下範例 JSON 政策授予額外的必要許可，以存取儲存貯體 *my-custom-installation-files* 及其內容。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToS3MediaBucket",
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::my-custom-installation-files",
        "arn:aws:s3::my-custom-installation-files/*"
      ]
    },
    {
      "Sid": "PermissionForByom",
      "Effect": "Allow",
      "Action": [
        "mediaimport:CreateDatabaseBinarySnapshot"
      ],
      "Resource": "*"
    }
  ]
}
```

您可以使用 S3 儲存貯體政策，將 Amazon S3 的類似許可授予呼叫者的帳戶。

從 CEV 建立資料庫執行個體所需的 IAM 許可

若要從現有 CEV 建立適用於 Oracle 資料庫執行個體的 RDS 自訂，IAM 主體需要下列其他許可。

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

下列範例 JSON 政策授予 AWS CloudTrail 驗證 IAM 角色和日誌資訊所需的必要許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ValidateIamRole",
      "Effect": "Allow",
      "Action": "iam:SimulatePrincipalPolicy",
      "Resource": "*"
    },
    {
      "Sid": "CreateCloudTrail",
      "Effect": "Allow",
      "Action": [
        "cloudtrail:CreateTrail",
        "cloudtrail:StartLogging"
      ],
      "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
    }
  ]
}
```

步驟 6：為 Oracle 設定適用於 RDS 的自訂 VPC 人雲端

根據 Amazon VPC 服務，您的 RDS Custom 資料庫執行個體位於虛擬私有雲端 (VPC)，就像是 Amazon EC2 執行個體或 Amazon RDS 執行個體。您可以提供和設定您自己的 VPC。與 RDS Custom for SQL Server 不同，RDS Custom for Oracle 不會建立存取控制清單或安全群組。您必須連接自己的安全群組、子網路和路由表。

您可以使用 CloudFormation 或手動程序來設定虛擬私有雲 (VPC)。

Important

我們強烈建議您使用 AWS CloudFormation。這是最簡單且最不容易出錯的方法。

主題

- [使用 CloudFormation \(建議\) 設定您的 VPC](#)
- [為 Oracle 的 RDS 自訂手動設定您的 VPC 人雲端](#)

使用 CloudFormation (建議) 設定您的 VPC

若您已經為不同的 RDS Custom 引擎設定 VPC，且要重複使用現有 VPC，請略過此步驟。此區段假設如下：

- 您已經習慣建立 CloudFormation 的 IAM 執行個體設定檔和角色。
- 您知道您的路由表 ID。

若要私有化資料庫執行個體，其須位於私有子網路。若要私有化子網路，則其不能與具有預設網際網路閘道的路由表相關聯。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[設定路由表](#)。

當您將 CloudFormation 範本用於 VPC 時，它會建立下列資源：

- 私有 VPC
- 名為的子網路群組 `rds-custom-private`
- 您的資料庫執行個體用來與相依 AWS 服務物件通訊的下列 VPC 端點：
 - `com.amazonaws.region.ec2messages`
 - `com.amazonaws.region.events`
 - `com.amazonaws.region.logs`
 - `com.amazonaws.region.monitoring`
 - `com.amazonaws.region.s3`
 - `com.amazonaws.region.secretsmanager`
 - `com.amazonaws.region.ssm`
 - `com.amazonaws.region.ssmmessages`

Note

對於使用現有帳戶進行複雜的網路設定，如果存取權不存在，建議您手動設定相依服務的存取權。如需詳細資訊，請參閱[確保您的 VPC 可以依賴存取 AWS 服務](#)。

若要使用來設定您的 VPC CloudFormation

1. [請在以下位置開啟 CloudFormation 主控台](https://console.aws.amazon.com/cloudformation)。 <https://console.aws.amazon.com/cloudformation>
2. 啟動「建立堆疊」精靈，選擇建立堆疊，然後選擇使用新資源 (標準)。
3. 在 Create stack (建立堆疊) 頁面上，執行下列動作：

- a. 針對 Prepare template (準備範本)，選擇 Template is ready (範本已就緒)。
 - b. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。
 - c. 針對 Choose file (選擇檔案)，導覽並選擇 custom-vpc.json。
 - d. 選擇下一步。
4. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，執行下列操作：
 - a. 針對堆疊名稱 輸入 **custom-vpc**。
 - b. 針對 Parameters (參數)，選擇要用於 RDS Custom 資料庫執行個體的私有子網路。
 - c. 選擇要用於 RDS Custom 資料庫執行個體的私有 VPC ID。
 - d. 輸入與私有子網路相關聯的路由表。
 - e. 選擇下一步。
 5. 在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。
 6. 在檢閱 custom-vpc 頁面上，選擇提交。

CloudFormation 設定您的私有 VPC。在左側面板中，當 custom-vpc 顯示 CREATE_COMPLETE 時，請繼續執行下一個步驟。

7. (選用) 檢閱私有 VPC 的詳細資訊。在堆疊窗格中，選擇 custom-vpc。在右側窗格中，執行下列作業：
 - a. 選擇堆疊資訊。您的堆疊具有格式為 `arn:aws:cloudformation:region:account-no:stack/custom-vpc/identifier` 的 ID。
 - b. 選擇資源。您應該會看到名為 rds-custom-private 的子網路群組，以及多個使用 `vpce-string` 命名格式的 VPC 端點。每個端點對應到 RDS 自訂需要與之通訊的 AWS 服務該端點。如需詳細資訊，請參閱 [確保您的 VPC 可以依賴存取 AWS 服務](#)。
 - c. 選擇參數。您應該會看到私有子網路、私有 VPC 和您在建立堆疊時指定的路由表。在建立資料庫執行個體時，您需要提供 VPC ID 和子網路群組。

為 Oracle 的 RDS 自訂手動設定您的 VPC 人雲端

除了使用自動化 VPC 建立之外 AWS CloudFormation，您還可以手動設定 VPC。當您有使用現有資源的複雜網路設定時，此選項可能是最佳選擇。

主題

- [確保您的 VPC 可以依賴存取 AWS 服務](#)

- [設定個體中繼資料服務](#)

確保您的 VPC 可以依賴存取 AWS 服務

RDS Custom 會將通訊從您的資料庫執行個體傳送到其他 AWS 服務。請確定可從建立 RDS 自訂資料庫執行個體的子網路存取下列服務：

- Amazon CloudWatch
- Amazon CloudWatch 日誌
- Amazon CloudWatch 活動
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

建立異地同步備份部署

- Amazon Simple Queue Service

如果 RDS Custom 無法與必要的服務通訊，則會發佈下列事件：

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS services. Make sure port 443 (HTTPS) allows outbound connections, and try again. "Failed to connect to the following services: s3 events"
```

若要避免 `incompatible-network` 錯誤，請確定 RDS 自訂資料庫執行個體之間的通訊涉及 VPC 元件，並 AWS 服務 符合下列需求：

- 資料庫執行個體會連接埠 443 上對其他 AWS 服務進行傳出連線。
- VPC 允許對來自您 RDS Custom 資料庫執行個體的請求進行傳入回應。
- RDS Custom 可以正確解析每個 AWS 服務的端點網域名稱。

若您已經為不同的 RDS Custom 資料庫引擎設定 VPC，您可重複使用該 VPC 並略過此程序。

設定個體中繼資料服務

請確定您的執行個體可以執行下列操作：

- 使用執行個體中繼資料服務版本 2 (IMDSv2) 存取執行個體中繼資料服務。
- 允許透過連接埠 80 (HTTP) 與 IMDS 連結 IP 地址進行傳出通訊。
- 從 <http://169.254.169.254> (IMDSv2 連結) 要求執行個體中繼資料。

如需詳細資訊，請參閱 Amazon EC2 使用者[指南中的使用 ImDSv2](#)。

RDS Custom for Oracle 自動化預設會藉由在基礎 Amazon EC2 執行個體上設定 `HttpTokens=enabled` 來使用 IMDSv2。不過，您也可以根據需要使用 IMDSV1。[如需詳細資訊，請參閱 Amazon EC2 使用者指南中的設定執行個體中繼資料選項](#)。

使用 Amazon RDS Custom for Oracle 的自訂引擎版本

Amazon RDS Custom for Oracle 的自訂引擎版本 (CEV) 是資料庫引擎和特定 Amazon Machine Image (AMI) 的二進位磁碟區快照。根據預設，RDS Custom for Oracle 會使用 RDS Custom 管理的最新可用 AMI，但您可以指定舊的 CEV 中使用的 AMI。您可以將資料庫安裝檔案存放在 Amazon S3 中。RDS Custom 會使用安裝檔案和 AMI，為您建立 CEV。

主題

- [準備建立 CEV](#)
- [建立 CEV](#)
- [修改 CEV 狀態](#)
- [檢視 CEV 詳細資訊](#)
- [刪除 CEV](#)

準備建立 CEV

若要建立 CEV，請存取存放在 Amazon S3 儲存貯體中任何下列版本的安裝檔案和修補程式：

- Oracle 資料庫 19c
- Oracle 資料庫 18c
- Oracle 資料庫 12c 版本 2 (12.2)
- Oracle 資料庫 12c 版本 1 (12.1)

例如，您可以對 Oracle 資料庫使用 19c 的 2021 年 4 月版 RU/RUR，或任何有效的安裝檔案和修補程式組合。如需 RDS Custom for Oracle 所支援版本和區域的詳細資訊，請參閱[搭配 RDS for Oracle 的 RDS Custom](#)。

主題

- [步驟 1 \(選用\)：下載清單檔案範本](#)
- [步驟 2：從 Oracle Software Delivery Cloud 下載資料庫安裝檔案和修補程式](#)
- [步驟 3：將您的安裝檔案上傳到 Amazon S3](#)
- [步驟 4 \(選用\)：在 S3 中共用您的安裝媒體 AWS 帳戶](#)
- [步驟 5：準備 CEV 清單檔案](#)

- [步驟 6 \(選用\)：驗證 CEV 清單檔案](#)
- [步驟 7：新增必要的 IAM 許可](#)

步驟 1 (選用)：下載清單檔案範本

CEV 清單檔案是 JSON 文件，其中包含 CEV 的資料庫安裝 .zip 檔案清單。若要建立 CEV，請執行下列動作：

1. 識別您要包含在 CEV 中的 Oracle 資料庫安裝檔案。
2. 下載安裝檔案。
3. 建立列出安裝檔案的 JSON 清單檔案。

RDS Custom for Oracle 會針對每個支援的 Oracle Database 版本，提供 JSON 清單檔案範本以及我們建議的 .zip 檔案。例如，下列範本適用於 19.17.0.0.0 RU。

```
{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V982063-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_190000_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [
    "p34419443_190000_Linux-x86-64.zip",
    "p34411846_190000_Linux-x86-64.zip"
  ],
  "otherPatchFileNames": [
    "p28852325_190000_Linux-x86-64.zip",
    "p29997937_190000_Linux-x86-64.zip",
    "p31335037_190000_Linux-x86-64.zip",
    "p32327201_190000_Linux-x86-64.zip",
    "p33613829_190000_Linux-x86-64.zip",
    "p34006614_190000_Linux-x86-64.zip",
    "p34533061_190000_Linux-x86-64.zip",
    "p34533150_190000_Generic.zip",
    "p28730253_190000_Linux-x86-64.zip",
    "p29213893_1917000DBRU_Generic.zip",
    "p33125873_1917000DBRU_Linux-x86-64.zip",
    "p34446152_1917000DBRU_Linux-x86-64.zip"
  ]
}
```

```
]
}
```

每個範本都有相關聯的 Readme，其中包括下載修補程式的指示、.zip 檔案的 URL，以及檔案總和檢查。您可以依原狀使用這些範本，也可以使用自己的修補程式來修改這些範本。若要檢閱範本，請將 [custom-oracle-manifest.zip](#) 下載至您的本機磁碟，然後使用檔案封存應用程式開啟它。如需詳細資訊，請參閱 [步驟 5：準備 CEV 清單檔案](#)。

步驟 2：從 Oracle Software Delivery Cloud 下載資料庫安裝檔案和修補程式

找出 CEV 所需的安裝檔案後，請將它們下載至您的本機系統。Oracle 資料庫安裝檔案和修補程式是由 Oracle Software Delivery Cloud 託管。每個 CEV 都需要基礎版本 (例如 Oracle Database 19c 或 Oracle Database 12c Release 2 (12.2))，以及選用的修補程式清單。

下載 Oracle Database 的資料庫安裝檔案

- 移至 <https://edelivery.oracle.com/> 並登入。
- 在搜尋方塊中，輸入 **Oracle Database Enterprise Edition** 或 **Oracle Database Standard Edition 2** 然後選擇「搜尋」。
- 選擇下列其中一個基礎版本：

資料庫版本	企業版本	Standard Edition 2
Oracle 資料庫 19c	DLP：甲骨文資料庫 19c 企業版 19.3.0.0.0 (甲骨文資料庫企業版)	DLP：甲骨文資料庫 19c 標準版 2 19.3.0.0.0 (甲骨文資料庫標準版 2)
Oracle 資料庫 18c	DLP：甲骨文資料庫 18c 企業版 18.0.0.0.0 (甲骨文資料庫企業版)	DLP：甲骨文資料庫標準版 2 18.0.0.0.0 (甲骨文資料庫標準版 2)
Oracle 資料庫 12c 版本 2 (12.2.0.1)	DLP：甲骨文資料庫 12c 企業版 12.2.0.1.0 (甲骨文資料庫企業版)	DLP：甲骨文資料庫標準版 2 12.2.0.1.0 (Oracle 資料庫標準版 2)
Oracle 資料庫 12c 版本 1 (12.1.0.2)	DLP：甲骨文資料庫 12c 企業版 12.1.0.2.0 (甲骨文資料庫企業版)	DLP：甲骨文資料庫標準版 2 12.1.0.2.0 (Oracle 資料庫標準版 2)

4. 選擇繼續。
5. 清除 Download Queue (下載佇列) 核取方塊。
6. 選擇對應至基礎版本的選項：
 - Oracle Database 19.3.0.0.0 - 長期版本。
 - Oracle Database 18.0.0.0.0
 - Oracle Database 12.2.0.1.0。
 - Oracle Database 12.1.0.2.0。
7. 在 Platform/Languages (平台/語言) 中選擇 Linux x86-64。
8. 選擇「繼續」，然後簽署「Oracle 授權合約」。
9. 選擇對應至資料庫版本的 .zip 檔案：

資料庫版本與版本	Zip 檔案	SHA-256 雜湊
19c 環保局和 SE2	V982063-01.zip	BA8329C757133DA313ED3B6D7F86C5AC42CD9970A28BF2E6233F3235233AA8D8
18c 環保及季 2	V978967-01.zip	C96A4FD768787AF98272008833FE10B172691CF84E42816B138C12D4DE63AB96
12.2.0.1 環境電腦和 SE2	V839960-01.zip	96ED97D21F15C1AC0CCE3749DA6C3DAC7059BB60672D76B008103FC754D22DDE
12.1.0.2 自由地	V46095-01_1of2.zip V46095-01_2of2.zip	31FDC2AF41687B4E547A3A18F796424D8C1AF36406D2160F65B0AF6A9CD47355 for V46095-01_1of2.zip 03DA14F5E875304B28F0F3BB02AF0EC33227885B99C9865DF70749D1E220ACCD for V46095-01_2of2.zip
第二季	V77388-01_1of2.zip V77388-01_2of2.zip	73873369753230F5A0921F95ACEADB591388CB06ED72A7F3AEA7BCBCEA2403BC 對於 V77388-01_1of2.zip

資料庫版本與版本	Zip 檔案	SHA-256 雜湊
		2492E1BE1E3E3531DA83D0843C09C08E435A C8CEFD9A00C0DF56BE4F15CEEBF3 對於 V77388-01 _2of2.zip

10. 從 updates.oracle.com 或 support.oracle.com 將所需的 Oracle 修補程式下載至本機系統。您可在下列位置尋找修補程式的 URL：

- 您在 [步驟 1 \(選用\)：下載清單檔案範本](#) 所下載的 .zip 檔案中的讀我檔案
- 修補程式列示在 [適用於 Oracle 的 Amazon Relational Database Service \(Amazon RDS\) 版本備註](#) 的每個版本更新 (RU) 中

步驟 3：將您的安裝檔案上傳到 Amazon S3

使用 AWS CLI 將 Oracle 安裝和修補檔案上傳至 Amazon S3。包含安裝檔案的 S3 儲存貯體必須與 CEV 位於相同的 AWS 區域。

本節中的範例使用下列預留位置：

- *install-or-patch-file.zip* – Oracle 安裝媒體檔案。例如，`p32126828_190000_Linux-x86-64.zip` 是修補程式。
- *DOC-EXAMPLE-DESTINATION-BUCKET* – 針對您的上傳安裝檔案指定的 Amazon S3 儲存貯體。
- *123456789012/cev1* – 您的 Amazon S3 儲存貯體的選用字首。
- *DOC-EXAMPLE-SOURCE-BUCKET* – Amazon S3 儲存貯體，您可以選擇性在其中暫存檔案。

主題

- [步驟 3a：確認您的 S3 儲存貯體位於正確的位置 AWS 區域](#)
- [步驟 3b：確定您的 S3 儲存貯體政策具有正確的許可](#)
- [步驟 3c：使用 cp 或 sync 命令上傳檔案](#)
- [步驟 3d：列出 S3 儲存貯體中的檔案](#)

步驟 3a：確認您的 S3 儲存貯體位於正確的位置 AWS 區域

確認您的 S3 儲存貯體位於您計劃執行 `create-custom-db-engine-version` 命令的 AWS 區域。

```
aws s3api get-bucket-location --bucket DOC-EXAMPLE-DESTINATION-BUCKET
```

步驟 3b：確定您的 S3 儲存貯體政策具有正確的許可

您可以從頭或從來源 CEV 建立 CEV。如果您計劃從來源 CEV 建立新 CEV，請確定您的 S3 儲存貯體政策具有正確的許可：

1. 識別 RDS Custom 保留的 S3 儲存貯體。儲存貯體名稱具有格式 `do-not-delete-rds-custom-account-region-string`。例如，儲存貯體可能名為 `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE`。
2. 確定下列許可附加至您的 S3 儲存貯體政策。用您的儲存貯體名稱取代 `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE`。

```
{
  "Sid": "AWSRDSCustomForOracleCustomEngineVersionGetObject",
  "Effect": "Allow",
  "Principal": {
    "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
    "s3:GetObject",
    "s3:GetObjectTagging"
  ],
  "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE/CustomEngineVersions/*"
}, ...
```

步驟 3c：使用 cp 或 sync 命令上傳檔案

請選擇下列其中一個選項：

- 使用 `aws s3 cp` 來上傳單一 .zip 檔案。

分別上傳每個安裝 .zip 檔案。請勿將 .zip 檔案合併成單一 .zip 檔案。

- 使用 `aws s3 sync` 來上傳目錄。

Example

下列範例會將 *install-or-patch-file.zip* 上傳到 RDS Custom Amazon S3 儲存貯體中的 *123456789012/cev1* 資料夾。針對您想要上傳的每個 .zip 分別執行 `aws s3` 命令。

對於LinuxmacOS、或Unix：

```
aws s3 cp install-or-patch-file.zip \  
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

在 Windows 中：

```
aws s3 cp install-or-patch-file.zip ^\  
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

Example

下列範例會將您的本機 *cev1* 資料夾上傳到 Amazon S3 儲存貯體中的 *123456789012/cev1* 資料夾。

對於LinuxmacOS、或Unix：

```
aws s3 sync cev1 \  
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

在 Windows 中：

```
aws s3 sync cev1 ^\  
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

Example

下列範例會將 *DOC-EXAMPLE-SOURCE-BUCKET* 中的所有檔案上傳到 Amazon S3 儲存貯體中的 *123456789012/cev1* 資料夾。

對於LinuxmacOS、或Unix：

```
aws s3 sync s3://DOC-EXAMPLE-SOURCE-BUCKET/ \  
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```



```
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

在 Windows 中：

```
aws s3 sync s3://DOC-EXAMPLE-SOURCE-BUCKET/ ^
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

步驟 3d：列出 S3 儲存貯體中的檔案

下列範例會使用 `s3 ls` 命令，列出 RDS Custom Amazon S3 儲存貯體中的檔案。

```
aws s3 ls \
s3://DOC-EXAMPLE-DESTINATION-BUCKET/123456789012/cev1/
```

步驟 4 (選用)：在 S3 中共用您的安裝媒體 AWS 帳戶

就本區段而言，Amazon S3 儲存貯體包含您上傳的 Oracle 安裝檔案便是媒體儲存貯體。您的組織可能會 AWS 帳戶在 AWS 區域。如果是這樣，您可能需要使用一個 AWS 帳戶來填充您的媒體存儲桶，而使用另一個 AWS 帳戶來創建 CEV。若您不想分享您的媒體儲存貯體，請跳至下一區段。

此區段假設如下：

- 您可以存取建立媒體儲存貯體的帳戶，以及要在其中建立 CEV 的其他帳戶。
- 您打算只在一個 AWS 區域中建立 CEV。如果您打算使用多個區域，請在每個區域中建立媒體儲存貯體。
- 您正在使用 CLI。如果您要使用 Amazon S3 主控台，請採納下列步驟：

若要設定媒體儲存貯體以便共用 AWS 帳戶

1. 登入包 AWS 帳戶含您上傳安裝媒體的 S3 儲存貯體。
2. 您可以從空白的 JSON 政策範本，或現有的政策開始採納。

下列命令會擷取現有的政策，並將其儲存為 `my-policy.json`。在此範例中，包含安裝檔案的 S3 儲存貯體名稱為 `DOC/EXAMPLE ####`。

```
aws s3api get-bucket-policy \
--bucket DOC-EXAMPLE-BUCKET \
--query Policy \
```

```
--output text > my-policy.json
```

3. 編輯媒體儲存貯體權限，如下所示：

- 在範本的 Resource 元素，請指定要上傳 Oracle Database 安裝檔案的 S3 儲存貯體。
- 在Principal元素中，為您要用來建立 CEV AWS 帳戶的所有項目指定 ARN。您可以將根、使用者或角色新增至 S3 儲存貯體允許清單。如需詳細資訊，請參閱 AWS Identity and Access Management 使用者指南中的 [IAM 識別符](#)。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "GrantAccountsAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-1:root",
          "arn:aws:iam::account-2:user/user-name-with-path",
          "arn:aws:iam::account-3:role/role-name-with-path",
          ...
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}
```

4. 將政策連接至您的媒體儲存貯體。

在下面的例子中，*####-###*是包含您的安裝文件的 S3 存儲桶的名稱，*#### .json ### JSON* 文件的名稱。

```
aws s3api put-bucket-policy \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --policy file://my-policy.json
```

5. 登入您要 AWS 帳戶 在其中建立 CEV 的。
6. 確認此帳戶可以存取建立 AWS 帳戶 該帳戶的媒體儲存貯體。

```
aws s3 ls --query "Buckets[].Name"
```

如需詳細資訊，請參閱AWS CLI 命令參考中的 [aws s3 ls](#)。

7. 遵循 [建立 CEV](#) 中的步驟建立 CEV。

步驟 5：準備 CEV 清單檔案

CEV 清單檔案是包含下列項目的 JSON 文件：

- (必要) 您上傳至 Amazon S3 的安裝 .zip 檔案清單。RDS Custom 會依資訊清單中列出的順序套用修補程式。
- (選用) 針對 Oracle 基本目錄、Oracle 主目錄，以及 UNIX/Linux 使用者和群組的識別符和名稱設定非預設值的安裝參數。請注意，您無法修改現有 CEV 或現有資料庫執行個體的安裝參數。當安裝參數具有不同的設定時，您也無法從一個 CEV 升級到另一個 CEV。

如需範例 CEV 清單檔案，請參閱您在 [步驟 1 \(選用\)：下載清單檔案範本](#) 中下載的 JSON 範本。您可以在中檢閱 [CEV 清單檔案範例](#) 中的範例。

主題

- [CEV 資訊清單中的 JSON 欄位](#)
- [建立 CEV 資訊清單](#)
- [CEV 清單檔案範例](#)

CEV 資訊清單中的 JSON 欄位

下列資料表說明資訊清單檔案中的 JSON 欄位。

CEV 資訊清單中的 JSON 欄位

JSON 欄位	描述
MediaImportTemplateVersion	CEV 資訊清單的版本。日期格式為 YYYY-MM-DD 。
databaseInstallationFileNames	資料庫安裝檔案的排序清單。
opatchFileNames	用於 Oracle 資料庫引擎的 OPatch 安裝程式排序清單。只有一個值有效。opatchFileNames 的值必須以 p6880880_ 開頭。
psuRuPatchFileNames	此資料庫的 PSU 和 RU 修補程式。 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>⚠ Important</p> <p>若包含 psuRuPatchFileNames ，則 opatchFileNames 為必要項目。opatchFileNames 的值必須以 p6880880_ 開頭。</p> </div>
OtherPatchFileNames	不在 PSU 和 RU 修補程式清單中的修補程式。RDS Custom 會在套用 PSU 和 RU 修補程式之後套用這些修補程式。 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>⚠ Important</p> <p>若包含 OtherPatchFileNames ，則 opatchFileNames 為必要項目。opatchFileNames 的值必須以 p6880880_ 開頭。</p> </div>
installationParameters	Oracle 基本目錄、Oracle 主目錄，以及 UNIX/Linux 使用者和群組的識別符和名稱的非預設值設定。您可以設定下列參數： code>oracleBase 其下安裝 Oracle 二進位檔案的目錄。它是存放您檔案的二進位磁碟區的掛載點。Oracle 基本目錄可以包含多個 Oracle 主目錄。例如，如果 /home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1 是您的其中

JSON 欄位	描述
	<p>一個 Oracle 主目錄，則 <code>/home/oracle</code> 是 Oracle 基本目錄。使用者指定的 Oracle 基本目錄不是符號連結。</p> <p>如果您未指定 Oracle 基本目錄，則預設目錄為 <code>/rdsdbbin</code>。</p> <p>OracleHome</p> <p>其中安裝 Oracle 資料庫二進位檔案的目錄。例如，如果您將 <code>/home/oracle/</code> 指定為 Oracle 基本目錄，則可將 <code>/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1/</code> 指定為 Oracle 主目錄。使用者指定的 Oracle 主目錄不是符號連結。<code>\$ORACLE_HOME</code> 環境變數會參考 Oracle 主目錄值。</p> <p>如果您未指定 Oracle 主目錄，預設命名格式為 <code>/rdsdbbin/oracle.<i>major-engine-version</i>.custom.r1.<i>engine-edition</i>.1</code>。</p> <p>unixUsername</p> <p>擁有 Oracle 軟體的 UNIX 使用者名稱。RDS Custom 會在執行本機資料庫命令時採用此使用者。如果您同時指定 <code>unixUid</code> 和 <code>unixUsername</code>，則 RDS Custom 會建立使用者 (如果不存在)，然後將 UID 指派給使用者 (如果此 UID 與初始 UID 不同)。</p> <p>預設使用者名為 <code>rdsdb</code>。</p> <p>unixUid</p> <p>擁有 Oracle 軟體的 UNIX 使用者 ID (UID)。如果您同時指定 <code>unixUid</code> 和 <code>unixUsername</code>，則 RDS Custom 會建立使用者 (如果不存在)，然後將 UID 指派給使用者 (如果此 UID 與初始 UID 不同)。</p> <p>預設 UID 為 <code>61001</code>。這是使用者 <code>rdsdb</code> 的 UID。</p> <p>unix GroupName</p> <p>UNIX 群組的名稱。擁有 Oracle 軟體的 UNIX 使用者屬於這個群組。</p>

JSON 欄位	描述
	<p>預設群組名稱為 rdsdb。</p> <p>unix GroupId</p> <p>UNIX 使用者所屬 UNIX 群組的識別符。</p> <p>預設群組 ID 為 1000。這是群組 rdsdb 的 ID。</p>

每個 Oracle 資料庫版本都有不同的支援安裝檔案清單。當您建立 CEV 資訊清單時，請務必指定僅適用於 Oracle 的 RDS 自訂所支援的檔案。否則，CEV 建立會失敗並顯示錯誤。支援[適用於 Oracle 的 Amazon Relational Database Service \(Amazon RDS\) 版本備註](#)中列出的所有修補程式。

建立 CEV 資訊清單

建立 CEV 清單檔案

1. 按照您要套用的順序，列出您計劃套用的所有安裝檔案。
2. 將安裝檔案與 [CEV 資訊清單中的 JSON 欄位](#) 中所述的 JSON 欄位建立關聯。
3. 執行下列任何一項：
 - 將 CEV 清單檔案建立為 JSON 文字檔案。
 - 在主控制台建立 CEV 時，編輯 CEV 清單檔案範本。如需詳細資訊，請參閱 [建立 CEV](#)。

CEV 清單檔案範例

下列範例顯示不同 Oracle 資料庫版本的 CEV 清單檔案。若您在資訊清單中包含 JSON 欄位，請確保不是空的。例如，下列 CEV 清單檔案無效，因為 otherPatchFileNames 空白。

```
{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V982063-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_190000_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [
    "p32126828_190000_Linux-x86-64.zip"
  ]
}
```

```
    ],  
    "otherPatchFileNames": [  
    ]  
  }  
}
```

主題

- [Sample CEV manifest for Oracle Database 12c Release 1 \(12.1\)](#)
- [Sample CEV manifest for Oracle Database 12c Release 2 \(12.2\)](#)
- [Sample CEV manifest for Oracle Database 18c](#)
- [Sample CEV manifest for Oracle Database 19c](#)

Example Oracle 資料庫 12c 版本 1 (12.1) 的 CEV 清單檔案範例

在 Oracle 資料庫 12c 版本 1 (12.1) 的 2021 年 7 月 PSU 範例中，RDS Custom 會指定的順序套用修補程式。因此，RDS Custom 會套用 p32768233，接著套用 p32876425，然後套用 p18759211，以此類推。此範例會針對 UNIX/Linux 使用者和群組，以及 Oracle 主目錄和 Oracle 基本目錄設定新值。

```
{  
  "mediaImportTemplateVersion":"2020-08-14",  
  "databaseInstallationFileNames":[  
    "V46095-01_1of2.zip",  
    "V46095-01_2of2.zip"  
  ],  
  "opatchFileNames":[  
    "p6880880_121010_Linux-x86-64.zip"  
  ],  
  "psuRuPatchFileNames":[  
    "p32768233_121020_Linux-x86-64.zip"  
  ],  
  "otherPatchFileNames":[  
    "p32876425_121020_Linux-x86-64.zip",  
    "p18759211_121020_Linux-x86-64.zip",  
    "p19396455_121020_Linux-x86-64.zip",  
    "p20875898_121020_Linux-x86-64.zip",  
    "p22037014_121020_Linux-x86-64.zip",  
    "p22873635_121020_Linux-x86-64.zip",  
    "p23614158_121020_Linux-x86-64.zip",  
    "p24701840_121020_Linux-x86-64.zip",  
    "p25881255_121020_Linux-x86-64.zip",  
    "p27015449_121020_Linux-x86-64.zip",  
  ]  
}
```

```

    "p28125601_121020_Linux-x86-64.zip",
    "p28852325_121020_Linux-x86-64.zip",
    "p29997937_121020_Linux-x86-64.zip",
    "p31335037_121020_Linux-x86-64.zip",
    "p32327201_121020_Linux-x86-64.zip",
    "p32327208_121020_Generic.zip",
    "p17969866_12102210119_Linux-x86-64.zip",
    "p20394750_12102210119_Linux-x86-64.zip",
    "p24835919_121020_Linux-x86-64.zip",
    "p23262847_12102201020_Linux-x86-64.zip",
    "p21171382_12102201020_Generic.zip",
    "p21091901_12102210720_Linux-x86-64.zip",
    "p33013352_12102210720_Linux-x86-64.zip",
    "p25031502_12102210720_Linux-x86-64.zip",
    "p23711335_12102191015_Generic.zip",
    "p19504946_121020_Linux-x86-64.zip"
  ],
  "installationParameters": {
    "unixGroupName": "dba",
    "unixGroupId": 12345,
    "unixUname": "oracle",
    "unixUid": 12345,
    "oracleHome": "/home/oracle/oracle.12.1.0.2",
    "oracleBase": "/home/oracle"
  }
}

```

Example Oracle 資料庫 12c 版本 2 (12.2) 的 CEV 清單檔案範例

在 Oracle 資料庫 12c 版本 2 (12.2) 的 2021 年 10 月 PSU 範例中，RDS Custom 會套用 p33261817，接著套用 p33192662，然後套用 p29213893，以此類推。此範例會針對 UNIX/Linux 使用者和群組，以及 Oracle 主目錄和 Oracle 基本目錄設定新值。

```

{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V839960-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_122010_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [
    "p33261817_122010_Linux-x86-64.zip"
  ]
}

```



```

    ],
    "otherPatchFileNames": [
      "p33192662_122010_Linux-x86-64.zip",
      "p29213893_122010_Generic.zip",
      "p28730253_122010_Linux-x86-64.zip",
      "p26352615_12201211019DBOCT2021RU_Linux-x86-64.zip",
      "p23614158_122010_Linux-x86-64.zip",
      "p24701840_122010_Linux-x86-64.zip",
      "p25173124_122010_Linux-x86-64.zip",
      "p25881255_122010_Linux-x86-64.zip",
      "p27015449_122010_Linux-x86-64.zip",
      "p28125601_122010_Linux-x86-64.zip",
      "p28852325_122010_Linux-x86-64.zip",
      "p29997937_122010_Linux-x86-64.zip",
      "p31335037_122010_Linux-x86-64.zip",
      "p32327201_122010_Linux-x86-64.zip",
      "p32327208_122010_Generic.zip"
    ],
    "installationParameters": {
      "unixGroupName": "dba",
      "unixGroupId": 12345,
      "unixUname": "oracle",
      "unixUid": 12345,
      "oracleHome": "/home/oracle/oracle.12.2.0.1",
      "oracleBase": "/home/oracle"
    }
  }
}

```

Example Oracle 資料庫 18c 的 CEV 清單檔案範例

在下列 Oracle 資料庫 18c 的 2021 年 10 月 PSU 範例中，RDS Custom 會套用 p32126855，接著套用 p28730253，然後套用 p27539475，依此類推。此範例會針對 UNIX/Linux 使用者和群組，以及 Oracle 主目錄和 Oracle 基本目錄設定新值。

```

{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V978967-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_180000_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [

```

```

    "p32126855_180000_Linux-x86-64.zip"
  ],
  "otherPatchFileNames": [
    "p28730253_180000_Linux-x86-64.zip",
    "p27539475_1813000DBRU_Linux-x86-64.zip",
    "p29213893_180000_Generic.zip",
    "p29374604_1813000DBRU_Linux-x86-64.zip",
    "p29782284_180000_Generic.zip",
    "p28125601_180000_Linux-x86-64.zip",
    "p28852325_180000_Linux-x86-64.zip",
    "p29997937_180000_Linux-x86-64.zip",
    "p31335037_180000_Linux-x86-64.zip",
    "p31335142_180000_Generic.zip"
  ]
  "installationParameters": {
    "unixGroupName": "dba",
    "unixGroupId": 12345,
    "unixUname": "oracle",
    "unixUid": 12345,
    "oracleHome": "/home/oracle/18.0.0.0.ru-2020-10.rur-2020-10.r1",
    "oracleBase": "/home/oracle/"
  }
}

```

Example Oracle 資料庫 19c 的 CEV 清單檔案範例

在下列 Oracle 資料庫 19c 的範例中，RDS Custom 會套用 p32126828，接著套用 p29213893，然後套用 p29782284，依此類推。此範例會針對 UNIX/Linux 使用者和群組，以及 Oracle 主目錄和 Oracle 基本目錄設定新值。

```

{
  "mediaImportTemplateVersion": "2020-08-14",
  "databaseInstallationFileNames": [
    "V982063-01.zip"
  ],
  "opatchFileNames": [
    "p6880880_190000_Linux-x86-64.zip"
  ],
  "psuRuPatchFileNames": [
    "p32126828_190000_Linux-x86-64.zip"
  ],
  "otherPatchFileNames": [
    "p29213893_1910000DBRU_Generic.zip",

```

```
    "p29782284_1910000DBRU_Generic.zip",
    "p28730253_190000_Linux-x86-64.zip",
    "p29374604_1910000DBRU_Linux-x86-64.zip",
    "p28852325_190000_Linux-x86-64.zip",
    "p29997937_190000_Linux-x86-64.zip",
    "p31335037_190000_Linux-x86-64.zip",
    "p31335142_190000_Generic.zip"
  ],
  "installationParameters": {
    "unixGroupName": "dba",
    "unixGroupId": 12345,
    "unixUname": "oracle",
    "unixUid": 12345,
    "oracleHome": "/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1",
    "oracleBase": "/home/oracle"
  }
}
```

步驟 6 (選用)：驗證 CEV 清單檔案

或者，藉由執行 `json.tool` Python 指令碼來確認資訊清單是有效的 JSON 檔案。例如，如果您變更為包含名為 `manifest.json` 之 CEV 資訊清單的目錄，則執行下列命令。

```
python -m json.tool < manifest.json
```

步驟 7：新增必要的 IAM 許可

請確保建立 CEV 的 IAM 主體具有 [步驟 5：將必要許可授予 IAM 使用者或角色](#) 中所述的必要政策。

建立 CEV

您可以使用 AWS Management Console 或建立 CEV。AWS CLI 指定多租戶或非多租戶架構。如需詳細資訊，請參閱 [多租戶架構考量](#)。

一般而言，建立 CEV 大約需要兩個小時。在建立 CEV 之後，您可以使用它來建立 RDS Custom 資料庫執行個體。如需詳細資訊，請參閱 [建立 RDS Custom for Oracle 資料庫執行個體](#)。

請注意下列建立 CEV 的需求和限制：

- 包含安裝檔案的 Amazon S3 儲存貯體必須與您的 CEV 位於 AWS 區域相同的位置。否則，建立程序會失敗。
- CEV 名稱的格式 `major-engine-version.customized_string` 必須如下所示。19.cdb_cev1

- CEV 名稱必須包含 1—50 個英數字元、底線、破折號或句號。
- CEV 名稱不能包含連續的句點，如中 19..cdb_cev1 所示。

主控台

若要建立 CEV

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。

Custom engine versions (自訂引擎版本) 頁面會顯示目前存在的所有 CEV。如果您尚未建立任何 CEV，則頁面為空白。


3. 選擇 Create custom engine version (建立自訂引擎版本)。
4. 在引擎選項中，執行下列動作：
 - a. 針對 Engine type (引擎類型)，請選擇 Oracle。
 - b. 針對架構設定，選擇性地選擇性地選擇「多租戶架構」來建立使用資料庫引擎或的 Oracle 多租戶 CEV。custom-oracle-ee-cdb custom-oracle-se2-cdb 您只能使用多租戶 CEV 建立 RDS Custom for Oracle CDB。如果您不選擇此選項，您的 CEV 是非 CDB，它使用引擎或 custom-oracle-ee custom-oracle-se2

Note

您選擇的架構是 CEV 的永久特性。您無法將 CEV 修改為稍後使用不同的架構。

- c. 請選擇下列其中一個選項：
 - 建立新 CEV - 從頭開始建立 CEV。在此情況下，您必須指定 JSON 清單檔案，其會指定資料庫二進位檔案。
 - 從來源建立 CEV - 在指定要複製的 CEV 中，選擇要用作來源 CEV 的現有 CEV。在此情況下，您可以指定新的 Amazon Machine Image (AMI)，但無法指定不同的資料庫二進位檔案。
 - d. 在引擎版本中，選擇主要引擎版本。
5. 在版本詳細資訊中，執行下列動作：
 - a. 在自訂引擎版本名稱中輸入有效名稱。例如，您輸入名稱 **19.cdb_cev1**。

- b. (選用) 輸入 CEV 的描述。
6. 在安裝媒體中，執行下列動作：
 - a. (選用) 針對 AMI ID，將欄位保留空白以使用最新服務提供的 AMI，或輸入您先前用來建立 CEV 的 AMI。若要取得有效的 AMI ID，請使用下列任一技巧：
 - 在主控台中，選擇左側導覽窗格中的自訂引擎版本，然後選擇 CEV 的名稱。CEV 使用的 AMI ID 會出現在組態索引標籤中。
 - 在中 AWS CLI，使用指 `describe-db-engine-versions` 令。搜尋輸出找出 ImageID。
 - b. 針對 S3 location of manifest files (資訊清單檔案的 S3 位置)，輸入您在 [步驟 3：將您的安裝檔案上傳到 Amazon S3](#) 中指定之 Amazon S3 儲存貯體的位置。例如，輸入 `s3://my-custom-installation-files/123456789012/cev1/`。

 Note

您 AWS 區域 在其中建立 CEV 的區域必須與 S3 儲存貯體位於相同的區域。

- c. (僅建立新的 CEV) 對於 CEV 清單檔案，輸入您已在 [建立 CEV 資訊清單](#) 建立的 JSON 清單檔案。
7. 在 KMS 金鑰區段中，選取 [輸入金鑰 ARN] 以列出可用的金 AWS KMS 鑰。然後從清單選取 KMS 金鑰。

RDS 自訂需要 AWS KMS 金鑰。如需詳細資訊，請參閱 [步驟 1：建立或重複使用對稱加密 AWS KMS 金鑰](#)。
 8. (選用) 選擇新增標籤，為 CEV 建立鍵值組。
 9. 選擇 Create custom engine version (建立自訂引擎版本)。

如果 JSON 清單檔案的格式無效，主控台會顯示驗證 CEV 清單檔案時發生錯誤。修正問題，然後重試。

Custom engine versions (自訂引擎版本) 頁面隨即出現。您的 CEV 會顯示狀態 Creating (正在建立)。建立 CEV 的程序大約需要兩個小時。

AWS CLI

若要使用建立 CEV AWS CLI，請執行 [create-custom-db-engine-version](#) 命令。

下列選項是必要的：

- `--engine`— 指定引擎類型。若為 CDB，請指定 `custom-oracle-ee-cdb` 或 `custom-oracle-se2-cdb`。對於非 CDB，請指定 `custom-oracle-ee`。您只能從使用 `custom-oracle-ee-cdb` 用或建立的 CEV 建立 CDB。您只能從使用或建立的 CEV 建立非 CDB。 `custom-oracle-ee` `custom-oracle-se2`
- `--engine-version` - 指定引擎版本。格式為 *major-engine-version*。####。CEV 名稱必須包含 1—50 個英數字元、底線、破折號或句號。CEV 名稱不能包含連續的句點，如 `19..cdb_cev1` 所示。
- `--kms-key-id`— 指定 AWS KMS key。
- `--manifest` - 指定 *manifest_json_string* 或 `--manifest file:file_name`。 *manifest_json_string* 中不允許換行字元。請務必轉義 JSON 程式碼中的雙引號 (")，方法是在引號前加上反斜杠 (\)。

下列範例為 19c 顯示了來自 [步驟 5：準備 CEV 清單檔案](#) 的 *manifest_json_string*。此範例會針對 Oracle 基本目錄、Oracle 主目錄，以及 UNIX/Linux 使用者和群組的識別符和名稱設定新值。如果複製此字串，請先移除所有換行字元，然後再將其貼到指令中。

```
{\"mediaImportTemplateVersion\": \"2020-08-14\",
\"databaseInstallationFileNames\": [\"V982063-01.zip\"],
\"opatchFileNames\": [\"p6880880_190000_Linux-x86-64.zip\"],
\"psuRuPatchFileNames\": [\"p32126828_190000_Linux-x86-64.zip\"],
\"otherPatchFileNames\": [\"p29213893_1910000DBRU_Generic.zip\",
\"p29782284_1910000DBRU_Generic.zip\", \"p28730253_190000_Linux-
x86-64.zip\", \"p29374604_1910000DBRU_Linux-x86-64.zip\",
\"p28852325_190000_Linux-x86-64.zip\", \"p29997937_190000_Linux-x86-64.zip
\", \"p31335037_190000_Linux-x86-64.zip\", \"p31335142_190000_Generic.zip
\"]\"installationParameters\":{ \"unixGroupName\": \"dba\",
\"unixUsername\": \"oracle\", \"oracleHome\": \"/home/oracle/
oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1\", \"oracleBase\": \"/
home/oracle/\"}}"
```

- `--database-installation-files-s3-bucket-name` - 指定您已在 [步驟 3：將您的安裝檔案上傳到 Amazon S3](#) 中指定的相同儲存貯體名稱。您執行的區域 `create-custom-db-engine-version` 必須與 Amazon S3 儲存貯體所 AWS 區域 在的區域相同。

您還可以指定下列選項：

- `--description` - 指定 CEV 的描述。
- `--database-installation-files-s3-prefix` - 指定您已在 [步驟 3：將您的安裝檔案上傳到 Amazon S3](#) 中指定的資料夾名稱。
- `--image-id` - 指定要重複使用的 AMI ID。若要尋找有效的 ID，請執行 `describe-db-engine-versions` 命令，然後搜尋輸出找出 ImageID。根據預設，RDS Custom for Oracle 會使用最新的可用 AMI。

下列範例會建立名為 `19.cdb_cev1` 的 Oracle 多租戶 CEV。此範例會重複使用現有的 AMI，而不是使用最新的可用 AMI。請確保 CEV 的名稱以主要引擎版本號為開頭。

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-custom-db-engine-version \  
  --engine custom-oracle-se2-cdb \  
  --engine-version 19.cdb_cev1 \  
  --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-  
installation-files \  
  --database-installation-files-s3-prefix 123456789012/cev1 \  
  --kms-key-id my-kms-key \  
  --description "test cev" \  
  --manifest manifest_string \  
  --image-id ami-012a345678901bcde
```

在 Windows 中：

```
aws rds create-custom-db-engine-version ^  
  --engine custom-oracle-se2-cdb ^  
  --engine-version 19.cdb_cev1 ^  
  --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-  
installation-files ^  
  --database-installation-files-s3-prefix 123456789012/cev1 ^  
  --kms-key-id my-kms-key ^  
  --description "test cev" ^  
  --manifest manifest_string ^  
  --image-id ami-012a345678901bcde
```

Example

使用 `describe-db-engine-versions` 命令取得 CEV 的詳細資訊。

```
aws rds describe-db-engine-versions \  
  --engine custom-oracle-se2-cdb \  
  --include-all
```

下列部分輸出顯示引擎、參數群組、清單檔案，以及其他資訊。

```
{  
  "DBEngineVersions": [  
    {  
      "Engine": "custom-oracle-se2-cdb",  
      "EngineVersion": "19.cdb_cev1",  
      "DBParameterGroupFamily": "custom-oracle-se2-cdb-19",  
      "DBEngineDescription": "Containerized Database for Oracle Custom SE2",  
      "DBEngineVersionDescription": "test cev",  
      "Image": {  
        "ImageId": "ami-012a345678901bcde",  
        "Status": "active"  
      },  
      "ValidUpgradeTarget": [],  
      "SupportsLogExportsToCloudwatchLogs": false,  
      "SupportsReadReplica": true,  
      "SupportedFeatureNames": [],  
      "Status": "available",  
      "SupportsParallelQuery": false,  
      "SupportsGlobalDatabases": false,  
      "MajorEngineVersion": "19",  
      "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-custom-  
installation-files",  
      "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",  
      "DBEngineVersionArn": "arn:aws:rds:us-east-1:123456789012:cev:custom-  
oracle-se2-cdb/19.cdb_cev1/abcd12e3-4f5g-67h8-i9j0-k1234156m789",  
      "KMSKeyId": "arn:aws:kms:us-  
east-1:732027699161:key/1ab2345c-6d78-9ef0-1gh2-3456i7j89k01",  
      "CreateTime": "2023-03-07T19:47:58.131000+00:00",  
      "TagList": [],  
      "SupportsBabelfish": false,  
      ...  
    }  
  ]  
}
```


無法建立 CEV

如果 CEV 建立的程序失敗，會發生 RDS Custom 問題 RDS-EVENT-0198 並且有訊息 `Creation failed for custom engine version major-engine-version.cev_name`，其中包含失敗的詳細資訊。例如，事件會列印遺失的檔案。

您無法修改失敗的 CEV。您只能刪除它，然後在修正失敗原因之後再試一次建立 CEV。如需有關 CEV 建立失敗原因進行疑難排解的詳細資訊，請參閱 [針對 RDS Custom for Oracle 的自訂引擎版本建立進行疑難排解](#)。

修改 CEV 狀態

您可以使用 AWS Management Console 或修改 CEV。AWS CLI 您可以修改 CEV 描述或其可用狀態。您的 CEV 有下列其中一個狀態值：

- `available` – 您可以使用此 CEV 建立新的 RDS Custom 資料庫執行個體或升級資料庫執行個體。這是新建立的 CEV 的預設狀態。
- `inactive` – 您無法使用此 CEV 建立或升級 RDS Custom 執行個體。您無法還原資料庫快照以使用此 CEV 建立新的 RDS Custom 資料庫執行個體。

您可以將 CEV 從任何支援狀態變更為任何其他支援狀態。您可以變更狀態，以防止意外使用 CEV，或讓已停產的 CEV 符合再次使用的資格。例如，您可以將 CEV 的狀態從 `available` 變更為 `inactive`，以及從 `inactive` 變更回 `available`。

主控台

若要修改 CEV

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。
3. 選擇您想要修改其描述或狀態的 CEV。
4. 在 Actions (動作) 中，選擇 Modify (修改)。
5. 進行下列任何變更：
 - 針對 CEV status settings (CEV 狀態設定)，選擇新的可用性狀態。
 - 針對 Version description (版本描述)，輸入新的描述。
6. 選擇 Modify CEV (修改 CEV)。

如果 CEV 正在使用中，主控台會顯示您無法修改 CEV 狀態。修正問題，然後重試。

Custom engine versions (自訂引擎版本) 頁面隨即出現。

AWS CLI

若要使用修改 CEV AWS CLI，請執行 [modify-custom-db-engine-version](#) 命令。您可以執行 [describe-db-engine-versions](#) 指令來尋找要修改的 CEV。

下列選項是必要的：

- `--engine engine-type`，其中####為custom-oracle-ee、custom-oracle-se2、custom-oracle-ee-cdb或 custom-oracle-se2-cdb
- `--engine-version cev`，其中 *cev* 是您要修改的自訂引擎版本的名稱。
- `--status status`，其中 *status* 是您要指派給 CEV 的可用性狀態。

下列範例會將名為 19.my_cev1 的 CEV 從目前的狀態變更為 inactive。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-custom-db-engine-version \  
  --engine custom-oracle-se2 \  
  --engine-version 19.my_cev1 \  
  --status inactive
```

在 Windows 中：

```
aws rds modify-custom-db-engine-version ^  
  --engine custom-oracle-se2 ^  
  --engine-version 19.my_cev1 ^  
  --status inactive
```

檢視 CEV 詳細資訊

您可以使用 AWS Management Console 或檢視 CEV 資訊清單的詳細資訊，以及用來建立 CEV 之命令的詳細資料。AWS CLI

主控台

檢視 CEV 詳細資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。

Custom engine versions (自訂引擎版本) 頁面會顯示目前存在的所有 CEV。如果您尚未建立任何 CEV，則頁面為空白。

3. 選擇要檢視的 CEV 名稱。
4. 選擇 Configuration (組態) 以檢視清單檔案中指定的安裝參數。

Configuration	Databases	Snapshots	Manifest
<h3>Configuration</h3>			
Edition Oracle Enterprise Edition	Amazon Resource Name (ARN) arn:aws:rds:us-west-2:██████████:custom-engine-version:██████████	DB installation parameters	
Major Version 19	KMS key ID ██████████	Oracle Base Directory /rdsdbbin	
Installation files location s3://██████████-us-west-2-██████████-aws-custom-engine-versions-██████████-oracle-19.0.0-2020-04		Oracle Home Directory /rdsdbbin/oracle.19.custom.r1.EE.1	
		Oracle User Name rdsdb	
		Oracle UID 61001	
		Oracle Group Name rdsdb	
		Oracle GID 1000	

5. 選擇 Manifest (清單檔案) 以檢視在 `create-custom-db-engine-version` 命令的 `--manifest` 選項中指定的安裝參數。您可以複製此文字、視需要取代這些值，然後在新命令中使用它們。

Configuration	Databases	Snapshots	Automated Backups	Tags	Manifest
<p>CEV manifest Copy</p> <pre>--manifest "{\"databaseInstallationFileNames\": [\"V982063-01.zip\"], \"mediaImportTemplateVersion\": \"2020-08-14\", \"opatchFileNames\": [\"p6880880_190000_1220119_Linux-x86-64.zip\"], \"psuRuPatchFileNames\": [\"p30783543_190000_Linux-x86-64.zip\", \"p30528704_197000DBRU_Linux-x86-64.zip\", \"p29213893_197000DBRU_Generic.zip\", \"p28730253_190000_Linux-x86-64.zip\", \"p28852325_190000_Linux-x86-64.zip\", \"p29997937_190000_Linux-x86-64.zip\", \"p29997959_190000_Generic.zip\"], \"installationParameters\": {\"oracleHome\": \"/rdsdbbin/oracle.19.custom.r1.EE.1\", \"oracleBase\": \"/rdsdbbin\", \"unixUid\": \"61001\", \"unixUsername\": \"rdsdb\", \"unixGroupId\": \"1000\", \"unixGroupName\": \"rdsdb\"}}"</pre>					

AWS CLI

若要使用檢視 CEV 的詳細資料 AWS CLI，請執行命 [describe-db-engine-versions](#) 令。

下列選項是必要的：

- `--engine engine-type`，其中####為 custom-oracle-ee、custom-oracle-se2、custom-oracle-ee-cdb 或 custom-oracle-se2-cdb
- `--engine-version major-engine-version.customized_string`

下列範例會建立使用企業版的非 CDB CEV。CEV 名稱以主要引擎版本號 19.my_cev1 開頭，這是必要的。

Example

對於 Linux macOS、或 Unix：

```
aws rds describe-db-engine-versions \
  --engine custom-oracle-ee \
  --engine-version 19.my_cev1
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^
  --engine custom-oracle-ee ^
  --engine-version 19.my_cev1
```

下列部分輸出顯示引擎、參數群組、清單檔案，以及其他資訊。

```

"DBEngineVersions": [
  {
    "Engine": "custom-oracle-ee",
    "MajorEngineVersion": "19",
    "EngineVersion": "19.my_cev1",
    "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-cev-customer-
installation-files",
    "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
    "CustomDBEngineVersionManifest": "{\n\"mediaImportTemplateVersion\":
\n\"2020-08-14\", \n\"databaseInstallationFileNames\": [\n\"V982063-01.zip\"\n],
\n\"installationParameters\": {\n\"oracleBase\": \"\"/tmp\", \n\"oracleHome\": \"\"/
tmp/Oracle\"\n}, \n\"opatchFileNames\": [\n\"p6880880_190000_Linux-x86-64.zip
\"\n], \n\"psuRuPatchFileNames\": [\n\"p32126828_190000_Linux-x86-64.zip
\"\n], \n\"otherPatchFileNames\": [\n\"p29213893_1910000DBRU_Generic.zip\", \n
\n\"p29782284_1910000DBRU_Generic.zip\", \n\"p28730253_190000_Linux-x86-64.zip\", \n
\n\"p29374604_1910000DBRU_Linux-x86-64.zip\", \n\"p28852325_190000_Linux-x86-64.zip\",
\n\"p29997937_190000_Linux-x86-64.zip\", \n\"p31335037_190000_Linux-x86-64.zip\", \n
\n\"p31335142_190000_Generic.zip\"\n]\n}\n",
    "DBParameterGroupFamily": "custom-oracle-ee-19",
    "DBEngineDescription": "Oracle Database server EE for RDS Custom",
    "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-
ee/19.my_cev1/0a123b45-6c78-901d-23e4-5678f901fg23",
    "DBEngineVersionDescription": "test",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/ab1c2de3-f4g5-6789-h012-
h3ijk4567l89",
    "CreateTime": "2022-11-18T09:17:07.693000+00:00",
    "ValidUpgradeTarget": [
      {
        "Engine": "custom-oracle-ee",
        "EngineVersion": "19.cev.2021-01.09",
        "Description": "test",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
      }
    ]
  }
]

```

刪除 CEV

您可以使用 AWS Management Console 或刪除 CEV。AWS CLI 通常需要幾分鐘的時間。

若要刪除 CEV，下列任何一項都無法使用該 CEV：

- RDS Custom 資料庫執行個體

- RDS Custom 資料庫執行個體的快照
- RDS Custom 資料庫執行個體的自動備份

主控台

若要刪除 CEV

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。
3. 選擇您想要刪除其描述或狀態的 CEV。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。

Delete **cev_name**? (是否刪除 cev_name?) 對話方塊隨即顯示。

5. 輸入 **delete me**，然後選擇 Delete (刪除)。

在 Custom engine versions (自訂引擎版本) 頁面上，橫幅會顯示您的 CEV 即將刪除。

AWS CLI

若要使用刪除 CEV AWS CLI，請執行 [delete-custom-db-engine-version](#) 命令。

下列選項是必要的：

- `--engine engine-type`，其中####為custom-oracle-ee、custom-oracle-se2、custom-oracle-ee-cdb或 custom-oracle-se2-cdb
- `--engine-version cev`，其中 *CEV* 是要刪除之自訂引擎版本的名稱

以下範例會刪除名為 19.my_cev1 的 CEV。

Example

對於LinuxmacOS、或Unix：

```
aws rds delete-custom-db-engine-version \  
  --engine custom-oracle-ee \  
  --engine-version 19.my_cev1
```

在 Windows 中：

```
aws rds delete-custom-db-engine-version ^  
  --engine custom-oracle-ee ^  
  --engine-version 19.my_cev1
```

設定 Amazon RDS Custom for Oracle 的資料庫執行個體

您可以建立 RDS Custom 資料庫執行個體，然後使用安全通訊殼層 (SSH) 或 AWS Systems Manager 與它連線。

主題

- [多租戶架構考量](#)
- [建立 RDS Custom for Oracle 資料庫執行個體](#)
- [RDS Custom 服務連結角色](#)
- [使用工作階段管理員連線至 RDS Custom 資料庫執行個體](#)
- [使用 SSH 連線到您的 RDS Custom 資料庫執行個體](#)
- [以 SYS 身分登入您的 RDS Custom for Oracle 資料庫](#)
- [在 RDS Custom for Oracle 資料庫執行個體上安裝其他軟體元件](#)

多租戶架構考量

如果您使用 Oracle 多租戶架構 (custom-oracle-ee-cdb或custom-oracle-se2-cdb引擎類型) 建立適用於 Oracle 資料庫的 Amazon RDS 自訂執行個體，則您的資料庫就是容器資料庫 (CDB)。如果您未指定 Oracle 多租戶架構，則資料庫是使用custom-oracle-ee或custom-oracle-se2引擎類型的傳統非 CDB。非 CDB 不能包含插入式資料庫 (PDB)。如需詳細資訊，請參閱 [Amazon RDS Custom for Oracle 的資料庫架構](#)。

當您建立 RDS Custom for Oracle CDB 執行個體時，請考慮下列事項：

- 您只能從 Oracle Database 19c CEV 建立多租戶資料庫。
- 您只能在 CEV 使用custom-oracle-ee-cdb或custom-oracle-se2-cdb引擎類型時建立 CDB 執行個體。
- 如果您使用標準版 2 建立 CDB 執行個體，CDB 最多可以包含 3 個 PDB。
- 依預設，您的 CDB 會命名為 RDSCDB (其也是 Oracle 系統 ID (Oracle SID) 的名稱)。您可以選擇不同名稱。
- 您的 CDB 只包含一個起始 PDB。PDB 名稱預設為 ORCL。您可以針對起始 PDB 選擇不同的名稱，但 Oracle SID 和 PDB 名稱不能相同。
- RDS Custom for Oracle 不會為 PDB 提供 API。若要建立其他 PDB，請使用 Oracle SQL 命令 CREATE PLUGGABLE DATABASE。RDS Custom for Oracle 不會限制您可以建立的 PDB 數目。通常，您負責建立和管理 PDB，如同內部部署中一般。

- 您無法使用 RDS API 建立、修改和刪除 PDB：您必須使用 Oracle SQL 陳述式。當您使用 Oracle SQL 建立 PDB 時，建議您稍後建立手動快照，以防您需要執行 point-in-time 復原 (PITR)。
- 您無法使用 Amazon RDS API 重新命名現有的 PDB。您也不能使用 `modify-db-instance` 命令重新命名 CDB。
- CDB 根的開放模式在主要資料庫上為 READ WRITE，而在掛載的待命資料庫上則為 MOUNTED。RDS Custom for Oracle 嘗試在開啟 CDB 時開啟所有 PDB。如果 RDS Custom for Oracle 無法開啟所有 PDB，它會發出事件 `tenant database shutdown`。

建立 RDS Custom for Oracle 資料庫執行個體

使用 AWS Management Console 或建立適用於 Oracle 資料庫執行個體的 Amazon RDS 自訂 AWS CLI。該程序與建立 Amazon RDS 資料庫執行個體的程序類型。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

如果您在 CEV 清單檔案中包含安裝參數，則資料庫執行個體會使用您指定的 Oracle 基本目錄、Oracle 主目錄，以及 UNIX/Linux 使用者和群組的識別符和名稱。Oracle Database 在安裝期間建立的 `oratab` 檔案會指向真實的安裝位置，而不是符號連結。當 RDS Custom for Oracle 執行命令時，其會以設定的作業系統使用者身分執行，而非預設使用者 `rdsdb`。如需詳細資訊，請參閱 [步驟 5：準備 CEV 清單檔案](#)。

在您嘗試建立或連線至 RDS Custom 資料庫執行個體之前，請先完成 [設定您的 Amazon RDS Custom for Oracle 的環境](#) 中的任務。


主控台

若要建立 RDS Custom for Oracle 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇 Create database (建立資料庫)。
4. 在 Choose a database creation method (選擇資料庫建立方法) 中，選取 Standard Create (標準建立)。
5. 在引擎選項區段中，執行下列動作：
 - a. 針對 Engine type (引擎類型)，請選擇 Oracle。
 - b. 針對 Database management type (引擎管理類型)，選擇 Amazon RDS Custom。

c. 對於管理設定，執行下列其中一項動作：

- 選取多租戶架構以建立容器資料庫 (CDB)。建立時，您的 CDB 包含一個 PDB 種子和一個起始 PDB。

 Note

僅 Oracle Database 19c 支援多租戶架構設定。

- 清除多租戶架構以建立非 CDB。非 CDB 不能包含 PDB。

d. 對於「版本」，請選擇「Oracle 企業版」或「Oracle 標準版 2」。

e. 對於自訂引擎版本，選擇現有的 RDS Custom 自訂引擎版本 (CEV)。CEV 採用下列格式：*major-engine-version.customized_string*。範例識別符為 19.cdb_cev1。

如果您在上一個步驟中選擇了多租戶架構，則只能指定使用 custom-oracle-ee-cdb 或 custom-oracle-se2-cdb 引擎類型的 CEV。主控台會篩選出使用不同引擎類型建立的 CEV。

6. 在 Templates (範本) 中，選擇 Production (生產)。

7. 在 Settings (設定) 區段中，執行下列動作：

- a. 對於資料庫執行個體識別符，輸入資料庫執行個體的唯一名稱。
- b. 對於主要使用者名稱，輸入使用者名稱。您稍後可以從主控台擷取此值。

當您連線至非 CDB 時，主要使用者就是非 CDB 的使用者。當您連線至 CDB 時，主要使用者就是 PDB 的使用者。若要連線至 CDB 根，請登入主機、啟動 SQL 用戶端，然後使用 SQL 命令建立管理使用者。

c. 清除自動產生密碼。

8. 選擇資料庫執行個體類別。

如需支援的類別，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#)。

9. 在 Storage (儲存) 區段中，執行下列動作：

a. 對於儲存體類型，選擇 SSD 類型：io1、gp2 或 gp3。您具有下列其他選項：

- 對於 io1 或 gp3，選擇佈建 IOPS 的速率。對於 io1，預設為 1000，而對於 gp3，預設為 12000。
- 對於 gp3，選擇儲存體輸送量的速率。預設值為 500 MiBps。

- b. 對於已配置的儲存體，選擇儲存體大小。預設為 40 GiB。
10. 對於連線，指定您的虛擬私有雲端 (VPC)、資料庫子網路群組和 VPC 安全群組 (防火牆)。
 11. 針對 RDS Custom security (RDS Custom 安全)，請執行下列操作：

- a. 針對 IAM instance profile (IAM 執行個體設定檔)，為您的 RDS Custom for Oracle 資料庫執行個體選擇執行個體設定檔。


例如 `AWSRDSCustom`，IAM 執行個體設定檔必須以開頭 `AWSRDSCustomInstanceProfileForRdsCustomInstance`。

- b. 在加密中，選擇輸入金鑰 ARN 以列出可用的 AWS KMS 金鑰。然後從清單選擇您的金鑰。

RDS 自訂需要 AWS KMS 金鑰。如需詳細資訊，請參閱 [步驟 1：建立或重複使用對稱加密 AWS KMS 金鑰](#)。


12. 對於資料庫選項，執行下列動作：

- a. (選用) 在系統 ID (SID) 中，輸入 Oracle SID 的值，也是 CDB 的名稱。SID 是管理資料庫檔案之 Oracle 資料庫執行個體的名稱。在此內容中，「Oracle 資料庫執行個體」一詞特指系統全區域 (SGA) 和 Oracle 背景程序。如果您未指定 SID，值會預設為 **RDSCDB**。
- b. (選用) 對於初始資料庫名稱，輸入一個名稱。預設值為 **ORCL**。在多租戶架構中，初始資料庫名稱是 PDB 名稱。

 Note

SID 和 PDB 名稱必須不同。

- c. 對於「選項」群組，請選擇選項群組或接受預設值。

 Note

RDS 自定義為甲骨文唯一支持的選項是 Timezone。如需詳細資訊，請參閱 [Oracle 時區](#)。

- d. 對於備份保留期，選擇一值。您無法選擇 0 天。
- e. 在其餘區段，指定您偏好的 RDS Custom 資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。以下設定不會出現在主控台中，也不受支援：

- 處理器功能

- 儲存體自動調整規模
- Database authentication (資料庫身分驗證) 中的 Password and Kerberos authentication (密碼和 Kerberos 身分驗證) 選項 (僅支援 Password authentication (密碼身分驗證))
- Performance Insights (績效詳情)
- Log exports (日誌匯出)
- Enable auto minor version upgrade (啟用自動次要版本升級)
- 刪除保護

13. 選擇建立資料庫。

Important

建立 RDS Custom for Oracle 資料庫執行個體時，您可能會收到下列錯誤：服務連結角色正在建立的程序中。請稍後再試。如果收到這個訊息，請等待幾分鐘的時間然後再次嘗試建立資料庫執行個體。

View credential details (檢視憑證詳細資訊) 按鈕會顯示在 Databases (資料庫) 頁面上。

若要檢視 RDS Custom 資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視憑證詳細資訊)。

若要以主要使用者的身分連線至資料庫執行個體，請使用出現的使用者名稱和密碼。

Important

您無法再次於主控台中檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。若要在 RDS Custom 資料庫執行個體可供使用之後變更主要使用者密碼，請登入資料庫並執行 ALTER USER 命令。您無法使用主控台內的修改選項來重設密碼。

14. 選擇 Databases (資料庫) 以檢視 RDS Custom 資料庫執行個體的清單。

15. 選擇您剛剛建立的 RDS Custom 資料庫執行個體。

在 RDS 主控台上，新 RDS Custom 資料庫執行個體的詳細資訊即會出現：

- 在 RDS Custom 資料庫執行個體建立完成且可供使用之前，資料庫執行個體會處於 Creating (建立中) 狀態。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視執行個體類別和分配的儲存空間而定，新的資料庫執行個體可能需要幾分鐘才能使用。

- Role (角色) 具有 Instance (RDS Custom) (執行個體 (RDS Custom)) 值。
- RDS Custom automation mode (RDS Custom 自動化模式) 具有 Full automation (全自動化) 值。此設定表示資料庫執行個體提供自動監控和執行個體復原。

AWS CLI

您可以使用[create-db-instance](#) AWS CLI 命令建立 RDS 自訂資料庫執行個體。

下列選項是必要的：

- `--db-instance-identifier`
- `--db-instance-class` (如需支援執行個體類別的清單，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#))
- `--engine engine-type`，其中####為custom-oracle-ee、custom-oracle-se2、custom-oracle-ee-cdb或 custom-oracle-se2-cdb
- `--engine-version cev` (其中 *cev* 是您在 [建立 CEV](#) 中指定的自訂引擎版本的名稱)
- `--kms-key-id my-kms-key`
- `--backup-retention-period days` (其中 *days* 是大於 0 的值)
- `--no-auto-minor-version-upgrade`
- `--custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1` (其中 *region* 是您要在其中建立資料庫執行個體的 AWS 區域)

下列範例會建立名為 my-cfo-cdb-instance 的 RDS Custom 資料庫執行個體。資料庫是具有非預設名稱 *MYCDB* 的 CDB。非預設 PDB 名為 *MYPDB*。備份保留期為 3 天。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --engine custom-oracle-ee-cdb \  
  --db-instance-identifier my-cfo-cdb-instance \  
  --engine-version 19.cdb_cev1 \  
  --db-name MYPDB \  
  --db-system-id MYCDB \  
  --allocated-storage 250 \  
  --no-auto-minor-version-upgrade
```

```
--db-instance-class db.m5.xlarge \  
--db-subnet-group mydbsubnetgroup \  
--master-username myuser \  
--master-user-password mypassword \  
--backup-retention-period 3 \  
--port 8200 \  
--kms-key-id my-kms-key \  
--no-auto-minor-version-upgrade \  
--custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --engine custom-oracle-ee-cdb ^  
  --db-instance-identifier my-cfo-cdb-instance ^  
  --engine-version 19.cdb_cev1 ^  
  --db-name MYPDB ^  
  --db-system-id MYCDB ^  
  --allocated-storage 250 ^  
  --db-instance-class db.m5.xlarge ^  
  --db-subnet-group mydbsubnetgroup ^  
  --master-username myuser ^  
  --master-user-password mypassword ^  
  --backup-retention-period 3 ^  
  --port 8200 ^  
  --kms-key-id my-kms-key ^  
  --no-auto-minor-version-upgrade ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

使用 `describe-db-instances` 命令取得執行個體的詳細資訊。

Example

```
aws rds describe-db-instances --db-instance-identifier my-cfo-cdb-instance
```

下列部分輸出顯示引擎、參數群組及其他資訊。

```
{
  "DBInstanceIdentifier": "my-cfo-cdb-instance",
  "DBInstanceClass": "db.m5.xlarge",
  "Engine": "custom-oracle-ee-cdb",
  "DBInstanceStatus": "available",
  "MasterUsername": "admin",
  "DBName": "MYPDB",
  "DBSystemID": "MYCDB",
  "Endpoint": {
    "Address": "my-cfo-cdb-instance.abcdefghijkl.us-
east-1.rds.amazonaws.com",
    "Port": 1521,
    "HostedZoneId": "A1B2CDEFGH34IJ"
  },
  "AllocatedStorage": 100,
  "InstanceCreateTime": "2023-04-12T18:52:16.353000+00:00",
  "PreferredBackupWindow": "08:46-09:16",
  "BackupRetentionPeriod": 7,
  "DBSecurityGroups": [],
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-0a1bcd2e",
      "Status": "active"
    }
  ],
  "DBParameterGroups": [
    {
      "DBParameterGroupName": "default.custom-oracle-ee-cdb-19",
      "ParameterApplyStatus": "in-sync"
    }
  ],
  ...
}
```

RDS Custom 服務連結角色

服務連結角色可讓 Amazon RDS 自訂存取您 AWS 帳戶的。它會讓使用 RDS Custom 更簡單，因為您不需要手動新增必要許可。RDS Custom 會定義其服務連結角色的許可，除非另有定義，否則僅有 RDS Custom 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能連接到任何其他 IAM 實體。

當您建立 RDS Custom 資料庫執行個體時，Amazon RDS 和 RDS Custom 服務連結角色兩者都會建立 (如果尚未存在) 並且使用。如需詳細資訊，請參閱 [使用 Amazon RDS 的服務連結角色](#)。

第一次建立 RDS Custom for Oracle 資料庫執行個體時，您可能會收到下列錯誤：服務連結角色正在建立的程序中。請稍後再試。如果收到這個訊息，請等待幾分鐘的時間然後再次嘗試建立資料庫執行個體。

使用工作階段管理員連線至 RDS Custom 資料庫執行個體

建立 RDS 自訂資料庫執行個體之後，您可以使用 AWS Systems Manager Session Manager。當您的資料庫執行個體無法公開存取時，這是偏好的技術。

工作階段管理員可讓您透過瀏覽器型 Shell 或透過 AWS CLI 存取 Amazon EC2 執行個體。如需詳細資訊，請參閱 [AWS Systems Manager 工作階段管理員](#)。

主控台

若要使用工作階段管理員連線至資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要連線的 RDS Custom 資料庫執行個體。
3. 選擇 Configuration (組態)。
4. 請記下資料庫執行個體的 Resource ID (資源 ID)。例如，資源 ID 可能是 db-ABCDEFGHIJKLMNOPS0123456。
5. 在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在導覽窗格中，選擇執行個體。
7. 尋找 EC2 執行個體的名稱，然後按一下與其相關聯的執行個體 ID。例如，執行個體 ID 可能是 i-abcdefghijklm01234。
8. 選擇連線。
9. 選擇 Session Manager (工作階段管理員)。
10. 選擇連線。

您的工作階段會開啟一個視窗。

AWS CLI

您可以使用 AWS CLI 連線至 RDS Custom 資料庫執行個體。此技術需要 AWS CLI 的工作階段管理員外掛程式。若要了解如何安裝外掛程式，請參閱 [安裝 AWS CLI 的工作階段管理員外掛程式](#)。

若要尋找 RDS Custom 資料庫執行個體的資料庫資源 ID，請使用 `aws rds describe-db-instances`。

```
aws rds describe-db-instances \  
  --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \  
  --output text
```

下列範例輸出顯示 RDS Custom 執行個體的資源 ID。字首是 db-。

```
db-ABCDEFGHIJKLMNOPS0123456
```

若要尋找資料庫執行個體的 EC2 執行個體 ID，請使用 `aws ec2 describe-instances`。以下範例使用 db-ABCDEFGHIJKLMNOPS0123456 做為資源 ID。

```
aws ec2 describe-instances \  
  --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPS0123456" \  
  --output text \  
  --query 'Reservations[*].Instances[*].InstanceId'
```

以下範例輸出顯示 EC2 執行個體 ID。

```
i-abcdefghijklm01234
```

使用 `aws ssm start-session` 命令，在 `--target` 參數中提供 EC2 執行個體 ID。

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

成功的連線看起來如下。

```
Starting session with SessionId: yourid-abcdefghijklm1234  
[ssm-user@ip-123-45-67-89 bin]$
```

使用 SSH 連線到您的 RDS Custom 資料庫執行個體

Secure Shell 通訊協定 (SSH) 是一種支援透過不安全網路進行加密通訊的網路通訊協定。建立 RDS Custom 資料庫執行個體之後，您可以使用 SSH 用戶端連線到此執行個體。如需詳細資訊，請參閱 [使用 SSH 連線至您的 Linux 執行個體](#)。

您的 SSH 連線技術取決於您的資料庫執行個體是否為私有，這表示它不接受來自公用網際網路的連線。在這種情況下，您必須使用 SSH 通道將 ssh 公用程式連線到執行個體。這項技術會在現有的 SSH 工作階段內使用專用資料串流 (通道) 傳輸資料。您可以使用 AWS Systems Manager 來設定 SSH 通道。

Note

支援存取私有執行個體的各種策略。若要了解如何使用堡壘主機，將 SSH 用戶端連線到私有執行個體，請參閱 [AWS 上的 Linux 堡壘主機](#)。若要了解如何設定連接埠轉送，請參閱 [使用 AWS Systems Manager Session Manager 進行連接埠轉送](#)。

如果您的資料庫執行個體位於公有子網路且具有公開可用的設定，則不需要 SSH 通道。您可以像使用公有 Amazon EC2 執行個體一樣使用 SSH 進行連線。

若要將 SSH 用戶端連線到您的資料庫執行個體，請完成下列步驟：

1. [步驟 1：將您的資料庫執行個體設定為允許 SSH 連線](#)
2. [步驟 2：擷取您的 SSH 私密金鑰和 EC2 執行個體 ID](#)
3. [步驟 3：使用 ssh 公用程式連線到 EC2 執行個體](#)

步驟 1：將您的資料庫執行個體設定為允許 SSH 連線

若要確保您的資料庫執行個體能夠接受 SSH 連線，請執行下列動作：

- 請確定您的資料庫執行個體安全群組允許 TCP 連接埠 22 上的傳入連線。

若要了解如何設定資料庫執行個體的安全群組，請參閱 [使用安全群組控制存取](#)。

- 如果您不打算使用 SSH 通道，請確保您的資料庫執行個體位於公有子網路且可公開存取。

在主控台中，可在資料庫詳細資訊頁面的連線能力與安全性索引標籤上公開存取相關欄位。若要在 CLI 中檢查您的設定，請執行下列命令：

```
aws rds describe-db-instances \
  --query 'DBInstances[*].
  {DBInstanceIdentifier:DBInstanceIdentifier,PubliclyAccessible:PubliclyAccessible}' \
  --output table
```

若要變更資料庫執行個體的協助工具設定，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

步驟 2：擷取您的 SSH 私密金鑰和 EC2 執行個體 ID

若要使用 SSH 連線到資料庫執行個體，您需要與執行個體相關聯的 SSH 金鑰對。RDS 自訂會代表您建立安全殼層 key pair，並以前置詞命名 `do-not-delete-rds-custom-ssh-privatekey-db-`。AWS Secrets Manager 將您的 SSH 私鑰存儲為秘密。

使用 AWS Management Console 或擷取安全殼層密鑰 AWS CLI。如果您的執行個體具有公用 DNS，且您不打算使用 SSH 通道，亦請擷取 DNS 名稱。您可以指定 DNS 名稱進行公有連線。

主控台

擷取秘密 SSH 金鑰

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要連線的 RDS Custom 資料庫執行個體。
3. 選擇 Configuration (組態)。
4. 請記下 Resource ID (資源 ID) 值。例如，資料庫執行個體資源 ID 可能是 `db-ABCDEFGHIJKLMNOPS0123456`。
5. 在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在導覽窗格中，選擇執行個體。
7. 尋找 EC2 執行個體的名稱，然後選擇與其相關聯的執行個體 ID。例如，EC2 執行個體 ID 可能是 `i-abcdefghijklm01234`。
8. 在 Details (詳細資訊) 中尋找 Key pair name (金鑰對名稱)。配對名稱包含資料庫執行個體資源 ID。例如，配對名稱可能是 `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPS0123456-0d726c`。
9. 如果您的 EC2 執行個體是公有的，請記下公有 IPv4 DNS。例如，公有網域名稱系統 (DNS) 地址可能是 `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`。
10. 開啟主 AWS Secrets Manager 控制台，網址為 <https://console.aws.amazon.com/secretsmanager/>。
11. 選擇與金鑰對具有相同名稱的密碼。
12. 選擇 Retrieve secret value (擷取秘密值)。
13. 將 SSH 私有金鑰複製到文字檔，然後以 `.pem` 副檔名儲存檔案。例如，將檔案儲存為 `/tmp/do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPS0123456-0d726c.pem`。

AWS CLI

若要擷取 SSH 私有金鑰並將其儲存在 .pem 檔案中，您可以使用 AWS CLI。

1. 使用 `aws rds describe-db-instances` 尋找 RDS Custom 資料庫執行個體的資料庫資源 ID。

```
aws rds describe-db-instances \  
  --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \  
  --output text
```

下列範例輸出顯示 RDS Custom 執行個體的資源 ID。字首是 db-。

```
db-ABCDEFGHIJKLMNOPS0123456
```

2. 使用 `aws ec2 describe-instances` 尋找資料庫執行個體的 EC2 執行個體 ID。以下範例使用 db-ABCDEFGHIJKLMNOPS0123456 做為資源 ID。

```
aws ec2 describe-instances \  
  --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPS0123456" \  
  --output text \  
  --query 'Reservations[*].Instances[*].InstanceId'
```

以下範例輸出顯示 EC2 執行個體 ID。

```
i-abcdefghijklm01234
```

3. 若要尋找金鑰名稱，請指定 EC2 執行個體 ID。下列範例描述 EC2 執行個體 *i-0bdc4219e66944afa*。

```
aws ec2 describe-instances \  
  --instance-ids i-0bdc4219e66944afa \  
  --output text \  
  --query 'Reservations[*].Instances[*].KeyName'
```

下列範例輸出會顯示金鑰名稱，該名稱使用字首 do-not-delete-rds-custom-ssh-privatekey-。

```
do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPS0123456-0d726c
```

4. 使用 `aws secretsmanager` 將私有金鑰儲存在以金鑰命名的 `.pem` 檔案中。下列範例會將檔案儲存在 `/tmp` 目錄中。

```
aws secretsmanager get-secret-value \  
  --secret-id do-not-delete-rds-custom-ssh-privatekey-db-  
  ABCDEFGHIJKLMNOPQRS0123456-0d726c \  
  --query SecretString \  
  --output text >/tmp/do-not-delete-rds-custom-ssh-privatekey-db-  
  ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

步驟 3：使用 ssh 公用程式連線到 EC2 執行個體

您的連線技術取決於您要連線到私有資料庫執行個體還是連線到公有執行個體。私有連線需要您透過 AWS Systems Manager 設定 SSH 通道。

使用 ssh 公用程式連線到 EC2 執行個體

1. 若為私人連線，請修改 SSH 組態檔，以透過 Proxy 將命令傳送至 AWS Systems Manager Session Manager。若為公有連線，請跳至步驟 2。

將下列這幾行新增至 `~/.ssh/config`。針對名稱開頭為 `i-` 或 `mi-` 的主機，這些行會透過 Proxy 傳送 SSH 命令。

```
Host i-* mi-*  
  ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-  
  StartSSHSession --parameters 'portNumber=%p'"
```

2. 變更為包含 `.pem` 檔案的目錄。使用 `chmod`，將許可設定為 `400`。

```
cd /tmp  
chmod 400 do-not-delete-rds-custom-ssh-privatekey-db-  
  ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

3. 執行 `ssh` 公用程式，指定 `.pem` 檔案和公有 DNS 名稱 (進行公用連線) 或 EC2 執行個體 ID (進行私有連線)。以使用者 `ec2-user` 身分登入。

下列範例會使用 DNS 名稱 `ec2-12-345-678-901.us-east-2.compute.amazonaws.com` 連線至公有執行個體。

```
ssh -i \  
  
```

```
"do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPS0123456-0d726c.pem" \
ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com
```

下列範例會使用 EC2 執行個體 ID `i-0bdc4219e66944afa` 連線到私有執行個體。

```
ssh -i \
  "do-not-delete-rds-custom-ssh-privatekey-db-
  ABCDEFGHIJKLMNOPS0123456-0d726c.pem" \
  ec2-user@i-0bdc4219e66944afa
```

以 SYS 身分登入您的 RDS Custom for Oracle 資料庫

在建立 RDS Custom 資料庫執行個體之後，您可以使用者 SYS 身分登入 Oracle 資料庫，這將給予您 SYSDBA 權限。您有以下登入選項：

- 從 Secrets Manager 取得 SYS 密碼，並在 SQL 用戶端中指定此密碼。
- 使用作業系統身分驗證來登入您的資料庫。在此情況下，您不需要密碼。

為您的 RDS Custom for Oracle 資料庫尋找 SYS 密碼

您可以在 API 呼叫中指定主要使用者名稱，以 SYS 或 SYSTEM 身分登入 Oracle 資料庫。SYS 和 SYSTEM 的密碼存放在 Secrets Manager 中。##### `do-not-delete-rds-###-## ID-uuid#`您可以使用 AWS Management Console 尋找密碼。

主控台

在 Secrets Manager 中為您的資料庫尋找 SYS 密碼

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 RDS 主控台中，完成下列步驟：
 - a. 在導覽窗格中，選擇 Databases (資料庫)。
 - b. 選擇 RDS Custom for Oracle 資料庫執行個體的名稱。
 - c. 選擇 Configuration (組態)。
 - d. 複製資源 ID 下方的值。例如，您的資源 ID 可能是 db-ABC12CDE3FGH4I5JKLMNO6PQR7。

3. 前往以下位置開啟機密管理員控制台：<https://console.aws.amazon.com/secretsmanager/>。
4. 在 Secrets Manager 主控台中，完成下列步驟：
 - a. 在左側導覽窗格中，選擇秘密。
 - b. 依您在步驟 5 中複製的資源 ID 篩選秘密。
 - c. 選擇名為-do-not-delete-rds自訂-**ID-uuid #####**識別碼是您在步驟 5 中複製的####
#。例如，如果您的資源識別碼是 DB-ABC12CDE3FGH4I5JK，則您的密碼將會被命名為-自訂 DB-自訂 DB 12CDE3F GH4I5JK。do-not-delete-rds
 - d. 在秘密值中，選擇擷取秘密值。
 - e. 在金鑰/值中，複製密碼的值。
5. 在資料庫執行個體上安裝 SQL*Plus，然後以 SYS 身分登入資料庫。如需詳細資訊，請參閱 [步驟 3：將您的 SQL 連線至 Oracle 資料庫執行個體](#)。

使用作業系統身分驗證，登入 RDS Custom for Oracle 資料庫

作業系統使用者 rdsdb 擁有 Oracle 資料庫二進位檔案。您可以切換到 rdsdb 使用者並登入 RDS Custom for Oracle 資料庫，無需密碼。

1. 使用 Connect 到您的資料庫執行個體 AWS Systems Manager。如需詳細資訊，請參閱 [使用工作階段管理員連線至 RDS Custom 資料庫執行個體](#)。
2. 在 Web 瀏覽器中，前往 <https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html>。
3. 如需出現在網頁上的最新資料庫版本，請複製即時用戶端基本套件和 SQL*Plus 套件的 .rpm 連結 (不是 .zip 連結)。例如，下列是針對 Oracle Database 21.9 版的連結：
 - [https://download.oracle.com/otn_software/linux/instantclient/219000/ 每oracle-instantclient-basic](https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic)分鐘
 - [https://download.oracle.com/otn_software/linux/instantclient/219000/ 每oracle-instantclient-sqlplus](https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus)分鐘
4. 在您的 SSH 工作階段中，執行 wget 命令，從您在上一個步驟中取得的連結下載 .rpm 檔案。下列範例會下載 Oracle Database 21.9 版的 .rpm 檔案：

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

5. 執行 yum 命令安裝套件，如下所示：

```
sudo yum install oracle-instantclient-*.rpm
```

6. 切換到 rdsdb 使用者。

```
sudo su - rdsdb
```

7. 使用作業系統身分驗證來登入您的資料庫。

```
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 12 20:11:08 2023  
Version 21.9.0.0.0
```

```
Copyright (c) 1982, 2020, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.10.0.0.0
```

在 RDS Custom for Oracle 資料庫執行個體上安裝其他軟體元件

在新建立的資料庫執行個體中，您的資料庫環境包含 Oracle 二進位檔、資料庫和資料庫接聽程式。您可能想要在資料庫執行個體的主機作業系統上安裝其他軟體。例如，您可能想要安裝 Oracle Application Express (APEX)、Oracle Enterprise Manager (OEM) 代理程式或 Guardium S-TAP 代理程式。如需指導方針和高階指示，請參閱詳細的 AWS 部落格文章在 [Amazon RDS 自訂適用於 Oracle 的安裝其他軟體元件](#)。

管理 Amazon RDS Custom for Oracle 資料庫執行個體

Amazon RDS Custom 支援 Amazon RDS 資料庫執行個體一般管理任務的子集。接著，您可以使用 AWS Management Console 和 AWS CLI 尋找支援的 RDS Custom for Oracle 管理任務的指示。

主題

- [使用 RDS Custom for Oracle 中的容器資料庫 \(CDB\)](#)
- [使用 RDS Custom for Oracle 的高可用性](#)
- [自訂 RDS Custom 環境](#)
- [修改您的 RDS Custom for Oracle 資料庫執行個體](#)
- [變更 RDS Custom for Oracle 資料庫執行個體的字元集](#)
- [在 RDS Custom for Oracle 中設定 NLS_LANG 值](#)
- [支援透明資料加密](#)
- [標記 RDS Custom for Oracle 資源](#)
- [刪除 RDS Custom for Oracle 資料庫執行個體](#)

使用 RDS Custom for Oracle 中的容器資料庫 (CDB)

您可以使用 Oracle 多租戶架構 (custom-oracle-ee-cdb 或 custom-oracle-se2-cdb 引擎類型) 或傳統的非 CDB 架構 (或引擎類型) 來建立適用於 Oracle 資料庫執行個體的 RDS 自訂。custom-oracle-ee 或 custom-oracle-se2 當您建立一個容器資料庫 (CDB) 時，它包含一個插入式資料庫 (PDB) 和一個 PDB 種子。您可以使用 Oracle SQL 手動建立額外的 PDB。

PDB 和 CDB 名稱

建立 RDS Custom for Oracle CDB 執行個體時，您會指定初始 PDB 的名稱。依預設，您的初始 PDB 會命名為 ORCL。您可以選擇不同名稱。

根據預設，您的 CDB 會命名為 RDSCDB。您可以選擇不同名稱。CDB 名稱也是 Oracle 系統識別符 (SID) 的名稱，可唯一識別管理 CDB 的記憶體和程序。如需 Oracle SID 的詳細資訊，請參閱《Oracle 資料庫概念》中的 [Oracle 系統識別符 \(SID\)](#)。

您無法使用 Amazon RDS API 重新命名現有的 PDB。您也不能使用 modify-db-instance 命令重新命名 CDB。

PDB 管理

在 RDS Custom for Oracle 共同責任模型中，您負責管理 PDB 並建立任何額外的 PDB。RDS Custom 不會限制 PDB 的數目。您可以透過連線至 CDB 根並執行 SQL 陳述式，來手動建立、修改和刪除 PDB。在 Amazon EBS 資料磁碟區上建立 PDB，以防止資料庫執行個體超出支援周邊。

若要修改您的 CDB 或 PDB，請完成下列步驟：

1. 暫停自動化以防止干擾 RDS Custom 動作。
2. 修改您的 CDB 或 PDB。
3. 備份任何已修改的 PDB。
4. 繼續 RDS Custom 自動化。

自動復原 CDB 根

RDS Custom 會以其將非 CDB 保持開啟狀態的同一方式，將 CDB 根保持開啟狀態。如果 CDB 根的狀態變更，監控和復原自動化會嘗試將 CDB 根復原到所需的狀態。當根 CDB 關閉 (RDS-EVENT-0004) 或重新啟動 (RDS-EVENT-0006) 時，您會收到 RDS 事件通知，類似於非 CDB 架構。RDS Custom 會嘗試在資料庫執行個體啟動時以 READ WRITE 模式開啟所有 PDB。如果某些 PDB 無法開啟，RDS Custom 會發佈下列事件：tenant database shutdown。

使用 RDS Custom for Oracle 的高可用性

若要支援 Oracle 資料庫執行個體的 RDS 自訂之間的複寫，您可以使用 Oracle 資料保全設定高可用性 (HA)。主要資料庫執行個體會自動將資料同步至待命執行個體。此功能僅在企業版中受支援。

您可以使用下列方式設定高可用性環境：

- 在不同可用區域 (AZ) 中設定待命執行個體，以因應 AZ 故障。
- 將待命資料庫放置於掛載或唯讀模式。
- 從主要資料庫容錯移轉或切換至待命資料庫，而不會遺失資料。
- 透過設定內部部署執行個體的高可用性，然後容錯移轉或切換至 RDS Custom 待命資料庫來遷移資料。

若要了解如何設定高可用性，請參閱白皮書 [使用僅供讀取複本建置 Amazon RDS Custom for Oracle 的高可用性](#)。可執行以下任務：

- 使用虛擬私有網路 (VPN) 通道為您的高可用性執行個體加密傳輸中的資料。RDS Custom 不會自動設定傳輸中加密。
- 設定 Oracle 快速容錯移轉觀察程式 (FSFO) 來監控您的高可用性執行個體。
- 當滿足必要條件時，允許觀察程式執行自動容錯移轉。

自訂 RDS Custom 環境

RDS Custom for Oracle 包含內建功能，可讓您自訂資料庫執行個體環境，而無需暫停自動化。例如，您可以使用 RDS API 自訂環境，如下所示：

- 建立並還原資料庫快照以建立複製環境。
- 建立僅供讀取複本。
- 修改儲存體設定。
- 變更 CEV 以套用版本更新

對於某些自訂項目，例如變更字元集，您無法使用 RDS API。在這些情況下，您需要以根使用者身分存取 Amazon EC2 執行個體，或以 SYSDBA 身分登入 Oracle 資料庫，才能手動變更環境。

若要手動自訂執行個體，您必須暫停並繼續 RDS Custom 自動化。此暫停可確保您的自訂不會干擾 RDS Custom 自動化。如此一來，您就可以避免中斷支援周邊，這會將執行個體置於 `unsupported-configuration` 狀態，直到您修正了基本問題為止。當您修改 RDS Custom for Oracle 資料庫執行個體時，暫停和繼續是唯一支援的自動化任務。

自訂 RDS Custom 環境的一般步驟

若要自訂 RDS Custom 資料庫執行個體，請完成下列步驟：

1. 使用主控台或 CLI 在指定期間暫停 RDS Custom 自動化。
2. 識別基礎 Amazon EC2 執行個體。
3. 使用 SSH 金鑰或 AWS Systems Manager 連線至基礎 Amazon EC2 執行個體。
4. 在資料庫或作業系統層驗證您目前的組態設定。

您可以透過比較初始組態與變更的組態來驗證您的變更。根據自訂類型，使用作業系統工具或資料庫查詢。

5. 視需要自訂您的 RDS Custom for Oracle 資料庫執行個體。
6. 如有需要，重新啟動您的執行個體或資料庫。

Note

在內部部署 Oracle CDB 中，您可以使用內建命令或在啟動觸發條件之後，為 PDB 保留指定的開啟模式。當 CDB 重新啟動時，此機制會將 PDB 帶至指定的狀態。開啟您的 CDB 時，RDS Custom 自動化會捨棄任何使用者指定的保留狀態，並嘗試開啟所有 PDB。如果 RDS Custom 無法開啟所有 PDB，則會發出下列事件：The following PDBs failed to open: *List-of-PDBs*。

7. 驗證新的組態設定，方法是將這些設定與先前的設定進行比較。
8. 採取下列其中一種方式繼續 RDS Custom 自動化：
 - 手動繼續自動化。
 - 等待暫停期間結束。在此情況下，RDS Custom 會自動繼續監控和執行個體復原。
9. 驗證 RDS Custom 自動化架構

如果您已正確地遵循上述步驟，RDS Custom 會啟動自動備份。主控台中的執行個體狀態會顯示可用。

[如需最佳實務和 step-by-step 指示，請參閱部 AWS 部落格文章對 Oracle 執行個體適用的 Amazon RDS 自訂進行組態變更：第 1 部分](#)和[為 Oracle 資料庫重新建立 Amazon RDS 自訂：第 2 部分](#)。

暫停和繼續 RDS Custom 資料庫執行個體

您可以使用主控台或 CLI，暫停和繼續資料庫執行個體的自動化。

主控台

若要暫停或繼續 RDS Custom 自動化

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的 RDS Custom 資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 針對 RDS Custom automation mode (RDS Custom 自動化模式)，選擇下列其中一個選項：
 - Paused (暫停) 會暫停 RDS Custom 資料庫執行個體的監控和執行個體復原。輸入您想要的 Automation mode duration (自動化模式持續時間) 的暫停持續時間 (以分鐘為單位)。最小值為 60 分鐘 (預設值)。最大值為 1,440 分鐘。

- Full automation (全自動化) 會繼續自動化。
5. 選擇 Continue (繼續) 以檢查修改的摘要。
訊息隨即顯示，指出 RDS Custom 將會立即套用變更。
 6. 如果您的變更正確，請選擇 Modify DB instance (修改資料庫執行個體)。或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

在 RDS 主控台上，修改的詳細資訊即會出現。如果您暫停了自動化，您的 RDS Custom 資料庫執行個體的 Status (狀態) 會指出 Automation paused (自動化已暫停)。

7. (選用) 在導覽窗格中，選擇 Databases (資料庫)，然後再執行 RDS Custom 資料庫執行個體。

在 Summary (摘要) 窗格中，RDS Custom automation mode (RDS Custom 自動化模式) 會指出自動化狀態。如果暫停自動化，則值為 Paused (暫停)。自動化會在 *num* 分鐘內繼續。

AWS CLI

若要暫停或繼續 RDS 自訂自動化，請使用 `modify-db-instance` AWS CLI 命令。使用所需的參數 `--db-instance-identifier` 識別資料庫執行個體。使用下列參數控制自動化模式：

- `--automation-mode` 會指定資料庫執行個體的暫停狀態。有效值為 `all-paused` (會暫停自動化) 以及 `full` (會繼續自動化)。
- `--resume-full-automation-mode-minutes` 會指定暫停的持續時間。預設值為 60 分鐘。

Note

無論您指定 `--no-apply-immediately` 或 `--apply-immediately`，RDS Custom 會盡快以非同步的方式套用修改。

在命令回應中，`ResumeFullAutomationModeTime` 會以 UTC 時間戳記的形式表示繼續時間。當自動化模式為 `all-paused` 時，您可以使用 `modify-db-instance` 來繼續自動化模式或延長暫停時間。不支援其他 `modify-db-instance` 加密選項。

下列範例會暫停 `my-custom-instance` 的自動化 90 分鐘。

Example

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --automation-mode all-paused \  
  --resume-full-automation-mode-minutes 90
```

在 Windows 中：

```
aws rds modify-db-instance ^\  
  --db-instance-identifier my-custom-instance ^\  
  --automation-mode all-paused ^\  
  --resume-full-automation-mode-minutes 90
```

下列範例會將暫停持續時間延長額外 30 分鐘。30 分鐘會新增至 ResumeFullAutomationModeTime 中顯示的原始時間。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --automation-mode all-paused \  
  --resume-full-automation-mode-minutes 30
```

在 Windows 中：

```
aws rds modify-db-instance ^\  
  --db-instance-identifier my-custom-instance ^\  
  --automation-mode all-paused ^\  
  --resume-full-automation-mode-minutes 30
```

下列範例會繼續 *my-custom-instance* 的全自動化。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --automation-mode full \  
  --resume-full-automation-mode-minutes 30
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier my-custom-instance ^
  --automation-mode full
```

在下面的部分範例輸出中，待定 AutomationMode 值為 full。

```
{
  "DBInstance": {
    "PubliclyAccessible": true,
    "MasterUsername": "admin",
    "MonitoringInterval": 0,
    "LicenseModel": "bring-your-own-license",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "0123456789abcdefg"
      }
    ],
    "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
    "CopyTagsToSnapshot": false,
    "OptionGroupMemberships": [
      {
        "Status": "in-sync",
        "OptionGroupName": "default:custom-oracle-ee-19"
      }
    ],
    "PendingModifiedValues": {
      "AutomationMode": "full"
    },
    "Engine": "custom-oracle-ee",
    "MultiAZ": false,
    "DBSecurityGroups": [],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.custom-oracle-ee-19",
        "ParameterApplyStatus": "in-sync"
      }
    ],
    ...
    "ReadReplicaDBInstanceIdentifiers": [],
    "AllocatedStorage": 250,
```

```
"DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
"BackupRetentionPeriod": 3,
"DBName": "ORCL",
"PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
"Endpoint": {
  "HostedZoneId": "ABCDEFGHIJKLMNO",
  "Port": 8200,
  "Address": "my-custom-instance.abcdefghijkl.us-west-2.rds.amazonaws.com"
},
"DBInstanceStatus": "automation-paused",
"IAMDatabaseAuthenticationEnabled": false,
"AutomationMode": "all-paused",
"EngineVersion": "19.my_cev1",
"DeletionProtection": false,
"AvailabilityZone": "us-west-2a",
"DomainMemberships": [],
"StorageType": "gp2",
"DbiResourceId": "db-ABCDEFGHIJKLMNQRSTUUVW",
"ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
"KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
"StorageEncrypted": false,
"AssociatedRoles": [],
"DBInstanceClass": "db.m5.xlarge",
"DbInstancePort": 0,
"DBInstanceIdentifier": "my-custom-instance",
"TagList": []
}
```

修改您的 RDS Custom for Oracle 資料庫執行個體

修改適用於 Oracle 資料庫執行個體的 RDS 自訂類似於修改 Amazon RDS 資料庫執行個體。您可以變更如下所示的設定：

- DB instance class (資料庫執行個體類別)
- 存儲分配和類型
- Backup retention period (備份保留期間)
- 刪除保護
- Option group (選項群組)
- CEV (請參閱[升級 RDS Custom for Oracle 資料庫執行個體](#))
- 連線埠

主題

- [修改資料庫執行個體儲存體時的需求和限制](#)
- [修改資料庫執行個體類別時的需求和限制](#)
- [當您修改執行個體類別時，RDS Custom 如何建立您的資料庫執行個體](#)
- [修改您的 RDS Custom for Oracle 資料庫執行個體](#)

修改資料庫執行個體儲存體時的需求和限制

修改 RDS Custom for Oracle 資料庫執行個體的儲存體時，請考慮下列需求和限制：

- 針對 RDS Custom for Oracle 配置的儲存體下限為 40 GiB，上限為 64 TiB。
- 與 Amazon RDS 一樣，您無法減少配置的儲存體。這是 Amazon EBS 磁碟區的限制。
- 針對 RDS Custom 資料庫執行個體不支援儲存體自動擴展。
- 手動連接至 RDS Custom 資料庫執行個體的任何儲存磁碟區都在支援周邊外面。

如需詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

- RDS Custom 不支援磁帶 (標準) Amazon EBS 儲存體。您只能選擇 io1、gp2 或 gp3 SSD 儲存體類型。

如需 Amazon EBS 儲存體的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。如需與儲存體修改相關的一般資訊，請參閱 [使用適合 Amazon RDS 資料庫執行個體的儲存體](#)。

修改資料庫執行個體類別時的需求和限制

修改 RDS Custom for Oracle 資料庫執行個體的執行個體類別時，請考慮下列需求和限制：

- 您的資料庫執行個體必須處於 available 狀態。
- 您的資料庫執行個體必須在根磁碟區、資料磁碟區和二進位磁碟區上至少具有 100 MiB 的可用空間。
- 使用預設彈性網絡介面 (ENI) 時，您只能將單一彈性 IP (EIP) 指派給 RDS Custom for Oracle 資料庫執行個體。如果您將多個 ENI 連接到資料庫執行個體，修改操作會失敗。
- 所有 RDS Custom for Oracle 標籤都必須存在。
- 如果您使用 RDS Custom for Oracle 複寫，請注意下列需求和限制：
 - 對於主要資料庫執行個體和僅供讀取複本，您一次只能變更一個資料庫執行個體的執行個體類別。

- 如果您的 RDS Custom for Oracle 資料庫執行個體具有內部部署主要或複本資料庫，請務必在修改完成後手動更新內部部署資料庫執行個體上的私有 IP 地址。此動作對於保留 Oracle DataGuard 功能是必要的。RDS Custom for Oracle 會在修改成功時發佈事件。
- 當主要複本或僅供讀取複本資料庫執行個體已設定 FSFO (快速啟動容錯移轉) 時，您無法修改 RDS Custom for Oracle 資料庫執行個體類別。

當您修改執行個體類別時，RDS Custom 如何建立您的資料庫執行個體

當您修改執行個體類別時，RDS Custom 如何建立您的資料庫執行個體

- 建立 Amazon EC2 執行個體。
- 從最新的資料庫快照建立根磁碟區。RDS Custom for Oracle 不會保留在最新資料庫快照之後新增至根磁碟區的資訊。
- 創建 Amazon CloudWatch 警報。
- 如果您刪除了原始金鑰對，請建立 Amazon EC2 SSH 金鑰對。否則，RDS Custom for Oracle 會保留原始金鑰對。
- 使用您啟動修改時連接至資料庫執行個體的標籤建立新資源。當標籤直接連接至基礎資源時，RDS Custom 不會將這些標籤傳輸至新資源。
- 將最近修改的二進位磁碟區和資料磁碟區傳輸到新的資料庫執行個體。
- 傳輸彈性 IP 地址 (EIP)。如果資料庫執行個體可公開存取，則 RDS Custom 會在傳輸 EIP 之前，暫時將公用 IP 地址連接至新的資料庫執行個體。如果資料庫執行個體無法公開存取，RDS Custom 不會建立公用 IP 地址。

修改您的 RDS Custom for Oracle 資料庫執行個體

您可以使用主控台或 RDS API 修改資料庫執行個體類別或儲存體。AWS CLI

主控台

修改 RDS Custom for Oracle 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。

4. 選擇 Modify (修改)。
5. (選擇性) 在執行個體組態中，選擇資料庫執行個體類別的值。如需支援的類別，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#)。
6. (選擇性) 在儲存體中，視需要進行下列變更：
 - a. 針對 Allocated storage (配置儲存) 輸入新的值。必須大於目前的值，並且從 40 GiB-64 TiB。
 - b. 將儲存體類型的值變更為一般用途 SSD (gp2)、一般用途 SSD (gp3) 或佈建 IOPS (io1)。
 - c. 如果您使用佈建 IOPS (io1) 或一般用途 SSD (gp3)，則可以變更佈建 IOPS 值。
7. (選擇性) 在其他組態中，視需要進行下列變更：
 - 在選項群組中，選擇新的選項群組。如需詳細資訊，請參閱 [在適用於 Oracle 的 RDS 自訂中使用選項群組](#)。
8. 選擇 Continue (繼續)。
9. 選擇 Apply immediately (立即套用) 或 Apply during the next scheduled maintenance window (在下次排定的維護時段套用)。
10. 選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要修改適用於 Oracle 資料庫執行個體的 RDS 自訂儲存區，請使用 [modify-db-instance](#) AWS CLI 指令。視需要設定下列參數：

- `--db-instance-class` - 新的執行個體類別。如需支援的類別，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#)。
- `--allocated-storage` - 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。必須大於目前的值，並且從 40—65,536 GiB。
- `--storage-type` - 儲存體類型：gp2、gp3 或 io1。
- `--iops` - 資料庫執行個體的佈建 IOPS，若是使用 io1 或 gp3 儲存體類型的話。
- `--apply-immediately` - 使用 `--apply-immediately` 可立即套用儲存變更。

或使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。

下列範例會將的資料庫執行個體類別變更 my-cfo-instance 為 db.m5.16xlarge。此命令也會將儲存區大小變更為 1 TiB，儲存區類型為 io1，將已佈建 IOPS 變更為 3000，並將選項群組變更為 cfo-ee-19-mt。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-cfo-instance \  
  --db-instance-class db.m5.16xlarge \  
  --storage-type io1 \  
  --iops 3000 \  
  --allocated-storage 1024 \  
  --option-group cfo-ee-19-mt \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-cfo-instance ^  
  --db-instance-class db.m5.16xlarge ^  
  --storage-type io1 ^  
  --iops 3000 ^  
  --allocated-storage 1024 ^  
  --option-group cfo-ee-19-mt ^  
  --apply-immediately
```

變更 RDS Custom for Oracle 資料庫執行個體的字元集

RDS Custom for Oracle 預設為字元集 US7ASCII。您可以指定不同的字元集，以滿足語言或多位元組字元需求。當您使用 RDS Custom for Oracle 時，您可以暫停自動化，然後手動變更資料庫的字元集。

若要變更 RDS Custom to Oracle 資料庫執行個體的字元集，需符合下列需求：

- 新佈建的 RDS Custom 執行個體具有空白資料庫或未含應用程式資料的入門資料庫，您只能在此類執行個體上變更字元。對於所有其他情況，請使用 DMU (Database Migration Assistant for Unicode) 變更字元集。
- 您只能變更 RDS for Oracle 支援的字元集。如需詳細資訊，請參閱 [支援的資料庫字元集](#)。

變更 RDS Custom for Oracle 資料庫執行個體的字元集

1. 暫停 RDS Custom 自動化。如需詳細資訊，請參閱 [暫停和繼續 RDS Custom 資料庫執行個體](#)。
2. 以具有 SYSDBA 權限的使用者身分登入您的資料庫。

3. 以受限制模式重新啟動資料庫、變更字元集，然後以正常模式重新啟動資料庫。

在您的 SQL 用戶端中執行下列指令碼：

```
SHUTDOWN IMMEDIATE;  
STARTUP RESTRICT;  
ALTER DATABASE CHARACTER SET INTERNAL_CONVERT AL32UTF8;  
SHUTDOWN IMMEDIATE;  
STARTUP;  
SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
```

確認輸出顯示正確的字元集。

```
VALUE  
-----  
AL32UTF8
```

4. 繼續 RDS Custom 自動化。如需詳細資訊，請參閱 [暫停和繼續 RDS Custom 資料庫執行個體](#)。

在 RDS Custom for Oracle 中設定 NLS_LANG 值

地區設定是處理語言和文化需求的一組資訊，這些資訊對應至特定語言和國家。若要指定 Oracle 軟體的地區設定行為，請在您的用戶端主機上設定 NLS_LANG 環境變數。此變數會設定用戶端應用程式在資料庫工作階段中所使用的語言、地區和字元集。

對於 RDS Custom for Oracle，您只能在 NLS_LANG 變數中設定語言：區域和字元使用預設值。該語言用於 Oracle 資料庫訊息、定序，日期名稱和月份名稱。每個支援的語言都有唯一名稱，例如美國英文、法文或德文。如果未指定語言，值預設為美國英文。

在建立 RDS Custom for Oracle 資料庫之後，您可以在用戶端主機上將 NLS_LANG 設定為英文以外的語言。若要查看 Oracle Database 支援的語言清單，請登入您的 RDS Custom for Oracle 資料庫並執行下列查詢：

```
SELECT VALUE FROM V$NLS_VALID_VALUES WHERE PARAMETER='LANGUAGE' ORDER BY VALUE;
```

您可以在主機命令列上設定 NLS_LANG。下列範例會在 Linux 上使用 Z Shell，將用戶端應用程式的語言設定為德文。

```
export NLS_LANG=German
```

您的應用程式會在啟動時讀取 NLS_LANG 值，然後在連線時將其傳達至資料庫。

如需詳細資訊，請參閱《Oracle Database 全球化支援指南》中的[使用 NLS_LANG 環境變數選擇地區設定](#)。

支援透明資料加密

RDS Custom 支援 RDS Custom for Oracle 資料庫執行個體的透明資料加密 (TDE)。

不過，您無法像在 RDS for Oracle 中一樣，使用自訂選項群組中的選項啟用 TDE。您可以手動開啟 TDE。如需使用 Oracle 透明資料加密的詳細資訊，請參閱 Oracle 文件中的[使用透明資料加密來保護儲存的資料](#)。

標記 RDS Custom for Oracle 資源

您可以將 RDS Custom 資源標記為與 Amazon RDS 資源一樣，但是有一些重要差異：

- 請勿建立或修改 RDS Custom 自動化所需的 AWSRDSCustom 標籤。如果您這樣做，可能會破壞自動化。
- Name 標籤會新增至字首值為 do-not-delete-rds-custom 的 RDS Custom 資源。金鑰的任何客戶傳遞值都會遭到覆寫。
- 在建立期間新增至 RDS Custom 資料庫執行個體的標籤會傳播至所有其他相關的 RDS Custom 資源。
- 當您在建立資料庫執行個體之後將標籤新增至 RDS Custom 資源時，不會傳播標籤。

如需資源標記的一般資訊，請參閱[標記 Amazon RDS 資源](#)。

刪除 RDS Custom for Oracle 資料庫執行個體

若要刪除 RDS Custom 資料庫執行個體，請執行以下操作：

- 提供資料庫執行個體的名稱。
- 清除讓您建立資料庫執行個體最終資料庫快照的選項。
- 選擇或清除保留自動化備份的選項。

您可以使用主控台或 CLI 刪除 RDS Custom 資料庫執行個體。刪除資料庫執行個體所需的時間會因備份保留期 (也就是要刪除的備份數量) 和要刪除的資料數量而有所不同。

主控台

若要刪除 RDS Custom 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要刪除的 RDS Custom 資料庫執行個體。RDS Custom 資料庫執行個體會顯示角色 Instance (RDS Custom) (執行個體 (RDS Custom))。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。
4. 若要保留自動備份，請選擇 Retain automated backups (保留自動備份)。
5. 在方塊中輸入 **delete me**。
6. 選擇 Delete (刪除)。

AWS CLI

您可以使用 [delete-db-instance](#) AWS CLI 命令刪除 RDS 自訂資料庫執行個體。使用所需的參數 `--db-instance-identifier` 識別資料庫執行個體。其餘參數與 Amazon RDS 資料庫執行個體的參數相同，但是有下列例外：

- `--skip-final-snapshot` 是必要的。
- `--no-skip-final-snapshot` 不支援。
- `--final-db-snapshot-identifier` 不支援。

下列範例會刪除名為 `my-custom-instance` 的 RDS Custom 資料庫執行個體，並保留自動化備份。

Example

對於 Linux/macOS、或 Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --skip-final-snapshot \  
  --no-delete-automated-backups
```

在 Windows 中：

```
aws rds delete-db-instance ^
```

```
--db-instance-identifier my-custom-instance ^  
--skip-final-snapshot ^  
--no-delete-automated-backups
```


使用 RDS Custom for Oracle 的 Oracle 複本

您可以為執行 Oracle 企業版的 Oracle 資料庫執行個體建立 RDS 自訂 Oracle 複本。同時支援容器資料庫 (CDB) 和非 CDB。標準版 2 不支援 Oracle 資料保全。

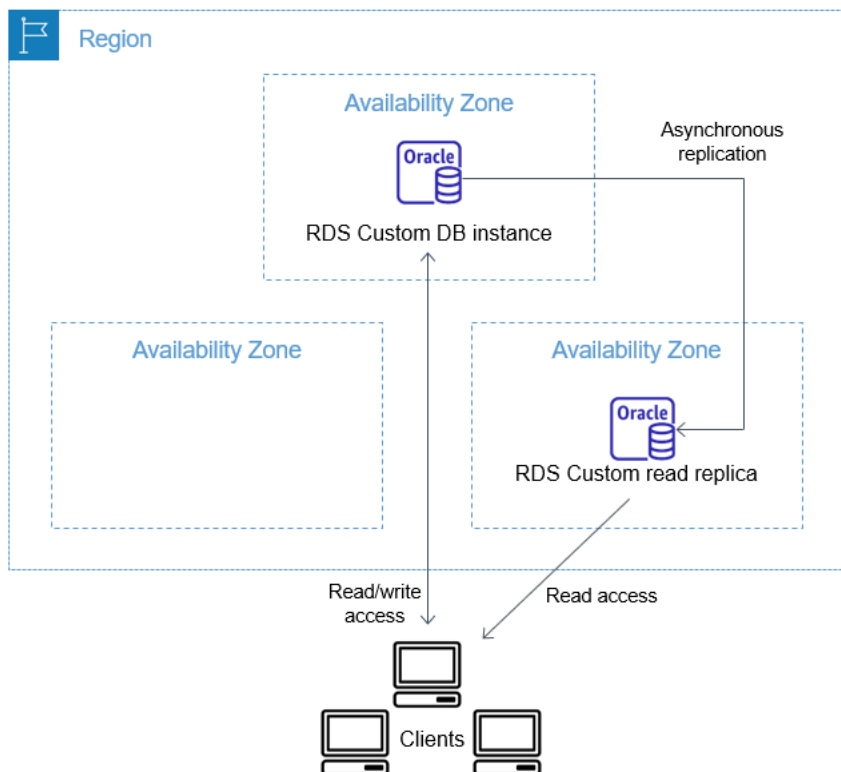
建立 RDS Custom for Oracle 複本與建立 RDS for Oracle 複本過程類似，但有一些重要差異。如需建立和管理 Oracle 複本的一般資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#) 和 [使用適用於 Amazon RDS for Oracle 的僅供讀取複本](#)。

主題

- [RDS Custom for Oracle 複本的概觀](#)
- [RDS Custom for Oracle 複本的指南和限制](#)
- [將 RDS Custom for Oracle 複本升級為獨立的資料庫執行個體](#)

RDS Custom for Oracle 複本的概觀

RDS Custom for Oracle 複本的架構是 RDS for Oracle 複本的類比。主要資料庫執行個體會以非同步方式複寫到一或多個 Oracle 複本。



複本數上限

與 RDS for Oracle 相同，您的 RDS Custom for Oracle 主要資料庫執行個體最多可以建立 5 個受管的 Oracle 複本。您也可以自行建立手動設定 (外部) 的 Oracle 複本。外部複本不會計入您的資料庫執行個體限制。外部複本也屬於 RDS Custom 支援範圍之外。如需支援周邊的詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

複本命名慣例

Oracle 複本名稱是以資料庫唯一名稱為基礎。格式為 `DB_UNIQUE_NAME_X`，按附加字母順序排序。例如，如果您的資料庫唯一名稱為 ORCL，則前兩個複本會命名為 ORCL_A 和 ORCL_B。前六個字母 A—F 會保留給 RDS Custom。RDS Custom 會從主要資料庫執行個體將資料庫參數複製到複本。如需詳細資訊，請參閱 Oracle 文件中的 [DB_UNIQUE_NAME](#)。

複本備份保留

根據預設，RDS Custom Oracle 複本會使用與主要資料庫執行個體相同的備份保留期間。您可以將備份保留期修改為 1 - 35 天。RDS 自訂支援備份、還原和 point-in-time 復原 (PITR)。如需備份和還原 RDS Custom 資料庫執行個體的詳細資訊，請參閱 [備份與還原 Amazon RDS Custom for Oracle DB 資料庫執行個體](#)。

Note

建立 Oracle 複本時，RDS Custom 會暫時暫停重做日誌的清理。如此一來，RDS Custom 可確保新的 Oracle 複本可供使用之後，將這些日誌套用至新的 Oracle 複本。

複本提升

您可以使用主控台、`promote-read-replica` AWS CLI 命令或 `PromoteReadReplica` API，在 RDS 自訂適用於 Oracle 中升級受管理的 Oracle 複本。如果您刪除主要資料庫執行個體，且所有複本都正常運作，則 RDS Custom for Oracle 會自動將受管複本提升為獨立執行個體。如果複本已暫停自動提升或位於支援範圍以外，您必須先修正複本，RDS Custom 才可以自動提升複本。您只能手動提升外部 Oracle 複本。

RDS Custom for Oracle 複本的指南和限制

建立 RDS Custom for Oracle 複本時，並非所有 RDS Oracle 複本選項都支援。

主題

- [RDS Custom for Oracle 複本的一般指南](#)
- [RDS Custom for Oracle 複本的一般限制](#)
- [RDS Custom for Oracle 複本的網路需求和限制](#)
- [RDS Custom for Oracle 的外部複本限制](#)
- [RDS Custom for Oracle 的複本提升限制](#)
- [RDS Custom for Oracle 的複本提升指南](#)

RDS Custom for Oracle 複本的一般指南

使用 RDS Custom for Oracle 的時候，請遵循下列指示：

- 您只能在 Oracle 企業版中使用 RDS 自訂進行 Oracle 複寫。不支援標準版 2。
- 請不要修改 RDS_DATAGUARD 使用者。此使用者是為 RDS Custom for Oracle 自動化保留的。修改此使用者可能會導致意外結果，例如，無法為您的 RDS Custom for Oracle 資料庫執行個體建立 Oracle 複本。
- 請勿變更複本使用者密碼。管理 RDS Custom 主機上的 Oracle Data Guard 組態時會需要該密碼。如果您變更密碼，RDS Custom for Oracle 可能會將 Oracle 複本置於支援範圍以外。如需詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

密碼儲存在中 AWS Secrets Manager，並以資料庫資源 ID 標記。每個 Oracle 複本在 Secrets Manager 中都有自己的秘密。秘密的格式如下。

```
do-not-delete-rds-custom-db-DB_resource_id-6-digit_UUID-dg
```

- 請勿變更主要資料庫執行個體的 DB_UNIQUE_NAME。變更名稱會導致任何還原操作停滯。
- 請勿在 RDS Custom CDB 的 CREATE PLUGGABLE DATABASE 命令中指定子句 STANDBYS=NONE。這樣一來，如果發生容錯移轉，待命 CDB 會包含所有 PDB。

RDS Custom for Oracle 複本的一般限制

RDS Custom for Oracle 複本具有下列限制：

- 您僅能在唯讀模式下建立 RDS Custom for Oracle 複本。不過，您可以手動將掛載複本的模式變更為唯讀，再從唯讀變更為掛載。如需詳細資訊，請參閱 [create-db-instance-read-replica](#) AWS CLI 命令的文件。
- 您無法建立跨區域 RDS Custom for Oracle 複本。

- 您無法變更 Oracle Data Guard CommunicationTimeout 值的參數。此參數針對 Oracle 資料庫執行個體設定為 15 秒。

RDS Custom for Oracle 複本的網路需求和限制

確定您的網路組態支援 RDS Custom for Oracle 複本。考慮下列各項：

- 請務必針對主要資料庫執行個體和所有複本，對於虛擬私有雲端 (VPC) 內的輸入和輸出通訊啟用連接埠 1140。這對於僅供讀取複本之間的 Oracle Data Guard 通訊而言是必要的。
- RDS Custom for Oracle 會於建立 Oracle 複本時驗證網路。如果主要資料庫執行個體和新的複本無法透過網路連線，則 RDS Custom for Oracle 不會建立複本，並將其置於 INCOMPATIBLE_NETWORK 狀態。
- 對於外部 Oracle 複本 (例如您在 Amazon EC2 或內部部署建立的複本)，請針對 Oracle Data Guard 複寫使用另一個連接埠和接聽程式。嘗試使用連接埠 1140 可能會導致與 RDS Custom 自動化發生衝突。
- /rdsdbdata/config/tnsnames.ora 檔案包含對應至接聽程式通訊協定位址的網路服務名稱。請注意以下要求和建議：
 - 在處理 Oracle 複本操作時，tnsnames.ora 中字首為 rds_custom_ 的項目是為 RDS Custom 保留的。

在 tnsnames.ora 中建立手動項目時，不使用此字首。

- 在某些情況下，您可能想要手動切換或容錯移轉，或使用容錯移轉技術，例如快速啟動容錯移轉 (FSFO)。若是如此，請確定手動將 tnsnames.ora 項目從主要資料庫執行個體同步到所有待命執行個體。此建議適用於由 RDS Custom 管理的 Oracle 複本和外部 Oracle 複本。

RDS Custom 自動化只會在主要資料庫執行個體上更新 tnsnames.ora 項目。同時務必在新增或移除 Oracle 複本時同步。

如果您未同步 tnsnames.ora 檔案並手動切換或容錯移轉，則主要資料庫執行個體上的 Oracle Data Guard 可能無法與 Oracle 複本通訊。

RDS Custom for Oracle 的外部複本限制

RDS Custom for Oracle 外部複本 (包含內部部署複本) 具有下列限制：

- RDS Custom for Oracle 不會對外部 Oracle 複本在手動容錯移轉 (例如 FSFO) 時偵測執行個體角色變更。

RDS Custom for Oracle 不會偵測受管複本的變更。角色變更會記錄在事件日誌中。您也可以使用 [describe-db-instances](#) AWS CLI 指令查看新狀態。

- RDS Custom for Oracle 不會偵測外部 Oracle 複本的高複寫延遲。

RDS Custom for Oracle 不會偵測受管複本的延遲。高複寫延遲會產生 Replication has stopped 事件。您也可以使用 [describe-db-instances](#) AWS CLI 命令查看複製狀態，但可能會有延遲更新。

- 若您刪除主要資料庫執行個體，RDS Custom for Oracle 不會自動提升外部 Oracle 複本。

自動提升功能僅適用於受管的 Oracle 複本。如需手動提升僅 Oracle 複本的相關資訊，請參閱白皮書 [使用 Amazon RDS Custom for Oracle 上的 Data Guard 來啟用高可用性](#)。

RDS Custom for Oracle 的複本提升限制

RDS Custom for Oracle 受管 Oracle 複本的升級與 RDS 受管複本的升級相同，但有一些差異。請留意 RDS Custom for Oracle 複本的下列限制：

- RDS Custom for Oracle 正在進行備份時，您無法提升複本。
- 提升 Oracle 複本時，您無法將備份保留期變更為 0。
- 若複本的運作狀態不佳，您便不能提升該複本。

如果您在主要資料庫執行個體上發出問題 `delete-db-instance`，RDS Custom for Oracle 會驗證每個受管的 Oracle 複本是否正常運作且可用於升級。複本可能會由於自動化已暫停或位於支援範圍之外而不符合升級。在這種情況下，RDS Custom for Oracle 會發佈事件說明問題，以便您可以手動修復 Oracle 複本。

RDS Custom for Oracle 的複本提升指南

提升複本時，請注意下列指示：

- RDS Custom for Oracle 升級複本時，請勿發起容錯移轉。否則，可能會導致提升工作流程停滯。
- 當 Oracle Custom for Oracle 正在升級您的 Oracle 複本時，請勿切換主要資料庫執行個體。否則，可能會導致提升工作流程停滯。
- 當 RDS Custom for Oracle 正在升級您的 Oracle 複本時，請勿關閉主要資料庫執行個體。否則，可能會導致提升工作流程停滯。

- 請勿嘗試以新提升的資料庫執行個體做為目標重新啟動複寫。RDS Custom for Oracle 提升您的 Oracle 複本之後，它會成為獨立的資料庫執行個體，而且不再具有複本角色。

如需詳細資訊，請參閱 [RDS Custom for Oracle 複本提升的疑難排解](#)。

將 RDS Custom for Oracle 複本升級為獨立的資料庫執行個體

就像使用 RDS for Oracle 一樣，您可以將 RDS Custom for Oracle 提升為獨立的資料庫執行個體。當您提升 Oracle 複本時，RDS Custom for Oracle 資料庫執行個體將在可使用前重新啟動。如需提升 Oracle 複本的相關詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

以下步驟顯示了提升 Oracle 複本至資料庫執行個體的一般流程：

1. 停止任何交易寫入主要資料庫執行個體。
2. 等待 RDS Custom for Oracle 套用所有更新至您的 Oracle 複本。
3. 選擇 Amazon RDS 主控台、AWS CLI 命令或 [PromoteReadReplica](#) Amazon RDS API 作業上的升級選項 [promote-read-replica](#)，以提升您的 Oracle 複本。

提升 Oracle 複本可能需要幾分鐘。在此程序期間，RDS Custom for Oracle 會停止複寫並重新啟動您的複本。當重新啟動完成，該 Oracle 複本便可用做獨立的資料庫執行個體。

主控台

若要將 RDS Custom for Oracle 複本提升為獨立的資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。

Databases (資料庫) 窗格隨即出現。每個 Oracle 複本會在 Role (角色) 欄中顯示 Replica (複本)。

3. 選擇您想提升的 RDS Custom for Oracle 複本。
4. 針對 Actions (動作)，選擇 Promote (提升)。
5. 在 Promote Oracle replica (提升 Oracle 複本) 頁面上，輸入新升級資料庫執行個體的備份保留期間和備份時間。您無法將此數值設為 0。
6. 依您所需完成設定後，選擇 Promote Oracle replica (提升 Oracle 複本)。

AWS CLI

若要將適用於 Oracle 複本的 RDS 自訂升級為獨立資料庫執行個體，請使用 AWS CLI [promote-read-replica](#) 指令。

Example

對於LinuxmacOS、或Unix：

```
aws rds promote-read-replica \  
--db-instance-identifier my-custom-read-replica \  
--backup-retention-period 2 \  
--preferred-backup-window 23:00-24:00
```

在 Windows 中：

```
aws rds promote-read-replica ^  
--db-instance-identifier my-custom-read-replica ^  
--backup-retention-period 2 ^  
--preferred-backup-window 23:00-24:00
```

RDS API

若要將 RDS Custom for Oracle 複本提升為獨立的資料庫執行個體，請使用所需參數 `PromoteReadReplica` 呼叫 Amazon RDS API [DBInstanceIdentifier](#) 操作。

備份與還原 Amazon RDS Custom for Oracle DB 資料庫執行個體

就像是 Amazon RDS 一樣，RDS Custom 會在資料庫執行個體的備份時段建立並儲存 RDS Custom for Oracle 資料庫執行個體的自動備份。您也可以手動備份您的資料庫執行個體。

程序與擷取 Amazon RDS 資料庫執行個體的快照相同。RDS Custom 資料庫執行個體的第一個快照會包含完整的資料庫執行個體資料。後續快照是增量。

使用 AWS Management Console 或還原資料庫快照 AWS CLI。

主題

- [建立 RDS Custom for Oracle 快照](#)
- [從 RDS Custom for Oracle 資料庫快照還原](#)
- [將 RDS Custom for Oracle 執行個體還原至某個時間點](#)
- [刪除 RDS Custom for Oracle 快照](#)
- [刪除 RDS Custom for Oracle 自動備份](#)

建立 RDS Custom for Oracle 快照

RDS Custom for Oracle 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。當您的資料庫執行個體包含容器資料庫 (CDB) 時，執行個體的快照會包含根 CDB 和所有 PDB。

建立 RDS Custom for Oracle 快照時，請指定要備份哪個 RDS Custom 資料庫執行個體。為快照命名，稍後即可從該快照進行還原。

建立快照時，RDS Custom for Oracle 會為連接到資料庫執行個體的每個磁碟區建立 Amazon EBS 快照。RDS Custom for Oracle 會使用根磁碟區的 EBS 快照來註冊新的 Amazon Machine Image (AMI)。為了讓快照更容易與特定資料庫執行個體產生關聯，它們透過 DBSnapshotIdentifier、DbiResourceId 和 VolumeType 加上標籤。

建立資料庫快照會導致短暫的輸入/輸出暫停。根據資料庫執行個體的大小和類別，暫停時間會持續幾秒鐘至幾分鐘。快照建立時間會隨著資料庫的大小而有所不同。由於快照包括整個儲存體磁碟區，所以檔案大小 (例如暫存檔案) 也會影響建立快照建立時間。若要進一步了解建立快照，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

使用主控台或 AWS CLI 建立 RDS Custom for Oracle 快照。

主控台

若要建立 RDS Custom 快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 在 RDS Custom 資料庫執行個體清單中，選擇您要建立快照的執行個體。
4. 針對 Actions (動作)，選擇 Take snapshot (建立快照)。

Take DB Snapshot (建立資料庫快照) 視窗隨即顯示。

5. 在 Snapshot name (快照名稱) 中輸入快照的名稱。
6. 選擇 Take Snapshot (擷取快照)。

AWS CLI

您可以使用 [create-db-snapshot](#) AWS CLI 命令建立 RDS 自訂資料庫執行個體的快照。

指定下列選項：

- `--db-instance-identifier` – 識別您要備份的 RDS Custom 資料庫執行個體
- `--db-snapshot-identifier` – 為您的 RDS Custom 快照命名以便稍後從該快照還原

在此範例中，您會為名為 *my-custom-instance* 的 RDS Custom 資料庫執行個體建立名為 *my-custom-snapshot* 的資料庫快照。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-snapshot \  
  --db-instance-identifier my-custom-instance \  
  --db-snapshot-identifier my-custom-snapshot
```

在 Windows 中：

```
aws rds create-db-snapshot ^  
  --db-instance-identifier my-custom-instance ^
```

```
--db-snapshot-identifier my-custom-snapshot
```

從 RDS Custom for Oracle 資料庫快照還原

還原 RDS Custom for Oracle 資料庫執行個體時，您要提供資料庫快照的名稱和新執行個體的名稱。您無法從快照還原至現有的 RDS Custom 資料庫執行個體。還原時，會建立新的 RDS Custom for Oracle 資料庫執行個體。

還原程序與在 Amazon RDS 中還原的方式有以下不同：

- 還原快照之前，RDS Custom for Oracle 會先備份現有的組態檔案。這些檔案可以在目錄 / rdsdbdata/config/backup 中的還原執行個體上取得。RDS Custom for Oracle 會使用預設參數還原資料庫快照，並使用現有的資料庫組態檔案覆寫先前的資料庫組態檔案。因此，還原的執行個體不會保留自訂參數和對資料庫組態檔案的變更。
- 還原的資料庫名稱與快照中的名稱相同。您無法指定不同的名稱。(對於 RDS Custom for Oracle，預設值為 ORCL。)

主控台

若要從資料庫快照還原 RDS Custom 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。
5. 在 Restore DB Instance (還原資料庫執行個體) 頁面上，針對 DB Instance Identifier (資料庫執行個體識別符)，輸入您還原的 RDS Custom 資料庫執行個體的名稱。
6. 選擇 Restore DB instance (還原資料庫執行個體)。

AWS CLI

您可以使用 [restore-db-instance-from-db-](#) AWS CLI 快照命令還原 RDS 自訂資料庫快照集。

如果您要從中還原的快照適用於私有資料庫執行個體，請務必指定正確的 db-subnet-group-name 和 no-publicly-accessible。否則，資料庫執行個體預設為可公開存取。下列選項是必要的：

- db-snapshot-identifier – 識別要從中進行還原的快照

- `db-instance-identifier` – 指定要從資料庫快照建立之 RDS Custom 資料庫執行個體的名稱
- `custom-iam-instance-profile` – 指定與 RDS Custom 資料庫執行個體的基礎 Amazon EC2 執行個體關聯的執行個體設定檔。

下列程式碼會為 `my-custom-instance` 還原名為 `my-custom-snapshot` 的快照。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-snapshot-identifier my-custom-snapshot \  
  --db-instance-identifier my-custom-instance \  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \  
  --no-publicly-accessible
```

在 Windows 中：

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-snapshot-identifier my-custom-snapshot ^  
  --db-instance-identifier my-custom-instance ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^  
  --no-publicly-accessible
```

將 RDS Custom for Oracle 執行個體還原至某個時間點

您可以將資料庫執行個體還原至特定時間點 (PITR)，建立新的資料庫執行個體。若要支援 PITR，您的資料庫執行個體必須將備份保留設定為非零值。

RDS Custom for Oracle 資料庫執行個體的最晚可還原時間取決於許多因素，但通常是目前時間的 5 分鐘內。若要查看資料庫執行個體的最新可還原時間，請使用 AWS CLI [describe-db-instances](#) 指令並查看資料庫執行個體在 `LatestRestorableTime` 欄位中傳回的值。若要在 Amazon RDS 主控台中查看每個資料庫執行個體的最新可還原時間，請選擇自動備份。

您可以還原至備份保留期間內的任何時間點。若要查看每個資料庫執行個體的最早可還原時間，請在 Amazon RDS 主控台中選擇自動備份。

如需 PITR 的一般資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

主題

- [RDS Custom for Oracle 的 PITR 考量](#)

RDS Custom for Oracle 的 PITR 考量

在 RDS Custom for Oracle 中，PITR 與 Amazon RDS 中的 PITR 有以下的重要不同：

- 還原的資料庫名稱與來源資料庫執行個體中的名稱相同。您無法指定不同的名稱。預設值為 ORCL。
- `AWSRDSCustomIamRolePolicy` 需要新許可。如需詳細資訊，請參閱 [步驟 2：將存取原則新增至 `AWSRDSCustomInstanceRoleForRdsCustomInstance`](#)。
- 所有 RDS Custom for Oracle 資料庫執行個體都必須將備份保留設定為非零值。
- 如果您變更作業系統或資料庫執行個體時區，PITR 可能無法運作。如需變更時區的相關資訊，請參閱 [Oracle 時區](#)。
- 如果將自動化設定為 ALL_PAUSED，RDS Custom 會暫停上傳已存檔重做日誌檔，包括在最近可還原時間 (LRT) 之前建立的記錄檔。我們建議您暫停自動化一段時間。

舉例來說，假設您的 LRT 是 10 分鐘前。您暫停自動化。暫停期間，RDS Custom 不會上傳封存的重新日誌。如果您的資料庫執行個體當機，您只能復原到暫停時存在的 LRT 之前的時間。當您繼續自動化時，RDS Custom 會繼續上傳日誌。LRT 會繼續。套用一般 PITR 規則。

- 在 RDS Custom 中，您可以手動指定保留封存的重新日誌的任意時數，以便在 RDS Custom 上傳後刪除之前保留這些封存的重新日誌。請如下指定小時數：

1. 建立名為 `/opt/aws/rdscustomagent/config/redo_logs_custom_configuration.json` 的文字檔案。
2. 以下列格式新增 JSON 物件：`{"archivedLogRetentionHours" : "num_of_hours"}`。
數字必須是介於 1—840 範圍內的整數。

- 假設您將非 CDB 插入容器資料庫 (CDB) 作為 PDB，然後嘗試 PITR。只在您先前已備份 PDB 時，操作才會成功。建立或修改 PDB 之後，建議您一律將其備份。
- 我們建議您不要自訂資料庫初始化參數。例如，修改下列參數會影響 PITR：
 - `CONTROL_FILE_RECORD_KEEP_TIME` 會影響上傳和刪除日誌的規則。
 - `LOG_ARCHIVE_DEST_n` 不支援多個目的地。
 - `ARCHIVE_LAG_TARGET` 會影響最近的可還原時間。`ARCHIVE_LAG_TARGET` 設定為 300 因為復原點目標 (RPO) 為 5 分鐘。為了實現此目標，RDS 每 5 分鐘切換一次線上重新日誌，並將其存放在 Amazon S3 儲存貯體中。如果記錄切換器的頻率造成 RDS 自訂 for Oracle 資料庫的效能問題，您可以將資料庫執行個體和儲存擴展到 IOPS 和輸送量較高的執行個體和儲存體。如果您的復原計劃有必要，您可以將 `ARCHIVE_LAG_TARGET` 初始化參數的設定調整為 60—7200 之間的值。
- 如果您自訂資料庫初始化參數，強烈建議您只自訂下列項目：
 - `COMPATIBLE`
 - `MAX_STRING_SIZE`

- DB_FILES
- UNDO_TABLESPACE
- ENABLE_PLUGGABLE_DATABASE
- CONTROL_FILES
- AUDIT_TRAIL
- AUDIT_TRAIL_DEST

針對其他所有初始化參數，RDS Custom 會還原預設值。如果您修改不在上述清單中的參數，它可能會對 PITR 產生不良影響，並導致無法預期的結果。例如，CONTROL_FILE_RECORD_KEEP_TIME 會影響上傳和刪除日誌的規則。

您可以使用 AWS Management Console、或 RDS API 將 RDS 自訂資料庫執行個體還原到 AWS CLI 某個時間點。

主控台

若要將 RDS Custom 資料庫執行個體還原至指定時間

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 選擇您想要還原的 RDS Custom 資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。

Restore to point in time (還原到時間點) 視窗隨即出現。

5. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇 Custom (自訂)，請輸入您希望執行個體叢集還原到什麼日期和時間。

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

6. 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入目標還原 RDS Custom 資料庫執行個體的名稱。名稱必須是唯一的。
7. 視需要選擇其他選項，例如資料庫執行個體類別。
8. 選擇 Restore to point in time (還原至時間點)。

AWS CLI

您可以使用 [restore-db-instance-to-point-in-time](#) AWS CLI 命令建立新的 RDS 自訂資料庫執行個體，將資料庫執行個體還原到指定的時間。

使用下列其中一個選項來指定要從中還原的備份：

- `--source-db-instance-identifier` *mysourcedbinstance*
- `--source-dbi-resource-id` *dbinstanceresourceID*
- `--source-db-instance-automated-backups-arn` *backupARN*

`custom-iam-instance-profile` 選項為必要項目。

下列範例會將 `my-custom-db-instance` 還原至名稱為 `my-restored-custom-db-instance` 的新資料庫執行個體，截至指定的時間。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier my-custom-db-instance \  
  --target-db-instance-identifier my-restored-custom-db-instance \  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \  
  --restore-time 2022-10-14T23:45:00.000Z
```

在 Windows 中：

```
aws rds restore-db-instance-to-point-in-time ^  
  --source-db-instance-identifier my-custom-db-instance ^  
  --target-db-instance-identifier my-restored-custom-db-instance ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^  
  --restore-time 2022-10-14T23:45:00.000Z
```

刪除 RDS Custom for Oracle 快照

當您不再需要時，可以刪除由 RDS Custom for Oracle 管理的資料庫快照。Amazon RDS 與 RDS Custom 資料庫執行個體的刪除程序相同。

二進位和根磁碟區的 Amazon EBS 快照會在您的帳戶中保留較長的時間，因為這些快照可能會連結到您帳戶中執行的某些執行個體或其他 RDS Custom for Oracle 快照。這些 EBS 快照不再與任何現有 RDS Custom for Oracle 資源 (資料庫執行個體或備份) 相關之後，會自動刪除這些快照。

主控台

若要刪除 RDS Custom 資料庫執行個體的快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要刪除的資料庫快照。
4. 針對 Actions (動作)，選擇 Delete Snapshot (刪除快照)。
5. 在確認頁面上，選擇 Delete (刪除)。

AWS CLI

若要刪除 RDS 自訂快照集，請使用 AWS CLI 命令 [delete-db-snapshot](#)。

下列選項是必要的：

- `--db-snapshot-identifier` – 要刪除的快照

以下範例會刪除 `my-custom-snapshot` 資料庫快照。

Example

對於LinuxmacOS、或Unix：

```
aws rds delete-db-snapshot \  
  --db-snapshot-identifier my-custom-snapshot
```

在 Windows 中：

```
aws rds delete-db-snapshot ^  
  --db-snapshot-identifier my-custom-snapshot
```

刪除 RDS Custom for Oracle 自動備份

不再需要保留 RDS Custom for Oracle 的自動備份時，刪除即可。該程序與刪除 Amazon RDS 備份的程序相同。

主控台

刪除保留的自動備份

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 選擇 Retained (已保留)。
4. 選擇您要刪除的已保留自動備份。
5. 對於 Actions (動作)，請選擇 Delete (刪除)。
6. 在確認頁面上，輸入 **delete me**，然後選擇 Delete (刪除)。

AWS CLI

您可以使用 AWS CLI 命令 [delete-db-instance-automated-backup](#) 刪除保留的自動備份。

下列選項用來刪除保留的自動備份：

- `--dbi-resource-id` – 來源 RDS Custom 資料庫執行個體的資源識別符。

您可以使用 AWS CLI 指令 [describe-db-instance-automated-backup](#)，找到保留自動備份的來源資料庫執行個體的資源識別碼。

下列範例刪除來源資料庫執行個體識別符為 `custom-db-123ABCEXAMPLE` 的已保留自動備份。

Example

對於 Linux/macOS、或 Unix：

```
aws rds delete-db-instance-automated-backup \  
  --dbi-resource-id custom-db-123ABCEXAMPLE
```

在 Windows 中：


```
aws rds delete-db-instance-automated-backup ^  
  --dbi-resource-id custom-db-123ABCEXAMPLE
```

在適用於 Oracle 的 RDS 自訂中使用選項群組

RDS 自訂使用選項群組來啟用和設定其他功能。選項群組會指定適用於 Oracle 資料庫執行個體的 RDS 自訂功能 (稱為選項)。選項擁有的設定，可以指定該選項如何運作。將 Oracle 資料庫執行個體的 RDS 自訂與選項群組建立關聯時，會啟用此執行個體的指定選項和選項設定。如需 Amazon RDS 中選項群組的一般資訊，請參閱[使用選項群組](#)。

主題

- [適用於 Oracle 的 RDS 自訂選項群組概觀](#)
- [Oracle 時區](#)

適用於 Oracle 的 RDS 自訂選項群組概觀

若要啟用您 Oracle 資料庫的選項，請將其新增至選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需詳細資訊，請參閱[使用選項群組](#)。

主題

- [適用於 Oracle 選項的 RDS 自訂摘要](#)
- [將選項新增至適用於 Oracle 資料庫執行個體的 RDS 自訂選項的基本步驟](#)
- [在 RDS 中建立選項群組自訂適用於甲骨文](#)
- [將選項群組與 Oracle 資料庫執行個體的 RDS 自訂產生關聯](#)

適用於 Oracle 選項的 RDS 自訂摘要

適用於 Oracle 的 RDS 自訂支援資料庫執行個體的下列選項。

選項	選項 ID	描述
Oracle 時區	Timezone	適用於 Oracle 資料庫執行個體的 RDS 自訂所使用的時區。

將選項新增至適用於 Oracle 資料庫執行個體的 RDS 自訂選項的基本步驟

將選項新增至適用於 Oracle 資料庫執行個體的 RDS 自訂的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。

2. 將選項新增至選項群組。
3. 建立或修改選項群組時，將選項群組與資料庫執行個體建立關聯。

在 RDS 中建立選項群組自訂適用於甲骨文

您可以建立新的選項群組，從預設選項群組衍生其設定。然後將一或多個選項新增至新的選項群組。或者，如果您已經有一個現有選項群組，您可以將包含其所有選項的選項群組複製到新的選項群組。若要瞭解如何複製選項群組，請參閱[刪除選項群組](#)。

適用於 Oracle 的 RDS 自訂的預設選項群組如下：

- default:custom-oracle-ee
- default:custom-oracle-se2
- default:custom-oracle-ee-cdb
- default:custom-oracle-se2-cdb

建立選項群組時，設定會衍生自預設選項群組。新增選項之後，您就可以將TIME_ZONE選項群組與資料庫執行個體建立關聯。

主控台

建立選項群組的其中一種方法是使用 AWS Management Console。

使用主控台建立新的選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 視窗中，執行下列動作：
 - a. 在 [名稱] 中，輸入 AWS 帳戶中唯一的選項群組名稱。名稱僅可包含字母、數字與連字號。
 - b. 在 Description (說明) 欄位中，輸入選項群組的簡要說明。用於顯示用途的說明。
 - c. 對於「引擎」，請選擇下列任何一個適用於 Oracle 資料庫引擎的 RDS 自訂：
 - custom-oracle-ee

- custom-oracle-se2
 - custom-oracle-ee-cdb
 - custom-oracle-se2-國別分組
- d. 對於主要引擎版本，請選擇適用於 Oracle 的 RDS 自訂支援的主要引擎版本。如需詳細資訊，請參閱 [適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。
5. 請選擇 Create (建立) 以繼續進行。若要取消操作，請改為選擇 Cancel (取消)。

AWS CLI

若要建立選項群組，請搭配下列必要參數使用 AWS CLI [create-option-group](#) 指令。

- --option-group-name
- --engine-name
- --major-engine-version
- --option-group-description

Example

以下範例會建立名為 testoptiongroup 的選項群組，該群組與 Oracle Enterprise Edition 資料庫引擎關聯。該說明包含在引號中。

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name testoptiongroup \  
  --engine-name custom-oracle-ee-cdb \  
  --major-engine-version 19 \  
  --option-group-description "Test option group for a Custom Oracle CDB"
```

在 Windows 中：

```
aws rds create-option-group ^  
  --option-group-name testoptiongroup ^  
  --engine-name custom-oracle-ee-cdb ^  
  --major-engine-version 19 ^  
  --option-group-description "Test option group for a Custom Oracle CDB"
```

RDS API

如要建立選項群組，請呼叫 Amazon RDS API [CreateOptionGroup](#) 操作。

將選項群組與 Oracle 資料庫執行個體的 RDS 自訂產生關聯

您可以將選項群組與新的或現有的資料庫執行個體建立關聯。

- 對於新的資料庫執行個體，請在建立執行個體時套用選項群組。如需詳細資訊，請參閱 [建立 RDS Custom for Oracle 資料庫執行個體](#)。
- 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。如需詳細資訊，請參閱 [修改您的 RDS Custom for Oracle 資料庫執行個體](#)。

Oracle 時區

若要變更 RDS 自訂適用於 Oracle 資料庫執行個體的系統時區，請使用時區選項。例如，您可以變更資料庫執行個體的時區，以與現場部署環境或舊型應用程式相容。時區選項會變更主機層級的時區。變更時區會影響所有日期欄和值，包括 SYSDATE 和 SYSTIMESTAMP。

主題

- [適用於 Oracle 的 RDS 自訂中的時區選項設定](#)
- [適用於 Oracle 的 RDS 自訂中的可用時區](#)
- [在適用於 Oracle 的 RDS 自訂中設定時區的考量事項](#)
- [適用於 Oracle 的 RDS 自訂中的時區設定限制](#)
- [將時區選項新增至選項群組](#)
- [移除時區選項](#)

適用於 Oracle 的 RDS 自訂中的時區選項設定

Amazon RDS 支援時區選項的下列設定。

選項設定	有效值	描述
TIME_ZONE	其中一個可用的時區。如需完整清單，請參閱 適用於 Oracle 的 RDS 自訂中的可用時區 。	資料庫執行個體的新時區。

適用於 Oracle 的 RDS 自訂中的可用時區

您可對時區選項使用下列值。

區域	時區
非洲	非洲/開羅，非洲/卡薩布蘭卡，非洲/哈拉雷，非洲/拉哥斯，非洲/魯安達，非洲/蒙羅維亞，非洲/奈洛比，非洲/的黎波里，非洲/溫吐克
美洲	美洲/阿拉瓜伊納，美洲/阿根廷/布宜諾斯艾利斯，美洲/亞松森，美洲/波哥大，美洲/卡拉卡斯，美洲/芝加哥，美洲/赤瓦瓦州，美洲/古雅巴，美洲/丹佛，美洲/底特律，美洲/福塔力沙，美洲/哥特哈布，美洲/瓜地馬拉，美洲/哈利法克斯，美洲/利馬，美洲/洛杉磯，美洲/瑪瑙斯，美洲/馬塔莫羅斯，美洲/墨西哥市，美洲/蒙特利，美洲/蒙特維多，美洲/紐約州，美洲/鳳凰城，美洲/聖地牙哥，美洲/聖保羅，美洲/提華納，美洲/多倫多
亞洲	亞洲/安曼，亞洲/阿什哈巴德，亞洲/巴格達，亞洲/巴庫，亞洲/曼谷，亞洲/貝魯特，亞洲/加爾各答，亞洲/大馬士革，亞洲/達卡，亞洲/香港，亞洲/伊爾庫次克，亞洲/雅加達，亞洲/耶路撒冷，亞洲/喀布爾，亞洲/喀拉蚩，亞洲/加德滿都，亞洲/加爾各答，亞洲/克拉斯諾雅斯克，亞洲/馬加丹，亞洲/馬尼拉，亞洲/馬斯喀特，亞洲/新西伯利亞，亞洲/仰光，亞洲/利雅德，亞洲/首爾，亞洲/上海，亞洲/新加坡，亞洲/台北，亞洲/德黑蘭，亞洲/東京，亞洲/庫倫，亞洲/海參崴，亞洲/雅庫次克，亞洲/葉里溫
大西洋	大西洋/亞速爾群島，大西洋/維德角群島
澳大利亞	澳洲/阿德萊德，澳洲/布里斯班，澳洲/達爾文，澳洲/尤克拉，澳洲/荷巴特，澳洲/羅豪，澳洲/伯斯，澳洲/雪梨
巴西	巴西/, 巴西/東DeNoronha部
加拿大	加拿大/紐芬蘭，加拿大/薩克其萬省
等等	等等/GMT-3
歐洲	歐洲/阿姆斯特丹，歐洲/雅典，歐洲/柏林，歐洲/都柏林，歐洲/赫爾辛基，歐洲/加里寧格勒，歐洲/倫敦，歐洲/馬德里，歐洲/莫斯科，歐洲/巴黎，歐洲/布拉格，歐洲/羅馬，歐洲/塞拉耶佛

區域	時區
太平洋	太平洋/亞庇，太平洋/奧克蘭，太平洋/查塔姆，太平洋/斐濟，太平洋/關島，太平洋/檀香山，太平洋/刻里提瑪斯，太平洋/馬克沙斯，太平洋/薩摩亞，太平洋/東加塔普，太平洋/威克
美國	美國/阿拉斯加，美國/中部，美國/東印第安納州，美國/東部，美國/太平洋
UTC	UTC

在適用於 Oracle 的 RDS 自訂中設定時區的考量事項

如果您選擇設定資料庫執行個體的時區，請考慮下列事項：

- 當您新增時區選項時，在資料庫執行個體自動重新啟動時會發生短暫的中斷。
- 如果您不小心設定為錯誤的時區，則必須將您的資料庫執行個體復原為先前的時區設定。因此，我們強烈建議您在將時區選項新增至執行個體之前，先使用下列其中一種策略：
 - 如果您的 RDS 自訂適用於 Oracle 資料庫執行個體使用預設選項群組，請建立資料庫執行個體的快照。如需詳細資訊，請參閱 [建立 RDS Custom for Oracle 快照](#)。
 - 如果您的資料庫執行個體目前使用非預設選項群組，請建立資料庫執行個體的快照，然後使用時區選項建立新的選項群組。
- 強烈建議您在套用 Timezone 選項後手動備份資料庫執行個體。
- 強烈建議您在將測試資料庫執行個體新增至生產資料庫執行個體之前，先在測試資料庫執行個體上測試時區選項。新增時區選項可能導致表格使用系統日期來新增日期或時間的問題。我們建議您分析資料和應用程式，來評估變更時區的影響。

適用於 Oracle 的 RDS 自訂中的時區設定限制

注意下列限制：

- 您無法直接在主機上變更時區，而不會將其移至支援範圍之外。若要變更資料庫時區，您必須建立選項群組。
- 由於時區選項是永久性選項 (但不是永久選項)，因此您無法執行下列動作：
 - 在新增該選項之後，從選項群組中將其移除。
 - 將選項的時區設定修改為其他時區。
- 您無法將多個選項群組與 Oracle 資料庫執行個體的 RDS 自訂產生關聯。

- 您無法在 CDB 中為個別 PDB 設定時區。

將時區選項新增至選項群組

適用於 Oracle 的 RDS 自訂的預設選項群組如下：


- default:custom-oracle-ee
- default:custom-oracle-se2
- default:custom-oracle-ee-cdb
- default:custom-oracle-se2-cdb

建立選項群組時，設定會衍生自預設選項群組。如需 Amazon RDS 中選項群組的一般資訊，請參閱[使用選項群組](#)。

主控台

將時區選項新增至選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您要修改的選項群組，然後選擇 Add option (新增選項)。
4. 在 Add option (新增選項) 視窗中，執行下列作業：
 - a. 選擇「時區」。
 - b. 在選項設定中，選擇時區。
 - c. 若要在新增所有相關聯的 Oracle 資料庫執行個體之 RDS 自訂時立即啟用該選項，請針對「立即套用」選擇是。如果您選擇否 (預設值)，則會在下一個維護時段期間為每個關聯的資料庫執行個體啟用該選項。
 - d.

 **Important**

如果您將時區選項新增至已附加至一個或多個資料庫執行個體的現有選項群組，則在所有資料庫執行個體自動重新啟動時會發生短暫的中斷。
5. 當您滿意設定後，選擇 Add option (新增選項)。

6. 備份其時區已更新之 Oracle 資料庫執行個體的 RDS 自訂功能。如需詳細資訊，請參閱 [建立 RDS Custom for Oracle 快照](#)。

AWS CLI

下列範例會使用 AWS CLI [add-option-to-option-group](#) 指令，將Timezone選項和選TIME_ZONE項設定新增至名為testoptiongroup的選項群組。時區會設為 America/Los_Angeles。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name "testoptiongroup" \  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/  
Los_Angeles}]" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name "testoptiongroup" ^  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/  
Los_Angeles}]" ^  
  --apply-immediately
```

移除時區選項

時區選項是永久性選項，但不是永久選項。在新增此選項之後，您無法從選項群組中移除它。若要取消舊選項群組與資料庫執行個體的關聯：

1. 使用更新的選項建立新Timezone選項群組。
2. 修改執行個體時，將新的選項群組與資料庫執行個體建立關聯。

將內部部署資料庫遷移至 RDS Custom for Oracle

將內部部署 Oracle 資料庫遷移至 RDS Custom for Oracle 之前，您必須考量下列因素：

- 應用程式可承受的停機時間
- 來源資料庫的大小
- 網路連線能力
- 備用計劃的需求
- 來源和目標 Oracle 資料庫版本和資料庫執行個體作業系統類型
- 可用的複寫工具，例如 AWS Database Migration Service、Oracle GoldenGate 或第三方複寫工具

根據這些因素，您可以選擇實體遷移、邏輯遷移或組合。如果選擇實體遷移，您可以使用下列技術：

RMAN 複製

作用中資料庫複製不需要備份來源資料庫。它會透過網路將資料庫檔案複製到輔助執行個體，將即時來源資料庫複製到目的地主機。RMAN DUPLICATE 指令會將必要檔案複製為映像複本或備份集。若要了解這項技術，請參閱 AWS 部落格文章：[使用 RMAN 複製將 Oracle 資料庫實體遷移至 Amazon RDS Custom。](#)

Oracle Data Guard

在此技術中，您會備份主要內部部署資料庫，並將這些備份複製到 Amazon S3 儲存貯體。然後，將這些備份複製到 RDS Custom for Oracle 待命資料庫執行個體。在執行必要的組態之後，您可以手動將主要資料庫切換至 RDS Custom for Oracle 待命資料庫。若要了解這項技術，請參閱 AWS 部落格文章：[使用資料防護將 Oracle 資料庫實體遷移至 Amazon RDS Custom。](#)

如需有關以邏輯方式將資料匯入至 RDS for Oracle 的一般資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle。](#)

升級 Amazon RDS Custom for Oracle 的資料庫執行個體

您可以藉由將 Amazon RDS Custom 資料庫執行個體修改為新的自訂引擎版本 (CEV) 來進行升級。如需升級的一般資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

主題

- [RDS Custom for Oracle 中的升級概觀](#)
- [RDS Custom for Oracle 升級的需求](#)
- [Oracle 資料庫升級的 RDS 自訂考量](#)
- [Oracle 作業系統升級的 RDS 自訂注意事項](#)
- [檢視 RDS Custom for Oracle 資料庫執行個體的有效 CEV 升級目標](#)
- [升級 RDS Custom for Oracle 資料庫執行個體](#)
- [檢視 RDS Custom 資料庫執行個體的待定資料庫升級](#)
- [RDS Custom for Oracle 資料庫執行個體升級失敗的疑難排解](#)

RDS Custom for Oracle 中的升級概觀

使用 RDS Custom for Oracle，您可以建立新的 CEV，然後修改執行個體以使用新 CEV，來修補 Oracle 資料庫或資料庫執行個體作業系統 (OS)。

主題

- [CEV 升級選項](#)
- [在沒有 CEV 的情況下進行修補](#)
- [使用 CEV 修補資料庫執行個體的一般步驟](#)

CEV 升級選項

當您建立升級的 CEV 時，您有下列互斥選項：

僅限資料庫

重複使用資料庫執行個體目前使用中的 Amazon Machine Image (AMI)，但指定不同的資料庫二進位檔案。RDS Custom 會配置新的二進位磁碟區，然後將其附加到現有的 Amazon EC2 執行個體。RDS Custom 會將整個資料庫磁碟區取代為使用目標資料庫版本的新磁碟區。

僅限作業系統

重複使用資料庫執行個體目前使用中的資料庫二進位檔案，但指定不同的 AMI。RDS Custom 會配置新的 Amazon EC2 執行個體，然後將現有的二進位磁碟區附加到新的執行個體。現有的資料庫磁碟區會加以保留。

如果您想要同時升級作業系統和資料庫，則必須升級 CEV 兩次。您可以先升級作業系統，然後再升級資料庫，或先升級資料庫，然後再升級作業系統。

Warning

當您修補作業系統時，您會遺失根磁碟區資料和任何現有的作業系統自訂。因此，我們強烈建議您不要使用安裝或儲存永久資料或檔案的根磁碟區。我們也建議您在升級前，先備份資料。

在沒有 CEV 的情況下進行修補

強烈建議您使用 CEV 升級 RDS Custom for Oracle 資料庫執行個體。RDS Custom for Oracle 會自動將修補程式中繼資料與資料庫執行個體上的資料庫二進位同步。

在特殊情況下，RDS Custom 支援使用 OPatch 公用程式直接將「一次性」資料庫修補程式套用至基礎 Amazon EC2 執行個體。有效的使用情況可能是您要立即套用的資料庫修補程式，但 RDS Custom 群組正在升級 CEV 功能，進而造成延遲。若要手動套用資料庫修補程式，請執行下列步驟：

1. 暫停 RDS Custom 自動化。
2. 將修補程式套用至 Amazon EC2 執行個體上的資料庫二進位。
3. 繼續 RDS Custom 自動化。

執行上述技巧的缺點是，您必須手動將資料庫修補程式套用至每個要升級的執行個體。相反，當您建立新的 CEV 時，您可以使用相同的 CEV 建立或升級多個資料庫執行個體。

使用 CEV 修補資料庫執行個體的一般步驟

無論您修補作業系統還是資料庫，都要執行下列基本步驟：

1. 根據您要修補資料庫還是作業系統，建立包含下列任一項目的 CEV：
 - 您要套用到資料庫執行個體的 Oracle Database RU
 - 不同的 AMI (最新的可用 AMI 或您指定的 AMI) 以及用作來源的現有 CEV

請遵循 [建立 CEV](#) 中的步驟。

2. (可供資料庫修補選擇) 執行 `describe-db-engine-versions` 以檢查可用的引擎版本升級。
3. 執行 `modify-db-instance` 以啟動修補程序。

要修補的執行個體狀態會有所不同，如下所示：

- RDS 修補資料庫時，資料庫執行個體的狀態將變更為升級中。
- RDS 修補作業系統時，資料庫執行個體的狀態將變更為修改中。

資料庫執行個體的狀態為可用時，表示修補已完成。

4. 執行 `describe-db-instances` 以確認您的資料庫執行個體使用新的 CEV。

RDS Custom for Oracle 升級的需求

若要將 RDS Custom for Oracle 資料庫執行個體升級為目標 CEV，請確定符合下列需求：

- 您要升級的目標 CEV 必須存在。
- 您必須在單一操作中升級作業系統或資料庫。不支援在單一 API 呼叫中同時升級作業系統和資料庫。
- 目標 CEV 必須使用目前 CEV 清單檔案中的安裝參數設定。例如，您無法將使用預設 Oracle 主目錄的資料庫升級至使用非預設 Oracle 主目錄的 CEV。
- 對於資料庫升級，目標 CEV 必須使用新的次要資料庫版本，而不是新的主要版本。例如，您無法從 Oracle 資料庫 12c CEV 升級至 Oracle 資料庫 19c CEV。但您可以從版本 21.0.0.0.ru-2023-04.rur-2023-04.r1 升級至版本 21.0.0.0.ru-2023-07.rur-2023-07.r1。
- 對於作業系統升級，目標 CEV 必須使用不同的 AMI，但具有相同的主要版本。

Oracle 資料庫升級的 RDS 自訂考量

如果您打算升級資料庫，請考慮下列事項：

- 當您升級主要資料庫執行個體中的資料庫二進位檔案時，RDS Custom for Oracle 會自動升級您的僅供讀取複本。升級作業系統時，您必須手動升級僅供讀取複本。
- 當您將容器資料庫 (CDB) 升級到新的資料庫版本時，Oracle 的 RDS 自訂會檢查所有 PDB 都已開啟或可開啟。如果不符合這些條件，RDS Custom 會停止檢查，並將資料庫恢復為原始狀態，而不會嘗試升級。如果符合條件，RDS Custom 會首先修補 CDB 根，然後平行修補所有其他 PDB (包括 PDB\$SEED)。

修補完成後，RDS 自訂會嘗試開啟所有 PDB。若有任何 PDB 無法開啟，您會收到下列事件：The following PDBs failed to open: *List-of-PDBs*。如果 RDS Custom 無法修補 CDB 根或任何 PDB，則執行個體會進入 PATCH_DB_FAILED 狀態。

- 您可能想要同時執行主要資料庫版本升級，以及將非 CDB 轉換為 CDB。在此情況下，建議您依照下述進行：
 1. 建立使用 Oracle 多租用戶架構的新 RDS 自訂適用於 Oracle 資料庫執行個體。
 2. 將非 CDB 插入您的 CDB 根中，將其建立為 PDB。確定非 CDB 的主要版本與您的 CDB 相同。
 3. 通過運行 `noncdb_to_pdb.sql` 甲骨文 SQL 腳本來轉換您的 PDB。
 4. 驗證您的 CDB 執行個體。
 5. 升級您的 CDB 執行個體。

Oracle 作業系統升級的 RDS 自訂注意事項

當您規劃作業系統升級時，請考慮下列事項：

- 您無法提供您自己的 AMI，以便在 Oracle CEV 的 RDS 自訂中使用。您可以指定預設 AMI 或先前已由 Oracle CEV 的 RDS 自訂使用的 AMI。

Note

Oracle 版 RDS 自訂會在發現常見弱點和暴露時發佈新的預設 AMI。沒有固定的時間表可用或保證。適用於甲骨文的 RDS 自定義傾向於每 30 天發布一次新的默認 AMI。

- 升級主要資料庫執行個體中的作業系統時，必須手動升級其關聯的僅供讀取複本。
- 開始修補作業系統之前，請為 AZ 中的執行個體類型保留足夠的 Amazon EC2 運算容量。

建立容量保留時，您必須指定 AZ、執行個體數目和執行個體屬性 (包括執行個體類型)。例如，如果您的資料庫執行個體使用基礎 EC2 執行個體類型 `r5.large`，建議您在 AZ 中為 `r5.large` 保留 EC2 容量。在作業系統修補期間，RDS Custom 會建立類型為 `db.r5.large` 的新主機，如果 AZ 缺少此執行個體類型的 EC2 容量，則該主機可能會停滯。如果您保留 EC2 容量，則可以降低因容量限制而導致修補遭到封鎖的風險。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [隨需容量保留](#)。

- 在升級其作業系統之前，請先備份資料庫執行個體。升級會移除您的根磁碟區資料和任何現有的作業系統自訂。

- 在共同的責任模型中，您必須負責讓作業系統保持在最新狀態。適用於 Oracle 的 RDS 自訂不會強制您將哪些修補程式套用至作業系統。如果適用於 Oracle 的 RDS 自訂功能正常，您可以無限期地使用與此 CEV 相關聯的 AMI。

檢視 RDS Custom for Oracle 資料庫執行個體的有效 CEV 升級目標

您可以在 AWS Management Console 中的 Custom engine versions (自訂引擎版本) 頁面中看到現有的 CEV。

您也可以使用[描述-db-Engine 版本](#) AWS CLI 命令來尋找升級資料庫執行個體時要使用的有效 CEV，如下列範例所示。此範例假設您已使用引擎版本 19.my_cev1 建立資料庫執行個體，並假設升級版本 19.my_cev2 和 19.my_cev 存在。

```
aws rds describe-db-engine-versions --engine custom-oracle-ee --engine-version
19.my_cev1
```

輸出結果與以下內容相似。ImageId 欄位會顯示 AMI ID。

```
{
  "DBEngineVersions": [
    {
      "Engine": "custom-oracle-ee",
      "EngineVersion": "19.my_cev1",
      ...
      "Image": {
        "ImageId": "ami-2345",
        "Status": "active"
      },
      "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-
oracle-ee/19.my_cev1/12a34b5c-67d8-90e1-2f34-gh56ijk78lm9"
      "ValidUpgradeTarget": [
        {
          "Engine": "custom-oracle-ee",
          "EngineVersion": "19.my_cev2",
          "Description": "19.my_cev2 description",
          "AutoUpgrade": false,
          "IsMajorVersionUpgrade": false
        },
        {
          "Engine": "custom-oracle-ee",
          "EngineVersion": "19.my_cev3",
```



```
        "Description": "19.my_cev3 description",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    }
]
...
```

升級 RDS Custom for Oracle 資料庫執行個體

若要升級 RDS Custom for Oracle 資料庫執行個體，請將其修改為使用新的 CEV。此 CEV 可以包含新的資料庫二進位檔案或新的 AMI。如果您想要升級資料庫和作業系統，則必須執行兩個不同的升級。

Note

如果您升級資料庫，RDS Custom 會在升級主要資料庫執行個體之後自動升級僅供讀取複本。如果您升級作業系統，您必須手動升級這些複本。

開始之前，請先檢閱 [RDS Custom for Oracle 升級的需求](#) 和 [Oracle 資料庫升級的 RDS 自訂考量](#)。

主控台

升級 RDS Custom for Oracle 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇資料庫，然後選擇您要升級的 RDS Custom for Oracle 資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 對於資料庫引擎版本，選擇新的 CEV。請執行下列操作：
 - 如果您要修補資料庫，請確定 CEV 指定的資料庫二進位檔與資料庫執行個體所使用的資料庫二進位檔案不同，而且未指定與資料庫執行個體目前所用 AMI 不同的 AMI。
 - 如果您要修補作業系統，請確定 CEV 指定的 AMI 與資料庫執行個體所使用的 AMI 不同，而且未指定不同的資料庫二進位檔案。

Warning

當您修補作業系統時，您會遺失根磁碟區資料和任何現有的作業系統自訂。

5. 選擇 Continue (繼續) 以檢查修改的摘要。

若要立即套用變更，請選擇 Apply immediately (立即套用)。

6. 如果您的變更正確，請選擇 Modify DB instance (修改資料庫執行個體)。或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

下列範例顯示可能的升級案例。這些範例假設您已建立 RDS Custom for Oracle 資料庫執行個體，其具有下列特性：

- 名為 my-custom-instance 的資料庫執行個體
- 名為 19.my_cev1 的 CEV
- Oracle Database 19c 使用非 CDB 架構
- 使用 AMI ami-1234 的 Oracle Linux 7.9

最新服務提供的 AMI 為 ami-2345。您可以執行 CLI 命令 describe-db-engine-versions 來尋找您的 AMI：

主題

- [升級作業系統](#)
- [升級資料庫](#)

升級作業系統

在此範例中，您想要將 ami-1234 升級至 ami-2345，這是最新服務提供的 AMI。因為這是作業系統升級，所以 ami-1234 和 ami-2345 的資料庫二進位檔案必須相同。您建立一個以 19.my_cev1 為基礎且名為 19.my_cev2 的新 CEV。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-custom-db-engine-version \  
  --engine custom-oracle-ee \  
  --engine-version 19.my_cev2 \  
  --description "Non-CDB CEV based on ami-2345" \  
  --kms-key-id key-name \  
  --tags key=value
```

```
--source-custom-db-engine-version-identifer arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-abcde123456789 \  
--image-id ami-2345
```

在 Windows 中：

```
aws rds create-custom-db-engine-version ^  
--engine custom-oracle-ee ^  
--engine-version 19.my_cev2 ^  
--description "Non-CDB CEV based on ami-2345" ^  
--kms-key-id key-name ^  
--source-custom-db-engine-version-identifer arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-abcde123456789 ^  
--image-id ami-2345
```

若要升級 RDS Custom 資料庫執行個體，請使用 [modify-db-instance](#) AWS CLI 命令搭配下列參數：

- `--db-instance-identifier` - 指定要升級的 RDS Custom for Oracle 資料庫執行個體。
- `--engine-version` - 指定具有新 AMI 的 CEV。
- `--no-apply-immediately` | `--apply-immediately` - 指定要立即執行升級，還是等到排定的維護時段。

下列範例會將 `my-custom-instance` 升級到版本 `19.my_cev2`。僅升級作業系統。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
--db-instance-identifier my-custom-instance \  
--engine-version 19.my_cev2 \  
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
--db-instance-identifier my-custom-instance ^  
--engine-version 19.my_cev2 ^
```

```
--apply-immediately
```

升級資料庫

在此範例中，您想要將 Oracle 修補程式 p35042068 套用至 RDS for Oracle 資料庫執行個體。因為您已在 [升級作業系統](#) 中升級作業系統，所以您的資料庫執行個體目前正在使用 19.my_cev2，其是以 ami-2345 為基礎。您建立名為 19.my_cev3 的新 CEV，其也會使用 ami-2345，但您可以在 \$MANIFEST 環境變數中指定新的 JSON 清單檔案。因此，資料庫二進位檔案只在新 CEV 和執行個體目前使用的 CEV 中不同。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-custom-db-engine-version \  
  --engine custom-oracle-ee \  
  --engine-version 19.my_cev3 \  
  --description "Non-CDB CEV with p35042068 based on ami-2345" \  
  --kms-key-id key-name \  
  --image-id ami-2345 \  
  --manifest $MANIFEST
```

在 Windows 中：

```
aws rds create-custom-db-engine-version ^  
  --engine custom-oracle-ee ^  
  --engine-version 19.my_cev3 ^  
  --description "Non-CDB CEV with p35042068 based on ami-2345" ^  
  --kms-key-id key-name ^  
  --image-id ami-2345 ^  
  --manifest $MANIFEST
```

下列範例會將 my-custom-instance 升級到引擎版本 19.my_cev3。僅升級資料庫。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --engine-version 19.my_cev3 \  
  --apply-immediately
```

```
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-custom-instance ^  
  --engine-version 19.my_cev3 ^  
  --apply-immediately
```

檢視 RDS Custom 資料庫執行個體的待定資料庫升級

您可以使用描述的 [DB 執行個體或描述暫停維護動作命令](#)，查看 Amazon RDS 自訂資料庫執行個體的 [擱置中資料庫升級](#)。AWS CLI

不過，如果您使用 `--apply-immediately` 選項或是升級正在進行中，則這個方法無法運作。

下列 `describe-db-instances` 命令會顯示 `my-custom-instance` 的待定資料庫升級。

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

輸出結果與以下內容相似。

```
{  
  "DBInstances": [  
    {  
      "DBInstanceIdentifier": "my-custom-instance",  
      "EngineVersion": "19.my_cev1",  
      ...  
      "PendingModifiedValues": {  
        "EngineVersion": "19.my_cev3"  
      }  
      ...  
    }  
  ]  
}
```

RDS Custom for Oracle 資料庫執行個體升級失敗的疑難排解

如果 RDS Custom 資料庫執行個體升級失敗，則會產生 RDS 事件且資料庫執行個體狀態會變成 `upgrade-failed`。

您可以使用 [describe-db-instance AWS CLI 命令](#) 來查看此狀態，如下列範例所示。

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

輸出結果與以下內容相似。

```
{
  "DBInstances": [
    {
      "DBInstanceIdentifier": "my-custom-instance",
      "EngineVersion": "19.my_cev1",
      ...
      "PendingModifiedValues": {
        "EngineVersion": "19.my_cev3"
        ...
      }
      "DBInstanceStatus": "upgrade-failed"
    }
  ]
}
```

升級失敗之後，除了修改資料庫執行個體以執行下列工作之外，所有資料庫動作都會遭到封鎖：

- 重試相同的升級
- 暫停和繼續 RDS Custom 自動化
- Point-in-time 回收 (PITR)
- 刪除資料庫執行個體

Note

如果已暫停 RDS Custom 資料庫執行個體的自動化，則在繼續自動化之前，您無法重試升級。相同的動作適用於 RDS 受管僅供讀取複本的升級失敗，與主要複本的升級失敗相同。

如需更多詳細資訊，請參閱 [針對 RDS Custom for Oracle 升級進行疑難排解](#)。

針對 Amazon RDS Custom for Oracle 的資料庫問題進行故障診斷

RDS Custom 的共同的責任模型提供作業系統殼層層級存取權和資料庫管理員存取權。RDS Custom 會執行您的帳戶中的資源，不像是 Amazon RDS，它會執行系統帳戶中的資源。責任越大，存取權也越大。在下列區段中，您可以了解如何針對 Amazon RDS Custom 資料庫執行個體的問題進行疑難排解。

Note

本節說明如何對 RDS Custom for Oracle 進行故障診斷。若要對 RDS Custom for SQL Server 進行故障診斷，請參閱 [針對 Amazon RDS Custom for SQL Server 的資料庫問題進行故障診斷](#)。

主題

- [檢視 RDS Custom 事件](#)
- [訂閱 RDS 自訂事件](#)
- [針對 RDS Custom for Oracle 的自訂引擎版本建立進行疑難排解](#)
- [修正 RDS Custom for Oracle 中不支援的組態](#)
- [針對 RDS Custom for Oracle 升級進行疑難排解](#)
- [RDS Custom for Oracle 複本提升的疑難排解](#)

檢視 RDS Custom 事件

RDS Custom 和 Amazon RDS 資料庫執行個體的檢視事件程序相同。如需詳細資訊，請參閱 [檢視 Amazon RDS 事件](#)。

若要使用檢視 RDS 自訂事件通知 AWS CLI，請使用 `describe-events` 命令。RDS Custom 引入數個新事件。事件類別與 Amazon RDS 的類別相同。如需事件清單，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

下列範例會擷取針對指定 RDS Custom 資料庫執行個體發生之事件的詳細資訊。

```
aws rds describe-events \  
  --source-identifier my-custom-instance \  
  --source-type db-instance
```

訂閱 RDS 自訂事件

RDS Custom 與 Amazon RDS 資料庫執行個體的訂閱事件程序相同。如需詳細資訊，請參閱 [訂閱 Amazon RDS 事件通知](#)。

若要使用 CLI 來訂閱 RDS Custom 事件通知，請使用 `create-event-subscription` 命令。包含下列必要參數：

- `--subscription-name`
- `--sns-topic-arn`

下列範例會在目前 AWS 帳戶中建立 RDS Custom 資料庫執行個體之備份和復原事件的訂閱。通知會傳送至 Amazon Simple Notification Service (Amazon SNS) 主題，由 `--sns-topic-arn` 指定。

```
aws rds create-event-subscription \  
  --subscription-name my-instance-events \  
  --source-type db-instance \  
  --event-categories '["backup","recovery"]' \  
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

針對 RDS Custom for Oracle 的自訂引擎版本建立進行疑難排解

當 CEV 建立失敗時，會發生 RDS Custom 問題 `RDS-EVENT-0198` 並且有訊息 `Creation failed for custom engine version major-engine-version.cev_name`，其中包含失敗的詳細資訊。例如，事件會列印遺失的檔案。

CEV 建立可能因為發生下列問題而失敗：

- 包含安裝檔案的 Amazon S3 儲存貯體與 CEV 位於不同的 AWS 區域。
- 當您第一次在中請求建立 CEV 時，RDS 自訂會建立用於存放 RDS 自訂資源 (例如 CEV 成品、AWS CloudTrail 記錄和交易記錄) 的 S3 儲存貯體。AWS 區域

如果 RDS Custom 無法建立 S3 儲存貯體，則 CEV 建立會失敗。呼叫者沒有如 [步驟 5：將必要許可授予 IAM 使用者或角色](#) 中所述的 S3 許可，或者 S3 儲存貯體的數量已達到限制。

- 呼叫者沒有從包含安裝媒體檔案的 S3 儲存貯體中取得檔案的許可。這些許可在 [步驟 7：新增必要的 IAM 許可](#) 中描述。
- 您的 IAM 政策具有 `aws:SourceIp` 條件。請務必遵循《AWS Identity and Access Management 使用者指南》中的 [AWS 根據來源 IP 拒絕存取 AWS](#)。另外，請確認呼叫者擁有 S3 許可，如 [步驟 5：將必要許可授予 IAM 使用者或角色](#) 中所述。

- CEV 資訊清單中列出的安裝媒體檔案不在 S3 儲存貯體中。
- RDS Custom 的安裝檔案的 SHA-256 檢查總和未知。

請確認提供的檔案的 SHA-256 檢查總和符合 Oracle 網站上的 SHA-256 檢查總和。如果檢查總和相符，請聯絡 [AWS 支援](#)，並提供失敗的 CEV 名稱、檔案名稱和檢查總和。

- OPatch 版本與修補程式檔案不相容。您可能會收到下列訊息：OPatch is lower than minimum required version. Check that the version meets the requirements for all patches, and try again。如要套用 Oracle 修補程式，必須使用相容版本的 OPatch 公用程式。您可在修補程式的讀我檔案中找到 Opatch 公用程式的所需版本。從 My Oracle Support 下載最新的 OPatch 公用程式，然後再次嘗試建立 CEV。
- CEV 資訊清單中指定的修補程式順序錯誤。

您可以在 RDS 主控台上 (在導覽窗格中選擇「事件」) 或使用 describe-events AWS CLI 命令來檢視 RDS 事件。預設持續時間為 60 分鐘。如果未傳回任何事件，請指定較長的持續時間，如以下範例所示。

```
aws rds describe-events --duration 360
```

目前，從 Amazon S3 匯入檔案以建立 CEV 的 MediaImport 服務並未與 AWS CloudTrail 整合。因此，如果您開啟 Amazon RDS 的資料記錄功能 CloudTrail，則不會記錄對 MediaImport 服務 (例如 CreateCustomDbEngineVersion 事件) 的呼叫。

不過，您可能會看到來自可存取 Amazon S3 儲存貯體的 API Gateway 的呼叫。這些呼叫來自 CreateCustomDbEngineVersion 事件的 MediaImport 服務。

修正 RDS Custom for Oracle 中不支援的組態

在共同的責任模型中，您必須負責修正讓 RDS Custom for Oracle 資料庫執行個體進入 unsupported-configuration 狀態的組態問題。如果問題出在 AWS 基礎結構上，您可以使用主控台或修正此問題。AWS CLI 如果問題與作業系統或資料庫組態相關，您可以登入主機進行修正。

Note

本節說明如何修正 RDS Custom for Oracle 中不支援的組態。如需 RDS Custom for SQL Server 的相關資訊，請參閱 [修正 RDS Custom for SQL Server 中不支援的組態](#)。

您可以在下表中找到支援周邊所傳送通知和事件的說明，以及了解如何進行修正。這些通知和支援周邊可能會變更。如需支援周邊的相關背景，請參閱 [RDS Custom 支援周邊](#)。如需事件說明，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

事件 ID	組態	RDS 事件訊息	動作
SP-00000	手動不受支援的	#於以下原因，RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]。	若要解決此問題，請建立 AWS Support 案例。
AWS 資源 (基礎架構)			
SP-O1001	Amazon Elastic Block Store (Amazon EBS) 磁碟區	## EBS ##### ## EC2 ##### ec2_id#####若要解決此問題，請從執行個體中卸離指定的磁碟區。	<p>除了從 Amazon 機器映像 (AMI) 建立的根磁碟區外，RDS 自訂還會建立兩種類型的 EBS 磁碟區，並將它們與 EC2 執行個體建立關聯：</p> <ul style="list-style-type: none"> • 資料庫軟體二進位檔所在的二進位磁碟區 • 資料庫檔案所在的資料磁碟區 <p>建立資料庫執行個體時，您指定的儲存組態會設定資料磁碟區。</p> <p>支援周邊會監控下列項目：</p> <ul style="list-style-type: none"> • 使用資料庫執行個體建立的初始 EBS 磁碟區仍與執行個體相關聯。 • 初始 EBS 磁碟區仍具有與初始設定相同的組態：儲存類型、大小、佈建 IOPS 和儲存輸送量。 • 沒有額外 EBS 磁碟區連接至資料庫執行個體。 <p>使用下列 CLI 命令來比較 EBS 磁碟區詳細資料的磁碟區類型和適用於 Oracle 資料庫執行個體的 RDS 自訂詳細資訊：</p> <pre>aws rds describe-db-instances \</pre>

事件 ID	組態	RDS 事件訊息	動作
			<pre>--db-instance-identifier db-instance-name grep StorageType</pre>
SP-O1002	Amazon Elastic Block Store (Amazon EBS) 磁碟區	<p><i>EBS ##### ### EC2 ##### [ec2_id] ###</i>您無法從這個執行個體卸離原始磁碟區。若要解決此問題，請將##### <i>### ec2_id #</i></p>	<p>除了從 Amazon 機器映像 (AMI) 建立的根磁碟區外，RDS 自訂還會建立兩種類型的 EBS 磁碟區，並將它們與 EC2 執行個體建立關聯：</p> <ul style="list-style-type: none"> • 資料庫軟體二進位檔所在的二進位磁碟區 • 資料庫檔案所在的資料磁碟區 <p>建立資料庫執行個體時，您指定的儲存組態會設定資料磁碟區。</p> <p>支援周邊會監控下列項目：</p> <ul style="list-style-type: none"> • 使用資料庫執行個體建立的初始 EBS 磁碟區仍與執行個體相關聯。 • 初始 EBS 磁碟區仍具有與初始設定相同的組態：儲存類型、大小、佈建 IOPS 和儲存輸送量。 • 沒有額外 EBS 磁碟區連接至資料庫執行個體。 <p>使用下列 CLI 命令來比較 EBS 磁碟區詳細資料的磁碟區類型和適用於 Oracle 資料庫執行個體的 RDS 自訂詳細資訊：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier db-instance-name grep StorageType</pre>

事件 ID	組態	RDS 事件訊息	動作
SP-O1003	Amazon Elastic Block Store (Amazon EBS) 磁碟區	<pre>### EC2 ### # ec2_id ## # EBS ##### ##### [X] # [Y]## [N] ## [M] # IOPS [J] # [K]#</pre> <p>若要解決此問題，請回復修改。</p>	<p>除了從 Amazon 機器映像 (AMI) 建立的根磁碟區外，RDS 自訂還會建立兩種類型的 EBS 磁碟區，並將它們與 EC2 執行個體建立關聯：</p> <ul style="list-style-type: none"> 資料庫軟體二進位檔所在的二進位磁碟區 資料庫檔案所在的資料磁碟區 <p>建立資料庫執行個體時，您指定的儲存組態會設定資料磁碟區。</p> <p>支援周邊會監控下列項目：</p> <ul style="list-style-type: none"> 使用資料庫執行個體建立的初始 EBS 磁碟區仍與執行個體相關聯。 初始 EBS 磁碟區仍具有與初始設定相同的組態：儲存類型、大小、佈建 IOPS 和儲存輸送量。 沒有額外 EBS 磁碟區連接至資料庫執行個體。 <p>使用下列 CLI 命令來比較 EBS 磁碟區詳細資料的磁碟區類型和適用於 Oracle 資料庫執行個體的 RDS 自訂詳細資訊：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier db-instance-name grep StorageType</pre>
SP-O1004	Amazon EC2 執行個體狀態	<p>自動復原使 EC2 執行個體 [<i>ec2_id</i>] 處於受損狀態。若要解決問題，請參閱疑難排解執行個體復原失敗。</p>	<p>若要檢查資料庫執行個體的狀態，請使用主控台或執行下列 AWS CLI 命令：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep DBInstanceStatus</pre>

事件 ID	組態	RDS 事件訊息	動作
SP-O1005	Amazon EC2 執行個體屬性	<p><i>EC2 ##</i> <i>[ec2_id] ####</i> <i>#### [att1] #</i> <i>[val-old] ##</i> <i># [VAL-#]###</i> <i>[att2] # [val-old] ### [VAL-#]#</i></p> <p>若要解決此問題，請回復為原始值。</p>	
SP-O1006	Amazon EC2 執行個體狀態	<p>EC2 執行個體 <i>[ec2_id]</i> 已終止或找不到。若要解決此問題，請刪除 RDS 自訂資料庫執行個體。</p>	<p>支援周邊會監控 EC2 執行個體狀態變更通知。EC2 執行個體必須永遠在執行中。</p> <p>刪除資料庫執行個體</p> <ol style="list-style-type: none"> 若要檢查資料庫執行個體的狀態，請使用主控台或執行下列 AWS CLI 命令： <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep DBInstanceStatus</pre> <ol style="list-style-type: none"> 刪除適用於 Oracle 資料庫執行個體的 RDS 自訂。

事件 ID	組態	RDS 事件訊息	動作
SP-O1007	Amazon EC2 執行個體狀態	EC2 執行個體 <code>[ec2_id]</code> 已停止。若要解決此問題，請啟動執行個體。	<p>支援周邊會監控 EC2 執行個體狀態變更通知。EC2 執行個體必須永遠在執行中。</p> <p>重新啟動資料庫執行個體</p> <ol style="list-style-type: none">若要檢查資料庫執行個體的狀態，請使用主控台或執行下列 AWS CLI 命令： <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep DBInstanceStatus</pre> <ol style="list-style-type: none">啟動資料庫執行個體。重新掛載二進位和資料磁碟區。

作業系統

事件 ID	組態	RDS 事件訊息	動作
SP-O2001	RDS Custom 代理程式狀態	RDS 自訂代理程式未在 EC2 執行個體 <code>[ec2_id]</code> 上執行。請確定代理程式在 <code>[ec2_id]</code> 上執行。	<p>在 RDS Custom for Oracle 上，如果 RDS Custom 代理程式停止，則資料庫執行個體就會在支援周邊外面。代理程式會 CloudWatch 每隔 30 秒將 <code>IamAlive</code> 指標發佈到 Amazon。如果超過 30 秒未發佈指標，則會觸發警示。支援周邊也會每 30 分鐘監控主機上的 RDS Custom 代理程式程序狀態。</p> <p>重新啟動 RDS 自訂代理程式</p> <ol style="list-style-type: none"> 1. 登入主機並確定 RDS Custom 代理程式正常執行。 2. 執行下列命令以尋找代理程式的狀態。 <pre>service rdscustomagent status</pre> <ol style="list-style-type: none"> 3. 使用下列命令啟動代理程式。 <pre>service rdscustomagent start</pre> <p>當 RDS 自訂代理程式再次執行時，指 <code>IamAlive</code> 標會發佈至 Amazon CloudWatch，並且警示會切換至該 OK 狀態。此切換會通知支援周邊代理程式正在執行。</p>
SP-O2002	AWS Systems Manager 代理程式 (SSM 代理程式) 狀態	無法存取 EC2 執行個體 <code>[ec2_id]</code> 上的系統管理員代理程式。確認您已正確設定網路、代理程式和 IAM 許可。	<p>SSM 代理程式必須始終在執行中。RDS Custom 代理程式必須負責確定 Systems Manager 代理程式在執行中。如果 SSM 代理程式已終止然後重新啟動，則 RDS 自訂代理程式會將指標發佈至 CloudWatch。RDS Custom 代理程式具有指標警示，設定為在每次重新啟動的前三分鐘觸發。支援範圍也會每隔 30 分鐘監視主機上 SSM 代理程式的處理序狀態。</p> <p>如需詳細資訊，請參閱 針對 SSM Agent 進行疑難排解。</p>

事件 ID	組態	RDS 事件訊息	動作
SP-02003	AWS Systems Manager 代理程式 (SSM 代理程式) 狀態	EC2 實例 [<i>ec2_id</i>] 上的系統管理器代理程式崩潰了多次。如需詳細資訊，請參閱 SSM 代理程式疑難排解說 明文件。	如需詳細資訊，請參閱 針對 SSM Agent 進行疑難排解 。
SP-O2004	作業系統時區	已變更 EC2 執行個體 [<i>ec2_id</i>] 上的時區。若要解決此問題，請將時區還原為先前的 [<i>previous-time-zone</i>] 設定。然後使用 RDS 選項群組來變更時區。	<p>RDS 自動化偵測到主機上的時區已變更，而不使用選項群組。此主機層級變更可能會導致 RDS 自動化失敗，因此 EC2 執行個體會置於狀態。unsupported-configuration</p> <p>修正時區設定</p> <ol style="list-style-type: none"> 登入 EC2 主機並檢查作業系統時區，如下所示： <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;"> <pre>timedatectl</pre> </div> <ol style="list-style-type: none"> 暫停 RDS Custom 自動化。如需詳細資訊，請參閱 暫停和繼續 RDS Custom 資料庫執行個體。 停止資料庫執行個體。 還原作業系統上的時區變更。 啟動資料庫執行個體。 繼續 RDS Custom 自動化。 <p>資料庫執行個體會 在 30 分鐘內變成可用。若要避免 future 移出周邊，請透過選項群組修改您的時區。如需詳細資訊，請參閱 Oracle 時區。</p>

事件 ID	組態	RDS 事件訊息	動作
SP-O2005	sudo 組態	EC2 執行個體 <code>[ec2_id]</code> 上的 sudo 組態缺少必要的許可。若要解決此問題，請將最近的變更還原至 sudo 組態。	<p>支援周邊會監控某些作業系統使用者可以在方塊上執行特定命令。它會監控 sudo 組態與支援的狀態。</p> <p>當 sudo 組態不受支援時，RDS Custom 會嘗試將它們覆寫回先前支援的狀態。如果成功，則會傳送下列通知：</p> <p>RDS Custom 已成功覆寫您的組態。</p> <p>若要調查 sudo 組態的變更</p> <ol style="list-style-type: none"> 1. 登入您的主機。 2. 執行下列命令。 <div data-bbox="776 865 1507 982" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>visudo -c -f /etc/sudoers.d/ <i>individual_sudo_files</i></pre> </div> 3. 視需要修改 sudo 模型組態。 <p>在支援範圍判斷 sudo 組態是否受支援之後，您的 RDS 自訂適用於 Oracle 資料庫執行個體會 30 分鐘內提供使用。</p>
SP-O2006	S3 儲存貯體存取	RDS 自訂自動化無法從 EC2 執行個體 <code>[ec2_id]</code> 上的 S3 儲存貯體下載檔案。檢查您的聯網組態，並確定執行個體允許與 S3 之間的連線。	

資料庫

事件 ID	組態	RDS 事件訊息	動作
SP-O3001	資料庫封存延遲目標	<p><i>EC2 #####</i> <i>[ec2_id] #####</i> <i>_##_#####</i> <i>#####</i> 若要解決此問題，請將參數設定為值範圍內的值。</p>	<p>支援範圍會監控ARCHIVE_LAG_TARGET 資料庫參數，以確認資料庫執行個體的最新可還原時間是否在合理的範圍內。</p> <p>變更存檔重做日誌的延遲目標</p> <ol style="list-style-type: none"> 1. 登入您的 EC2 主機 2. Connect 到適用於 Oracle 資料庫執行個體的 RDS 自訂 3. 將ARCHIVE_LAG_TARGET 參數變更為介於 60—7200 之間的值。例如，使用下列 SQL 陳述式。 <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;</pre> </div> <p>資料庫執行個體會 在 30 分鐘內變成可用。</p>

事件 ID	組態	RDS 事件訊息	動作
SP-O3002	Oracle Data Guard 角色	在 EC2 執行個體 <code>[ec2_id] ## Oracle ##### [#####]</code> 。若要解決此問題，請將 DATABASE_ROLE 參數設定為主要或實體待命。	<p>支援周邊每 15 秒會監視目前的資料庫角色，並在資料庫角色變更時傳送 CloudWatch 通知。Oracle Data Guard DATABASE_ROLE 參數必須是 PRIMARY 或 PHYSICAL STANDBY。</p> <p>將 Oracle 資料保全資料庫角色還原為支援的值</p> <ol style="list-style-type: none"> 執行下列陳述式，以檢查「Oracle 資料保全」角色： <pre>SELECT DATABASE_ROLE FROM V\$DATABASE;</pre> <ol style="list-style-type: none"> 如果您的資料庫執行個體是獨立的，請使用下列其中一個陳述式將其變更回 PRIMARY 角色： <pre>ALTER DATABASE COMMIT TO SWITCHOVER PRIMARY; ALTER DATABASE ACTIVATE STANDBY DATABASE;</pre> <p>如果您的資料庫執行個體是複本，請使用下列陳述式將其變更回 PHYSICAL STANDBY 角色：</p> <pre>ALTER DATABASE CONVERT TO PHYSICAL STANDBY;</pre> <p>在支援周邊判斷支援資料庫角色之後，RDS Custom for Oracle 資料庫執行個體會 在 15 秒內變成可用。</p>

事件 ID	組態	RDS 事件訊息	動作
SP-O3003	資料庫運作狀態	甲骨文數據庫的 SMON 進程處於殭屍狀態。若要解決此問題，請手動復原 EC2 執行個體 [ec2_id] 上的資料庫，開啟資料庫，然後立即備份。如需更多協助，請聯絡 AWS Support。	<p>支援周邊會監控資料庫執行個體狀態。它也會監控前一小時和前一天發生的重新啟動次數。當執行個體處於仍然存在的狀態時，您會收到通知，但無法與其互動。</p> <p>使支撐周邊評估您的執行個體狀態</p> <ol style="list-style-type: none"> 登入主機並判斷資料庫狀態。 <pre>ps -eo pid,state,command grep smon</pre> <ol style="list-style-type: none"> 必要時，請重新啟動資料庫執行個體。如果重新啟動失敗，請繼續執行下一個步驟。 如有必要，請重新啟動 EC2 主機。 <p>資料庫執行個體重新啟動後，RDS Custom 代理程式會偵測到您的資料庫執行個體不再處於無回應狀態。然後，它會通知支援周邊重新評估您的資料庫執行個體狀態。</p>

事件 ID	組態	RDS 事件訊息	動作
SP-O3004	資料庫日誌模式	<p><i>EC2 #####</i> <i>[ec2_id] ###</i> <i>#####</i> <i>[#]#</i>若要解決此問題，請將記錄模式設定為 [<i>value_a</i>]。</p>	<p>將資料庫執行個體記錄模式變更為 ARCHIVELOG</p> <ol style="list-style-type: none"> 登入您的 EC2 主機。 Connect 到資料庫並執行下列陳述式： <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 5px 0;"> <pre>SELECT LOG_MODE FROM V\$DATABASE;</pre> </div> <p>或者您可以在 SQL*Plus 中運行以下命令：</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 5px 0;"> <pre>ARCHIVE LOG LIST</pre> </div> 執行下列 SQL*Plus 命令以起始一致的關閉。 <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 5px 0;"> <pre>SHUTDOWN IMMEDIATE</pre> </div> <p>RDS 自訂代理程式會自動重新啟動資料庫執行個體，並將記錄模式設定為ARCHIVELOG。資料庫執行個體會在 30 分鐘內變成可用。</p>
SP-O3005	Oracle 本位目錄路徑	<p><i>EC2 #####</i> <i>[ec2_id] ###</i> <i>#####</i> <i>###</i>若要解決此問題，請將設定還原為 <i>old_path</i>。</p>	

事件 ID	組態	RDS 事件訊息	動作
SP-O3006	資料庫唯一名稱	<p><i>EC2 ##</i> <i>[ec2_id] ###</i> <i>#####</i> <i>##</i>若要解決此問題，請將名稱還原為 <i>old_value</i> 。</p>	<p>變更資料庫執行個體的資料庫唯一名稱</p> <ol style="list-style-type: none"> 登入您的 EC2 主機。 Connect 至資料庫並執行下列陳述式： <pre>SELECT DB_UNIQUE_NAME FROM V\$DATABASE;</pre> <ol style="list-style-type: none"> 使用指令指定原始資料庫唯一名稱 ALTER SYSTEM SET DB_UNIQUE_NAME 。 執行下列 SQL 陳述式以起始一致性關閉。 <pre>SHUTDOWN IMMEDIATE;</pre> <p>RDS 自訂代理程式會自動重新啟動資料庫執行個體，並將記錄模式設定為 ARCHIVELOG 。資料庫執行個體會在 30 分鐘內變成可用。</p>

針對 RDS Custom for Oracle 升級進行疑難排解

您的 RDS Custom for Oracle 執行個體升級可能會失敗。接著，您可以找到在升級 RDS Custom DB for Oracle 資料庫執行個體期間可以使用的技巧：

- 檢查資料庫執行個體 /tmp 目錄中的升級輸出日誌檔。日誌名稱取決於您的資料庫引擎版本。例如，您可能會看到包含字串 catupgrd 或 catup 的日誌。
- 檢查位於 /rdsdbdata/log/trace 目錄的 alert.log 檔案。
- 在 root 目錄中執行下列 grep 命令以追蹤升級作業系統程序。此命令會顯示寫入日誌檔的位置，並判斷升級程序的狀態。

```
ps -aux | grep upg
```

下面顯示了範例輸出。

```

root      18884  0.0  0.0 235428  8172 ?          S<   17:03   0:00 /usr/bin/
sudo -u rdsdb /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-
UPGRADE/2.upgrade.sh
rdsdb     18886  0.0  0.0 153968 12164 ?          S<   17:03   0:00 /usr/bin/perl -T -
w /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-UPGRADE/2.upgrade.sh
rdsdb     18887  0.0  0.0 113196  3032 ?          S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb     18900  0.0  0.0 113196  1812 ?          S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb     18901  0.1  0.0 167652 20620 ?          S<   17:03   0:07 /rdsdbbin/oracle/
perl/bin/perl catctl.pl -n 4 -d /rdsdbbin/oracle/rdbms/admin -l /tmp catupgrd.sql
root      29944  0.0  0.0 112724  2316 pts/0      S+   18:43   0:00 grep --color=auto
upg

```

- 執行下列 SQL 查詢以驗證元件的目前狀態，以尋找資料庫版本和資料庫執行個體上安裝的選項。

```

SET LINESIZE 180
COLUMN COMP_ID FORMAT A15
COLUMN COMP_NAME FORMAT A40 TRUNC
COLUMN STATUS FORMAT A15 TRUNC
SELECT COMP_ID, COMP_NAME, VERSION, STATUS FROM DBA_REGISTRY ORDER BY 1;

```

輸出結果與以下內容相似。

COMP_NAME	STATUS	PROCEDURE
Oracle Database Catalog Views	VALID	
DBMS_REGISTRY_SYS.VALIDATE_CATALOG		
Oracle Database Packages and Types	VALID	
DBMS_REGISTRY_SYS.VALIDATE_CATPROC		
Oracle Text	VALID	VALIDATE_CONTEXT
Oracle XML Database	VALID	DBMS_REGXDB.VALIDATEXDB

4 rows selected.

- 執行下列 SQL 查詢，以檢查是否有可能會干擾升級程序的無效物件。

```

SET PAGES 1000 LINES 2000
COL OBJECT FOR A40
SELECT SUBSTR(OWNER,1,12) OWNER,

```

```
SUBSTR(OBJECT_NAME,1,30) OBJECT,  
SUBSTR(OBJECT_TYPE,1,30) TYPE, STATUS,  
CREATED  
FROM DBA_OBJECTS  
WHERE STATUS <> 'VALID'  
AND OWNER IN ('SYS', 'SYSTEM', 'RDSADMIN', 'XDB');
```

RDS Custom for Oracle 複本提升的疑難排解

您可以使用主控台、`promote-read-replica` AWS CLI 命令或 `PromoteReadReplica` API，在 RDS 自訂適用於 Oracle 中升級受管理的 Oracle 複本。如果您刪除主要資料庫執行個體，且所有複本都正常運作，則 RDS Custom for Oracle 會自動將受管複本提升為獨立執行個體。如果複本已暫停自動化或位於支援範圍之外，您必須先修正複本，RDS Custom 才能自動提升複本。如需詳細資訊，請參閱 [RDS Custom for Oracle 的複本提升限制](#)。

複本提升工作流程可能會下列情況中停滯：

- 主要資料庫執行個體處於狀態 `STORAGE_FULL`。
- 主要資料庫無法封存其所有線上重做日誌。
- 您的 Oracle 複本中的封存重做日誌檔案與主要資料庫之間存在差距。

若要回應卡住的工作流程

1. 在 Oracle 複本資料庫執行個體上同步重做日誌差距。
2. 強制將僅供讀取複本提升至最新套用的重做日誌。執行下列 SQL*Plus 命令：

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;  
SHUTDOWN IMMEDIATE  
STARTUP
```

3. 聯絡 AWS Support 並要求他們將您的資料庫執行個體移至 `available` 狀態。

使用 RDS Custom for SQL Server

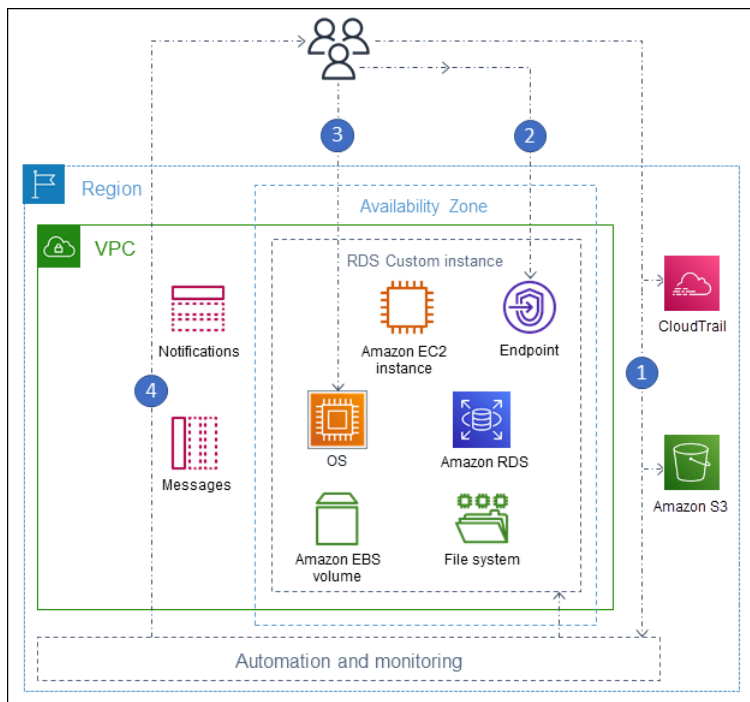
在下列內容，您可以找到建立、管理和維護 RDS Custom for SQL Server 資料庫執行個體的指示。

主題

- [RDS Custom for SQL Server 工作流程](#)
- [Amazon RDS Custom for SQL Server 的需求和限制](#)
- [設定您的 Amazon RDS Custom for SQL Server 的環境](#)
- [透過 RDS Custom for SQL Server 使用自有媒體](#)
- [使用 RDS Custom for SQL Server 的自訂引擎版本](#)
- [建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體](#)
- [管理 Amazon RDS Custom for SQL Server 資料庫執行個體](#)
- [管理 RDS Custom for SQL Server 的多可用區部署](#)
- [備份與還原 Amazon RDS Custom for SQL Server 資料庫執行個體](#)
- [將內部部署資料庫遷移到 Amazon RDS Custom for SQL Server](#)
- [升級 Amazon RDS Custom for SQL Server 的資料庫執行個體](#)
- [針對 Amazon RDS Custom for SQL Server 的資料庫問題進行故障診斷](#)

RDS Custom for SQL Server 工作流程

下圖顯示 RDS Custom for SQL Server 的典型工作流程。



步驟如下：

1. 從 RDS Custom 提供的引擎版本建立 RDS Custom for SQL Server 資料庫執行個體。

如需詳細資訊，請參閱 [建立 RDS Custom for SQL Server 資料庫執行個體](#)。

2. 將應用程式連線到 RDS Custom 資料庫執行個體端點。

如需更多詳細資訊，請參閱 [使用連接到 RDS 自訂資料庫執行個體 AWS Systems Manager](#) 及 [使用 RDS 連線到您的 RDS Custom 資料庫執行個體](#)。

3. (選用) 存取主機以自訂您的軟體。
4. 監控 RDS Custom 自動化產生的通知和訊息。

建立 RDS Custom 資料庫執行個體

您可以使用 `create-db-instance` 命令建立 RDS Custom 資料庫執行個體。該程序類似於建立 Amazon RDS 執行個體的程序。但是，某些參數不同。如需詳細資訊，請參閱 [建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體](#)。

資料庫連線

如同 Amazon RDS 資料庫執行個體，您的 RDS Custom for SQL Server 資料庫執行個體位於 VPC 中。您的應用程式會使用用戶端 (例如 SQL Server Management Studio (SSMS)) 連線至 RDS Custom 執行個體，就像在 RDS for SQL Server 一樣。

RDS Custom 自訂

您可以存取 RDS Custom 主機以安裝或自訂軟體。若要避免變更與 RDS Custom 自動化之間發生衝突，您可以在指定期間暫停自動化。在此期間，RDS Custom 不會執行監控或執行個體復原。在期間結束時，RDS Custom 會繼續全自動化。如需詳細資訊，請參閱 [暫停和繼續 RDS Custom 自動化](#)。

Amazon RDS Custom for SQL Server 的需求和限制

您可在下方找到 Amazon RDS Custom for SQL Server 需求和限制的摘要，以進行快速參考。需求和限制也會出現在相關章節中。

主題

- [區域和版本可用性](#)
- [RDS Custom for SQL Server 的一般需求](#)
- [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)
- [RDS Custom for SQL Server 的限制](#)
- [RDS Custom for SQL Server 資料庫執行個體的定序和字元支援](#)
- [RDS Custom for SQL Server 資料庫執行個體的本機時區](#)
- [將服務主要金鑰與 SQL 伺服器的 RDS 自訂搭配使用](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 Amazon RDS 搭配 Amazon RDS Custom for SQL Server 版本和區域可用性的詳細資訊，請參閱 [適用於 SQL 伺服器的 RDS 自訂支援的區域和資料庫引擎](#)。

RDS Custom for SQL Server 的一般需求

請確保遵循 Amazon RDS Custom for SQL Server 的以下需求：

- 使用 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#) 中顯示的執行個體類別。支持的唯一存儲類型是 gp2，gp3，io1 和 io2 塊快速類型的固態驅動器 (SSD)。儲存限制上限為 16 TiB。
- 請確定您擁有可建立 RDS 自訂資料庫執行個體的對稱加密 AWS KMS 金鑰。如需詳細資訊，請參閱 [確定您擁有對稱式加密金鑰 AWS KMS](#)。
- 請務必建立 AWS Identity and Access Management (IAM) 角色和執行個體設定檔。如需詳細資訊，請參閱 [手動建立 IAM 角色和執行個體設定檔](#) 及 [自動建立執行個體設定檔，使 AWS Management Console](#)。
- 請確定提供 RDS 自訂可用來存取其他網路組態 AWS 服務。如需特定需求，請參閱 [步驟 2：設定網路、執行個體設定檔和加密](#)。
- RDS Custom 和 Amazon RDS 資料庫執行個體的合併數量不能超過您的配額限制。例如，如果您的配額為 40 個資料庫執行個體，則可以擁有 20 個 RDS Custom for SQL Server 資料庫執行個體和 20 個 Amazon RDS 資料庫執行個體。

- RDS 自訂會自動建立名稱開頭為的 AWS CloudTrail 追蹤do-not-delete-rds-custom-。RDS 自訂支援範圍會依賴事件 CloudTrail 來判斷您的動作是否會影響 RDS 自訂自動化。RDS Custom 在您建立第一個資料庫執行個體時建立追蹤。若要使用已存在的 CloudTrail，請聯絡 Sup AWS port 部門。如需詳細資訊，請參閱 [AWS CloudTrail](#)。

RDS Custom for SQL Server 的資料庫執行個體類別支援

使用 [describe-orderable-db-instance-](#) options 命令檢查您的區域是否支援資料庫執行個體類別。

適用於 SQL Server 的 RDS 自訂支援下表所示的資料庫執行個體類別：

SQL Server 版本	RDS Custom 支援
企業版本	db.r5.xlarge–db.r5.24xlarge 大型資料庫 .r5b db.m5.xlarge–db.m5.24xlarge 大型資料庫 .r6. 大型資料庫 .r6 .32 數據庫 .m6. X 大 — 數據庫. m6. db.x2iedn.xlarge–db.x2iedn.32xlarge
標準版本	db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.8xlarge db.m5.large–db.m5.24xlarge 大型資料庫 .r6 l.8 DB.m6. 大 — 數據庫. m6 l.8-大 資料庫 .x2. 擴大 — 資料庫 .x2 內建 .8 倍大
開發人員版本	db.r5.xlarge–db.r5.24xlarge

SQL Server 版本	RDS Custom 支援
	大型資料庫 .r5b db.m5.xlarge–db.m5.24xlarge 大型資料庫 .r6. 大型資料庫 .r6 .32 數據庫 .m6. X 大 — 數據庫. m6. db.x2iedn.xlarge–db.x2iedn.32xlarge
Web 版本	db.r5.large–db.r5.4xlarge db.m5.large–db.m5.4xlarge db.r6i.large–db.r6i.4xlarge DB.m6. 大 — 數據庫. m6 l.4X 大 db.r5b.large–db.r5b.4xlarge

下列建議適用於 db.x2iedn 類別類型：

- 在創建時，本地存儲是一個原始和未分配的設備。在搭配此執行個體類別使用資料庫執行個體之前，您必須掛載本機儲存並格式化。之後，tempdb在其上進行配置以確保最佳性能。[如需詳細資訊，請參閱使用本機執行個體儲存體在適用於 SQL Server 的 Amazon RDS 自訂中優化臨時資料庫效能。](#)
- 當您執行資料庫執行個體操作 (例如擴展運算、執行個體取代、快照還原或 point-in-time復原 (PITR) 時，本機儲存體會還原為其原始和未配置狀態。在這些情況下，您必須重新掛載、重新格式化和重新設定磁碟機，tempdb以及還原功能。
- 對於異地同步備份執行個體，建議您在待命資料庫執行個體上執行組態。如此一來，如果發生容錯移轉，系統會繼續運作而不會出現問題，因為組態已在待命執行個體上就位。

RDS Custom for SQL Server 的限制

下列限制適用於 RDS Custom for SQL Server：

- 您無法在 Amazon RDS for RDS Custom for SQL Server 資料庫執行個體中建立僅供讀取複本。不過，您可以搭配多可用區部署自動設定高可用性。如需詳細資訊，請參閱 [管理 RDS Custom for SQL Server 的多可用區部署](#)。
- 您無法修改現有適用於 SQL Server 資料庫執行個體的 RDS 自訂資料庫執行個體識別碼。
- 對於非使用自訂引擎版本 (CEV) 建立的適用於 SQL Server 資料庫執行個體的 RDS 自訂，不保證對 Microsoft 視窗作業系統的變更會持續存在。例如，當您初始化快照或 point-in-time 還原作業時，會遺失這些變更。如果 RDS Custom for SQL Server 資料庫執行個體是使用 CEV 建立的，則系統會保存這些變更。
- 並非所有選項都支援。例如，當您建立 RDS Custom for SQL Server 資料庫執行個體時，您無法執行下列操作：
 - 變更資料庫執行個體類別中每個核心的 CPU 核心數量和每個核心的執行緒數量。
 - 開啟儲存體自動擴展。
 - 使用 AWS Management Console 設定 Kerberos 身分驗證。不過，您可以手動設定 Windows 身分驗證並且使用 Kerberos。
 - 指定您自己的資料庫參數群組、選項群組或字元集。
 - 開啟績效詳情。
 - 開啟自動次要版本升級。
- 資料庫執行個體儲存體上限為 16 TiB。

RDS Custom for SQL Server 資料庫執行個體的定序和字元支援

RDS Custom for SQL Server 可針對 SQL_Latin、Japanese、German 和 Arabic 地區設定，支援使用傳統和 UTF-8 編碼的各種不同伺服器定序。預設的伺服器定序為 SQL_Latin1_General_CP1_CI_AS，不過，您可以選取其他支援的定序來使用。您可以選取與 RDS for SQL Server 使用相同程序的定序。如需詳細資訊，請參閱 [Microsoft SQL Server 的定序和字元集](#)。

在 RDS Custom for SQL Server 上使用伺服器定序時，須遵循下列需求和限制：

- 您可以在建立 RDS Custom for SQL Server 資料庫執行個體時設定伺服器定序。資料庫執行個體建立之後，您就無法修改伺服器層級定序。
- 從資料庫快照還原時或在時間點復原 (PITR) 期間，您無法修改伺服器層級定序。

- 當您從 RDS Custom for SQL Server CEV 建立資料庫執行個體時，該資料庫執行個體不會從 CEV 繼承伺服器定序，而是會使用預設伺服器定序 SQL_Latin1_General_CP1_CI_AS。如果您已在 RDS Custom for SQL Server CEV 上設定非預設伺服器定序，並且想要在新的資料庫執行個體上使用相同的伺服器定序，則從 CEV 建立資料庫執行個體時，請務必選取相同的定序。

Note

如果您在建立資料庫執行個體時選取的定序與 CEV 的定序不同，則會在新的 RDS Custom for SQL Server 資料庫執行個體上重建 Microsoft SQL Server 系統資料庫，以使用更新的定序。重建程序只會在新的 RDS Custom for SQL Server 資料庫執行個體上執行，而且不會對 CEV 本身造成任何影響。一旦重建系統資料庫，您先前對 CEV 上的系統資料庫所做的任何修改都不會保留在新的 RDS Custom for SQL Server 資料庫執行個體上。一些修改的範例包括：master 資料庫中的使用者定義物件、msdb 資料庫中的已排程工作，或是對 CEV 上 model 資料庫中的預設資料庫設定所做的變更。您可以在新的 RDS Custom for SQL Server 資料庫執行個體建立之後，手動重新建立修改。

- 當您從 RDS Custom for SQL Server 自訂引擎版本 (CEV) 建立資料庫執行個體，並選取與 CEV 不同的定序時，請確定用於建立 CEV 的黃金映像 (AMI) 符合下列需求，如此才能在新的資料庫執行個體上重建 Microsoft SQL Server 系統資料庫：
 - 對於 SQL 伺服器 2022，請確定 setup.exe 檔案位於下列路徑中：C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap\SQL2022\setup.exe
 - 若是 SQL Server 2019，請確定 setup.exe 檔案位於下列路徑中：C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap\SQL2019\setup.exe
 - master、model 和 msdb 資料庫的資料和日誌範本副本必須存在各自的預設位置。如需詳細資訊，請參閱 Microsoft 公開文件中的 [重建系統資料庫](#)。
 - 請確定您的 SQL Server Database Engine 使用 NT Service\MSSQLSERVER 或 NT AUTHORITY\NETWORK SERVICE 作為服務帳戶。為資料庫執行個體設定非預設伺服器定序時，任何其他帳戶都不會擁有 C:\ 磁碟機上的必要許可。
- 如果為新資料庫執行個體選取的伺服器定序與 CEV 上的設定相同，則新的 RDS Custom for SQL Server 資料庫執行個體上的 Microsoft SQL Server 系統資料庫將不會進行重建程序。您先前對 CEV 上系統資料庫所做的任何修改，都會自動保留在新的 RDS Custom for SQL Server 資料庫執行個體中。

您可以將定序設為下表中列出的其中一個值。

伺服器定序	描述
阿拉伯文	阿拉伯語 -100，二進制排序
阿拉伯文	阿拉伯語 -100、二進制碼點比較排序
阿拉伯人	阿拉伯語 -100，不區分大小寫，重音不敏感，不敏感，寬
阿拉伯文 _ 西阿拉伯	阿拉伯語 -100，不區分大小寫，重音不敏感，角色敏感，
阿拉伯文阿拉伯文	阿拉伯語 -100，不區分大小寫，不區分重音，角色敏感，
阿拉伯文阿拉伯文	阿拉伯語 -100，不區分大小寫，不區分重音，角色敏感，
阿拉伯文 _ 西班牙	阿拉伯語 -100，不區分大小寫，重音不敏感，角色敏感，
阿拉伯文 _ 中國人民共和國	阿拉伯語 -100，不區分大小寫，不區分重音，角色敏感，
阿拉伯文 _ 西班牙文 _ 西班牙文	阿拉伯語 -100，不區分大小寫，不區分重音，角色敏感，
阿拉伯文	阿拉伯語 -100，不區分大小寫，不區分重音，不敏感角色
阿拉伯文阿拉伯文	阿拉伯語 -100，不區分大小寫，不區分重音，不敏感角色
阿拉伯文 _ 西阿拉伯	阿拉伯語 -100，不區分大小寫，重音不敏感，不敏感，寬
阿拉伯文 _ 中國人民共和國	阿拉伯語 -100，不區分大小寫，不區分重音，不敏感角色
阿拉伯文 _ 西阿拉伯文 _ 英文	阿拉伯語 -100，不區分大小寫，不區分重音，不敏感角色
阿拉伯文	阿拉伯語 -100，不區分大小寫，重音敏感，不敏感角色，
阿拉伯文	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，寬
阿拉伯阿拉伯	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，不
阿拉伯文阿拉伯文	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，寬
阿拉伯文	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，寬
阿拉伯語	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，寬
阿拉伯文 _ 阿拉伯文 _ 西班牙文	阿拉伯語 -100，不區分大小寫，重音敏感，角色敏感，寬

阿拉伯	阿拉伯語 -100, 不區分大小寫, 重音敏感, 不敏感角色,
阿拉伯文阿拉伯文	阿拉伯語 -100, 不區分大小寫, 重音敏感, 不敏感角色,
阿拉伯文	阿拉伯語 -100, 不區分大小寫, 重音敏感, 不敏感, 寬度
阿拉伯語	阿拉伯語 -100, 不區分大小寫, 重音敏感, 不敏感角色,
阿拉伯文_西阿拉伯文	阿拉伯語 -100, 不區分大小寫, 重音敏感, 不敏感角色,
阿拉伯人_人工智能	阿拉伯語 -100, 區分大小寫, 重音不敏感, 不敏感, 寬度
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音不敏感, 角色敏感, 寬
阿拉伯文_阿拉伯文	阿拉伯文 -100、區分大小寫、不區分重音、角色敏感、不
阿拉伯文阿拉伯文	阿拉伯語 -100, 區分大小寫, 不區分重音, 角色敏感, 寬
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音不敏感, 角色敏感, 寬
阿拉伯文_爱加利亚国际	阿拉伯文 -100、區分大小寫、不區分重音、角色敏感、寬
阿拉伯文_ 中华人民共和国	阿拉伯語 -100, 區分大小寫, 不區分重音, 角色敏感, 寬
阿拉伯文_100 阿拉伯	阿拉伯文 -100、區分大小寫、不區分重音、不敏感角色、
阿拉伯文阿拉伯文	阿拉伯語 -100, 區分大小寫, 不區分重音, 不敏感角色,
阿拉伯文_100 名艾西	阿拉伯語 -100, 區分大小寫, 重音不敏感, 不敏感, 寬度
阿拉伯文_ 法国人民共和国	阿拉伯語 -100, 區分大小寫, 不區分重音, 不敏感角色,
阿拉伯文_ 中华人民共和国	阿拉伯語 -100, 區分大小寫, 不區分重音, 不敏感角色,
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 不敏感, 寬度不
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 角色敏感, 不區
阿拉伯文_阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 角色敏感, 不區
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度

阿拉伯文	阿拉伯文 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度
阿拉伯文阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度
阿拉伯文	阿拉伯文 -100、區分大小寫、重音區分、不敏感角色、不
阿拉伯文_阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 不敏感卡納特,
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 不敏感, 寬度敏
阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 不敏感角色, 寬
阿拉伯文_阿拉伯文	阿拉伯語 -100, 區分大小寫, 重音敏感, 不敏感卡納特,
阿拉伯文	阿拉伯語, 二進制排
阿拉伯文	阿拉伯語, 二進制碼點比較排序
阿拉伯語	阿拉伯語, 不區分大小寫, 不區分重音, 不敏感角色, 寬度
阿拉伯文阿拉伯文	阿拉伯語, 不區分大小寫, 不區分重音, 角色敏感, 寬度不
阿拉伯西阿拉伯文	阿拉伯語, 不區分大小寫, 不區分重音, 角色敏感, 寬度每
阿拉伯文西阿拉伯文	阿拉伯語, 不區分大小寫, 不區分重音, 不敏感角色, 寬度
Arabic_CI_AS	阿拉伯文、不區分大小寫、區分重音、不區分假名、不區分
阿拉伯文	阿拉伯語, 不區分大小寫, 重音敏感, 角色敏感, 寬度不每
阿拉伯文西阿拉伯	阿拉伯語, 不區分大小寫, 重音敏感, 角色敏感, 寬度敏
阿拉伯文西阿拉伯	阿拉伯語, 不區分大小寫, 重音敏感, 不敏感角色, 寬度每
阿拉伯人工智能	阿拉伯文, 區分大小寫, 不區分重音, 不敏感角色, 寬度不
阿拉伯文阿拉伯文	阿拉伯文, 區分大小寫, 不區分重音, 角色敏感, 寬度不每
阿拉伯文社会科技	阿拉伯文, 區分大小寫, 不區分重音, 角色敏感, 寬度敏
阿拉伯文 CS_AI_WS	阿拉伯文, 區分大小寫, 不區分重音, 不敏感角色, 寬度每
Arabic_CS_AS	阿拉伯文, 區分大小寫, 重音敏感, 不敏感角色, 寬度不每

阿拉伯文	阿拉伯文，區分大小寫，重音敏感，角色敏感，寬度不敏感
阿拉伯文	阿拉伯文，區分大小寫，重音敏感，角色敏感，寬度敏感
阿拉伯文	阿拉伯文，區分大小寫，重音敏感，不敏感角色，寬度敏感
Chinese_PRC_BIN2	中华人民共和国二进制码点比较排序
Chinese_PRC_CI_AS	Chinese-PRC、不區分大小寫、區分重音、不區分假名、不區分寬度
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke、不區分大小寫、區分重音、不區分假名、不區分寬度
Danish_Norwegian_CI_AS	Danish-Norwegian、不區分大小寫、區分重音、不區分假名、不區分寬度
Finnish_Swedish_CI_AS	芬蘭語-瑞典語，不區分大小寫，重音敏感，不敏感角色，不區分寬度
French_CI_AS	法文、不區分大小寫、區分重音、不區分假名、不區分寬度
德國 _ PhoneBook	德語 PhoneBook -100，二進制排序
德國 _ PhoneBook	德文 PhoneBook -100，二進制碼點比較排序
德國 _ 西安 _ 艾 PhoneBook	德語 PhoneBook -100，不區分大小寫，重音不敏感，不區分寬度
德國 _ 俄羅斯 _ 愛 _ K PhoneBook	德語 PhoneBook -100，不區分大小寫，重音不敏感，角色敏感，不區分寬度
德國 _ PhoneBook 中國愛加州	德文 PhoneBook -100、不區分大小寫、不區分重音、角色敏感，不區分寬度
德國 _ 西班牙 _ 中國 _ 英文 PhoneBook	德文 PhoneBook -100，不區分大小寫，不區分重音，角色敏感，不區分寬度
德國 _ PhoneBook 中國愛西蘭	德語 PhoneBook -100，不區分大小寫，重音不敏感，角色敏感，不區分寬度
德國 _ PhoneBook 中國愛加州	德文 PhoneBook -100，不區分大小寫，不區分重音，角色敏感，不區分寬度
德國 _ 西班牙 _ 西班牙語 _ 英語 PhoneBook	德文 PhoneBook -100，不區分大小寫，不區分重音，角色敏感，不區分寬度
德國 _ PhoneBook 中國	德語 PhoneBook -100，不區分大小寫，不區分重音，不區分寬度
德國 _ 西班牙 _ 英文 PhoneBook	德語 PhoneBook -100，不區分大小寫，不區分重音，不區分寬度
德國 _ PhoneBook 中國愛新聞	德語 PhoneBook -100，不區分大小寫，不區分重音，不區分寬度
德國 _ 中國愛華盛頓 PhoneBook	德語 PhoneBook -100，不區分大小寫，不區分重音，不區分寬度

德國 _ 西班牙 _ 英語 _ 英語 PhoneBook	德文 PhoneBook -100, 不區分大小寫, 不區分重音, 不敏感
德國 _ PhoneBook 西亞	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ 西亞 _ PhoneBook	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ 阿PhoneBook斯加州	德文 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ PhoneBook 西阿斯加州	德文 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ PhoneBook 中國	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ 中國 _ 華盛頓 PhoneBook	德文 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ 中國 _ 西班牙語 _ 英語 PhoneBook	德文 PhoneBook -100, 不區分大小寫, 重音敏感, 角色每
德國 _ 阿PhoneBook斯加	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ 西亞 _ 英文 PhoneBook	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ PhoneBook 中國	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ 中國 _ 華盛頓 PhoneBook	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ 俄羅斯 _ PhoneBook 西班牙文	德語 PhoneBook -100, 不區分大小寫, 重音敏感, 不敏感
德國 _ 人PhoneBook工智能	德語 PhoneBook -100, 區分大小寫, 重音不敏感, 不敏感
德國 _ _ 俄羅斯 _ 艾克PhoneBook斯	德語 PhoneBook -100, 區分大小寫, 重音不敏感, 角色每
德國 _ 阿拉伯PhoneBook聯合大學	德文 PhoneBook -100、區分大小寫、不區分重音、角色每
德國 _ _ PhoneBook 中國人民幣	德文 PhoneBook -100, 區分大小寫, 不區分重音, 角色每
德國 _ 愛PhoneBook加州	德語 PhoneBook -100, 區分大小寫, 重音不敏感, 角色每
德國 _ _ 艾克斯 _ 華盛頓 PhoneBook	德文 PhoneBook -100, 區分大小寫, 不區分重音, 角色每
德國 _ _ 英國 _ 英語 _W _ 英文 PhoneBook	德文 PhoneBook -100, 區分大小寫, 不區分重音, 角色每
德國 _ 俄羅斯 _ PhoneBook 中國	德文 PhoneBook -100, 區分大小寫, 不區分重音, 不敏感
德國 _ 俄羅斯 _ 英文 PhoneBook	德文 PhoneBook -100, 區分大小寫, 不區分重音, 不敏感

德國 _ 社會責任 _ 愛PhoneBook新聞	德語 PhoneBook -100 , 區分大小寫 , 重音不敏感 , 不敏感角
德國 __ 中國人民共PhoneBook和國	德文 PhoneBook -100 , 區分大小寫 , 不區分重音 , 不敏感角
德國 __ 艾西 _ 英文 PhoneBook	德文 PhoneBook -100 , 區分大小寫 , 不區分重音 , 不敏感角
德國 __ PhoneBook	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 不敏感角
德國 _____ PhoneBook	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德國 __ 阿PhoneBook斯加州	德文 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德文 _____ PhoneBook	德文 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德國 _ PhoneBook 新聞	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德國 __ 阿PhoneBook斯加州	德文 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德文 _____ 英文 PhoneBook	德文 PhoneBook -100 , 區分大小寫 , 重音敏感 , 角色敏感
德國 __ 阿PhoneBook斯加	德文 PhoneBook -100、區分大小寫、重音區分、不敏感角
德文 _____ PhoneBook	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 不敏感角
德國 _ PhoneBook 新聞	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 不敏感角
德國 __ 俄羅斯 _ 華夏 PhoneBook	德文 PhoneBook -100 , 區分大小寫 , 重音敏感 , 不敏感角
德文 _____ PhoneBook	德語 PhoneBook -100 , 區分大小寫 , 重音敏感 , 不敏感角
德國 _ 賓 PhoneBook	德語PhoneBook , 二進制排序
德國 _ PhoneBook	德文PhoneBook、二進制碼點比較排序
德國 _ PhoneBook 中國	德語 , 不區分大小寫PhoneBook , 不區分重音 , 不敏感角
德國 _ 西班牙 _ 愛 _ K PhoneBook	德語 , 不區分大小寫PhoneBook , 不區分重音 , 角色敏感
德國 _ 中國 _ 愛PhoneBook加州	德文 , 不區分大小寫PhoneBook , 不區分重音 , 角色敏感
德國 _ 中國 _ 愛PhoneBook新聞	德語 , 不區分大小寫PhoneBook , 不區分重音 , 不敏感角
德國 _ PhoneBook 西阿聯酋	德語 , 不區分大小寫PhoneBook , 重音敏感 , 不敏感角色

德國 _ 阿斯加州 _ PhoneBook	德文，不區分大小寫PhoneBook，重音敏感，角色敏感，
德國 _ 中國 _ PhoneBook 西班牙	德文，不區分大小寫PhoneBook，重音敏感，角色敏感，
德國 _ PhoneBook 中國	德文，不區分大小寫PhoneBook，重音敏感，不敏感角色
德國 _ 人PhoneBook工智能	德語PhoneBook，區分大小寫，不區分重音，不敏感角色
德國 _ 社會責任 _ 艾克PhoneBook斯	德文PhoneBook，區分大小寫，不區分重音，角色敏感，
德國 _ 社會責任 _ 愛滋病 PhoneBook	德文PhoneBook，區分大小寫，不區分重音，角色敏感，
德國 _ 社會責任 _ 愛PhoneBook新聞	德文PhoneBook，區分大小寫，不區分重音，不敏感角色
德文 _ PhoneBook	德文PhoneBook，區分大小寫，重音敏感，不敏感角色，
德國 _ _ _ _ _ PhoneBook	德文PhoneBook，區分大小寫，重音敏感，角色敏感，寬
德國 _ 社PhoneBook會責任	德文PhoneBook，區分大小寫，重音敏感，角色敏感，寬
德國 _ 社PhoneBook會責任	德文PhoneBook，區分大小寫，重音敏感，不敏感角色，
Hebrew_BIN	希伯來文、二進位排序
Hebrew_CI_AS	希伯來文、不區分大小寫、區分重音、不區分假名、不區分
日本文	日文 90, 二進位排序
日本 _90	日文 -90、二進制碼點比較排序
日本人 _ 西班牙	日語 90，不區分大小寫，不區分重音，不敏感角色，寬度
日本人 _ 西班牙	日語 90，不區分大小寫，不區分重音，角色敏感，寬度不
日語 _ 西班牙語	日文 -90、不區分大小寫、不區分重音、角色敏感、不區分
日文 _ 西班牙文 _ 西班牙文	日文-90、不區分大小寫、不區分重音、角色敏感、不區分
日文 _ 西班牙文	日文 -90，不區分大小寫，不區分重音，角色敏感，寬度每
日文 _ 中国人民共和国	日文 -90、不區分大小寫、不區分重音、角色敏感、寬度每
日文 _ 西班牙文 _ 西班牙文	日文-90、不區分大小寫、不區分重音、角色敏感、寬度敏

日本_中国	日文 -90、不區分大小寫、不區分重音、不敏感角色、不區分寬度
日文_西班牙文_西班牙文	日文 -90、不區分大小寫、不區分重音、不敏感角色、不區分寬度
日文_西班牙文	日語 90，不區分大小寫，不區分重音，不敏感角色，寬度不敏感
日文_中华人民共和国	日文 -90、不區分大小寫、不區分重音、不敏感角色、寬度不敏感
日文_西班牙文_西班牙文	日文-90、不區分大小寫、不區分重音、不敏感角色、寬度不敏感
日文_西班牙	日語 90，不區分大小寫，重音敏感，不敏感角色，寬度不敏感
日文_西班牙	日語 90，不區分大小寫，重音敏感，角色敏感，寬度不敏感
日本阿斯加州	日文 -90、不區分大小寫、重音敏感、角色敏感、不區分寬度
日文_西班牙文_西班牙文	日文-90、不區分大小寫、重音敏感、角色敏感、不區分寬度
日文_西班牙文	日語 90，不區分大小寫，重音敏感，角色敏感，寬度敏感
日語_西班牙語	日文 90、不區分大小寫、重音敏感、角色敏感、寬度敏感
日文_西班牙文_西班牙文	日文 90、不區分大小寫、重音敏感、角色敏感、寬度敏感
日文_西班牙	日文 -90、不區分大小寫、重音區分、不敏感角色、不區分寬度
日文_西班牙文_西班牙文	日文 -90、不區分大小寫、重音區分、不敏感角色、不區分寬度
日文_西班牙	日語 90，不區分大小寫，重音敏感，不敏感角色，寬度敏感
日文_西班牙_西班牙	日文 -90、不區分大小寫、重音區分、不敏感角色、寬度每字節
日文_西班牙文_西班牙文	日文 -90、不區分大小寫、重音區分、不敏感角色、寬度每字節
日本人工智能	日語 90，區分大小寫，不區分重音，不敏感角色，寬度不敏感
日本人_90_艾克	日文 -90，區分大小寫，不區分重音，角色敏感，寬度不敏感
日文_中华人民共和国	日文 -90、區分大小寫、不區分重音、角色敏感、不區分寬度
日文_西班牙文_西班牙文	日文 -90、區分大小寫、不區分重音、角色敏感、不區分寬度
日本第 90 期愛滋病	日文 -90、區分大小寫、不區分重音、角色敏感、寬度敏感

日本人_90_愛加拿大	日文 90、區分大小寫、不區分重音、角色敏感、寬度敏感
日文_中文_英文_W_英文	日文 90、區分大小寫、不區分重音、角色敏感、寬度敏感
日本_90_艾斯克	日文 -90、區分大小寫、不區分重音、不敏感角色、不區分寬度
日文_中文_英文	日文 -90、區分大小寫、不區分重音、不敏感角色、不區分寬度
日本第 90 期愛新聞	日語 90，區分大小寫，不區分重音，不敏感角色，寬度敏感
日文_中国人民共和国	日文 90、區分大小寫、不區分重音、不敏感角色、寬度敏感
日文_中文_英文_W-英文	日文 -90、區分大小寫、不區分重音、不敏感角色、寬度敏感
日文_	日語 90，區分大小寫，重音敏感，不敏感角色，寬度不敏感
日本_90_	日語 90，區分大小寫，重音敏感，角色敏感，寬度不敏感
日文_阿斯加州	日文 90、區分大小寫、重音敏感、角色敏感、不區分寬度
日文_西班牙文_英文	日文 90、區分大小寫、重音敏感、角色敏感、不區分寬度
日文第 90 期	日文 -90，區分大小寫，重音敏感，角色敏感，寬度敏感
日本第 90 期	日文 90、區分大小寫、重音敏感、角色敏感、寬度敏感、
日文第 90 期_英文_W_英文	日文 90、區分大小寫、重音敏感、角色敏感、寬度敏感、
日本_阿斯加	日文 -90、區分大小寫、重音區分、不敏感角色、不區分寬度
日文_90_	日文 -90、區分大小寫、重音區分、不敏感角色、不區分寬度
日文第 90 期	日文 90，區分大小寫，重音敏感，不敏感角色，寬度敏感
日文_法国	日文 90、區分大小寫、重音區分、不敏感角色、寬度敏感
日文_90_	日文 90、區分大小寫、重音區分、不敏感角色、寬度敏感
Japanese_BIN	日文、二進位排序
日本文	日文、二進位碼點比較排序
日本料理_武州角城	日語-武州角城 -100, 二進制排序

日本料理 _ 武州角城 _100	日語-武州角 -100, 二進制碼點比較排序
日本料理 _ 武州角城 _ 西艾	日語-武士角 -100, 不區分大小寫, 重音不敏感, 不敏感,
日本武州角城 _100 日本人	日語-武士角 -100, 不區分大小寫, 重音不敏感, 角質敏感
日本武州角城 _ 中国人民共和国	日語-武士角 -100, 不區分大小寫, 不區分重音, 角色敏感
日本武州角城 _ 西班牙文化中心	日語-武士角 -100, 不區分大小寫, 不區分重音, 角色敏感, 不
日本武州角城 _ 中心 _ 西班牙	日語-武士角 -100, 不區分大小寫, 重音不敏感, 角質敏感
日本武州角城 _100 中国人民共和国	日語-武士角 -100, 不區分大小寫, 不區分重音, 角色敏感
日本武州角城 _100 中心 _ 西班牙文化	日語-武士角 -100, 不區分大小寫, 不區分重音, 角質敏感
日本武州角城 _100 中國	日語-武士角 -100, 不區分大小寫, 不區分重音, 不敏感角
日本武州角城 _100 中心 _ 西班牙文	日語-武士角 -100, 不區分大小寫, 不區分重音, 不敏感角色
日本武州角城 _100 名中心	日語-武士角 -100, 不區分大小寫, 不區分重音, 不敏感角
日本武州角城 _100 中國愛華盛頓	日語-武士角 -100, 不區分大小寫, 不區分重音, 不敏感角
日本武州角城 _ 中心 _ 西班牙文化	日語-武士角 -100, 不區分大小寫, 不區分重音, 不敏感角
日本武州角城	日語-武士角 -100, 不區分大小寫, 重音敏感, 不敏感角色
日本武州角城 _ 西方	日語-武士角 -100, 不區分大小寫, 重音敏感, 角質敏感,
日本武州角城 _ 中国	日語-武士角 -100, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城 _ 西阿拉伯文	日語-武士角 -100, 不區分大小寫, 重音敏感, 角色敏感, 不區
日本武州角城 _100 中國	日語-武士角 -100, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城 _100 加州	日語-武士角 -100, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城 _ 西南部 _ 西班牙文	日語-武士角 -100, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城	日語-武士角 -100, 不區分大小寫, 重音敏感, 不區分角色
日本武州角城 _ 西洋文化	日語-武士角 -100, 不區分大小寫, 重音敏感, 不敏感角度

日本武州角城_100 新聞	日語-武士角 -100, 不區分大小寫, 重音敏感, 不敏感角色
日本武州角城_西南	日語-武士角 -100, 不區分大小寫, 重音敏感, 不敏感角色
日本武州角城_100 中心_西班牙文	日語-武士角 -100, 不區分大小寫, 重音敏感, 不敏感角色
日本武州角城_100	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色
日本料理_武州角城_100	日語-武士角 -100, 區分大小寫, 重音不敏感, 角質敏感,
日本武州角城_艾克斯	日文-武士角 -100、區分大小寫、不區分重音、角色敏感、
日本武州角城_100 人民幣_英文版	日語-武士角 -100, 區分大小寫, 不區分重音, 角色敏感, 不區
日本武州角城_100 人民幣	日語-武士角 -100, 區分大小寫, 不區分重音, 角質敏感,
日本武州角城_100 人民共和國	日語-武士角 -100, 區分大小寫, 不區分重音, 角色敏感,
日本武州角城_100_英文版本	日語-武士角 -100, 區分大小寫, 不區分重音, 角色敏感, 寬度
日本武州角城	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色
日本武州角城_100 人工智能	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色, 不
日本武州角城_100 人工智能	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色
日本武州角城_100 人工智能	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色
日本武州角城_100 人工智能文化	日語-武士角 -100, 區分大小寫, 不區分重音, 不敏感角色
日本武州角城	日語-武士角 -100, 區分大小寫, 重音敏感, 不敏感角色,
日本武州角城_100	日語-武士角 -100, 區分大小寫, 重音敏感, 角質敏感, 寬
日本武州角城	日文-武士角 -100、區分大小寫、區分重音、角色敏感、不
日本武州角城_西班牙文化	日語-武士角 -100, 區分大小寫, 重音敏感, 角質敏感, 不
日本武州角城_100	日語-武士角 -100, 區分大小寫, 重音敏感, 角質敏感, 寬
日本武州角城_100 加州	日文-武士角 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度每
日本武州角城_100_英文版	日語-武士角 -100, 區分大小寫, 重音敏感, 角色敏感, 寬度每

日本武州角城 _100	日語-武士角 -100 , 區分大小寫 , 重音敏感 , 不區分角色 ,
日本武州角城 _ 法国	日語-武士角 -100, 區分大小寫, 重音敏感, 不敏感角色, 不區分
日本武州角城 _100	日語-武士角 -100 , 區分大小寫 , 重音敏感 , 不敏感角色 ,
日本武州角城 _100	日語-武士角 -100 , 區分大小寫 , 重音敏感 , 不敏感角色 ,
日本武州角城 _100 _ 新加坡文化	日語-武士角 -100 , 區分大小寫 , 重音敏感 , 不敏感角色 ,
日本料理 _ 武州角城	日語-武州角城 -140, 二進制排序
日本料理 _ 武州角城 _140	日語-武州角城 -140, 二進制碼點比較排序
日本料理 _ 武州角城	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日本武州角城 _140 日本人	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日文武宿 _ 角城 _ 西班牙文	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日本武州角城 _ 中国人民共和国	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日文武宿 _ 角城 _ 西班牙文化中心	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日本武州角城 _140 中國	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日文武州角城 _ 西班牙文化中心	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日本武州角城 _140 中国人民共和国高速公司	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日文武宿 _ 角城 _ 西班牙文化中心	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 角色敏感
日本武州角城 _ 西班牙文	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日本武州角城 _ 中国人民共和国	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日文武宿 _ 角城 _ 西班牙文字塔	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日本武州角城 _140 日本	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日文武州角城 _140 中心 _ 西班牙文	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角
日本武州角城 _ 中国人民共和国	日語-武士角 -140 , 不區分大小寫 , 不區分重音 , 不敏感角

日本武州市角城_140 中心_ 西班牙文	日語-武士角 -140, 不區分大小寫, 不區分重音, 不敏感角
日本武州角城	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日本武州角城市	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日文武宿_ 角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城_ 西班牙	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日本武宿_ 角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城市	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日文武州角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日本武州角城_ 中国高速公司	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日文武州角城_ 西班牙文化中心	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色敏感,
日文武州角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日本武州角城_ 西班牙	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日文武宿_ 角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日本武州角城市	日語-武士角 -140, 不區分大小寫, 重音敏感, 角色不敏感
日文武州角城_ 西班牙文	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日本武州角城_ 中国高速公司	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日文武州角城_ 西班牙文化站	日語-武士角 -140, 不區分大小寫, 重音敏感, 不敏感角
日本料理_ 武州角城	日語-武士角 -140, 區分大小寫, 不區分重音, 不敏感角
日本武州角城_140	日語-武士角 -140, 區分大小寫, 重音不敏感, 角色敏感,
日本武州角城_140 英文版	日語-武士角 -140, 區分大小寫, 重音不敏感, 角色敏感,
日本武州角城市_ 人工智能	日語-武士角 -140, 區分大小寫, 不區分重音, 角色敏感,
日本武州市角城_140 英文版	日語-武士角 -140, 區分大小寫, 不區分重音, 角色敏感,

日本武州角城市	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 角色敏感 ,
日文武州角城 _140 英文版	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 角色敏感 ,
日本武州角城市 _ 愛滋病網	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 角色敏感 ,
日本武州角城 _140 英文版本	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 角色敏感 ,
日本武州角城 _ 英文	日語-武士角 -140 , 區分大小寫 , 不區分重音 , 不敏感角色
日本武州角城 _14	日語-武士角 -140 , 區分大小寫 , 不區分重音 , 不敏感角色
日本武州角城 _140 英文版	日語-武士角 -140 , 區分大小寫 , 不區分重音 , 不敏感角色
日本武州角城市	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 不敏感角色
日文武州角城 _140 英文版	日語-武士角 -140 , 區分大小寫 , 重音不敏感 , 不敏感角色
日本武州角城市 _ 人工智能資源	日語-武士角 -140 , 區分大小寫 , 不區分重音 , 不敏感角色
日文武州角城 _140 英文版本	日語-武士角 -140 , 區分大小寫 , 不區分重音 , 不敏感角色
日本武州角城	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 不敏感角度 ,
日本武州角城	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日文武州角城 _ 法国	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日本武州角城 _ 法国	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日本武州角城 _140 日本文	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日本武州角城市	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日文武州角城 _ 法国文化学院	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日本武州角城 _ 法国新加坡	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日文武州角城 _140 加州文化中心	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬
日本武州角城 _ 英文	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 不敏感角色 ,
日本武州角城市	日語-武士角 -140 , 區分大小寫 , 重音敏感 , 不敏感角度 ,

日文武州角城_14	日語-武士角 -140, 區分大小寫, 重音敏感, 不敏感角度,
日本武州角城市	日語-武士角 -140, 區分大小寫, 重音敏感, 角色不敏感,
日文武州角城_法国文化学院	日語-武士角 -140, 區分大小寫, 重音敏感, 不敏感角色,
日本武州角城市_法定	日語-武士角 -140, 區分大小寫, 重音敏感, 不敏感角色,
日文武州角城_14. 新聞中心	日語-武士角 -140, 區分大小寫, 重音敏感, 不敏感角色,
日本人	日語, 不區分大小寫, 不區分重音, 不敏感角色, 寬度不敏感,
日語西班牙	日語, 不區分大小寫, 不區分重音, 角色敏感, 寬度不敏感,
日文愛西班牙文	日語, 不區分大小寫, 不區分重音, 角色敏感, 寬度敏感,
日文西班牙文	日語, 不區分大小寫, 不區分重音, 不敏感角色, 寬度敏感,
Japanese_CI_AS	日文、不區分大小寫、區分重音、不區分假名、不區分寬度,
日文加州	日語, 不區分大小寫, 重音敏感, 角色敏感, 寬度不敏感,
日文西班牙文	日語, 不區分大小寫, 重音敏感, 角色敏感, 寬度敏感,
日文西班牙文	日語, 不區分大小寫, 重音敏感, 不敏感角色, 寬度敏感,
日本人工智能	日語, 區分大小寫, 不區分重音, 不敏感角色, 寬度不敏感,
日本人民共和国	日語, 區分大小寫, 不區分重音, 角色敏感, 寬度不敏感,
日本企業社會責任	日語, 區分大小寫, 不區分重音, 角色敏感, 寬度敏感,
日本企業社會責任	日語, 區分大小寫, 不區分重音, 不敏感角色, 寬度敏感,
Japanese_CS_AS	日文、區分大小寫、區分重音、不區分假名、不區分寬度,
日文_反社会	日語, 區分大小寫, 重音敏感, 角色敏感, 寬度不敏感,
日文社会委员会	日語, 區分大小寫, 重音敏感, 角色敏感, 寬度敏感,
日文企業社會責任	日語, 區分大小寫, 重音敏感, 不敏感角色, 寬度敏感,
日本尤尼科德宾	日本萬國碼, 二進位排序

日本尤尼科德 BIN2	日文-Unicode, 二進制碼點比較排序
日本人 _ 尤尼科德 (Ci_AI)	日語-Unicode, 不區分大小寫, 重音不區分, 不敏感角色
日本唯科德 Ci_AI_KS	日語-Unicode, 不區分大小寫, 重音不敏感, 角色敏感, 寬度不
日本联合国国家航空公司	日語-Unicode, 不區分大小寫, 重音不敏感, 角色敏感, 寬度不
日本联合国国家航空公司	日語-Unicode, 不區分大小寫, 重音不區分, 不敏感角色
日本唯科德 (CI_A)	日語-Unicode, 不區分大小寫, 重音敏感, 不敏感角色, 寬度不
日本唯科德 _ 西阿斯克	日語-Unicode, 不區分大小寫, 重音敏感, 角色敏感, 寬度不
日本聯合國國家聯盟	日語-Unicode, 不區分大小寫, 重音敏感, 角色敏感, 寬度不
日本联合国国家航空公司	日語-Unicode, 不區分大小寫, 重音敏感, 不敏感角色, 寬度不
日本聯合國企業社會工智能	日語-Unicode, 區分大小寫, 重音不區分, 不敏感角色, 寬度不
日本联合国联合会	日語-Unicode, 區分大小寫, 重音不區分, 角色敏感, 寬度不
日本联合会社会科学院	日語-Unicode, 區分大小寫, 重音不區分, 角色敏感, 寬度不
日本聯合國企業社會責任	日語-Unicode, 區分大小寫, 重音不區分, 不敏感角色, 寬度不
日本联合国联合会	日語-Unicode, 區分大小寫, 重音敏感, 不敏感角色, 寬度不
日本联合国联合会	日語-Unicode, 區分大小寫, 重音敏感, 角色敏感, 寬度不
日本联合国联合社会科学院	日文-Unicode, 區分大小寫, 重音敏感, 角色敏感, 寬度不
日本联合国联合会	日語-Unicode, 區分大小寫, 重音敏感, 不敏感角色, 寬度不
日本西吉斯	日本-西吉斯 100, 二進制排序
日本西班牙文 _ 二号	日文-xJIS-100, 二進制碼點比較排序
日本西吉斯 _ 西班牙	日語 xJIS-100, 不區分大小寫, 不區分重音, 不敏感角色
日本西班牙文 _ 西班牙	日語 xJIS-100, 不區分大小寫, 不區分重音, 角質敏感, 寬度不
日語西班牙語 _ 西班牙語	日文-xJIS-100、不區分大小寫、不區分重音、角色敏感、寬度不

日文西班牙文 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、不區分重音、角色敏感、
日語西班牙語 _ 西班牙語	日語 xJIS-100，不區分大小寫，不區分重音，角色敏感，
日本西班牙文 _ 中国西班牙语	日文-xJIS-100、不區分大小寫、不區分重音、角色敏感、
日文西班牙文 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、不區分重音、角色敏感、
日本西班牙文 _ 西班牙	日文-xJIS-100、不區分大小寫、不區分重音、不敏感角色
日語 _ 西班牙語 _ 西班牙語	日文-xJIS-100、不區分大小寫、不區分重音、不敏感角色
日本西班牙文 _ 西班牙	日文 xJIS-100，不區分大小寫，不區分重音，不敏感角色
日本西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、不區分重音、不敏感角色
日文西班牙文 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、不區分重音、不敏感角色
日本西班牙文 _ 西班牙	日語 xJIS-100，不區分大小寫，重音敏感，不敏感角色，
日本新聞 _ 西班牙語	日語 xJIS-100，不區分大小寫，重音敏感，角色敏感，寬
日本西班牙語 _ 西班牙語	日文-xJIS-100、不區分大小寫、重音敏感、角色敏感、不
日文新聞 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、重音敏感、角色敏感、不
日本新聞 _ 西班牙語	日語 xJIS-100，不區分大小寫，重音敏感，角色敏感，寬
日本新聞中心 _ 西班牙語	日文-xJIS-100、不區分大小寫、重音敏感、角色敏感、寬
日文新聞 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、重音敏感、角色敏感、寬
日本西班牙文 _ 西班牙	日文-xJIS-100、不區分大小寫、重音區分、不敏感角色、
日文新聞 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、重音區分、不敏感角色、
日本新聞 _ 西班牙語	日語 xJIS-100，不區分大小寫，重音敏感，不敏感角色，
日語西班牙語 _ 西班牙語	日文-xJIS-100、不區分大小寫、重音區分、不敏感角色、
日文新聞 _ 西班牙文 _ 西班牙文	日文-xJIS-100、不區分大小寫、重音區分、不敏感角色、
日本西吉斯 _ 人工智能	日語 xJIS-100，區分大小寫，不區分重音，不敏感角色，

日本日語 _ 西班牙文	日語 xJIS-100 , 區分大小寫 , 不區分重音 , 角質敏感 , 不
日本西班牙文 _ 西班牙文	日文-xJIS-100、區分大小寫、不區分重音、角色敏感、不
日文西班牙文 _ 西班牙文 _ 英文	日文-xJIS-100、區分大小寫、不區分重音、角色敏感、不
日本新聞 _ 西班牙文	日語 xJIS-100 , 區分大小寫 , 不區分重音 , 角色敏感 , 寬
日本西班牙文 _ 中国人民共和国	日文-xJIS-100、區分大小寫、不區分重音、角色敏感、寬
日本新聞 _ 西班牙文 _ 英語 _ W-英文	日文-xJIS-100、區分大小寫、不區分重音、角色敏感、寬
日本西班牙文 _ 中国	日文-xJIS-100、區分大小寫、不區分重音、不敏感角色、
日文版本 _ 西班牙文 _ 西班牙文	日文-xJIS-100、區分大小寫、不區分重音、不敏感角色、
日本新聞 _ 西班牙文 _ 愛新聞	日文 xJIS-100 , 區分大小寫 , 不區分重音 , 不敏感角色 ,
日本西班牙文 _ 中国人民共和国	日文-xJIS-100、區分大小寫、不區分重音、不敏感角色、
日文版本 _ 西班牙文 _ 正文 _ 英文	日文-xJIS-100、區分大小寫、不區分重音、不敏感角色、
日本西班牙文 _ 正如	日語 xJIS-100 , 區分大小寫 , 重音敏感 , 不敏感角色 , 寬
日語 _ 日語 _ 正常	日語 xJIS-100 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬度
日本西班牙文 _ 阿斯加州	日文-xJIS-100、區分大小寫、重音敏感、角色敏感、不區
日文新聞 _ 正文 _ _ _ _ _ 英文	日文-xJIS-100、區分大小寫、重音敏感、角色敏感、不區
日本新聞 _ 西班牙文	日語 xJIS-100 , 區分大小寫 , 重音敏感 , 角色敏感 , 寬度
日本新聞 _ 西班牙語	日文-xJIS-100、區分大小寫、重音敏感、角色敏感、寬度
日文新聞 _ 正文 _ _ _ _ _ _ _ _ _ _	日文-xJIS-100、區分大小寫、重音敏感、角色敏感、寬度
日本西班牙文 _ 阿斯加	日文-xJIS-100、區分大小寫、重音區分、不敏感角色、不
日本新聞 _ 正確的方法	日文-xJIS-100、區分大小寫、重音區分、不敏感角色、不
日本新聞 _ 社會責任	日語 xJIS-100 , 區分大小寫 , 重音敏感 , 不敏感角色 , 寬
日本新聞中心 _ 法國	日文-xJIS-100、區分大小寫、重音區分、不敏感角色、寬

日文新聞 _ 反正文 _ 新聞 _ 英文

日文-xJIS-100、區分大小寫、重音區分、不敏感角色、寬

日本西吉斯文

日語-西吉 -140, 二進制排序

日文日文 _ 西班牙文

日文-xJIS-140, 二進制碼點比較排序

日本西部 _ 西班牙

日語 xjis-140 , 不區分大小寫 , 重音不敏感 , 不敏感角色 ,

日本西班牙文 _ 西班牙

日語 xjis-140 , 不區分大小寫 , 重音不敏感 , 角色敏感 , 寬

日語 _ 西班牙語 _ 西班牙語

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 角色敏感 ,

日語西班牙語 _ 西班牙語

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 角色敏感 ,

日文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 角色敏感 ,

日語西班牙語 _ 西班牙語

日語 xjis-140 , 不區分大小寫 , 重音不敏感 , 角色敏感 , 寬

日文西班牙文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 重音不敏感 , 角色敏感 ,

日本西班牙文 _ 中国人民共和国高等教育中心

日文 xjis-140 , 不區分大小寫 , 重音不敏感 , 角色敏感 , 寬

日文西班牙文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 重音不敏感 , 角色敏感 ,

日語 _ 西班牙語 _ 西班牙語

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

日本西班牙文 _ 中国人民共和国

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

日文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

日語西班牙語 _ 西班牙

日語 xjis-140 , 不區分大小寫 , 不區分重音 , 不敏感角色 ,

日文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

日本西班牙文 _ 中国人民共和国西班牙语

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

日文 _ 西班牙文 _ 西班牙文 _ 英文

日語-xJIS-140 , 不區分大小寫 , 不區分重音 , 不敏感角色

Japanese_XJIS_140_CI_AS

Japanese-XJIS-140、不區分大小寫、區分重音、不區分作

日文 _ 西班牙文 _ 西班牙

日語 xjis-140 , 不區分大小寫 , 重音敏感 , 角色敏感 , 寬度

日文 _ 西班牙文 _ 西班牙文

日語-xJIS-140 , 不區分大小寫 , 重音敏感 , 角色敏感 , 寬

Japanese_XJIS_140_CI_AS_KS_VSS

日文 _ 西班牙文 _ 西班牙文

日本新聞 _ 西班牙語

日文 _ 西班牙文 _ 西班牙文

日本新聞中心 _ 西班牙語 _ WSS

日文新聞 _ 西班牙文 _ 西班牙文

日文 _ 西班牙文 _ 西班牙文

Japanese_XJIS_140_CI_AS_VSS

日文 _ 西班牙文 _ 西班牙文

日本新聞 _ 西班牙

日文 _ 西班牙文 _ 西班牙文

日文新聞中心 _ 西班牙文 _ VSS

日文新聞 _ 西班牙文 _ W _ 英文

日本西吉斯 _ 人工智能

日本西班牙文 _ 西班牙

日文西班牙文 _ 西班牙文

日本西班牙文 _ 中华人民共和国

日本新聞 _ 西班牙文 _ 英文版

日本新聞 _ 西班牙文

日文西班牙文 _ 西班牙文 _ 英文

日本新聞中心 _ 愛滋病學院

日文版本 _ 西班牙文 _ 西班牙文

Japanese-XJIS-140、不區分大小寫、區分重音、區分假名

日語-xJIS-140，不區分大小寫，重音敏感，角色敏感，不

日語 xJIS-140，不區分大小寫，重音敏感，角色敏感，寬

日語-xJIS-140，不區分大小寫，重音敏感，角色敏感，寬

日文 XJIS-140、不區分大小寫、重音敏感、角色敏感、寬

日語-xJIS-140，不區分大小寫，重音敏感，角色敏感，寬

日語-xJIS-140，不區分大小寫，重音敏感，不敏感角色，

Japanese-XJIS-140、不區分大小寫、區分重音、不區分假

日語-xJIS-140，不區分大小寫，重音敏感，不敏感角色，

日語 xjis-140，不區分大小寫，重音敏感，不敏感角色，寬

日語-xJIS-140，不區分大小寫，重音敏感，不敏感角色，

日文-xjis-140，不區分大小寫，重音敏感，不敏感角色，寬

日語-xJIS-140，不區分大小寫，重音敏感，不敏感角色，

日語-xJIS-140，區分大小寫，不區分重音，不敏感角色，

日語-xJIS-140，區分大小寫，不區分重音，角色敏感，寬

日語-xJIS-140，區分大小寫，不區分重音，角色敏感，寬

日文 xjis-140、區分大小寫、不區分重音、角色敏感、不區

日語-xJIS-140，區分大小寫，不區分重音，角色敏感，寬

日語 xJIS-140，區分大小寫，不區分重音，角色敏感，寬

日語-xJIS-140，區分大小寫，不區分重音，角色敏感，寬

日文 XJIS-140、區分大小寫、不區分重音、角色敏感、寬

日文-xJIS-140、區分大小寫、不區分重音、角色敏感、寬

日文 _ 西班牙文 _ 西班牙文	日語-xJIS-140, 區分大小寫, 不區分重音, 不敏感角色,
日語 _ 日語 _ 日語 _ 新聞資源	日語-xJIS-140, 區分大小寫, 不區分重音, 不敏感角色,
日文版本 _ 西班牙文 _ 英文	日語-xJIS-140, 區分大小寫, 不區分重音, 不敏感角色,
日本新聞 _ 西班牙文	日語 xjis-140, 區分大小寫, 不區分重音, 不敏感角色, 寬
日文 _ 西班牙文 _ 西班牙文	日語-xJIS-140, 區分大小寫, 不區分重音, 不敏感角色,
日本新聞 _ 社會責任 _ 英文版	日文 xjis-140、區分大小寫、不區分重音、不敏感角色、寬
日文版本 _ 西班牙文 _ 英文 .V.	日語-xJIS-140, 區分大小寫, 不區分重音, 不敏感角色,
日本新聞 _ 正如	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
日本新聞 _ 正式	日文 xjis-140, 區分大小寫, 重音敏感, 角色敏感, 不區分
日文新聞 _ 西班牙文	日語-xJIS-140, 區分大小寫, 重音敏感, 角色敏感, 寬度
日文新加索 _ 法律系統	日文 XJIS-140、區分大小寫、重音敏感、角色敏感、不區
日本新聞 _ 西班牙文 _ 英文	日語-xJIS-140, 區分大小寫, 重音敏感, 角色敏感, 不區
Japanese_XJIS_140_CS_AS_KS_WS	日文 XJIS-140, 區分大小寫, 重音敏感, 角色敏感, 寬度
日文新聞 _ 西班牙文 _ 英文	日語-xJIS-140, 區分大小寫, 重音敏感, 角色敏感, 寬度
日文新聞中心 _ 社會科學院	日文 XJIS-140、區分大小寫、重音敏感、角色敏感、寬度
日文新聞 _ 正確版本 _ 英文 .WS_VS_W	日文-xJIS-140、區分大小寫、重音敏感、角色敏感、寬度
日文版本 _ 西班牙文	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
日本新聞 _ 俄羅斯 _ 社會責任	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
日文版本 _ 西班牙文	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 不
日本新聞 _ 西班牙文	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
日文正版英文 .W-英文	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
日本新聞 _ 日語 _ 社會責任 _ 新聞網	日文 XJIS-140、區分大小寫、重音區分、不敏感角色、寬

日文版本_新聞_英文.WS_W	日語-xJIS-140, 區分大小寫, 重音敏感, 不敏感角色, 寬
Korean_Wansung_CI_AS	Korean-Wansung、不區分大小寫、區分重音、不區分假名
Latin1_General_100_BIN	Latin1-General-100、二進位排序
Latin1_General_100_BIN2	Latin1-General-100、二進位代碼點比較排序
Latin1_General_100_BIN2_UTF8	拉丁語-通用 -100, 二進制碼點比較排序, UTF8
Latin1_General_100_CI_AS	Latin1-General-100、不區分大小寫、區分重音、不區分假名
Latin1_General_100_CI_AS_SC_UTF8	拉丁裔 1-一般 -100、不區分大小寫、重音區分、不敏感角
Latin1_General_BIN	Latin1-General、二進位排序
Latin1_General_BIN2	Latin1-General、二進位代碼點比較排序
Latin1_General_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名
Latin1_General_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、
Latin1_General_CI_AS_KS	Latin1-General、不區分大小寫、區分重音、區分假名、不
Latin1_General_CS_AS	Latin1-General、區分大小寫、區分重音、不區分假名、不
Modern_Spanish_CI_AS	Modern-Spanish、不區分大小寫、區分重音、不區分假名
SQL_1xCompat_CP850_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、
SQL_Latin1_General_CP1_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名
SQL_拉丁語一般_CP1_C_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、
SQL_Latin1_General_CP1_CS_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、
拉丁語 1-通用_中国	拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音
拉丁語 1-一般_	拉丁語 1-針對 Unicode 資料的一般、區分大小寫、區分重
拉丁語素_通用_中国	拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音
拉丁語 1-通用_	拉丁文 1-針對 Unicode 資料的一般、區分大小寫、區分重

拉丁语系列_通用_中国人工智能

拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、不區分 1253 上的 SQL 伺服器排序順序 124

拉丁语系列

拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音上的 SQL Server 排序順序 114

拉丁語 1-通用_

拉丁文 1-針對 Unicode 資料的一般、區分大小寫、區分重

拉丁语字母_通用_CP1254_AS

對於 Unicode 數據，土耳其語，不區分大小寫，重音敏感

SQL_拉丁语一般_CP1254_CS_AS

對於 Unicode 資料，土耳其文，區分大小寫，區分重音，

拉丁語 1-通用_

拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音

拉丁語 1-通用_

拉丁語 1-針對 Unicode 資料的一般、區分大小寫、區分重

SQL_Latin1_General_CP1256_CI_AS

Latin1-General、不區分大小寫、區分重音、不區分假名、

拉丁語 1-通用_

拉丁文 1-針對 Unicode 資料的一般、區分大小寫、區分重

拉丁語 1-通用_中文

拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音上的 SQL Server 排序順序 154

拉丁語 1-通用_

拉丁語 1-對於 Unicode 數據的一般，區分大小寫，區分大排序順序 153

拉丁语素_通用型_中文

拉丁語 1-一般，Unicode 數據的二進制排序，非 Unicode

拉丁语字母表_通用_CP437

拉丁語 1-用於 Unicode 數據的一般二進制代碼點比較排序

SQL_Latin1_General_CP437_CI_AI

Latin1-General、不區分大小寫、不區分重音、不區分假名

拉丁語 1-通用_中文

拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音資料)

SQL_拉丁语一般_CP437_CS_AS

拉丁語 1-針對 Unicode 資料的一般、區分大小寫、區分重料)

SQL_Latin1_General_CP850_BIN

拉丁文 1-一般，Unicode 資料的二進位排序，非 Unicode

SQL_Latin1_General_CP850_BIN2

Latin1-General、二進位代碼點比較排序，適用於非 Unico

SQL_Latin1_General_CP850_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名、Unicode 資料)
SQL_Latin1_General_CP850_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、
SQL_拉丁语一般_CP850_CS_AS	拉丁文 1-針對 Unicode 資料的一般、區分大小寫、區分重
SQL_拉丁 1_一般_前列_中文	拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音
SQL_拉丁语一般_前列表	拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音 資料)
SQL_拉丁语一般_前列表	拉丁文 1-針對 Unicode 資料的一般、不區分大小寫、重音
Thai_CI_AS	泰語、不區分大小寫、區分重音、不區分假名、不區分寬度

RDS Custom for SQL Server 資料庫執行個體的本機時區

RDS Custom for SQL Server 資料庫執行個體的時區是根據預設來設定的。目前預設值是世界協調時間 (UTC)。您也可以將資料庫執行個體的時區改設為本機時區，以符合應用程式的時區。

您可以在第一次建立資料庫執行個體時設定時區。您可以使用 Amazon RDS API 建立資料庫執行 [AWS Management Console](#) 個體動作或命令來 [建立資料庫執行個體](#)。AWS CLI [create-db-instance](#)

如果您的資料庫執行個體是多可用區部署的一部分，則在容錯移轉時，時區仍會維持在您設定的本機時區。

當您要求 point-in-time 還原時，您可以指定還原的時間。時間以您當地的時區顯示。如需更多詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

下列是在資料庫執行個體上設定本機時區的限制：

- 您可以在建立執行個體期間設定資料庫執行個體的時區，但無法修改現有適用於 SQL Server 資料庫執行個體的 RDS Custom 時區。
- 如果修改現有適用於 SQL Server 資料庫執行個體的 RDS Custom 時區，則 RDS Custom 會將資料庫執行個體狀態變更為 unsupported-configuration，並傳送事件通知。
- 您無法將快照從某個時區的資料庫執行個體還原至另一個時區的資料庫執行個體。

- 強烈建議您不要將備份檔案從某個時區還原至不同時區。如果您將備份檔案從某個時區還原至不同時區，則必須稽核您的查詢及應用程式，是否受到時區變更的影響。如需更多詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

支援的時區

您可以將本機時區設為下表中列出的其中一個值。

RDS Custom for SQL Server 支援的時區。

時區	標準時間偏移	描述	備註
阿富汗標準時間	(UTC+04:30)	喀布爾	此時區不會遵守日光節約時間。
阿拉斯加標準時間	(UTC-09:00)	阿拉斯加州	
阿留申標準時間	(UTC-10:00)	阿留申群島	
阿爾泰標準時間	(UTC+07:00)	巴爾瑙爾，戈爾諾-阿爾泰斯克	
沙烏地阿拉伯標準時間	(UTC+03:00)	科威特，利雅德	此時區不會遵守日光節約時間。
阿拉伯標準時間	(UTC+04:00)	阿布達比、馬斯喀特	
阿拉伯標準時間	(UTC+03:00)	巴格達	此時區不會遵守日光節約時間。
阿根廷標準時間	(UTC-03:00)	布宜諾斯艾利斯市	此時區不會遵守日光節約時間。
阿斯特拉罕標準時間	(UTC+04:00)	阿斯特拉罕，烏里揚諾夫斯克	
大西洋標準時間	(UTC-04:00)	大西洋時間 (加拿大)	
AUS 中央標準時間	(UTC+09:30)	達爾文	此時區不會遵守日光節約時間。

時區	標準時間偏移	描述	備註
澳洲中西部標準時間	(UTC+08:45)	尤克拉	
AUS 東部標準時間	(UTC+10:00)	坎培拉、墨爾本、雪梨	
亞塞拜然標準時間	(UTC+04:00)	巴庫	
亞速爾標準時間	(UTC-01:00)	亞速爾	
巴伊亞標準時間	(UTC-03:00)	薩爾瓦多	
孟加拉標準時間	(UTC+06:00)	達卡	此時區不會遵守日光節約時間。
白俄羅斯標準時間	(UTC+03:00)	明斯克	此時區不會遵守日光節約時間。
布干維爾標準時間	(UTC+11:00)	布干維爾島	
加拿大中央標準時間	(UTC-06:00)	薩斯喀徹溫	此時區不會遵守日光節約時間。
維德角標準時間	(UTC-01:00)	維德角群島	此時區不會遵守日光節約時間。
高加索標準時間	(UTC+04:00)	葉里溫	
中央 澳洲標準時間	(UTC+09:30)	阿德萊德	
中美洲標準時間	(UTC-06:00)	中央美洲	此時區不會遵守日光節約時間。
中亞標準時間	(UTC+06:00)	阿斯塔納	此時區不會遵守日光節約時間。
中央巴西標準時間	(UTC-04:00)	古雅巴	

時區	標準時間偏移	描述	備註
中歐標準時間	(UTC+01:00)	貝爾格勒、布拉提斯拉瓦、布達佩斯、盧布亞納、布拉格	
中歐標準時間	(UTC+01:00)	塞拉耶佛、斯高彼亞、華沙、札格雷布	
中央太平洋標準時間	(UTC+11:00)	索羅門群島、新喀里多尼亞	此時區不會遵守日光節約時間。
中央標準時間	(UTC-06:00)	中央時間 (美國和加拿大)	
中央標準時間 (墨西哥)	(UTC-06:00)	瓜達拉哈拉、墨西哥城、蒙特利	
查塔姆群島標準時間	(UTC+12:45)	查塔姆群島	
中國標準時間	(UTC+08:00)	北京、重慶、香港、烏魯木齊	此時區不會遵守日光節約時間。
古巴標準時間	(UTC-05:00)	哈瓦那	
國際換日線標準時間	(UTC-12:00)	國際換日線以西	此時區不會遵守日光節約時間。
東部非洲標準時間	(UTC+03:00)	奈洛比	此時區不會遵守日光節約時間。
東部澳洲標準時間	(UTC+10:00)	布里斯班	此時區不會遵守日光節約時間。
東部歐洲標準時間	(UTC+02:00)	奇西瑙	
東部南美洲標準時間	(UTC-03:00)	巴西利亞	
復活節島標準時間	(UTC-06:00)	復活節島	

時區	標準時間偏移	描述	備註
東部標準時間	(UTC-05:00)	東部時間 (美國和加拿大)	
東部標準時間 (墨西哥)	(UTC-05:00)	切圖馬爾	
埃及標準時間	(UTC+02:00)	開羅	
葉卡捷琳堡標準時間	(UTC+05:00)	葉卡捷琳堡	
斐濟標準時間	(UTC+12:00)	斐濟	
FLE 標準時間	(UTC+02:00)	赫爾辛基, 基輔, 里加, 索菲亞, 塔林, 維爾紐斯	
喬治亞標準時間	(UTC+04:00)	提比里斯	此時區不會遵守日光節約時間。
GMT 標準時間	(UTC)	都柏林、愛丁堡、里斯本、倫敦	此時區與格林威治標準時間不同。此時區會遵守日光節約時間。
格陵蘭標準時間	(UTC-03:00)	格陵蘭	
格林威治標準時間	(UTC)	蒙羅維亞、雷克雅維克	此時區不會遵守日光節約時間。
GTB 標準時間	(UTC+02:00)	雅典、布加勒斯特	
海地標準時間	(UTC-05:00)	海地	
夏威夷標準時間	(UTC-10:00)	夏威夷州	
印度標準時間	(UTC+05:30)	清奈、加爾各答、孟買、新德里	此時區不會遵守日光節約時間。
伊朗標準時間	(UTC+03:30)	德黑蘭	

時區	標準時間偏移	描述	備註
以色列標準時間	(UTC+02:00)	耶路撒冷	
約旦標準時間	(UTC+02:00)	安曼	
卡里寧格勒標準時間	(UTC+02:00)	卡里寧格勒	
堪察加標準時間	(UTC+12:00)	彼得保羅夫斯克-堪察加斯克 – 老	
韓國標準時間	(UTC+09:00)	首爾	此時區不會遵守日光節約時間。
利比亞標準時間	(UTC+02:00)	的黎波里	
來因群島標準時間	(UTC+14:00)	刻里提瑪斯島	
羅豪標準時間	(UTC+10:30)	羅豪島	
馬加丹標準時間	(UTC+11:00)	馬加丹	此時區不會遵守日光節約時間。
麥哲倫標準時間	(UTC-03:00)	蓬塔阿雷納斯	
馬克薩斯標準時間	(UTC-09:30)	馬克薩斯群島	
模里西斯標準時間	(UTC+04:00)	路易士港	此時區不會遵守日光節約時間。
中東標準時間	(UTC+02:00)	貝魯特	
蒙特維多標準時間	(UTC-03:00)	蒙特維多	
摩洛哥標準時間	(UTC+01:00)	卡薩布蘭卡	
山地標準時間	(UTC-07:00)	山地時間 (美國和加拿大)	
山地標準時間 (墨西哥)	(UTC-07:00)	赤瓦瓦州、拉巴斯、馬薩特蘭	

時區	標準時間偏移	描述	備註
緬甸標準時間	(UTC+06:30)	仰光	此時區不會遵守日光節約時間。
北部中亞標準時間	(UTC+07:00)	新西伯利亞	
納米比亞標準時間	(UTC+02:00)	溫吐克	
尼泊爾標準時間	(UTC+05:45)	加德滿都	此時區不會遵守日光節約時間。
紐西蘭標準時間	(UTC+12:00)	奧克蘭、威靈頓	
紐芬蘭標準時間	(UTC-03:30)	紐芬蘭	
諾福克標準時間	(UTC+11:00)	諾福克島	
東北亞標準時間	(UTC+08:00)	伊爾庫次克	
北亞標準時間	(UTC+07:00)	克拉斯諾亞爾斯克	
北韓標準時間	(UTC+09:00)	平壤	
鄂木斯克標準時間	(UTC+06:00)	鄂木斯克	
太平洋 SA 標準時間	(UTC-03:00)	聖地牙哥	
太平洋標準時間	(UTC-08:00)	太平洋時間 (美國和加拿大)	
太平洋標準時間 (墨西哥)	(UTC-08:00)	下加利福尼亞州	
巴基斯坦標準時間	(UTC+05:00)	伊斯蘭馬巴德，喀拉蚩	此時區不會遵守日光節約時間。
巴拉圭標準時間	(UTC-04:00)	亞松森	
羅馬標準時間	(UTC+01:00)	布魯塞爾、哥本哈根、馬德里、巴黎	

時區	標準時間偏移	描述	備註
俄羅斯時區 10	(UTC+11:00)	查庫爾達克	
俄羅斯時區 11	(UTC+12:00)	阿納底，彼得保羅夫斯克-堪察加斯克	
俄羅斯時區 3	(UTC+04:00)	伊熱夫斯克，薩馬拉	
俄羅斯標準時間	(UTC+03:00)	莫斯科、聖彼得堡、伏爾加格勒	此時區不會遵守日光節約時間。
南美洲東部標準時間	(UTC-03:00)	卡宴，福塔雷薩	此時區不會遵守日光節約時間。
SA 太平洋標準時間	(UTC-05:00)	波哥大、利馬、基多、里約布蘭科	此時區不會遵守日光節約時間。
南美洲西部標準時間	(UTC-04:00)	喬治城，拉帕茲，瑪瑙斯，聖胡安	此時區不會遵守日光節約時間。
聖皮爾標準時間	(UTC-03:00)	聖皮埃赫及密克隆	
薩哈林標準時間	(UTC+11:00)	薩哈林	
薩摩亞標準時間	(UTC+13:00)	薩摩亞	
聖多美標準時間	(UTC+01:00)	聖多美普林西比	
薩拉托夫標準時間	(UTC+04:00)	薩拉托夫	
SE 亞洲標準時間	(UTC+07:00)	曼谷、河內、雅加達	此時區不會遵守日光節約時間。
新加坡標準時間	(UTC+08:00)	吉隆坡、新加坡	此時區不會遵守日光節約時間。
南非標準時間	(UTC+02:00)	哈拉雷，比勒陀利亞	此時區不會遵守日光節約時間。

時區	標準時間偏移	描述	備註
斯里蘭卡標準時間	(UTC+05:30)	斯里哈亞華登尼普拉	此時區不會遵守日光節約時間。
蘇丹標準時間	(UTC+02:00)	喀土穆	
敘利亞標準時間	(UTC+02:00)	大馬士革	
台北標準時間	(UTC+08:00)	台北	此時區不會遵守日光節約時間。
塔斯馬尼亞標準時間	(UTC+10:00)	荷巴特	
托坎廷斯標準時間	(UTC-03:00)	阿拉瓜	
東京標準時間	(UTC+09:00)	大阪、札幌、東京	此時區不會遵守日光節約時間。
托木斯克標準時間	(UTC+07:00)	托木斯克	
東加標準時間	(UTC+13:00)	努瓜婁發	此時區不會遵守日光節約時間。
外貝加爾標準時間	(UTC+09:00)	赤塔	
土耳其標準時間	(UTC+03:00)	伊斯坦堡	
土克斯及開科斯群島標準時間	(UTC-05:00)	土克斯及開科斯群島	
烏蘭巴托標準時間	(UTC+08:00)	烏蘭巴托	此時區不會遵守日光節約時間。
美國東部標準時間	(UTC-05:00)	印第安那州 (東部)	
美國山地標準時間	(UTC-07:00)	亞利桑那州	此時區不會遵守日光節約時間。
UTC	UTC	國際標準時間	此時區不會遵守日光節約時間。

時區	標準時間偏移	描述	備註
UTC-02	(UTC-02:00)	國際標準時間-02	此時區不會遵守日光節約時間。
UTC-08	(UTC-08:00)	國際標準時間-08	
UTC-09	(UTC-09:00)	國際標準時間-09	
UTC-11	(UTC-11:00)	國際標準時間-11	此時區不會遵守日光節約時間。
UTC+12	(UTC+12:00)	國際標準時間+12	此時區不會遵守日光節約時間。
UTC+13	(UTC+13:00)	國際標準時間+13	
委內瑞拉標準時間	(UTC-04:00)	加拉加斯	此時區不會遵守日光節約時間。
符拉迪沃斯托克標準時間	(UTC+10:00)	符拉迪沃斯托克	
伏爾加格勒標準時間	(UTC+04:00)	伏爾加格勒	
西部澳洲標準時間	(UTC+08:00)	珀斯	此時區不會遵守日光節約時間。
西部西非標準時間	(UTC+01:00)	中西非	此時區不會遵守日光節約時間。
西部歐洲標準時間	(UTC+01:00)	阿姆斯特丹、柏林、羅馬、斯德哥爾摩、維也納	
西部蒙古標準時間	(UTC+07:00)	霍夫德	
西亞標準時間	(UTC+05:00)	阿什哈巴德，塔什干	此時區不會遵守日光節約時間。
約旦河西岸標準時間	(UTC+02:00)	加薩，希伯倫	

時區	標準時間偏移	描述	備註
西太平洋標準時間	(UTC+10:00)	關島，莫爾斯貝港	此時區不會遵守日光節約時間。
亞庫茲克標準時間	(UTC+09:00)	亞庫茲克	

將服務主要金鑰與 SQL 伺服器的 RDS 自訂搭配使用

適用於 SQL 伺服器的 RDS 自訂支援使用服務主要金鑰 (SMK)。RDS 自訂會在您的 RDS 自訂 SQL 伺服器資料庫執行個體的整個生命週期內保留相同的 SMK。透過保留相同的 SMK，您的資料庫執行個體可以使用以 SMK 加密的物件，例如連結的伺服器密碼和認證。如果您使用異地同步備份部署，RDS Custom 也會在主要和次要資料庫執行個體之間同步處理和維護 SMK。

主題

- [區域和版本可用性](#)
- [支援的功能](#)
- [使用 TDE](#)
- [配置功能](#)
- [要求與限制](#)

區域和版本可用性

所有可用於 SQL 伺服器的 RDS 自訂區域都支援使用 SMK，適用於 RDS 自訂上所有可用的 SQL 伺服器版本。如需有關適用於 SQL 伺服器的 Amazon RDS 與 RDS 自訂版本和區域可用性的詳細資訊，請參閱[適用於 SQL 伺服器的 RDS 自訂支援的區域和資料庫引擎](#)。

支援的功能

將 SMK 與 SQL 伺服器的 RDS 自訂搭配使用時，支援下列功能：

- 透明資料加密 (TDE)
- 列級加密
- 資料庫郵件
- 連結伺服器
- SQL 伺服器整合服務 (SSIS)

使用 TDE

SMK 能夠設定透明資料加密 (TDE)，該加密功能會在資料寫入儲存裝置之前加密資料，並在從儲存區讀取資料時自動解密資料。與 SQL 伺服器 RDS 不同，在 RDS 自訂 SQL 伺服器資料庫執行個體上設定 TDE 不需要使用選項群組。建立憑證和資料庫加密金鑰後，您可以執行下列指令，在資料庫層級開啟 TDE：

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON;
```

如需將 TDE 與 RDS 搭配 SQL 伺服器使用的詳細資訊，請參閱[支援 SQL Server 的透明資料加密](#)。

如需 Microsoft SQL Server 中 TDE 的詳細資訊，請參閱 Microsoft 文件中的[透明資料加密](#)。

配置功能

如需設定使用 SMK 搭配 SQL 伺服器自訂之功能的詳細步驟，您可以在 Amazon RDS 資料庫部落格中使用下列文章：

- 連結伺服器：[設定連結伺服器上的 RDS 自訂 SQL 伺服器](#)。
- SSIS：[將 SSIS 套件移轉至 SQL 伺服器的 RDS 自訂](#)。
- TDE：[使用 RDS 自訂 SQL 伺服器上的 TDE 來保護您的資料](#)。

要求與限制

將 SMK 與 SQL Server 資料庫執行個體的 RDS 自訂搭配使用時，請記住下列需求和限制：

- 如果您在資料庫執行個體上重新產生 SMK，則應立即執行手動資料庫快照。我們建議盡可能避免重新產生 SMK。
- 您必須維護伺服器憑證和資料庫主要金鑰密碼的備份。如果您不維護這些備份，則可能會導致數據丟失。
- 如果您設定 SSIS，則應該使用 SSM 文件將適用於 SQL Server 資料庫執行個體的 RDS 自訂加入網域，以便進行擴展運算或主機取代。
- 啟用 TDE 或資料行加密時，資料庫備份會自動加密。當您執行快照還原或時間點復原時，將還原來自來源資料庫執行個體的 SMK 以解密還原的資料，並且會產生新的 SMK 以重新加密還原執行個體上的資料。

如需有關 Microsoft SQL Server 中的服務主要金鑰的詳細資訊，請參閱 Microsoft 文件中的[SQL 伺服器和資料庫加密金鑰](#)。

設定您的 Amazon RDS Custom for SQL Server 的環境

建立和管理 Amazon RDS Custom for SQL Server 資料庫執行個體的資料庫執行個體之前，請務必執行下列任務。

內容

- [設定 RDS Custom for SQL Server 的先決條件](#)
 - [自動建立執行個體設定檔，使 AWS Management Console](#)
- [步驟 1：將所需許可授予 IAM 主體](#)
- [步驟 2：設定網路、執行個體設定檔和加密](#)
 - [配置 AWS CloudFormation](#)
 - [所需的參數 CloudFormation](#)
 - [下載 AWS CloudFormation 範本檔](#)
 - [使用配置資源 CloudFormation](#)
 - [手動設定](#)
 - [確定您擁有對稱式加密金鑰 AWS KMS](#)
 - [手動建立 IAM 角色和執行個體設定檔](#)
 - [建立 AWSRDSCustomSQLServerInstanceRole IAM 角色](#)
 - [將存取原則新增至 AWSRDSCustomSQLServerInstanceRole](#)
 - [建立您的 RDS Custom for SQL Server 執行個體設定檔](#)
 - [新增 AWSRDSCustomSQLServerInstanceRole 至您的 RDS 自訂 SQL 伺服器執行個體設定檔](#)
 - [手動設定您的 VPC](#)
 - [設定您的 VPC 安全群組](#)
 - [設定相依項目的端點 AWS 服務](#)
 - [設定個體中繼資料服務](#)
- [跨執行個體限制](#)

Note

如需如何設定必要條件以及啟動適用於 SQL 伺服器的 Amazon RDS 自訂 step-by-step 教學課程，請參閱使用 [CloudFormation 範本開始使用適用於 SQL 伺服器的 Amazon RDS 自訂 \(網路設定\)](#) 和 [探索建立 SQL Server 執行個體專用 Amazon RDS 自訂所需的先決條件](#)。

設定 RDS Custom for SQL Server 的先決條件

建立 RDS Custom to SQL Server 資料庫執行個體之前，請確定您的環境符合本主題中所述的需求。您也可以使用 CloudFormation 範本來設定 AWS 帳戶。如需更多資訊，請參閱 [配置 AWS CloudFormation](#)

適用於 SQL 伺服器的 RDS 自訂需要您設定下列必要條件：

- 設定建立執行個體所需的 AWS Identity and Access Management (IAM) 許可。這是向 RDS 發出 `create-db-instance` 請求所需的 AWS Identity and Access Management (IAM) 使用者或角色。
- 為 SQL Server 資料庫執行個體設定 RDS 自訂所需的先決條件資源：
 - 設定 RDS 自訂執行個體加密所需的 AWS KMS 金鑰。RDS Custom 需要在建立執行個體時使用客戶受管金鑰以進行加密。KMS 金鑰 ARN、識別碼、別名 ARN 或別名名稱會做為建立 RDS 自訂資料庫執行個體的請求中的 `kms-key-id` 參數傳遞。
 - 在 SQL Server 資料庫執行個體的 RDS 自訂中設定所需的權限。RDS Custom 會在建立時將執行個體設定檔附加至資料庫執行個體，並將其用於資料庫執行個體內的自 RDS 自訂建立要求 `custom-iam-instance-profile` 中的執行個體設定檔名稱設定為。您可以透過建立執行個體設定檔，也可 AWS Management Console 以手動建立執行個體設定檔。如需詳細資訊，請參閱 [自動建立執行個體設定檔，使 AWS Management Console](#) 及 [手動建立 IAM 角色和執行個體設定檔](#)。
 - 根據適用於 SQL 伺服器的 RDS 自訂需求設定網路設定。RDS 自訂執行個體位於您在建立執行個體時提供的子網路 (使用資料庫子網路群組設定) 中。這些子網路必須允許 RDS 自訂執行個體與 RDS 自動化所需的服務通訊。

Note

針對上述需求，請確定沒有任何服務控制原則 (SCP) 限制帳戶層級權限。如果您正在使用的帳戶是 AWS 組織的一部分，則可能會有限制帳戶層級許可的服務控制政策 (SCP)。確保 SCP 不會限制您使用下列程序所建立使用者與角色上的許可。

如需 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的[服務控制政策 \(SCP\)](#)。使用[描述-組織](#) AWS CLI 命令來檢查您的帳戶是否為組織的一部分。AWS 若要取得有關「Organ AWS izations」的更多資訊，請參閱《AWS Organizations 使用指南》中的〈Organ [AWS izations](#)〉

如需適用於 RDS Custom for SQL Server 的一般需求，請參閱 [RDS Custom for SQL Server 的一般需求](#)。

自動建立執行個體設定檔，使 AWS Management Console

RDS 自訂需要您建立並設定執行個體設定檔，以啟動任何適用於 SQL Server 資料庫執行個體的 RDS 自訂。使用在 AWS Management Console 單一步驟中建立和附加新執行個體設定檔。此選項位於 [建立資料庫]、[還原快照] 和 [還原到時間主控台] 頁面中的 [RDS 自訂安全性] 區段下提供。選擇 [建立新執行個體設定檔] 以提供執行個體設定檔名稱尾碼。會 AWS Management Console 建立具有 RDS 自動化工作所需權限的新執行個體設定檔。若要自動建立新的執行個體設定檔，您的登入 AWS Management Console 使用者必須擁有 iam:CreateInstanceProfile、iam:AddRoleToInstanceProfile、iam:CreateRole、和 iam:AttachRolePolicy 權限。

Note

此選項僅在中可用 AWS Management Console。如果您使用 CLI 或 SDK，請使用 RDS 自訂提供的 CloudFormation 範本或手動建立執行個體設定檔。如需詳細資訊，請參閱 [手動建立 IAM 角色和執行個體設定檔](#)。

步驟 1：將所需許可授予 IAM 主體

請確定您有足夠的存取權來建立 RDS 自訂執行個體。使用主控台或 CLI 建立適用於 SQL Server 資料庫執行個體的 RDS 自訂 IAM 角色或 IAM 使用者 (稱為 IAM 主體) 必須具有下列其中一項政策，才能成功建立資料庫執行個體：

- AdministratorAccess 政策
- 具有下列額外許可的 AmazonRDSFullAccess 政策：

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

```
s3:CreateBucket
s3:PutBucketPolicy
s3:PutBucketObjectLockConfiguration
s3:PutBucketVersioning
kms:CreateGrant
kms:DescribeKey
```

RDS 自訂會在建立執行個體時使用這些權限。這些權限會在您的帳戶中設定 RDS 自訂作業所需的資源。

如需 kms:CreateGrant 許可的詳細資訊，請參閱 [AWS KMS key 管理](#)。

以下範例 JSON 政策授予必要許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ValidateIamRole",
      "Effect": "Allow",
      "Action": "iam:SimulatePrincipalPolicy",
      "Resource": "*"
    },
    {
      "Sid": "CreateCloudTrail",
      "Effect": "Allow",
      "Action": [
        "cloudtrail:CreateTrail",
        "cloudtrail:StartLogging"
      ],
      "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
    },
    {
      "Sid": "CreateS3Bucket",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutBucketPolicy",
        "s3:PutBucketObjectLockConfiguration",
        "s3:PutBucketVersioning"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
    }
  ]
}
```

```
    },
    {
      "Sid": "CreateKmsGrant",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

此外，IAM 主體在 IAM 角色上需要 `iam:PassRole` 許可。這必須連接到在請求中以 `custom-iam-instance-profile` 參數傳遞的執行個體，才能建立 RDS Custom 資料庫執行個體。稍後會在 [步驟 2：設定網路、執行個體設定檔和加密](#) 中建立執行個體設定檔及其連接的角色。

Note

確保先前列出的許可不受服務控制政策 (SCP)、許可界限或與 IAM 主體相關聯之工作階段政策的限制。

步驟 2：設定網路、執行個體設定檔和加密

您可以使用下列其中一個程序來設定 IAM 執行個體設定檔角色、虛擬私有雲 (VPC) 和 AWS KMS 對稱加密金鑰：

- [配置 AWS CloudFormation](#) (建議)
- [手動設定](#)

Note

如果您的帳戶屬於任何帳戶 AWS Organizations，請確定執行個體設定檔角色所需的權限不受服務控制政策 (SCP) 的限制。

本主題中的網路組態最適用於無法公開存取的資料庫執行個體。您無法從 VPC 外部直接連線到此類資料庫執行個體。

配置 AWS CloudFormation

若要簡化設定，您可以使用 AWS CloudFormation 範本檔案建立 CloudFormation 堆疊。CloudFormation 範本會根據 RDS Custom 的需求，建立所有網路、執行個體設定檔和加密資源。

若要瞭解如何建立堆疊，請參閱 [AWS CloudFormation 使用者指南](#) 中的 [在 AWS CloudFormation 主控台上建立堆疊](#)。

如需有關如何使用 AWS CloudFormation 範本啟動適用於 SQL 伺服器的 Amazon RDS 自訂教學課程，請參閱使用 AWS 資料庫部落格中的 [AWS CloudFormation 範本開始使用 SQL 伺服器的 Amazon RDS 自訂](#)。

主題

- [所需的參數 CloudFormation](#)
- [下載 AWS CloudFormation 範本檔](#)
- [使用配置資源 CloudFormation](#)

所需的參數 CloudFormation

若要設定 RDS 自訂必要條件資源，需要下列參數 CloudFormation：

參數群組	參數名稱	預設值	描述
可用性組態	選取必要條件設定的可用性組態	Multi-AZ	指定要在單一同步備份或異地同步備份組態中為 RDS 自訂執行個體設定先決條件。如果在此組態中至少需要一個異地同步備份資料庫執行個體，則應使用異地同步備份組態。
網路組態	適用於 VPC 雲端的 IPv4 CIDR 區塊	10.0.0.0/16	指定 VPC 的 IPv4 CIDR 區塊 (或 IP 位址範圍)。此 VPC 設定為建立和使用 RDS 自訂資料庫執行個體。

參數群組	參數名稱	預設值	描述
	適用於 2 個私有子網路中的第 1 個的 IPv4 CIDR 區塊	10.0.128.0/20	為您的第一個私有子網路指定 IPv4 CIDR 區塊 (或 IP 位址範圍)。這是可以建立 RDS 自訂資料庫執行個體的兩個子網路之一。這是一個無法訪問互聯網的私有子網。
	適用於 2 個私有子網路的 IPv4 CIDR 區塊 (共 2 個)	10.0.144.0/20	指定第二個私有子網路的 IPv4 CIDR 區塊 (或 IP 位址範圍)。這是可以建立 RDS 自訂資料庫執行個體的兩個子網路之一。這是一個無法訪問互聯網的私有子網。
	適用於公有子網路的 IPv4 CIDR 區塊	10.0.0.0/20	為您的公用子網路指定 IPv4 CIDR 區塊 (或 IP 位址範圍)。這是 EC2 執行個體可以與 RDS 自訂資料庫執行個體連線的子網路之一。這是一個可以訪問互聯網的公共子網。

參數群組	參數名稱	預設值	描述
RDP 存取組態	您的來源的 IPv4 CIDR 區塊	-	指定來源的 IPv4 CIDR 區塊 (或 IP 位址範圍)。這是您在公有子網路中建立 RDP 連線至 EC2 執行個體的 IP 範圍。如果未設定，則不會設定與 EC2 執行個體的 RDP 連線。
	安裝 RDS 的 RDP 存取自訂 SQL 伺服器執行個體	否	指定是否啟用從 EC2 執行個體到適用於 SQL 伺服器執行個體的 RDS 自訂 RDP 連線。依預設，不會設定從 EC2 執行個體到資料庫執行個體的 RDP 連線。

資源建立者 CloudFormation

使用預設設定成功建立 CloudFormation 堆疊會在您的中建立下列資源 AWS 帳戶：

- 對稱加密 KMS 金鑰，用於加密由 RDS Custom 管理的資料。
- 執行個體設定檔與 IAM 角色相關聯，AmazonRDSCustomInstanceProfileRolePolicy 以提供 RDS 自訂所需的許可。如需詳細資訊，請參閱《AWS 受管原則參考指南》CustomServiceRolePolicy 中的 [AmazonRDS](#)。
- 將 CIDR 範圍指定為參數的 CloudFormation VPC。預設值為 10.0.0.0/16。
- 兩個私有子網路，具有參數中指定的 CIDR 範圍，以及 AWS 區域中兩個不同的可用區域。子網路 CIDR 的預設值為 10.0.128.0/20 和 10.0.144.0/20。
- 一個公用子網路，其中包含在參數中指定的 CIDR 範圍。子網路 CIDR 的預設值為 10.0.0.0/20。EC2 執行個體位於此子網路中，可用來連線至 RDS 自訂執行個體。
- 為 VPC 設定的 DHCP 選項，其網域名稱會解析為 Amazon Domain Name System (DNS) 伺服器。
- 與兩個私有子網路相關聯且無法存取網際網路的路由表。

- 路由表與公共子網關聯，並可以訪問互聯網。
- 與 VPC 相關聯的網際網路閘道，可允許網際網路存取公有子網路。
- 與兩個私有子網路相關聯的網路存取控制清單 (ACL)，並在 VPC 中限制存取 HTTPS 和資料庫連接埠。
- 與 RDS Custom 執行個體相關聯的 VPC 安全群組。從 EC2 執行個體安全群組 RDS 自訂資料庫連接埠和傳入資料庫連接埠所需的 AWS 服務端點，對外 HTTPS 的存取受到限制。
- 要與公有子網路中 EC2 執行個體相關聯的 VPC 安全群組。對 RDS 自訂執行個體安全性群組的輸出資料庫連接埠存取受到限制。
- 要與針 AWS 服務對 RDS 自訂所需之端點建立的 VPC 端點相關聯的 VPC 安全性群組。
- 在其中建立 RDS Custom 執行個體的資料庫子網路群組。由此範本建立的兩個私人子網路會新增至資料庫子網路群組。
- RDS 自訂所需之每個 AWS 服務端點的 VPC 端點。

將可用性組態設定為異地同步備份除了上述清單之外，還會建立下列資源：

- 允許私有子網路之間通訊的網路 ACL 規則。
- 對與 RDS 自訂執行個體關聯的 VPC 安全性群組內異地同步備份連接埠的輸入和輸出存取。
- VPC 端點到異地同步備份通訊所需的 AWS 服務端點。

此外，設定 RDP 存取組態會建立下列資源：

- 從您的來源 IP 位址設定公用子網路的 RDP 存取權：
 - 允許從來源 IP 連線至公用子網路的 RDP 規則的網路 ACL 規則。
 - 從來源 IP 到與 EC2 執行個體相關聯的 VPC 安全群組輸入 RDP 連接埠的存取權。
- 將 RDP 存取從公有子網路中的 EC2 執行個體設定為私有子網路中的 RDS 自訂執行個體：
 - 網路 ACL 規則允許從公用子網路到私有子網路的 RDP 連線。
 - 從與 EC2 執行個體相關聯的 VPC 安全群組到與 RDS 自訂執行個體相關聯的 VPC 安全群組對 RDP 連接埠進行輸入存取。

使用下列程序建立適用於 SQL 伺服器的 RDS 自訂 CloudFormation 堆疊。

下載 AWS CloudFormation 範本檔

下載範本檔案

1. 開啟 [custom-sqlserver-onboard.zip](#) 連結的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「另存連結為」。
2. 將檔案儲存並擷取到您的電腦。

使用配置資源 CloudFormation

若要使用來設定資源 CloudFormation

1. 開啟主 CloudFormation 控制台，網址為 <https://console.aws.amazon.com/cloudformation>。
2. 若要啟動「建立堆疊」精靈，請選擇 Create Stack (建立堆疊)。

Create stack (建立堆疊) 頁面隨即出現。

3. 對於 Prepare template (準備範本)，請選擇 Template is ready (範本已準備就緒)。
4. 針對 Specify template 執行下列操作：
 - a. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。
 - b. 針對選擇檔案，導覽至該處並選擇正確檔案。
5. 選擇下一步。

Specify stack details (指定堆疊詳細資訊) 頁面隨即出現。

6. 針對堆疊名稱 輸入 **rds-custom-sqlserver**。
7. 對於 Parameters (參數)，請執行下列動作：
 - a. 若要保留預設選項，請選擇 Next (下一步)。
 - b. 若要變更選項，請選擇適當的可用性組態、網路組態和 RDP 存取組態，然後選擇 [下一步]。

在變更參數之前，請先仔細閱讀每個參數的說明。

Note

如果您選擇在此 CloudFormation 堆疊中建立至少一個異地同步備份執行個體，請確定 CloudFormation 堆疊參數 [選取必要條件設定的可用性組態] 已設為 Multi-AZ。如果您將

CloudFormation 堆疊建立為單一可用區，請在建立第一個異地同步備份執行個體之前將 CloudFormation 堆疊更新為異地同步備份組態。

8. 在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。
9. 在「複查 rds-custom-sqlserver」頁面上，執行下列動作：
 - a. 針對 Capabilities (功能)，選取 I acknowledge that AWS CloudFormation might create IAM resources with custom names (確認可能使用自訂名稱建立 IAM 資源) 核取方塊。
 - b. 選擇 Create Stack (建立堆疊)。

Note

請勿直接從資源頁面更新從此 AWS CloudFormation 堆疊建立的資源。這樣可以防止您使用 AWS CloudFormation 範本將 future 的更新套用至這些資源。

CloudFormation 建立適用於 SQL 伺服器的 RDS 自訂所需的資源。如果堆疊建立失敗，請詳讀 Events 索引標籤，查看哪些資源建立失敗及其狀態原因。

主控台中此 CloudFormation 堆疊的 [輸出] 索引標籤應包含要作為建立 SQL Server 資料庫執行個體 RDS 自訂參數傳遞的所有資源的相關資訊。請務必使用 CloudFormation 針對 RDS 自訂資料庫執行個體建立的 VPC 安全群組和資料庫子網路群組。依預設，RDS 會嘗試連接預設的 VPC 安全群組，此群組可能沒有您需要的存取權。

如果您曾經 CloudFormation 建立資源，則可以略過[手動設定](#)。

更新 CloudFormation 堆疊

您也可以在建立之後更新 CloudFormation 堆疊上的某些組態。可以更新的配置如下：

- 適用於 SQL 伺服器的 RDS 自訂的可用性組態
 - 選取必要條件設定的可用性組態：更新此參數可在單一同步備份和異地同步備份組態之間切換。如果您要將此 CloudFormation 堆疊用於至少一個異地同步備份執行個體，則必須更新堆疊以選擇異地同步備份組態。
- 適用於 SQL 伺服器的 RDS 自訂 RDP 存取組態
 - 來源的 IPv4 CIDR 區塊：您可以透過更新此參數來更新來源的 IPv4 CIDR 區塊 (或 IP 位址範圍)。將此參數設定為空白，會從來源 CIDR 區塊移除 RDP 存取組態至公用子網路。

- 為 SQL 伺服器設定 RDS 自訂的 RDP 存取權：啟用或停用從 EC2 執行個體到適用於 SQL 伺服器執行個體的 RDS 自訂 RDP 連線。

刪除 CloudFormation 堆疊

您可以在刪除使用 CloudFormation 堆疊資源的所有 RDS 自訂執行個體之後刪除堆疊。RDS Custom 不會追蹤 CloudFormation 堆疊，因此當有使用堆疊資源的資料庫執行個體時，它不會封鎖堆疊的刪除。刪除堆疊時，請確定沒有使用堆疊資源的 RDS 自訂資料庫執行個體。

Note

刪除 CloudFormation 堆疊時，除了 KMS 金鑰外，堆疊所建立的所有資源都會遭到刪除。KMS 金鑰會進入暫緩刪除狀態，並在 30 天後刪除。若要保留 KMS 金鑰，請在 30 天寬限期內執行 [CancelKey 刪除](#) 作業。

手動設定

如果您選擇手動設定資源，請執行以下任務。

Note

若要簡化設定，您可以使用 AWS CloudFormation 範本檔案來建立 CloudFormation 堆疊，而不是手動設定。如需詳細資訊，請參閱 [配置 AWS CloudFormation](#)。

您也可以使用 AWS CLI 來完成本節。如果是這樣，請下載並安裝最新的 CLI。

主題

- [確定您擁有對稱式加密金鑰 AWS KMS](#)
- [手動建立 IAM 角色和執行個體設定檔](#)
- [手動設定您的 VPC](#)

確定您擁有對稱式加密金鑰 AWS KMS

RDS 自訂需要 AWS KMS key 對稱加密。當您建立適用於 SQL Server 資料庫執行個體的 RDS 自訂時，請務必提供 KMS 金鑰識別碼做為參數 `kms-key-id`。如需詳細資訊，請參閱 [建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體](#)。

您有下列選項：

- 如果您的中有現有的客戶受管 KMS 金鑰 AWS 帳戶，您可以將其與 RDS 自訂搭配使用。無需採取進一步動作。
- 若您已為不同的 RDS Custom 引擎建立客戶受管對稱加密 KMS 金鑰，您可以重複使用相同的 KMS 金鑰。無需採取進一步動作。
- 若您的帳戶中並無現有的客戶受管對稱加密 KMS 金鑰，請藉由遵循《AWS Key Management Service 開發人員指南》中的[建立金鑰](#)來建立 KMS 金鑰。
- 如果您要建立 CEV 或 RDS 自訂資料庫執行個體，而您的 KMS 金鑰位於不同的 AWS 帳戶，請務必使用 AWS CLI。您無法搭配跨帳戶 KMS 金鑰使用 AWS 主控台。

⚠ Important

RDS 自訂不支援 AWS 受管 KMS 金鑰。

確保您的對稱加密金鑰授 `kms:Decrypt` 與存取 IAM 執行個體設定檔中 AWS Identity and Access Management (IAM) 角色的存取權和 `kms:GenerateDataKey` 操作。若您的帳戶中有新的對稱加密金鑰，則不需要變更。否則，請確定對稱加密金鑰政策授予這些操作的存取權。

如需詳細資訊，請參閱 [步驟 4：設定適用於 Oracle 的 RDS 自訂 IAM](#)。

手動建立 IAM 角色和執行個體設定檔

您可以手動建立執行個體設定檔，並使用它來啟動 RDS 自訂執行個體。如果您打算在中建立執行個體 AWS Management Console，請略過本節。AWS Management Console 可讓您建立執行個體設定檔，並將其附加至 RDS 自訂資料庫執行個體。如需詳細資訊，請參閱 [自動建立執行個體設定檔，使 AWS Management Console](#)。

手動建立執行個體設定檔時，請將執行個體設定檔名稱做為 `custom-iam-instance-profile` 參數傳遞至 `create-db-instance` CLI 指令。RDS Custom 會使用與此執行個體設定檔相關聯的角色執行自動化來管理執行個體。

建立 RDS Custom for SQL Server 的 IAM 執行個體設定檔和 IAM 角色

1. 建立名為 `AWSRDSCustomSQLServerInstanceRole` 的 IAM 角色，具有可讓 Amazon EC2 擔任此角色的信任政策。

2. 將 AWS 受管理的策略新增 AmazonRDSCustomInstanceProfileRolePolicy 至 AWSRDSCustomSQLServerInstanceRole。
3. 為 RDS Custom for SQL Server 建立名為 AWSRDSCustomSQLServerInstanceProfile 的 IAM 執行個體設定檔。
4. 將 AWSRDSCustomSQLServerInstanceRole 新增至執行個體設定檔。

建立 AWSRDSCustomSQLServerInstanceRole IAM 角色

下列範例會建立 AWSRDSCustomSQLServerInstanceRole 角色。信任政策可讓 Amazon EC2 擔任角色。

```
aws iam create-role \  
  --role-name AWSRDSCustomSQLServerInstanceRole \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Action": "sts:AssumeRole",  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "ec2.amazonaws.com"  
        }  
      }  
    ]  
  }'
```

將存取原則新增至 AWSRDSCustomSQLServerInstanceRole

若要提供必要的權限，請將 AWS 受管理的原則附加 AmazonRDSCustomInstanceProfileRolePolicy 至 AWSRDSCustomSQLServerInstanceRole。AmazonRDSCustomInstanceProfileRolePolicy 允許 RDS 自訂執行個體傳送和接收訊息，以及執行各種自動化動作。

Note

確定存取政策中的許可不受與執行個體設定檔角色相關聯的 SCP 或許可界限所限制。

下列範例會將 AWS 受管理的原則附加到 `AWSRDSCustomSQLServerIamRolePolicy` 至 `AWSRDSCustomSQLServerInstanceRole` 角色。

```
aws iam attach-role-policy \  
  --role-name AWSRDSCustomSQLServerInstanceRole \  
  --policy-arn arn:aws:iam::aws:policy/AmazonRDSCustomInstanceProfileRolePolicy
```

建立您的 RDS Custom for SQL Server 執行個體設定檔

執行個體設定檔是包含單一 IAM 角色的容器。RDS Custom 會使用執行個體設定檔，將角色傳遞到執行個體。

如果您使用 AWS Management Console 為 Amazon EC2 建立角色，主控台會自動建立執行個體設定檔，並為其提供與建立角色時相同的名稱。如下所示建立您的執行個體設定檔，將它命名為 `AWSRDSCustomSQLServerInstanceProfile`。

```
aws iam create-instance-profile \  
  --instance-profile-name AWSRDSCustomSQLServerInstanceProfile
```

新增 `AWSRDSCustomSQLServerInstanceRole` 至您的 RDS 自訂 SQL 伺服器執行個體設定檔

將 `AWSRDSCustomInstanceRoleForRdsCustomInstance` 角色新增至先前建立的 `AWSRDSCustomSQLServerInstanceProfile` 設定檔。

```
aws iam add-role-to-instance-profile \  
  --instance-profile-name AWSRDSCustomSQLServerInstanceProfile \  
  --role-name AWSRDSCustomSQLServerInstanceRole
```

手動設定您的 VPC

根據 Amazon VPC 服務，您的 RDS Custom 資料庫執行個體位於虛擬私有雲端 (VPC)，就像是 Amazon EC2 執行個體或 Amazon RDS 執行個體。您可以提供和設定您自己的 VPC。因此，您可以完全控制您的執行個體網絡設定。

RDS Custom 會將通訊從您的資料庫執行個體傳送到其他 AWS 服務。請確定可從建立 RDS 自訂資料庫執行個體的子網路存取下列服務：

- Amazon CloudWatch

- Amazon CloudWatch 日誌
- Amazon CloudWatch 活動
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

建立異地同步備份部署

- Amazon Simple Queue Service

如果 RDS Custom 無法與必要的服務通訊，則會發佈下列事件：

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS services. Make sure port 443 (HTTPS) allows outbound connections, and try again. "Failed to connect to the following services: s3 events"
```

若要避免incompatible-network錯誤，請確定 RDS 自訂資料庫執行個體之間的通訊涉及 VPC 元件，並 AWS 服務 符合下列需求：

- 資料庫執行個體會連接埠 443 上對其他 AWS 服務進行傳出連線。
- VPC 允許對來自您 RDS Custom 資料庫執行個體的請求進行傳入回應。
- RDS Custom 可以正確解析每個 AWS 服務的端點網域名稱。

若您已經為不同的 RDS Custom 資料庫引擎設定 VPC，您可重複使用該 VPC 並略過此程序。

主題

- [設定您的 VPC 安全群組](#)
- [設定相依項目的端點 AWS 服務](#)
- [設定個體中繼資料服務](#)

設定您的 VPC 安全群組

安全群組會做為 VPC 執行個體的虛擬防火牆，控制傳入及傳出流量。RDS 自訂資料庫執行個體具有連接到其網路介面的安全群組，以保護執行個體。請確定您的安全性群組允許 RDS 自訂和其他 AWS 服務透過 HTTPS 之間的流量。您可以在執行個體建立要求中傳遞這個安全性群組做為 `vpc-security-group-ids` 參數。

若要設定 RDS Custom 的安全群組

1. 登入 AWS Management Console 並開啟 Amazon VPC 主控台，網址為 <https://console.aws.amazon.com/vpc>。
2. 允許 RDS Custom 使用預設安全群組，或建立您自己的安全群組。

如需詳細說明，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

3. 請確定您的安全群組允許連接埠 443 上的傳出連線。RDS Custom 需要此連接埠與相依 AWS 服務通訊。
4. 如果您具備私有 VPC 並使用 VPC 端點，請確定與資料庫執行個體相關聯的安全群組允許連接埠 443 上傳出至 VPC 端點的連線。此外還要確定與 VPC 端點相關聯的安全群組允許連接埠 443 上從資料庫執行個體傳入的連線。

如果不允許傳入的連線，則 RDS Custom 執行個體無法連線到 AWS Systems Manager 和 Amazon EC2 端點。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 [建立 Virtual Private Cloud 端點](#)。

5. 對於 SQL Server 異地同步備份執行個體的 RDS 自訂，請確定與資料庫執行個體關聯的安全群組允許連接埠 1120 上的輸入和輸出連線至此安全群組本身。這對於異地同步備份 RDS 自訂 SQL Server 資料庫執行個體上的對等主機連線是必要的。

如需安全群組的詳細資訊，請參閱《Amazon VPC 開發人員指南》中的 [您的 VPC 的安全群組](#)。

設定相依項目的端點 AWS 服務

建議您使用下列指示將每個服務的端點新增至 VPC。但是，您可以使用任何可讓 VPC 與 AWS 服務端點通訊的解決方案。例如，您可以使用網路位址轉譯 (NAT) 或 AWS Direct Connect。

設定 RDS 自訂可 AWS 服務 使用的端點

1. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在導覽列上，使用區域選取器來選擇 AWS 區域。

3. 在導覽窗格中選擇端點。在主窗格中，選擇 Create Endpoint (建立端點)。
4. 對於 Service category (服務類別)，選擇 AWS 服務。
5. 針對 Service Name (服務名稱)，選擇表格中顯示的端點。
6. 在 VPC 中，選擇您的 VPC。
7. 在 Subnets (子網路) 中，請從每個可用區域選擇要包含的子網路。

VPC 端點可跨越多個可用區域。AWS 在您選擇的每個子網路中，為 VPC 端點建立 elastic network interface。每個網路界面具有網域名稱系統 (DNS) 主機名稱和私有 IP 地址。

8. 針對 Security group (安全群組)，選擇或建立安全群組。

您可以使用安全群組來控制對端點的存取，就像使用防火牆一樣。確定安全群組允許來自資料庫執行個體的連接埠 443 上的輸入連線。如需安全群組的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [VPC 的安全群組](#)。

9. 或者，您可以將政策連接至 VPC 端點。端點策略可以控制對您 AWS 服務要連線的對象的存取。預設政策允許所有請求通過端點。如果您使用的是自訂政策，請確定該政策中允許來自資料庫執行個體的請求。
10. 選擇建立端點。

下表說明如何尋找您的 VPC 傳出通訊所需的端點清單。

服務	端點格式	備註和連結
AWS Systems Manager	使用下列端點格式： <ul style="list-style-type: none"> • <code>ssm.<i>region</i>.amazonaws.com</code> • <code>ssmmessages.<i>region</i>.amazonaws.com</code> 	如需每個區域的端點清單，請參閱 Amazon Web Services 一般參考中的 AWS Systems Manager 端點和配額 。
AWS Secrets Manager	使用端點格式 <code>secretsmanager.<i>region</i>.amazonaws.com</code> 。	如需每個區域的端點清單，請參閱 Amazon Web Services 一般參考中的 AWS Secrets Manager 端點和配額 。
Amazon CloudWatch	使用下列端點格式：	如需每個區域中的端點清單，請參閱：

服務	端點格式	備註和連結
	<ul style="list-style-type: none"> 對於 CloudWatch 指標，請使用 <code>monitoring.<i>region</i>.amazonaws.com</code> 對於 CloudWatch 事件，使用 <code>events.<i>region</i>.amazonaws.com</code> 對於 CloudWatch 記錄檔，請使用 <code>logs.<i>region</i>.amazonaws.com</code> 	<ul style="list-style-type: none"> Amazon CloudWatch 端點和配額 Amazon Web Services 一般參考 Amazon CloudWatch 日誌端點和配額 Amazon Web Services 一般參考 Amazon CloudWatch 活動端點和配額 Amazon Web Services 一般參考
Amazon EC2	使用下列端點格式： <ul style="list-style-type: none"> <code>ec2.<i>region</i>.amazonaws.com</code> <code>ec2messages.<i>region</i>.amazonaws.com</code> 	如需每個區域中的端點清單，請參閱 Amazon Web Services 一般參考中的 Amazon Elastic Compute Cloud 端點和配額 。
Amazon S3	使用端點格式 <code>s3.<i>region</i>.amazonaws.com</code> 。	如需每個區域中的端點清單，請參閱 Amazon Web Services 一般參考中的 Amazon Simple Storage Service 端點和配額 。 若要進一步了解 Amazon S3 的閘道端點，請參閱《Amazon VPC 開發人員指南》中的 Amazon S3 的端點 。 若要了解如何建立存取點，請參閱《Amazon VPC 開發人員指南》中的 建立存取點 。 若要瞭解如何為 Amazon S3 建立閘道端點，請參閱 閘道 VPC 端點 。

服務	端點格式	備註和連結
Amazon Simple Queue Service	使用端點格式 <code>sqs.<i>region</i>.amazonaws.com</code>	如需每個區域中的端點清單，請參閱 Amazon 簡單佇列服務端點和配額 。

設定個體中繼資料服務

請確定您的執行個體可以執行下列操作：

- 使用執行個體中繼資料服務版本 2 (IMDSv2) 存取執行個體中繼資料服務。
- 允許透過連接埠 80 (HTTP) 與 IMDS 連結 IP 地址進行傳出通訊。
- 從 `http://169.254.169.254` (IMDSv2 連結) 要求執行個體中繼資料。

如需詳細資訊，請參閱 Amazon EC2 使用者 [指南中的使用 IMDSv2](#)。

跨執行個體限制

當您依照上述步驟建立執行個體設定檔時，它會使用 AWS 受管理的原則，AmazonRDSCustomInstanceProfileRolePolicy 為 RDS Custom 提供必要的權限，讓執行個體管理和監控自動化。受管理的原則可確保權限僅允許存取 RDS Custom 執行自動化所需的資源。我們建議您使用受管政策來支援新功能，並解決安全性需求，這些功能會自動套用至現有的執行個體設定檔，而無需手動。如需詳細資訊，請參閱 [AWS 受管政策：亞馬遜 R DSCustom InstanceProfile RolePolicy](#)

受 AmazonRDSCustomInstanceProfileRolePolicy 管理政策會限制執行個體設定檔具有跨帳戶存取權，但可能允許存取相同帳戶內 RDS Custom 執行個體之間的某些 RDS Custom 受管資源。根據您的需求，您可以使用權限界限進一步限制跨執行個體存取。權限界限定義了以身分識別為基礎的原則可以授與給實體的最大權限，但不會自行授與權限。如需詳細資訊，請參閱 [使用界限評估有效權限](#)。

例如，下列原則會限制執行個體設定檔角色存取特定金 AWS KMS 鑰，並限制所有使用不同 AWS KMS 金鑰的執行個體對 RDS Custom 受管資源的存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyOtherKmsKeyAccess",
```

```
        "Effect": "Deny",
        "Action": "kms:*",
        "NotResource": "arn:aws:kms:region:acct_id:key/KMS_key_ID"
    },
    {
        "Sid": "NoBoundarySetByDefault",
        "Effect": "Allow",
        "Action": "*",
        "Resource": "*"
    }
]
}
```

Note

請確定權限界限不會封鎖AmazonRDSCustomInstanceProfileRolePolicy授與 RDS 自訂的任何權限。

透過 RDS Custom for SQL Server 使用自有媒體

RDS Custom for SQL Server 支援兩種授權模式：已包含授權 (LI) 和使用自有媒體 (BYOM)。

透過 BYOM，您可以執行下列操作：

1. 在 AWS EC2 視窗 AMI 上提供並安裝您自己的 Microsoft SQL Server 二進位檔案，並提供支援的累積更新 (CU)。
2. 將 AMI 另存為黃金映像，可將其用於建立自訂引擎版本 (CEV) 的範本。
3. 從您的黃金映像建立 CEV。
4. 使用 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。

然後，Amazon RDS 會為您管理您的資料庫執行個體。

Note

如果您也有 SQL Server 資料庫執行個體的已包含授權 (LI) RDS Custom，則無法將此資料庫執行個體中的 SQL Server 軟體與 BYOM 搭配使用。您必須對 BYOM 提供自己的 SQL Server 二進位檔案。

RDS Custom for SQL Server 的 BYOM 要求

適用於 RDS Custom for SQL Server 引擎版本的相同一般需求也適用於 BYOM。如需詳細資訊，請參閱 [RDS Custom for SQL Server CEV 的要求](#)。

使用 BYOM 時，請務必滿足下列額外要求：

- 使用下列其中一個受支援的版本：SQL Server 2022 或 2019 企業版、標準版或開發人員版。
- 將 SQL Server 系統管理員 (SA) 伺服器角色權限授予 NT AUTHORITY\SYSTEM。
- 一律將 Windows Server 作業系統設定為 UTC 時間。

Amazon EC2 Windows 執行個體預設為 UTC 時區。如需有關檢視和變更 Windows 執行個體時間的詳細資訊，請參閱 [設定 Windows 執行個體的時間](#)。

- 開啟 TCP 埠 1433 和 UDP 連接埠 1434 以允許 SSM 連線。

RDS Custom for SQL Server 的限制

適用於 RDS Custom for SQL Server 的相同一般限制也適用於 BYOM。如需詳細資訊，請參閱 [Amazon RDS Custom for SQL Server 的需求和限制](#)。

使用 BYOM 時，適用下列額外限制：

- 僅支援預設 SQL Server 執行個體 (MSSQLSERVER)。不支援具名的 SQL Server 執行個體。RDS Custom for SQL Server 只會偵測並監控預設 SQL Server 執行個體。
- 每個 AMI 僅支援安裝一次 SQL Server。不支援不同 SQL Server 版本的多重安裝。
- BYOM 不支援 SQL Server 網頁版本。
- BYOM 不支援 SQL Server 版本的評估版本。當您安裝 SQL Server 時，請勿選擇使用試用版本的核取方塊。
- 功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需詳細資訊，請參閱 [RDS Custom for SQL Server CEV 的區域可用性](#) 及 [RDS Custom for SQL Server CEV 的版本支援](#)。

使用 BYOM 建立 RDS Custom for SQL Server 資料庫執行個體

若要使用 BYOM 準備和建立 RDS Custom for SQL Server 資料庫執行個體，請參閱 [使用自有媒體 \(BYOM\) 準備 CEV](#)。

使用 RDS Custom for SQL Server 的自訂引擎版本

RDS Custom for SQL Server 的自訂引擎版本 (AMI) 是包含 Microsoft SQL Server 的 Amazon Machine Image (AMI)。

CEV 工作流程的基本步驟如下：

1. 選擇一個 AWS EC2 Windows AMI 用作 CEV 的基礎映像。您可以選擇使用預先安裝型 Microsoft SQL Server 或使用自有媒體來安裝 SQL Server。
2. 在作業系統 (OS) 上安裝其他軟體，並自訂作業系統和 SQL Server 的組態，以符合您的企業需求。
3. 將 AMI 儲存為黃金映像
4. 從您的黃金映像建立自訂引擎版本 (CEV)。
5. 使用 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。

然後，Amazon RDS 會為您管理這些資料庫執行個體。

CEV 可讓您維護作業系統和資料庫的偏好基準組態。使用 CEV 可確保主機組態 (例如任何第三方代理程式安裝或其他作業系統自訂) 會保存在 RDS Custom for SQL Server 資料庫執行個體上。有了 CEV，您可以使用相同組態快速部署 RDS Custom for SQL Server 資料庫執行個體的機群。

主題

- [準備建立 RDS Custom for SQL Server 的 CEV](#)
- [建立 RDS Custom for SQL Server 的 CEV](#)
- [修改 RDS Custom for SQL Server 的 CEV](#)
- [檢視 Amazon RDS Custom for SQL Server 的 CEV 詳細資訊](#)
- [刪除 RDS Custom for SQL Server 的 CEV](#)

準備建立 RDS Custom for SQL Server 的 CEV

您可以使用 Amazon Machine Image (AMI) 建立 CEV，該映像含有預先安裝的已包含授權 (LI) Microsoft SQL Server，或者使用安裝了自有 SQL Server 安裝媒體 (BYOM) 的 AMI。

內容

- [使用自有媒體 \(BYOM\) 準備 CEV](#)

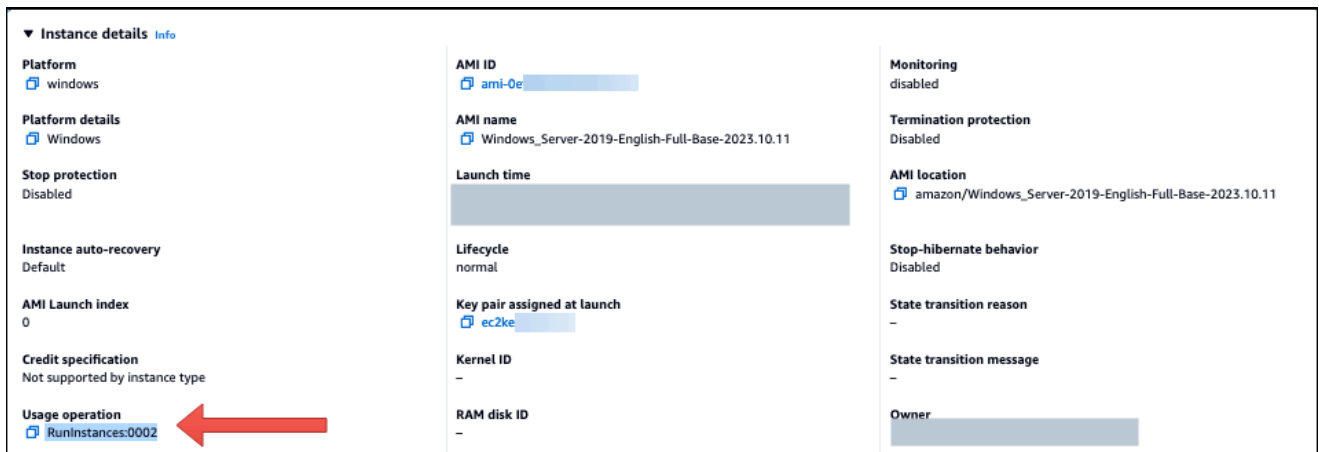
- [使用預先安裝型 SQL Server \(LI\) 準備 CEV](#)
- [RDS Custom for SQL Server CEV 的區域可用性](#)
- [RDS Custom for SQL Server CEV 的版本支援](#)
- [RDS Custom for SQL Server CEV 的要求](#)
- [RDS Custom for SQL Server CEV 的限制](#)

使用自有媒體 (BYOM) 準備 CEV

以下步驟使用 AMI 與視窗伺服器 2019 基礎作為一個例子。

若要使用 BYOM 建立 CEV

1. 在 Amazon EC2 主控台上，選擇啟動執行個體。
2. 在「名稱」中，輸入例證的名稱。
3. 在快速入門下，選擇視窗。
4. 選擇 Microsoft 視窗伺服器 2019 基地。
5. 選擇適當的執行個體類型、key pair、網路和儲存設定，然後啟動執行個體。
6. 啟動或建立 EC2 執行個體之後，請確定已從步驟 4 選取正確的 Windows AMI：
 - a. 在亞馬遜 EC2 主控台中選取 Amazon EC2 執行個體。
 - b. 在「詳細資料」區段中，檢查「使用」作業，並確定其設定為:RunInstances0002。



7. 登入 EC2 執行個體，然後將 SQL Server 安裝媒體複製到該執行個體。

Note

如果您正在使用 SQL Server 開發人員版本構建 CEV，您可能需要使用您的 [Microsoft 視覺工作室訂閱](#) 獲取安裝媒體。

8. 安裝 SQL Server。請確定執行下列操作：
 - a. 審查 [RDS Custom for SQL Server 的 BYOM 要求](#) 和 [RDS Custom for SQL Server CEV 的版本支援](#)。
 - b. 將執行個體根目錄設定為預設值 C:\Program Files\Microsoft SQL Server\。請勿變更此目錄。
 - c. 將 SQL Server 資料庫引擎帳戶名稱設定為 NT Service\MSSQLSERVER 或 NT AUTHORITY\NETWORK SERVICE。
 - d. 將 SQL Server 啟動模式設定為手動。
 - e. 將 SQL Server 驗證模式選擇為混合。
 - f. 保留預設資料目錄和 TempDB 位置的現有設定。
9. 將 SQL Server 系統管理員 (SA) 伺服器角色權限授予 NT AUTHORITY\SYSTEM：

```
USE [master]
GO
EXEC master..sp_addsrvrolemember @loginame = N'NT AUTHORITY\SYSTEM' , @rolename =
  N'sysadmin'
GO
```

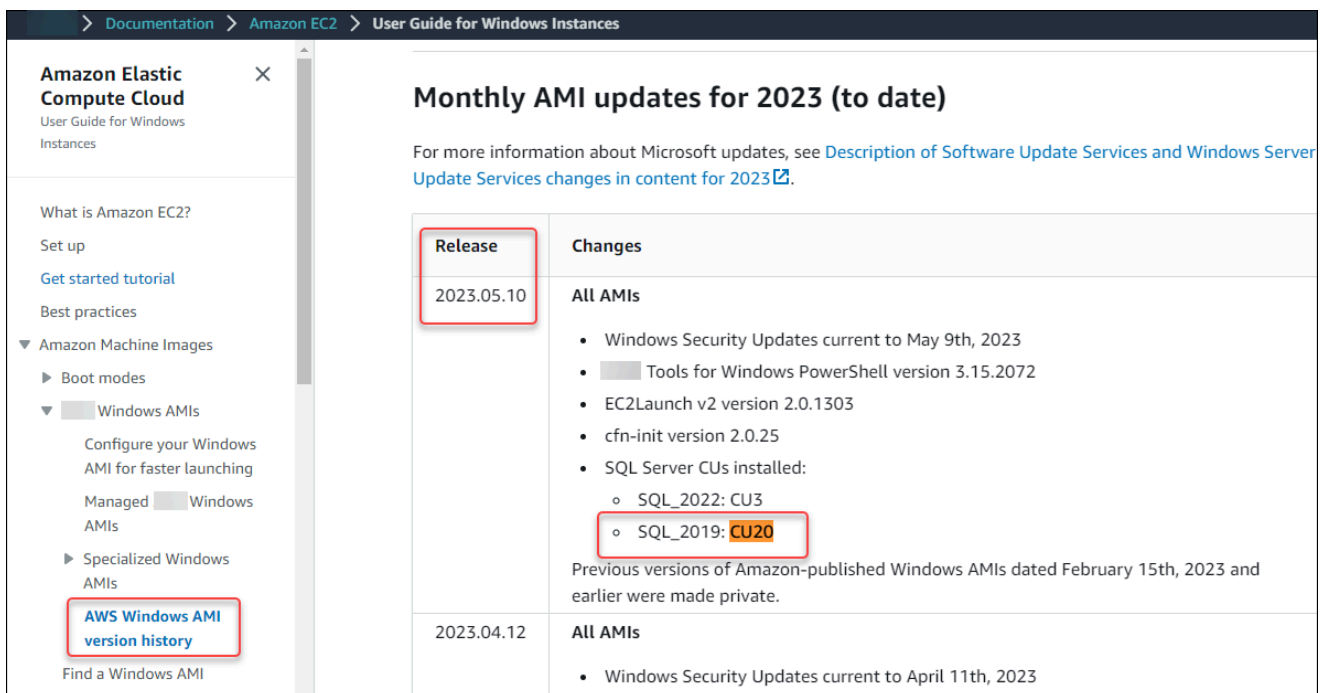
10. 安裝其他軟體或自訂作業系統和資料庫組態以符合您的需求。
11. 在 EC2 執行個體上執行 Sysprep。如需詳細資訊，請參閱 [使用 Sysprep 建立標準化 Amazon Machine Image \(AMI\)](#)。
12. 儲存含有已安裝 SQL Server 版本、其他軟體和自訂項目的 AMI。這將是你的黃金映像。
13. 透過提供您所建立映像的 AMI ID 來建立新的 CEV。如需詳細步驟，請參閱 [建立 RDS Custom for SQL Server 的 CEV](#)。
14. 使用 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。如需詳細步驟，請參閱 [從 CEV 建立 RDS Custom for SQL Server 資料庫執行個體](#)。

使用預先安裝型 SQL Server (LI) 準備 CEV

下列步驟使用預先安裝了 Microsoft SQL Server (LI) 的 AMI 建立 CEV，並以使用 SQL Server CU20 (版本號碼 2023.05.10) 的 AMI 作為範例。建立 CEV 時，請選擇具有最新版本號碼的 AMI。如此可確保您使用受支援的 Windows Server 和 SQL Server 版本，以及最新的累積更新 (CU)。

使用預先安裝型 Microsoft SQL Server (LI) 建立 CEV 的方法如下

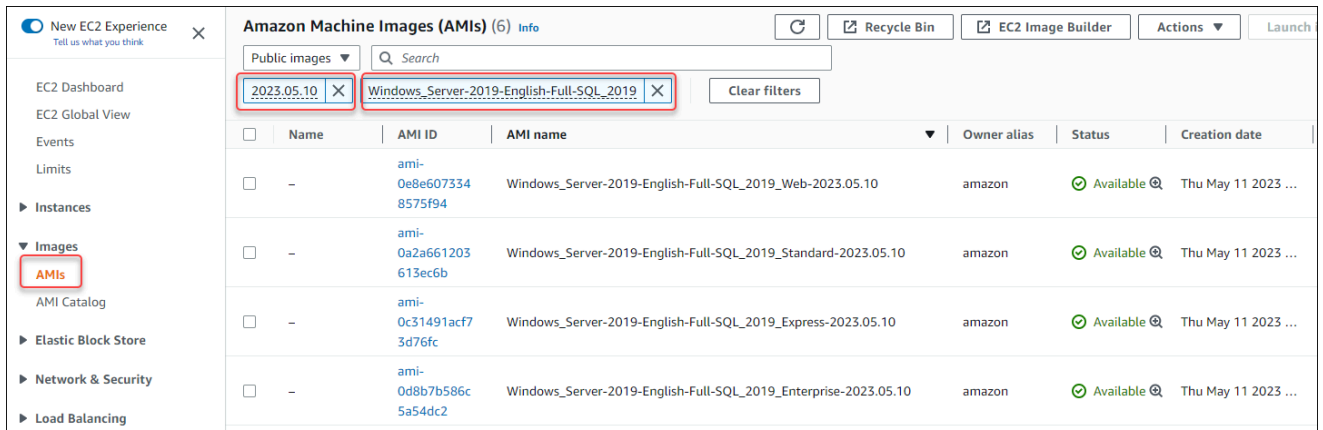
1. 選擇最新可用的 AWS EC2 視窗 Amazon 機器映像 (AMI) 包括許可證 (LI) Microsoft 視窗服務器和 SQL 服務器。
 - a. 在 [Windows AMI 版本歷史記錄](#) 內搜尋 CU20。
 - b. 請記下版本編號。對於 SQL Server 2019 CU20，版本編號為 2023.05.10。



The screenshot shows the 'Monthly AMI updates for 2023 (to date)' page. The left sidebar has 'AWS Windows AMI version history' highlighted. The main content area shows a table of updates:

Release	Changes
2023.05.10	All AMIs <ul style="list-style-type: none">Windows Security Updates current to May 9th, 2023Tools for Windows PowerShell version 3.15.2072EC2Launch v2 version 2.0.1303cfn-init version 2.0.25SQL Server CUs installed:<ul style="list-style-type: none">SQL_2022: CU3SQL_2019: CU20 <p>Previous versions of Amazon-published Windows AMIs dated February 15th, 2023 and earlier were made private.</p>
2023.04.12	All AMIs <ul style="list-style-type: none">Windows Security Updates current to April 11th, 2023

- c. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
- d. 在 Amazon EC2 主控台的左導覽窗格中，選擇 Images (映像)，然後選擇 AMIs。
- e. 選擇公有映像。
- f. 將 2023.05.10 輸入搜尋方塊中。會出現 AMI 清單。
- g. 將 Windows_Server-2019-English-Full-SQL_2019 輸入搜尋方塊中以篩選結果。下列結果應該出現。



h. 選擇具有您要使用之 SQL Server 版本的 AMI。

- 從您選擇的 AMI 建立或啟動 EC2 執行個體。
- 登入 EC2 執行個體並安裝其他軟體，或自訂作業系統和資料庫組態以符合您的需求。
- 在 EC2 執行個體上執行 Sysprep。如需使用 Sysprep 準備 AMI 的詳細資訊，請參閱[使用 Sysprep 建立標準化 Amazon Machine Image \(AMI\)](#)。
- 儲存含有已安裝 SQL Server 版本、其他軟體和自訂項目的 AMI。這將是你的黃金映像。
- 透過提供您所建立映像的 AMI ID 來建立新的 CEV。如需建立 CEV 的詳細步驟，請參閱[建立 RDS Custom for SQL Server 的 CEV](#)。
- 使用 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。如需詳細步驟，請參閱[從 CEV 建立 RDS Custom for SQL Server 資料庫執行個體](#)。

RDS Custom for SQL Server CEV 的區域可用性

適用於 SQL 伺服器的 RDS 自訂引擎版本 (CEV) 支援可在下列項 AWS 區域目中使用：

- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (奧勒岡)
- 亞太區域 (孟買)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太區域 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)

- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- 歐洲 (斯德哥爾摩)
- 南美洲 (聖保羅)

RDS Custom for SQL Server CEV 的版本支援

適用於 SQL 伺服器的 RDS 建立 CEV 自訂支援下列 AWS EC2 視窗 AMI：

- 對於使用預先安裝媒體的 CEV，AWS EC2 視窗 AMI 包含授權 (LI) Microsoft 視窗伺服器 2019 年 (作業系統) 和 SQL 伺服器 2022 或 2019
- 對於使用自攜媒體 (BYOM) 的執行長官，AWS EC2 視窗 AMI 與 Microsoft 視窗伺服器 2019 (作業系統)

下列作業系統和資料庫版本支援 RDS Custom for SQL Server 的 CEV 建立程序：

- 對於使用預先安裝媒體的 CEV：
 - 含 CU9 的 SQL 伺服器 2022，適用於企業版、標準版和網頁版
 - SQL 伺服器 2019 搭配 CU17、CU18、CU20 和 CU24，適用於企業版、標準版和網頁版
- 對於使用自攜媒體 (BYOM) 的執行長官：
 - 含 CU9 的 SQL 伺服器 2022，適用於企業版、標準版和開發人員版
 - SQL 伺服器 2019 搭配 CU17、CU18、CU20 和 CU24，適用於企業版、標準版和開發人員版
- 對於使用預先安裝媒體或自攜媒體 (BYOM) 的 CEV，Windows 伺服器 2019 是唯一支援的作業系統。

如需詳細資訊，請參閱 [AWS Windows AMI 版本歷程記錄](#)。

RDS Custom for SQL Server CEV 的要求

下列需求適用於建立 RDS for RDS Custom for SQL 的 CEV：

- 用來建立 CEV 的 AMI 必須以 RDS Custom for SQL Server 所支援的作業系統和資料庫組態為基礎。如需所支援組態的詳細資訊，請參閱 [Amazon RDS Custom for SQL Server 的需求和限制](#)。
- CEV 必須具有唯一名稱。您無法建立與現有 CEV 同名的 CEV。

- 您必須使用 SQL Server 主要版本 + 次要版本 + 自訂字串的命名模式來命名 CEV。主要版本 + 次要版本必須符合 AMI 隨附的 SQL Server 版本。例如，您可以將具有 SQL Server 2019 CU17 的 AMI 命名為 15.00.4249.2.my_cevtest。
- 您必須使用 Sysprep 準備 AMI。如需使用 Sysprep 準備 AMI 的詳細資訊，請參閱[使用 Sysprep 建立標準化 Amazon Machine Image \(AMI\)](#)。
- 您負責維護 AMI 的生命週期。從 CEV 建立的 RDS Custom for SQL Server 資料庫執行個體不會存放 AMI 的複本。它會維護一個指標，指向您用來建立 CEV 的 AMI。AMI 必須存在，RDS Custom for SQL Server 資料庫執行個體才能保持可操作狀態。

RDS Custom for SQL Server CEV 的限制

下列限制適用於具有 RDS Custom for SQL Server 的自訂引擎版本：

- 若有資源 (例如資料庫執行個體或資料庫快照) 與 CEV 相關聯，則無法刪除該 CEV。
- 若要建立 RDS Custom for SQL Server 資料庫執行個體，CEV 必須具有 pending-validation、available、failed 或 validating 的狀態。如果 CEV 狀態為 incompatible-image-configuration，則無法使用 CEV 建立 RDS Custom for SQL Server 資料庫執行個體。
- 若要修改 RDS Custom for SQL Server 資料庫執行個體，以使用新的 CEV，CEV 必須具有 available 的狀態。
- 無法從現有 RDS Custom for SQL Server 資料庫執行個體建立 AMI 或 CEV。
- 您無法修改現有 CEV 以使用不同的 AMI。不過，您可以修改 RDS Custom for SQL Server 資料庫執行個體，以使用不同的 CEV。如需詳細資訊，請參閱[修改 RDS Custom for SQL Server 資料庫執行個體](#)。
- 不支援 CEV 的跨區域複本。
- 不支援 CEV 的跨帳戶複本。
- CEV 刪除後就無法還原或復原。不過，您可以從相同 AMI 建立新的 CEV。
- RDS Custom for SQL Server 資料庫執行個體會將您的 SQL Server 資料庫檔案存放在 D:\ 磁碟機中。與 CEV 相關聯的 AMI 應該將 Microsoft SQL Server 系統資料庫檔案儲存在 C:\ 磁碟機中。
- RDS Custom for SQL Server 資料庫執行個體會保留您對 SQL Server 所做的組態變更。不會保留對執行中 RDS Custom for SQL Server 資料庫執行個體 (從 CEV 建立) 上作業系統所做的任何組態變更。如果您需要對作業系統進行永久組態變更，並將其保留為新的基準組態，請建立新 CEV 並修改資料庫執行個體以使用新的 CEV。

⚠ Important

修改 RDS Custom for SQL Server 資料庫執行個體以使用新的 CEV，這是離線操作。您可以立即執行修改，或排程在每週維護時段期間進行修改。

- 修改 CEV 後，Amazon RDS 不會將這些修改推送到任何相關聯的 RDS Custom for SQL Server 資料庫執行個體。您必須修改每個 RDS Custom for SQL Server 資料庫執行個體，才能使用新的或更新的 CEV。如需詳細資訊，請參閱 [修改 RDS Custom for SQL Server 資料庫執行個體](#)。

⚠ Important

如果刪除 CEV 使用的 AMI，則可能需要進行主機取代的任何修改 (例如擴展運算) 都將失敗。然後，RDS Custom for SQL Server 資料庫執行個體會放置在 RDS 支援周邊之外。我們建議您避免刪除與 CEV 相關聯的任何 AMI。

建立 RDS Custom for SQL Server 的 CEV

您可以使用 AWS Management Console 或 AWS CLI 建立自訂引擎版本 (CEV)。然後，您可以使用 CEV 建立 RDS Custom for SQL Server 資料庫執行個體。

請確定 Amazon Machine Image (AMI) 位於與 CEV 相同的 AWS 帳戶和區域中。否則，建立 CEV 的程序會失敗。

如需詳細資訊，請參閱 [建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體](#)。

⚠ Important

對於使用預先安裝型 SQL Server 建立的 AMI 與使用自有媒體 (BYOM) 建立的 AMI，兩者建立 CEV 的步驟皆為相同。

主控台

若要建立 CEV

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。

Custom engine versions (自訂引擎版本) 頁面會顯示目前存在的所有 CEV。如果您尚未建立任何 CEV，則資料表為空白。

3. 選擇 Create custom engine version (建立自訂引擎版本)。
4. 針對 Engine type (引擎類型) 中，選擇 Microsoft SQL Server。
5. 針對版本，選擇您要使用的資料庫引擎版本。
6. 針對 Major version (主要版本)，選擇您 AMI 上安裝的主要引擎版本。
7. 在 Version details (版本詳細資訊) 的 Custom engine version name (自訂引擎版本名稱) 中輸入有效名稱。

名稱格式為 *major-engine-version.minor-engine-version.customized_string*。您可以使用 1–50 個英數字元、字元、底線、破折號和句點。例如，您輸入名稱 **15.00.4249.2.my_cevtest**。

選擇性輸入 CEV 的描述。

8. 針對 Installation Media (安裝媒體)，瀏覽或輸入您要從中建立 CEV 的 AMI ID。
9. 在 Tags (標籤) 區段中，新增任何標籤以識別 CEV。
10. 選擇 Create custom engine version (建立自訂引擎版本)。

Custom engine versions (自訂引擎版本) 頁面隨即出現。即會顯示您的 CEV，狀態為 pending-validation

AWS CLI

若要使用建立 CEVAWS CLI，請執行 [create-custom-db-engine-version](#) 命令。

下列選項是必要的：

- --engine
- --engine-version
- --image-id

您還可以指定下列選項：

- --description
- --region

- `--tags`

以下範例會建立名為 `15.00.4249.2.my_cevtest` 的 CEV。請確保 CEV 的名稱以主要引擎版本號碼為開頭。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-custom-db-engine-version \  
  --engine custom-sqlserver-ee \  
  --engine-version 15.00.4249.2.my_cevtest \  
  --image-id ami-0r93cx31t5r596482 \  
  --description "Custom SQL Server EE 15.00.4249.2 cev test"
```

下列部分輸出顯示引擎、參數群組及其他資訊。

```
"DBEngineVersions": [  
  {  
    "Engine": "custom-sqlserver-ee",  
    "MajorEngineVersion": "15.00",  
    "EngineVersion": "15.00.4249.2.my_cevtest",  
    "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for RDS Custom for  
SQL Server",  
    "DBEngineVersionArn": "arn:aws:rds:us-east-1:<my-account-id>:cev:custom-sqlserver-  
ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",  
    "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",  
  
    "Image": [  
      "ImageId": "ami-0r93cx31t5r596482",  
      "Status": "pending-validation"  
    ],  
    "CreateTime": "2022-11-20T19:30:01.831000+00:00",  
    "SupportsLogExportsToCloudwatchLogs": false,  
    "SupportsReadReplica": false,  
    "Status": "pending-validation",  
    "SupportsParallelQuery": false,  
    "SupportsGlobalDatabases": false,  
    "TagList": []  
  }  
]
```

如果建立 CEV 的程序失敗，RDS Custom for SQL Server 會發出 RDS-EVENT-0198，訊息為 Creation failed for custom engine version *major-engine-version.cev_name*。該訊息包含有關故障的詳細資訊，例如事件列印遺失的檔案。若要尋找 CEV 建立問題的疑難排解想法，請參閱 [針對 RDS Custom for SQL Server 的 CEV 錯誤進行疑難排解](#)。

從 CEV 建立 RDS Custom for SQL Server 資料庫執行個體

成功建立 CEV 後，CEV status (CEV 狀態) 會顯示 pending-validation。您現在可以使用 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。若要從 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體，請參閱 [建立 RDS Custom for SQL Server 資料庫執行個體](#)。

CEV 的生命週期

CEV 生命週期包含下列狀態。

CEV 狀態	描述	故障診斷建議
pending-validation	已建立 CEV，且其正在等待相關聯 AMI 的驗證。CEV 會保留在 pending-validation 中，直到從中建立了 RDS Custom for SQL Server 資料庫執行個體為止。	如果沒有現有的任務，請從 CEV 建立新的 RDS Custom for SQL Server 資料庫執行個體。建立 RDS Custom for SQL Server 資料庫執行個體時，系統會嘗試驗證 CEV 的相關聯 AMI。
validating	基於新 CEV 的 RDS Custom for SQL Server 資料庫執行個體的建立任務正在進行中。建立 RDS Custom for SQL Server 資料庫執行個體時，系統	等待現有 RDS Custom for SQL Server 資料庫執行個體的建立任務完成。您可以使用 RDS EVENTS 主控台，檢閱詳細的事件訊息以進行疑難排解。

CEV 狀態	描述	故障診斷建議
	會嘗試驗證 CEV 的相關聯 AMI。	
available	已成功驗證 CEV。一旦成功從 CEV 建立了 RDS Custom for SQL Server 資料庫執行個體，該 CEV 就會進入 available 狀態。	CEV 不需要任何額外的驗證。它可以用來建立其他 RDS Custom for SQL Server 資料庫執行個體，或修改現有的執行個體。
inactive	CEV 已修改為非作用中狀態。	您無法使用此 CEV 建立或升級 RDS Custom 資料庫執行個體。您也無法還原資料庫快照，以使用此 CEV 建立新的 RDS Custom 資料庫執行個體。若要取得如何將狀態變更為 ACTIVE 的資訊，請參閱 修改 RDS Custom for SQL Server 的 CEV 。
failed	在此 CEV 可以驗證 AMI 之前，無法對其進行建立資料庫執行個體步驟。或者，CEV 使用的基礎 AMI 未處於可用狀態。	針對根本原因進行疑難排解，找出系統為何無法建立資料庫執行個體。檢視詳細的錯誤訊息，然後嘗試再次建立新的資料庫執行個體。確保 CEV 使用的基礎 AMI 處於可用狀態。

CEV 狀態	描述	故障診斷建議
incompatible-image-configuration	驗證 AMI 時發生錯誤。	<p>檢視錯誤的技術詳細資訊。您無法再次嘗試使用此 CEV 驗證 AMI。請檢閱下列建議：</p> <ul style="list-style-type: none"> • 確定您的 CEV 是使用 SQL Server 主要版本 + 次要版本 + 自訂字串的必要命名模式來命名。 • 確定 CEV 名稱中的 SQL Server 版本與 AMI 隨附的版本相符。 • 確定作業系統組建版本符合所需的最低組建版本。 • 確定作業系統主要版本符合所需的最低主要版本。 <p>使用正確的資訊建立新的 CEV。</p> <p>如有需要，請使用支援的 AMI 建立新的 EC2 執行個體，並在其上執行 Sysprep 程序。</p>

修改 RDS Custom for SQL Server 的 CEV

您可以使用 AWS Management Console 或 AWS CLI 修改 CEV。您可以修改 CEV 描述或其可用狀態。您的 CEV 有下列其中一個狀態值：

- **available** – 您可以使用此 CEV 建立新的 RDS Custom 資料庫執行個體或升級資料庫執行個體。這是新建立的 CEV 的預設狀態。
- **inactive** – 您無法使用此 CEV 建立或升級 RDS Custom 資料庫執行個體。您無法還原資料庫快照以使用此 CEV 建立新的 RDS Custom 資料庫執行個體。

您可以將 CEV 狀態從 **available** 變更為 **inactive** 或從 **inactive** 變更為 **available**。您可以將狀態變更為 **INACTIVE**，以防止意外使用 CEV，或讓已中止的 CEV 符合再次使用的資格。

主控台

若要修改 CEV

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。
3. 選擇您想要修改其描述或狀態的 CEV。
4. 在 Actions (動作) 中，選擇 Modify (修改)。
5. 進行下列任何變更：
 - 針對 CEV status settings (CEV 狀態設定)，選擇新的可用性狀態。
 - 針對 Version description (版本描述)，輸入新的描述。
6. 選擇 Modify CEV (修改 CEV)。

如果 CEV 正在使用中，主控台會顯示您無法修改 CEV 狀態。修正問題，然後再試一次。

Custom engine versions (自訂引擎版本) 頁面隨即出現。

AWS CLI

若要使用修改 CEVAWS CLI，請執行 [modify-custom-db-engine-version](#) 命令。您可以執行 [describe-db-engine-versions](#) 指令來尋找要修改的 CEV。

下列選項是必要的：

- `--engine`
- `--engine-version cev`，其中 *cev* 是您要修改的自訂引擎版本的名稱。
- `--status status`，其中 *status* 是您要指派給 CEV 的可用性狀態。

下列範例會將名為 `15.00.4249.2.my_cevtest` 的 CEV 從目前的狀態變更為 `inactive`。

Example

對於 Linux/macOS、或 Unix：

```
aws rds modify-custom-db-engine-version \
```

```
--engine custom-sqlserver-ee \  
--engine-version 15.00.4249.2.my_cevtest \  
--status inactive
```

在Windows中：

```
aws rds modify-custom-db-engine-version ^  
--engine custom-sqlserver-ee ^  
--engine-version 15.00.4249.2.my_cevtest ^  
--status inactive
```

修改 RDS Custom for SQL Server 資料庫執行個體以使用新的 CEV

您可以修改現有的 RDS Custom for SQL Server 資料庫執行個體，以使用不同的 CEV。您可以進行的變更包括：

- 變更 CEV
- 變更資料庫執行個體類別
- 變更備份保留期和備份時段
- 變更維護時段

主控台

若要修改 RDS Custom for SQL Server 資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。
4. 選擇 Modify (修改)。
5. 視需要進行下列變更：
 - a. 對於 DB engine version (資料庫引擎版本)，選擇不同的 CEV。
 - b. 變更 DB instance class (資料庫執行個體類別) 的值。如需支援的類別，請參閱 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)。
 - c. 變更 Backup retention period (備份保留期) 的值。
 - d. 針對 Backup window (備份時段)，設定 Start time (開始時間) 和 Duration (持續時間) 的值。

- e. 針對 DB instance maintenance window (資料庫執行個體維護時段)，設定 Start day (開始日期)、Start time (開始時間) 和 Duration (持續時間) 的值。
6. 選擇 Continue (繼續)。
7. 選擇 Apply immediately (立即套用) 或 Apply during the next scheduled maintenance window (在下次排定的維護時段套用)。
8. 選擇 Modify DB instance (修改資料庫執行個體)。

Note

將資料庫執行個體從一個 CEV 修改為另一個 CEV (例如，升級次要版本時) 時，SQL Server 系統資料庫 (包括其資料和組態) 會從目前的 RDS Custom for SQL Server 資料庫執行個體保存下來。

AWS CLI

若要使用修改資料庫執行個體以使用不同的 CEVAWS CLI，請執行命 [modify-db-instance](#) 令。

下列選項是必要的：

- `--db-instance-identifier`
- `--engine-version cev`，其中 *cev* 是自訂引擎版本的名稱，您想要資料庫執行個體變更為該版本。

下列範例會修改名為 `my-cev-db-instance` 的資料庫執行個體，以使用名為 `15.00.4249.2.my_cevtest_new` 的 CEV，並立即套用變更。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-cev-db-instance \  
  --engine-version 15.00.4249.2.my_cevtest_new \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^
```

```
--db-instance-identifier my-cev-db-instance ^  
--engine-version 15.00.4249.2.my_cevtest_new ^  
--apply-immediately
```

檢視 Amazon RDS Custom for SQL Server 的 CEV 詳細資訊

您可以使用 AWS Management Console 或 AWS CLI 來檢視 CEV 的詳細資訊。

主控台

檢視 CEV 詳細資訊

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。

Custom engine versions (自訂引擎版本) 頁面會顯示目前存在的所有 CEV。如果您尚未建立任何 CEV，則頁面為空白。

3. 選擇要檢視的 CEV 名稱。
4. 選擇 Configuration (組態) 來檢視詳細資訊。

RDS > Custom engine versions > 15.00.4249.2.test-cev-v1

15.00.4249.2.test-cev-v1

Summary

Name	15.00.4249.2.test-cev-v1	Status	Available	Date created	12/12/2022, 4:50:24 PM
Description	test-cev-v1 gul testing	Engine	SQL Server Standard Edition		

Configuration

Edition	SQL Server Standard Edition	Amazon Resource Name (ARN)	arn:aws:rds:us-west-2:123456789012:cev:custom-sqlserver-se/15.00.4249.2.test-cev-v1/d5d0adcc-2ff7-44d4-ba33-b53d7adb24ab
Major Version	15.00	KMS key ID	-
AMI	ami-063e		

AWS CLI

若要使用 AWS CLI 檢視 CEV 的詳細資訊，請執行 [describe-db-engine-versions](#) 命令。

您還可以指定下列選項：

- `--include-all`，檢視具有任何生命週期狀態的所有 CEV。若沒有 `--include-all` 選項，只會傳回處於 `available` 生命週期狀態的 CEV。

```
aws rds describe-db-engine-versions --engine custom-sqlserver-ee --engine-version
15.00.4249.2.my_cevtest --include-all
{
  "DBEngineVersions": [
    {
      "Engine": "custom-sqlserver-ee",
      "MajorEngineVersion": "15.00",
      "EngineVersion": "15.00.4249.2.my_cevtest",
      "DBParameterGroupFamily": "custom-sqlserver-ee-15.0",
      "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for custom
RDS",
      "DBEngineVersionArn": "arn:aws:rds:us-east-1:{my-account-id}:cev:custom-
sqlserver-ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
      "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",
      "Image": {
        "ImageId": "ami-0r93cx31t5r596482",
        "Status": "pending-validation"
      },
      "DBEngineMediaType": "AWS Provided",
      "CreateTime": "2022-11-20T19:30:01.831000+00:00",
      "ValidUpgradeTarget": [],
      "SupportsLogExportsToCloudwatchLogs": false,
      "SupportsReadReplica": false,
      "SupportedFeatureNames": [],
      "Status": "pending-validation",
      "SupportsParallelQuery": false,
      "SupportsGlobalDatabases": false,
      "TagList": [],
      "SupportsBabelfish": false
    }
  ]
}
```

您可以使用篩選條件，檢視具有特定生命週期狀態的 CEV。例如，若要檢視生命週期狀態為 `pending-validation`、`available` 或 `failed` 的 CEV：

```
aws rds describe-db-engine-versions engine custom-sqlserver-ee
      region us-west-2 include-all query 'DBEngineVersions[?Status ==
pending-validation ||
      Status == available || Status == failed]'
```

刪除 RDS Custom for SQL Server 的 CEV

您可以使用 AWS Management Console 或 AWS CLI 刪除 CEV。通常，此任務需要幾分鐘的時間。

刪除 CEV 前，請確定下列任何一項未使用該 CEV：

- RDS Custom 資料庫執行個體
- RDS Custom 資料庫執行個體的快照
- RDS Custom 資料庫執行個體的自動備份

主控台

若要刪除 CEV

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Custom engine versions (自訂引擎版本)。
3. 選擇您想要刪除其描述或狀態的 CEV。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。

Delete *cev_name*? (是否刪除 *cev_name*?) 對話方塊隨即顯示。

5. 輸入 **delete me**，然後選擇 Delete (刪除)。

在 Custom engine versions (自訂引擎版本) 頁面上，橫幅會顯示您的 CEV 即將刪除。

AWS CLI

若要使用刪除 CEVAWS CLI，請執行 [delete-custom-db-engine-version](#) 命令。

下列選項是必要的：

- `--engine custom-sqlserver-ee`
- `--engine-version cev`，其中 *CEV* 是要刪除之自訂引擎版本的名稱

以下範例會刪除名為 `15.00.4249.2.my_cevtest` 的 CEV。

Example

對於LinuxmacOS、或Unix：

```
aws rds delete-custom-db-engine-version \  
  --engine custom-sqlserver-ee \  
  --engine-version 15.00.4249.2.my_cevtest
```

在Windows中：

```
aws rds delete-custom-db-engine-version ^  
  --engine custom-sqlserver-ee ^  
  --engine-version 15.00.4249.2.my_cevtest
```

建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體

您可以建立 RDS 自訂資料庫執行個體，然後使用 AWS Systems Manager 或遠端桌面通訊協定 (RDP) 連線至該執行個體。

Important

您必須完成 [設定您的 Amazon RDS Custom for SQL Server 的環境](#) 中的任務，才能建立或連線至 RDS Custom for SQL Server 資料庫執行個體。

當您建立 RDS Custom 資料庫執行個體時可以為它加上標籤，但是請勿建立或修改 RDS Custom 自動化所需的 AWSRDSCustom 標籤。如需詳細資訊，請參閱 [標記 RDS Custom for SQL Server 資源](#)。

第一次建立 RDS Custom for SQL Server 資料庫執行個體時，您可能會收到下列錯誤：服務連結角色正在建立的程序中。請稍後再試。如果收到這個訊息，請等待幾分鐘的時間然後再次嘗試建立資料庫執行個體。

主題

- [建立 RDS Custom for SQL Server 資料庫執行個體](#)
- [RDS Custom 服務連結角色](#)
- [使用連接到 RDS 自訂資料庫執行個體 AWS Systems Manager](#)
- [使用 RDS 連線到您的 RDS Custom 資料庫執行個體](#)

建立 RDS Custom for SQL Server 資料庫執行個體

使用 AWS Management Console 或建立適用於 SQL 伺服器資料庫執行個體的 Amazon RDS 自訂 AWS CLI。該程序與建立 Amazon RDS 資料庫執行個體的程序類型。

如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

主控台

若要建立 RDS Custom for SQL Server 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

3. 選擇 Create database (建立資料庫)。
4. 選擇 Standard Create (標準建立) 做為資料庫建立方法。
5. 針對 Engine options (引擎選項)，選擇 Microsoft SQL Server 做為引擎類型。
6. 針對 Database management type (引擎管理類型)，選擇 Amazon RDS Custom。
7. 在 Edition (版本) 區段中，選擇您想要使用的資料庫引擎版本。
8. (選用) 如果您想要從 CEV 建立資料庫執行個體，請核取 Use custom engine version (CEV) (使用自訂引擎版本 (CEV)) 核取方塊。在下拉式清單中選取您的 CEV。
9. 對於資料庫版本，請保留預設值版本。
10. 在 Templates (範本) 中，選擇 Production (生產)。
11. 在 Settings (設定) 區段中，為 DB instance identifier (資料庫執行個體識別符) 輸入唯一的名稱。
12. 若要輸入您的主要密碼，請執行以下動作：
 - a. 在 Settings (設定) 區段，開啟 Credential Settings (登入資料設定)。
 - b. 清除 Auto generate a password (自動產生密碼) 核取方塊。
 - c. 變更 Master username (主要使用者名稱) 值並在 Master password (主要密碼) 和 Confirm password (確認密碼) 中輸入相同的密碼。

根據預設，新 RDS Custom 資料庫執行個體會將自動產生的密碼用於主要使用者。

13. 在 DB instance size (資料庫執行個體大小) 區段中，為 DB instance class (資料庫執行個體類別) 選擇值。

如需支援的類別，請參閱 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)。

14. 選擇 Storage (儲存) 設定。
15. 針對 RDS Custom security (RDS Custom 安全)，請執行下列操作：
 - a. 對於 IAM 執行個體設定檔，您有兩個選項可選擇適用於 SQL Server 資料庫執行個體的 RDS 自訂執行個體設定檔。
 1. 選擇 [建立新的執行個體設定檔] 並提供執行個體設定檔名稱尾碼 如需詳細資訊，請參閱 [自動建立執行個體設定檔，使 AWS Management Console](#)。
 2. 選擇現有的執行個體設定檔。從 d' 下拉式清單中，選擇開頭為的執行個體設定檔。AWSRDSCustom
 - b. 針對 Encryption (加密)，選擇 Enter a key ARN (輸入金鑰 ARN) 以列出可用的 AWS KMS 金鑰。然後從清單選擇您的金鑰。

RDS 自訂需要 AWS KMS 金鑰。如需詳細資訊，請參閱 [確定您擁有對稱式加密金鑰 AWS KMS](#)。

16. 在其餘區段，指定您偏好的 RDS Custom 資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。以下設定不會出現在主控台中，也不受支援：

- 處理器功能
- 儲存體自動調整規模
- Availability & durability (可用性與持久性)
- Database authentication (資料庫身分驗證) 中的 Password and Kerberos authentication (密碼和 Kerberos 身分驗證) 選項 (僅支援 Password authentication (密碼身分驗證))
- Additional configuration (其他組態) 中的 Database options (資料庫選項)
- Performance Insights (績效詳情)
- Log exports (日誌匯出)
- Enable auto minor version upgrade (啟用自動次要版本升級)
- 刪除保護

支援 Backup retention period (備份保留期)，但是您無法選擇 0 days (0 天)。

17. 選擇 Create database (建立資料庫)。

View credential details (檢視憑證詳細資訊) 按鈕會顯示在 Databases (資料庫) 頁面上。

若要檢視 RDS Custom 資料庫執行個體的主要使用者名稱和密碼，請選擇 View credential details (檢視憑證詳細資訊)。

若要以主要使用者的身分連線至資料庫執行個體，請使用出現的使用者名稱和密碼。

Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。若要在 RDS Custom 資料庫執行個體可供使用後變更主要使用者密碼，請修改資料庫執行個體。如需修改 資料庫執行個體的詳細資訊，請參閱 [管理 Amazon RDS Custom for SQL Server 資料庫執行個體](#)。

18. 選擇 Databases (資料庫) 以檢視 RDS Custom 資料庫執行個體的清單。

19. 選擇您剛剛建立的 RDS Custom 資料庫執行個體。

在 RDS 主控台上，新 RDS Custom 資料庫執行個體的詳細資訊即會出現：

- 在 RDS Custom 資料庫執行個體建立完成且可供使用之前，資料庫執行個體會處於 Creating (建立中) 狀態。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視執行個體類別和分配的儲存空間而定，新的資料庫執行個體可能需要幾分鐘才能使用。
- Role (角色) 具有 Instance (RDS Custom) (執行個體 (RDS Custom)) 值。
- RDS Custom automation mode (RDS Custom 自動化模式) 具有 Full automation (全自動化) 值。此設定表示資料庫執行個體提供自動監控和執行個體復原。

AWS CLI

您可以使用建立 DB 執行個體命令來[建立 RDS 自訂資料庫執行個體](#) AWS CLI。

下列選項是必要的：

- `--db-instance-identifier`
- `--db-instance-class` (如需支援執行個體類別的清單，請參閱 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#))
- `--engine` (`custom-sqlserver-ee`、`custom-sqlserver-se` 或 `custom-sqlserver-web`)
- `--kms-key-id`
- `--custom-iam-instance-profile`

下列範例會建立名為 `my-custom-instance` 的 RDS Custom for SQL Server 資料庫執行個體。備份保留期為 3 天。

Note

若要從自訂引擎版本 (CEV) 建立資料庫執行個體，請將現有的 CEV 名稱提供給 `--engine-version` 參數。例如：`--engine-version 15.00.4249.2.my_cevtest`

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-db-instance \
```

```
--engine custom-sqlserver-ee \  
--engine-version 15.00.4073.23.v1 \  
--db-instance-identifier my-custom-instance \  
--db-instance-class db.m5.xlarge \  
--allocated-storage 20 \  
--db-subnet-group mydbsubnetgroup \  
--master-username myuser \  
--master-user-password mypassword \  
--backup-retention-period 3 \  
--no-multi-az \  
--port 8200 \  
--kms-key-id mykmskey \  
--custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --engine custom-sqlserver-ee ^  
  --engine-version 15.00.4073.23.v1 ^  
  --db-instance-identifier my-custom-instance ^  
  --db-instance-class db.m5.xlarge ^  
  --allocated-storage 20 ^  
  --db-subnet-group mydbsubnetgroup ^  
  --master-username myuser ^  
  --master-user-password mypassword ^  
  --backup-retention-period 3 ^  
  --no-multi-az ^  
  --port 8200 ^  
  --kms-key-id mykmskey ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

使用 `describe-db-instances` 命令取得執行個體的詳細資訊。

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

下列部分輸出顯示引擎、參數群組及其他資訊。

```
{
  "DBInstances": [
    {
      "PendingModifiedValues": {},
      "Engine": "custom-sqlserver-ee",
      "MultiAZ": false,
      "DBSecurityGroups": [],
      "DBParameterGroups": [
        {
          "DBParameterGroupName": "default.custom-sqlserver-ee-15",
          "ParameterApplyStatus": "in-sync"
        }
      ],
      "AutomationMode": "full",
      "DBInstanceIdentifier": "my-custom-instance",
      "TagList": []
    }
  ]
}
```

RDS Custom 服務連結角色

服務連結角色可讓 Amazon RDS 自訂存取您 AWS 帳戶的。它會讓使用 RDS Custom 更簡單，因為您不需要手動新增必要許可。RDS Custom 會定義其服務連結角色的許可，除非另有定義，否則僅有 RDS Custom 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能連接到任何其他 IAM 實體。

當您建立 RDS Custom 資料庫執行個體時，Amazon RDS 和 RDS Custom 服務連結角色兩者都會建立 (如果尚未存在) 並且使用。如需詳細資訊，請參閱 [使用 Amazon RDS 的服務連結角色](#)。

第一次建立 RDS Custom for SQL Server 資料庫執行個體時，您可能會收到下列錯誤：服務連結角色正在建立的程序中。請稍後再試。如果收到這個訊息，請等待幾分鐘的時間然後再次嘗試建立資料庫執行個體。

使用連接到 RDS 自訂資料庫執行個體 AWS Systems Manager

在建立 RDS Custom 資料庫執行個體之後，您可以使用 AWS Systems Manager 工作階段管理員連線到該執行個體。「工作階段管理員」是 Systems Manager 功能，您可以用來透過瀏覽器型殼層或透過 AWS CLI 管理 Amazon EC2 執行個體。如需詳細資訊，請參閱 [AWS Systems Manager 工作階段管理員](#)。

主控台

若要使用工作階段管理員連線至資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要連線的 RDS Custom 資料庫執行個體。
3. 選擇 Configuration (組態)。
4. 請記下資料庫執行個體的 Resource ID (資源 ID) 值。例如，資源 ID 可能是 db-ABCDEFGHIJKLMNOPS0123456。
5. 在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在導覽窗格中，選擇執行個體。
7. 尋找 EC2 執行個體的名稱，然後選擇與其相關聯的執行個體 ID。例如，執行個體 ID 可能是 i-abcdefghijklm01234。
8. 選擇連線。
9. 選擇 Session Manager (工作階段管理員)。
10. 選擇連線。

您的工作階段會開啟一個視窗。

AWS CLI

您可以使用 AWS CLI 連線至 RDS Custom 資料庫執行個體。此技術需要 AWS CLI 的工作階段管理員外掛程式。若要了解如何安裝外掛程式，請參閱 [安裝 AWS CLI 的工作階段管理員外掛程式](#)。

若要尋找 RDS Custom 資料庫執行個體的資料庫資源 ID，請使用 [describe-db-instances](#)。

```
aws rds describe-db-instances \  
  --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \  
  --output text
```

下列範例輸出顯示 RDS Custom 執行個體的資源 ID。字首是 db-。

```
db-ABCDEFGHIJKLMNOPS0123456
```

若要尋找資料庫執行個體的 EC2 執行個體 ID，請使用 `aws ec2 describe-instances`。以下範例使用 db-ABCDEFGHIJKLMNOPS0123456 做為資源 ID。

```
aws ec2 describe-instances \  
  --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \  
  --output text \  
  --query 'Reservations[*].Instances[*].InstanceId'
```

以下範例輸出顯示 EC2 執行個體 ID。

```
i-abcdefghijklm01234
```

使用 `aws ssm start-session` 命令，在 `--target` 參數中提供 EC2 執行個體 ID。

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

成功的連線看起來如下。

```
Starting session with SessionId: yourid-abcdefghijklm1234  
[ssm-user@ip-123-45-67-89 bin]$
```

使用 RDS 連線到您的 RDS Custom 資料庫執行個體

建立 RDS Custom 資料庫執行個體之後，您可以使用 RDP 用戶端連線到此執行個體。此程序與連線 Amazon EC2 執行個體的程序相同。如需詳細資訊，請參閱[連線至您的 Windows 執行個體](#)。

若要連線到資料庫執行個體，您需要與執行個體相關聯的金鑰對。RDS Custom 會為您建立金鑰對。配對名稱使用字首 `do-not-delete-rds-custom-DBInstanceIdentifier`。AWS Secrets Manager 會將您的私有金鑰儲存為秘密。

完成下列步驟中的任務：

1. [將您的資料庫執行個體設定為允許 RDP 連線](#)。
2. [擷取您的私密金鑰](#)。
3. [使用 RDP 公用程式連線到 EC2 執行個體](#)。

將您的資料庫執行個體設定為允許 RDP 連線

若要允許 RDP 連線，請設定 VPC 安全群組並在主機上設定防火牆規則。

設定您的 VPC 安全群組

請確定與資料庫執行個體相關聯的 VPC 安全群組允許傳輸控制通訊協定 (TCP) 的連接埠 3389 上的傳入連線。若要了解如何設定 VPC 安全群組，請參閱 [設定您的 VPC 安全群組](#)。

在主機上設定防火牆規則

若要允許 TCP 的連接埠 3389 上的傳入連線，請在主機上設定防火牆規則。下列範例示範其做法。

建議您使用特定的 `-Profile` 值：`Public`、`Private` 或 `Domain`。使用 `Any` 指的是全部三個值。您也可指定以逗號分隔的值組合。如需有關設定防火牆規則的詳細資訊，請參閱 Microsoft 文件中的 [設定規 NetFirewall 則](#)。

使用 Systems Manager 工作階段管理員來設定防火牆規則

1. 連線至工作階段管理員，如 [使用連接到 RDS 自訂資料庫執行個體 AWS Systems Manager](#) 中所示。
2. 執行下列命令。

```
Set-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)" -Direction Inbound -LocalAddress Any -Profile Any
```

使用 Systems Manager CLI 命令來設定防火牆規則

1. 使用下列命令在主機上開啟 RDP。

```
OPEN_RDP_COMMAND_ID=$(aws ssm send-command --region $AWS_REGION \
  --instance-ids $RDS_CUSTOM_INSTANCE_EC2_ID \
  --document-name "AWS-RunPowerShellScript" \
  --parameters '{"commands":["Set-NetFirewallRule -DisplayName \"Remote Desktop - User Mode (TCP-In)\" -Direction Inbound -LocalAddress Any -Profile Any"]}' \
  --comment "Open RDP port" | jq -r ".Command.CommandId")
```

2. 使用輸出中傳回的命令 ID，以取得前一個命令的狀態。若要使用以下查詢傳回命令 ID，請確定已安裝 jq 外掛程式。

```
aws ssm list-commands \
  --region $AWS_REGION \
  --command-id $OPEN_RDP_COMMAND_ID
```

擷取您的私密金鑰

使用 AWS Management Console 或擷取您的秘密金鑰 AWS CLI。

主控台

若要擷取秘密金鑰

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要連線的 RDS Custom 資料庫執行個體。
3. 選擇 Configuration (組態) 索引標籤。
4. 請記下資料庫執行個體的 DB instance ID (資料庫執行個體 ID)，例如 *my-custom-instance*。
5. 在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在導覽窗格中，選擇執行個體。
7. 尋找 EC2 執行個體的名稱，然後選擇與其相關聯的執行個體 ID。

在此範例中，執行個體 ID 為 i-abcdefghijklm01234。

8. 在 Details (詳細資訊) 中尋找 Key pair name (金鑰對名稱)。配對名稱包含資料庫識別符。在此範例中，配對名稱是 do-not-delete-rds-custom-*my-custom-instance-0d726c*。
9. 在執行個體摘要中，尋找公有 IPv4 DNS。例如，公有 DNS 可能是 ec2-12-345-678-901.us-east-2.compute.amazonaws.com。
10. [請在以下位置開啟 AWS Secrets Manager 主控台。](https://console.aws.amazon.com/secretsmanager/) <https://console.aws.amazon.com/secretsmanager/>
11. 選擇與金鑰對具有相同名稱的密碼。
12. 選擇 Retrieve secret value (擷取秘密值)。

AWS CLI

若要擷取私有金鑰

1. 藉由呼叫 `aws rds describe-db-instances` 命令來取得 RDS Custom 資料庫執行個體的清單。

```
aws rds describe-db-instances \  
  --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \  
  --output text
```

```
--output text
```

- 從範例輸出中選擇資料庫執行個體識別符，例如 `do-not-delete-rds-custom-my-custom-instance`。
- 藉由呼叫 `aws ec2 describe-instances` 命令來尋找資料庫執行個體的 EC2 執行個體 ID。下列範例使用 EC2 執行個體名稱來描述資料庫執行個體。

```
aws ec2 describe-instances \  
  --filters "Name=tag:Name,Values=do-not-delete-rds-custom-my-custom-instance" \  
  --output text \  
  --query 'Reservations[*].Instances[*].InstanceId'
```

以下範例輸出顯示 EC2 執行個體 ID。

```
i-abcdefghijklm01234
```

- 藉由指定 EC2 執行個體 ID 來尋找金鑰名稱，如以下範例所示。

```
aws ec2 describe-instances \  
  --instance-ids i-abcdefghijklm01234 \  
  --output text \  
  --query 'Reservations[*].Instances[*].KeyName'
```

下列範例輸出會顯示金鑰名稱，該名稱使用字首 `do-not-delete-rds-custom-DBInstanceIdentifier`。

```
do-not-delete-rds-custom-my-custom-instance-0d726c
```

使用 RDP 公用程式連線到 EC2 執行個體

請按照 Amazon EC2 使用者指南中的[使用 RDP Connect 到 Windows 執行個體](#)中的程序進行操作。此程序假設您已建立包含私有金鑰的 `.pem` 檔案。

管理 Amazon RDS Custom for SQL Server 資料庫執行個體

Amazon RDS Custom for SQL Server 支援 Amazon RDS 資料庫執行個體一般管理任務的子集。接著，您可以使用 AWS Management Console 和 AWS CLI 尋找支援的 RDS Custom for SQL Server 管理任務的指示。

主題

- [暫停和繼續 RDS Custom 自動化](#)
- [修改 RDS Custom for SQL Server 資料庫執行個體](#)
- [修改 RDS Custom for SQL Server 資料庫執行個體的儲存體](#)
- [標記 RDS Custom for SQL Server 資源](#)
- [刪除 RDS Custom for SQL Server 資料庫執行個體](#)
- [啟動與停止 RDS Custom for SQL Server 資料庫執行個體](#)

暫停和繼續 RDS Custom 自動化

RDS Custom 會自動為 RDS Custom for SQL Server 資料庫執行個體提供監控和執行個體復原。如果需要自訂執行個體，請執行下列動作：

1. 在指定期間暫停 RDS Custom 自動化。暫停可確保您的自訂不會干擾 RDS Custom 自動化。
2. 視需要自訂 RDS Custom for SQL Server 資料庫執行個體。
3. 執行下列任何一項：
 - 手動繼續自動化。
 - 等待暫停期間結束。在此情況下，RDS Custom 會自動繼續監控和執行個體復原。

Important

修改 RDS Custom for SQL Server 資料庫執行個體時，暫停和繼續自動化是唯一支援的自動化任務。

主控台

若要暫停或繼續 RDS Custom 自動化

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要修改的 RDS Custom 資料庫執行個體。
3. 選擇 Modify (修改)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
4. 針對 RDS Custom automation mode (RDS Custom 自動化模式)，選擇下列其中一個選項：
 - Paused (暫停) 會暫停 RDS Custom 資料庫執行個體的監控和執行個體復原。輸入您想要的 Automation mode duration (自動化模式持續時間) 的暫停持續時間 (以分鐘為單位)。最小值為 60 分鐘 (預設值)。最大值為 1,440 分鐘。
 - Full automation (全自動化) 會繼續自動化。
5. 選擇 Continue (繼續) 以檢查修改的摘要。

訊息隨即顯示，指出 RDS Custom 將會立即套用變更。

6. 如果您的變更正確，請選擇 Modify DB instance (修改資料庫執行個體)。或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

在 RDS 主控台上，修改的詳細資訊即會出現。如果您暫停了自動化，您的 RDS Custom 資料庫執行個體的 Status (狀態) 會指出 Automation paused (自動化已暫停)。

7. (選用) 在導覽窗格中，選擇 Databases (資料庫)，然後再執行 RDS Custom 資料庫執行個體。

在 Summary (摘要) 窗格中，RDS Custom automation mode (RDS Custom 自動化模式) 會指出自動化狀態。如果暫停自動化，則值為 Paused (暫停)。自動化會在 *num* 分鐘內繼續。

AWS CLI

若要暫停或繼續 RDS 自訂自動化，請使用 `modify-db-instance` AWS CLI 命令。使用所需的參數 `--db-instance-identifier` 識別資料庫執行個體。使用下列參數控制自動化模式：

- `--automation-mode` 會指定資料庫執行個體的暫停狀態。有效值為 `all-paused` (會暫停自動化) 以及 `full` (會繼續自動化)。
- `--resume-full-automation-mode-minutes` 會指定暫停的持續時間。預設值為 60 分鐘。

Note

無論您指定 `--no-apply-immediately` 或 `--apply-immediately`，RDS Custom 會盡快以非同步的方式套用修改。

在命令回應中，`ResumeFullAutomationModeTime` 會以 UTC 時間戳記的形式表示繼續時間。當自動化模式為 `all-paused` 時，您可以使用 `modify-db-instance` 來繼續自動化模式或延長暫停時間。不支援其他 `modify-db-instance` 加密選項。

下列範例會暫停 `my-custom-instance` 的自動化 90 分鐘。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --automation-mode all-paused \  
  --resume-full-automation-mode-minutes 90
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-custom-instance ^  
  --automation-mode all-paused ^  
  --resume-full-automation-mode-minutes 90
```

下列範例會將暫停持續時間延長額外 30 分鐘。30 分鐘會新增至 `ResumeFullAutomationModeTime` 中顯示的原始時間。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --automation-mode all-paused \  
  --resume-full-automation-mode-minutes 30
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier my-custom-instance ^
  --automation-mode all-paused ^
  --resume-full-automation-mode-minutes 30
```

下列範例會繼續 `my-custom-instance` 的全自動化。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \
  --db-instance-identifier my-custom-instance \
  --automation-mode full \
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier my-custom-instance ^
  --automation-mode full
```

在下面的部分範例輸出中，待定 `AutomationMode` 值為 `full`。

```
{
  "DBInstance": {
    "PubliclyAccessible": true,
    "MasterUsername": "admin",
    "MonitoringInterval": 0,
    "LicenseModel": "bring-your-own-license",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "0123456789abcdefg"
      }
    ],
    "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
    "CopyTagsToSnapshot": false,
    "OptionGroupMemberships": [
      {
        "Status": "in-sync",
        "OptionGroupName": "default:custom-oracle-ee-19"
      }
    ]
  }
}
```

```

    ],
    "PendingModifiedValues": {
      "AutomationMode": "full"
    },
    "Engine": "custom-oracle-ee",
    "MultiAZ": false,
    "DBSecurityGroups": [],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.custom-oracle-ee-19",
        "ParameterApplyStatus": "in-sync"
      }
    ],
    ...
    "ReadReplicaDBInstanceIdentifiers": [],
    "AllocatedStorage": 250,
    "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
    "BackupRetentionPeriod": 3,
    "DBName": "ORCL",
    "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
    "Endpoint": {
      "HostedZoneId": "ABCDEFGHIJKLMNO",
      "Port": 8200,
      "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
    },
    "DBInstanceStatus": "automation-paused",
    "IAMDatabaseAuthenticationEnabled": false,
    "AutomationMode": "all-paused",
    "EngineVersion": "19.my_cev1",
    "DeletionProtection": false,
    "AvailabilityZone": "us-west-2a",
    "DomainMemberships": [],
    "StorageType": "gp2",
    "DbiResourceId": "db-ABCDEFGHIJKLMNQRSTUUVW",
    "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
    "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
    "StorageEncrypted": false,
    "AssociatedRoles": [],
    "DBInstanceClass": "db.m5.xlarge",
    "DbInstancePort": 0,
    "DBInstanceIdentifier": "my-custom-instance",
    "TagList": []

```

```
}
```

修改 RDS Custom for SQL Server 資料庫執行個體

修改 RDS Custom for SQL Server 資料庫執行個體類似於針對 Amazon RDS 執行這個操作，但是您可以進行的變更限制為下列項目：

- 變更資料庫執行個體類別
- 變更備份保留期和備份時段
- 變更維護時段
- 當新版本可用時升級資料庫引擎版本
- 變更已配置的儲存體、佈建 IOPS 和儲存體類型
- 變更資料庫連接埠
- 變更資料庫執行個體識別符
- 變更主要憑證
- 允許和移除多可用區部署
- 允許公開存取
- 變更安全群組
- 變更子網路群組

修改 RDS Custom for SQL Server 資料庫執行個體時有下列限制：

- 不支援自訂資料庫選項和參數群組。
- 手動連接至 RDS Custom 資料庫執行個體的任何儲存磁碟區都在支援周邊外面。

如需詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

主控台

若要修改 RDS Custom for SQL Server 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。

4. 選擇 Modify (修改)。
5. 視需要進行下列變更：
 - a. 在 DB engine version (資料庫引擎版本) 中，選擇新版本。
 - b. 變更 DB instance class (資料庫執行個體類別) 的值。如需支援的類別，請參閱 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)。
 - c. 變更 Backup retention period (備份保留期) 的值。
 - d. 針對 Backup window (備份時段)，設定 Start time (開始時間) 和 Duration (持續時間) 的值。
 - e. 針對 DB instance maintenance window (資料庫執行個體維護時段)，設定 Start day (開始日期)、Start time (開始時間) 和 Duration (持續時間) 的值。
6. 選擇 Continue (繼續)。
7. 選擇 Apply immediately (立即套用) 或 Apply during the next scheduled maintenance window (在下次排定的維護時段套用)。
8. 選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要修改適用於 SQL Server 資料庫執行個體的 RDS 自訂，請使用 [modify-db-instance](#) AWS CLI 命令。視需要設定下列參數：

- `--db-instance-class` – 如需支援的類別，請參閱 [RDS Custom for SQL Server 的資料庫執行個體類別支援](#)
- `--engine-version` – 您要升級的資料庫引擎版本號碼。
- `--backup-retention-period` – 保留自動化備份的期間，從 0–35 天。
- `--preferred-backup-window` – 每日時間範圍，自動化備份會在這段期間建立。
- `--preferred-maintenance-window` – 系統維護可能發生期間的每週時間範圍 (格式為 UTC)。
- `--apply-immediately` – 使用 `--apply-immediately` 可立即套用儲存變更。

或使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。

修改 RDS Custom for SQL Server 資料庫執行個體的儲存體

修改 RDS Custom for SQL Server 資料庫執行個體的儲存體類似於修改 Amazon RDS 資料庫執行個體的儲存體，但是您只能執行下列動作：

- 增加配置的儲存體大小。
- 變更儲存體類型。您可以使用可用的儲存體類型，例如一般用途或佈建 IOPS。gp3、io1 和 io2 區塊快速儲存區類型支援佈建的 IOPS。
- 如果您使用支援佈建 IOPS 的磁碟區類型，請變更佈建的 IOPS。

修改 RDS Custom for SQL Server 資料庫執行個體的儲存體時有下列限制：

- 針對 RDS Custom for SQL Server 配置的儲存體大小下限為 20 GiB，而支援的儲存體大小上限為 16 TiB。
- 與 Amazon RDS 一樣，您無法減少配置的儲存體。這是 Amazon Elastic Block Store (Amazon EBS) 磁碟區的限制。如需更多資訊，請參閱[使用適合 Amazon RDS 資料庫執行個體的儲存體](#)
- RDS Custom for SQL Server 執行個體不支援儲存體自動擴展。
- 不會考慮對您手動連接至 RDS Custom 資料庫執行個體的任何儲存磁碟區進行儲存體擴展。只有 RSS 提供的預設資料磁碟區 (即 D 磁碟機)，才會考慮對其進行儲存體擴展。

如需詳細資訊，請參閱 [RDS Custom 支援周邊](#)。

- 擴展儲存體通常不會導致資料庫執行個體的任何中斷或效能下降。修改完資料庫執行個體的儲存空間大小之後，其狀態呈現為 storage-optimization (儲存最佳化)。
- 儲存體最佳化可能需要幾個小時。在六 (6) 小時或在執行個體上完成儲存體最佳化 (以時間較長者為準) 後，您才能進行進一步的儲存體修改。如需更多資訊，請參閱[使用適合 Amazon RDS 資料庫執行個體的儲存體](#)

如需儲存體的詳細資訊，請參閱[Amazon RDS 資料庫執行個體儲存體](#)。

如需與儲存體修改相關的一般資訊，請參閱 [使用適合 Amazon RDS 資料庫執行個體的儲存體](#)。

主控台

修改 RDS Custom for SQL Server 資料庫執行個體的儲存體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。
4. 選擇 Modify (修改)。

5. 視需要進行下列變更：
 - a. 針對 Allocated storage (配置儲存) 輸入新的值。它必須大於目前的值，並且從 20 GiB 至 64 TiB。
 - b. 變更 Storage type (儲存體類型) 的值。您可以選擇可用的一般用途或佈建 IOPS 儲存區類型。gp3、io1 和 io2 區塊快速儲存區類型支援佈建的 IOPS。
 - c. 如果您指定支援佈建 IOPS 的儲存區類型，則可以定義佈建 IOPS 值。
6. 選擇 Continue (繼續)。
7. 選擇 Apply immediately (立即套用) 或 Apply during the next scheduled maintenance window (在下次排定的維護時段套用)。
8. 選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要修改適用於 SQL Server 資料庫執行個體的 RDS 自訂儲存區，請使用 [modify-db-instance](#) AWS CLI 命令。視需要設定下列參數：

- `--allocated-storage` – 要配置給資料庫執行個體的儲存量 (以 GiB 為單位)。它必須大於目前的值，並且從 20 至 16,384 GiB。
- `--storage-type`— 儲存類型，例如 GP2、gp3、io1 或 io2。
- `--iops` – 資料庫執行個體的佈建 IOPS。您只能為支援佈建 IOPS (gp3、io1 和 io2) 的儲存區類型指定此項目。
- `--apply-immediately` – 使用 `--apply-immediately` 可立即套用儲存變更。

或使用 `--no-apply-immediately` (預設值) 在下個維護時段套用變更。

下列範例會將儲存區大小變更 my-custom-instance 為 200 GiB、儲存區類型為 io1，將已佈建 IOPS 變更為 3000。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --storage-type io1 \  
  --iops 3000 \  
  --apply-immediately
```

```
--allocated-storage 200 \  
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-custom-instance ^  
  --storage-type io1 ^  
  --iops 3000 ^  
  --allocated-storage 200 ^  
  --apply-immediately
```

標記 RDS Custom for SQL Server 資源

您可以將 RDS Custom 資源標記為與 Amazon RDS 資源一樣，但是有一些重要差異：

- 請勿建立或修改 RDS Custom 自動化所需的 AWSRDSCustom 標籤。如果您這樣做，可能會破壞自動化。
- Name 標籤會新增至字首值為 do-not-delete-rds-custom 的 RDS Custom 資源。金鑰的任何客戶傳遞值都會遭到覆寫。
- 在建立期間新增至 RDS Custom 資料庫執行個體的標籤會傳播至所有其他相關的 RDS Custom 資源。
- 當您在建立資料庫執行個體之後將標籤新增至 RDS Custom 資源時，不會傳播標籤。

如需資源標記的一般資訊，請參閱 [標記 Amazon RDS 資源](#)。

刪除 RDS Custom for SQL Server 資料庫執行個體

若要刪除 RDS Custom for SQL Server 資料庫執行個體，請執行以下操作：

- 提供資料庫執行個體的名稱。
- 選擇或清除讓您建立資料庫執行個體最終資料庫快照的選項。
- 選擇或清除保留自動化備份的選項。

您可以使用主控台或 CLI 刪除 RDS Custom for SQL Server 資料庫執行個體。刪除資料庫執行個體所需的時間會因備份保留期間 (也就是要刪除的備份數量)、刪除的資料量，以及是否要拍攝最終快照而有所不同。

⚠ Warning

刪除 RDS Custom for SQL Server 資料庫執行個體會永久刪除 EC2 執行個體和相關聯的 Amazon EBS 磁碟區。您不應隨時終止或刪除這些資源，否則刪除和最終快照建立可能會失敗。

ℹ Note

如果資料庫執行個體的状态為 `creating`、`failed`、`incompatible-create`、`incompatible-restore` 或 `incompatible-network`，則無法建立該資料庫執行個體的最終資料庫快照。如需詳細資訊，請參閱 [檢視 Aurora 叢集中 資料庫執行個體狀態](#)。

⚠ Important

當您選擇建立最終快照時，建議您避免在資料庫執行個體刪除過程中將資料寫入至資料庫執行個體。一旦啟動了資料庫執行個體刪除，最終快照就不保證會擷取資料變更。

主控台

若要刪除 RDS Custom 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要刪除的 RDS Custom for SQL Server 資料庫執行個體。RDS Custom for SQL Server 資料庫執行個體會顯示角色 Instance (RDS Custom for SQL Server) (執行個體 (RDS Custom for SQL Server))。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。
4. 若要建立最終快照，請選擇 Create final snapshot (建立最終快照)，並提供 Final snapshot name (最終快照名稱) 的名稱。
5. 若要保留自動備份，請選擇 Retain automated backups (保留自動備份)。
6. 在方塊中輸入 **delete me**。
7. 選擇 Delete (刪除)。

AWS CLI

您可以使用[delete-db-instance](#) AWS CLI 命令刪除適用於 SQL Server 資料庫執行個體的 RDS 自訂。使用所需的參數 `--db-instance-identifier` 識別資料庫執行個體。其餘參數與 Amazon RDS 資料庫執行個體的參數相同。

下列範例會刪除名為 `my-custom-instance` 的 RDS Custom for SQL Server 資料庫執行個體、建立最終快照，並保留自動化備份。

Example

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier my-custom-instance \  
  --no-skip-final-snapshot \  
  --final-db-snapshot-identifier my-custom-instance-final-snapshot \  
  --no-delete-automated-backups
```

在 Windows 中：

```
aws rds delete-db-instance ^  
  --db-instance-identifier my-custom-instance ^  
  --no-skip-final-snapshot ^  
  --final-db-snapshot-identifier my-custom-instance-final-snapshot ^  
  --no-delete-automated-backups
```

若要建立最終快照，`--final-db-snapshot-identifier` 為必要選項，且必須加以指定。

若要略過最終快照，請在命令中指定 `--skip-final-snapshot` 選項，而非 `--no-skip-final-snapshot` 和 `--final-db-snapshot-identifier` 選項。

若要刪除自動備份，請在命令中指定 `--delete-automated-backups` 選項，而非 `--no-delete-automated-backups` 選項。

啟動與停止 RDS Custom for SQL Server 資料庫執行個體

您可以啟動與停止 RDS Custom for SQL Server 資料庫執行個體。RDS for SQL Server 資料庫執行個體的一般需求和限制，也適用於停止與啟動 RDS Custom for SQL Server 資料庫執行個體。如需詳細資訊，請參閱 [暫時停止 Amazon RDS 資料庫執行個體](#)。

以下考量同樣適用於啟動與停止 RDS Custom for SQL Server 資料庫執行個體：

- 不支援在資料庫執行個體為 STOPPED 時，修改 RDS Custom for SQL Server 資料庫執行個體的 EC2 執行個體屬性。
- 您只能停止和啟動針對單一可用區域設定的 RDS Custom for SQL Server 資料庫執行個體。您無法停止多可用區域組態中的 RDS Custom for SQL Server 資料庫執行個體。
- 當您停止 RDS Custom for SQL Server 資料庫執行個體時，會建立 SYSTEM 快照。當您再次啟動 RDS Custom for SQL Server 資料庫執行個體時，快照會自動刪除。
- 如果您在 RDS Custom for SQL Server 資料庫執行個體停止時刪除 EC2 執行個體，當您再次啟動 RDS Custom for SQL Server 資料庫執行個體時，將會更換 C: 磁碟機。
- 當您停止 RDS Custom for SQL Server 資料庫執行個體時，只要您不修改執行個體類型，C:\ 磁碟機、主機名稱及您的自訂組態就會保留。
- 下列動作會導致 RDS Custom 將資料庫執行個體置於支援周邊外，而您仍需支付資料庫執行個體時數的費用：
 - 在 Amazon RDS 停止時啟動基礎 EC2 執行個體。若要解決問題，您可以呼叫 `start-db-instance` Amazon RDS API，或是停止 EC2，讓 RDS Custom 執行個體恢復 STOPPED。
 - 在 RDS Custom for SQL Server 資料庫執行個體為 ACTIVE 時停止基礎 EC2 執行個體。

如需停止與啟動資料庫執行個體的詳細資訊，請參閱 [暫時停止 Amazon RDS 資料庫執行個體](#)，以及 [啟動之前已停止的 Amazon RDS 資料庫執行個體](#)。

管理 RDS Custom for SQL Server 的多可用區部署

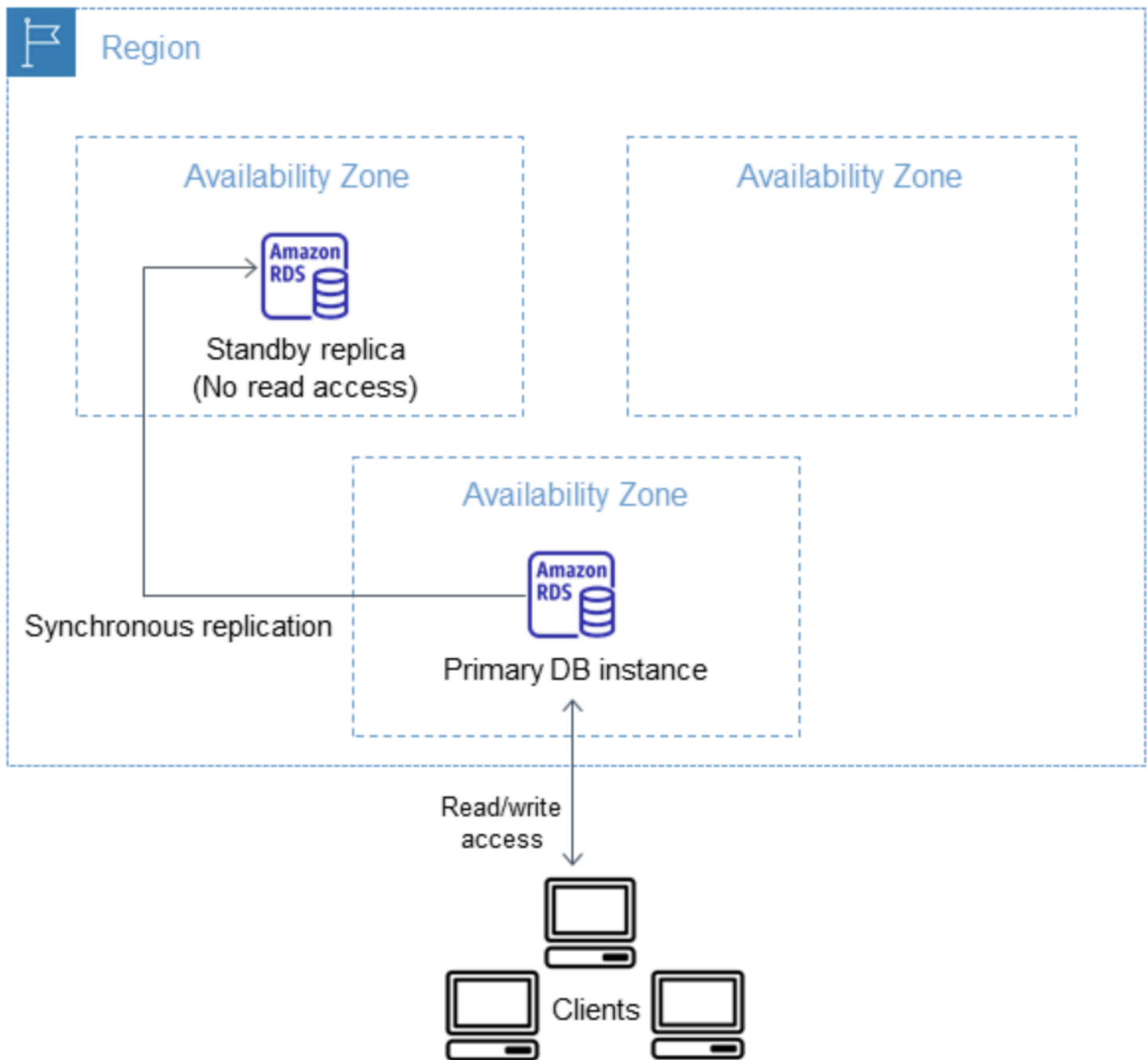
在 RDS Custom for SQL Server 的多可用區域資料庫執行個體部署中，Amazon RDS 會自動佈建，並在不同的可用區域 (AZ) 中維持同步待命複本。主要資料庫執行個體會在待命複本的可用區域間進行同步複製，以提供資料備援。

Important

RDS Custom for SQL Server 的多可用區部署與 RDS for SQL Server 的異地同步備份不同。不同於 RDS for SQL Server 的異地同步備份，您必須先設定 RDS Custom for SQL Server 先決條件，才能建立異地同步備份資料庫執行個體，因為 RDS Custom 會在您自己的帳戶內執行，因此需要使用權限。

若您未完成先決條件，異地同步備份資料庫執行個體就可能無法執行，或自動還原為單一可用區資料庫執行個體。如需先決條件的詳細資訊，請參閱 [RDS Custom for SQL Server 多可用區部署的先決條件](#)。

執行具有高可用性的資料庫執行個體，可以在規劃好的系統維護期間增強可用性。如果發生規劃的資料庫維護或意外服務中斷，Amazon RDS 會自動容錯移轉到 up-to-date 次要資料庫執行個體。此功能可讓資料庫操作迅速恢復，而不需要手動介入。主要和待命執行個體使用相同的端點，而此端點的實體網路地址在容錯移轉過程中會轉移到次要複本。當容錯移轉發生時，您不必重新設定應用程式。



您可以在建立 RDS Custom 資料庫執行個體時指定異地同步備份，以建立 RDS Custom for SQL Server 多可用區部署。您可以使用主控台，藉由修改資料庫執行個體並指定多可用區域選項，將現有 RDS Custom for SQL Server 資料庫執行個體轉換為異地同步備份部署。您也可以使用 AWS CLI 或 Amazon RDS API 來指定多可用區域資料庫執行個體部署。

RDS 主控台會顯示待命複本的可用區域 (次要 AZ)。您也可以使用 `describe-db-instances` CLI 命令或 `DescribeDBInstances` API 操作來尋找次要 AZ。

相較於單一可用區域部署，多可用區部署的 RDS Custom for SQL Server 資料庫執行個體會有增加的寫入和遞交延遲。這可能是資料庫執行個體之間的同步資料複寫造成。儘管 AWS 在可用區域間設計了低延遲網路連線，但如果您的部署容錯移轉到待命複本，延遲時間可能會變更。

Note

對於生產應用程式，我們建議使用佈建 IOPS (每秒輸入/輸出操作) 資料庫執行個體類別來取得快速、一致的效能。如需資料庫執行個體類別的詳細資訊，請參閱 [Amazon RDS Custom for SQL Server 的需求和限制](#)。

主題

- [區域和版本可用性](#)
- [RDS Custom for SQL Server 多可用區部署的限制](#)
- [RDS Custom for SQL Server 多可用區部署的先決條件](#)
- [建立 RDS Custom for SQL Server 多可用區部署](#)
- [將 RDS Custom for SQL Server 單一可用區部署修改為多可用區部署](#)
- [將 RDS Custom for SQL Server 多可用區部署修改為單一可用區部署](#)
- [RDS Custom for SQL Server 多可用區部署的容錯移轉程序](#)
- [使用 RDS Custom for SQL Server 多可用區部署的應用程式存留時間 \(TTL\) 設定](#)

區域和版本可用性

下列 SQL Server 版次支援 RDS Custom for SQL Server 的多可用區部署：

- SQL 伺服器 2022 和 2019 年：企業版、標準版、網頁和開發人員版

Note

SQL 伺服器 2019 CU8 或更低版本不支援 RDS 的異地同步備份部署自訂。

RDS Custom for SQL Server 的多可用區部署可在所有可用 RDS Custom for SQL Server 的區域使用。如需詳細了解 RDS Custom for SQL Server 多可用區部署的區域可用性，請參閱 [適用於 SQL 伺服器的 RDS 自訂支援的區域和資料庫引擎](#)。

RDS Custom for SQL Server 多可用區部署的限制

RDS Custom for SQL Server 多可用區部署具有下列限制：

- 不支援跨區域多可用區部署。
- 您無法設定次要資料庫執行個體來接受資料庫讀取活動。
- 當您搭配使用自訂引擎版本 (CEV) 與多可用區部署時，次要資料庫執行個體也會使用相同的 CEV。次要資料庫執行個體不能使用不同的 CEV。

RDS Custom for SQL Server 多可用區部署的先決條件

如果您有現有的 RDS Custom for SQL Server 單一可用區部署，您需要下列其他先決條件，才能將其修改為多可用區部署。您可以選擇手動或使用提供的 CloudFormation 範本完成先決條件。最新的 CloudFormation 範本包含單一可用區和異地同步備份部署的必要條件。

Important

若要簡化設定，建議您使用網路設定指示中提供的最新 AWS CloudFormation 範本檔案來建立先決條件。如需詳細資訊，請參閱[配置 AWS CloudFormation](#)。

Note

當您將現有 RDS Custom for SQL Server 單一可用區部署修改為多可用區部署時，您必須完成這些先決條件。若您未完成先決條件，不可進行多可用區設定。請遵循 [將 RDS Custom for SQL Server 單一可用區部署修改為多可用區部署](#) 中的步驟以完成先決條件。

- 更新 RDS 安全群組輸入和輸出規則，以允許連接埠 1120。
- 在您的私人網路存取控制清單 (ACL) 中新增規則，以允許資料庫執行個體 VPC 使用 TCP 連接埠 0-65535。
- 建立新的 Amazon SQS VPC 端點，讓 RDS Custom for SQL Server 資料庫執行個體與 SQS 通訊。
- 更新執行個體設定檔角色中的 SQS 權限。

建立 RDS Custom for SQL Server 多可用區部署

若要建立 RDS Custom for SQL Server 多可用區部署，請遵循 [建立並連線至 Amazon RDS Custom for SQL Server 的資料庫執行個體](#) 中的步驟。

Important

若要簡化設定，建議您使用網路設定指示中提供的最新 AWS CloudFormation 範本檔案。如需詳細資訊，請參閱[配置 AWS CloudFormation](#)。

多可用區部署的建立需要幾分鐘。

將 RDS Custom for SQL Server 單一可用區部署修改為多可用區部署

您可以將現有 RDS Custom for SQL Server 資料庫執行個體從單一可用區部署修改為多可用區部署。當您修改資料庫執行個體時，Amazon RDS 會執行多個動作：

- 拍攝主要資料庫執行個體的快照。
- 從快照為待命複本建立新磁碟區。這些磁碟區會在背景中初始化，並在資料完全初始化之後達到最大磁碟區效能。
- 開啟主要和次要資料庫執行個體之間的同步區塊層級複本。

Important

建議您避免在尖峰活動期間，將 RDS Custom for SQL Server 資料庫執行個體從單一可用區修改為生產資料庫執行個體上的多可用區部署。

AWS 使用快照建立待命執行個體，避免從單一可用區轉換為多可用區域的停機時間，但在轉換為多個可用區期間和之後可能會對效能產生影響。對於對寫入延遲敏感的工作負載而言，這個影響可能很重大。雖然此功能可以從快照中快速還原大量磁碟區，但由於是同步複寫，可能會導致 I/O 作業延遲。此延遲可能會影響您的資料庫效能。

主題

- [設定必要條件，以使用以修改單一可用區至異地同步備份部署 CloudFormation](#)
- [設定先決條件，以手動將單一可用區修改為多可用區部署](#)
- [使用 RDS 主控台、AWS CLI 或 RDS API 進行修改。](#)

設定必要條件，以使用以修改單一可用區至異地同步備份部署 CloudFormation

若要使用異地同步備份部署，您必須確定已套用具有先決條件的最新 CloudFormation 範本，或手動設定最新的必要條件。如果您已經套用最新的 CloudFormation 必要條件範本，則可以略過這些步驟。

若要使用設定適用於 SQL Server 異地同步備份部署的 RDS 自訂必要條件 CloudFormation

1. [請在以下位置開啟 CloudFormation 主控台。](https://console.aws.amazon.com/cloudformation) <https://console.aws.amazon.com/cloudformation>
2. 若要啟動「建立堆疊」精靈，請選取您用來建立單一可用區部署的現有堆疊，然後選擇更新。

建立堆疊頁面隨即出現。

3. 對於先決條件 - 準備範本，請選擇取代現有範本。
4. 針對 Specify template 執行下列操作：
 - a. 下載最新的 AWS CloudFormation 範本檔案。開啟 [custom-sqlserver-onboard.zip](#) 連結的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「另存連結為」。
 - b. 將 custom-sqlserver-onboard.json 檔案儲存並擷取到您的電腦。
 - c. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。
 - d. 針對 Choose file (選擇檔案)，導覽至 custom-sqlserver-onboard.json 並加以選擇。
5. 選擇下一步。

Specify stack details (指定堆疊詳細資訊) 頁面隨即出現。

6. 若要保留預設選項，請選擇 Next (下一步)。

進階選項頁面旋即出現。

7. 若要保留預設選項，請選擇 Next (下一步)。
8. 若要保留預設選項，請選擇 Next (下一步)。
9. 在檢閱變更頁面上，執行下列動作：
 - a. 針對 Capabilities (功能)，選取 I acknowledge that AWS CloudFormation might create IAM resources with custom names (確認可能使用自訂名稱建立 IAM 資源) 核取方塊。
 - b. 選擇提交。
10. 確認更新是否成功。成功作業的狀態會顯示 UPDATE_COMPLETE。

如果更新失敗，更新過程指定的任何新組態都會復原。現有的資源仍然可以使用。例如，如果您新增編號為 18 和 19 的網路 ACL 規則，但現有規則的編號相同，則更新會傳回下列錯誤：Resource

handler returned message: "The network acl entry identified by 18 already exists. 在此情況中，您可以將現有 ACL 規則修改為使用小於 18 的數字，然後重試更新。

設定先決條件，以手動將單一可用區修改為多可用區部署

Important

若要簡化設定，建議您使用網路設定指示中提供的最新 AWS CloudFormation 範本檔案。如需詳細資訊，請參閱[設定必要條件，以使用以修改單一可用區至異地同步備份部署 CloudFormation](#)。

如果您選擇手動設定先決條件，請執行以下任務。

1. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 選擇端點。Create Endpoint (建立端點) 頁面隨即出現。
3. 在服務類別中，選擇 AWS 服務。
4. 在服務中，搜尋 **SQS**
5. 在 VPC 中，選擇部署 RDS Custom for SQL Server 資料庫執行個體的 VPC。
6. 在子網路中，選擇部署 RDS Custom for SQL Server 資料庫執行個體的字網路。
7. 在「安全群組」中，選擇「-」vpc-endpoint-sg 群組。
8. 針對政策，選擇自訂
9. 在您的自訂政策中，以您自己的值取代 **AWS ###**、**##**、**accountId** 和 **IAM-Instance-role**。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
        }
      },
      "Action": [
        "SQS:SendMessage",
        "SQS:ReceiveMessage",
        "SQS:DeleteMessage",
        "SQS:GetQueueUrl"
      ]
    }
  ]
}
```

```

        ],
        "Resource": "arn:${AWS::Partition}:sqs:${AWS::Region}:
${AWS::AccountId}:do-not-delete-rds-custom-*",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/{IAM-
Instance-role}"
        }
    }
]
}

```

10. 以 Amazon SQS 存取權限更新執行個體設定檔。以您自己的值取代 *AWS ###*、*##* 和 *accountId*。


```

        {
            "Sid": "SendMessageToSQSQueue",
            "Effect": "Allow",
            "Action": [
                "SQS:SendMessage",
                "SQS:ReceiveMessage",
                "SQS:DeleteMessage",
                "SQS:GetQueueUrl"
            ],
            "Resource": [
                {
                    "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
                }
            ],
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                }
            }
        }
    }
}

```

11. 更新 Amazon RDS 安全群組輸入和輸出規則，以允許連接埠 1120。

- a. 在「安全群組」中，選擇「-」`rds-custom-instance-sg` 群組。
 - b. `##### TCP #####rds-custom-instance-sg##### 1120#`
 - c. 對於輸出規則，請建立自訂 TCP 規則，以允許連接埠 `1120` 連接至目的地 `-rds-custom-instance-sg` 群組。
12. 在您的私人網路存取控制清單 (ACL) 中新增規則，以允許資料庫執行個體的來源子網路使用 TCP 連接埠 `0-65535`。

 Note

建立輸入規則和輸出規則時，請記下現有最高的規則編號。您建立的新規則必須具有小於 100 的規則編號，且不重複任何現有的規則編號。

- a. 在 [網路 ACL] 中，選擇 [-] `private-network-acl` 群組。
- b. 針對輸入規則，請建立所有 TCP 規則，以允許來自來源 `privatesubnet1` 和 `privatesubnet2` 的 TCP 連接埠 `0-65535`。
- c. 針對輸出規則，請建立所有 TCP 規則，以允許到目的地 `privatesubnet1` 和 `privatesubnet2` 的 TCP 連接埠 `0-65535`。

使用 RDS 主控台、AWS CLI 或 RDS API 進行修改。

完成先決條件之後，您可以使用 RDS 主控台、AWS CLI 或 RDS API，將 RDS Custom for SQL Server 資料庫執行個體從單一可用區修改為多可用區部署。

主控台

將現有 RDS Custom for SQL Server 單一可用區修改為多可用區部署

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。

Databases (資料庫) 窗格隨即出現。

3. 選擇您想要修改的 RDS Custom for SQL Server 資料庫執行個體。
4. 從動作中，選擇轉換為多可用區部署。

5. 在確認頁面上，選擇立即套用，來立即套用變更。選擇此選項並不會產生停機時間，但可能會對效能產生影響。或者，您也可以選擇在下個維護時段套用更新。如需詳細資訊，請參閱[排程修改設定](#)。
6. 在確認頁面上，選擇轉換為多可用區。

AWS CLI

若要使用轉換為異地同步備份資料庫執行個體部署AWS CLI，請呼叫[modify-db-instance](#)命令並設定選--multi-az項。指定資料庫執行個體識別符，以及您要修改的其他選項值。如需每個選項的詳細資訊，請參閱[資料庫執行個體的設定](#)。

Example

下列代碼包含 --multi-az 選項，以修改 mycustomdbinstance。使用 --no-apply-immediately，會在下一次維護時段期間由系統套用變更。使用 --apply-immediately 可立即套用變更。如需詳細資訊，請參閱[排程修改設定](#)。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mycustomdbinstance \  
  --multi-az \  
  --no-apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mycustomdbinstance ^  
  --multi-az \ ^  
  --no-apply-immediately
```

RDS API

若要使用 RDS API 轉換為多個可用區資料庫執行個體部署，請呼叫 [ModifyDBInstance](#) 操作，並將 MultiAZ 參數設為 true。

將 RDS Custom for SQL Server 多可用區部署修改為單一可用區部署

您可以將現有 RDS Custom for SQL Server 資料庫執行個體從多可用區部署修改為單一可用區部署。

主控台

將 RDS Custom for SQL Server 資料庫執行個體從多可用區部署修改為單一可用區部署。

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。

Databases (資料庫) 窗格隨即出現。
3. 選擇您想要修改的 RDS Custom for SQL Server 資料庫執行個體。
4. 針對多可用區部署，請選擇否。
5. 在確認頁面上，選擇立即套用，來立即套用變更。選擇此選項並不會產生停機時間，但可能會對效能產生影響。或者，您也可以選擇在下個維護時段套用更新。如需詳細資訊，請參閱[排程修改設定](#)。
6. 在確認頁面上，選擇修改資料庫執行個體。

AWS CLI

若要使用修改單一可用區部署的異地同步備份部署 AWS CLI，請呼叫命 [modify-db-instance](#) 令並包含選項 `--no-multi-az`。指定資料庫執行個體識別符，以及您要修改的其他選項值。如需每個選項的詳細資訊，請參閱[資料庫執行個體的設定](#)。

Example

下列代碼包含 `--no-multi-az` 選項，以修改 `mycustomdbinstance`。使用 `--no-apply-immediately`，會在下一次維護時段期間由系統套用變更。使用 `--apply-immediately` 可立即套用變更。如需詳細資訊，請參閱[排程修改設定](#)。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mycustomdbinstance \  
  --no-multi-az \  
  --no-apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mycustomdbinstance ^
```



```
--no-multi-az \ ^  
--no-apply-immediately
```

RDS API

若要使用 RDS API 將多可用區部署修改為單一可用區部署，請呼叫 [ModifyDBInstance](#) 操作並將參數 MultiAZ 設為 false。

RDS Custom for SQL Server 多可用區部署的容錯移轉程序

如果您已啟用多可用區域，當您的資料庫執行個體因為基礎設施瑕疵而發生計劃中或非計劃的中斷時，Amazon RDS 會自動切換到另一個可用區域中的待命複本。完成容錯移轉所需的時間取決於主要資料庫執行個體失效時的資料庫活動和其他條件。通常容錯移轉時間是 60–120 秒。不過，大型交易或冗長復原程序可能會增加容錯移轉時間。容錯移轉完成後，RDS 控制台使用者界面可能需要多一點時間來顯示新的可用區域。

Note

重新啟動容錯移轉的資料庫執行個體時，您可以手動強制容錯移轉。如需重新啟動資料庫執行個體的相關資訊，請參閱 [重新啟動中的資料庫執行個體](#)

Amazon RDS 會自動處理容錯移轉，因此您可以盡快恢復資料庫操作，而無須人為介入管理。如果發生下表所述的任何條件，主要資料庫執行個體會自動切換至待命複本。您可以在 RDS 事件日誌中檢視這些容錯移轉原因。

容錯移轉原因	描述
The operating system for the RDS Custom for SQL Server Multi-AZ DB instance is being patched in an offline operation	作業系統修補或安全更新的維護期間觸發容錯移轉。如需詳細資訊，請參閱 維持資料庫執行個體 。
The primary host of the RDS Custom for SQL Server Multi-	多可用區域資料庫執行個體部署偵測到主要資料庫執行個體受損並容錯移轉。

容錯移轉原因	描述
AZ DB instance is unhealthy.	
The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unreachable due to loss of network connectivity.	RDS 監控偵測到主要資料庫執行個體的網路連線失敗，並觸發容錯移轉。
The RDS Custom for SQL Server Multi-AZ DB instance was modified by the customer.	資料庫執行個體修改觸發了容錯移轉。如需詳細資訊，請參閱 修改 RDS Custom for SQL Server 資料庫執行個體 。
The storage volume of the primary host of the RDS Custom for SQL Server Multi-AZ DB instance experienced a failure.	多可用區域資料庫執行個體部署在主要資料庫執行個體上偵測到儲存問題並容錯移轉。
The user requested a failover of the RDS Custom for SQL Server Multi-AZ DB instance.	RDS Custom for SQL Server 多可用區資料庫執行個體已透過容錯移轉重新啟動。如需詳細資訊，請參閱 重新啟動中的資料庫執行個體 。

容錯移轉原因	描述
The RDS Custom for SQL Server Multi-AZ primary DB instance is busy or unresponsive.	<p>主要資料庫執行個體沒有回應。建議您採取下列步驟：</p> <ul style="list-style-type: none">• 檢查事件記錄檔和記 CloudWatch 錄檔是否有過多的 CPU、記憶體或交換空間使用量。如需詳細資訊，請參閱使用 Amazon RDS 事件通知。• 建立由 Amazon RDS 事件觸發的規則。如需詳細資訊，請參閱建立由 Amazon RDS 事件觸發的規則。• 評估您的工作負載，以判斷您是否正在使用適當的資料庫執行個體類別。如需詳細資訊，請參閱資料庫執行個體類別。

若要判斷您的多可用區域資料庫執行個體是否已容錯移轉，您可以執行下列動作：

- 設定資料庫事件訂閱，透過電子郵件或 SMS 通知您已啟動容錯移轉。如需事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。
- 您可以使用 RDS 主控台或 API 操作來檢視資料庫事件。
- 您可以使用 RDS 主控台、CLI 和 API 操作，檢視 RDS Custom for SQL Server 多可用區域資料庫執行個體部署的目前狀態。

使用 RDS Custom for SQL Server 多可用區部署的應用程式存留時間 (TTL) 設定

容錯移轉機制會自動將資料庫執行個體的網域名稱系統 (DNS) 記錄變更為指向待命資料庫執行個體。因此，您必須重新建立資料庫執行個體任何現有的連線。請確定任何 DNS 快取 time-to-live (TTL) 組態值都很低，並驗證您的應用程式不會在較長的時間內快取 DNS。高 TTL 值可能導致應用程式在容錯移轉後，無法快速重新連線至資料庫執行個體。

備份與還原 Amazon RDS Custom for SQL Server 資料庫執行個體

與 Amazon RDS 一樣，RDS 自訂會在啟用備份保留時，建立並儲存適用於 SQL Server 資料庫執行個體的 RDS 自訂執行個體的自動備份。您也可以手動備份您的資料庫執行個體。自動備份由快照備份和交易記錄備份組成。在指定的備份時段期間，會針對資料庫執行個體的整個儲存磁碟區進行快照備份。交易記錄備份會定期針對符合 PITR 資格的資料庫進行備份。RDS Custom 會根據您指定的備份保留期，儲存資料庫執行個體的自動備份。您可以使用自動備份將資料庫執行個體復原到備份保留期內的某個時間點。

您也可以手動進行快照備份。您可以隨時從這些快照備份建立新的資料庫執行個體。如需手動建立資料庫快照的詳細資訊，請參閱 [建立 RDS Custom for SQL Server 快照](#)。

雖然快照備份在運作上做為完整備份，但您只需支付增量儲存使用的費用。RDS Custom 資料庫執行個體的第一個快照會包含完整的資料庫執行個體資料。相同資料庫的後續快照均採用遞增處理，這表示系統僅會儲存上次執行快照後有所變更的資料。

主題

- [建立 RDS Custom for SQL Server 快照](#)
- [從 RDS Custom for SQL Server 資料庫快照還原](#)
- [將 RDS Custom for SQL Server 執行個體還原至某個時間點](#)
- [刪除 RDS Custom for SQL Server 快照](#)
- [刪除 RDS Custom for SQL Server 自動備份](#)

建立 RDS Custom for SQL Server 快照

RDS Custom for SQL Server 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。建立 RDS Custom 快照時，請指定要備份哪個 RDS Custom for SQL Server 資料庫執行個體。為快照命名，稍後即可從該快照進行還原。

當您建立快照時，適用於 SQL Server 的 RDS 自訂會為磁碟區建立 Amazon EBS 快照(D:)，也就是連接到資料庫執行個體的資料庫磁碟區。為了讓快照更容易與特定資料庫執行個體產生關聯，它們透過 DBSnapshotIdentifier、DbiResourceId 和 VolumeType 加上標籤。

建立資料庫快照會導致短暫的輸入/輸出暫停。根據資料庫執行個體的大小和類別，暫停時間會持續幾秒鐘至幾分鐘。快照建立時間會隨資料庫的總計數和大小而有所不同。若要深入瞭解適用於時間點還原 (PITR) 作業的資料庫數目，請參閱 [每個執行個體類別類型符合 PITR 資格的資料庫數目](#)。

由於快照包括整個儲存體磁碟區，所以檔案大小 (例如暫存檔案) 也會影響建立快照建立時間。若要進一步了解建立快照，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

使用主控台或 AWS CLI 建立 RDS Custom for SQL Server 快照。

主控台

若要建立 RDS Custom 快照

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 在 RDS Custom 資料庫執行個體清單中，選擇您要建立快照的執行個體。
4. 針對 Actions (動作)，選擇 Take snapshot (建立快照)。

Take DB Snapshot (建立資料庫快照) 視窗隨即顯示。

5. 在 Snapshot name (快照名稱) 中輸入快照的名稱。
6. 選擇 Take Snapshot (擷取快照)。

AWS CLI

您可以使用 [create-db-snapshot](#) AWS CLI 命令建立 RDS 自訂資料庫執行個體的快照。

指定下列選項：

- `--db-instance-identifier` – 識別您要備份的 RDS Custom 資料庫執行個體
- `--db-snapshot-identifier` – 為您的 RDS Custom 快照命名以便稍後從該快照還原

在此範例中，您會為名為 *my-custom-instance* 的 RDS Custom 資料庫執行個體建立名為 *my-custom-snapshot* 的資料庫快照。

Example

對於 Linux macOS、或 Unix：

```
aws rds create-db-snapshot \  
  --db-instance-identifier my-custom-instance \  
  --db-snapshot-identifier my-custom-snapshot
```

在Windows中：

```
aws rds create-db-snapshot ^  
  --db-instance-identifier my-custom-instance ^  
  --db-snapshot-identifier my-custom-snapshot
```

從 RDS Custom for SQL Server 資料庫快照還原

還原 RDS Custom for SQL Server 資料庫執行個體時，您要提供資料庫快照的名稱和新執行個體的名稱。您無法從快照還原至現有的 RDS Custom 資料庫執行個體。還原時，會建立新的 RDS Custom for SQL Server 資料庫執行個體。

從快照還原會將儲存磁碟區還原至拍攝快照的時間點。這將包括(D:)磁碟區上存在的所有資料庫和任何其他檔案。

主控台

若要從資料庫快照還原 RDS Custom 資料庫執行個體

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore snapshot (還原快照)。
5. 在 Restore DB Instance (還原資料庫執行個體) 頁面上，針對 DB Instance Identifier (資料庫執行個體識別符)，輸入您還原的 RDS Custom 資料庫執行個體的名稱。
6. 選擇 Restore DB instance (還原資料庫執行個體)。

AWS CLI

您可以使用 [restore-db-instance-from-db-](#) AWS CLI 快照命令還原 RDS 自訂資料庫快照集。

如果您要從中還原的快照適用於私有資料庫執行個體，請務必指定正確的 `db-subnet-group-name` 和 `no-publicly-accessible`。否則，資料庫執行個體預設為可公開存取。下列選項是必要的：

- `db-snapshot-identifier` – 識別要從中進行還原的快照
- `db-instance-identifier` – 指定要從資料庫快照建立之 RDS Custom 資料庫執行個體的名稱

- `custom-iam-instance-profile` – 指定與 RDS Custom 資料庫執行個體的基礎 Amazon EC2 執行個體關聯的執行個體設定檔。

下列程式碼會為 `my-custom-instance` 還原名為 `my-custom-snapshot` 的快照。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-snapshot-identifier my-custom-snapshot \  
  --db-instance-identifier my-custom-instance \  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \  
  --no-publicly-accessible
```

在Windows中：

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-snapshot-identifier my-custom-snapshot ^  
  --db-instance-identifier my-custom-instance ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^  
  --no-publicly-accessible
```

將 RDS Custom for SQL Server 執行個體還原至某個時間點

您可以將資料庫執行個體還原至特定時間點 (PITR)，建立新的資料庫執行個體。若要支援 PITR，您的資料庫執行個體必須啟用備份保留。

RDS Custom for SQL Server 資料庫執行個體的最晚可還原時間取決於許多因素，但通常是目前時間的 5 分鐘內。若要查看資料庫執行個體的最新可還原時間，請使用 AWS CLI [describe-db-instances](#) 指令並查看資料庫執行個體在 `LatestRestorableTime` 欄位中傳回的值。若要在 Amazon RDS 主控台中查看每個資料庫執行個體的最新可還原時間，請選擇自動備份。

您可以還原至備份保留期間內的任何時間點。若要查看每個資料庫執行個體的最早可還原時間，請在 Amazon RDS 主控台中選擇自動備份。

如需 PITR 的一般資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

主題

- [RDS Custom for SQL Server 的 PITR 考量](#)
- [每個執行個體類別類型符合 PITR 資格的資料庫數目](#)
- [使資料庫不符合 PITR 的資格](#)
- [Amazon S3 中的交易日誌](#)
- [使用AWS Management Console、或 RDS API 進行 AWS CLI PITR 還原。](#)

RDS Custom for SQL Server 的 PITR 考量

在 RDS Custom for SQL Server 中，PITR 與 Amazon RDS 中的 PITR 有以下的重要不同：

- PITR 只會還原資料庫執行個體中的資料庫。它不會還原作業系統或 C: 磁碟機上的檔案。
- 對於 RDS Custom for SQL Server 資料庫執行個體，資料庫會自動備份，而且只有在下列條件下符合 PITR 的資格：
 - 資料庫在線上。
 - 其復原模型設定為 FULL。
 - 可寫入。
 - 在 D: 磁碟機上有實體檔案。
 - 並未列在 `rds_pitr_blocked_databases` 資料表中。如需詳細資訊，請參閱[使資料庫不符合 PITR 的資格](#)。
- 符合 PITR 資格的資料庫由其資料庫 ID 的順序決定。RDS Custom for SQL Server 最多允許每個資料庫執行個體有 5,000 個資料庫。不過，針對 SQL Server 資料庫執行個體的 RDS 自訂，PITR 作業還原的資料庫數目上限取決於執行個體類別類型。如需詳細資訊，請參閱[每個執行個體類別類型符合 PITR 資格的資料庫數目](#)。

不屬於 PITR 的其他資料庫可以從資料庫快照還原，包括用於 PITR 的自動快照備份。

- 新增資料庫、重新命名資料庫或還原符合 PITR 資格的資料庫，會啟動資料庫執行個體的快照。
- 當資料庫執行個體執行擴展運算作業時，符合 PITR 資格的資料庫數目上限會變更，視目標執行個體類別類型而定。如果執行個體擴充，允許執行個體上的更多資料庫符合 PITR 的資格，就會建立新的快照集。
- 還原的資料庫名稱與來源資料庫執行個體中的名稱相同。您無法指定不同的名稱。
- `AWSRDSCustomSQLServerIamRolePolicy` 需要訪問其他 AWS 服務。如需詳細資訊，請參閱[將存取原則新增至 AWSRDSCustomSQLServerInstanceRole](#)。
- 不支援 RDS Custom for SQL Server 的時區變更。如果您變更作業系統或資料庫執行個體時區，PITR (和其他自動化) 無法運作。

每個執行個體類別類型符合 PITR 資格的資料庫數目

下表顯示根據執行個體類別類型，符合 PITR 資格的資料庫數目上限。

執行個體類別類型	符合 PITR 資格的資料庫數目上限				
db.*.large	100				
數據庫 *.x 大到數據庫 *.2XL	150				
分貝 *.4X 大 到數據庫 *.8	300				
數據庫 *.12 大到數據庫	600				
分貝 *.24 倍 大，分貝 . 32 倍大	1000				

*代表不同的實體類別類型。

資料庫執行個體上符合 PITR 資格的資料庫數目上限取決於執行個體類別類型。在適用於 SQL Server 的 RDS 自訂所支援的最大執行個體類別類型上，數字範圍從最小 100 到 1000 個不等。SQL 伺服器系統資料庫(master, model, msdb, tempdb)不包含在此限制中。擴展或縮減資料庫執行個體時，RDS Custom 會根據目標執行個體類別類型自動更新符合 PITR 資格的資料庫數量。RDS-EVENT-0352當資料庫執行個體上符合 PITR 資格的資料庫數目上限變更時，SQL Server 的 RDS 自訂會傳送。如需詳細資訊，請參閱[自訂引擎版本事件](#)。

Note

只有在 2023 年 8 月 26 日之後建立的資料庫執行個體才能支援 100 個以上的資料庫執行個體。對於 2023 年 8 月 26 日之前建立的執行個體，無論執行個體類別為何，符合 PITR 資格的資料庫數目上限都是 100。若要在 2023 年 8 月 26 日之前建立的資料庫執行個體上啟用 100 多個資料庫的 PITR 支援，您可以執行下列動作：

- 將資料庫引擎版本升級至 15.00.4322.v1 或更新版本

在 PITR 作業期間，RDS Custom 會在還原時還原來源資料庫執行個體上屬於 PITR 一部分的所有資料庫。目標資料庫執行個體完成還原作業後，如果啟用備份保留，資料庫執行個體將根據目標資料庫執行個體上符合 PITR 資格的資料庫數目上限開始備份。

例如，如果您的資料庫執行個體在具有 200 個資料庫 db.*.xlarge 的執行個體上執行：

1. 適用於 SQL 伺服器的 RDS 自訂會選擇前 150 個資料庫 (依其資料庫識別碼排序) 進行 PITR 備份。
2. 您可以修改例證，以向上縮放至 db.*.4xlarge。
3. 調整計算作業完成後，SQL Server 的 RDS 自訂會選擇前 300 個資料庫 (依其資料庫識別碼排序) 進行 PITR 備份。符合 PITR 要求條件的 200 個資料庫中的每一個資料庫都有資格申請 PITR。
4. 現在，您可以修改例證以縮小至 db.*.xlarge。
5. 調整計算作業完成後，SQL Server 的 RDS 自訂將再次選取前 150 個資料庫 (依其資料庫識別碼排序) 進行 PITR 備份。

使資料庫不符合 PITR 的資格

您可以選擇從 PITR 排除個別資料庫。若要做到這一點，請將它們的 `database_id` 值放到 `rds_pitr_blocked_databases` 資料表。使用下列 SQL 指令碼以建立資料表。

若要建立 `rds_pitr_blocked_databases` 資料表

- 執行以下 SQL 指令碼。

```
create table msdb..rds_pitr_blocked_databases
(
  database_id INT NOT NULL,
  database_name SYSNAME NOT NULL,
  db_entry_updated_date datetime NOT NULL DEFAULT GETDATE(),
  db_entry_updated_by SYSNAME NOT NULL DEFAULT CURRENT_USER,
  PRIMARY KEY (database_id)
);
```

如需符合資格和不符合資格資料庫的清單，請參閱 Amazon S3 儲存貯體 `do-not-delete-rds-custom-$ACCOUNT_ID-$REGION-unique_identifier` 的 `RDSCustomForSQLServer/`

Instances/*DB_instance_resource_ID*/TransactionLogMetadata 目錄中的 RI.End 檔案。如需 RI.End 詳細資訊，請參閱 [Amazon S3 中的交易日誌](#)。

您也可以使用下列 SQL 指令碼判斷 PITR 適用的資料庫清單。將 @limit 變數設定為符合執行個體類別 PITR 資格的資料庫數目上限。如需詳細資訊，請參閱 [每個執行個體類別類型符合 PITR 資格的資料庫數目](#)。

判斷資料庫執行個體類別上 PITR 合格的資料庫清單

- 執行以下 SQL 指令碼。

```
DECLARE @Limit INT;
SET @Limit = (insert-database-instance-limit-here);

USE msdb;
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND
TABLE_NAME = 'rds_pitr_blocked_databases'))
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
        'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
(sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
        SELECT dbs.database_id as DatabaseId, sysdbs.name as DatabaseName,
        'OPTOUT' as Reason,
        CASE WHEN dbs.database_name = sysdbs.name THEN NULL ELSE
dbs.database_name END AS DatabaseNameOnPitrTable
        FROM msdb.dbo.rds_pitr_blocked_databases dbs
        INNER JOIN sys.databases sysdbs ON dbs.database_id = sysdbs.database_id
        WHERE sysdbs.database_id > 4
    ),
    TABLE2 as (
        SELECT
        db.name AS DatabaseName,
        db.create_date AS CreateDate,
        db.state_desc AS DatabaseState,
        db.database_id AS DatabaseId,
        rs.database_guid AS DatabaseGuid,
        rs.last_log_backup_lsn AS LastLogBackupLSN,
```

```

rs.recovery_fork_guid RecoveryForkGuid,
rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
db.recovery_model_desc AS RecoveryModel,
db.is_auto_close_on AS IsAutoClose,
db.is_read_only as IsReadOnly,
NEWID() as FileName,
CASE WHEN(db.state_desc = 'ONLINE'
          AND db.recovery_model_desc != 'SIMPLE'
          AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
          AND db.is_read_only != 1
          AND db.user_access = 0
          AND db.source_database_id IS NULL
          AND db.is_in_standby != 1
          THEN 1 ELSE 0 END AS IsPartOfSnapshot,
CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
FROM sys.databases db
INNER JOIN sys.database_recovery_status rs
ON db.database_id = rs.database_id
WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE1) AND
db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
),
TABLE3 as(
    Select @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE2
where TABLE2.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
)
SELECT TOP(SELECT TotalNumberOfDatabases from TABLE3)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE2 where
TABLE2.IsPartOfSnapshot=1
ORDER BY TABLE2.DatabaseID ASC
ELSE
WITH TABLE0 AS (
    SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
FROM sys.dm_hadr_database_replica_states hdrs
INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
(sdb.name, 'Updateability') = 'READ_ONLY')
),
TABLE1 as (
    SELECT

```

```

        db.name AS DatabaseName,
        db.create_date AS CreateDate,
        db.state_desc AS DatabaseState,
        db.database_id AS DatabaseId,
        rs.database_guid AS DatabaseGuid,
        rs.last_log_backup_lsn AS LastLogBackupLSN,
        rs.recovery_fork_guid AS RecoveryForkGuid,
        rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
        db.recovery_model_desc AS RecoveryModel,
        db.is_auto_close_on AS IsAutoClose,
        db.is_read_only as IsReadOnly,
        NEWID() as FileName,
        CASE WHEN(db.state_desc = 'ONLINE'
                AND db.recovery_model_desc != 'SIMPLE'
                AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
                AND db.is_read_only != 1
                AND db.user_access = 0
                AND db.source_database_id IS NULL
                AND db.is_in_standby != 1
                THEN 1 ELSE 0 END AS IsPartOfSnapshot,
        CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
    FROM sys.databases db
    INNER JOIN sys.database_recovery_status rs
    ON db.database_id = rs.database_id
    WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
    db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
),
TABLE2 as(
    SELECT @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE1
where TABLE1.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
)
select top(select TotalNumberOfDatabases from TABLE2)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE1 where
TABLE1.IsPartOfSnapshot=1
ORDER BY TABLE1.DatabaseID ASC

```

Note

只有符號連結的資料庫也會從符合 PITR 作業資格的資料庫中排除。上述查詢不會根據此條件進行篩選。

Amazon S3 中的交易日誌

備份保留期決定是否會自動擷取 RDS Custom for SQL Server 資料庫執行個體的交易日誌並上傳到 Amazon S3。非零值表示會建立自動備份，而且 RDS Custom 代理程式每 5 分鐘會將交易日誌上傳到 S3。

S3 上的交易日誌檔會使用您在建立資料庫執行個體時提供的 AWS KMS key 進行靜態加密。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用伺服器端加密保護資料](#)。

每個資料庫的交易日誌會上傳至名稱為 `do-not-delete-rds-custom-$ACCOUNT_ID-$REGION-unique_identifier` 的 S3 儲存貯體。S3 儲存貯體中的 `RDSCustomForSQLServer/Instances/DB_instance_resource_ID` 目錄包含兩個子目錄：

- TransactionLogs – 包含每個資料庫的交易日誌及其各自的中繼資料。

交易日誌檔名稱遵循 `yyyyMMddHHmm.database_id.timestamp` 模式，例如：

```
202110202230.11.1634769287
```

具有尾碼 `_metadata` 的相同檔案名稱包含交易日誌的相關資訊，例如日誌序號、資料庫名稱和 `RdsChunkCount`。`RdsChunkCount` 會決定多少實體檔案表示單一交易日誌檔。您可能會看到具有尾碼 `_0001`、`_0002` 等等的檔案，這表示交易日誌檔的實際區塊。如果您想要使用區塊的交易日誌檔，請務必在下載區塊之後合併。

請考量具有下列檔案的案例：

- 202110202230.11.1634769287
- 202110202230.11.1634769287_0001
- 202110202230.11.1634769287_0002
- 202110202230.11.1634769287_metadata

`RdsChunkCount` 為 3。合併檔案的順序如下：202110202230.11.1634769287、202110202230.11.1634769287_0001、202110202230.11.1634769287_0002。

- `TransactionLogMetadata` – 包含交易日誌擷取之每個反覆項目的相關中繼資料資訊。

RI.End 檔案包含其交易日誌已擷取之所有資料庫的資訊，以及已存在但是其交易日誌未擷取之所有資料庫的資訊。RI.End 檔案名稱遵循 `yyyyMMddHHmm.RI.End.timestamp` 模式，例如：

```
202110202230.RI.End.1634769281
```

使用 AWS Management Console、或 RDS API 進行 AWS CLI PITR 還原。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 將 RDS Custom for SQL Server 資料庫執行個體還原至某個時間點。

主控台

若要將 RDS Custom 資料庫執行個體還原至指定時間

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 選擇您想要還原的 RDS Custom 資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。

Restore to point in time (還原到時間點) 視窗隨即出現。

5. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇 Custom (自訂)，請輸入您希望執行個體叢集還原到什麼日期和時間。

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

6. 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入目標還原 RDS Custom 資料庫執行個體的名稱。名稱必須是唯一的。
7. 視需要選擇其他選項，例如資料庫執行個體類別。
8. 選擇 Restore to point in time (還原至時間點)。

AWS CLI

您可以使用 [restore-db-instance-to-point-in-time](#) AWS CLI 命令建立新的 RDS 自訂資料庫執行個體，將資料庫執行個體還原到指定的時間。

使用下列其中一個選項來指定要從中還原的備份：

- `--source-db-instance-identifier` *mysourcedbinstance*
- `--source-dbi-resource-id` *dbinstanceresourceID*
- `--source-db-instance-automated-backups-arn` *backupARN*

`custom-iam-instance-profile` 選項為必要項目。

下列範例會將 `my-custom-db-instance` 還原至名為 `my-restored-custom-db-instance` 的新資料庫執行個體，截至指定的時間。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier my-custom-db-instance \  
  --target-db-instance-identifier my-restored-custom-db-instance \  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \  
  --restore-time 2022-10-14T23:45:00.000Z
```

在Windows中：

```
aws rds restore-db-instance-to-point-in-time ^  
  --source-db-instance-identifier my-custom-db-instance ^  
  --target-db-instance-identifier my-restored-custom-db-instance ^  
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^  
  --restore-time 2022-10-14T23:45:00.000Z
```

刪除 RDS Custom for SQL Server 快照

當您不再需要時，可以刪除由 RDS Custom for SQL Server 管理的資料庫快照。Amazon RDS 與 RDS Custom 資料庫執行個體的刪除程序相同。

二進位和根磁碟區的 Amazon EBS 快照會在您的帳戶中保留較長的時間，因為這些快照可能會連結到您帳戶中執行的某些執行個體或其他 RDS Custom for SQL Server 快照。這些 EBS 快照不再與任何現有 RDS Custom for SQL Server 資源 (資料庫執行個體或備份) 相關之後，會自動刪除這些快照。

主控台

若要刪除 RDS Custom 資料庫執行個體的快照

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要刪除的資料庫快照。
4. 針對 Actions (動作)，選擇 Delete Snapshot (刪除快照)。
5. 在確認頁面上，選擇 Delete (刪除)。

AWS CLI

若要刪除 RDS 自訂快照集，請使用 AWS CLI 命令 [delete-db-snapshot](#)。

下列選項是必要的：

- `--db-snapshot-identifier` – 要刪除的快照

以下範例會刪除 `my-custom-snapshot` 資料庫快照。

Example

對於 Linux/macOS、或 Unix：

```
aws rds delete-db-snapshot \  
  --db-snapshot-identifier my-custom-snapshot
```

在 Windows 中：

```
aws rds delete-db-snapshot ^  
  --db-snapshot-identifier my-custom-snapshot
```

刪除 RDS Custom for SQL Server 自動備份

不再需要保留 RDS Custom for SQL Server 的自動備份時，刪除即可。該程序與刪除 Amazon RDS 備份的程序相同。

主控台

刪除保留的自動備份

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
3. 選擇 Retained (已保留)。
4. 選擇您要刪除的已保留自動備份。
5. 對於 Actions (動作)，請選擇 Delete (刪除)。
6. 在確認頁面上，輸入 **delete me**，然後選擇 Delete (刪除)。

AWS CLI

您可以使用 AWS CLI 命令 [delete-db-instance-automated-backup](#) 刪除保留的自動備份。

下列選項用來刪除保留的自動備份：

- `--dbi-resource-id` – 來源 RDS Custom 資料庫執行個體的資源識別符。

您可以使用 AWS CLI 指令 [describe-db-instance-automated-backup](#)，找到保留自動備份的來源資料庫執行個體的資源識別碼。

下列範例刪除來源資料庫執行個體識別符為 `custom-db-123ABCEXAMPLE` 的已保留自動備份。

Example

對於 Linux macOS、或 Unix：

```
aws rds delete-db-instance-automated-backup \  
  --dbi-resource-id custom-db-123ABCEXAMPLE
```

在 Windows 中：

```
aws rds delete-db-instance-automated-backup ^  
  --dbi-resource-id custom-db-123ABCEXAMPLE
```

將內部部署資料庫遷移到 Amazon RDS Custom for SQL Server

您可以使用下列程序，使用原生備份和還原將內部部署 Microsoft SQL Server 資料庫遷移到 Amazon RDS Custom for SQL Server：

1. 對內部部署資料庫執行個體上的資料庫進行完整備份。
2. 將備份檔案上傳到 Amazon S3。
3. 將備份檔案從 S3 下載至 RDS Custom for SQL Server 資料庫執行個體。
4. 使用 RDS Custom for SQL Server 資料庫執行個體上下載的備份檔案還原資料庫。

此程序說明如何使用原生完整備份和還原，將資料庫從內部部署遷移到 RDS Custom for SQL Server。若要減少遷移程序期間的切換時間，您也可以考慮使用差異或日誌備份。

如需 RDS for SQL Server 其原生備份和還原的一般資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

主題

- [先決條件](#)
- [備份內部部署資料庫](#)
- [將備份檔案上傳到 Amazon S3](#)
- [從 Amazon S3 下載備份檔案](#)
- [將備份檔案還原至 RDS Custom for SQL Server 資料庫執行個體](#)

先決條件

在遷移資料庫之前，請先執行以下任務：

1. 為 RDS Custom for SQL Server 資料庫執行個體設定遠端桌面連線 (RDP)。如需詳細資訊，請參閱 [使用 RDS 連線到您的 RDS Custom 資料庫執行個體](#)。
2. 設定對 Amazon S3 的存取權限，以便上傳和下載資料庫備份檔案。如需詳細資訊，請參閱 [整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

備份內部部署資料庫

您可以使用 SQL Server 原生備份，在內部部署資料庫執行個體上取得資料庫的完整備份。

下列範例顯示名為 `mydatabase` 的資料庫備份，已指定 `COMPRESSION` 選項以減少備份檔案大小。

若要備份內部部署資料庫

1. 使用 SQL Server Management Studio (SSMS)，連接至內部部署 SQL Server 執行個體。
2. 執行下列 T-SQL 命令。

```
backup database mydatabase to  
disk = 'C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\mydb-  
full-compressed.bak'  
with compression;
```

將備份檔案上傳到 Amazon S3

您可以使用 AWS Management Console 將備份檔案 `mydb-full-compressed.bak` 上傳到 Amazon S3。

若要將備份檔案上傳到 S3

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台。
2. 針對 Buckets (儲存貯體)，選擇您要上傳檔案的目標儲存貯體名稱。
3. 選擇 Upload (上傳)。
4. 在 Upload (上傳) 視窗中，執行下列其中一個操作：
 - 將 `mydb-full-compressed.bak` 拖放到 Upload (上傳) 視窗。
 - 選擇 Add file (新增檔案)、選擇 `mydb-full-compressed.bak`，然後選擇 Open (開啟)。

Amazon S3 會將備份檔案上傳為 S3 物件。上傳完成後，您可以在 Upload: status (上傳：狀態) 頁面上看到成功訊息。

從 Amazon S3 下載備份檔案

您可以使用主控台將備份檔案從 S3 下載到 RDS Custom for SQL Server 資料庫執行個體。

若要從 S3 下載備份檔案

1. 使用 RDP，連接至 RDS Custom for SQL Server 資料庫執行個體。

2. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台。
3. 在 Buckets (儲存貯體) 清單中，選擇其中包含您的備份檔案的儲存貯體名稱。
4. 選擇備份檔案 mydb-full-compressed.bak。
5. 針對 Actions (動作)，選擇 Download as (下載為)。
6. 開啟提供之連結的內容 (右鍵) 功能表，然後選擇 Save As (另存新檔)。
7. 將 mydb-full-compressed.bak 儲存到 D:\rdsdbdata\BACKUP 目錄。

將備份檔案還原至 RDS Custom for SQL Server 資料庫執行個體

您使用 SQL Server 原生還原將備份檔案還原至 RDS Custom for SQL Server 資料庫執行個體。

在此範例中會指定 MOVE 選項，因為資料和日誌檔目錄與內部部署資料庫執行個體不同。

若要還原備份檔案

1. 使用 SSMS，連接至 RDS Custom for SQL Server 資料庫執行個體。
2. 執行下列 T-SQL 命令。

```
restore database mydatabase from disk='D:\rdsdbdata\BACKUP\mydb-full-  
compressed.bak'  
with move 'mydatabase' to 'D:\rdsdbdata\DATA\mydatabase.mdf',  
move 'mydatabase_log' to 'D:\rdsdbdata\DATA\mydatabase_log.ldf';
```

升級 Amazon RDS Custom for SQL Server 的資料庫執行個體

您可以藉由將 Amazon RDS Custom for SQL Server 資料庫執行個體修改為新的資料庫引擎版本來進行升級，就像是對 Amazon RDS 所做的一樣。

升級 RDS Custom for SQL Server 資料庫執行個體的相同限制，一般而言也適用於修改 RDS Custom for SQL Server 資料庫執行個體。如需詳細資訊，請參閱 [修改 RDS Custom for SQL Server 資料庫執行個體](#)。

如需與升級資料庫執行個體相關的一般資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

如果您在異地同步備份部署中升級適用於 SQL Server 資料庫的 RDS 自訂執行個體，則 Amazon RDS 會執行滾動式升級，因此您只會在容錯移轉期間發生中斷。如需詳細資訊，請參閱 [多可用區域和記憶體內最佳化的考量](#)。

主要版本升級

適用於 SQL 伺服器的 Amazon RDS 自訂目前支援以下主要版本升級。

目前版本	支援的升級版本
SQL Server 2019	SQL 伺服器

您可以使用 AWS CLI 查詢 (例如下列範例) 來尋找特定資料庫引擎版本的可用升級。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine sqlserver-se \  
  --engine-version 15.00.4322.2.v1 \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \  
  --output table
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --engine sqlserver-se ^
```

```
--engine-version 15.00.4322.2.v1 ^  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^  
--output table
```

資料庫相容性等級

您可以使用 Microsoft SQL Server 資料庫相容性等級來調整某些資料庫行為，以模仿舊版的 SQL Server。如需詳細資訊，請參閱 Microsoft 文件中的[相容性等級](#)。

升級資料庫執行個體時，所有現有的資料庫都會保持其原來的相容性等級。例如，如果您從 SQL 伺服器 2019 升級到 SQL 伺服器 2022，則所有現有資料庫的相容性層級都是 150。升級後建立的任何新資料庫都具有相容性等級 160。

您可以使用 ALTER DATABASE 命令來變更資料庫的相容性等級。例如，若要變更名 customeracct 為與 SQL Server 2022 相容的資料庫，請執行下列命令：

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 160
```


針對 Amazon RDS Custom for SQL Server 的資料庫問題進行故障診斷

RDS Custom 的共同的責任模型提供作業系統殼層層級存取權和資料庫管理員存取權。RDS Custom 會執行您的帳戶中的資源，不像是 Amazon RDS，它會執行系統帳戶中的資源。責任越大，存取權也越大。在下列各節中，您可以了解如何針對 Amazon RDS Custom for SQL Server 資料庫執行個體的問題進行故障診斷。

Note

本節說明如何對 RDS Custom for SQL Server 進行故障診斷。若要對 RDS Custom for Oracle 進行故障診斷，請參閱 [針對 Amazon RDS Custom for Oracle 的資料庫問題進行故障診斷](#)。

主題

- [檢視 RDS Custom 事件](#)
- [訂閱 RDS 自訂事件](#)
- [針對 RDS Custom for SQL Server 的 CEV 錯誤進行疑難排解](#)
- [修正 RDS Custom for SQL Server 中不支援的組態](#)
- [SQL 伺服器 RDS 自訂 Storage-Full 中的疑難排解](#)

檢視 RDS Custom 事件

RDS Custom 和 Amazon RDS 資料庫執行個體檢視事件程序相同。如需詳細資訊，請參閱 [檢視 Amazon RDS 事件](#)。

若要使用檢視 RDS 自訂事件通知 AWS CLI，請使用 `describe-events` 命令。RDS Custom 引入數個新事件。事件類別與 Amazon RDS 的類別相同。如需事件清單，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

下列範例會擷取針對指定 RDS Custom 資料庫執行個體發生之事件的詳細資訊。

```
aws rds describe-events \  
  --source-identifier my-custom-instance \  
  --source-type db-instance
```

訂閱 RDS 自訂事件

RDS Custom 與 Amazon RDS 資料庫執行個體的訂閱事件程序相同。如需詳細資訊，請參閱 [訂閱 Amazon RDS 事件通知](#)。

若要使用 CLI 來訂閱 RDS Custom 事件通知，請使用 `create-event-subscription` 命令。包含下列必要參數：

- `--subscription-name`
- `--sns-topic-arn`

下列範例會在目前 AWS 帳戶中建立 RDS Custom 資料庫執行個體之備份和復原事件的訂閱。通知會傳送至 Amazon Simple Notification Service (Amazon SNS) 主題，由 `--sns-topic-arn` 指定。

```
aws rds create-event-subscription \  
  --subscription-name my-instance-events \  
  --source-type db-instance \  
  --event-categories '["backup","recovery"]' \  
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

針對 RDS Custom for SQL Server 的 CEV 錯誤進行疑難排解

當您嘗試建立 CEV 時，其可能會失敗。在此情況下，RDS Custom 會發出 RDS-EVENT-0198 事件訊息。如需檢視 RDS 事件的詳細資訊，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

使用下列資訊來協助您處理可能的原因。

訊息	故障診斷建議
Custom Engine Version creation expected a Sysprep'd AMI. Retry creation using a Sysprep'd AMI.	在您從 AMI 建立的 EC2 執行個體上執行 Sysprep。如需使用 Sysprep 準備 AMI 的詳細資訊，請參閱 使用 Sysprep 建立標準化 Amazon Machine Image (AMI) 。
EC2 Image permissions for image (AMI_ID) weren't found for customer (Customer_ID). Verify	驗證用於建立的帳戶和設定檔是否在所選 AMI 的 <code>create EC2 Instance</code> 和 <code>Describe Images</code> 上具有所需的許可。

訊息	故障診斷建議		
customer (Customer_ID) has valid permissions on the EC2 Image.			
Failed to rebuild databases with server collation (collation name) due to missing setup.exe file for SQL Server.	確認 setup 檔案位於 C:\Program Files\Microsoft SQL Server\... \SQLnnnn\setup.exe 。		
Image (AMI_ID) doesn't exist in your account (ACCOUNT_ID). Verify (ACCOUNT_ID) is the owner of the EC2 image.	確定 AMI 存在於相同的客戶帳戶中。		
Image id (AMI_ID) isn't valid. Specify a valid image id, and try again.	AMI 的名稱不正確。確定提供了正確的 AMI ID。		
Image (AMI_ID) operating system platform isn't supported. Specify a valid image, and try again.	<p>選擇具有 Windows Server 搭配 SQL Server Enterprise、Standard 或 Web Edition 的受支援 AMI。從 EC2 Marketplace 選擇具有下列其中一個使用操作的 AMI：</p> <ul style="list-style-type: none"> • RunInstances:0102-使用 SQL 伺服器企業版的視窗 • RunInstances:0006-使用 SQL 伺服器標準的視窗 • RunInstances:0202-使用 SQL 伺服器網頁版的視窗 		

訊息	故障診斷建議		
<p>SQL Server Web Edition isn't supported for creating a Custom Engine Version using Bring Your Own Media. Specify a valid image, and try again.</p>	<p>使用包含受支援 SQL Server 版本的 AMI。如需詳細資訊，請參閱 RDS Custom for SQL Server CEV 的版本支援。</p>		
<p>The custom engine version can't be the same as the OEV engine version. Specify a valid CEV, and try again.</p>	<p>不支援傳統 RDS Custom for SQL Server 引擎版本。例如，版本 15.00.4073.23.v1。使用支援的版本號碼。</p>		
<p>The custom engine version isn't in an active state. Specify a valid CEV, and try again.</p>	<p>CEV 必須處於 AVAILABLE 狀態才能完成操作。將 CEV 從 INACTIVE 修改為 AVAILABLE。</p>		
<p>The custom engine version isn't valid for an upgrade. Specify a valid CEV with an engine version greater or equal to (X), and try again.</p>	<p>目標 CEV 無效。檢查有效升級路徑的需求。</p>		
<p>The custom engine version isn't valid. Names can include only lowercase letters (a-z), dashes (-), underscores (_), and periods (.). Specify a valid CEV, and try again.</p>	<p>遵循所需的 CEV 命名慣例。如需詳細資訊，請參閱 RDS Custom for SQL Server CEV 的要求。</p>		

訊息	故障診斷建議		
<p>The custom engine version isn't valid. Specify valid database engine version, and try again. Example: 15.00.4073.23-cev123.</p>	<p>已提供不受支援的 DB 引擎版本。請使用支援的資料庫引擎版本。</p>		
<p>The expected architecture is (X) for image (AMI_ID), but architecture (Y) was found.</p>	<p>使用建置在 x86_64 架構上的 AMI。</p>		
<p>The expected owner of image (AMI_ID) is customer account ID (ACCOUNT_ID), but owner (ACCOUNT_ID) was found.</p>	<p>從您對其具有許可的 AMI 建立 EC2 執行個體。在 EC2 執行個體上執行 Sysprep，以建立並儲存基礎映像。</p>		
<p>The expected platform is (X) for image (AMI_ID), but platform (Y) was found.</p>	<p>使用透過 Windows 平台建置的 AMI。</p>		
<p>The expected root device type is (X) for image %s, but root device type (Y) was found.</p>	<p>使用 EBS 裝置類型建立 AMI。</p>		

訊息	故障診斷建議		
<p>The expected SQL Server edition is (X), but (Y) was found.</p>	<p>選擇具有 Windows Server 搭配 SQL Server Enterprise、Standard 或 Web Edition 的受支援 AMI。從 EC2 Marketplace 選擇具有下列其中一個使用操作的 AMI：</p> <ul style="list-style-type: none"> • RunInstances:0102-使用 SQL 伺服器企業版的視窗 • RunInstances:0006-使用 SQL 伺服器標準的視窗 • RunInstances:0202-使用 SQL 伺服器網頁版的視窗 		
<p>The expected state is (X) for image (AMI_ID), but the following state was found: (Y).</p>	<p>確定 AMI 處於 AVAILABLE 的狀態。</p>		
<p>The provided Windows OS name (X) isn't valid. Make sure the OS is one of the following: (Y).</p>	<p>使用支援的 Windows 作業系統。</p>		
<p>Unable to find bootstrap log file in path.</p>	<p>確認日誌檔案位於 C:\Program Files\Microsoft SQL Server\{nnn}\Setup Bootstrap\Log\Ssummary.txt。</p>		
<p>RDS expected a Windows build version greater than or equal to (X), but found version (Y)..</p>	<p>使用最低作業系統組建版本為 14393 的 AMI。</p>		

訊息	故障診斷建議
RDS expected a Windows major version greater than or equal to (X), but found version (Y)..	使用最低作業系統主要版本為 10.0 或更新版本的 AMI。

修正 RDS Custom for SQL Server 中不支援的組態

由於採取共同的責任模型，您必須負責修正讓 RDS Custom for SQL Server 資料庫執行個體進入 unsupported-configuration 狀態的組態問題。如果問題出在 AWS 基礎結構上，您可以使用主控台或修正此問題。AWS CLI 如果問題與作業系統或資料庫組態相關，您可以登入主機進行修正。

Note

本節說明如何修正 RDS Custom for SQL Server 中不支援的組態。如需 RDS Custom for Oracle 的相關資訊，請參閱 [修正 RDS Custom for Oracle 中不支援的組態](#)。

您可以在下表中找到支援周邊所傳送通知和事件的說明，以及了解如何進行修正。這些通知和支援周邊可能會變更。如需支援周邊的相關背景，請參閱 [RDS Custom 支援周邊](#)。如需事件說明，請參閱 [Amazon RDS 事件類別和事件訊息](#)。

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S0000	手動不支援的組	由於以下原因，RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]：X	若要解決此問題，請建立支援案例。
AWS 資源 (基礎架構)			
SP-S1001	EC2 執行個體狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：基礎 EC2 執行個體 %s 在未停止	若要檢查資料庫執行個體的狀態，請使用主控台或執行下列 AWS CLI 命令：

事件代碼	組態區域	RDS 事件訊息	驗證程序
		<p>RDS 執行個體的情況下已停止。您可以啟動基礎 EC2 執行個體，並確保已連接二進位和資料磁碟區來解決此問題。如果您打算停止 RDS 執行個體，請先確定基礎 EC2 執行個體處於 [可用] 狀態，然後使用 RDS 主控台或 CLI 停止 RDS 執行個體。</p>	<pre>aws rds describe-db-instances \ --db-instance-identifier db-instance-name grep DBInstanceStatus</pre>
SP-S1002	EC2 執行個體狀態	<p>RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：RDS 資料庫執行個體狀態設定為，STOPPED但基礎 EC2 執行個體 %s 已啟動。您可以停止基礎 EC2 執行個體來解決此問題。如果您打算啟動 RDS 執行個體，請使用主控台或 CLI。</p>	<p>使用下列 AWS CLI 命令檢查資料庫執行個體的狀態：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep DBInstanceStatus</pre> <p>您也可以使用 EC2 主控台查看 EC2 執行個體的狀態。</p> <p>若要啟動資料庫執行個體，請使用主控台或執行下列 AWS CLI 命令：</p> <pre>aws rds start-db-instance \ --db-instance-identifier <i>db-instance-name</i></pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1003	EC2 執行個體類別	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：EC2 主機的預期和已設定的資料庫執行個體類別不相符。您可以將資料庫執行個體類別修改為其原始類別類型來解決此問題。	<p>使用下列 CLI 命令來檢查預期的資料庫執行個體類別：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep DBInstanceClass</pre>
SP-S1004	無法存取 EBS 儲存磁碟區	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：目前無法存取與 EC2 執行個體關聯的原始 EBS 儲存磁碟區 %s。	
SP-S1005	已卸除的 EBS 儲存磁碟區	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：未連接原始 EBS 儲存磁碟區「磁碟區 ID」。您可以透過連接至 EC2 執行個體關聯的 EBS 磁碟區來解決此問題。	<p>重新連接 EBS 磁碟區之後，請使用下列 CLI 命令來檢查 EBS 磁碟區「磁碟區識別碼」是否已正確連接至 RDS 執行個體：</p> <pre>aws ec2 describe-volumes \ --volume-ids <i>volume-id</i> grep InstanceId</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1006	EBS 儲存磁碟區大小	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：EBS 儲存磁碟區「磁碟區 ID」的預期和已設定設定設定不相符。磁碟區大小已在 EC2 層級從原始值 [%s] 手動變更。若要解決此問題，請建立支援案例。	<p>使用下列 CLI 命令來比較 EBS 磁碟區「磁碟區識別碼」詳細資料的磁碟區大小和 RDS 執行個體詳細資訊：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Allocated Storage</pre> <p>使用下列 CLI 命令來檢視實際配置的磁碟區大小：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep Size</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1007	EBS 儲存磁碟區組態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：EBS 儲存磁碟區「磁碟區 ID」的預期和已設定設定不相符。您可以在 EC2 層級將 EBS 儲存磁碟區組態 [IOPS、輸送量、磁碟區類型] 修改為原始值 [IOPS: %s，輸送量:%s，磁碟區類型:%s] 來解決此問題。如需 future 的儲存區修改，請使用 RDS 主控台或 CLI。磁碟區大小也已在 EC2 層級從原始值 [%s] 手動變更。若要解決此問題，請建立支援案例。	<p>使用下列 CLI 命令來比較 EBS 磁碟區 '磁碟區識別碼' 詳細資料的磁碟區類型和 RDS 執行個體詳細資訊。請確定 EBS 層級的值與 RDS 層級的值相符：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep StorageType</pre> <p>若要在 RDS 層級取得儲存體輸送量的預期值：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep StorageThroughput</pre> <p>若要在 RDS 層級取得磁碟區 IOPS 的預期值：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Iops</pre> <p>若要在 EC2 層級取得目前的儲存類型：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep VolumeType</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
			<p>若要取得 EC2 層級的儲存輸送量目前值：</p> <pre data-bbox="987 331 1507 491">aws ec2 describe-volumes \ --volume-ids grep Throughpu t</pre> <p>若要在 EC2 層級取得磁碟區 IOPS 的目前值：</p> <pre data-bbox="987 646 1507 764">aws ec2 describe-volumes \ --volume-ids grep Iops</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1008	EBS 儲存磁碟區大小和組態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：EBS 儲存磁碟區「磁碟區 ID」的預期和已設定設定不相符。您可以在 EC2 層級將 EBS 儲存磁碟區組態 [IOPS、輸送量、磁碟區類型] 修改為原始值 [IOPS: %s，輸送量:%s，磁碟區類型:%s] 來解決此問題。如需 future 的儲存區修改，請使用 RDS 主控台或 CLI。磁碟區大小也已在 EC2 層級從原始值 [%s] 手動變更。若要解決此問題，請建立支援案例。	<p>使用下列 CLI 命令來比較 EBS 磁碟區 '磁碟區識別碼' 詳細資料的磁碟區類型和 RDS 執行個體詳細資訊。請確定 EBS 層級的值與 RDS 層級的值相符：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep StorageType</pre> <p>若要在 RDS 層級取得儲存體輸送量的預期值：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep StorageThroughput</pre> <p>若要在 RDS 層級取得磁碟區 IOPS 的預期值：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Iops</pre> <p>若要在 EC2 層級取得目前的儲存類型：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep VolumeType</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
			<p>若要取得 EC2 層級的儲存輸送量目前值：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep Throughput</pre> <p>若要在 EC2 層級取得磁碟區 IOPS 的目前值：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep Iops</pre> <p>若要取得預期的分配磁碟區大小：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Allocated Storage</pre> <p>要獲取實際分配的卷大小：</p> <pre>aws ec2 describe-volumes \ --volume-ids grep Size</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1009	SQS 權限	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：IAM 執行個體設定檔缺少 Amazon Simple Queue Service (SQS) 許可。您可以通過確保與主機關聯的 IAM 配置文件具有以下權限來解決此問題：["SQS : Send Message", "SQS : ReceiveMessage", "SQS : DeleteMessage"]。GetQueue	
SP-S1010	虛擬私人雲端端點	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：VPC 端點政策封鎖了 Amazon Simple Queue Service (SQS) 操作。您可以修改 VPC 端點原則以允許必要的 SQS 動作來解決此問題。	
作業系統			

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S2001	SQL 服務狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：未啟動 SQL Server 服務。您可以重新啟動主機上的 SQL Server 服務來解決此問題。如果此資料庫執行個體是異地同步備份資料庫執行個體，且重新啟動失敗，請停止並啟動主機以啟動容錯移轉。	
SP-S2002	RDS 自訂代理程式狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：未安裝 RDS 自訂代理程式服務或無法啟動。您可以檢閱 Windows 事件記錄檔來判斷服務無法啟動的原因，並採取適當的步驟來修正問題，藉此解決此問題。如需其他協助，請建立支援案例。	

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S1009	SQS 權限	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：IAM 執行個體設定檔缺少 Amazon Simple Queue Service (SQS) 許可。您可以通過確保與主機關聯的 IAM 配置文件具有以下權限來解決此問題：["SQS : Send Message", "SQS : ReceiveMessage", "SQS : 網址DeleteMessage"]。GetQueue	
SP-S1010	虛擬私人雲端端點	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：VPC 端點政策封鎖了 Amazon Simple Queue Service (SQS) 操作。您可以修改 VPC 端點原則以允許必要的 SQS 動作來解決此問題。	
作業系統			

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S2001	SQL 服務狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：未啟動 SQL Server 服務。您可以重新啟動主機上的 SQL Server 服務來解決此問題。如果此資料庫執行個體是異地同步備份資料庫執行個體，且重新啟動失敗，請停止並啟動主機以啟動容錯移轉。	
SP-S2002	RDS 自訂代理程式狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：未安裝 RDS 自訂代理程式服務或無法啟動。您可以檢閱 Windows 事件記錄檔來判斷服務無法啟動的原因，並採取適當的步驟來解決此問題。如需其他協助，請建立支援案例。	<p>登入主機並確定 RDS Custom 代理程式正在執行。</p> <p>您可以使用下列命令來檢視代理程式狀態。</p> <pre>\$name = "RDSCustomAgent" \$service = Get-Service \$name Write-Host \$service.Status</pre> <p>如果狀態不是 Running，您可以使用下列命令啟動服務：</p> <pre>Start-Service \$name</pre> <p>如果代理程式無法啟動，請檢查 Windows 事件以瞭解無法啟動的原因。代理程式需要 Windows 使用者才能啟動服務。確定 Windows 使用者存在且具有執行服務的權限。</p>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S2003	SSM 代理程式狀態	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：Amazon SSM 代理程式服務無法連線。您可以使用 Get-Service AmazonSSMAgent PowerShell 命令檢查服務狀態或使用啟動服務來進行疑難排解 Start-Service AmazonSSMAgent。請確定允許 HTTPS (連接埠 443) 輸出流量傳送至 SSM、SSM 訊息和 ec2 訊息地區端點。	如需詳細資訊，請參閱 針對 SSM Agent 進行疑難排解 。 若要疑難排解 SSM 端點，請參閱 無法連線至 SSM 端點 和 使用 ssm-cli 疑難排解受管節點可用性 。
SP-S2004	RDS 自訂代理程式登入	SP-S2004RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：SQL 登入時發生未預期的問題 "\$HOSTNAME/RDSAgent"。若要解決此問題，請建立支援案例。	

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S2005	時區	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：Amazon EC2 執行個體 [%s] 上的時區已變更。您可以將時區修改回執行個體建立期間指定的設定來解決此問題。如果您想要建立具有特定時區的執行個體，請參閱 RDS 自訂說明文件。	執行 Get-Timezone PowerShell 指令以確認時區。 如需詳細資訊，請參閱 RDS Custom for SQL Server 資料庫執行個體的本機時區 。
SP-S2006	高可用性軟體解決方案版	RDS Custom DB 執行個體狀態設定為 [不支援的組態]，原因是：目前執行個體的高可用性軟體解決方案與預期的版本不同。若要解決此問題，請建立支援案例。	

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S2007	高可用性軟體解決方案組	RDS Custom DB 執行個體狀態設定為 [不支援的組態]，原因是：高可用性軟體解決方案的組態設定已修改為執行個體 %s 上的未預期值。若要修正此問題，請重新啟動 EC2 執行個體。當您重新啟動 EC2 執行個體時，它會自動將設定更新為高可用性軟體解決方案的所需組態。	
資料庫			
SP-S3001	SQL 伺服器共用記憶體協定	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：SQL Server 共用記憶體通訊協定已停用。您可以在 SQL Server 組態管理員中啟用共用記憶體通訊協定來解決此問題。	您可以通過檢查以下方式進行驗證：SQL Server 組態管理員 > SQL 伺服器網路組態 > MSSQLSERVER 的通訊協定 > 共用記憶體為已啟用。啟用通訊協定之後，請重新啟動 SQL Server 處理序。

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S3002	服務主要金鑰	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：RDS 自動化無法將服務主要金鑰 (SMK) 的備份做為新 SMK 產生的一部分。若要解決此問題，請建立支援案例。	
SP-S3003	服務主要金鑰	由於以下原因，RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]：與服務主要金鑰 (SMK) 相關的中繼資料遺失或不完整。若要解決此問題，請建立支援案例。	

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S3004	資料庫引擎版本和版本	預期和安裝的 SQL Server 版本和版本之間存在不相符的情況。SQL 伺服器的 RDS 自訂不支援修改 SQL 伺服器版本。此外，不支援在 RDS 自訂 EC2 執行個體上手動變更 SQL 伺服器版本。若要解決此問題，請建立支援案例。	<p>執行下列查詢以取得 SQL 版本：</p> <pre>select @@version</pre> <p>執行下列 AWS CLI 命令以取得 RDS SQL 引擎版本和版本：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep EngineVersion aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Engine</pre> <p>如需詳細資訊，請參閱 修改 RDS Custom for SQL Server 資料庫執行個體 及 升級資料庫執行個體引擎版本。</p>
SP-S3005	資料庫引擎版本	目前的 SQL 伺服器版本不符合預期的 SQL 伺服器版本 [%s]。SQL 伺服器的 RDS 自訂不支援修改 SQL 伺服器版本。若要解決此問題，請建立支援案例。	<p>執行下列查詢以取得 SQL 版本：</p> <p>Example</p> <pre>select @@version</pre> <p>執行下列 AWS CLI 命令以取得 RDS SQL 引擎版本：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep Engine</pre>

事件代碼	組態區域	RDS 事件訊息	驗證程序
SP-S3006	資料庫引擎版本	目前的 SQL 伺服器版本與預期的 SQL 伺服器版本 [%s] 不符。您無法手動變更 RDS 自訂 EC2 執行個體上的 SQL 伺服器版本。若要解決此問題，請建立支援案例。對於 future 對 SQL Server 版本的任何修改，您可以從 AWS RDS 主控台或透過 modify-db-instance CLI 命令修改執行個體。	<p>執行下列查詢以取得 SQL 版本：</p> <p>Example</p> <pre>select @@version</pre> <p>執行下列 AWS CLI 命令以取得 RDS SQL 引擎版本：</p> <pre>aws rds describe-db-instances \ --db-instance-identifier <i>db-instance-name</i> grep EngineVersion</pre> <p>如需詳細資訊，請參閱 修改 RDS Custom for SQL Server 資料庫執行個體 及 升級資料庫執行個體引擎版本。</p>
SP-S3007	資料庫檔案位置	RDS 自訂資料庫執行個體狀態設定為 [不支援的組態]，原因是：資料庫檔案設定在 D:\ 磁碟機之外。您可以通過確保所有數據庫文件（包括 ROW，LOG，FILESTREAM 等）都存儲在 D:\ 驅動器上來解決此問題。	<p>執行下列查詢，以列出不在預設路徑中的資料庫檔案位置：</p> <pre>USE master; SELECT physical_name as files_not_in_default_path FROM sys.master_files WHERE SUBSTRING(physical_name,1,3)!='D:\';</pre>

SQL 伺服器 RDS 自訂Storage-Full中的疑難排解

RDS 自訂會監控適用於 SQL Server 資料庫執行個體的 RDS 自訂磁碟區的根 (C:) 和資料 (D:) 磁碟區上的可用空間。當任一磁碟區的可用磁碟空間少於 500 MiB 時，RDS 自訂會將執行個體Storage-Full狀態移至狀態。若要擴展執行個體儲存體，請參閱[修改 RDS Custom for SQL Server 資料庫執行個體的儲存體](#)。

Note

擴展儲存後，執行個體最多Storage-Full可能需要 30 分鐘才能解決問題。

與 Amazon RDS 上的工作 AWS Outposts

Amazon RDS 將適用於 SQL 伺服器的 RDS、適用於 MySQL 的 RDS 和適用於 PostgreSQL 資料庫的 RDS 資料庫 AWS Outposts 延伸到 AWS Outposts 環境中。AWS Outposts 使用與公共場所相同的硬體，AWS 區域 將 AWS 服務、基礎架構和作業模型帶入內部部署。透過 Outposts 上的 RDS，您可以將受管資料庫執行個體佈建至接近必須在內部部署執行的商業應用程式之處。如需有關的更多資訊 AWS Outposts，請參閱[AWS Outposts](#)。

您可以使用相同的 AWS Management Console AWS CLI、和 RDS API 佈建和管理 Outposts 資料庫執行個體上的內部部署 RDS，就像在中執行的 RDS 資料庫執行個體一樣。AWS 雲端 RDS on Outposts 會將 Amazon S3 中的資料庫佈建、作業系統和資料庫修補、備份及長期封存等任務自動化。

Outposts 上的 RDS 支援自動備份資料庫執行個體。您必須備份和還原資料庫執行個體之間的 AWS 區域 Outpost 與您的網路連線。來自 Outpost 的所有數據庫快照和事務日誌都存儲在您的 AWS 區域。從 AWS 區域中，您可以將資料庫執行個體從資料庫快照還原到不同的 Outpost。如需更多詳細資訊，請參閱 [備份簡介](#)。

Outposts 上的 RDS 支援自動維護和升級資料庫執行個體。如需詳細資訊，請參閱 [維持資料庫執行個體](#)。

RDS on Outposts 會透過您的 AWS KMS key，對資料庫執行個體和資料庫快照使用靜態加密。如需靜態加密的詳細資訊，請參閱 [加密 Amazon RDS 資源](#)。

根據預設，Outpost 子網路中的 EC2 執行個體可以使用 Amazon Route 53 DNS 服務將網域名稱解析為 IP 地址。您可能會與 Route 53 經歷較長的 DNS 解析時間，取決於您的 Outpost 和 AWS 區域之間的路徑延遲。在這種情況下，您可以使用內部部署環境中本機安裝的 DNS 伺服器。如需詳細資訊，請參閱《AWS Outposts 使用者指南》中的 [DNS](#)。

當網路連線 AWS 區域 無法使用時，您的資料庫執行個體會繼續在本機執行。您可以將本機 DNS 伺服器設定為輔助伺服器，繼續使用 DNS 名稱解析存取資料庫執行個體。但是，您無法建立新的資料庫執行個體或修改現有的資料庫執行個體。沒有連線時，自動備份不會執行。如果發生資料庫執行個體失敗，在連線還原前，不會自動取代資料庫執行個體。建議您盡快還原網路連線。

主題

- [Amazon RDS 上的先決條件 AWS Outposts](#)
- [Amazon RDS 功能的 AWS Outposts 上的 Amazon RDS 支援](#)
- [AWS Outposts 上的 Amazon RDS 支援的資料庫執行個體類別](#)
- [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#)

- [在 AWS Outposts 上使用 Amazon RDS 的異地同步備份部署](#)
- [建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體](#)
- [在上建立 Amazon RDS 的僅供讀取複本 AWS Outposts](#)
- [在 AWS Outposts 上還原 Amazon RDS 上資料庫執行個體的注意事項](#)

Amazon RDS 上的先決條件 AWS Outposts

下列是使用 AWS Outposts 上的 Amazon RDS 的先決條件：

- 安裝 AWS Outposts 在您的內部部署資料中心。如需有關的更多資訊 AWS Outposts，請參閱[AWS Outposts](#)。
- 確認您至少有一個可供 Outposts 上的 RDS 使用的子網路。您可以使用相同的子網路處理其他工作負載。
- 確認 Outpost 與 AWS 區域之間有可靠的網路連線。

Amazon RDS 功能的 AWS Outposts 上的 Amazon RDS 支援

下表說明 AWS Outposts 上的 Amazon RDS 支援的 Amazon RDS 功能。

功能	支援	備註	其他資訊
資料庫執行個體佈建	是	<p>您只能為 RDS for SQL Server、RDS for MySQL 和 RDS for PostgreSQL 資料庫引擎建立資料庫執行個體。以下為可支援的版本：</p> <ul style="list-style-type: none"> Microsoft SQL Server : <ul style="list-style-type: none"> 15.00.4043.16.v1 和更新的 2019 版本 14.00.3294.2.v1 和更新的 2017 版本 13.00.5820.21.v1 和更新的 2016 版本 MySQL 8.0.28 版和更新的 MySQL 8.0 版本 PostgreSQL 版本以及第 PostgreSQL 及更 PostgreSQL 本 	建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體
使用 Microsoft SQL Server Management Studio 連線至 Microsoft SQL Server 資料庫執行個體。	是	<p>某些 TLS 版本和加密密碼可能不安全。若要關閉它們，請遵照 設定安全性通訊協定和密碼 中的指示。</p>	連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體
修改主要使用者密碼	是	—	修改 Amazon RDS 資料庫執行個體

功能	支援	備註	其他資訊
重新命名資料庫執行個體	是	—	修改 Amazon RDS 資料庫執行個體
將資料庫執行個體重新開機	是	—	重新啟動中的資料庫執行個體
停止資料庫執行個體	是	—	暫時停止 Amazon RDS 資料庫執行個體
啟動資料庫執行個體	是	—	啟動之前已停止的 Amazon RDS 資料庫執行個體
異地同步備份部署	是	MySQL 和 PostgreSQL 資料庫執行個體支援異地同步備份部署。 多可用區部署不支援直接 VPC 路由 (DVR)。	建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體 設定及管理多可用區部署
資料庫參數群組	是	—	使用參數群組
僅供讀取複本	是	MySQL 和 PostgreSQL 資料庫執行個體支援僅供讀取複本。 僅供讀取複本不支援直接 VPC 路由 (DVR)。	在上建立 Amazon RDS 的僅供讀取複本 AWS Outposts
靜態加密	是	Outposts 上的 RDS 不支援未加密的資料庫執行個體。	加密 Amazon RDS 資源
AWS Identity and Access Management (IAM) 資料庫身分驗證	否	—	適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證

功能	支援	備註	其他資訊
將 IAM 角色與資料庫執行個體建立關聯	否	—	add-role-to-db-實例命令 AWS CLI AddRoleTo資料庫執行個體 RDS API 作業
Kerberos 身分驗證	否	—	Kerberos 身分驗證
標記 Amazon RDS 資源	是	—	標記 Amazon RDS 資源
選項群組	是	—	使用選項群組
修改維護時段	是	—	維持資料庫執行個體
自動次要版本升級	是	—	自動升級次要引擎版本
修改備份時段	是	—	備份簡介 修改 Amazon RDS 資料庫執行個體
變更資料庫執行個體類別	是	—	修改 Amazon RDS 資料庫執行個體
變更配置儲存	是	—	修改 Amazon RDS 資料庫執行個體
儲存自動擴展	是	—	使用 Amazon RDS 儲存體自動擴展自動管理容量

功能	支援	備註	其他資訊
手動和自動資料庫執行個體快照	是	<p>您可將自動化備份或手動快照存放在您的 AWS 區域中。或者，您可以在本機將它們存放在 Outpost 上。</p> <p>MySQL 和 PostgreSQL 資料庫執行個體支援本機備份。</p> <p>若要將備份存放在您的 Outpost 上，請確定您已在 Outpost 上設定 Amazon S3。</p> <p>多可用區執行個體部署不支援本機備份。</p>	<p>建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體</p> <p>Amazon S3 on Outposts</p> <p>為單一可用區資料庫執行個體建立資料庫快照</p>
從資料庫快照還原	是	<p>您可將還原的資料庫執行個體的自動備份和手動快照存放在父系 AWS 區域中，或本機的 Outpost 上。</p>	<p>在 AWS Outposts 上還原 Amazon RDS 上資料庫執行個體的注意事項</p> <p>從資料庫快照還原</p>
從 Amazon S3 還原資料庫執行個體	否	—	將備份還原至 MySQL 資料庫執行個體
將快照資料匯出至 Amazon S3	否	—	將資料庫快照資料匯出至 Amazon S3
Point-in-time 回收	是	<p>您可將還原的資料庫執行個體的自動備份和手動快照存放在父系 AWS 區域中，或本機的 Outpost 上，但有一個例外。</p>	<p>在 AWS Outposts 上還原 Amazon RDS 上資料庫執行個體的注意事項</p> <p>將資料庫執行個體還原至指定的時間</p>

功能	支援	備註	其他資訊
Enhanced monitoring (增強型監控)	否	—	使用增強型監控來監控作業系統指標
Amazon CloudWatch 監控	是	您可在 AWS 區域 中檢視資料庫可用的相同指標集。	使用 Amazon CloudWatch 監控 Amazon RDS 指標
將資料庫引擎記錄發佈至 CloudWatch 記錄	是	—	將資料庫日誌發佈至 Amazon CloudWatch Logs
事件通知	是	—	使用 Amazon RDS 事件通知
Amazon RDS 績效詳情	否	—	在 Amazon RDS 上使用績效詳情監控資料庫負載
檢視或下載資料庫日誌	否	<p>RDS on Outposts 不支援使用主控台檢視資料庫日誌，或使用 AWS CLI 或 RDS API 描述資料庫日誌。</p> <p>RDS on Outposts 不支援使用主控台下載資料庫日誌，或使用 AWS CLI 或 RDS API 下載資料庫日誌。</p>	監控 Amazon RDS 日誌檔案
Amazon RDS 代理	否	—	使用 Amazon RDS Proxy
Amazon RDS for MySQL 的預存程序	是	—	RDS for MySQL 預存程序參考

功能	支援	備註	其他資訊
透過 RDS for MySQL 的外部資料庫進行複寫	否	—	使用外部來源執行個體設定二進位日誌檔案位置複寫
Amazon RDS for Microsoft SQL Server 的原生備份與還原	是	—	使用原生備份和還原，匯入和匯出 SQL Server 資料庫

AWS Outposts 上的 Amazon RDS 支援的資料庫執行個體類別

AWS Outposts 上的 Amazon RDS 支援下列資料庫執行個體類別：

- 一般用途資料庫執行個體類別
 - db.m5.24xlarge
 - db.m5.12xlarge
 - db.m5.4xlarge
 - db.m5.2xlarge
 - db.m5.xlarge
 - db.m5.large
- 記憶體最佳化資料庫執行個體類別
 - db.r5.24xlarge
 - db.r5.12xlarge
 - db.r5.4xlarge
 - db.r5.2xlarge
 - db.r5.xlarge
 - db.r5.large

依據您的 Outpost 設定方式，您可能無法使用所有的這些類別。例如，若您尚未為 Outpost 購買 db.r5 類別，則其無法與 RDS on Outposts 一起使用。

Outposts 上的 RDS 資料庫執行個體僅支援一般用途 SSD 儲存體。如需資料庫執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。

Amazon RDS 管理資料庫執行個體的維護和復原，並且需要 Outpost 的作用中容量才能執行這項操作。建議您為生產環境中的每個資料庫執行個體類別設定 N+1 個 EC2 執行個體。RDS on Outposts 可以使用這些 EC2 執行個體的額外容量來進行維護和修復操作。例如，如果您的生產環境有 3 個 db.m5.large 和 5 個 db.r5.xlarge 資料庫執行個體類別，我們建議它們至少有 4 個 m5.large EC2 執行個體和 6 個 r5.xlarge EC2 執行個體。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的 [AWS Outposts 恢復能力](#)。

適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址

AWS Outposts 上的 Amazon RDS 會使用您提供的內部部署網路相關資訊，來建立地址集區。此集區稱為「客戶擁有的 IP 地址集區」(CoIP 集區)。客戶擁有的 IP 地址 (CoIP) 可透過內部部署網路，為您 Outpost 子網路中的資源提供本機或外部連線能力。如需 CoIP 的詳細資訊，請參閱 AWS Outposts 使用者指南中的[客戶擁有的 IP 地址](#)。

每個 Outposts 上的 RDS 資料庫執行個體都有一個私有 IP 地址，用於其 Virtual Private Cloud (VPC) 內的流量。此私人 IP 位址無法公開存取。您可以使用 Public (公有) 選項來設定除了私有 IP 地址之外，資料庫執行個體是否還具有公有 IP 地址。將公有 IP 地址用於連線時，便會透過網際網路傳送該地址，且在某些情況下可能會導致高延遲。

Outposts 上的 RDS 會透過其子網路支援，使用資料庫執行個體的 CoIP，而非使用這些私有和公有 IP 地址。當您使用 Outposts 資料庫執行個體上 RDS 的 CoIP 時，會使用資料庫執行個體端點連線到資料庫執行個體。RDS on Outposts 會自動將 CoIP 用於所有來自 VPC 內部和外部的連線。

CoIP 可為 Outposts 上的 RDS 資料庫執行個體提供以下優勢：

- 較低的連線延遲
- 增強的安全性

使用 CoIP

您可使用 AWS Management Console、AWS CLI 或 RDS API，來開啟或關閉 Outposts 資料庫執行個體上 RDS 的 CoIP：

- 透過 AWS Management Console，選擇 Access type (存取類型) 中的 Customer-owned IP address (CoIP) (客戶擁有的 IP 地址) 設定來使用 CoIP。選擇其他設定之一將其關閉。

▼ **Additional configuration**

Access type [Info](#)

Private
RDS will not assign a public IP address to the database. Amazon EC2 instances and devices inside the VPC can connect to your database. EC2 instances and devices outside your VPC can't connect unless they use AWS Site-to-Site VPN or AWS Direct Connect.

Customer-owned IP address (CoIP)
Devices on your on-premises network can connect to your database through a CoIP.

Public
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices can connect to the database.

Database port
TCP/IP port that the database will use for application connections.

- 透過 AWS CLI，使用 `--enable-customer-owned-ip` | `--no-enable-customer-owned-ip` 選項。
- 透過 RDS API，使用 `EnableCustomerOwnedIp` 參數。

您可在執行下列任一動作時，開啟或關閉 CoIP：

- 建立資料庫執行個體

如需更多詳細資訊，請參閱 [建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體](#)。

- 修改資料庫執行個體

如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

- 建立僅供讀取複本

如需更多詳細資訊，請參閱 [在上建立 Amazon RDS 的僅供讀取複本 AWS Outposts](#)。

- 從快照中還原資料庫執行個體

如需更多詳細資訊，請參閱 [從資料庫快照還原](#)。

- 將資料庫執行個體還原至指定的時間

如需更多詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

Note

在某些情況下，您可能會開啟資料庫執行個體的 CoIP，但 Amazon RDS 無法配置資料庫執行個體的 CoIP。在這類情況下，資料庫執行個體狀態會變更為 incompatible-network。如需執行資料庫個體狀態的詳細資訊，請參閱 [檢視 Aurora 叢集中 資料庫執行個體狀態](#)。

限制

下列限制適用於 Outposts 上的 RDS 資料庫執行個體的 CoIP 支援：

- 使用資料庫執行個體的 CoIP 時，請確認資料庫執行個體的公有存取性已關閉。
- 請確認您 VPC 安全群組的傳入規則包括 CoIP 地址範圍 (CIDR 區塊)。如需設定安全群組的詳細資訊，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。
- 您無法將 CoIP 集區中的 CoIP 指派給資料庫執行個體。當您使用資料庫執行個體的 CoIP 時，Amazon RDS 會自動將 CoIP 集區的 CoIP 指派給資料庫執行個體。
- 您必須使用擁有 Outpost 資源 (擁有者) 的 AWS 帳戶 帳戶，或與相同組織中的其他 AWS 帳戶 帳戶 (消費者) 共用以下資源。
 - Outpost
 - 資料庫執行個體 VPC 的本機閘道 (LGW) 路由表
 - LGW 路由表的 CoIP 集區或集區

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的 [使用共享 AWS Outposts 資源](#)。

在 AWS Outposts 上使用 Amazon RDS 的異地同步備份部署

對於多可用區部署，Amazon RDS 會在某個 AWS Outpost 上建立主要資料庫執行個體。RDS 會將資料同步複製到不同 Outpost 上的待命資料庫執行個體。

AWS Outposts 上的異地同步備份部署與 AWS 區域中異地同步備份部署的作業類似，但有下列不同處：

- 其需要在兩個或多個 Outpost 之間建立本機連線。
- 其需要客戶擁有的 IP (CoIP) 集區。如需更多詳細資訊，請參閱 [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#)。
- 在您的本機網路上執行複製。

AWS Outposts 上的多可用區域適用於 Outpost 上 RDS 上的所有受支援的 MySQL 和 PostgreSQL 版本。異地同步備份部署不支援本機備份。如需更多詳細資訊，請參閱 [建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體](#)。

使用共同的責任模型

儘管 AWS 使用商業上合理的努力來提供設定為高可用性的資料庫執行個體，但可用性會使用共同的責任模型。Outpost 上的 RDS 進行容錯移轉和修復資料庫執行個體的能力需要您每個 Outpost 皆連接至其 AWS 區域。

Outpost 上的 RDS 還需要在託管主資料庫執行個體的 Outpost 和託管備用資料庫執行個體的 Outpost 之間建立連線以進行同步複製。對此連線的任何影響皆可阻止在 Outpost 上 RDS 執行容錯移轉。

由於同步資料複製，您可能會看到標準資料庫執行個體部署的延遲增加狀況。託管主資料庫執行個體的 Outpost 與託管備用資料庫執行個體的 Outpost 之間的連線頻寬和延遲會直接影響延遲。如需更多詳細資訊，請參閱 [先決條件](#)。

提高可用性

我們建議採取下列措施來提高可用性：

- 為您的任務關鍵型應用程式分配足夠的額外容量，以便在存在潛在主機問題時進行恢復和容錯移轉。此適用於資料庫子網路群組中包含子網路的所有 Outpost。如需詳細資訊，請參閱 [AWS Outposts 中的恢復能力](#)。
- 為您的 Outposts 提供備援網路連線。

- 使用兩個以上的 Outpost。擁有兩個以上的 Outposts 允許 Amazon RDS 復原資料庫執行個體。如果目前 Outpost 遇到失敗時，RDS 會將資料庫執行個體移動至另一個 Outpost 來執行此復原。
- 為您的 Outpost 提供雙電源和冗餘網路連線。

我們建議您的本機網路如下：

- 託管主資料庫執行個體的 Outpost 與託管備用資料庫執行個體的 Outpost 之間的往返時間 (RTT) 延遲直接影響寫入延遲。將 AWS Outposts 之間的 RTT 延遲保持在低個位數毫秒內。我們建議不超過 5 毫秒，但您的要求可能會有所不同。

您可在 WriteLatency 的 Amazon CloudWatch 指標中找到對網路延遲的淨影響。如需更多詳細資訊，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。

- Outpost 之間連線的可用性會影響資料庫執行個體的整體可用性。在 Outpost 之間有冗餘的網路連線。

先決條件

Outpost 上 RDS 上的異地同步備份部署具有下列先決條件：

- 至少有兩個 Outpost，透過本機連線連接並附加至 AWS 區域中的不同可用區域。
- 請確定您的資料庫子網路群組包含下列資料：
 - 指定的 AWS 區域中至少兩個可用區域中有至少兩個子網路。
 - 僅在 Outpost 中的子網路。
 - 相同 virtual private cloud (VPC) 中至少兩個 Outpost 中至少有兩個子網路。
- 將您資料庫執行個體的 VPC 與您所有本機閘道路由表產生關聯。此關聯是必要的，因為複寫會使用 Outpost 的本機閘道在您的本機網路上執行。

例如，假設您的 VPC 在 Outpost-A 中包含 subnet-A，在 Outpost-B 中包含 subnet-B。Outpost-A 會使用 LocalGateway-A (LGW-A)，而 Outpost-B 會使用 LocalGateway-B (LGW-B)。LGW-A 具有 RouteTable-A，而 LGW-B 具有 RouteTable-B。您想對複寫流量使用 RouteTable-A 和 RouteTable-B。如要執行此作業，請將您的 VPC 與 RouteTable-A 和 RouteTable-B 產生關聯。

如需如何建立關聯的詳細資訊，請參閱 Amazon EC2 [create-local-gateway-route-table-vpc-association](#) AWS CLI 命令。

- 確保您的 Outpost 使用客戶擁有的 IP (CoIP) 路由。每個路由表還必須至少有一個地址集區。Amazon RDS 為主資料庫執行個體和備用資料庫執行個體分配一個額外的 IP 地址，以進行資料同步。
- 確保擁有 RDS 資料庫執行個體的 AWS 帳戶 擁有本機閘道路由表和 CoIP 集區。或確保它屬於 Resource Access Manager 共用的一部分，可存取本機閘道路由表和 CoIP 集區。
- 確保 CoIP 集區中的 IP 地址可從一個 Outpost 本機閘道路由至其他 Outpost。
- 請確保 VPC 的 CIDR 區塊 (例如 10.0.0.0/4) 和 COIP 集區 CIDR 區塊不包含來自 E 類別 (240.0.0.0/4) 的 IP 地址。RDS 在內部使用這些 IP 地址。
- 請確保正確設定傳出和相關傳入流量。

Outpost 上的 RDS 建立主資料庫執行個體和備用資料庫執行個體之間的虛擬私有網路 (VPN) 連線。若要使其正常運作，您的本機網路必須允許網際網路安全關聯和金鑰管理通訊協定 (ISAKMP) 的傳出和相關傳入流量。其使用使用者資料包通訊協定 (UDP) 連接埠 500 和 IP 安全 (IPsec) 網路位址轉譯周遊 (NAT-T) 使用 UDP 連接埠 4500 來執行此作業。

如需 CoIP 的更多詳細資訊，請參閱此指南中的 [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#) 和《AWS Outposts 使用者指南》中的 [客戶擁有的 IP 地址](#)。

針對 Amazon EC2 許可權限使用 API 作業

無論您是否在 AWS Outposts 上使用資料庫執行個體的 CoIP，RDS 需要存取您的 CoIP 集區資源。RDS 可代表您為異地同步備份部署呼叫 CoIP 下列 EC2 許可權限 API 作業：

- `CreateCoipPoolPermission` – 在 Outpost 上 RDS 上建立多可用區域資料庫執行個體時
- `DeleteCoipPoolPermission` – 在 Outpost 上 RDS 上刪除多可用區域資料庫執行個體時

這些 API 作業可對內部 RDS 帳戶授予或從中移除許可權限，以從該許可權限指定的 CoIP 集區分配彈性 IP 地址。您可使用 `DescribeCoipPoolUsage` API 作業檢視這些 IP 地址。如需 CoIP 的更多詳細資訊，請參閱《AWS Outposts 使用者指南》中的 [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#) 和 [客戶擁有的 IP 地址](#)。

RDS 還可代表您為異地同步備份部署呼叫本機閘道路由表的下列 EC2 許可 API 作業：

- `CreateLocalGatewayRouteTablePermission` – 在 Outpost 上 RDS 上建立多可用區域資料庫執行個體時
- `DeleteLocalGatewayRouteTablePermission` – 在 Outpost 上 RDS 上刪除多可用區域資料庫執行個體時

這些 API 作業可對內部 RDS 帳戶授予或從中移除許可權限，以使內部 RDS VPC 與本機閘道路由表產生關聯。您可使用 `DescribeLocalGatewayRouteTableVpcAssociations` API 作業來檢視這些路由表-VPC 關聯。

建立 AWS Outposts 上的 Amazon RDS 的資料庫執行個體

建立 AWS Outposts 上的 Amazon RDS 資料庫執行個體類似於在 AWS 雲端建立 Amazon RDS 資料庫執行個體。不過，請務必指定與 Outpost 相關聯的資料庫子網路群組。

以 Amazon VPC 服務為基礎的 Virtual Private Cloud (VPC) 可以跨越 AWS 區域 中的所有可用區域。您可新增 Outpost 子網路，將 AWS 區域 中的任何 VPC 延伸至 Outpost。若要將 Outpost 子網路新增至 VPC，請在建立子網路時指定 Outpost 的 Amazon Resource Name (ARN)。

建立 Outposts 上的 RDS 資料庫執行個體之前，您可以建立資料庫子網路群組，其中包含一個與 Outpost 相關聯的子網路。建立 Outposts 上的 RDS 資料庫執行個體時，請指定此資料庫子網路群組。您也可以選擇在建立資料庫執行個體時，建立新的資料庫子網路群組。

如需如何設定 AWS Outposts 的資訊，請參閱 [AWS Outposts 使用者指南](#)。

主控台

建立資料庫子網路群組

建立資料庫子網路群組，其中包含一個與 Outpost 相關聯的子網路。

您也可以在建​​立資料庫執行個體時，建立 Outpost 的新資料庫子網路群組。如果您希望執行此作業，請略過此程序。

Note

若要為 AWS 雲端 建立資料庫子網路群組，請指定至少兩個子網路。

若要建立 Outpost 的資料庫子網路群組

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇您要建立資料庫子網路群組的 AWS 區域。
3. 選擇 Subnet groups (子網路群組)，然後選擇 Create DB Subnet Group (建立資料庫子網路群組)。

Create DB subnet group (建立資料庫子網路群組) 頁面隨即出現。

RDS > Subnet groups > Create DB subnet group

Create DB Subnet Group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

4. 針對 Name (名稱)，輸入資料庫子網路群組的名稱。
5. 針對 Description (描述)，選擇資料庫子網路群組的描述。
6. 針對 VPC，選擇您想為其建立資料庫子網路群組的 VPC。
7. 在可用區域中，選擇 Outpost 的可用區域。

8. 在子網路中，選擇供 Outposts 上的 RDS 使用的子網路。
9. 選擇 Create (建立) 來建立資料庫子網路群組。

建立 RDS on Outposts 資料庫執行個體

建立資料庫執行個體，然後選擇資料庫執行個體的 Outpost。

使用主控台建立 Outposts 上的 RDS 資料庫執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要在其上建立資料庫執行個體之 Outpost 所附加的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。

AWS Management Console 會偵測您已設定的可用 Outposts，並在資料庫位置區段中顯示內部部署選項。

Note

如果您尚未設定任何 Outposts，則表示資料庫位置區段沒有出現，或選擇內部部署建立方法區段中沒有 Outposts 上的 RDS 選項。

5. 針對 Database location (資料庫位置)，選擇 On-premises (內部部署)。
6. 針對 On-premises creation method (內部部署建立方法)，選擇 RDS on Outposts。
7. 指定 Outposts Connectivity (Outposts 連線能力) 的設定。這些設定適用於使用 VPC (具有資料庫執行個體的資料庫子網路) 的 Outpost。您的 VPC 必須以 Amazon VPC 服務為基礎。
 - a. 針對 Virtual Private Cloud (VPC)，選擇包含資料庫執行個體之資料庫子網路群組的 VPC。
 - b. 針對 VPC security group (VPC 安全群組)，選擇適用於您的資料庫執行個體的 Amazon VPC 安全群組。
 - c. 針對 DB Subnet group (資料庫子網路群組)，選擇適用於您的資料庫執行個體的資料庫子網路群組。

您可選擇與 Outpost 相關聯的現有資料庫子網路群組—例如，若您執行 [建立資料庫子網路群組](#) 中的程序。

您也可以為 Outpost 建立新的資料庫子網路群組。

8. 若為 Multi-AZ deployment (異地同步備份部署)，請選擇 Create a standby instance (recommended for production usage)(建立備用執行個體，推薦用於生產用途)，在另一個 Outpost 中建立備用資料庫執行個體。

Note

此選項無法用於 Microsoft SQL Server。

若選擇建立異地同步備份部署，則無法將備份存放於 Outpost 上。

9. 在 Backup (備份) 底下，執行下列動作：

- a. 針對 Backup target (備份目標)，選擇下列其中一項：

- AWS 雲端 在父系 AWS 區域 中存放自動備份和手動快照。
- Outposts (on-premises) (Outposts (現場部署)) 以建立本機備份。

Note

如要在您的 Outpost 上存放備份，您的 Outpost 必須具有 Simple Storage Service (Amazon S3) 功能。如需詳細資訊，請參閱 [Amazon S3 on Outposts](#)。

多可用區部署或僅供讀取複本不支援本機備份。

- b. 選擇啟用自動備份以建立資料庫執行個體的 point-in-time 快照。

若您開啟自動備份，則可選擇 Backup retention period (備份保留期) 和 Backup window (備份時段)，或保留預設值。

10. 視需要指定其他資料庫執行個體設定。

如需建立資料庫執行個體時各項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

11. 選擇建立資料庫。

Databases (資料庫) 頁面隨即出現。橫幅告訴您正在建立資料庫執行個體，並顯示 View credential details (檢視憑證詳細資訊) 按鈕。

檢視資料庫執行個體詳細資訊

建立資料庫執行個體之後，您可以檢視其憑證和其他詳細資訊。

若要檢視資料庫執行個體詳細資訊

1. 若要檢視資料庫執行個體的主要使用者名稱和密碼，請選擇 Databases (資料庫) 頁面上的 View credential details (檢視憑證詳細資訊)。

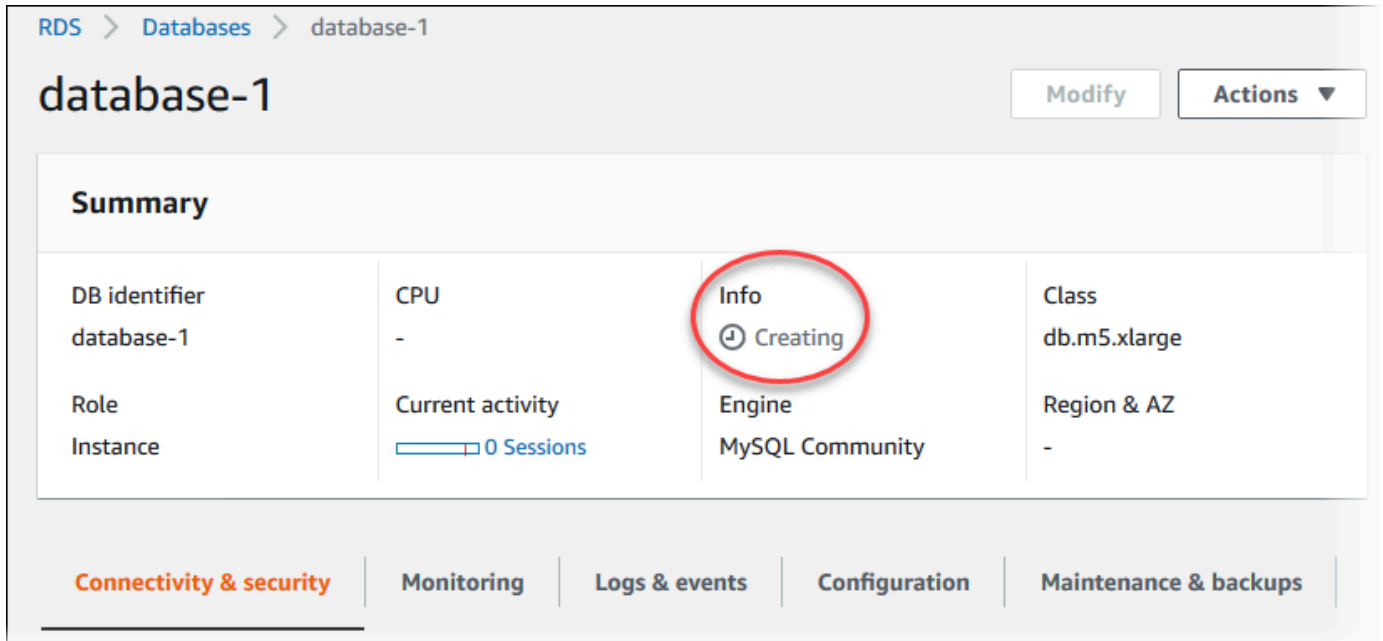
您可以使用這些憑證，以主要使用者身分連線至資料庫執行個體。

Important

您無法再次檢視主要使用者密碼。如果您沒有記錄下來，您可能需要進行變更。若要在資料庫執行個體可供使用後變更主要使用者密碼，請修改資料庫執行個體。如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

2. 在 Databases (資料庫) 頁面上，選擇新資料庫執行個體的名稱。

在 RDS 主控台上，新資料庫執行個體的詳細資訊即會出現。在資料庫執行個體建立完成且可供使用之前，資料庫執行個體會處於 Creating (建立中) 狀態。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和分配的儲存空間而定，新的資料庫執行個體可能需要幾分鐘才能使用。



RDS > Databases > database-1

database-1

Modify Actions

Summary

DB identifier database-1	CPU -	Info 🕒 Creating	Class db.m5.xlarge
Role Instance	Current activity 0 Sessions	Engine MySQL Community	Region & AZ -

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups

在資料庫執行個體可供使用之後，管理起來就像在 AWS 雲端 中管理 RDS 資料庫執行個體一樣。

AWS CLI

在 Outpost 中使用 AWS CLI 建立新的資料庫執行個體之前，請先建立資料庫子網路群組，以供 RDS on Outposts 使用。

若要建立 Outpost 的資料庫子網路群組

- 使用 [create-db-subnet-group](#) 命令。針對 `--subnet-ids`，指定 Outpost 中的子網路群組，以供 RDS on Outposts 使用。

對於LinuxmacOS、或Unix：

```
aws rds create-db-subnet-group \  
  --db-subnet-group-name myoutpostdbsubnetgr \  
  --db-subnet-group-description "DB subnet group for RDS on Outposts" \  
  --subnet-ids subnet-abc123
```

在Windows中：

```
aws rds create-db-subnet-group ^  
  --db-subnet-group-name myoutpostdbsubnetgr ^  
  --db-subnet-group-description "DB subnet group for RDS on Outposts" ^  
  --subnet-ids subnet-abc123
```

若要使用 AWS CLI 建立 RDS on Outposts 資料庫執行個體

- 使用 [create-db-instance](#) 命令。指定 Outpost 的可用區域、與 Outpost 建立關聯的 Amazon VPC 安全群組，以及為 Outpost 建立的資料庫子網路群組。可以包含下列選項：
 - `--db-instance-identifier`
 - `--db-instance-class`
 - `--engine` – 資料庫引擎。使用以下其中一個值：
 - MySQL – 指定 `mysql`。
 - PostgreSQL – 指定 `postgres`。
 - Microsoft SQL Server – 指定 `sqlserver-ee`、`sqlserver-se`、或 `sqlserver-web`。
 - `--availability-zone`
 - `--vpc-security-group-ids`

- `--db-subnet-group-name`
- `--allocated-storage`
- `--max-allocated-storage`
- `--master-username`
- `--master-user-password`
- `--multi-az` | `--no-multi-az` – (選用) 是否在不同的可用區域建立備用資料庫執行個體。預設值為 `--no-multi-az`。

`--multi-az` 選項無法用於 SQL Server。

- `--backup-retention-period`
- `--backup-target` – (選用) 存放自動備份和手動快照的位置。使用以下其中一個值：
 - `outposts` – 將它們本機存放在您的 Outpost 上。
 - `region` – 將其存放於父系 AWS 區域中。這是預設值。

若您使用 `--multi-az` 選項，則無法使用 `--backup-target` 的 `outposts`。此外，如果您將 `outposts` 用於 `--backup-target`，資料庫執行個體就無法具有僅供讀取複本。

- `--storage-encrypted`
- `--kms-key-id`

Example

下列範例會建立名為 `myoutpostdbinstance` 的 MySQL 資料庫執行個體，其備份存放在您的 Outpost 上。

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier myoutpostdbinstance \  
  --engine-version 8.0.17 \  
  --db-instance-class db.m5.large \  
  --engine mysql \  
  --availability-zone us-east-1d \  
  --vpc-security-group-ids outpost-sg \  
  --db-subnet-group-name myoutpostdbsubnetgr \  
  --allocated-storage 100 \  
  --max-allocated-storage 1000 \  
  --master-username masterawsuser \  
  --master-user-password password
```



```
--manage-master-user-password \  
--backup-retention-period 3 \  
--backup-target outposts \  
--storage-encrypted \  
--kms-key-id mykey
```

在Windows中：

```
aws rds create-db-instance ^  
--db-instance-identifier myoutpostdbinstance ^  
--engine-version 8.0.17 ^  
--db-instance-class db.m5.large ^  
--engine mysql ^  
--availability-zone us-east-1d ^  
--vpc-security-group-ids outpost-sg ^  
--db-subnet-group-name myoutpostdbsubnetgr ^  
--allocated-storage 100 ^  
--max-allocated-storage 1000 ^  
--master-username masterawsuser ^  
--manage-master-user-password ^  
--backup-retention-period 3 ^  
--backup-target outposts ^  
--storage-encrypted ^  
--kms-key-id mykey
```

如需建立資料庫執行個體時各項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

RDS API

若要使用 RDS API 在 Outpost 中建立新的資料庫執行個體，請先呼叫 [CreateSubnetGroup](#) 作業，建立供 Outposts 上 RDS 使用的資料庫子網路群組。針對 SubnetIds，指定 Outpost 中的子網路群組，以供 RDS on Outposts 使用。

接下來，請以下列參數呼叫 [CreateDBInstance](#) 作業。指定 Outpost 的可用區域、與 Outpost 建立關聯的 Amazon VPC 安全群組，以及為 Outpost 建立的資料庫子網路群組。

- AllocatedStorage
- AvailabilityZone
- BackupRetentionPeriod
- BackupTarget

如果您要建立多可用區域資料庫執行個體部署，則無法將 outposts 用於 BackupTarget。此外，如果您將 outposts 用於 BackupTarget，資料庫執行個體就無法具有僅供讀取複本。

- DBInstanceClass
- DBInstanceIdentifier
- VpcSecurityGroupIds
- DBSubnetGroupName
- Engine
- EngineVersion
- MasterUsername
- MasterUserPassword
- MaxAllocatedStorage (選用)
- MultiAZ (選用)
- StorageEncrypted
- KmsKeyId

如需建立資料庫執行個體時各項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

在上建立 Amazon RDS 的僅供讀取複本 AWS Outposts

Amazon RDS on AWS Outposts 使用 MySQL 和 PostgreSQL 資料庫引擎的內建複寫功能，從來源資料庫執行個體建立僅供讀取複本。來源資料庫執行個體會成為主要資料庫執行個體。對主要資料庫執行個體所做的更新，會以非同步方式複製到僅供讀取複本。您可以將讀取查詢從應用程式路由到僅供讀取複本，以減少主要資料庫執行個體的負載。使用僅供讀取複本，您可以彈性地擴展超越單一資料庫執行個體的容量限制，以處理高讀取量的資料庫工作負載。

當您從 RDS on Outposts 資料庫執行個體建立僅供讀取複本時，僅供讀取複本可以使用客戶擁有的 IP 地址 (CoIP)。如需詳細資訊，請參閱 [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#)。

RDS on Outposts 上的僅供讀取複本具有下列限制：

- 您無法在 RDS on Outposts 資料庫執行個體上建立 RDS for SQL Server 的僅供讀取複本。
- RDS on Outposts 上不支援跨區域僅供讀取複本。
- RDS on Outposts 上不支援階層式僅供讀取複本。
- 來源 RDS on Outposts 資料庫執行個體不能具有本機備份。來源資料庫執行個體的備份目標必須是您的 AWS 區域。
- 僅供讀取複本需要客戶擁有的 IP (CoIP) 集區。如需詳細資訊，請參閱 [適用於 AWS Outposts 上 Amazon RDS 之客戶擁有的 IP 地址](#)。
- Outposts 上 RDS 上的僅供讀取複本只能在與來源資料庫執行個體相同的虛擬私有雲端 (VPC) 中建立。
- Outposts 上 RDS 上的僅供讀取複本可以位於與來源資料庫執行個體相同的 VPC 中的相同 Outpost 或另一個 Outpost 上。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 從 Outposts 上的 RDS 資料庫執行個體建立僅供讀取複本。如需僅供讀取複本的詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

主控台

若要從來源資料庫執行個體建立僅供讀取複本

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇要作為僅供讀取複本來源的資料庫執行個體。

4. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
5. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入僅供讀取複本的名稱。
6. 指定 Outposts Connectivity (Outposts 連線能力) 的設定。這些設定適用於使用虛擬私有雲端 (VPC) (具有資料庫執行個體的資料庫子網路) 的 Outpost。您的 VPC 必須以 Amazon VPC 服務為基礎。
7. 選擇您的 DB instance class (資料庫執行個體類別)。建議您使用與僅供讀取複本來源資料庫執行個體相同或更大的資料庫執行個體類別和儲存體類型。
8. 若為 Multi-AZ deployment (多可用區部署)，請選擇 Create a standby instance (recommended for production usage) (建立備用執行個體，推薦用於生產用途)，在不同的可用區域中建立備用資料庫執行個體。

建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。

9. (選用) 在 Connectivity (連線能力) 下，設定 Subnet Group (子網路群組) 和 Availability Zone (可用區域) 的值。

如果您同時指定 Subnet Group (子網路群組) 和 Availability Zone (可用區域) 的值，則系統會在與資料庫子網路群組中可用區域相關聯的 Outpost 上建立僅供讀取複本。

如果您針對 Subnet Group (子網路群組) 指定一值，並針對 Availability Zone (可用區域) 指定 No preference (無偏好設定)，則系統會在資料庫子網路群組的隨機 Outpost 上建立僅供讀取複本。

10. 對於 AWS KMS key，選擇 KMS 金鑰的 AWS KMS key 識別碼。

僅供讀取複本必須加密。

11. 視需要選擇其他選項。
12. 選擇 Create read replica (建立僅供讀取複本)。

建立讀取複本之後，您可以在 RDS 主控台的 Databases (資料庫) 頁面上看到它。它在 Role (角色) 欄中顯示 Replica (複本)。

AWS CLI

[若要從來源 MySQL 或 PostgreSQL 資料庫執行個體建立僅供讀取複本，請使用 AWS CLI 指令 `create-db-instance-read` 複本。](#)

您可以指定 `--db-subnet-group-name` 和 `--availability-zone` 選項來控制建立僅供讀取複本的位置：

- 如果您同時指定 `--db-subnet-group-name` 和 `--availability-zone` 選項，則系統會在與資料庫子網路群組中可用區域相關聯的 Outpost 上建立僅供讀取複本。
- 如果您指定 `--db-subnet-group-name` 選項，但未指定 `--availability-zone` 選項，則系統會在資料庫子網路群組的隨機 Outpost 上建立僅供讀取複本。
- 如果您未指定任何一選項，則系統會在與來源 RDS on Outposts 資料庫執行個體相同的 Outpost 上建立僅供讀取複本。

下列範例會透過包含 `--db-subnet-group-name` 和 `--availability-zone` 選項，建立複本並指定僅供讀取複本的位置。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --source-db-instance-identifier mydbinstance \  
  --db-subnet-group-name myoutpostdbsubnetgr \  
  --availability-zone us-west-2a
```

在Windows中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^  
  --source-db-instance-identifier mydbinstance ^  
  --db-subnet-group-name myoutpostdbsubnetgr ^  
  --availability-zone us-west-2a
```

RDS API

若要從來源 MySQL 或 PostgreSQL 資料庫執行個體建立僅供讀取複本，請使用下列必要參數呼叫 Amazon RDS API [建立資料庫InstanceReadReplica](#)操作：

- DBInstanceIdentifier
- SourceDBInstanceIdentifier

您可以指定 DBSubnetGroupName 和 AvailabilityZone 參數來控制建立僅供讀取複本的位置：

- 如果您同時指定 DBSubnetGroupName 和 AvailabilityZone 參數，則系統會在與資料庫子網路群組中可用區域相關聯的 Outpost 上建立僅供讀取複本。
- 如果您指定 DBSubnetGroupName 參數，但未指定 AvailabilityZone 參數，則系統會在資料庫子網路群組的隨機 Outpost 上建立僅供讀取複本。
- 如果您未指定任何一參數，則系統會在與來源 RDS on Outposts 資料庫執行個體相同的 Outpost 上建立僅供讀取複本。

在 AWS Outposts 上還原 Amazon RDS 上資料庫執行個體的注意事項

在 AWS Outposts 上還原 Amazon RDS 中的資料庫執行個體時，您可為還原資料庫執行個體的自動備份和手動快照選擇儲存位置。

- 從手動資料庫快照進行還原時，您可將備份儲存於父系 AWS 區域中或本機存放在您的 Outpost 上。
- 從自動備份進行還原 (時間點復原)，您的選擇較少：
 - 若從父系 AWS 區域進行還原，您可將備份存放在 AWS 區域中或 Outpost 上。
 - 如果從您的 Outpost 還原，您只能將備份存放在您的 Outpost 上。

使用 Amazon RDS Proxy

透過使用 Amazon RDS Proxy，您可以允許應用程式集中和共用資料庫連線，以改善其擴展能力。RDS Proxy 會自動連線至待命資料庫執行個體，同時保留應用程式連線，使應用程式更具有資料庫故障彈性。透過使用 RDS Proxy，您也可以針對資料庫強制執行 AWS Identity and Access Management (IAM) 身份驗證，並將登入資料安全地存放在 AWS Secrets Manager。

使用 RDS Proxy，您可以處理資料庫流量中無法預測的突增情況。否則，這些激增可能會因為超額訂閱連線或以快速建立新連線而導致問題。RDS Proxy 會建立資料庫連線集區，並重複使用此集區中的連線。這種方法可以避免每次開啟新資料庫連線的記憶體和 CPU 額外負荷。若要防止資料庫過度訂閱，您可以控制建立的資料庫連線數目。

RDS Proxy 會佇列或節流無法立即從連線集區提供服務的應用程式連線。雖然延遲可能會增加，但您的應用程式可以繼續擴展，而不會突然失敗或讓資料庫無法負荷。如果連線請求超過了您指定的限制，則 RDS Proxy 會拒絕應用程式連線 (也就是說，它會分散負載)，同時，它會針對 RDS 可以使用可用容量的負載維持可預測的效能。

您可以減少處理登入憑證，並為每個新連線建立安全連線的負荷。RDS Proxy 可以代表資料庫處理其中的一些工作。

RDS Proxy 與其支援的引擎版本完全相容。您可以針對大部分的應用程式啟用 RDS Proxy，而不需要變更程式碼。

主題

- [區域和版本可用性](#)
- [RDS Proxy 的配額和限制](#)
- [規劃在哪裡使用 RDS Proxy](#)
- [RDS Proxy 概念和術語](#)
- [RDS Proxy 入門](#)
- [管理 RDS Proxy](#)
- [使用 Amazon RDS Proxy 端點](#)
- [使用 Amazon 監控 RDS 代理指標 CloudWatch](#)
- [使用 RDS Proxy 事件](#)
- [RDS Proxy 命令列範例](#)

- [RDS Proxy 的故障診斷](#)
- [將 RDS Proxy 與 AWS CloudFormation 搭配使用](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 Amazon RDS 搭配 RDS Proxy 版本和區域可用性的詳細資訊，請參閱 [支援 Amazon RDS 代理伺服器的區域和資料庫引擎](#)。

RDS Proxy 的配額和限制

下列配額和限制適用於 RDS Proxy：

- 每個 AWS 帳戶 ID 限制為 20 個代理伺服器。如果您的應用程式需要更多 Proxy，請透過 AWS Management Console 在「Service Quotas」頁面中，選取 Amazon Relational Database Service (Amazon RDS) 並找到代理伺服器以請求增加配額。AWS 可以自動增加您的配額或等待審核您的請求 AWS Support。
- 每個代理最多可以有 200 個相關聯的 Secrets Manager 秘密。因此，每個代理可在任何指定時間連接至多達 200 個不同的使用者帳戶。
- 每個代理都有一個默認端點。您也可以為每個代理伺服器新增最多 20 個代理主機端點。您可以建立、檢視、修改和刪除這些端點。
- 對於複寫組態中的 RDS 資料庫執行個體，您只能將代理與寫入器資料庫執行個體，而不是與僅供讀取複本關聯。
- 您的 RDS Proxy 必須位於與資料庫相同的 Virtual Private Cloud (VPC) 中。代理不可以公開存取，但資料庫可以。例如，如果您要在本機主機上建立資料庫的原型，則無法連線到 Proxy，除非您設定必要的網路需求以允許連線到 Proxy。這是因為您的本機主機不在 Proxy 的 VPC 之外。
- 您無法將 RDS Proxy 與其租用設定為 dedicated 的 VPC 搭配使用。
- 如果您將 RDS Proxy 與已啟用 IAM 驗證的 RDS 資料庫執行個體 搭配使用，請檢查使用者驗證。所有透過代理連線的使用者都必須透過登入憑證進行身分驗證。如需 RDS Proxy 中 Secrets Manager 和 IAM 支援的詳細資訊，請參閱 [設定資料庫認證 AWS Secrets Manager](#) 和 [設定 AWS Identity and Access Management \(IAM\) 政策](#)。
- 使用 SSL 主機名稱驗證時，您無法將 RDS Proxy 與自訂 DNS 搭配使用。
- 每個代理都可以與單一目標資料庫執行個體相關聯。不過，您可以將多個代理與同一個資料庫執行個體建立關聯。

- 文字大小大於 16 KB 的任何陳述式會導致代理將工作階段鎖定至目前的連線。
- 某些區域具有要在建立 Proxy 時考量的可用區域 (AZ) 限制。美國東部 (維吉尼亞北部) 區域不支援 use1-az3 可用區域中的 RDS Proxy。美國西部 (加利佛尼亞北部) 區域不支援 usw1-az2 可用區域中的 RDS Proxy。在建立 Proxy 時，若選取子網路，請確定您未在上述可用區域中選取子網路。
- 目前，RDS Proxy 不支援任何全域條件內容金鑰。

如需有關全域條件內容索引鍵的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容索引鍵](#)。

如需每個資料庫引擎的其他限制，請參閱下列各節：

- [RDS for MariaDB 的其他限制](#)
- [RDS for Microsoft SQL Server 的其他限制](#)
- [RDS for MySQL 的其他限制](#)
- [RDS for PostgreSQL 的其他限制](#)

RDS for MariaDB 的其他限制

下列額外限制適用於具有 RDS for MariaDB 資料庫的 RDS Proxy：

- 目前，所有代理都在 MariaDB 的連接埠 3306 上接聽。代理仍會使用您在資料庫設定中指定的連接埠來連線至您的資料庫。
- 您無法使用 RDS Proxy，搭配 Amazon EC2 執行個體中的自我管理 MariaDB 資料庫。
- 您無法將 RDS Proxy 與其資料庫參數群組中 `read_only` 參數設定為 1 的 RDS for MariaDB 資料庫執行個體搭配使用。
- RDS 代理伺服器不支援 MariaDB 壓縮模式。例如，其不支援 `mysql` 命令的 `--compress` 或 `-C` 選項使所用的壓縮。
- 某些 SQL 陳述式和函數可變更連線狀態，而不會導致鎖定。如需最新的鎖定行為，請參閱 [避免鎖定](#)。
- RDS Proxy 不支援 MariaDB `auth_ed25519` 外掛程式。
- RDS Proxy 不支援 MariaDB 資料庫的 Transport Layer Security (TLS) 1.3 版。
- 當 RDS Proxy 重複使用相同的資料庫連線來執行另一個查詢時，處理 `GET DIAGNOSTIC` 命令的資料庫連線可能會傳回不正確的資訊。當 RDS Proxy 多工進行資料庫連線時，可能會發生這種情況。如需詳細資訊，請參閱 [RDS Proxy 概念概觀](#)。

⚠ Important

對於與 MariaDB 資料庫相關聯的代理，請勿在初始化查詢中 `sql_auto_is_null` 將配置參數設置為 `true` 或非零值。這樣做可能會導致應用程式行為不正確。

RDS for Microsoft SQL Server 的其他限制

下列額外限制適用於具有 RDS for Microsoft SQL Server 資料庫的 RDS Proxy：

- 您需要為代理建立的 Secrets Manager 秘密數目取決於資料庫執行個體使用的定序。例如，假設您的資料庫執行個體使用區分大小寫的定序。如果您的應用程式同時接受 "Admin" 和 "admin"，則您的代理需要兩個單獨的秘密。如需有關 SQL Server 中定序的詳細資訊，請參閱 [Microsoft SQL Server](#) 文件。
- RDS Proxy 不支援使用 Active Directory 的連線。
- 您無法使用 IAM 身分驗證，搭配不支援權杖屬性的用戶端。如需詳細資訊，請參閱 [使用 Microsoft SQL Server 連線至代理的考量](#)。
- @@IDENTITY、@@ROWCOUNT、和 SCOPE_IDENTITY 的結果未必永遠準確。作為解決方法，請在同一工作階段陳述式中擷取其值，以確保它們傳回正確的資訊。
- 如果連線使用多個作用中結果集 (MARS)，則 RDS Proxy 不會執行初始化查詢。如需 MARS 的相關資訊，請參閱 [Microsoft SQL Server](#) 文件。
- 目前，RDS 代理伺服器不支援在主要版本 SQL 伺服器 2022 上執行的 SQL 伺服器資料庫執行個體的 RDS。
- RDS 代理不支援在主要版本 SQL 伺服器 2014 上執行的 SQL 伺服器資料庫執行個體的 RDS。
- RDS Proxy 不支援無法處理單一 TLS 記錄中多個回應訊息的用戶端應用程式。

RDS for MySQL 的其他限制

下列額外限制適用於具有 RDS for MySQL 資料庫的 RDS Proxy：

- RDS Proxy 不支援 MySQL `sha256_password` 和 `caching_sha2_password` 身分驗證外掛程式。這些外掛程式會實作使用者帳戶密碼的 SHA-256 雜湊。
- 目前，所有代理都在 MySQL 的連接埠 3306 上接聽。代理仍會使用您在資料庫設定中指定的連接埠來連線至您的資料庫。
- 您不能將 RDS Proxy 用於 EC2 執行個體中的自我管理 MySQL 資料庫。

- 您無法將 RDS Proxy 與其資料庫參數群組中 `read_only` 參數設定為 1 的 RDS for MySQL 資料庫執行個體搭配使用。
- RDS Proxy 不支援 MySQL 壓縮模式。例如，其不支援 `mysql` 命令的 `--compress` 或 `-C` 選項所使用的壓縮。
- 當 RDS Proxy 重複使用相同的資料庫連線來執行另一個查詢時，處理 `GET DIAGNOSTIC` 命令的資料庫連線可能會傳回不正確的資訊。當 RDS Proxy 多工進行資料庫連線時，可能會發生這種情況。
- 某些 SQL 陳述式和函式 (例如) `SET LOCAL` 可以變更連線狀態，而不會造成釘選。如需最新的鎖定行為，請參閱 [避免鎖定](#)。
- 不支援在多重陳述式查詢中使用此 `ROW_COUNT()` 函數。
- RDS Proxy 不支援無法處理單一 TLS 記錄中多個回應訊息的用戶端應用程式。

Important

對於與 MySQL 資料庫相關聯的代理，請勿在初始化查詢中 `sql_auto_is_null` 將配置參數設置為 `true` 或非零值。這樣做可能會導致應用程式行為不正確。

RDS for PostgreSQL 的其他限制

下列額外限制適用於具有 RDS for PostgreSQL 資料庫的 RDS Proxy：

- RDS Proxy 不支援 PostgreSQL 工作階段鎖定篩選條件。
- 目前，所有代理都在 PostgreSQL 的連接埠 5432 上接聽。
- 針對 PostgreSQL，RDS Proxy 目前不支援藉由發出 `CancelRequest` 取消來自用戶端的查詢。例如，使用 `Ctrl+C` 在互動式 `psql` 工作階段中取消長時間執行的查詢時，就是這種情況。
- PostgreSQL 函數 [lastval](#) 的結果未必永遠準確。您可以採取的因應措施是，將 [INSERT](#) 陳述式與 `RETURNING` 子句搭配使用。
- RDS 代理目前不支援串流複寫模式。
- 使用適用於 PostgreSQL 16 的 RDS，對 `scram_iterations` 值的修改僅會影響代理伺服器與資料庫之間的驗證程序。具體來說，如果您設定 `ClientPasswordAuthType` 為 `scram-sha-256`，對該 `scram_iterations` 值進行的任何自訂都不會影響 `client-to-proxy` 密碼驗證。相反地，`client-to-proxy` 密碼驗證的迭代值固定為 4096。

⚠ Important

對於具有 PostgreSQL 資料庫的現有代理，如果您將資料庫驗身分證修改為僅使用 SCRAM，則該代理會變成無法使用，最多持續 60 秒。若要避免發生此問題，請執行下列其中一項：

- 確定資料庫同時允許 SCRAM 和 MD5 身分驗證。
- 若只要使用 SCRAM 身分驗證，請建立新代理、將您的應用程式流量遷移至新代理，然後刪除先前與資料庫相關聯的代理。

規劃在哪裡使用 RDS Proxy

您可以決定哪些資料庫執行個體、叢集和應用程式可能因為使用 RDS Proxy 而獲益最大。若要執行這項操作，請考慮下列因素：

- 任何遇到「線數太多」錯誤的資料庫執行個體都是與代理關聯的理想候選者。這通常以高度 ConnectionAttempts CloudWatch 量值為特徵。代理讓應用程式能開啟多個用戶端連線，代理則管理較少數量到資料庫執行個體的長時間連線。
- 對於使用較小執行個體類別 (AWS 例如 T2 或 T3) 的資料庫執行個體，使用 Proxy 可協助避免 out-of-memory 情況。它也可以協助減少建立連線的 CPU 額外負荷。處理大量連線時，可能會發生這些情況。
- 您可以監控某些 Amazon CloudWatch 指標，以判斷資料庫執行個體是否接近特定類型的限制。這些限制是指連線數目以及與連線管理相關聯的記憶體。您也可以監視某些 CloudWatch 指標，以判斷資料庫執行個體是否正在處理許多短期連線。開啟和關閉這類連線可能會對資料庫造成效能負荷。如需欲監控指標的相關資訊，請參閱 [使用 Amazon 監控 RDS 代理指標 CloudWatch](#)。
- AWS Lambda 函數也是搭配代理使用的理想候選者。這些函數經常進行短暫的資料庫連線，這些連線將受益於 RDS Proxy 提供的連線集區。您可以利用 Lambda 函數中已有的任何 IAM 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫登入資料。
- 通常會大量開啟和關閉的資料庫連線，而且沒有內建的連線集區機制的應用程式，都很適合使用 proxy。
- 長期保持大量連線開啟的應用程式通常是搭配代理使用的理想候選者。軟體即服務 (SaaS) 或電子商務等產業中的應用程式通常會保持連線開啟，將資料庫要求的延遲降至最低。使用 RDS Proxy，應用程式在直接連線至資料庫執行個體時，可以保持更多的連線開啟。
- 您可能因為針對所有資料庫執行個體設定身分驗證的複雜性，尚未採用 IAM 身分驗證和 Secrets Manager。如果是這樣，您可以保留現有的身分驗證方法，並將身分驗證委派給代理。代理可以針對

特定應用程式強制執行用戶端連線的身分驗證政策。您可以利用 Lambda 函數中已有的任何 IAM 身分驗證，而不是在 Lambda 應用程式的程式碼中管理資料庫登入資料。

- RDS Proxy 可協助讓應用程式對資料庫失敗更具彈性且更為透明。RDS Proxy 會略過網域名稱系統 (DNS) 快取，將 Amazon RDS Multi-AZ 資料庫執行個體的容錯移轉時間縮短高達 66%。RDS Proxy 也會自動將流量路由到新的資料庫執行個體，同時保留應用程式連線。這使得應用程式的容錯移轉更加透明。

RDS Proxy 概念和術語

您可以使用 RDS Proxy 來簡化 Amazon RDS 資料庫執行個體的連線管理。

RDS Proxy 處理用戶端應用程式與資料庫之間的網路流量。它以主動方式達成此功能，首先了解資料庫協定，然後根據應用程式的 SQL 操作和資料庫中的結果集調整其行為。

RDS Proxy 減輕了資料庫連線管理的記憶體和 CPU 負荷。當應用程式同時開啟許多連線時，資料庫需要的記憶體和 CPU 資源比較少。它關閉並重新開啟長時間閒置的連線時，也不需要依據應用程式的邏輯。同樣地，當資料庫發生問題時，重新建立連線需要的應用程式邏輯也比較少。

RDS Proxy 的基礎設施具備高度可用性，並可部署到數個可用區域 (AZ) 上。RDS Proxy 的運算、記憶體和儲存體獨立於 RDS 資料庫執行個體。這種分離有助於降低資料庫伺服器的負荷，使它們可以投入資源來服務資料庫工作負載。RDS Proxy 運算資源可在無伺服器的狀態下執行，根據您的資料庫工作負載自動擴展。

主題

- [RDS Proxy 概念概觀](#)
- [連線集區](#)
- [RDS Proxy 安全性](#)
- [容錯移轉](#)
- [交易](#)

RDS Proxy 概念概觀

RDS Proxy 運用基礎設施來執行連線集區以及以下各節描述的其他功能。您可以在代理頁面上看到 RDS 主控台中所代表的代理。

每個代理處理與單一 RDS 資料庫執行個體的連線。Proxy 會自動決定 RDS Multi-AZ 資料庫執行個體的目前寫入器執行個體。

Proxy 保持開啟且可供資料庫應用程式使用的連線組成連線集區。

根據預設，RDS Proxy 可以在工作階段中的每一筆交易結束之後重新使用連線。此一交易層級重複使用稱為多工。當 RDS Proxy 從連線集區暫時移除連線以重複使用時，該操作就是所謂的借用連線。在這麼做是安全的情況下，RDS Proxy 會將該連線傳回至連線集區。

在某些情況下，RDS Proxy 無法確定在目前工作階段之外重複使用資料庫連線是否安全。這時，它會讓工作階段保持在同一連線上，直到該工作階段結束。這種回復行為稱為鎖定。

代理具有預設端點。當使用 Amazon RDS 資料庫執行個體時，您會連線到此端點。您會這樣做，而不是連線到直接連線至執行個體的讀取/寫入端點。對於 RDS 資料庫叢集，您也可以建立其他讀取/寫入和唯讀端點。如需詳細資訊，請參閱 [代理端點概觀](#)。

例如，您仍然可以連線到叢集端點以進行讀取/寫入連線，而不需要連線集區。您仍然可以連線至讀取器端點，以進行唯讀連線的負載平衡。您仍然可以連線到執行個體端點，以便對叢集內的特定資料庫執行個體進行診斷和疑難排解。如果您使用其他 AWS 服務 (例如 AWS Lambda 連線至 RDS 資料庫)，請變更其連線設定以使用 Proxy 端點。例如，您可以指定代理端點，以允許 Lambda 函數存取您的資料庫，同時利用 RDS Proxy 功能。

每個代理都包含一個目標群組。此目標群組體現了代理可連線的 RDS 資。與代理相關聯的 RDS 資料庫執行個體 稱為該代理的目標。為了方便起見，當您透過主控台建立代理伺服器時，RDS Proxy 也會建立對應的目標群組，並自動註冊關聯的目標。

引擎系列是一組使用相同資料庫協定的相關資料庫引擎。您可以為您建立的每一個代理選擇引擎系列。

連線集區

每個 Proxy 都會針對其相關聯 RDS 資料庫的寫入器執行個體執行連線集區。連線集區是一種最佳化，可以減輕開啟和關閉連線，以及保持許多連線同時開啟的相關負荷。此一負荷包含處理每個新連線所需的記憶體。它還涉及關閉每個連線和開啟新連線的 CPU 額外負荷。範例包括 Transport Layer Security/Secure Sockets Layer (TLS/SSL) 交握、身分驗證、協商功能等。連線集區簡化了應用程式的邏輯。您不需要撰寫應用程式碼，以便同時開啟的連線數降到最低。

每個 Proxy 也會執行連線多工，又稱為連線重用。RDS Proxy 運用「多工」，使用一個基礎資料庫連線執行一筆交易的所有操作。然後，RDS 可以使用不同的連線處理下一筆交易。您可以同時開啟到代理的多個連線，代理則使較少的資料庫執行個體或叢集連線保持開啟。這樣做可以進一步減輕資料庫伺服器上連線的記憶體負荷。此技術也可以降低發生「連線數太多」錯誤的可能性。

RDS Proxy 安全性

RDS Proxy 會使用現有的 RDS 安全機制，例如 TLS/SSL 和 AWS Identity and Access Management (IAM)。如需這些安全功能的一般資訊，請參閱 [Amazon RDS 中的安全](#)。此外，請務必熟悉 RDS 如何使用身分驗證、授權和其他安全領域。

RDS Proxy 可以充當用戶端應用程式和基礎資料庫之間的額外安全層。例如，即使基礎資料庫執行個體支援舊版 TLS，您也可以使用 TLS 1.3 連線到代理伺服器。您可以使用 IAM 角色連線到代理。即使代理使用原生使用者和密碼身分驗證方法來連線至資料庫，仍然如此。您可以使用這種技術，強制資料庫應用程式執行嚴格的身分驗證要求，而無需在資料庫執行個體上進行昂貴的遷移工作。

您可以將 RDS 代理使用的資料庫認證儲存在中 AWS Secrets Manager。代理伺服器存取的 RDS 資料庫執行個體的每個資料庫使用者都必須在 Secret Manager 中具有對應的密碼。您也可以為 RDS Proxy 的使用者設定 IAM 身分驗證。如此一來，即使資料庫仍使用原生密碼身分驗證，您也可以對資料庫存取強制執行 IAM 身分驗證。與其將資料庫登入資料嵌入在應用程式的程式碼中，建議您使用這些安全功能。

搭配 RDS Proxy 使用 TLS/SSL

您可以使用 TLS/SSL 通訊協定連線到 RDS Proxy。

Note

RDS 代理伺服器會使用來自 AWS Certificate Manager (ACM) 的憑證。如果您正在使用 RDS Proxy，您不需要下載 Amazon RDS 憑證或更新使用 RDS Proxy 連線的應用程式。

若要針對 Proxy 和資料庫之間的所有連線強制執行 TLS，您可以在中建立或修改 Proxy 時指定「需要傳輸層安全性」設定 AWS Management Console。

RDS Proxy 也能確保您的工作階段在用戶端與 RDS Proxy 端點之間使用 TLS/SSL。若要讓 RDS Proxy 執行此作業，請在用戶端指定需求。SSL 工作階段變數不會針對使用 RDS Proxy 之資料庫的 SSL 連線進行設定。

- 若為 RDS for MySQL，請在執行 `mysql` 命令時，使用 `--ssl-mode` 參數在用戶端指定需求。
- 若為 Amazon RDS PostgreSQL，請在執行 `psql` 命令時，指定 `sslmode=require` 做為 `conninfo` 字串的一部分。

RDS 代理伺服器支援 TLS 通訊協定 1.0、1.1、1.2 和 1.3 版本。您可以使用比基礎資料庫中所使用之 TLS 更新的版本連線至代理。

根據預設，用戶端程式會使用 RDS Proxy 建立加密連線，並且可透過 `--ssl-mode` 選項取得進一步的控制。RDS Proxy 在用戶端支援所有 SSL 模式。

針對用戶端，SSL 模式如下：

PREFERRED

SSL 為第一個選擇，但並非必要。

DISABLED

不允許任何 SSL。

REQUIRED

強制採用 SSL。

VERIFY_CA

強制採用 SSL 並驗證憑證授權單位 (CA)。

VERIFY_IDENTITY

強制採用 SSL 並驗證 CA 和 CA 主機名稱。

在搭配 `--ssl-mode VERIFY_CA` 或 `VERIFY_IDENTITY` 使用用戶端時，請指定指向 `--ssl-ca` 格式 CA 的 `.pem` 選項。對於要使用的 `.pem` 檔案，請從 [Amazon Trust Services](#) 下載所有根 CA PEM 並將它們放入單一 `.pem` 檔案。

RDS Proxy 使用萬用字元憑證，這些憑證同時適用於網域及其子網域。如果您使用 `mysql` 用戶端來利用 SSL 模式 `VERIFY_IDENTITY` 連接，則目前必須使用 MySQL 8.0 相容的 `mysql` 命令。

容錯移轉

容錯移轉是一種高可用性功能，當原始執行個體無法使用時，用另一個來取代該資料庫執行個體。發生容錯移轉可能是因為資料庫執行個體發生問題。它也可能是正常維護程序的一部分，例如在資料庫升級期間。容錯移轉適用於多可用區域組態中的 RDS 資料庫執行個體。

透過 Proxy 連線可讓您的應用程式對資料庫容錯移轉更具彈性。當原始資料庫執行個體無法使用時，RDS Proxy 會連線至待命資料庫，而不會捨棄閒置的應用程式連線。這有助於加速並簡化容錯移轉程序。與典型的重新啟動或資料庫問題相比，這對應用程式的干擾程度較小。

如果沒有 RDS Proxy，容錯移轉將導致短暫的執行中斷。在中斷期間，您無法在容錯移轉中對資料庫執行寫入作業。任何現有的資料庫連線都會中斷，您的應用程式必須將其重新開啟。當提升唯讀資料庫執行個體，以取代無法使用的執行個體時，該資料庫將可使用於新連線和寫入操作。

在資料庫容錯移轉期間，RDS Proxy 繼續接受位於相同 IP 地址的連線，並自動將連線導向至新的主要資料庫執行個體。透過 RDS Proxy 連線的用戶端不會受到下列項目影響：

- 容錯移轉時的網域名稱系統 (DNS) 傳播延遲。
- 本機 DNS 快取。
- 連線逾時。
- 不確定哪個資料庫執行個體是目前的寫入器。
- 等候來自己不可使用，卻未關閉連線的前寫入器的查詢回應。

對於自行維護連線集區的應用程式而言，經歷 RDS Proxy 意味著大部分的連線在容錯移轉或其他中斷期間保持活動狀態。只會取消處於交易或 SQL 陳述式中間的連線。RDS Proxy 會立即接受新的連線。當資料庫寫入器無法使用時，RDS Proxy 會將傳入的請求排入佇列。

對於不自行維護連線集區的應用程式，RDS Proxy 提供了更快的連線速率和更多的開放連線。它免除了從資料庫頻繁重新連接的昂貴負荷。它會重複使用 RDS Proxy 的連線集區中維護的資料庫連線，而達成此功能。此方法對於 TLS 連線來說尤其重要，因為設定成本很大。

交易

單一交易中的所有陳述式總是使用相同的基礎資料庫連線。當交易結束時，連線將可提供給不同的工作階段使用。使用交易做為粒度單位會產生下列後果：

- 開啟 RDS for MySQL `autocommit` 設定時，連線可能會在每一個別陳述式之後重複使用。
- 相反地，`autocommit` 設定關閉時，您在工作階段中發出的第一個陳述式會開始新的交易。例如，假設您輸入一連串的 `SELECT`、`INSERT`、`UPDATE` 和其他資料操作語言 (DML) 陳述式。在此情況下，直到您發出 `COMMIT`、`ROLLBACK` 或以另外方式結束交易之前，都不會重複使用連線。
- 輸入資料定義語言 (DDL) 陳述式將使交易在陳述式完成後結束。

RDS Proxy 透過資料庫用戶端應用程式使用的網路協定偵測交易何時結束。交易偵測並不依賴 SQL 陳述式的文字中出現的 `COMMIT` 或 `ROLLBACK` 等關鍵字。

在某些情況下，RDS Proxy 可能會偵測到資料庫要求，使得將工作階段移到不同的連線變得不切實際。這時，它會在剩餘的工作階段關閉該連線的多工功能。如果 RDS Proxy 無法確定多工在此工作階

段中實際可行，則適用相同的規則。此操作稱為鎖定。如需偵測和最小化鎖定的方法，請參閱 [避免鎖定](#)。

RDS Proxy 入門

在以下各節中，您可以找到如何設定和管理 RDS 代理伺服器。您也可以找到如何設定相關的安全選項。這些選項可控制誰可以存取每個 Proxy，以及每個 Proxy 連線到資料庫執行個體的方式。

主題

- [設定網路先決條件](#)
- [設定資料庫認證 AWS Secrets Manager](#)
- [設定 AWS Identity and Access Management \(IAM\) 政策](#)
- [建立 RDS Proxy](#)
- [檢視 RDS Proxy](#)
- [透過 RDS Proxy 連線至資料庫](#)

設定網路先決條件

使用 RDS 代理時，您必須在 RDS 資料庫執行個體和 RDS 代理之間擁有一個通用的虛擬私有雲端 (VPC)。此 VPC 應至少有兩個位於不同可用區域的子網路。您的帳戶可以擁有這些子網路，或與其他帳戶共用。如需 VPC 共用的詳細資訊，請參閱 [使用共用 VPC](#)。

您的用戶端應用程式資源 (例如 Amazon EC2、Lambda 或 Amazon ECS) 可以位於與代理相同的 VPC 中。或者，它們可以位於與代理不同的 VPC 中。如果您已成功連線到任何 RDS 資料庫執行個體，表示您已擁有所需的網路資源。

主題

- [取得子網路的相關資訊](#)
- [規劃 IP 地址容量](#)

取得子網路的相關資訊

若要建立 Proxy，您必須提供代理伺服器在其中運作的子網路和 VPC。下列 Linux 範例顯示用於檢查您擁有的 VPC 和子網路的 AWS CLI 命令。AWS 帳戶特別是，當您使用 CLI 建立代理時，您會傳遞子網路 ID 做為參數。

```
aws ec2 describe-vpcs
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

下列 Linux 範例顯示用於判斷與特定 RDS 資料庫執行個體 Aurora 資料庫對應之子網路 ID 的 AWS CLI 命令。尋找資料庫執行個體的 VPC ID。檢查 VPC 以尋找其子網路。下面的 Linux 示例演示了如何。

```
$ #From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet IDs.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].[DBSubnetGroup][0][0][Subnets][0][*].SubnetIdentifier' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
...
```

```
$ #From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].[DBSubnetGroup][0][0].VpcId' --output text
```

```
my_vpc_id
```

```
$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].[SubnetId]' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
subnet_id_4
subnet_id_5
subnet_id_6
```

規劃 IP 地址容量

RDS Proxy 會根據註冊的資料庫執行個體大小和數目，視需要自動調整其容量。某些作業可能還需要額外的 Proxy 容量，例如增加已註冊資料庫的大小或內部 RDS Proxy 維護作業。在這些操作期間，您

的代理可能需要更多的 IP 地址來佈建額外的容量。這些額外的位址可讓您的代理擴展，而不會影響您的工作負載。子網路中若少了可用的 IP 地址會阻止代理縱向擴展。這可能會導致更高的查詢延遲或用戶端連線失敗。當您的子網路中沒有足夠的可用 IP 地址時，RDS 會透過事件 RDS-EVENT-0243 通知您。如需此金鑰的相關資訊，請參閱 [使用 RDS Proxy 事件](#)。

以下是根據資料庫執行個體類別大小，建議在子網路中為 Proxy 保留免費的 IP 位址下限。

DB instance class (資料庫執行個體類別)	可用 IP 地址數目下限
db.*.xlarge 或更小	10
db.*.24xlarge	15
db.*.24xlarge	25
db.*.24xlarge	45
db.*.24xlarge	60
db.*.24xlarge	75
db.*.24xlarge	110

這些建議的 IP 位址數目是僅具有預設端點的 Proxy 預估值。具有其他端點或僅供讀取複本的代理可能需要更多可用的 IP 地址。對於每個額外的端點，建議您多保留三個 IP 地址。對於每個僅供讀取複本，建議您根據該僅供讀取複本的大小，保留資料表中指定的額外 IP 地址。

Note

RDS 代理伺服器在 VPC 擬私人雲端中不支援超過 215 個 IP 位址。

設定資料庫認證 AWS Secrets Manager

對於您建立的每個代理，您必須先使用 Secrets Manager 服務來儲存使用者名稱和密碼登入資料集。您可以為代理伺服器在 RDS 資料庫執行個體上連線的每個資料庫使用者帳戶建立個別的 Secret Manager 密碼。

在 Secrets Manager 主控台中，您可以使用 `username` 和 `password` 欄位的值來建立這些密碼。這樣做可讓代理連線至與代理關聯之 RDS 資料庫執行個體上的對應資料庫使用者。若要這樣做，您可以使用其他資料庫的登入資料、RDS 資料庫的登入資料或其他類型的秘密等設定。在 [使用者名稱] 和 [密碼] 欄位中填入適當的值，以及任何其他必要欄位的值。如果秘密中有 Host (主機) 和 Port (連接埠) 等其他欄位，代理會忽略這些欄位。這些詳細資訊由代理自動提供。

您也可以選擇 `Other type of secrets` (其他類型的秘密)。在此情況下，您可以使用名為 `username` 和 `password` 的金鑰來建立秘密。

若要以特定資料庫使用者身分透過 Proxy 連線，請確定與密碼相關聯的密碼與該使用者的資料庫密碼相符。如果不相符，您可以在 Secrets Manager 中更新相關聯的私密。在這種情況下，您仍可連線至私密登入資料和資料庫密碼確實相符的其他帳戶。

Note

對於 SQL Server 的 RDS，RDS 代理伺服器需要密碼管理員中的密碼，不論資料庫執行個體定序設定為何，都必須與應用程式程式碼區分大小寫。例如，如果您的應用程式可以同時使用使用者名稱「Admin」或「admin」，請同時為「管理員」和「admin」設定代理伺服器。在用戶端與 Proxy 之間的驗證程序中，RDS Proxy 無法容納使用者名稱不區分大小寫的情況。如需有關 SQL Server 中定序的詳細資訊，請參閱 [Microsoft SQL Server](#) 文件。

當您透過 AWS CLI 或 RDS API 建立代理伺服器時，您可以指定對應密碼的 Amazon 資源名稱 (ARN)。您可以針對代理可以存取的所有資料庫使用者帳戶執行此操作。在中 AWS Management Console，您可以依其描述性名稱來選擇密碼。

如需有關在 Secrets Manager 中建立秘密的指示，請參閱 Secrets Manager 文件中的 [建立秘密](#) 頁面。請使用下列技術其中之一：

- 在主控台中使用 [Secrets Manager](#)。
- 若要使用 CLI 建立 Secrets Manager 秘密以搭配 RDS Proxy 使用，請使用如下所示的命令。

```
aws secretsmanager create-secret
  --name "secret_name"
  --description "secret_description"
  --region region_name
  --secret-string '{"username":"db_user","password":"db_user_password"}'
```

- 您也可以建立自訂金鑰來加密 Secrets Manager 密碼。下列命令會建立範例金鑰。

```
PREFIX=my_identifier
aws kms create-key --description "$PREFIX-test-key" --policy '{
  "Id": "$PREFIX-kms-policy",
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::account_id:root"},
      "Action": "kms:*", "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal":
      {
        "AWS":
          ["$USER_ARN", "arn:aws:iam:account_id::role/Admin"]
      },
      "Action":
      [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms>Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {"AWS": "$ROLE_ARN"},

```

```

        "Action":["kms:Decrypt","kms:DescribeKey"],
        "Resource": "*"
    }
]
}'

```

例如，下列命令會為兩個資料庫使用者建立 Secret Manager 密碼：

```

aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'

```

若要建立使用自訂 AWS KMS 金鑰加密的這些密碼，請使用下列指令：

```

aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}' \
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}' \
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id

```

若要查看您 AWS 帳戶所擁有的密碼，請使用如下指令。

```
aws secretsmanager list-secrets
```

當您使用 CLI 建立代理時，您會將一個或多個秘密的 Amazon Resource Names (ARN) 傳遞給 `--auth` 參數。下面的 Linux 示例演示了如何準備報告，只使用您的 AWS 帳戶擁有的每個密鑰的名稱和 ARN。此範例使用 `--output table` 版本 2 中可用的 AWS CLI 參數。如果您使用的是 AWS CLI 版本 1，請 `--output text` 改用。

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

若要確認您是否在祕密中以正確的格式儲存了正確的登入資料，請使用下列命令。將簡短名稱或祕密的 ARN 替換為 *your_secret_name*。

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

輸出應該包含一行，顯示如下所示的 JSON 編碼值。

```
"SecretString": "{\"username\":\"your_username\",\"password\":\"your_password\"}"
```

設定 AWS Identity and Access Management (IAM) 政策

在 Secrets Manager 中建立祕密後，您建立可存取這些祕密的 IAM 政策。如需使用 IAM 的一般資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。

Tip

如果您使用的是 IAM 主控台，則適用下列程序。如果您使用 AWS Management Console 適用於 RDS，RDS 可以自動為您建立 IAM 政策。在這種情況下，您可以略過下列程序。

如何建立可存取 Secrets Manager 私密以與代理搭配使用的 IAM 政策

1. 登入 IAM 主控台。遵循建立角色程序，如 [建立 IAM 角色](#)，選擇 [建立角色以將權限委派給 AWS 服務中](#) 所述。

對於信任的實體類型，選擇 AWS 服務。在使用案例下，從其他 AWS 服務的使用案例下拉式清單中選取 RDS。選取 RDS - 將角色新增至資料庫。

2. 對於新角色，請執行新增內嵌政策步驟。使用與 [編輯 IAM 政策](#) 中所述的相同一般程序。將下列 JSON 貼入 JSON 文字方塊中。替換為您自己的帳戶 ID。將您的 AWS 地區替換為 us-east-2。以 Amazon Resource Names (ARNs) 替代您建立的祕密，請參閱 [在 IAM 政策陳述式中指定 KMS 金鑰](#)。對於 kms:Decrypt 動作，請替換預設值的 ARN AWS KMS key 或您自己的 KMS 金鑰。您使用哪一個取決於您已使用哪一個加密 Secrets Manager 祕密。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
```



```

    "Action": "secretsmanager:GetSecretValue",
    "Resource": [
      "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
      "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
    ]
  },
  {
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
      }
    }
  }
]
}

```

3. 編輯此 IAM 角色的信任政策。將下列 JSON 貼入 JSON 文字方塊中。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

下列命令透過 AWS CLI 執行相同的操作。

```

PREFIX=my_identifier
USER_ARN=$(aws sts get-caller-identity --query "Arn" --output text)

aws iam create-role --role-name my_role_name \

```

```

--assume-role-policy-document '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"Service":
["rds.amazonaws.com"]},"Action":"sts:AssumeRole"}]}'

ROLE_ARN=arn:aws:iam::account_id:role/my_role_name

aws iam put-role-policy --role-name my_role_name \
--policy-name $PREFIX-secret-reader-policy --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
        "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
        }
      }
    }
  ]
}'

```

建立 RDS Proxy

若要管理指定的一組資料庫執行個體的連線，您可以建立代理。您可以將代理與 RDS for MariaDB、RDS for Microsoft SQL Server、RDS for MySQL 或 RDS for PostgreSQL 資料庫執行個體建立關聯。

AWS Management Console

若要建立代理

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 選擇 Create Proxy (建立代理)。
4. 選擇代理的所有設定。

針對代理組態，提供下列項目的資訊：

- Engine family (引擎系列)。這個設定會決定在解譯往返資料庫的網路流量時，代理會辨識哪些資料庫網路通訊協定。對於 RDS for MariaDB 或 RDS for MySQL，請選擇 MariaDB and MySQL (MariaDB 和 MySQL)。對於 RDS for PostgreSQL，選擇 PostgreSQL。對於 RDS for SQL Server，請選擇 SQL Server。
- 代理識別碼。在您的 AWS 帳戶 ID 和當前 AWS 區域中指定一個唯一的名稱。
- 閒置用戶端連線逾時。選擇代理伺服器關閉用戶端連線之前可閒置的時段。預設值為 1,800 秒 (30 分鐘)。上一個要求完成後若應用程式未在的指定時間內提交新要求，用戶端連線就會被視為閒置。基礎資料庫連線保持開啟狀態，並傳回至連線集區。因此，它可供新的用戶端連線使用。

若要讓 Proxy 主動移除過時的連線，請降低閒置的用戶端連線逾時。當工作負載激增時，為了節省建立連接的成本，請增加閒置的客戶端連接超時。」

針對目標群組組態，提供下列項目的資訊：

- 資料庫。選擇一個 RDS 資料庫執行個體 以透過此代理存取。此清單僅包含具有相容資料庫引擎、引擎版本和其他設定的資料庫執行個體和叢集。如果清單是空的，請建立與 RDS Proxy 相容的新資料庫執行個體或叢集。若要執行此作業，請依照 [建立 Amazon RDS 資料庫執行個體](#) 中的程序進行。然後嘗試再次建立代理。
- 連線集區最大連線數。指定介於 1 到 100 之間的值。此設定代表 RDS Proxy 可用於其連線的 max_connections 值的百分比。如果您只打算在此資料庫執行個體或叢集中使用一個代理，可將此值設定為 100。如需 RDS Proxy 如何使用此設定的詳細資訊，請參閱 [MaxConnections 百分比](#)。

- 工作階段鎖定篩選條件。(選擇性) 此選項可讓您強制 RDS Proxy 不釘選偵測到的特定類型工作階段狀態。這樣做會規避用戶端連線之間多工處理資料庫連線的預設安全措施。目前，PostgreSQL 不支援此設定。唯一的選擇是 EXCLUDE_VARIABLE_SETS。

啟用此設定可能會導致某個連線的工作階段變數影響其他連線。如果您的查詢取決於目前交易之外設定的工作階段變數值，這樣做可能會造成錯誤或正確性問題。請先確認應用程式可安全地在用戶端連線之間共用資料庫連線，再考慮使用此選項。

以下模式可視為安全：

- SET 陳述式，其中有效的工作階段變數值沒有變更，也就是工作階段變數沒有變更。
- 您會在同一交易中變更工作階段變數值和執行陳述式。

如需詳細資訊，請參閱 [避免鎖定](#)。

- 連線借用逾時。在某些情況下，您可能預期代理有時會使用所有可用的資料庫連線。在這種情況下，您可以指定代理在傳回逾時錯誤之前等待資料庫連線變成可用的時間。您可以指定的期間上限為 5 分鐘。只有在代理開啟的連線數已達上限，而且所有連線都已在使用中時，才會套用此設定。
- 初始化查詢。(選用) 開啟每個新資料庫連線時，您可為要執行的代理指定一或多個 SQL 陳述式。此設定通常與 SET 陳述式搭配使用，以確定每個連線都有相同的設定，例如時區和字元集。對於多個陳述式，請使用分號作為分隔符號。您也可以將單一 SET 陳述式中包含多個變數，例如 SET x=1, y=2。

針對 Authentication (身分驗證)，請提供下列項目的資訊：

- IAM 角色。選擇具有存取您先前所選擇 Secrets Manager 私密之許可的 IAM 角色。或者，您可以從建立新的 IAM 角色 AWS Management Console。
- Secrets Manager 秘密。選擇至少一個包含資料庫使用者認證的 Secret Secrets Manager 碼，這些密碼可讓代理伺服器存取 RDS 資料庫執行個體。
- Client authentication type (用戶端身分驗證類型)。選擇代理從用戶端進行連線所使用的身分驗證類型。您的選擇適用於您將其與此代理建立關聯的所有 Secrets Manager 機密。如果您需要為每個密碼指定不同的用戶端驗證類型，請改用 AWS CLI 或 API 來建立 Proxy。
- IAM authentication (IAM 身分驗證)。選擇是否要求、允許或不允許連線到代理的連線進行 IAM 身分驗證。允許選項僅適用於 RDS for SQL Server 的代理。您的選擇適用於您將其與此代理建立關聯的所有 Secrets Manager 機密。如果您需要為每個密碼指定不同的 IAM 身份驗證，請改用 AWS CLI 或 API 來建立 Proxy。

針對連線，提供下列項目的資訊：

- 需要 Transport Layer Security。如果您希望代理對所有用戶端連線強制執行 TLS/SSL，請選擇此設定。對於代理的加密或未加密連線，代理會在連線到基礎資料庫時使用相同的加密設定。
- 子網路。此欄位會預先填入與 VPC 關聯的所有子網路。您可以移除此代理不需要的任何子網路。您至少必須保留兩個子網路。

提供其他連線組態：

- VPC 安全群組。選擇現有的 VPC 安全群組。或者，您可以從建立新的安全性群組 AWS Management Console。您必須設定輸入規則，以允許應用程式存取 Proxy。您還必須設定輸出規則，以允許來自資料庫目標的流量。

Note

此安全群組必須允許從代理至目標資料庫的連線。相同的安全群組會用於從應用程式到代理的輸入，以及從代理到資料庫的輸出。例如，假設您將同一個安全群組用於您的資料庫和代理。在這種情況下，請務必指定安全群組中的資源能與同一個安全群組中的其他資源進行通訊。

使用共用 VPC 時，您無法使用 VPC 的預設安全群組，或屬於另一個帳戶的預設安全群組。選擇屬於您帳戶的安全群組。如果沒有，請先建立一個。如需此限制的詳細資訊，請參閱[使用共用的 VPC](#)。

RDS 會在多個可用區域上部署代理，以確保高可用性。若要為此類代理啟用跨可用區域通訊，代理子網路的網路存取控制清單 (ACL) 必須允許引擎連接埠專屬的輸出，以及允許所有連接埠輸入。如需網路 ACL 的詳細資訊，請參閱[使用網路 ACL 控制到子網路的流量](#)。如果您的代理和目標的網路 ACL 相同，您必須新增 TCP 通訊協定輸入規則，並將其中的來源設定為 VPC CIDR。您也必須新增引擎連接埠特定 TCP 通訊協定輸出規則，其中目的地設定為 VPC CIDR。

(選用) 提供進階組態：

- 啟用增強型日誌。您可以啟用此設定，以疑難排解代理相容性或效能問題。

啟用此設定時，RDS Proxy 會在其記錄檔中包含有關代理伺服器效能的詳細資訊。此資訊可協助您偵錯涉及 SQL 行為或代理連線的效能和可擴展性的問題。因此，請僅啟用此設定以進行偵錯，以及當您有安全性措施時，才能保護記錄檔中出現的任何敏感資訊。

為最小化與代理相關聯的負荷，RDS Proxy 會自動在您啟用此設定的 24 小時後將其關閉。您可以暫時啟用此設定以針對特定問題進行故障診斷。

5. 選擇 Create Proxy (建立代理)。

AWS CLI

若要使用建立代理 AWS CLI，請使用下列必要參數呼叫 [create-db-proxy](#) 命令：

- `--db-proxy-name`
- `--engine-family`
- `--role-arn`
- `--auth`
- `--vpc-subnet-ids`

`--engine-family` 值會區分大小寫。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-proxy \  
  --db-proxy-name proxy_name \  
  --engine-family { MYSQL | POSTGRESQL | SQLSERVER } \  
  --auth ProxyAuthenticationConfig_JSON_string \  
  --role-arn iam_role \  
  --vpc-subnet-ids space_separated_list \  
  [--vpc-security-group-ids space_separated_list] \  
  [--require-tls | --no-require-tls] \  
  [--idle-client-timeout value] \  
  [--debug-logging | --no-debug-logging] \  
  [--tags comma_separated_list]
```

在 Windows 中：

```
aws rds create-db-proxy ^
  --db-proxy-name proxy_name ^
  --engine-family { MYSQL | POSTGRESQL | SQLSERVER } ^
  --auth ProxyAuthenticationConfig_JSON_string ^
  --role-arn iam_role ^
  --vpc-subnet-ids space_separated_list ^
  [--vpc-security-group-ids space_separated_list] ^
  [--require-tls | --no-require-tls] ^
  [--idle-client-timeout value] ^
  [--debug-logging | --no-debug-logging] ^
  [--tags comma_separated_list]
```

下列範例是 --auth 選項的 JSON 值。此範例將不同的用戶端驗證類型套用至每個密碼。

```
[
  {
    "Description": "proxy description 1",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret/1234abcd-12ab-34cd-56ef-1234567890ab",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_SCRAM_SHA_256"
  },
  {
    "Description": "proxy description 2",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:seret/1234abcd-12ab-34cd-56ef-1234567890cd",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_MD5"
  },
  {
    "Description": "proxy description 3",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122221111:secret/1234abcd-12ab-34cd-56ef-1234567890ef",
    "IAMAuth": "REQUIRED"
  }
]
```

]

i Tip

如果您還不知道要用於 `--vpc-subnet-ids` 參數的子網路 ID，請參閱 [設定網路先決條件](#) 以取得如何尋找它們的範例。

i Note

安全群組必須允許存取代理連線的目標資料庫。相同的安全群組會用於從應用程式到代理的輸入，以及從代理到資料庫的輸出。例如，假設您將同一個安全群組用於您的資料庫和代理。在這種情況下，請務必指定安全群組中的資源能與同一個安全群組中的其他資源進行通訊。使用共用 VPC 時，您無法使用 VPC 的預設安全群組，或屬於另一個帳戶的預設安全群組。選擇屬於您帳戶的安全群組。如果沒有，請先建立一個。如需此限制的詳細資訊，請參閱 [使用共用的 VPC](#)。

若要為代理伺服器建立正確的關聯，您也可以使用暫存器 `db-proxy-` 目標指令。指定目標群組類型 `default`。RDS Proxy 會在您建立每個代理時，自動建立的目標群組。

```
aws rds register-db-proxy-targets
  --db-proxy-name value
  [--target-group-name target_group_name]
  [--db-instance-identifiers space_separated_list] # rds db instances, or
  [--db-cluster-identifiers cluster_id]           # rds db cluster (all instances)
```

RDS API

若要建立 RDS 代理，請呼叫 Amazon RDS API 操作 [CreateDBProxy](#)。您傳遞具有 [AuthConfig](#) 資料結構的參數。

RDS Proxy 在您建立每個代理時，自動建立名為 `default` 的目標群組。您可以呼叫函數 [註冊 ProxyTargets](#) 資料庫資料庫，將 RDS 資料庫執行個體與目標群組建立關聯。

檢視 RDS Proxy

在建立一個或多個 RDS 代理後，您可以檢視它們全部。這樣做可讓您檢查其組態詳細資訊，並選擇那些內容需要修改、刪除等。

若要讓資料庫應用程式使用 Proxy，您必須在連接字串中提供 Proxy 端點。

AWS Management Console

如何檢視代理

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在的右上角 AWS Management Console，選擇您在其中建立 RDS 代理的 AWS 區域。
3. 在導覽窗格中，選擇 Proxies (代理)。
4. 選擇 RDS 代理名稱，以顯示其詳細資訊。
5. 在詳細資料頁面上，[目標群組] 區段會顯示代理伺服器與特定 RDS 資料庫執行個體 Aurora 資料庫的關聯方式。您可以依照預設目標群組頁面的連結，查看代理與資料庫之間關聯的詳細資訊。此頁面可讓您查看您在建立代理時所指定的設定。其中包括最大連線百分比、連線借用逾時、引擎系列，以及工作階段鎖定篩選條件。

CLI

若要使用 CLI 檢視代理，請使用 [describe-db-proxies](#) 命令。默認情況下，它顯示您的 AWS 帳戶擁有的所有代理。若要查看單一代理的詳細資料，請使用 `--db-proxy-name` 參數指定其名稱。

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

若要檢視與代理相關聯的其他資訊，請使用下列命令。

```
aws rds describe-db-proxy-target-groups --db-proxy-name proxy_name
```

```
aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

使用下列命令序列查看與代理關聯的事物的詳細資訊：

1. 若要取得代理清單，請執行 [describe-db-proxies](#)。
2. 若要顯示連線參數 (例如代理可以使用的連線百分比上限)，請執行 [describe-db-proxy-target-groups](#) `--db-proxy-name`。使用代理的名稱做為參數值。
3. 若要查看與傳回的目標群組相關聯的 RDS 資料庫執行個體 的詳細資訊，請執行 [描述](#)-db-Proxy 目標。

RDS API

若要使用 RDS API 檢視代理，請使用 [DescribeDBProxies](#) 操作。它會傳回 [DBProxy](#) 資料類型的值。

若要查看 Proxy 連線設定的詳細資訊，請將此傳回值中的代理主機識別碼與 [描述 BedB ProxyTarget 群組](#) 作業搭配使用。它返回 [DB ProxyTarget 組](#) 數據類型的值。

若要查看與代理相關聯的 RDS 執行個體或 Aurora 資料庫叢集，請使用 [描述 B 作ProxyTargets](#) 業。它返回 [DB ProxyTarget](#) 數據類型的值。

透過 RDS Proxy 連線至資料庫

透過 Proxy 或連線至資料庫連線至 RDS 資料庫執行個體的方式通常相同。如需詳細資訊，請參閱 [代理端點概觀](#)。

主題

- [使用原生身分驗證連線到代理](#)
- [使用 IAM 身分驗證連線到代理](#)
- [使用 Microsoft SQL Server 連線至代理的考量](#)
- [使用 PostgreSQL 連線至代理的考量事項](#)

使用原生身分驗證連線到代理

使用下列步驟來使用原生驗證連線到 Proxy：

1. 尋找代理端點。在中 AWS Management Console，您可以在對應 Proxy 的詳細資料頁面上找到端點。透過 AWS CLI，您可以使用 [描述-db-](#) 代理伺服器指令。下列範例會顯示作法。

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*]'.
{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
  [
    {
      "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
      "DBProxyName": "the-proxy"
    },
    {
      "Endpoint": "the-proxy-other-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
```

```
        "DBProxyName": "the-proxy-other-secret"
    },
    {
        "Endpoint": "the-proxy-rds-secret.proxy-demo.us-east-1.rds.amazonaws.com",
        "DBProxyName": "the-proxy-rds-secret"
    },
    {
        "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
        "DBProxyName": "the-proxy-t3"
    }
]
]
```

2. 在用戶端應用程式的連接字串中，將端點指定為 `host` 參數。例如，指定代理端點做為 `mysql -h` 選項或 `psql -h` 選項的值。
3. 請提供和平常一樣的資料庫使用者名稱和密碼。

使用 IAM 身分驗證連線到代理

當您將 IAM 驗證與 RDS Proxy 搭配使用時，請設定您的資料庫使用者，以使用一般使用者名稱和密碼進行驗證。IAM 驗證可讓 RDS Proxy 從 Secrets Manager 擷取使用者名稱和密碼登入資料。從 RDS Proxy 連至底層資料庫的連線不會透過 IAM。

若要使用 IAM 身分驗證連線至 RDS Proxy，請使用與透過 RDS 資料庫執行個體 Aurora 資料庫進行 IAM 驗證相同的一般連線程序。如需使用 IAM 的一般資訊，請參閱 [Amazon RDS 中的安全](#)。

RDS Proxy 的 IAM 使用方式的主要差異包含下列項目：

- 您不會使用身分驗證外掛程式設定每個個別資料庫使用者。資料庫使用者在資料庫內仍有一般使用者名稱和密碼。您設定包含這些使用者名稱和密碼的 Secrets Manager 私密，然後授權 RDS Proxy 擷取來自 Secrets Manager 的登入資料。

IAM 身分驗證會套用到您用戶端程式與代理之間的連線。接著，代理會使用從 Secrets Manager 中擷取的使用者名稱和密碼登入資料，向資料庫進行身分驗證。

- 您會指定代理端點，而非執行個體、叢集或讀取器端點。如需代理端點的詳細資訊，請參閱 [使用 IAM 身分驗證連接至資料庫執行個體](#)。
- 在直接資料庫 IAM 身分驗證案例中，您會選擇性地選擇資料庫使用者，並設定他們以便透過特殊身分驗證外掛程式進行識別。接著，您可以使用 IAM 身分驗證連線至這些使用者。

在 Proxy 使用案例中，您需要提供具有秘密的 Proxy，其包含某些使用者的使用者名稱和密碼 (原生身分驗證)。然後，您可以使用 IAM 身分驗證連線到 Proxy。在這裡，您可以藉由使用 Proxy 端點 (不是資料庫端點) 產生身分驗證字符來完成此操作。您也可以針對您提供的秘密，使用符合其中一個使用者名稱的使用者名稱。

- 使用 IAM 身分驗證連接至 Proxy 時，請務必使用 Transport Layer Security (TLS)/Secure Sockets Layer (SSL)。

您可以透過修改 IAM 政策，授與特定使用者對代理的存取權。範例如下。

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKL01234/db_user"
```

使用 Microsoft SQL Server 連線至代理的考量

如需使用 IAM 身分驗證連線到代理，請不要使用密碼欄位。相反地，您可以在權杖欄位中為每種類型的資料庫驅動程式提供適當的權杖屬性。例如，針對 JDBC 使用 `accessToken` 屬性，或針對 ODBC 使用 `sql_copt_ss_access_token` 屬性。或者使用 .NET SqlClient 驅動程式的 `AccessToken` 屬性。您無法使用 IAM 身分驗證，搭配不支援權杖屬性的用戶端。

在某些情況下，代理無法共用資料庫連線，而是將用戶端應用程式到代理的連線鎖定為專用的資料庫連線。如需這些條件的詳細資訊，請參閱 [避免鎖定](#)。

使用 PostgreSQL 連線至代理的考量事項

針對 PostgreSQL，當用戶端開始 PostgreSQL 資料庫的連線時，它會傳送啟動訊息。此訊息包含包含參數名稱和值字串的組合。如需詳細資訊，請參閱 PostgreSQL 文件的 [PostgreSQL 訊息格式](#) 中的 `StartupMessage`。

透過 RDS 代理連線時，啟動訊息可包含下列目前辨識的參數：

- `user`
- `database`

啟動訊息也可包含下列其他執行時間參數：

- [application_name](#)
- [client_encoding](#)

- [DateStyle](#)
- [TimeZone](#)
- [extra_float_digits](#)
- [search_path](#)

如需 PostgreSQL 傳訊的詳細資訊，請參閱 PostgreSQL 文件中的[前端/後端通訊協定](#)。

對於 PostgreSQL，如果您使用 JDBC，我們建議您執行以下操作以避免鎖定：

- 將 JDBC 連線參數 `assumeMinServerVersion` 至少設為 9.0 以避免鎖定。這樣可防止 JDBC 驅動程式在執行連線啟動期間執行 `SET extra_float_digits = 3` 行額外的往返。
- 將 JDBC 連線參數 `ApplicationName` 設為 *any/your-application-name* 以避免鎖定。這麼做可防止在執行 `SET application_name = "PostgreSQL JDBC Driver"` 時，JDBC 驅動程式在連線啟動期間執行額外的往返。請注意，JDBC 參數為 `ApplicationName`，但 PostgreSQL `StartupMessage` 參數為 `application_name`。

如需詳細資訊，請參閱[避免鎖定](#)。如需如何使用 JDBC 連線的詳細資訊，請參閱 PostgreSQL 文件中的[連線至資料庫](#)。

管理 RDS Proxy

本節提供如何管理 RDS 代理伺服器作業和組態的相關資訊。這些程序可協助應用程式以最有效率的方式使用資料庫連線，並達到最大程度的連線重複使用。您越能利用連線重複使用，就越能節省更多 CPU 和記憶體負荷。這樣可以縮短應用程式的延遲，並讓資料庫投入更多資源來處理應用程式要求。

主題

- [修改 RDS Proxy](#)
- [新增新的資料庫使用者](#)
- [變更資料庫使用者的密碼](#)
- [用戶端與資料庫連線](#)
- [配置連線設定](#)
- [避免鎖定](#)
- [刪除 RDS Proxy](#)

修改 RDS Proxy

您可以在建立代理之後變更與該代理相關聯的特定設定。您可以修改代理本身、它的關聯目標群組，或兩者都修改。每個代理都有一個關聯的目標群組。

AWS Management Console

Important

Client authentication type (用戶端身分驗證類型) 和 IAM authentication (IAM 身分驗證) 欄位中的值適用於與此代理相關聯的所有 Secrets Manager 機密。若要為每個密碼指定不同的值，請改用 AWS CLI 或 API 來修改 Proxy。

若要修改代理的設定

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 在代理清單中，選擇您要修改其設定的代理伺服器，或移至其詳細資訊頁面。
4. 在 Actions (動作) 中，選擇 Modify (修改)。
5. 輸入或選擇要修改的屬性。您可以修改下列選項：
 - Proxy identifier (Proxy 識別碼) – 輸入新的識別碼，以便將 Proxy 重新命名。
 - Idle client connection timeout (閒置用戶端連線逾時) – 輸入閒置用戶端連線逾時的期間。
 - IAM role (IAM 角色) – 變更從 Secrets Manager 中擷取秘密所使用的 IAM 角色。
 - Secrets Manager secrets (Secrets Manager 秘密) – 新增或移除 Secrets Manager 秘密。這些秘密對應於資料庫使用者名稱和密碼。
 - Client authentication type (用戶端身分驗證類型) – (僅限 PostgreSQL) 變用戶端連線至代理時的身分驗證類型。
 - IAM authentication (IAM 身分驗證) – 要求或不允許連線至 Proxy 時進行 IAM 身分驗證。
 - Require Transport Layer Security – 開啟或關閉 Transport Layer Security (TLS) 的需求。
 - VPC security group (VPC 安全群組) – 新增或移除 Proxy 要使用的 VPC 安全群組。
 - Enable enhanced logging (啟用增強型日誌) – 啟用或停用增強型日誌。
6. 選擇 Modify (修改)。

如果您找不到您要變更的列出設定，請使用下列程序更新代理的目標群組。與代理相關聯的目標群組可控制與實體資料庫連線相關的設定。每個代理都有一個名為 default 的關聯目標群組，這些群組將與代理一起自動建立。

您只能從代理詳細資訊頁面修改目標群組，而不能從代理頁面上的清單修改。

若要修改代理目標群組的設定

1. 在 Proxies (代理) 頁面上，前往代理的詳細資訊頁面。
2. 在目標群組中選擇 default 連結。目前，所有的代理都有一個名為 default 的目標群組。
3. 在預設目標群組的詳細資訊頁面上，選擇 Modify (修改)。
4. 選擇您可修改之屬性的新設定：
 - 資料庫 – 選擇不同的 RDS 資料庫執行個體或叢集。
 - Connection pool maximum connections (連線集區最大連線數) – 調整 Proxy 可以使用的最大可用連線數的百分比。
 - Session pinning filters (工作階段鎖定篩選條件) – (選用) 選擇工作階段鎖定篩選條件。這樣做會規避用戶端連線之間多工處理資料庫連線的預設安全措施。目前，PostgreSQL 不支援此設定。唯一的選擇是 EXCLUDE_VARIABLE_SETS。

啟用此設定可能會導致某個連線的工作階段變數影響其他連線。如果您的查詢取決於目前交易之外設定的工作階段變數值，這樣做可能會造成錯誤或正確性問題。請先確認應用程式可安全地在用戶端連線之間共用資料庫連線，再考慮使用此選項。

以下模式可視為安全：

- SET 陳述式，其中有效的工作階段變數值沒有變更，也就是工作階段變數沒有變更。
- 您會在同一交易中變更工作階段變數值和執行陳述式。

如需詳細資訊，請參閱 [避免鎖定](#)。

- Connection borrow timeout (連線借用逾時) – 調整連線借用逾時間隔。在已為代理使用連線數上限時，則會套用此設定。此設定會決定代理在傳回逾時錯誤之前等待連線變成可用的時間。
- Initialization query (初始化查詢) – (選用) 新增初始化查詢，或修改目前的查詢。開啟每個新資料庫連線時，您可為要執行的代理指定一或多個 SQL 陳述式。該設定通常與 SET 陳述式搭配使用，確保每個連線都有相同的設定，例如時區和字元集。對於多個陳述式，請使用分號作為分隔符號。您也可以將單一 SET 陳述式中包含多個變數，例如 SET x=1, y=2。

您無法變更某些屬性，例如目標群組識別符和資料庫引擎。

5. 選擇 Modify target group (修改目標群組)。

AWS CLI

[若要使用修改代理伺服器，請使用指令修改-DB 代理伺服器 AWS CLI、修改-DB 代理-目標群組、取消註冊-DB 代理-目標，以及註冊-DB 代理-目標。](#)

使用 `modify-db-proxy` 命令，您可以變更屬性，如下所示：

- 代理使用的 Secrets Manager 秘密組。
- 是否需要 TLS。
- 閒置用戶端逾時。
- 是否記錄來自 SQL 陳述式的額外資訊以供偵錯。
- 用於擷取 Secrets Manager 秘密的 IAM 角色。
- 代理使用的安全群組。

下列範例顯示如何將現有的代理重新命名。

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

若要修改連線相關設定或重新命名目標群組，請使用 `modify-db-proxy-target-group` 命令。目前，所有的代理都有一個名為 `default` 的目標群組。使用此目標群組時，您指定代理的名稱，並將目標群組的名稱指定為 `default`。

下列範例顯示如何先檢查代理的 `MaxIdleConnectionsPercent` 設定，然後使用目標群組將其變更。

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy

{
  "TargetGroups": [
    {
      "Status": "available",
      "UpdatedDate": "2019-11-30T16:49:30.342Z",
      "ConnectionPoolConfig": {
        "MaxIdleConnectionsPercent": 50,
        "ConnectionBorrowTimeout": 120,
        "MaxConnectionsPercent": 100,
        "SessionPinningFilters": []
      }
    }
  ]
}
```



```

    },
    "TargetGroupName": "default",
    "CreateDate": "2019-11-30T16:49:27.940Z",
    "DBProxyName": "the-proxy",
    "IsDefault": true
  }
]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name
default --connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
  "DBProxyTargetGroup": {
    "Status": "available",
    "UpdatedDate": "2019-12-02T04:09:50.420Z",
    "ConnectionPoolConfig": {
      "MaxIdleConnectionsPercent": 75,
      "ConnectionBorrowTimeout": 120,
      "MaxConnectionsPercent": 100,
      "SessionPinningFilters": []
    },
    "TargetGroupName": "default",
    "CreateDate": "2019-11-30T16:49:27.940Z",
    "DBProxyName": "the-proxy",
    "IsDefault": true
  }
}

```

使用 `deregister-db-proxy-targets` 和 `register-db-proxy-targets` 命令，您可以透過其目標群組變更 Proxy 與哪一個 RDS 資料庫執行個體相關聯。目前，每個代理主機都可以連線到一個 RDS 資料庫執行個體 目標群組追蹤異地同步備份組態中所有 RDS 資料庫執行個體的連線詳細資料，體。

下列範例將從與名為 `cluster-56-2020-02-25-1399` 的 Aurora MySQL 叢集關聯的代理開始。此範例顯示如何變更代理，讓它可以連線到名為 `provisioned-cluster` 的不同叢集。

當您使用 RDS 資料庫執行個體時，請指定 `--db-instance-identifier` 選項。

下列範例會修改 Aurora MySQL 代理。Aurora PostgreSQL 代理具有連接埠 5432。

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy
```

```
{
  "Targets": [
    {
      "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "instance-9814"
    },
    {
      "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "instance-8898"
    },
    {
      "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "instance-1018"
    },
    {
      "Type": "TRACKED_CLUSTER",
      "Port": 0,
      "RdsResourceId": "cluster-56-2020-02-25-1399"
    },
    {
      "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "instance-4330"
    }
  ]
}
```

```
aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
cluster-56-2020-02-25-1399
```

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy
```

```
{
  "Targets": []
}
```

```
aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
provisioned-cluster

{
  "DBProxyTargets": [
    {
      "Type": "TRACKED_CLUSTER",
      "Port": 0,
      "RdsResourceId": "provisioned-cluster"
    },
    {
      "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "gkldje"
    },
    {
      "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
      "Type": "RDS_INSTANCE",
      "Port": 3306,
      "RdsResourceId": "provisioned-1"
    }
  ]
}
```

RDS API

[若要修改使用 RDS API 的代理，您可以使用作業 ModifyDBProxy、修改資料庫ProxyTarget群組、取消ProxyTargets註冊資料庫和註冊資料庫作業。ProxyTargets](#)

使用 ModifyDBProxy，您可以變更屬性，如下所示：

- 代理使用的 Secrets Manager 秘密組。
- 是否需要 TLS。
- 閒置用戶端逾時。
- 是否記錄來自 SQL 陳述式的額外資訊以供偵錯。
- 用於擷取 Secrets Manager 秘密的 IAM 角色。
- 代理使用的安全群組。

您可以使用 `ModifyDBProxyTargetGroup` 修改連線相關設定或重新命名目標群組。目前，所有的代理都有一個名為 `default` 的目標群組。使用此目標群組時，您指定代理的名稱，並將目標群組的名稱指定為 `default`。

使用 `DeregisterDBProxyTargets` 和 `RegisterDBProxyTargets`，您可以透過其目標群組變更代理與哪個 RDS 資料庫執行個體相關聯。目前，每個 Proxy 都可以連線到一個 RDS 資料庫執行個體。目標群組會追蹤多可用區域組態中 RDS 資料庫執行個體的連線詳細資訊。

新增新的資料庫使用者

在某些情況下，您可能會將新的資料庫使用者新增至與 Proxy 相關聯的 RDS 資料庫執行個體或叢集。如果是這樣，請新增或重新規劃 Secrets Manager 秘密來儲存該使用者的登入資料。若要這麼做，選擇以下選項其中之一：

1. 使用 [設定資料庫認證 AWS Secrets Manager](#) 中描述的程序建立新的 Secrets Manager 秘密。
2. 更新 IAM 角色，以授予 RDS Proxy 存取新 Secrets Manager 私密的權限。若要這麼做，請更新 IAM 角色政策的資源區段。
3. 修改 RDS Proxy，在 Secrets Manager 密碼下新增新的 Secrets Manager 密碼。
4. 如果新使用者取代了現有使用者，請為現有使用者更新儲存於代理的 Secrets Manager 秘密中的登入資料。

將新的資料庫使用者新增至 PostgreSQL 資料庫

將新使用者新增至 PostgreSQL 資料庫時，如果您已執行下列命令：

```
REVOKE CONNECT ON DATABASE postgres FROM PUBLIC;
```

授予 `rdsproxyadmin` 使用者 `CONNECT` 權限，讓使用者可以監控目標資料庫上的連線。

```
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

您也可以透過在上述命令中將 `rdsproxyadmin` 變更為資料庫使用者，來允許其他目標資料庫使用者執行運作狀態檢查。

變更資料庫使用者的密碼

在某些情況下，您可能會在與 Proxy 相關聯的 RDS 資料庫執行個體中變更資料庫使用者的密碼。如果是這樣，請使用新密碼更新對應的 Secrets Manager 秘密。

用戶端與資料庫連線

從應用程式到 RDS Proxy 的連線稱為用戶端連線。從 Proxy 到資料庫的連線為資料庫連線。使用 RDS Proxy 時，用戶端連線會在 Proxy 終止，而資料庫連線則是在 RDS Proxy 內管理。

應用程式端連線集區可提供減少應用程式與 RDS Proxy 之間週期性連線建立的好處。

實作應用程式端連線集區之前，請考慮下列組態層面：

- **用戶端連線最長壽命**：RDS Proxy 會強制執行用戶端連線的最長壽命為 24 小時。此值不可設定。設定集區的最長連線壽命少於 24 小時，以避免意外的用戶端連線中斷。
- **用戶端連線閒置逾時**：RDS Proxy 會強制執行用戶端連線的閒置時間上限。設定閒置連線逾時值低於 RDS Proxy 用戶端連線閒置逾時設定的集區，以避免意外連線中斷。

在應用程式端連線集區中設定的用戶端連線數目上限不一定限制為 RDS Proxy 的 `max_Connect` 設定。

用戶端連線集區會延長用戶端連線壽命。如果您的連線遭遇綁定，則集區用戶端連線可能會降低多工效率。固定但閒置在應用程式端連線集區中的用戶端連線會繼續保留資料庫連線，並防止其他用戶端連線重複使用資料庫連線。檢閱您的 Proxy 記錄，以檢查您的連線是否經歷釘選。

Note

RDS Proxy 會在不再使用資料庫連線 24 小時後的某個時間，關閉資料庫連線。無論閒置連線數上限設定的值為何，Proxy 都會執行此動作。

配置連線設定

若要調整 RDS Proxy 的連線集區，可以修改以下設定：

- [IdleClient逾時](#)
- [MaxConnections百分比](#)
- [MaxIdleConnectionsPercent](#)
- [ConnectionBorrow逾時](#)

IdleClient逾時

您可以指定 Proxy 關閉用戶端連線之前的閒置時間長度。預設值為 1,800 秒 (30 分鐘)。

上一個要求完成後若應用程式未能在指定時間內提交新要求，用戶端連線就會被視為閒置。基礎資料庫連線保持開啟狀態，並傳回至連線集區。因此，它可供新的用戶端連線使用。如果您希望 Proxy 主動移除過時的連線，請降低閒置的用戶端連線逾時。如果您的工作負載與 Proxy 建立頻繁的連線，請提高閒置的用戶端連線逾時，以節省建立連線的成本。

RDS 主控台中的 [閒置用戶端連線逾時] 欄位以及和 API 中的 `IdleClientTimeout` 設定表示此設定。AWS CLI 若要了解如何在 RDS 主控台中變更 `Idle client connection timeout` (閒置用戶端連線逾時) 欄位的值，請參閱 [AWS Management Console](#)。若要了解如何變更 `IdleClientTimeout` 設定的值，請參閱 CLI 命令 [modify-db-proxy](#) 或 API 操作 [ModifyDBProxy](#)。

MaxConnections百分比

您可以限制 RDS Proxy 可以與目標資料庫建立的連線數。需以資料庫的可用最大連線數百分比形式指定限制。此設定由 RDS 主控台中的 [連線集區最大連線數] 欄位以及和 API 中 AWS CLI 的 `MaxConnectionsPercent` 設定表示。

`MaxConnectionsPercent` 值會以目標群組所使用的 RDS 資料庫執行個體的 `max_connections` 設定百分比來表示。Proxy 不會事先保留這些連線。此設定可讓 Proxy 在工作負載需要時建立這些連線。

例如，對於 `max_connections` 設定為 1000 且 `MaxConnectionsPercent` 設為 95 的已註冊資料庫目標，RDS Proxy 會將 950 個連線設定為同時連線至該資料庫目標的上限。

工作負載達到允許的資料庫連線數目上限的常見副作用，是會增加整體查詢延遲，以及提高 `DatabaseConnectionsBorrowLatency` 指標。您可以透過比較 `DatabaseConnections` 和 `MaxDatabaseConnectionsAllowed` 指標，來監控目前使用的資料庫連線和允許的總數。

設定此參數時，請注意下列最佳實務：

- 為工作負載模式的變更提供足夠的連線預留空間。建議將參數設定至少比您最近監控的最大使用量高出 30%。由於 RDS Proxy 會在多個節點之間重新分配資料庫連線配額，因此內部容量變更可能需要至少 30% 的成長空間才能進行其他連線，以避免增加借用延遲。
- RDS Proxy 會保留特定數量的連線以進行主動監控，以支援快速容錯移轉、流量路由和內部操作。`MaxDatabaseConnectionsAllowed` 指標不包括這些保留的連線。它代表可用於服務工作負載的連線數，並且可以低於從 `MaxConnectionsPercent` 設定衍生的值。

最小建議 `MaxConnectionsPercent` 值

- db.t3.small : 30
- db.t3.medium 或以上 : 20

若要了解如何在 RDS 主控台中變更 Connection pool maximum connections (連線集區最大連線數) 欄位的值，請參閱 [AWS Management Console](#)。若要瞭解如何變更MaxConnectionsPercent設定值，請參閱 [CLI 命令修改-db-代理目標群組或 API 作業修改資料庫群組。ProxyTarget](#)

如需資料庫連線數限制的詳細資訊，請參閱[資料庫連線數上限](#)。

MaxIdleConnectionsPercent

您可以控制 RDS Proxy 可在連線集區中保留的閒置資料庫連線數。根據預設，當連線上沒有任何活動五分鐘時，RDS Proxy 會將其集區中的資料庫連線視為閒置。

此MaxIdleConnectionsPercent值以 RDS 資料庫執行個體目標群組的max_connections設定百分比表示。預設值為 MaxConnectionsPercent 的 50%，上限為 MaxConnectionsPercent 的值。例如MaxConnectionsPercent，如果是 80，則預設值MaxIdleConnectionsPercent為 40。如果MaxConnectionsPercent未指定的值，則對於 SQL Server 的 RDS，則MaxIdleConnectionsPercent為 5，而對於所有其他引擎，預設值為 50。

值大時，Proxy 會將高百分比的閒置資料庫連線維持在開啟狀態。值小時，Proxy 就會關閉高百分比的閒置資料庫連線。如果您的工作負載無法預測，請考慮為 MaxIdleConnectionsPercent 設定一個高值。這麼做表示 RDS Proxy 可以因應活動中的突增情況，而無需開啟大量新的資料庫連線。

此設定由 AWS CLI 和 API DBProxyTargetGroup 中的MaxIdleConnectionsPercent設定表示。若要瞭解如何變更MaxIdleConnectionsPercent設定值，請參閱 [CLI 命令修改-db-代理目標群組或 API 作業修改資料庫群組。ProxyTarget](#)

如需資料庫連線數限制的詳細資訊，請參閱[資料庫連線數上限](#)。

ConnectionBorrow逾時

您可以選擇 RDS Proxy 在傳回逾時錯誤之前，等待連線集區中的資料庫連線變成可用的時間。預設值為 120 秒。此設定適用於連線數達到最大值，因此連線集區中沒有可用的連線時。當沒有適當的資料庫執行個體可用來處理要求時，例如當容錯移轉作業正在處理中時，它也適用。使用此設定，您可以設定應用程式的最佳等待期間，而不需要變更應用程式程式碼中的查詢逾時。

此設定由 RDS 主控台連線借用逾時欄位或 AWS CLI 或 API DBProxyTargetGroup 中的ConnectionBorrowTimeout設定表示。若要了解如何在 RDS 主控台中變更 Connection

borrow timeout (連線借用逾時) 欄位的值，請參閱 [AWS Management Console](#)。若要瞭解如何變更 `ConnectionBorrowTimeout` 設定值，請參閱 [CLI 命令修改-db-代理目標群組或 API 作業修改資料庫群組](#)。ProxyTarget

避免鎖定

當資料庫請求不依賴先前請求的狀態資訊時，多工會更有效率。在這種情況下，RDS Proxy 可以在每一筆交易結束時重複使用連線。這類狀態資訊的範例包括您可以透過 SET 或 SELECT 陳述式變更的大部分變數和組態參數。根據預設，用戶端連線上的 SQL 交易可以在基礎資料庫連線之間進行多工。

您與代理的連線可能會進入一種稱為鎖定的狀態。連線被鎖定時，每一筆後續交易都會使用相同的基礎資料庫連線，直到工作階段結束為止。其他用戶端連線也無法重複使用該資料庫連線，直到工作階段結束為止。工作階段會在用戶端連線中斷時結束。

在偵測到不適用於其他工作階段的工作階段狀態變更時，RDS Proxy 會自動將用戶端連線鎖定至特定資料庫連線。鎖定會降低連線重複使用的有效性。如果所有或幾乎所有的連線都遭遇鎖定，請考慮修改應用程式的程式碼或工作負載，以減少導致鎖定的情況。

例如，您的應用程式會變更工作階段變數或組態參數。在此情況下，稍後的陳述式可能會依賴新的變數或參數才會生效。因此，當 RDS Proxy 處理請求，而變更工作階段變數或組態設定時，會將該工作階段到資料庫的連線鎖定。這樣一來，工作階段狀態對於同一工作階段中的所有後續交易仍然有效。

對於部分資料庫引擎，此規則不適用於您可以設定的所有參數。RDS Proxy 會追蹤某些陳述式和變數。因此，當您修改它們時，RDS 代理不會釘選工作階段。在這種情況下，RDS Proxy 只會重複使用其他工作階段的連線，這些工作階段具有與這些設定相同的值。如需 RDS Proxy 針對資料庫引擎追蹤哪些項目的詳細資訊，請參閱下列內容：

- [RDS Proxy 針對 RDS for SQL Server 資料庫追蹤哪些項目](#)
- [RDS Proxy 針對 RDS for MariaDB 和 RDS for MySQL 資料庫追蹤哪些項目](#)

RDS Proxy 針對 RDS for SQL Server 資料庫追蹤哪些項目

下列是 RDS Proxy 追蹤的 SQL Server 陳述式：

- USE
- SET ANSI_NULLS
- SET ANSI_PADDING
- SET ANSI_WARNINGS

- SET ARITHABORT
- SET CONCAT_NULL_YIELDS_NULL
- SET CURSOR_CLOSE_ON_COMMIT
- SET DATEFIRST
- SET DATEFORMAT
- SET LANGUAGE
- SET LOCK_TIMEOUT
- SET NUMERIC_ROUNDABORT
- SET QUOTED_IDENTIFIER
- SET TEXTSIZE
- SET TRANSACTION ISOLATION LEVEL

RDS Proxy 針對 RDS for MariaDB 和 RDS for MySQL 資料庫追蹤哪些項目

以下是 RDS 代理跟踪 MariaDB 和 MySQL 語句：

- DROP DATABASE
- DROP SCHEMA
- USE

下列為 RDS Proxy 追蹤的 MySQL 和 MariaDB 變數：

- AUTOCOMMIT
- AUTO_INCREMENT_INCREMENT
- CHARACTER SET (or CHAR SET)
- CHARACTER_SET_CLIENT
- CHARACTER_SET_DATABASE
- CHARACTER_SET_FILESYSTEM
- CHARACTER_SET_CONNECTION
- CHARACTER_SET_RESULTS
- CHARACTER_SET_SERVER

- COLLATION_CONNECTION
- COLLATION_DATABASE
- COLLATION_SERVER
- INTERACTIVE_TIMEOUT
- NAMES
- NET_WRITE_TIMEOUT
- QUERY_CACHE_TYPE
- SESSION_TRACK_SCHEMA
- SQL_MODE
- TIME_ZONE
- TRANSACTION_ISOLATION (or TX_ISOLATION)
- TRANSACTION_READ_ONLY (or TX_READ_ONLY)
- WAIT_TIMEOUT

盡可能減少鎖定

RDS Proxy 的效能調校涉及試圖藉由將鎖定降到最低，讓交易層級的連線重複使用 (多工) 達到最大。

您可以執行下列操作，盡可能減少鎖定：

- 避免可能導致鎖定的不必要資料庫要求。
- 在所有連線中一致地設定變數和組態設定。這樣，稍後的工作階段更有可能重複使用具有這些特定設置的連線。

但是，對於 PostgreSQL，設定變數會導致工作階段鎖定。

- 若為 MySQL 引擎系列資料庫，將工作階段鎖定篩選條件套用至代理。如果您知道鎖定並不會影響應用程式的正確操作，就可以免除某些類型的操作將工作階段鎖定。
- 透過監控 Amazon CloudWatch 指標 DatabaseConnectionsCurrentlySessionPinned 查看釘選發生的頻率。如需此測量結果和其他 CloudWatch 量度的資訊，請參閱 [使用 Amazon 監控 RDS 代理指標 CloudWatch](#)。
- 如果您使用 SET 陳述式為每個用戶端連線執行相同的初始化，您可以執行此操作，同時保留交易層級的多工。在此情況下，您可以將設定初始工作階段狀態的陳述式移至代理所使用的初始化查詢中。此屬性為一包含一個或多個以分號分隔的 SQL 陳述式的字串。

例如，您可以為設定特定組態參數的代理定義初始化查詢。然後，每當它為該代理設定新連線時，RDS Proxy 會套用這些設定。您可以從應用程式的程式碼中移除相應的 SET 陳述式，這樣它們就不會干擾交易層級的多工。

如需代理發生鎖定頻率的指標，請參閱 [使用 Amazon 監控 RDS 代理指標 CloudWatch](#)。

導致所有引擎系列鎖定的條件

在下列情況下，多工可能會導致意外行為，代理將工作階段鎖定於目前連線：

- 文字大小大於 16 KB 的任何陳述式會導致代理鎖定工作階段。

導致 RDS for Microsoft SQL Server 鎖定的條件

若為 RDS for SQL Server，下列互動也會導致鎖定：

- 使用多個作用中結果集 (MARS)。如需 MARS 的詳細資訊，請參閱 [SQL Server](#) 文件。
- 使用分散式交易協調器 (DTC) 通訊。
- 建立暫存資料表、交易、游標或預備陳述式。
- 使用下列 SET 陳述式：
 - SET ANSI_DEFAULTS
 - SET ANSI_NULL_DFLT
 - SET ARITHIGNORE
 - SET DEADLOCK_PRIORITY
 - SET FIPS_FLAGGER
 - SET FMONLY
 - SET FORCEPLAN
 - SET IDENTITY_INSERT
 - SET NOCOUNT
 - SET NOEXEC
 - SET OFFSETS
 - SET PARSEONLY
 - SET QUERY_GOVENOR_COST_LIMIT

- SET REMOTE_PROC_TRANSACTIONS
- SET ROWCOUNT
- SET SHOWPLAN_ALL、SHOWPLAN_TEXT 與 SHOWPLAN_XML
- SET STATISTICS
- SET XACT_ABORT

導致 RDS for MariaDB 和 RDS for MySQL 鎖定的條件

對於 MariaDB 和 MySQL，以下互動也會導致固定：

- 明確的資料表鎖定陳述式 LOCK TABLE、LOCK TABLES 或 FLUSH TABLES WITH READ LOCK 會導致代理鎖定工作階段。
- 使用 GET_LOCK 建立具名鎖定會導致代理鎖定工作階段。
- 設定使用者變數或系統變數 (有某些例外) 會導致代理鎖定工作階段。如果這種情況減少了太多的連線重複使用率，請選擇不會造成釘選的 SET 作業。如需如何設定篩選屬性來執行此作業的資訊，請參閱 [建立 RDS Proxy](#) 和 [修改 RDS Proxy](#)。
- 建立一個暫時資料表會導致代理鎖定工作階段。這樣，無論交易邊界如何，暫時資料表的內容在整個工作階段中都會保留。
- 呼叫函數 ROW_COUNT、FOUND_ROWS 和 LAST_INSERT_ID 有時會導致鎖定。
- 預備陳述式導致代理鎖定工作階段。無論預備陳述式使用 SQL 文字或二進位協定，此規則都適用。
- 當您使用 SET LOCAL 時，RDS Proxy 不會釘選連線。
- 呼叫存放程序和存放函數不會導致鎖定。RDS Proxy 不會偵測由此類呼叫導致的任何工作階段狀態變更。如果您依賴該會話狀態在事務之間持續存儲，請確保您的應用程式不會更改存儲例程中的會話狀態。例如，RDS Proxy 目前與建立在所有交易中持續存在之暫存資料表的預存程序不相容。

如果您對應用程式行為具有專業知識，您可以略過特定應用程式陳述式的鎖定行為。若要這麼做，請在建立代理時選擇工作階段鎖定篩選條件選項。您目前可以選擇退出工作階段鎖定，以便設定工作階段變數和組態設定。

導致 RDS for PostgreSQL 鎖定的條件

若是 PostgreSQL，下列互動也會導致鎖定：

- 使用 SET 命令。

- 使用PREPARE、DISCARDDEALLOCATE、或EXECUTE命令來管理準備好的陳述式。
- 創建臨時序列，表或視圖。
- 宣告游標。
- 捨棄工作階段狀態。
- 在通知頻道上收聽。
- 載入程式庫模組，例如auto_explain。
- 使用函數操作序列，例如nextval和setval。
- 使用諸如pg_advisory_lock和之類的功能與鎖進行交互pg_try_advisory_lock。

Note

RDS 代理伺服器不會在交易層級建議鎖定上釘選 pg_advisory_xact_lockpg_advisory_xact_lock_shared，特別是pg_try_advisory_xact_lock、和pg_try_advisory_xact_lock_shared。

- 設定參數，或將參數重設為其預設值。具體來說，使用SET和set_config命令將默認值分配給會話變量。
- 呼叫存放程序和存放函數不會導致鎖定。RDS Proxy 不會偵測由此類呼叫導致的任何工作階段狀態變更。如果您依賴該會話狀態在事務之間持續存儲，請確保您的應用程式不會更改存儲例程中的會話狀態。例如，RDS Proxy 目前與建立在所有交易中持續存在之暫存資料表的預存程序不相容。

刪除 RDS Proxy

您可以在不再需要代理伺服器時刪除代理伺服器。或者，如果您將與其關聯的資料庫執行個體或叢集停止服務，則可以刪除 Proxy。

AWS Management Console

若要刪除代理

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 從清單中選擇要刪除的代理。
4. 選擇 Delete Proxy (刪除代理)。

AWS CLI

要刪除數據庫代理，請使用 AWS CLI 命令 [刪除](#)-DB 代理。若要移除相關的關聯，請同時使用 [deregister-db-proxy-targets](#) 命令。

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
  --db-proxy-name proxy_name
  [--target-group-name target_group_name]
  [--target-ids comma_separated_list]           # or
  [--db-instance-identifiers instance_id]       # or
  [--db-cluster-identifiers cluster_id]
```

RDS API

若要刪除資料庫代理，請呼叫 Amazon RDS API 函數 [DeleteDBProxy](#)。要刪除相關項目和關聯，您還可以調用函數 [DeleteDB ProxyTarget 組](#)和取消註冊數據庫。 [ProxyTargets](#)

使用 Amazon RDS Proxy 端點

了解 RDS Proxy 的端點，以及如何使用它們。透過使用 Proxy 端點，您可以利用下列功能：

- 您可以使用多個端點搭配代理，來獨立監控和不同應用程式的連線並進行疑難排解。
- 您可以使用跨 VPC 端點，以允許從資源存取一個 VPC 中的資料庫，例如 Amazon EC2 執行個體。

主題

- [代理端點概觀](#)
- [多可用區域資料庫叢集的 Proxy 端點](#)
- [跨 VPC 存取 RDS 資料庫](#)
- [建立代理端點](#)
- [檢視代理端點](#)
- [修改代理端點](#)
- [刪除代理端點](#)
- [代理端點的限制](#)

代理端點概觀

使用 RDS Proxy 端點涉及與 RDS 執行個體端點相同的程序類型。如果您不熟悉 RDS 端點，請在[連線至執行 MySQL 資料庫引擎的資料庫執行個體](#)和[連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)中查閱詳細資訊。

對於您建立的代理端點，您還可以將端點與代理本身所用的不同 Virtual Private Cloud (VPC) 建立關聯。如此一來，您可以從不同的 VPC 連線至 Proxy，例如，組織內不同應用程式使用的 VPC。

如需與代理端點關聯的限制相關資訊，請參閱[代理端點的限制](#)。

在 RDS Proxy 日誌中，每個項目均會以關聯的代理端點名稱作為前綴。此名稱可以是您針對使用者定義端點所指定的名稱。或者，它可以是執行讀取/寫入要求的 Proxy 預設端點的特殊名稱 default。

每個 Proxy 端點都有自己的一組 CloudWatch 指標。您可以監控代理所有端點的指標。您還可以監控特定端點的指標，或監控代理所有讀取/寫入或唯讀端點的指標。如需詳細資訊，請參閱[使用 Amazon 監控 RDS 代理指標 CloudWatch](#)。

代理端點會使用與相關聯代理相同的身分驗證機制。RDS Proxy 會自動設定使用者定義端點的許可和授權，且與相關聯代理的屬性一致。

多可用區域資料庫叢集的 Proxy 端點

依預設，您將 RDS Proxy 與多可用區域資料庫叢集搭配使用時連線的端點具有讀取/寫入功能。因此，此端點會將所有請求傳送至叢集的寫入器執行個體。所有這些連線都會計入寫入器執行個體的 max_connections 值。如果您的 Proxy 與多可用區域資料庫叢集相關聯，您可以為該 Proxy 建立額外的讀取/寫入或唯讀端點。

您可以將唯讀端點與您的代理搭配使用，進行唯讀查詢。執行此操作的方式與您將讀取器端點用於多可用區域資料庫叢集的方式相同。這樣做有助於您利用多可用區域資料庫叢集的讀取可擴展性，而此叢集具有一個或多個讀取器資料庫執行個體。您可以使用唯讀端點，並視需將更多讀取器資料庫執行個體新增至多可用區域資料庫叢集，來執行更多同時查詢並建立更多同時連線。這些讀取器端點有助於改善查詢密集型應用程式的讀取可擴展性。如果叢集中的讀取器資料庫執行個體變得無法使用，讀取器端點也有助於改善連線的可用性。

多可用區域資料庫叢集的讀取器端點

使用 RDS Proxy，您可以建立和使用讀取器端點。不過，這些端點僅適用於與多可用區域資料庫叢集相關聯的 Proxy。如果您使用 RDS CLI 或 API，您可能會看到具有值 TargetRole 的 READ_ONLY 屬性。您可以將代理目標從 RDS 資料庫執行個體變更為異地同步備份資料庫叢集，以利用此類 Proxy。

當您將 RDS Proxy 與多可用區域資料庫叢集搭配使用時，您可以建立並連線至稱為讀取器端點的唯讀端點。

讀取器端點如何協助應用程式變得可用

在某些情況下，叢集中的讀取器執行個體可能無法使用。如果發生此情況，使用資料庫 Proxy 讀取器端點的連線可以比使用多可用區域資料庫叢集讀取器端點的連線更快復原。RDS Proxy 只會將連線路由傳送至叢集中可用的讀取器執行個體。當執行個體變得無法使用時，DNS 快取不會有延遲。

如果連線為多路復用，RDS Proxy 會將後續查詢導向至不同的讀取器執行個體，而不會中斷您的應用程式。如果讀取器執行個體處於無法使用狀態，則會關閉該執行個體端點的所有用戶端連線。

如果連線鎖定，則連線的下一個查詢會傳回錯誤。不過，您的應用程式可以立即重新連線至相同的 Proxy 端點。RDS Proxy 會將連線路由至不同但為 available 狀態的讀取器資料庫執行個體。手動重新連線時，RDS Proxy 不會檢查新舊讀取器執行個體之間的複寫延遲。

如果多可用區域資料庫叢集沒有任何可用的讀取器執行個體，RDS Proxy 會在其可用時嘗試連線至讀取器端點。如果連線借用逾時期間內沒有讀取器執行個體變得可用，則連線嘗試會失敗。如果讀取器執行個體變得可用，則連線嘗試會成功。

讀取器端點如何協助實現查詢可擴展性

Proxy 的讀取器端點透過以下方式協助實現多可用區域資料庫叢集查詢可擴展性：

- 在實際情況下，RDS Proxy 針對使用特定讀取器端點連線的所有查詢問題，使用相同的讀取器資料庫執行個體。如此一來，相同資料表上的一組相關查詢可以利用特定資料庫執行個體上的快取、計劃最佳化等。
- 如果讀取器資料庫執行個體變得無法使用，則對應用程式的影響取決於工作階段是多路復用還是鎖定。如果工作階段是多路復用，RDS Proxy 會將任何後續查詢路由至不同的讀取器資料庫執行個體，而無需執行任何動作。如果工作階段已鎖定，您的應用程式會收到錯誤，並且必須重新連線。您可以立即重新連線至讀取器端點，並且 RDS Proxy 會將連線路由至可用的讀取器資料庫執行個體。如需代理工作階段的多工處理和鎖定的詳細資訊，請參閱[RDS Proxy 概念概觀](#)。

跨 VPC 存取 RDS 資料庫

依預設，Aurora 技術堆疊的元件全都在同一個 Amazon VPC 中。例如，假設在 Amazon EC2 執行個體上執行的應用程式連線至 Amazon RDS 資料庫執行個體。在此情況下，應用程式伺服器和資料庫必須都位於同一個 VPC 內。

使用 RDS 代理，您可以從另一個 VPC (例如 EC2 執行個體) 中的資源，在一個 VPC 中設定對 Amazon RDS 資料庫執行個體的存取權。例如，您的組織可能有多個存取相同資料庫資源的應用程式。每個應用程式都可能在其自己的 VPC 中。

若要啟用跨 VPC 存取，您可以為代理建立新的端點。Proxy 本身駐留在與 Amazon RDS 資料庫執行個體相同的 VPC 中。然而，跨 VPC 端點駐留在其他 VPC 中，以及 EC2 執行個體等其他資源。跨 VPC 端點與 EC2 和其他資源相同的 VPC 子網路和安全群組關聯。這些關聯可讓您從因 VPC 限制而無法存取資料庫的應用程式連線至端點。

下列步驟說明如何透過 RDS Proxy 來建立和存取跨 VPC 端點：

1. 建立兩個 VPC，或選擇兩個已用於和 RDS 工作的 VPC。每個 VPC 都應該有自己相關聯的網路資源，例如網際網路閘道、路由表、子網路和安全群組。如果您只有一個 VPC，可以參閱[Amazon RDS 入門](#)，了解設定另一個 VPC 以成功使用 RDS 的步驟。您也可以 Amazon EC2 主控台中檢查現有的 VPC，以查看要連接在一起的資源種類。
2. 建立與您要連線的 Amazon RDS 資料庫執行個體相關聯的資料庫 Proxy。請遵循 [建立 RDS Proxy](#) 中的程序。
3. 於 RDS 主控台代理的詳細資訊頁面上，在 Proxy endpoints (代理端點) 區段中，選擇 Create endpoint (建立端點)。請遵循 [建立代理端點](#) 中的程序。
4. 選擇要讓跨 VPC 端點讀取/寫入還是唯讀。
5. 不是接受與 Amazon RDS 資料庫執行個體相同 VPC 的預設值，而是選擇不同的 VPC。此 VPC 必須與代理所駐留的 VPC 位於相同的 AWS 區域。
6. 現在，不是接受與 Amazon RDS 資料庫執行個體相同 VPC 的子網路和安全群組預設值，而是選擇新的選項。根據您選擇的 VPC 中的子網路和安全群組進行這些設定。
7. 您不需要變更 Secrets Manager 機密的任何設定。相同的登入資料適用於代理的所有端點，無論每個端點所在的 VPC 為何。
8. 等待新端點為可用狀態。
9. 記下完整的端點名稱。這是以 *Region_name*.rds.amazonaws.com 結尾的值，作為資料庫應用程式連線字串的一部分提供。
10. 從與端點相同的 VPC 中的資源存取新端點。測試此程序的簡單方法是，在此 VPC 中建立一個新的 EC2 執行個體。然後，登入 EC2 執行個體，並使用連接字串中的端點值執行mysql或psql命令進行連線。

建立代理端點

主控台

若要建立代理伺服器端點

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 按一下您要為其建立新端點的代理名稱。

該代理的詳細資訊頁面即會出現。

4. 在 Proxy endpoints (代理端點) 區段中，選擇 Create proxy endpoint (建立代理端點)。

Create proxy endpoint (建立代理端點) 視窗即會出現。

5. 對於 Proxy endpoint name (代理端點名稱)，輸入您選擇的描述性名稱。
6. 對於 Target role (目標角色) 下，選擇要讓端點讀取/寫入還是唯讀。

使用讀取/寫入端點的連線可以執行任何類型的作業，例如資料定義語言 (DDL) 陳述式、資料操作語言 (DML) 陳述式和查詢。這些端點一律連線至 RDS 資料庫叢集的主要執行個體。當您在應用程式中僅使用單一端點時，您可以使用讀取/寫入端點來進行一般資料庫操作。您也可以將讀取/寫入端點用於管理作業、線上交易處理 (OLTP) 應用程式和 extract-transform-load (ETL) 工作。

使用唯讀端點的連線只能執行查詢。RDS Proxy 可以針對端點的每個連線使用其中一個讀取器執行個體。如此一來，查詢密集型應用程式可以利用多可用區域資料庫叢集的叢集功能。這些唯讀連線不會對叢集的主要執行個體施加任何額外負荷。如此一來，您的報告和分析查詢不會減慢 OLTP 應用程式的寫入操作。

7. 對於 Virtual Private Cloud (VPC) (VPC)，請選擇預設值，以便從通常用於存取 Proxy 或其關聯資料庫的相同 EC2 執行個體或其他資源存取端點。若要設定此代理的跨 VPC 存取，請選擇預設值以外的 VPC。如需跨 VPC 存取的詳細資訊，請參閱[跨 VPC 存取 RDS 資料庫](#)。
8. 對於子網路，RDS Proxy 預設會填入與關聯代理相同的子網路。若要將端點的存取限制為僅能夠連線至 VPC 位址範圍的一部分，請移除一或多個子網路。
9. 若為 VPC Security groups (VPC 安全群組)，您可以選擇現有的安全群組，或建立新的安全群組。依據預設，RDS Proxy 會填入與相關聯代理相同的安全群組或群組。如果 Proxy 的輸入和輸出規則適用於此端點，請保留預設選擇。

如果您選擇建立新的安全群組，請在此頁面上指定安全群組的名稱。然後稍後從 EC2 主控台編輯安全群組設定。

10. 選擇 Create proxy endpoint (建立代理端點)。

AWS CLI

若要建立代理端點，請使用AWS CLI [create-db-proxy-endpoint](#) 指令。

包含下列必要參數：

- `--db-proxy-name` *value*
- `--db-proxy-endpoint-name` *value*
- `--vpc-subnet-ids` *list_of_ids*. 以空格分隔子網路 ID。您沒有指定 VPC 本身的 ID。

您還可以包含下列選用參數：

- `--target-role` { `READ_WRITE` | `READ_ONLY` } 此參數預設為 `READ_WRITE`。如果代理伺服器與異地同步備份資料庫叢集關聯，只包含寫入器資料庫執行，則無法指定 `READ_ONLY`。如需將唯讀端點與 搭配使用的詳細資訊，請參閱 [多可用區域資料庫叢集的讀取器端點](#)。
- `--vpc-security-group-ids` *value*. 以空格分隔安全群組 ID。如果省略此參數，RDS Proxy 會使用預設的 VPC 安全群組。RDS Proxy 會根據您為 `--vpc-subnet-ids` 參數指定的子網路 ID 來決定 VPC。

Example

下列範例會建立名為 `my-endpoint` 的代理端點

對於LinuxmacOS、或Unix：

```
aws rds create-db-proxy-endpoint \  
  --db-proxy-name my-proxy \  
  --db-proxy-endpoint-name my-endpoint \  
  --vpc-subnet-ids subnet_id subnet_id subnet_id ... \  
  --target-role READ_ONLY \  
  --vpc-security-group-ids security_group_id ]
```

在Windows中：

```
aws rds create-db-proxy-endpoint ^  
  --db-proxy-name my-proxy ^  
  --db-proxy-endpoint-name my-endpoint ^  
  --vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^  
  --target-role READ_ONLY ^
```

```
--vpc-security-group-ids security_group_id
```

RDS API

若要建立代理主機端點，請使用 RDS API 建立 [資料庫動作ProxyEndpoint](#)。

檢視代理端點

主控台

若要檢視代理端點的詳細資訊

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 在清單中，選擇您要檢視其端點的代理。按一下代理名稱，以檢視其詳細資訊頁面。
4. 在 Proxy endpoints (代理端點) 區段中，選擇您要檢視的端點。按一下其名稱，以檢視詳細資訊頁面。
5. 檢查您感興趣的參數值。您可以檢查如下所示屬性：
 - 端點是讀取/寫入還是唯讀。
 - 您在資料庫連線字串中使用的端點地址。
 - 與端點關聯的 VPC、子網路和安全群組。

AWS CLI

若要檢視一或多個 Proxy 端點，請使用 AWS CLI [describe-db-proxy-endpoints](#) 指令。

您可以包含下列任一選用參數：

- --db-proxy-endpoint-name
- --db-proxy-name

以下範例描述 my-endpoint 代理端點。

Example

對於 Linux macOS、或 Unix：

```
aws rds describe-db-proxy-endpoints \  
  --db-proxy-endpoint-name my-endpoint
```

在Windows中：

```
aws rds describe-db-proxy-endpoints ^  
  --db-proxy-endpoint-name my-endpoint
```

RDS API

若要描述一或多個代理主機端點，請使用 RDS API [描述 B 作ProxyEndpoints](#)業。

修改代理端點

主控台

修改一個或多個代理端點

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Proxies (代理)。
3. 在清單中，選擇您要修改其端點的代理。按一下代理伺服器名稱以檢視其
4. 在 Proxy endpoints (代理端點) 區段中，選擇您要修改的端點。您可以在清單中選取或按一下其名稱，以檢視詳細資訊頁面。
5. 在代理詳細資訊頁面的 Proxy endpoints (代理端點) 區段中，選擇 Edit (編輯)。或者，在 Proxy 端點詳細資料頁面上，針對「動作」選擇「編輯」。
6. 變更您要修改的參數值。
7. 選擇儲存變更。

AWS CLI

若要修改 Proxy 端點，請使用具有下列必要參數的AWS CLI [modify-db-proxy-endpoint](#)指令：

- `--db-proxy-endpoint-name`

使用下列一個或多個參數來指定對端點屬性的變更：

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. 以空格分隔安全群組 ID。

下列範例會將 `my-endpoint` 代理端點重新命名為 `new-endpoint-name`。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-proxy-endpoint \  
  --db-proxy-endpoint-name my-endpoint \  
  --new-db-proxy-endpoint-name new-endpoint-name
```

在Windows中：

```
aws rds modify-db-proxy-endpoint ^  
  --db-proxy-endpoint-name my-endpoint ^  
  --new-db-proxy-endpoint-name new-endpoint-name
```

RDS API

若要修改代理主機端點，請使用 RDS API 修[改資料庫ProxyEndpoint](#)作業。

刪除代理端點

您可以如下所述使用主控台刪除代理的端點。

Note

您無法刪除 RDS Proxy 為每個代理自動建立的預設代理主機端點。
當您刪除代理時，RDS Proxy 會自動刪除所有關聯的端點。

主控台

若要使用 AWS Management Console 來刪除代理端點

1. 在導覽窗格中，選擇 Proxies (代理)。
2. 在清單中，選擇您要刪除其端點的代理。按一下代理名稱，以檢視其詳細資訊頁面。

3. 在 Proxy endpoints (代理端點) 區段中，選擇您要刪除的端點。您可以在清單中選取一個或多個端點，或者按一下單一端點的名稱，以檢視詳細資訊頁面。
4. 在代理詳細資訊頁面的 Proxy endpoints (代理端點) 區段中，選擇 Delete (刪除)。或者，在 Proxy 端點詳細資料頁面上，針對「動作」選擇「刪除」。

AWS CLI

若要刪除 Proxy 端點，請使用下列必要參數執行 [delete-db-proxy-endpoint](#) 命令：

- `--db-proxy-endpoint-name`

下列命令會刪除名為 `my-endpoint` 的代理端點。

對於 Linux/macOS、或 Unix：

```
aws rds delete-db-proxy-endpoint \  
  --db-proxy-endpoint-name my-endpoint
```

在 Windows 中：

```
aws rds delete-db-proxy-endpoint ^  
  --db-proxy-endpoint-name my-endpoint
```

RDS API

若要使用 RDS API 刪除代理主機端點，請執行 [刪除資料庫 ProxyEndpoint](#) 作業。指定代理端點 `DBProxyEndpointName` 參數的名稱。

代理端點的限制

RDS 代理主機端點有下列限制：

- 每個代理都有一個預設端點，您可以修改，但無法建立或刪除。
- 代理的使用者定義端點數目上限為 20。因此，代理最多可以有 21 個端點：預設端點，以及您建立的 20 個端點。
- 當您將其他端點與代理建立關聯時，RDS Proxy 會自動確定叢集中要用於每個端點的資料庫執行個體。

使用 Amazon 監控 RDS 代理指標 CloudWatch

您可以通過使用 Amazon 監控 RDS 代理 CloudWatch。CloudWatch 從代理收集原始數據並將其處理為可讀的 near-real-time 指標。若要在 CloudWatch 主控台中尋找這些指標，請選擇「指標」，然後選擇「RDS」，然後選擇「每個代理指標」。如需詳細資訊，請參閱 [Amazon 使用者指南中的使用 CloudWatch 用 Amazon 指 CloudWatch 標](#)。

Note

RDS 會針對與代理關聯的每個基礎 Amazon EC2 執行個體發佈這些指標。單一代理可能由一個以上的 EC2 執行個體服務。使用 CloudWatch 統計資料來彙總所有相關執行處理的 Proxy 值。

其中一些指標可能會在 Proxy 第一次成功連線之後才顯示。

在 RDS Proxy 日誌中，每個項目均會以關聯的代理端點名稱作為前綴。此名稱可以是您為使用者定義端點指定的名稱，也可以是執行讀取/寫入要求之 Proxy 預設端點的特殊名稱 default。

所有 RDS Proxy 指標都在群組 proxy 中。

每個 Proxy 端點都有自己的 CloudWatch 指標。您可以獨立監控每個代理端點的用量。如需代理端點的詳細資訊，請參閱 [使用 Amazon RDS Proxy 端點](#)。

您可以使用下列其中一個維度集來彙總每個指標的值。例如，藉由使用 ProxyName 維度集，您可以分析特定代理的所有流量。透過使用其他維度集，您可以用不同的方式來分割指標。您可以根據每個代理的不同端點或目標資料庫，或者每個資料庫的讀取/寫入和唯讀流量來分割指標。

- 維度集 1 : ProxyName
- 維度集 2 : ProxyName、EndpointName
- 維度集 3 : ProxyName、TargetGroup、Target
- 維度集 4 : ProxyName、TargetGroup、TargetRole

指標	描述	有效期間	CloudWatch 標註集
AvailabilityPercentage	目標群組在維度所指出的角色中可供使用的時間百分比。此指	1 分鐘	Dimension set 4

指標	描述	有效期間	CloudWatch 標註集
	標每分鐘報告一次。 此指標最實用的統計資料是 Average。		
ClientConnections	目前的用戶端連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 2
ClientConnectionsClosed	已關閉的用戶端連線數。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
ClientConnectionsNoTLS	不具有 Transport Layer Security (TLS) 的目前用戶端連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
ClientConnectionsReceived	接收到的用戶端連線要求數。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
ClientConnectionsSetupFailedAuth	因身分驗證或 TLS 設定錯誤而失敗的用戶端連線嘗試次數。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
ClientConnectionsSetupSucceeded	使用任何具有或不具 TLS 的身分驗證機制成功建立的用戶端連線數。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2

指標	描述	有效期間	CloudWatch 標註集
ClientConnectionsTLS	具有 TLS 的目前用戶端連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
DatabaseConnectionRequests	建立資料庫連線的要求數目。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionRequestsWithTLS	要透過 TLS 建立資料庫連線的要求數目。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnections	目前的資料庫連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionBorrowLatency	代理為了取得資料庫連線而受到監控所花費的時間 (以毫秒為單位)。此指標最實用的統計資料是 Average。	1 分鐘或以上	Dimension set 1 , Dimension set 2
DatabaseConnectionCurrentlyBorrowed	目前借用狀態中的資料庫連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4

指標	描述	有效期間	CloudWatch 標註集
DatabaseConnectionsCurrentlyInTransaction	交易中的目前資料庫連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionsCurrentlySessionPinned	由於變更工作階段狀態之用戶端要求中的操作而遭到鎖定的目前資料庫連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionsSetupFailed	失敗的資料庫連線要求數目。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionsSetupSucceeded	使用或不使用 TLS 成功建立的資料庫連線數。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 3 , Dimension set 4
DatabaseConnectionsWithTLS	具有 TLS 的目前資料庫連線數。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4
MaxDatabaseConnectionsAllowed	允許的資料庫連線數上限。此指標每分鐘報告一次。此指標最實用的統計資料是 Sum。	1 分鐘	Dimension set 1 , Dimension set 3 , Dimension set 4

指標	描述	有效期間	CloudWatch 標註集
QueryDatabaseResponseLatency	資料庫回應查詢已花費的時間 (以毫秒為單位)。此指標最實用的統計資料是 Average。	1 分鐘或以上	Dimension set 1 , Dimension set 2 , Dimension set 3 , Dimension set 4
QueryRequests	收到的查詢數。包含多個陳述式的查詢會計算為一個查詢。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
QueryRequestsNoTLS	從非 TLS 連線收到的查詢數。包含多個陳述式的查詢會計算為一個查詢。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
QueryRequestsTLS	從 TLS 連線收到的查詢數。包含多個陳述式的查詢會計算為一個查詢。此指標最實用的統計資料是 Sum。	1 分鐘或以上	Dimension set 1 , Dimension set 2
QueryResponseLatency	取得查詢請求與代理回應請求之間的時間 (以毫秒為單位)。此指標最實用的統計資料是 Average。	1 分鐘或以上	Dimension set 1 , Dimension set 2

您可以在中找到 RDS 代理活動 CloudWatch 的記錄檔AWS Management Console。每個代理在日誌群組頁面中都有一筆記錄。

⚠ Important

這些日誌檔是供人類使用，用於故障診斷，而非程式設計存取。日誌檔的格式和內容可能會變更。

尤其是，較舊的日誌不包含指示每個請求端點的任何前綴。在較新的日誌中，每個項目均會以關聯的代理端點名稱作為前綴。此名稱可以是您針對使用者定義端點所指定的名稱，也可以是請求的 default 特殊名稱 (使用代理的預設端點)。

使用 RDS Proxy 事件

事件表示環境中的變更，例如來自軟體即服務 (SaaS) 合作夥伴的 AWS 環境或服務或應用程式。或者，它可以是您自己的自定義應用程式或服務之一。例如，建立或修改 RDS Proxy 時，Amazon RDS 會產生一個事件。Amazon RDS 以近乎即時的方 EventBridge 式向 Amazon 提供活動。以下列出可訂閱的 Amazon RDS Proxy 事件清單以及 RDS Proxy 事件範例。

如需使用事件的詳細資訊，請參閱下列內容：

- 如需如何使用 AWS Management Console、AWS CLI 或 RDS API 檢視事件的指示，請參閱[檢視 Amazon RDS 事件](#)。
- 若要了解如何設定要將事件傳送到的 Amazon RDS EventBridge，請參閱[建立由 Amazon RDS 事件觸發的規則](#)。

RDS Proxy 事件

下表顯示 RDS Proxy 為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	訊息	備註
組態變更	RDS-EVENT-0204	RDS 已修改資料庫代理 <i>name</i> 。	
組態變更	RDS-EVENT-0207	RDS 已修改資料庫代理 <i>name</i> 的端點。	
組態變更	RDS-EVENT-0213	RDS 偵測到新增了資料庫執行個體，並自動將其新增	

類別	RDS 事件 ID	訊息	備註
		到資料庫代理 <i>name</i> 的目標群組。	
組態變更	RDS-EVENT-0213	RDS 偵測到建立了資料庫執行個體 <i>name</i> ，並自動將其新增至資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中。	
組態變更	RDS-EVENT-0214	RDS 偵測到刪除了資料庫執行個體 <i>name</i> ，並自動將其從資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中移除。	
組態變更	RDS-EVENT-0215	RDS 偵測到刪除了資料庫叢集 <i>name</i> ，並自動將其從資料庫代理 <i>name</i> 的目標群組 <i>name</i> 中移除。	
建立	RDS-EVENT-0203	RDS 已建立資料庫代理 <i>name</i> 。	
建立	RDS-EVENT-0206	RDS 已建立資料庫代理 <i>name</i> 的端點 <i>name</i> 。	
刪除	RDS-EVENT-0205	RDS 已刪除資料庫代理 <i>name</i> 。	
刪除	RDS-EVENT-0208	RDS 已刪除資料庫代理 <i>name</i> 的端點 <i>name</i> 。	

類別	RDS 事件 ID	訊息	備註
失敗	RDS-EVENT-0243	RDS 無法佈建代理 <i>name</i> 的容量，因為您的子網路中沒有足夠的可用 IP 地址： <i>name</i> 。若要修正此問題，請確定您的子網路具有最低數量的未使用 IP 地址，如 RDS Proxy 文件所建議。	若要確定執行個體類別的建議數量，請參閱 規劃 IP 地址容量 。
失敗	RDS-EVENT-0275	<i>RDS #####</i> <i>##</i> 從用戶端到 Proxy 的同時連線要求數目已超過上限。	

以下是 JSON 格式的 RDS Proxy 事件範例。事件顯示 RDS 修改了 my-endpoint RDS Proxy 的 my-rds-proxy 端點。事件 ID 是 RDS-EVENT-0207。

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Proxy Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PROXY",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "RDS modified endpoint my-endpoint of DB Proxy my-rds-proxy.",
    "SourceIdentifier": "my-endpoint",
    "EventID": "RDS-EVENT-0207"
  }
}
```

}

RDS Proxy 命令列範例

若要查看連線命令和 SQL 陳述式的組合如何與 RDS Proxy 互動，請參閱下列範例。

範例

- [Preserving Connections to a MySQL Database Across a Failover](#)
- [Adjusting the max_connections Setting for an Aurora DB Cluster](#)

Example 透過容錯移轉保留與 MySQL 資料庫的連線

此 MySQL 範例將示範已開啟的連線如何在容錯移轉期間繼續運作。例如，當您重新啟動資料庫，或資料庫因由於有問題而無法使用時。本範例使用一個名為 the-proxy 的代理，以及一個具有資料庫執行個體 instance-8898 和 instance-9814 的 Aurora 資料庫叢集。從 Linux failover-db-cluster 命令列執行命令時，代理連線的寫入器執行個體會變更為不同的資料庫執行個體。您可以看到與代理關聯的資料庫執行個體改變了，但連線仍然保持開啟。

```
$ mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p
Enter password:
...

mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-9814      |
+-----+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                  mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
    -u admin_user -p
$ # Initially, instance-9814 is the writer.
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-8898 is the writer.
$ fg
```



```
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-8898      |
+-----+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                  mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
    -u admin_user -p
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-9814 is the writer again.
$ fg
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-9814      |
+-----+
1 row in set (0.01 sec)
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| hostname      | ip-10-1-3-178 |
+-----+-----+
1 row in set (0.02 sec)
```

Example 調整 Aurora 資料庫叢集的 max_connections 設定

此範例示範如何調整 Aurora MySQL 資料庫叢集的 max_connections 設定。若要這麼做，請根據 MySQL 5.7 相容叢集的預設參數設定，建立自己的資料庫叢集參數群組。您可以為 max_connections 設定指定一個值，以取代設定預設值的公式。將該資料庫叢集參數群組與資料庫叢集相關聯。

```
export REGION=us-east-1
export CLUSTER_PARAM_GROUP=rds-proxy-mysql-57-max-connections-demo
```

```
export CLUSTER_NAME=rds-proxy-mysql-57

aws rds create-db-parameter-group --region $REGION \
  --db-parameter-group-family aurora-mysql5.7 \
  --db-parameter-group-name $CLUSTER_PARAM_GROUP \
  --description "Aurora MySQL 5.7 cluster parameter group for RDS Proxy demo."

aws rds modify-db-cluster --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP

echo "New cluster param group is assigned to cluster:"
aws rds describe-db-clusters --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --query '*[*].{DBClusterParameterGroup:DBClusterParameterGroup}'

echo "Current value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"

echo -n "Enter number for max_connections setting: "
read answer

aws rds modify-db-cluster-parameter-group --region $REGION --db-cluster-parameter-
group-name $CLUSTER_PARAM_GROUP \
  --parameters "ParameterName=max_connections,ParameterValue=$
$answer,ApplyMethod=immediate"

echo "Updated value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"
```

RDS Proxy 的故障診斷

接下來，您可以找到一些常見 RDS Proxy 問題的疑難排解想法，以及 RDS Proxy 的 CloudWatch 記錄檔上的資訊。

在 RDS Proxy 日誌中，每個項目均會以關聯的代理端點名稱作為前綴。此名稱可以是您針對使用者定義端點所指定的名稱。或者，它可以是執行讀取/寫入要求的 Proxy 預設端點的特殊名稱 default。如需代理端點的詳細資訊，請參閱[使用 Amazon RDS Proxy 端點](#)。

主題

- [驗證代理的連線能力](#)
- [常見問題與解決方案](#)

驗證代理的連線能力

您可以使用下列命令來確認連線中的所有元件 (例如 Proxy、資料庫和計算執行個體) 都可以彼此通訊。

使用[describe-db-proxies](#)指令檢查代理本身。還可以使用 [describe-db-proxy-target-groups](#) 指令來檢查相關聯的目標群組。檢查目標的詳細資訊是否符合您打算與 Proxy 建立關聯的 RDS 資料庫執行個體。使用如下命令。

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

若要確認代理主機可以連線至基礎資料庫，請使用[describe-db-proxy-targets](#)命令檢查目標群組中指定的目標。使用如下命令。

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

[describe-db-proxy-targets](#)命令的輸出包括一個 TargetHealth 字段。您可以檢查 State 內的 Reason、Description 和 TargetHealth 欄位，以檢查代理是否能與基礎資料庫執行個體進行通訊。

- State 的 AVAILABLE 值表示代理可連線至資料庫執行個體。
- State 的 UNAVAILABLE 值表示暫時或永久的連線問題。在這種情況下，請檢查 Reason 和 Description 欄位。例如，如果 Reason 具有 PENDING_PROXY_CAPACITY 的值，請在代理完成其擴展操作後嘗試再次連線。如果 Reason 具有 UNREACHABLE、CONNECTION_FAILED 或 AUTH_FAILURE 的值，請使用來自 Description 欄位的解說，協助您診斷此問題。
- State 欄位在變更為 REGISTERING 或 AVAILABLE 前，可能短暫具有 UNAVAILABLE 的值。

如果下列 Netcat 命令 (nc) 成功，您可以從 EC2 執行個體或登入的其他系統存取代理端點。如果您與代理和關聯的資料庫不在同一 VPC 中，則此命令會回報失敗。您可能可以直接登入資料庫，而不需要同一個 VPC 中。然而除非您位於同一個 VPC 中，否則無法登入代理。

```
nc -zx MySQL_proxy_endpoint 3306

nc -zx PostgreSQL_proxy_endpoint 5432
```

您可以使用下列命令確定您的 EC2 執行個體具有需要的屬性。特別是，EC2 執行個體的 VPC 必須與 Proxy 連線的 VPC 相同。

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

檢查用於代理的 Secrets Manager 秘密。

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

請確定顯示的 SecretString 欄位 get-secret-value 已編碼為包含 username 和 password 欄位的 JSON 字串。下列範例顯示 SecretString 欄位的格式。

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreateDate": some_timestamp
}
```

常見問題與解決方案

本節說明使用 RDS Proxy 時的一些常見問題和可能的解決方案。

執行 `aws rds describe-db-proxy-targets` CLI 命令之後，如果 TargetHealth 說明狀態 Proxy does not have any registered credentials，請驗證下列項目：

- 有已註冊的憑證，供使用者存取代理。
- 存取代理伺服器使用的秘 Secrets Manager 碼的 IAM 角色有效。

建立或連線至資料庫代理時，您可能會遇到下列 RDS 事件。

類別	RDS 事件 ID	描述
失敗	RDS-EVENT-0243	RDS 無法佈建代理的容量，因為子網路中沒有足夠的可用 IP 地址。若要修正此問題，請確認您的子網路具有最低數目的未使用 IP 地址。若要確定執行個體類別的建議數量，請參閱 規劃 IP 地址容量 。
失敗	RDS-EVENT-0275	<i>RDS #####</i> 從用戶端到 Proxy 的同時連線要求數目已超過上限。

建立新代理或連線至代理時，可能會遇到下列問題。

錯誤	原因或因應措施
403: The security token included in the request is invalid	選取現有的 IAM 角色，而不是選擇建立新的 IAM 角色。

連線到 MySQL 代理時可能會遇到下列問題。

錯誤	原因或因應措施
ERROR 1040 (HY000): Connections rate limit	從用戶端到代理的連線要求速率已超過限制。

錯誤	原因或因應措施
exceeded (<i>limit_value</i>)	
ERROR 1040 (HY000): IAM authentication rate limit exceeded	從用戶端到代理器的 IAM 身分驗證同時請求數目已超過限制。
ERROR 1040 (HY000): Number simultane ous connectio ns exceeded (<i>limit_value</i>)	從用戶端到代理的同時連線要求數目超過限制。
ERROR 1045 (28000): Access denied for user ' <i>DB_USER</i> '@'%' (usi password: YES)	代理使用的 Secrets Manager 私密不符合現有資料庫使用者的使用者名稱和密碼。更新 Secrets Manager 私密中的登入資料，或確認資料庫使用者存在，且具有與私密中相同的密碼。
ERROR 1105 (HY000): Unknown error	發生未知的錯誤。
ERROR 1231 (42000): Variable 'character _set_cl ient'' can't be set to the value of <i>value</i>	為 <code>character_set_client</code> 參數設定的值無效。例如，值 <code>ucs2</code> 是無效的，因為它可能會讓 MySQL 伺服器當機。

錯誤	原因或因應措施
ERROR 3159 (HY000): This RDS Proxy requires TLS connections.	<p>您已在代理中啟用需要 Transport Layer Security 設定，但您的連線已在 MySQL 用戶端中包含參數 <code>ssl-mode=DISABLED</code>。執行下列任何一項：</p> <ul style="list-style-type: none"> 為代理停用需要 Transport Layer Security 設定。 使用 MySQL 用戶端中 <code>ssl-mode=REQUIRED</code> 的最小設定來連線至資料庫。
ERROR 2026 (HY000): SSL connection error: Internal Server <i>Error</i>	<p>TLS 與代理的交握失敗。一些可能的原因包括：</p> <ul style="list-style-type: none"> SSL 是必要的，但伺服器不支援它。 發生內部伺服器錯誤。 發生錯誤的交握。
ERROR 9501 (HY000): Timed-out waiting to acquire database connection	<p>代理等候取得資料庫連線逾時。一些可能的原因包括：</p> <ul style="list-style-type: none"> 代理無法建立資料庫連線，因為已達到最大連線數 代理無法建立資料庫連線，因為資料庫無法使用。

連線到 PostgreSQL 代理時可能會遇到下列問題。

錯誤	原因	解決方案
IAM authentication is allowed only with SSL connections.	使用者嘗試使用 PostgreSQL 用戶端中的 <code>sslmode=disable</code> 設定，透過 IAM 身分驗證連線至資料庫。	使用者需要使用 PostgreSQL 用戶端中 <code>sslmode=require</code> 的最小設定連接到資料庫。如需詳細資訊，請參閱 PostgreSQL SSL 支援 文件。
This RDS Proxy requires TLS connections.	使用者啟用需要 Transport Layer Security 選項，但嘗試使用 PostgreSQL 用戶端中的 <code>sslmode=disable</code> 連線。	若要修正此錯誤，請執行下列其中一項操作： <ul style="list-style-type: none"> 停用代理的需要 Transport Layer Security 選項。

錯誤	原因	解決方案
<p>IAM authentication failed for user <i>user_name</i> . Check the IAM token for this user and try again.</p>	<p>此錯誤可能由下列原因造成：</p> <ul style="list-style-type: none"> • 用戶端提供了不正確的 IAM 使用者名稱。 • 用戶端為使用者提供了不正確的 IAM 授權字符。 • 用戶端使用的 IAM 政策沒有必要許可。 • 用戶端為使用者提供了過期的 IAM 授權字符。 	<ul style="list-style-type: none"> • 使用 PostgreSQL 用戶端中 <code>sslmode=allow</code> 的最小設定連線至資料庫。 <p>若要修正此錯誤，請執行下列動作：</p> <ol style="list-style-type: none"> 1. 確認提供的 IAM 使用者存在。 2. 確認 IAM 授權字符屬於所提供的 IAM 使用者。 3. 確認 IAM 政策具有適當的 RDS 許可。 4. 檢查所使用之 IAM 授權字符的有效性。
<p>This RDS proxy has no credentials for the role <i>role_name</i> . Check the credentials for this role and try again.</p>	<p>這個角色沒有 Secrets Manager 秘密</p>	<p>新增此角色的 Secrets Manager 秘密。如需詳細資訊，請參閱設定 AWS Identity and Access Management (IAM) 政策。</p>
<p>RDS supports only IAM, MD5, or SCRAM authentication.</p>	<p>用來連線到代理的資料庫用戶端正在使用代理目前不支援的身分驗證機制。</p>	<p>如果您未使用 IAM 身分驗證，請使用 MD5 或 SCRAM 密碼身分驗證。</p>
<p>A user name is missing from the connection startup packet. Provide a user name for this connection.</p>	<p>嘗試建立連線時，用來連線至 Proxy 的資料庫用戶端並未傳送使用者名稱。</p>	<p>使用您選擇的 PostgreSQL 用戶端設定與代理連線時，請務必定義使用者名稱。</p>

錯誤	原因	解決方案
Feature not supported : RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.	用於連線到代理的 PostgreSQL 用戶端使用的通訊協定早於 3.0。	使用支援 3.0 訊息通訊協定的較新 PostgreSQL 用戶端。如果您使用的是 PostgreSQL psql CLI，請使用大於或等於 7.4 的版本。
Feature not supported : RDS Proxy currently doesn't support streaming replication mode.	用來連線至 Proxy 的 PostgreSQL 用戶端正在嘗試使用串流複寫模式，而 RDS Proxy 目前不支援這種模式。	關閉用於連線的 PostgreSQL 用戶端中的串流複寫模式。
Feature not supported : RDS Proxy currently doesn't support the option <i>option_name</i> .	透過啟動訊息，用來連線至 Proxy 的 PostgreSQL 用戶端正在要求 RDS Proxy 目前不支援的選項。	在用於連線的 PostgreSQL 用戶端中，關閉上述訊息中顯示為不支援的選項。
The IAM authentication failed because of too many competing requests.	從用戶端到代理的 IAM 身分驗證同時請求數目已超過限制。	降低使用 PostgreSQL 用戶端的 IAM 驗證建立連線的速率。
The maximum number of client connections to the proxy exceeded <i>number_value</i> .	從用戶端到代理的同時連線要求數目超過限制。	減少從 PostgreSQL 用戶端到此 RDS Proxy 的作用中連線數目。
Rate of connection to proxy exceeded <i>number_value</i> .	從用戶端到代理的連線要求速率已超過限制。	降低從 PostgreSQL 用戶端建立連線的速率。
The password that was provided for the role <i>role_name</i> is wrong.	此角色的密碼與 Secrets Manager 密碼不符。	在 Secrets Manager 檢查此角色的密碼，以查看密碼是否與 PostgreSQL 用戶端中使用的密碼相同。

錯誤	原因	解決方案
The IAM authentication failed for the role <i>role_name</i> . Check the IAM token for this role and try again.	用於 IAM 驗證的 IAM 符記發生問題。	產生一個新的身分驗證符記，並在新的連線中加以使用。
IAM is allowed only with SSL connections.	用戶端嘗試使用 IAM 驗證進行連線，但未啟用 SSL。	在用戶端中啟用 SSL。
Unknown error.	發生未知的錯誤。	聯絡 AWS Support，以便調查問題。
Timed-out waiting to acquire database connection.	代理等候取得資料庫連線逾時。一些可能的原因包括： <ul style="list-style-type: none"> 代理無法建立資料庫連線，因為已達到最大連線數 代理無法建立資料庫連線，因為資料庫無法使用。 	可能的解決方案如下： <ul style="list-style-type: none"> 檢查狀態的目標，以查看它是否無法使用。 檢查是否有長時間執行和/或查詢正在執行的交易。他們可以使用來自連線集區的資料庫連線很長一段時間。
Request returned an error: <i>database_error</i> .	從 Proxy 建立的資料庫連線傳回錯誤。	解決方案取決於特定的資料庫錯誤。其中一個範例為 Request returned an error: database "your-database-name" does not exist。這表示指定的資料庫名稱不存在於資料庫伺服器上。或者，它表示用作資料庫名稱的使用者名稱 (如果未指定資料庫名稱) 不存在於伺服器上。

將 RDS Proxy 與 AWS CloudFormation 搭配使用

您可以將 RDS Proxy 與 AWS CloudFormation 搭配使用。這可協助您建立相關資源群組。這類群組可以包含可連線至新建立之 Amazon RDS 資料庫執行個體的 Proxy。AWS CloudFormation 中的 RDS Proxy 支援包含兩個新的註冊類型：DBProxy 和 DBProxyTargetGroup。

下列清單會顯示 RDS Proxy 的範例 AWS CloudFormation 範本。

```
Resources:
  DBProxy:
    Type: AWS::RDS::DBProxy
    Properties:
      DBProxyName: CanaryProxy
      EngineFamily: MYSQL
      RoleArn:
        Fn::ImportValue: SecretReaderRoleArn
      Auth:
        - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IMAuth: DISABLED}
      VpcSubnetIds:
        Fn::Split: [",", "Fn::ImportValue": SubnetIds]

  ProxyTargetGroup:
    Type: AWS::RDS::DBProxyTargetGroup
    Properties:
      DBProxyName: CanaryProxy
      TargetGroupName: default
      DBInstanceIdentifiers:
        - Fn::ImportValue: DBInstanceName
    DependsOn: DBProxy
```

如需詳細了解此範例中的資源，請參閱 [DBProxy](#) 和 [DBProxyTargetGroup](#)。

如需有關您可以使用 AWS CloudFormation 建立的資源的詳細資訊，請參閱 [RDS 資源類型參考](#)。

使用與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽)

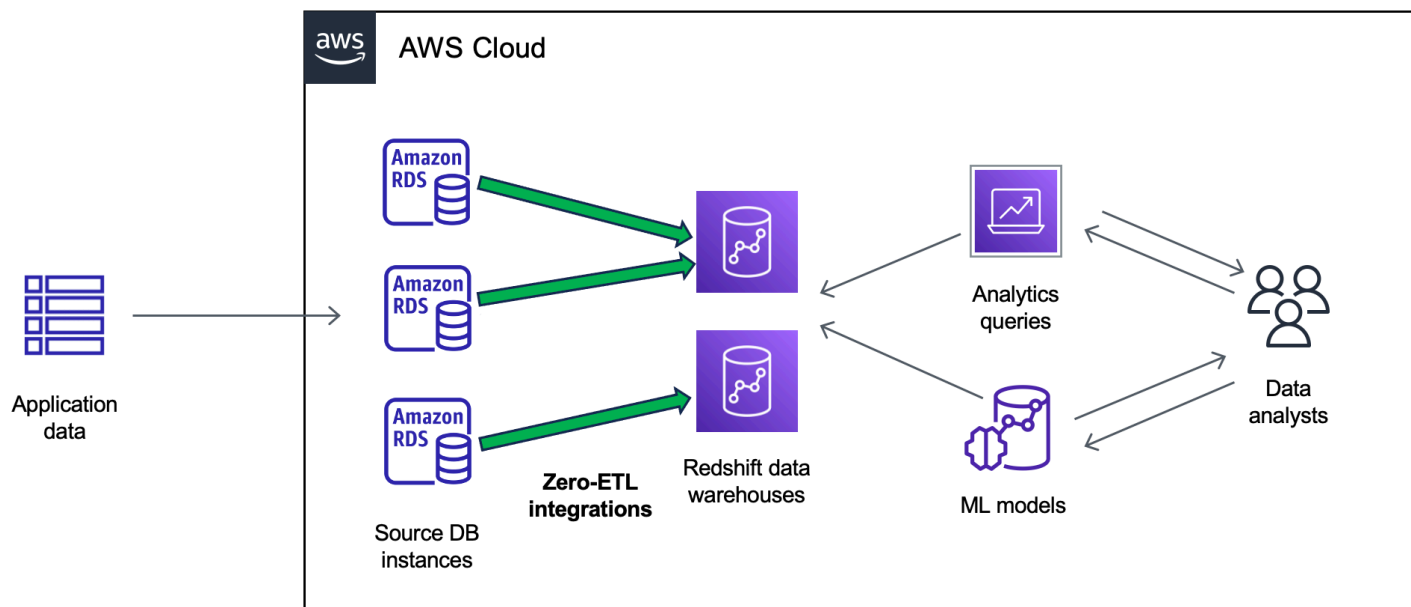
這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

與 Amazon Redshift 的 Amazon RDS 零 ETL 整合可在來自 RDS 的 PB 級交易資料上使用 Amazon Redshift 啟用近乎即時的分析和機器學習 (ML)。這是一個全受管的解決方案，可讓交易資料寫入 RDS 資料庫 Aurora 資料庫後，在 Amazon Redshift 中使用。擷取、轉換和載入 (ETL) 是將來自多個來源的資料合併為大型中央資料倉儲的程序。

零 ETL 整合可讓 RDS 資料庫 Aurora 資料庫中的資料以近乎即時的的速度在 Amazon Redshift 中使用。資料存放在 Amazon Redshift 之後，您可以使用 Amazon Redshift 的內建功能 (例如機器學習、具體化檢視、資料共用、聯合存取多個資料存放區和資料湖的聯合存取，以及與 Amazon、Amazon SageMaker 等整合)，為您的分析、機器學習和 AI 工作負載提供支援。QuickSight AWS 服務

若要建立零 ETL 整合，請將 RDS 資料庫 Aurora 資料庫指定為來源，並將 Amazon Redshift 資料倉儲指定為目標。整合會將來源資料庫中的資料複寫到目標資料倉儲。

下圖說明此功能。



整合會監控資料管道的運作狀態，並在可能的情況下從問題中復原。您可以從多個 RDS 資料庫 Aurora 資料庫集建立整合到單一 Amazon Redshift 命名空間，以便跨多個應用程式獲得洞見。

主題

- [優勢](#)
- [重要概念](#)
- [預覽限制](#)
- [配額](#)
- [支援地區](#)
- [開始使用與 Amazon Redshift 的 Amazon RDS 零 ETL 整合](#)
- [建立與 Amazon Redshift 的 Amazon RDS 零 ETL 整合](#)
- [將資料新增至來源 RDS 資料庫，並在 Amazon Redshift 中進行查詢](#)
- [檢視和監控與 Amazon Redshift 的 Amazon RDS 零 ETL 整合](#)
- [刪除與 Amazon Redshift 的 Amazon RDS 零 ETL 整合](#)
- [對與 Amazon Redshift 的 Amazon RDS 零 ETL 整合進行疑難排解](#)

優勢

與 Amazon Redshift 的 RDS 零 ETL 整合具有下列主要優勢：

- 協助您從多個資料來源衍生整體洞見。
- 無需建置和維護執行擷取、轉換和載入 (ETL) 操作的複雜資料管道。Zero-ETL 整合會透過為您佈建和管理管道，免除建置和管理這些管道所帶來的挑戰。
- 減少操作負擔和成本，讓您專注於改善應用程式。
- 讓您利用 Amazon Redshift 的分析和機器學習功能，從交易和其他資料衍生洞察，以有效地回應關鍵、時間敏感的事件。

重要概念

當您開始使用零 ETL 整合時，請考慮下列概念：

整合

一種全受管資料管道，可自動將交易資料和結構描述從 RDS 資料庫 複製到 Amazon Redshift 資料倉儲。

來源資料庫

從中複製資料的 RDS 資料庫。您可以指定單一可用區或異地同步備份資料庫執行個體。

目標資料倉儲

將資料複製至其中的 Amazon Redshift 資料倉儲。資料倉儲有兩種類型：[佈建的叢集](#)資料倉儲和[無伺服器](#)資料倉儲。佈建的叢集資料倉儲是稱為節點的運算資源集合，這些節點會組織成稱為叢集的群組。無伺服器資料倉儲由存放運算資源的工作群組，以及為資料庫物件和使用者提供空間的命名空間所組成。這兩個資料倉儲都會執行 Amazon Redshift 引擎，並包含一或多個資料庫。

多個來源資料庫資料庫可以寫入相同的目標。

如需詳細資訊，請參閱《Amazon Redshift 開發人員指南》中的[資料倉儲系統架構](#)。

預覽限制

以下限制適用於與 Amazon Redshift 的 RDS 零 ETL 整合。

主題

- [一般限制](#)
- [RDS for MySQL 限制](#)
- [Amazon Redshift 限制](#)

一般限制

- 來源資料庫必須與目標 Amazon Redshift 資料倉儲位於相同的區域。
- 如果資料庫資料庫個體具有現有整合，則無法重新命名該叢集。
- 您無法刪除具有現有整合功能的資料庫資料庫。您必須先刪除所有相關聯的整合。
- 如果停止來源資料庫資料，則在您恢復資料庫集之前，最後幾個交易可能不會複製到目標資料倉儲。
- 如果來源資料庫已停止，則無法刪除整合。
- Amazon RDS 僅支援單一可用區和異地同步備份資料庫執行個體部署做為整合來源。它目前不支援異地同步備份資料庫叢集。

- 零 ETL 整合目前不支援資料篩選。
- 如果您的資料庫是藍/綠部署的來源，則在切換期間，藍色和綠色環境就無法擁有現有的 Zero-ETL 整合。您必須先刪除整合再進行轉換，然後重新建立該整合。
- 您無法為正在主動建立另一個整合的來源資料庫建立整合。
- 當您一開始建立整合時或當資料表重新同步時，從來源植入目標的資料可能需要 20 到 25 分鐘或更長時間，取決於來源資料庫的大小。此延遲可能會導致複本延遲增加。
- 不支援某些資料類型。如需詳細資訊，請參閱 [the section called “資料類型差異”](#)。
- 不支援具有預先定義資料表更新的外部索引鍵參照。具體而言，ON DELETE和動作不支援和ON UPDATE規SET DEFAULT則。CASCADE SET NULL嘗試建立或更新對另一個資料表具有這類參考的資料表，會將該資料表置於失敗狀態。
- ALTER TABLE分割區操作會導致您的資料表重新同步，以便將資料從 RDS 重新載入到 Amazon Redshift。資料表將無法在重新同步處理時進行查詢。如需詳細資訊，請參閱 [the section called “我的一個或多個 Amazon Redshift 資料表需要重新同步”](#)。
- 不支援 XA 交易。
- 物件識別符 (包括資料庫名稱、資料表名稱、資料欄名稱等) 只能包含英數字元、數字、\$ 和 _ (底線)。

RDS for MySQL 限制

- 您的來源資料庫必須執行適用於 MySQL 的 RDS 版本 8.0.32 或更高版本。
- 零 ETL 整合依賴 MySQL 二進位記錄 (binlog)，來擷取持續的資料變更。請勿使用 binlog 型資料篩選，因為這可能會導致來源和目標資料庫之間的資料不一致。
- RDS for MySQL 系統資料表、暫時資料表和檢視不會複寫到 Amazon Redshift。
- 僅針對設定為使用 InnoDB 儲存體引擎的資料庫支援零 ETL 整合。
- 來源資料庫叢集無法使用憑證授權單位 (CA) 來設定 rds-ca-ecc384-g1。

Amazon Redshift 限制

如需與零 ETL 整合相關的 Amazon Redshift 限制清單，請參閱 Amazon Red shift 管理指南中的 [考量事項](#)。

配額

您的帳戶具有與 Amazon Redshift 的 RDS 零 ETL 整合相關的下列配額。除非另有說明，否則每個配額都是根據區域而定。

名稱	預設	描述
整合	100	AWS 帳戶內的整合總數。
每個目標資料倉儲的整合	50	將資料傳送至單一目標 Amazon Redshift 資料倉儲的整合數目。
每個來源執行個體的整合	1	從單一來源資料庫執行個體資料庫傳送資料的整合數目。

此外，Amazon Redshift 會對每個資料庫執行個體或叢集節點中允許的資料表數目設定某些限制。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的 [Amazon Redshift 中的配額和限制](#)。

支援地區

RDS 零 ETL 與 Amazon Redshift 集成在一個子集成中提供。AWS 區域如需支援的區域的清單，請參閱 [the section called “零 ETL 整合”](#)。

開始使用與 Amazon Redshift 的 Amazon RDS 零 ETL 整合

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

在使用 Amazon Redshift 建立零 ETL 整合之前，請先使用必要的參數和許可設定 RDS 資料庫和 Amazon Redshift 資料倉儲。安裝期間，您將完成以下步驟：

1. [建立自訂資料庫參數群組。](#)
2. [建立來源資料庫資料庫。](#)

3. [建立目標 Amazon Redshift 資料倉儲。](#)

完成這些步驟後，請繼續[the section called “建立零 ETL 整合”](#)。

步驟 1：建立自訂資料庫參數群組。

Amazon RDS 零 ETL 與 Amazon Redshift 整合需要特定的資料庫參數值，才能控制二進位記錄 (BINLOG)。若要設定二進位記錄，您必須先建立自訂資料庫參數群組，然後將其與來源資料庫建立關聯。

根據您的來源資料庫參數群組。如需建立參數群組的指示，請參閱 [the section called “使用資料庫參數群組”](#)。

- binlog_format=ROW
- binlog_row_image=full
- binlog_checksum=NONE

此外，請確定binlog_row_value_options 參數未設定為 PARTIAL_JSON。

步驟 2：選取或建立來源資料庫資料庫

建立自訂資料庫參數群組後，請選擇或建立適用於 MySQL 的 RDS 單一可用區或異地同步備份資料庫執行個體 SQL 資料庫叢集。此資料庫將成為複寫至 Amazon Redshift 的資料來源。

資料庫必須執行適用於 MySQL 的 RDS 版本 8.0.32 或更高版本，SQL 15.4 和零 ETL Support)。如需建立單一可用區或異地同步備份資料庫執行個體資料庫的指示，請參閱 [the section called “建立資料庫執行個體”](#)

在 [其他組態] 下，將預設資料庫參數群組變更為您在上一個步驟中建立的自訂參數群組。

Note

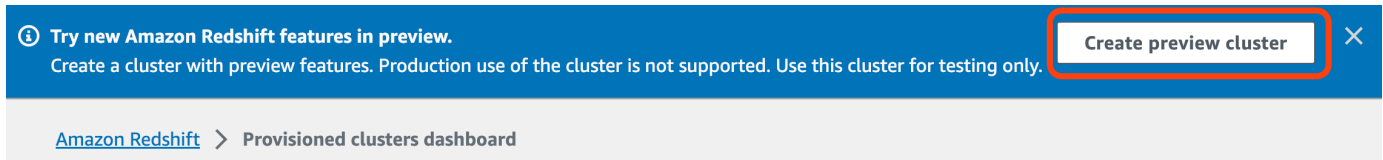
如果在建立資料庫叢集之後將參數群組與資料庫建立關聯，則必須重新啟動個體以套用變更，然後才能建立零 ETL 整合。如需說明，請參閱[the section called “將資料庫執行個體重新開機”](#)。

此外，請確定已在資料庫上啟用自動備份。如需詳細資訊，請參閱 [the section called “啟用自動備份”](#)。

步驟 3：建立目標 Amazon Redshift 資料倉儲

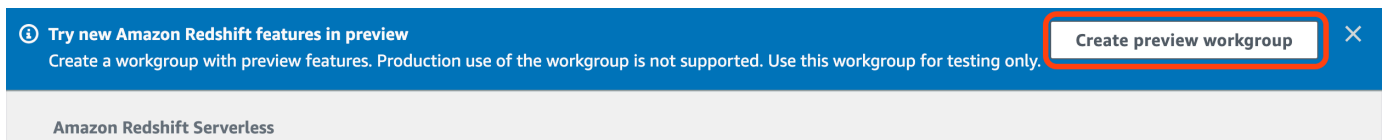
建立來源資料庫資料庫之後，您必須在 Amazon Redshift 中建立和設定目標資料倉儲。資料倉儲必須符合下列需求：

- 在預覽中建立
 - 若要在預覽中建立已佈建的叢集，請從已佈建的叢集儀表板上的橫幅選擇建立預覽叢集。如需詳細資訊，請參閱[建立預覽叢集](#)。



建立叢集時，請將預覽追蹤設定為 `preview_2023`。

- 若要在預覽中建立 Redshift Serverless 工作群組，請從無伺服器儀表板上的橫幅中選擇建立預覽工作群組。如需詳細資訊，請參閱[建立預覽工作群組](#)。



- 使用具有至少兩個節點的 RA3 節點類型 (`ra3.x1plusra3.4xlarge`、或 `ra3.16xlarge`)，或 Redshift 無伺服器。
- 已加密 (如果使用已佈建的叢集)。如需詳細資訊，請參閱 [Amazon Redshift 資料庫加密](#)。

如需建立資料倉儲的指示，請參閱[建立叢集](#) (適用於佈建的叢集)，或[使用命名空間建立工作群組](#) (適用於 Redshift Serverless)。

在資料倉儲上啟用區分大小寫

若要成功整合，必須為資料倉儲啟用區分大小寫參數 (`enable_case_sensitive_identifier`)。依預設，所有佈建的叢集和 Redshift Serverless 工作群組上都會停用區分大小寫。

若要啟用區分大小寫，請根據您的資料倉儲類型執行下列步驟：

- 佈建的叢集 – 若要在佈建的叢集上啟用區分大小寫，請建立已啟用 `enable_case_sensitive_identifier` 參數的自訂參數群組。接著，將該參數群組與叢集建立關聯。如需指示，請參閱[使用主控台管理參數群組](#)或[使用 AWS CLI 設定參數值](#)。

Note

在將自訂參數群組與叢集建立關聯之後，請記得重新啟動該叢集。

- 無伺服器工作群組 - 若要在 Redshift Serverless 工作群組上啟用區分大小寫，您必須使用 AWS CLI。Amazon Redshift 主控台目前不支援修改 Redshift Serverless 參數值。傳送下列[更新工作群組要求](#)：

```
aws redshift-serverless update-workgroup \  
  --workgroup-name target-workgroup \  
  --config-parameters  
  parameterKey=enable_case_sensitive_identifier,parameterValue=true
```

在修改工作群組的參數值之後，您不需要重新啟動該工作群組。

設定資料倉儲的授權

建立資料倉儲之後，您必須將來源 RDS 資料庫 Aurora 資料庫設定為授權的整合來源。如需指示，請參閱[設定 Amazon Redshift 資料倉儲的授權](#)。

後續步驟

透過來源 RDS 資料庫和 Amazon Redshift 目標資料倉儲，您現在可以建立零 ETL 整合並複寫資料。如需說明，請參閱 [the section called “建立零 ETL 整合”](#)。

建立與 Amazon Redshift 的 Amazon RDS 零 ETL 整合

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

建立 Amazon RDS 零 ETL 整合時，您必須指定來源 RDS 單一可用區或異地同步備份資料庫執行個體 Aurora 資料庫和目標 Amazon Redshift 資料倉儲。您也可以自訂加密設定和新增標籤。Amazon RDS 會在來源資料庫資料庫及其目標之間建立整合。整合啟用後，您插入來源資料庫資料庫的任何資料都會複寫到已設定的 Amazon Redshift 目標。

主題

- [必要條件](#)
- [所需的許可](#)
- [建立零 ETL 整合](#)
- [後續步驟](#)

必要條件

在建立零 ETL 整合之前，您必須先建立來源資料庫資料庫和目標 Amazon Redshift 資料倉儲。您也必須將資料庫資料庫新增為授權整合來源，以允許複寫至資料倉儲。

如需完成上述每個步驟的指示，請參閱 [the section called “開始使用零 ETL 整合”](#)。

所需的許可

建立零 ETL 整合需要特定 IAM 許可。您至少需要執行下列動作的許可：

- 為來源 RDS 資料庫 資料庫叢集建立零 ETL 整合。
- 檢視並刪除所有零 ETL 整合。
- 建立目標資料倉儲的傳入整合。如果相同帳戶擁有 Amazon Redshift 資料倉儲，且此帳戶是該資料倉儲的授權主體，則您不需要此許可。如需新增授權主體的相關資訊，請參閱 [設定 Amazon Redshift 資料倉儲的授權](#)。

下列範例政策示範建立和管理整合所需的 [最低權限許可](#)。如果您的使用者或角色擁有更廣泛的權限 (例如 AdministratorAccess 受管理原則)，您可能不需要這些確切的權限。

Note

Redshift Amazon Resource Name (ARN) 具有下列格式。請注意在無伺服器命名空間 UUID 之前使用正斜線 (/)，而不是冒號 (:)。

- 佈建的叢集 – `arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid`
- 無伺服器 – `arn:aws:redshift-serverless:{region}:{account-id}:namespace/namespace-uuid`

範例政策

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "rds:CreateIntegration"
    ],
    "Resource": [
      "arn:aws:rds:{region}:{account-id}:db:source-db",
      "arn:aws:rds:{region}:{account-id}:integration:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "rds:DescribeIntegrations"
    ],
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": [
      "rds>DeleteIntegration"
    ],
    "Resource": [
      "arn:aws:rds:{region}:{account-id}:integration:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "redshift:CreateInboundIntegration"
    ],
    "Resource": [
      "arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid"
    ]
  }
]}
```

在不同帳戶中選擇目標資料倉儲

如果您打算指定位於另一個目標 Amazon Redshift 資料倉儲 AWS 帳戶，則必須建立一個角色，以允許目前帳戶中的使用者存取目標帳戶中的資源。如需詳細資訊，請參閱在[您擁有的其他 AWS 帳戶 IAM 使用者中提供存取權限](#)。

角色必須具有下列許可，允許使用者檢視目標帳戶中可用的 Amazon Redshift 佈建叢集和 Redshift 無伺服器命名空間。

必要許可和信任政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:DescribeClusters",
        "redshift-serverless:ListNamespaces"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

角色必須具有下列信任策略，其可指定目標帳戶 ID。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{external-account-id}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需建立角色的指示，請參閱[使用自訂信任政策建立角色](#)。

建立零 ETL 整合

您可以使用 AWS Management Console、或 RDS API 建立。AWS CLI

依預設，RDS for MySQL 會立即清除二進位日誌檔。由於 Zero-ETL 整合依賴二進位記錄檔將資料從來源複寫到目標，因此來源資料庫的保留期間必須至少為一小時。建立整合後，Amazon RDS 會立即檢查所選來源資料庫的二進位日誌檔保留期。如果目前的值為 0 小時，Amazon RDS 會自動將其變更為 1 小時。否則，此值保持不變。

RDS 主控台

建立零 ETL 整合

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在左側導覽窗格中，選擇零 ETL 整合。
3. 選擇建立零 ETL 整合。
4. 對於整合識別符，輸入整合的名稱。此名稱最多可有 63 個英數字元，且可包含連字號。
5. 選擇下一步。
6. 對於來源，請選取資料來源的 RDS 資料庫。資料庫必須執行適 MySQL 15.4 和零 ETL Support)。

Note

RDS 會在資料庫參數未正確設定時通知您。如果您收到此訊息，您可以選擇為我修正或手動設定它們。如需手動修正它們的指示，請參閱 [the section called “步驟 1：建立自訂資料庫參數群組”](#)。

修改資料庫參數需要重新開機。建立整合之前，必須先完成重新開機，且新參數值必須順利套用至資料庫。

7. 成功設定來源資料庫資料庫之後，請選擇下一步。
8. 針對目標，執行下列動作：
 1. (選擇性) 若要針對 Amazon Redshift 目標使 AWS 帳戶用不同的帳戶，請選擇「指定其他帳戶」。然後，輸入 IAM 角色的 ARN，該角色擁有顯示您資料倉儲的許可。如需建立 IAM 角色的指示，請參閱 [the section called “在不同帳戶中選擇目標資料倉儲”](#)。

- 對於 Amazon Redshift 資料倉儲，請從來源資料庫資料庫中選取複寫資料的目標。您可以選擇佈建的 Amazon Redshift 叢集或 Redshift 無伺服器命名空間做為目標。

Note

如果未正確設定所指定資料倉儲的資源政策或區分大小寫設定，RDS 即會通知您。如果您收到此訊息，您可以選擇為我修正或手動設定它們。如需手動修正它們的指示，請參閱《Amazon Redshift 管理指南》中的[開啟資料倉儲的區分大小寫](#)和[設定資料倉儲的授權](#)。修改已佈建 Redshift 叢集的區分大小寫需要重新開機。在可以建立整合之前，必須先完成重新開機，且新參數值必須成功套用至叢集。

如果您選取的來源和目標位於不同的 AWS 帳戶中，則 Amazon RDS 無法為您修正這些設定。您必須導覽至其他帳戶，然後在 Amazon Redshift 中手動修正這些設定。

- 正確設定了目標資料倉儲後，即可選擇下一步。
- (選用) 對於標籤，將一或多個標籤新增至整合。如需詳細資訊，請參閱 [the section called “標記 RDS 資源”](#)。
- 對於加密，指定您想要加密整合的方式。根據預設，RDS 會將 AWS 擁有的金鑰若要改為選擇客戶受管金鑰，請啟用自訂加密設定，然後選擇要用於加密的 KMS 金鑰。如需詳細資訊，請參閱 [the section called “加密 Amazon RDS 資源”](#)。

Note

如果您指定自訂 KMS 金鑰，金鑰政策必須允許 Amazon Redshift 服務主體 (redshift.amazonaws.com) 的 kms:CreateGrant 動作。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[建立金鑰政策](#)。

可選擇性地新增加密內容。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的[加密內容](#)。

- 選擇下一步。
- 檢閱您的整合設定，然後選擇建立零 ETL 整合。

如果建立失敗，請參閱 [the section called “我無法建立零 ETL 整合”](#)，取得疑難排解步驟。

整合建立時的狀態為 Creating，而目標 Amazon Redshift 資料倉儲的狀態為 Modifying。在此期間，您無法查詢資料倉儲或對其進行任何組態變更。

成功建立整合時，整合和目標 Amazon Redshift 資料倉儲的狀態都會變更為 Active。

AWS CLI

若要使用建立零 ETL 整合 AWS CLI，請搭配下列選項使用 [建立整合](#) 指令：

- `--integration-name` - 指定整合的名稱。
- `--source-arn`— 指定 RDS 資料庫 Aurora 資料庫將成為整合的來源。
- `--target-arn` - 指定 Amazon Redshift 資料倉儲的 ARN，它將會成為整合目標。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-integration \  
  --integration-name my-integration \  
  --source-arn arn:aws:rds:{region}:{account-id}:my-db \  
  --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

在 Windows 中：

```
aws rds create-integration ^  
  --integration-name my-integration ^  
  --source-arn arn:aws:rds:{region}:{account-id}:my-db ^  
  --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

RDS API

若要使用 Amazon RDS API 建立零 ETL 整合，請搭配下列參數使用 [CreateIntegration](#) 操作：

- `IntegrationName` - 指定整合的名稱。
- `SourceArn`— 指定 RDS 單一可用區或異地同步備份資料庫執行個體 Aurora 資料庫將成為整合的來源。
- `TargetArn` - 指定將成為整合目標之 Amazon Redshift 資料倉儲的 ARN。

後續步驟

在成功建立零 ETL 整合之後，您必須在目標 Amazon Redshift 叢集或工作群組內建立目的地資料庫。然後，您就可以開始將資料新增到來源 RDS 資料庫，並在 Amazon Redshift 中進行查詢。如需指示，請參閱[在 Amazon Redshift 中建立目的地資料庫](#)。

將資料新增至來源 RDS 資料庫，並在 Amazon Redshift 中進行查詢

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱[AWS 服務條款](#)中的 Beta 版和預覽版。

若要完成建立零 ETL 整合，將資料從 Amazon RDS 複製到 Amazon Redshift，您必須在 Amazon Redshift 中建立目的地資料庫。

首先，連線到您的 Amazon Redshift 叢集或工作群組，並建立參考整合識別碼的資料庫。然後，您可以將資料新增到來源 RDS 資料庫 Aurora 資料庫，並在 Amazon Redshift 中查看已複寫的資料。

主題

- [在 Amazon Redshift 中建立目的地資料庫](#)
- [將資料新增至來源資料庫資料](#)
- [在 Amazon 紅移中查詢您的亞馬遜 RDS 數據](#)
- [RDS 與 Amazon Redshift 資料庫之間的資料類型差異](#)

在 Amazon Redshift 中建立目的地資料庫

在建立整合之後，您必須在目標資料倉儲中建立目的地資料庫，然後才能開始將資料複寫到 Amazon Redshift。此目的地資料庫必須包含整合識別符的參考。您可以使用 Amazon Redshift 主控台或查詢編輯器第 2 版來建立資料庫。

如需建立目的地資料庫的指示，請參閱[在 Amazon Redshift 中建立目的地資料庫](#)。

將資料新增至來源資料庫資料

設定整合之後，您可以將一些資料新增至要複寫到 Amazon Redshift 資料倉儲的 RDS 資料庫。

Note

Amazon RDS 與 Amazon Redshift 中的資料類型之間存在差異。如需資料類型映射的資料表，請參閱 [the section called “資料類型差異”](#)。

首先，使用您選擇的 MySQL。如需說明，請參閱 [the section called “連接至執行 MySQL 的資料庫執行個體”](#)。

然後，建立資料表並插入一系列範例資料。

Important

請確定資料表具有主索引鍵。否則，無法將其複寫到目標資料倉儲。

下列範例使用 [MySQL 工作台公用程式](#)。

```
CREATE DATABASE my_db;  
  
USE my_db;  
  
CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author  
  VARCHAR(50) NOT NULL,  
  Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));  
  
INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977, 'Supernatural  
  fiction');
```

在 Amazon 紅移中查詢您的亞馬遜 RDS 數據

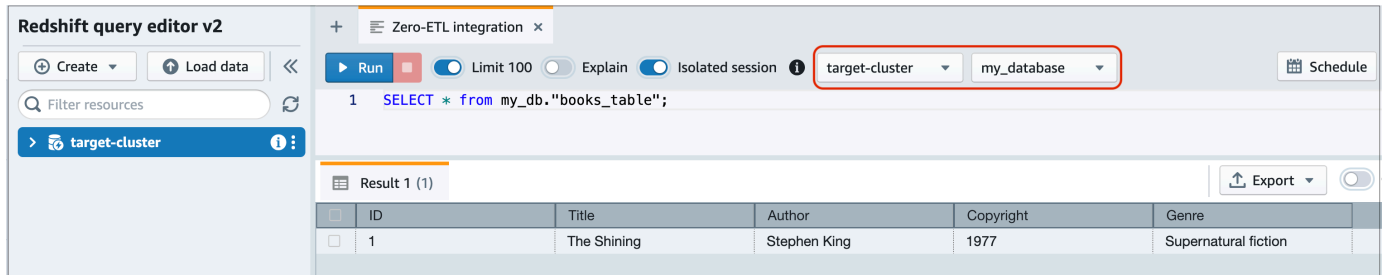
將資料新增至 RDS 資料庫之後，資料就會複寫到 Amazon Redshift，並可供查詢。

查詢複製的資料

1. 導覽至 Amazon Redshift 主控台，然後從左側導覽窗格中選擇查詢編輯器第 2 版。
2. 連線到您的叢集或工作群組，然後從下拉式功能表中選擇您已從整合中建立的目的地資料庫 (在此範例中為 `destination_database`)。如需建立目的地資料庫的指示，請參閱 [在 Amazon Redshift 中建立目的地資料庫](#)。

3. 使用 SELECT 陳述式來查詢您的資料。在此範例中，您可以執行下列命令，從您在來源 RDS 資料庫中建立的表格中選取所有資料：

```
SELECT * from my_db."books_table";
```



- *my_db* 是 RDS 資料庫結構描述名稱。
- *books_table* 是 RDS 資料表名稱。

您也可以使用命令列用戶端查詢資料。例如：

```
destination_database=# select * from my_db."books_table";
```

```

ID |          Title |          Author |      Copyright |          Genre | txn_seq |
txn_id
-----+-----+-----+-----+-----+-----+-----
+-----+
  1 | The Shining | Stephen King |      1977 | Supernatural fiction |      2 |
12192

```

Note

如需區分大小寫，請針對結構描述、資料表和資料欄名稱使用雙引號 (" ")。如需詳細資訊，請參閱 [enable_case_sensitive_identifier](#)。

RDS 與 Amazon Redshift 資料庫之間的資料類型差異

下表顯示適用於 MySQL 的 RDS 表格的對應顯示 Aurora MySQL SQL 應的 Amazon Redshift 資料類型的對應。Amazon RDS Amazon 目前僅支援這些資料類型進行零 ETL 整合。

如果來源資料庫資料庫中的資料表包含不受支援的資料類型，表格會不同步，而且 Amazon Redshift 目標無法使用該資料表。從來源到目標的串流會繼續進行，但是無法使用其中資料類型不受支援的資料表。若要修正資料表並使其可在 Amazon Redshift 中使用，您必須手動還原重大變更，然後執行 [ALTER DATABASE...INTEGRATION REFRESH](#) 以重新整理整合。

MySQL 版

RDS for MySQL 資料類型	Amazon Redshift 資料類型	描述	限制
INT	INTEGER	帶正負號的 4 位元組整數	
SMALLINT	SMALLINT	帶正負號的 2 位元組整數	
TINYINT	SMALLINT	帶正負號的 2 位元組整數	
MEDIUMINT	INTEGER	帶正負號的 4 位元組整數	
BIGINT	BIGINT	帶正負號的 8 位元組整數	
INT UNSIGNED	BIGINT	帶正負號的 8 位元組整數	
TINYINT UNSIGNED	SMALLINT	帶正負號的 2 位元組整數	
MEDIUMINT UNSIGNED	INTEGER	帶正負號的 4 位元組整數	
BIGINT UNSIGNED	DECIMAL(20,0)	可選擇精確度 (有效位數) 的精確數值	

RDS for MySQL 資料類型	Amazon Redshift 資料類型	描述	限制
十進制 (p , s) = 數字 (p , s)	DECIMAL(p,s)	可選擇精確度 (有效位數) 的精確數值	不支援大於 38 且大於 37 的精確度
十進制 (P , s) 無符號 = 數字 (p , s) 無符號	DECIMAL(p,s)	可選擇精確度 (有效位數) 的精確數值	不支援大於 38 且大於 37 的精確度
FLOAT4/REAL	REAL	單精度浮點數	
FLOAT4/REAL UNSIGNED	REAL	單精度浮點數	
DOUBLE/REAL/FLOAT8	DOUBLE PRECISION	雙精度浮點數	
DOUBLE/REAL/FLOAT8 UNSIGNED	DOUBLE PRECISION	雙精度浮點數	
位元 (n)	VARBYTE(8)	可變長度二進位值	
BINARY(n)	瓦字節 (n)	可變長度二進位值	
VARBINARY(n)	瓦字節 (n)	可變長度二進位值	
CHAR(n)	VARCHAR(n)	可變長度字串值	
VARCHAR(n)	VARCHAR(n)	可變長度字串值	
TEXT	VARCHAR(65535)	可變長度字串值 最多 65535 個位元組	
TINYTEXT	VARCHAR(255)	可變長度字串值 最多 255 個位元組	

RDS for MySQL 資料類型	Amazon Redshift 資料類型	描述	限制
ENUM	VARCHAR(1020)	可變長度字串值 最多 1020 個位元組	
SET	VARCHAR(1020)	可變長度字串值 最多 1020 個位元組	
DATE	DATE	日曆日期 (年、月、日)	
DATETIME	TIMESTAMP	日期和時間 (未使用時區)	
TIMESTAMP(p)	TIMESTAMP	日期和時間 (未使用時區)	
TIME	VARCHAR(18)	可變長度字串值 最多 18 個位元組	
YEAR	VARCHAR(4)	可變長度字串值 最多 4 個位元組	
JSON	SUPER	半結構化資料或文件作為值	

檢視和監控與 Amazon Redshift 的 Amazon RDS 零 ETL 整合

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

您可以檢視 Amazon RDS 零 ETL 整合的詳細資訊，以查看其組態資訊和目前狀態。您也可以透過在 Amazon Redshift 中查詢特定系統檢視來監控整合的狀態。此外，Amazon Redshift 將某些與整合相關的指標發佈到 Amazon CloudWatch，您可以在亞 Amazon Redshift 主控台中查看這些指標。

主題

- [檢視整合](#)
- [使用系統資料表監控整合](#)
- [監控與 Amazon 的集成 EventBridge](#)

檢視整合

您可以使用 AWS Management Console、或 RDS API 檢視 Amazon RDS 零 ETL 與亞馬遜紅移整合。AWS CLI

主控台

檢視零 ETL 整合的詳細資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從左側導覽窗格中，選擇零 ETL 整合。
3. 選取整合以檢視其詳細資訊，例如其來源資料庫資料庫和目標資料倉儲。

The screenshot displays the AWS Management Console interface for a Zero-ETL integration. The breadcrumb navigation shows 'RDS > Zero-ETL integrations > my-integration'. The main heading is 'my-integration', with two buttons: 'View CloudWatch metrics for source DB' and 'Delete'. Below this is a section titled 'Zero-ETL integration details' containing three columns of information:

General settings	Source	Destination
Integration name my-integration	Source type RDS for MySQL	Destination type Redshift provisioned cluster
Date created Sept 28, 2024, 04:30:00 (UTC-07:00)	DB identifier source-instance	Data warehouse 670a7cf1-f27a-4596-aede-935ad771378f
Integration ARN arn:aws:rds:us-east-1:123456789012:integration:264853b4-2571-44c5-b45d-08633fc5c688	Source ARN arn:aws:rds:us-east-1:123456789012:db:source-instance	Destination ARN arn:aws:redshift:us-east-1:123456789012:namespace:670a7cf1-f27a-4596-aede-935ad771378f
Status Active		

整合可以具有下列狀態：

- **Creating** – 正在建立整合。
- **Active** – 整合正在將交易資料傳送至目標資料倉儲。
- **Syncing** – 整合遇到可復原的錯誤，且正在重新植入資料。在完成重新同步之前，受影響的資料表無法在 Amazon Redshift 中進行查詢。
- **Needs attention** – 整合發生事件或錯誤，需要手動介入才能解決此問題。若要修正問題，請遵循整合詳細資訊頁面上錯誤訊息中的指示。
- **Failed** – 整合發生無法復原的事件或無法修正的錯誤。您必須刪除並重新建立整合。
- **Deleting** – 正在刪除整合。

AWS CLI

若要使用檢視目前帳戶中的所有零 ETL 整合 AWS CLI，請使用 [描述整合](#) 指令並指定選項。--integration-identifier

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-integrations \  
  --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

在 Windows 中：

```
aws rds describe-integrations ^  
  --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

RDS API

若要使用 Amazon RDS API 檢視零 ETL 整合，請搭配 IntegrationIdentifier 參數使用 [DescribeIntegrations](#) 操作：

使用系統資料表監控整合

Amazon Redshift 具有系統資料表和檢視，其中包含系統如何運作的相關資訊。您可以使用查詢任何其他資料庫資料表的方式，來查詢這些系統資料表和檢視。如需 Amazon Redshift 中系統資料表和檢視的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員》指南中的 [系統資料表參考](#)。

您可以查詢下列系統檢視和表格，以取得有關 零 ETL 與 Amazon Redshift 整合的相關資訊：

- [SVV_INTEGRATION](#) – 提供整合的組態詳細資料。
- [SVV_INTEGRATION_TABLE_STATE](#) – 描述整合內每個資料表的狀態。
- [SYS_INTEGRATION_TABLE_STATE_CHANGE](#) - 顯示整合的資料表狀態變更日誌。
- [SYS_INTEGRATION_ACTIVITY](#) – 提供已完成整合執行的相關資訊。

所有與整合相關的 Amazon CloudWatch 指標均來自 Amazon Redshift。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的[監控零 ETL 整合](#)。目前，Amazon RDS Amazon 不會將任何整合指標發佈到 CloudWatch。

監控與 Amazon 的集成 EventBridge

Amazon Redshift 將與整合相關的事件發送到 Amazon EventBridge。如需事件及其對應事件 ID 的清單，請參閱 [Amazon Redshift 管理指南 EventBridge 中的零 ETL 整合事件通知](#)。

刪除與 Amazon Redshift 的 Amazon RDS 零 ETL 整合

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

當您刪除零 ETL 整合時，Amazon RDS Amazon 資料庫叢集中移除該整合。您的交易資料不會從 Amazon RDS 或 Amazon Redshift 中刪除，但 Amazon RDS 不會將新資料傳送到 Amazon Redshift。

只有當整合狀態為、或時 ActiveFailed, Syncing 您才能刪除整合 Needs attention。

您可以使用 AWS Management Console、或 RDS API 刪除零 ETL 整合。AWS CLI

主控台

刪除零 ETL 整合

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從左側導覽窗格中，選擇零 ETL 整合。

3. 選取您要刪除的零 ETL 整合。
4. 選擇動作、刪除，並確認刪除。

AWS CLI

若要刪除零 ETL 整合，請使用 [delete-integration](#) 命令並指定 `--integration-identifier` 選項。

Example

對於LinuxmacOS、或Unix：

```
aws rds delete-integration \  
  --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

在 Windows 中：

```
aws rds delete-integration ^  
  --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

RDS API

若要使用 Amazon RDS API 刪除零 ETL 整合，請搭配 `IntegrationIdentifier` 參數使用 [DeleteIntegration](#) 操作：

對與 Amazon Redshift 的 Amazon RDS 零 ETL 整合進行疑難排解

這是與 Amazon Redshift 的 Amazon RDS 零 ETL 整合 (預覽版本) 的發行前版本文件。文件和功能會隨時變更。我們建議僅在測試環境中使用此功能，不要在生產環境中使用。如需預覽版條款和條件，請參閱 [AWS 服務條款](#) 中的 Beta 版和預覽版。

您可以在 Amazon Redshift 中查詢 [SVV_INTEGRATION](#) 系統資料表，以檢查零 ETL 整合的狀態。如果 `state` 資料欄具有 `ErrorState` 的值，表示有問題。如需詳細資訊，請參閱 [the section called “使用系統資料表進行監視”](#)。

使用下列資訊，對與 Amazon Redshift 的 Amazon RDS 零 ETL 整合的常見問題進行疑難排解。

主題

- [我無法建立零 ETL 整合](#)
- [我的整合停留在一種狀態 Syncing](#)
- [我的表沒有複製到 Amazon Redshift](#)
- [我的一個或多個 Amazon Redshift 資料表需要重新同步](#)

我無法建立零 ETL 整合

如果您無法建立零 ETL 整合，請確定下列情況對於您的來源資料庫執行個體是正確的：

- 您的來源資料庫資料庫正在執行適用於 MySQL 的 RDS 8.0.32 版或更高版本，SQL 15.4 和零 ETL Support)。若要驗證引擎版本，請選擇資料庫資料庫的 [組態] 索引標籤，然後檢查 Engine 版本。
- 您已正確設定資料庫參數。如果必要參數設定不正確或未與資料庫執行個體相關聯，則建立會失敗。請參閱[the section called “步驟 1：建立自訂資料庫參數群組。”](#)。

此外，請確定下列情況對於您的目標資料倉儲是正確的：

- 已啟用區分大小寫。請參閱[開啟資料倉儲的區分大小寫](#)。
- 您已新增正確的授權主體和整合來源。請參閱[為您的 Amazon Redshift 資料倉儲設定授權](#)。
- 資料倉儲已加密 (如果是佈建的叢集)。請參閱 [Amazon Redshift 資料庫加密](#)。

我的整合停留在一種狀態 Syncing

Syncing如果您變更其中一個所需資料庫參數的值，您的整合可能會持續顯示狀態。

若要修正此問題，請檢查與來源資料庫資料庫相關聯之參數群組中的參數值，並確定它們符合所需的值。如需詳細資訊，請參閱 [the section called “步驟 1：建立自訂資料庫參數群組。”](#)。

如果您修改任何參數，請務必重新啟動資料庫資料庫以套用變更。

我的表沒有複製到 Amazon Redshift

您的資料可能無法複製，因為一或多個來源資料表沒有主索引鍵。Amazon Redshift 中的監控儀表板會將這些表格的狀態顯示為Failed，且整體零 ETL 整合的狀態會變更為 Needs attention

若要解決這個問題，您可以識別資料表中可以成為主索引鍵的現有索引鍵，或者您可以新增合成的主索引鍵。如需詳細解決方案，請參閱[使用亞馬遜 Amazon Redshift 建立 Amazon Aurora MySQL 或 Amazon RDS for MySQL 零 ETL 整合時處理沒有主鍵的表格](#)。

我的一個或多個 Amazon Redshift 資料表需要重新同步

在來源資料庫執行個體上執行某些命令，可能需要重新同步您的資料表。在這些情況下，[SVV_INTEGRATION_TABLE_STATE](#) 系統檢視會顯示 ResyncRequired 的 table_state，這表示整合必須將該特定資料表的資料從 MySQL 完全重新載入至 Amazon Redshift。

當資料表開始重新同步時，其會進入 Syncing 的狀態。您不需要採取任何手動動作，即可重新同步資料表。資料表資料正在重新同步處理時，您無法在 Amazon Redshift 中存取該資料。

以下是一些可以將資料表置於 ResyncRequired 狀態的範例操作，以及可以考慮的替代方法。

作業	範例	備用
將資料欄新增到特定位置	<pre>ALTER TABLE <i>table_name</i> ADD COLUMN <i>column_name</i> INTEGER NOT NULL first;</pre>	<p>Amazon Redshift 不支援使用 first 或 after 關鍵字將資料欄新增到特定位置。如果目標資料表中的資料欄順序並不重要，請使用更簡單的命令，將資料欄新增至資料表的尾端：</p> <pre>ALTER TABLE <i>table_name</i> ADD COLUMN <i>column_name</i> <i>column_type</i> ;</pre>
新增具有預設 CURRENT_T	<pre>ALTER TABLE <i>table_name</i> ADD COLUMN <i>column_name</i> TIMESTAMP</pre>	現有資料表資料列的

作業	範例	備用
IMESTAMP 的時間戳記資料欄	<pre>NOT NULL DEFAULT CURRENT_TIMESTAMP;</pre>	<p>CURRENT_TIMESTAMP 值由 RDS for MySQL 計算，如果沒完整的資料表資料重新同步，就無法在 Amazon Redshift 中進行模擬。</p> <p>如果可能，請將預設值切換為 2023-01-01 00:00:15 之類的常值常數，避免資料表可用性中的延遲。</p>
在單一命令內執行多個資料欄操作	<pre>ALTER TABLE <i>table_name</i> ADD COLUMN <i>column_1</i>, RENAME COLUMN <i>column_2</i> TO <i>column_3</i>;</pre>	<p>考慮將命令分成兩個單獨的操作 (ADD 和 RENAME)，這不需要重新同步。</p>

適用於 Db2 的 Amazon RDS

Amazon RDS 支援執行下列版本的資料庫執行個體 IBM Db2：

- Db2 Standard Edition
- Db2 Advanced Edition

Amazon RDS 支援執行下列 Db2 版本的資料庫執行個體：

- DB2 11.5

如需次要版本的支援詳情，請參閱[Amazon RDS 版本上的 Db2](#)。

在建立資料庫執行個體之前，請先完成本使用指南—[設定 Amazon RDS](#)節中的步驟。當您使用主要使用者建立資料庫執行個體時，使用者會取得 DBADM 授權，但有一些限制。使用此使用者執行管理工作，例如建立其他資料庫帳戶。您無法使用 SYSADM、SYSCTRL、SYSMAINT 執行個體層級授權或 SECADM 資料庫層級授權單位。

您可以建立下列作業：

- 資料庫執行個體
- 資料庫快照
- Point-in-time 還原
- 自動化儲存備份
- 手動儲存備份

您可以在虛擬私有雲 (VPC) 內使用執行 Db2 的資料庫執行個體。您也可以啟用各種選項，將功能新增至 Amazon RDS for Db2 資料庫執行個體。Amazon RDS 支援 RDS for Db2 的異地同步備份部署作為高可用性的容錯移轉解決方案。

Important

為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。它也會限制對需要提升權限的特定系統程序和表格的存取。您可以使用標準 SQL 用戶端存取資料庫，例如 IBM Db2 CLP。不過，您無法使用 Telnet 或 Secure Shell (SSH) 直接存取主機。

主題

- [Db2 在 Amazon RDS 上概述](#)
- [建立適用於 Db2 資料庫執行個體的 Amazon RDS 的先決條件](#)
- [連接到您的 Amazon RDS 用 Db2 資料庫執行個體](#)
- [保護 Amazon RDS for Db2 資料庫執行個體連線](#)
- [管理適用於 Db2 資料庫執行個體的 Amazon RDS](#)
- [將 Amazon RDS for Db2 資料庫執行個體與 Amazon S3 整合](#)
- [在 Amazon RDS 上將數據遷移到 Db2](#)
- [適用於 Db2 資料庫執行個體的 Amazon RDS 選項](#)
- [適用於 Db2 的 Amazon RDS 的外部存儲程序](#)
- [適用於 Db2 的 Amazon RDS 的已知問題和限制](#)
- [適用於 Db2 的 Amazon RDS 存儲過程參考](#)
- [用於 Db2 的 Amazon RDS 用戶定義函數參考](#)

Db2 在 Amazon RDS 上概述

您可以閱讀以下各節以取得 Db2 在 Amazon RDS 上的概觀。


主題

- [適用於 Db2 的 Amazon RDS 功能](#)
- [Amazon RDS 版本上的 Db2](#)
- [Amazon 適用於 Db2 的授權選項](#)
- [Amazon RDS 適用於 Db2 執行個體類別](#)
- [適用於 Db2 的 Amazon RDS 參數](#)
- [Amazon RDS 上 Db2 資料庫的 EBCDC 定序](#)
- [適用於 Db2 資料庫執行個體的 Amazon RDS 的當地時區](#)

適用於 Db2 的 Amazon RDS 功能

適用於 Db2 的 Amazon RDS 支援 IBM Db2 資料庫的大部分功能和功能。部分功能的支援或權限可能有所限制。[如需特定 Db2 版本之 Db2 資料庫功能的詳細資訊，請參閱文件 IBM Db2。](#)

您可以在[資料庫的最新資訊](#)頁面上篩選新的 Amazon RDS 功能。對於 Products (產品)，請選擇 Amazon RDS。然後，您可以使用關鍵字進行搜索，例如**Db2 2023**。

 Note

以下列表並不詳盡。

主題

- [適用於 Db2 的 RDS 中支援的功能](#)
- [適用於 Db2 的 RDS 中不支援的功能](#)

適用於 Db2 的 RDS 中支援的功能

RDS for Db2 支援的功能包括原生功能以 IBM Db2 及 Amazon RDS 核心的功能。

原生的功能 IBM Db2

適用於 Db2 的 RDS 支援下列 Db2 資料庫功能：

- 建立使用客戶定義的程式碼集、定序、頁面大小和區域的標準資料庫。使用 Amazon RDS [rdsadmin.create_database](#) 存儲程序。
- 新增、刪除或修改本機使用者和群組。使用的 Amazon RDS 存儲程序 [授與和撤銷權限](#)。
- 使用 Amazon RDS [rdsadmin.create_role](#) 存放程序建立角色。
- Support 標準列組織表格。
- Support 資料行組織表格的分析工作負載。
- 能夠定義 DB2 相容性功能，Oracle 例如和。MySQL
- Support Java 基於外部預存程序。
- 使用 SSL/TLS Support 傳輸中的資料加密。
- 監視資料庫的狀態 (ALIVEDOWN、STORAGE_FULL、UNKNOWN、和 STANDBY_CONNECTABLE)。
- 還原客戶提供的離線或線上 Linux (LE) 資料庫。使用 Amazon RDS 存儲程序的 [管理資料庫](#)。
- 套用客戶提供的 Db2 封存記錄，讓資料庫與自我管理的 Db2 資料庫保持同步。使用 Amazon RDS 存儲程序的 [管理資料庫](#)。
- Support Db2 執行個體層級和資料庫層級稽核。
- Support 同質聯合。

- 能夠從亞馬遜簡單儲存服務 (Amazon S3) 中的數據文件加載表。
- 授與使用者、群組或角色的授權 CONNECTSYSMON , 例如 ACCESSCTRLDATAACCESS、SQLADM、WLMADM、EXPLAINLOAD、或 IMPLICIT_SCHEMA

Amazon RDS 的核心功能

適用於 Db2 的 RDS 支援下列核心 Amazon RDS 功能：

- 要指派給資料庫執行個體的自訂參數群組。
- 建立、修改和刪除資料庫執行個體。
- 還原自我管理的 Db2 離線或線上 Linux (LE) 資料庫備份。

Note

若要還原備份，請勿在建立資料庫執行個體時提供資料庫名稱。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

- Support gp3、io2 和 io1 儲存區類型。
- AWS Managed Microsoft AD 針對 Db2 的 RDS 進行 Kerberos 驗證和 LDAP 群組授權的使用。
- 修改現有 Db2 執行個體的安全群組、連接埠、執行個體類型、儲存體、備份保留期及其他設定。
- 資料庫執行個體的刪除保護。
- 跨區域 point-in-time 復原 (PITR)。
- 使用 AWS Key Management Service (AWS KMS) 進行儲存加密和靜態加密。
- 異地同步備份資料庫執行個體，具備一個高可用性的
- 重新啟動資料庫執行個體。
- 主密碼的更新。
- 將資料庫執行個體還原到特定時間。
- 使用儲存層級 Backup 來備份和還原資料庫執行個體。
- 啟動和停止資料庫執行個體。
- 資料庫執行個體的維護。

適用於 Db2 的 RDS 中不支援的功能

適用於 Db2 的 RDS 不支援下列 Db2 資料庫功能：

- SYSADMSECADM、和主要使用者的SYSMAINT存取權。
- 以 C、C++ 或 Cobol 撰寫的外部預存程序。
- 單一主機上的多個 Db2 資料庫執行個體。
- 單一 RDS for Db2 資料庫執行個體上的多個 Db2 資料庫。
- 用於驗證的外部 GSS-API 插件。
- 備份或還原 Db2 資料庫的外部協力廠商外掛程式。
- 多節點大規模 parallel 處理 (MPP) ，例如. IBM Db2 Warehouse
- IBM Db2 pureScale.
- 高可用性災難復原 (HADR)。
- 原生資料庫加密。
- Db2 的異質聯合會。
- 用於加密備份的跨區域 point-in-time-recovery (PITR)。
- 創建非圍欄例程。如需詳細資訊，請參閱 [非圍欄例程](#)。
- 建立新的非自動儲存體表格空間。如需詳細資訊，請參閱 [移轉期間的非自動儲存體表格空間](#)

Amazon RDS 版本上的 Db2

對於 Db2，版本號碼採用主要. 次要修訂的形式，例如，11.5.9.0.sb0000.r 1。我們的版本實現與 Db2 的版本實現匹配。

重大的

主要版本號碼既是整數，也是版本號碼的第一個小數部分，例如 11.5。如果主要版本號碼發生變更，版本變更就會被視為主要變更 (例如，從 11.5 版變更到 12.1 版)。

未成年

次要版本號碼同時是版本號碼的第三部分和第四部分，例如，11.5.9.0 中的 9.0。第三部分表示 Db2 包，例如 9.0 中的 9。第四部分表示 Db2 修正套件，例如 9.0 中的 0。如果 Db2 modpack 或 Db2 修正套件變更，則版本變更會被視為次要變更 (例如，從 11.5.9.0 版變更至 11.5.9.1 版，或從 11.5.9.0 變更至 11.5.10.0)，但提供目錄資料表更新的例外情況除外。(Amazon RDS 會處理這些例外情況。)

建立

組建編號是版本號碼的第五部分，例如，在 11.5.9.0.sb00000 中。編號部分全部為零的內部版本號表示標準組建。編號部分不是全部為零的內部版本號表示特殊的構建。如果現有 Db2 版本的安全性

修正程式或特殊組建，組建編號會變更。組建編號變更也表示 Amazon RDS 自動套用新的次要版本。

修訂版

版本號碼是版本號碼的第六部分，例如，在 11.5.9.0.sb00000.r1 中的 r1。修訂版本是對現有 Db2 發行版本的 Amazon RDS 修訂版本。修訂編號變更表示 Amazon RDS 會自動套用新的次要版本。

主題

- [在 Amazon RDS 上支援的 Db2 次要版本](#)
- [在 Amazon RDS 上支持的 Db2 主要版本](#)

在 Amazon RDS 上支援的 Db2 次要版本

下表顯示 Amazon RDS 目前支援的次要版本 Db2。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。

Db2 引擎版本	IBM發行日期	RDS 發佈日期	RDS 標準支援結束日期
11.5			
11.5.9.0	2023 年 11 月 15 日	2023 年十一月二十七	

您可以在建立新的資料庫執行個體時指定任何目前支援的 Db2 版本。您可以為指定的主要版本指定主要版本 (例如 Db2 11.5) 和任何支援的次要版本。若未指定版本，Amazon RDS 會預設使用支援的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 預設會使用您指定之主要版本的最新發行版本。若要查看支援的版本清單以及新建立的資料庫執行個體的預設值，請使用 [describe-db-engine-versions](#) AWS Command Line Interface (AWS CLI) 命令。

例如，若要列出適用於 Db2 的 Amazon RDS 引擎版本，請執行下列 AWS CLI 命令。將##取代為您的 AWS 區域。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --filters Name=engine,Values=db2-ae,db2-se \  
  --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion, \  
DBParameterGroupFamily:DBParameterGroupFamily}" \  
  --region region
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^ \  
  --filters Name=engine,Values=db2-ae,db2-se ^ \  
  --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion, \  
DBParameterGroupFamily:DBParameterGroupFamily}" ^ \  
  --region region
```

此命令會產生類似下列範例的輸出：

```
[  
  {  
    "Engine": "db2-ae",  
    "EngineVersion": "11.5.9.0.sb00000000.r1",  
    "DBParameterGroupFamily": "db2-ae-11.5"  
  },  
  {  
    "Engine": "db2-se",  
    "EngineVersion": "11.5.9.0.sb00000000.r1",  
    "DBParameterGroupFamily": "db2-se-11.5"  
  }  
]
```

預設的 Db2 版本可能因而異。AWS 區域若要建立具有特定次要版本的資料庫執行個體，請在建立資料庫執行個體期間指定次要版本。您可以執行 `describe-db-engine-versions` 指令來決定 AWS 區域 for db2-ae 和 db2-se 資料庫引擎的預設版本。下列範例會傳回美國東部 (維吉尼亞北部) 的預設版本。db2-ae

對於 Linux macOS、或 Unix：

```
aws rds describe-db-engine-versions \  
  --default-only --engine db2-ae \  
  --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion, \  
DBParameterGroupFamily:DBParameterGroupFamily}" \  
  --region region
```

```
--region us-east-1
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --default-only --engine db2-ae ^  
  --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,  
  DBParameterGroupFamily:DBParameterGroupFamily}" ^  
  --region us-east-1
```

此命令會產生類似下列範例的輸出：

```
[  
  {  
    "Engine": "db2-ae",  
    "EngineVersion": "11.5.9.0.sb00000000.r1",  
    "DBParameterGroupFamily": "db2-ae-11.5"  
  }  
]
```

使用 Amazon RDS，您可以控制將 Db2 執行個體升級到 Amazon RDS 支援的新主要版本的時機。您可以維持與特定 Db2 版本的相容性、在生產環境中部署之前測試應用程式的新版本，以及在最符合您排程的時間執行主要版本升級。

啟用自動次要版本升級時，Amazon RDS 會自動將您的資料庫執行個體升級到新的 Db2 次要版本，因為它們受到 Amazon RDS 的支援。此修補會在您排程的維護時段發生。您可以修改資料庫執行個體，以啟用或停用自動次要版本升級。

除了 Db2 版本 11.5.9.1 和 11.5.10.0 之外，自動升級至新的 Db2 次要版本包括自動升級至新的組建和修訂版本。對於 11.5.9.1 和 11.5.10.0，請手動升級次要版本。

若您選擇不要自動排程升級，可以按照更新主要版本的相同程序，手動升級至支援的次要版本。如需相關資訊，請參閱「[升級資料庫執行個體引擎版本](#)」。

在 Amazon RDS 上支持的 Db2 主要版本

RDS for Db2 主要版本至少可在對應 IBM 版本的支援 IBM 結束 (基本) 之前，在標準支援下使用。下表顯示可用來規劃測試和升級週期的日期。如果 Amazon 將 RDS for Db2 版本的支援延長時間超過原始說明的時間，我們計劃更新此表格以反映之後的日期。

您可以使用下列日期來規劃測試和升級週期。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。

Db2 主要版本	IBM發行日期	RDS 發佈日期	IBM支撐結束 (基準)	IBM支援結束 (延伸)	RDS 標準支援結束日期
DB2 11.5	2019 年 6 月 27 日	2023 年十一月二十七	二零二五年九月	終止支援後 4 年	

Amazon 適用於 Db2 的授權選項

Amazon RDS for Db2 有兩種授權選項：使用您自有的授權 (BYOL) 和 Db2 授權透過。AWS Marketplace

主題

- [為 Db2 帶來您自己的授權](#)
- [通過 Db2 許可證 AWS Marketplace](#)
- [在 Db2 授權之間切換](#)

為 Db2 帶來您自己的授權

在 BYOL 模型中，您可以使用現有的 Db2 資料庫授權在 Amazon RDS 上部署資料庫。確認您具有要執行的資料庫執行個體類別和 Db2 資料庫版本的適當 Db2 資料庫授權。您也必須遵循在雲端運算環境中授權 IBM 資料庫軟體的政策。

Note

異地同步備份資料庫執行個體為冷待機狀態，因為 Db2 資料庫已安裝但未執行。待機型無法讀取、執行中或提供要求。如需詳細資訊，請參閱 IBM 網站上的 [IBM Db2 授權資訊](#)。

在此模型中，您會繼續使用您的使用中 IBM 支援帳戶，並 IBM 直接連絡 Db2 資料庫服務要求。如果您擁有提供案例支援的 AWS Support 帳戶，您可以聯絡以 AWS Support 取得 Amazon RDS 問題。Amazon Web Services 並針對 IBM 對需要兩個組織協助的案例提供多廠商支援程序。

Amazon RDS 支援的和的 BYOL 模型。Db2 Standard Edition Db2 Advanced Edition

主題

- [IBMDB2 使用您自己的授權的 ID](#)
- [將IBM識別碼新增至 Db2 資料庫執行個體之 RDS 的參數群組](#)
- [與整合 AWS License Manager](#)

IBMDB2 使用您自己的授權的 ID

在 BYOL 模型中，您需要 IBM Customer ID 和您 IBM Site ID 的建立、修改或還原 Db2 資料庫執行個體的 RDS。在建立 RDS for Db2 資料庫執行個體 IBM Site ID 之前 IBM Customer ID，您必須先使用您的和您的建立自訂參數群組。如需詳細資訊，請參閱 [將IBM識別碼新增至 Db2 資料庫執行個體之 RDS 的參數群組](#)。您可以使用不同 IBM Customer IDs 且 IBM Site IDs 在相同的 AWS 帳戶 或 AWS 區域中執行多個 RDS for Db2 資料庫執行個體。

Important

如果您是現有 IBM Db2 客戶，您可以 IBM Site ID 在「權利證明」憑證中找到您 IBM Customer ID 和您的「」IBM。如需詳細資訊，請參閱 [有關如何在 IBM 網站 IBM Site ID 上檢視您的 IBM Customer ID 和的指示](#)。

如果您是新的 IBM Db2 客戶，則必須先從 [IBM](#) 購買 Db2 軟體授權。購買 Db2 軟體授權後，您將收到授權證明 IBM，其中會列出您 IBM Customer ID 和您 IBM Site ID 的。

如果我們無法透過您 IBM Customer ID 和您的授權驗證 IBM Site ID，我們可能會終止任何使用這些未驗證授權執行的資料庫執行個體。

將 IBM 識別碼新增至 Db2 資料庫執行個體之 RDS 的參數群組

由於您無法修改預設參數群組，因此必須建立自訂參數群組，然後修改它以包含您 IBM Customer ID 和您的 IBM Site ID。如需參數群組的詳細資訊，請參閱 [在資料庫執行個體中使用資料庫參數群組](#)。

Important

在建立 RDS for Db2 資料庫執行個體 IBM Site ID 之前 IBM Customer ID，您必須先使用您的和您的建立自訂參數群組。

請使用下表中的參數設定。

參數	Value
<code>rds.ibm_customer_id</code>	<your IBM Customer ID>
<code>rds.ibm_site_id</code>	<your IBM Site ID>
<code>ApplyMethod</code>	<code>immediate</code> , <code>pending-reboot</code>

這些參數是動態的，這表示對這些參數的任何變更都會立即生效，而且您不需要重新啟動資料庫執行個體。如果您不希望變更立即生效，您可以將這些變更設定 `ApplyMethod` 為 `pending-reboot` 並排在維護時段期間進行。

您可以使用 AWS Management Console、或 Amazon RDS API 來建立和修改自訂參數群組。AWS CLI

主控台

若要將您的 IBM Customer ID 和您的 IBM Site ID 加入參數群組

1. 建立新的資料庫參數群組。如需建立資料庫參數群組的詳細資訊，請參閱 [建立資料庫參數群組](#)。
2. 修改您建立的參數群組。如需修改參數群組的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

AWS CLI

若要將您的 IBM Customer ID 和您的 IBM Site ID 加入參數群組

1. 透過執行 [create-db-parameter-group](#) 指令建立自訂參數群組。

包含下列必要選項：

- `--db-parameter-group-name`— 您正在建立之參數群組的名稱。
- `--db-parameter-group-family`— Db2 引擎版本和主要版本。有效值：`db2-se-11.5`、`db2-ae-11.5`。
- `--description`— 此參數群組的描述。

如需建立資料庫參數群組的詳細資訊，請參閱 [建立資料庫參數群組](#)。

2. 修改透過執行 [modify-db-parameter-group](#) 指令建立的自訂參數群組中的參數。

包含下列必要選項：

- `--db-parameter-group-name`— 您建立的參數群組名稱。
- `--parameters`— 參數名稱、值以及參數更新的應用程式方法陣列。

如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

RDS API

若要將您的IBM Customer ID和您的IBM Site ID加入參數群組

1. 使用 Amazon RDS API [CreateDBParameterGroup](#)操作建立自訂資料庫參數群組。

包含下列必要參數：

- `DBParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

如需建立資料庫參數群組的詳細資訊，請參閱[建立資料庫參數群組](#)。

2. 修改您使用 RDS API [ModifyDBParameterGroup](#)作業建立的自訂參數群組中的參數。

包含下列必要參數：

- `DBParameterGroupName`
- `Parameters`

如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

現在，您可以建立資料庫執行個體，並將自訂參數群組附加到資料庫執行個體。如需詳細資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#) 及 [將資料庫參數群組與資料庫執行個體建立關聯](#)。

與整合 AWS License Manager

若要協助在 BYOL 模型中監控 RDS 的 Db2 授權使用情況，請與適用於 Db2 的 RDS [AWS License Manager](#) 整合。License Manager 支援根據虛擬 CPU (vCPUs) 追蹤適用於 Db2 引擎的 RDS 版本。您也可以搭配使用 License Manager AWS Organizations 來集中管理所有組織帳戶。

下表顯示適用於 Db2 之 RDS 的產品資訊篩選器。

篩選條件	名稱	描述
引擎版本	db2-se	Db2 標準版
	db2-ae	Db2 進階版

若要追蹤 RDS for Db2 資料庫執行個體的授權使用情況，您可以建立自我管理的授權。在此情況下，符合產品資訊篩選器的 RDS for Db2 資源會自動與自我管理授權相關聯。探索適用於 Db2 資料庫執行個體的 RDS 可能需要長達 24 小時的時間。

主控台

若要建立自我管理的授權，以追蹤 RDS for Db2 資料庫執行個體的授權使用情況

- 移至 <https://console.aws.amazon.com/license-manager/>。
- 建立自我管理授權。

如需指示，請參閱 AWS License Manager 使用指南中的 [建立自我管理授權](#)。

在 Product Information (產品資訊) 面板中，新增 RDS Product Information Filter (RDS 產品資訊篩選條件) 的規則。

如需詳細資訊，請參閱 AWS License Manager API 參考 [ProductInformation](#) 中的。

AWS CLI

若要使用建立自我管理的授權 AWS CLI，請呼叫 [建立授權組態](#) 指令。使用 `--cli-input-json` 或 `--cli-input-yaml` 參數，將參數傳遞至命令。

Example

下列程式碼會建立 Db2 標準版的自我管理授權。

```
aws license-manager create-license-configuration --cli-input-json file:///rds-db2-se.json
```

下列是範例中使用的範例 `rds-db2-se.json` 檔案。

```
{
  "Name": "rds-db2-se",
  "Description": "RDS Db2 Standard Edition",
  "LicenseCountingType": "vCPU",
  "LicenseCountHardLimit": false,
  "ProductInformationList": [
    {
      "ResourceType": "RDS",
      "ProductInformationFilterList": [
        {
          "ProductInformationFilterName": "Engine Edition",
          "ProductInformationFilterValue": ["db2-se"],
          "ProductInformationFilterComparator": "EQUALS"
        }
      ]
    }
  ]
}
```

如需關於產品資訊的詳細資訊，請參閱《AWS License Manager 使用者指南》中的[自動化探索資源庫存](#)。

如需關於 `--cli-input` 參數的詳細資訊，請參閱《AWS CLI 使用者指南》中的[從 JSON 或 YAML 輸入檔案產生AWS CLI Skeleton 及輸入參數](#)。

通過 Db2 許可證 AWS Marketplace

在 Db2 授權透過 AWS Marketplace 模式中，您需要支付每小時費率來訂閱 Db2 授權。此模型可協助您快速開始使用 RDS for Db2，而無需購買授權。

若要透過使用 Db2 授權 AWS Marketplace，您需要針對您要使用的特定 IBM Db2 版本提供有效的 AWS Marketplace 訂閱。如果您還沒有，請[AWS Marketplace 訂閱](#)該 IBM Db2 版本。

Amazon RDS 通過 AWS Marketplace IBM Db2 標準版和 IBM Db2 高級版支持 Db2 許可證。

主題

- [術語](#)

- [付款和帳單](#)
- [訂閱 Db2 Marketplace 清單並註冊 IBM](#)

術語

本頁面在與之討論 Amazon RDS 整合時使用下列術語 AWS Marketplace。

SaaS 訂閱

在中 AWS Marketplace，software-as-a-service (SaaS) 產品 (例如 pay-as-you-go 授權模式) 採用以使用為基礎的訂閱模式。IBM 是 Db2 的軟體銷售商，會追蹤您的使用情況，而您只需為使用量付費。

公開發售

公開優惠允許您直接從購買 AWS Marketplace 產品 AWS Management Console。

Db2 Marketplace 費用

Db2 軟體授權使用方式收取的費用 IBM。這些服務費用是透過計量的，AWS Marketplace 並顯示在您的 AWS 帳單上 AWS Marketplace。

Amazon RDS 費用

RDS for Db2 AWS 服務的費用，使用 AWS Marketplace Db2 授權時不包括授權。費用是透過使用中的 Amazon RDS 服務計算，並顯示在您的 AWS 帳單上。

付款和帳單

適用於 Db2 的 RDS 與整合，可 AWS Marketplace 為 Db2 提供每小時的 pay-as-you-go 授權。Db2 Marketplace 費用涵蓋了 Db2 軟體的授權成本，而 Amazon RDS 費用則涵蓋 RDS for Db2 資料庫執行個體使用量的成本。如需定價的相關資訊，請參閱[適用於 Db2 的 Amazon RDS 定價](#)。

若要停止這些費用，您必須刪除任何適用於 Db2 資料庫執行個體的 RDS。此外，您可以移除 Db2 授權的訂閱。AWS Marketplace 如果您在不刪除資料庫執行個體的情況下移除訂閱，Amazon RDS 將繼續向您收取使用資料庫執行個體的費用。如需詳細資訊，請參閱 [the section called “刪除資料庫執行個體”](#)。

[您可以 AWS Marketplace 在主控台中檢視使用 Db2 授權的 RDS for Db2 資料庫執行個體的帳單和管理付款。AWS Billing](#) 您的帳單包含兩項費用：一個用於透過使用 Db2 授權 AWS Marketplace，另一個用於使用 Amazon RDS。如需有關帳單的詳細資訊，請參閱 AWS Billing and Cost Management 使用指南中的 [檢視帳單](#)。

訂閱 Db2 Marketplace 清單並註冊 IBM

若要透過使用 Db2 授權 AWS Marketplace，您必須使用 AWS Management Console 來完成下列兩項工作。您無法透過 AWS CLI 或 RDS API 完成這些工作。

Note

如果要使用 AWS CLI 或 RDS API 建立資料庫執行個體，必須先完成這兩項工作。

主題

- [工作 1：在中訂閱 Db2 AWS Marketplace](#)
- [工作 2：註冊您的訂閱 IBM](#)

工作 1：在中訂閱 Db2 AWS Marketplace

若要搭配使用 Db2 授權 AWS Marketplace，您必須擁有 Db2 的有效 AWS Marketplace 訂閱。由於訂閱與特定 IBM Db2 版本相關聯，因此您必須 AWS Marketplace 針對您要使用的每個 Db2 版本訂閱中的 Db2：[IBM Db2 進階版](#)、[IBM Db2 標準版](#)。如需 AWS Marketplace 訂閱的相關資訊，請參閱 AWS Marketplace [購買指南中的 SaaS 使用量訂閱](#)。

建議您在開始[建立資料庫執行個體](#)之 AWS Marketplace 前，先訂閱中的 Db2。

工作 2：註冊您的訂閱 IBM

在中訂閱 Db2 之後 AWS Marketplace，請從您選擇的 Db2 訂閱類型 AWS Marketplace 頁面完成 IBM 訂單的註冊。在 AWS Marketplace 頁面上，選擇 [檢視購買選項]，然後選擇 [設定您的帳戶]。您可以使用現有 IBM 帳戶或創建免費 IBM 帳戶進行註冊。

在 Db2 授權之間切換

您可以在適用於 Db2 的 RDS 中的 Db2 授權之間切換。例如，您可以使用「自帶授權」開始，然後透過 AWS Marketplace 切換至 Db2 授權。

Important

如果您想要透過切換至 Db2 授權 AWS Marketplace，請確定您有您要使用的 IBM Db2 版本有效的 AWS Marketplace 訂閱。如果不這樣做，請先[訂閱該 Db2 版本 AWS Marketplace 的 Db2](#)，然後完成還原程序。

主控台

若要在 Db2 授權之間切換

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Automated backups (自動備份)。
自動備份顯示在 Current Region (目前區域) 索引標籤上。
3. 選擇您想要還原的資料庫執行個體。
4. 針對 Actions (動作)，選擇 Restore to point in time (還原至時間點)。

Restore to point in time (還原到時間點) 視窗隨即出現。

5. 選擇 Latest restorable time (最近的可還原時間) 以還原最近一次的可能時間，或選擇 Custom (自訂) 以選擇一個時間。

如果您選擇「自訂」，請輸入要還原執行個體的目標日期和時間。

Note

會以您的當地時區顯示時間，根據國際標準時間 (UTC) 的時差來表示。例如，UTC-5 是東部標準時間/中部日光節約時間。

6. 若為資料庫引擎，請選擇您要使用的 Db2 授權。
7. 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入目標還原資料庫執行個體的名稱。名稱必須是唯一的。
8. 視需要選擇其他選項，例如資料庫執行個體類別、儲存體以及是否要使用儲存體自動擴充功能。

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

9. 選擇 Restore to point in time (還原至時間點)。

如需詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

AWS CLI

若要在 Db2 授權之間切換，請使用 AWS CLI 指令 [恢復-DB](#) 執行個體對時間點。下列範例會還原最新 point-in-time 版本、將資料庫引擎設定為 IBM Db2 進階版，並透過 AWS Marketplace 將授權模式設定為 Db2 授權。

您可以指定其他設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier my_source_db_instance \  
  --target-db-instance-identifier my_target_db_instance \  
  --use-latest-restorable-time \  
  --engine db2-ae \  
  --license-model marketplace-license
```

在 Windows 中：

```
aws rds restore-db-instance-to-point-in-time ^  
  --source-db-instance-identifier my_source_db_instance ^  
  --target-db-instance-identifier my_target_db_instance ^  
  --use-latest-restorable-time ^  
  --engine db2-ae ^  
  --license-model marketplace-license
```

如需詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

RDS API

若要在 Db2 授權之間切換，請使用下列參數呼叫 Amazon RDS API [RestoreDBInstanceToPointInTime](#)操作：

- SourceDBInstanceIdentifier
- TargetDBInstanceIdentifier
- RestoreTime
- Engine
- LicenseModel

如需更多詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

Amazon RDS 適用於 Db2 執行個體類別

資料庫執行個體的運算和記憶體容量由其執行個體類別決定。您需要的資料庫執行個體類別取決於您的處理能力和記憶體需求。

適用於 Db2 執行個體類別的支援 RDS

支援的 Amazon RDS 適用於 Db2 執行個體類別是 Amazon RDS 資料庫執行個體類別的子集。如需 Amazon RDS 執行個體類別的完整清單，請參閱 [資料庫執行個體類別](#)。

下表列出 Db2 資料庫 11.5.9.0 支援的所有執行個體類別。

DB2 版	DB2 版本
Db2 Standard Edition 使用自有授權 (BYOL) 通過 Db2 許可證 AWS Marketplace	一般用途執行個體類別，包含第 3 代 Intel Xeon Scalable 處理器、SSD 儲存和網路最佳化 大型資料庫 .m6 ID.8 位元大 由第三代 Intel Xeon Scalable 處理器支援的一般用途執行個體類 大数据库 .m6 英寸 一般用途實體類別 DB.m6. 大 — 數據庫 .m6 l.8-大 記憶體最佳化執行個體類別，搭配本機 NVMe SSD，由第 3 代處理器提供支援 Intel Xeon Scalable db.x2iedn.xlarge 記憶體最佳化執行個體類別，由第 3 代 Intel Xeon Scalable 處理 大型資料庫 .r6 位元 .6 位元 .4 記憶體最佳化執行個體類別，由第 3 代 Intel Xeon Scalable 處理 大-數據庫 .r6 英寸 記憶體最佳化資料庫執行個體類別

DB2 版	DB2 版本
	db.r6i.large–db.r6i.4xlarge
	爆量效能執行個體類別
	db.t3.small–db.t3.2xlarge
Db2 Advanced Edition 使用自有授權 (BYOL) 通過 Db2 許可證 AWS Marketplace	一般用途執行個體類別，包含第 3 代 Intel Xeon Scalable 處理器、SSD 儲存和網路最佳化 資料庫 .m6 域 .12 倍大 — 資料庫 .m6 ID.32 X 大
	由第三代 Intel Xeon Scalable 處理器支援的一般用途執行個體類
	數據庫 .m6 英寸大 — 數據庫 .m6 英寸
	一般用途實體類別
	DB.m6. 12 X 大 — 資料庫 .m6 .32 X 大
	記憶體最佳化執行個體類別，搭配本機 NVMe SSD，由第 3 代處理器提供支援 Intel Xeon Scalable
	資料庫 .x2 加大 — 資料庫 .x2 整個 .32 X 大
	記憶體最佳化執行個體類別，由第 3 代 Intel Xeon Scalable 處理
	資料庫 .r6 區域 .8 倍大 — 資料庫 .r6 區域 .32 X 大
	記憶體最佳化執行個體類別，由第 3 代 Intel Xeon Scalable 處理
	數據庫 .r6 英寸大-數據庫 .r6 .32
	記憶體最佳化資料庫執行個體類別
	資料庫 .r6 .8 X 大 — 資料庫 .r6 .32 大

適用於 Db2 的 Amazon RDS 參數

Amazon RDS for Db2 支援透過參數群組修改資料庫管理員 (執行個體層級) 參數和 Db2 登錄參數。資料庫參數只能透過[rdsadmin.update_db_param](#)預存程序修改。

根據預設，RDS for Db2 資料庫執行個體使用 Db2 資料庫和資料庫執行個體專用的資料庫參數群組。此參數群組包含 IBM Db2 資料庫引擎的參數。如需使用參數群組和設定參數的相關資訊，請參閱[使用參數群組](#)。

RDS for Db2 參數會設定為您所選儲存引擎的預設值。如需 Db2 參數的詳細資訊，請參閱文件中的[Db2 資料庫組態參數](#)。IBM Db2

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 檢視特定 Db2 版本的可用參數。如需有關在主控台中檢視 Db2 參數群組中的參數的資訊，請參閱[檢視資料庫參數群組的參數值](#)。

使用 AWS CLI，您可以執行[describe-engine-default-parameters](#)指令來檢視 Db2 版本的參數。為 `--db-parameter-group-family` 選項指定下列其中一個值：

- db2-ae-11.5
- db2-se-11.5

例如，若要檢視 Db2 Standard Edition 11.5 的參數，請執行下列命令。

```
aws rds describe-engine-default-parameters --db-parameter-group-family db2-se-11.5
```

此命令會產生類似下列範例的輸出。

```
{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "agent_stack_sz",
        "ParameterValue": "1024",
        "Description": "You can use this parameter to determine the amount of memory that is allocated by Db2 for each agent thread stack.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "integer",
```

```

        "AllowedValues": "256-32768",
        "IsModifiable": false
    },
    {
        "ParameterName": "agentpri",
        "ParameterValue": "-1",
        "Description": "This parameter controls the priority given to all
agents and to other database manager instance processes and threads by the operating
system scheduler. This priority determines how CPU time is allocated to the database
manager processes, agents, and threads relative to other processes and threads running
on the machine.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "integer",
        "AllowedValues": "1-99",
        "IsModifiable": false
    },
    ...
]
}
}

```

若只要列出 Db2 Standard Edition 11.5 的可修改參數，請執行下列命令：

對於LinuxmacOS、或Unix：

```

aws rds describe-engine-default-parameters \
  --db-parameter-group-family db2-se-11.5 \
  --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'

```

在 Windows 中：

```

aws rds describe-engine-default-parameters ^
  --db-parameter-group-family db2-se-11.5 ^
  --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'

```

主題

- [確定哪些參數是可修改的](#)
- [修改參數](#)

確定哪些參數是可修改的

若要判斷您可以修改哪些資料庫管理員、資料庫和登錄參數，請執行下列命令。

1. Connect 至您的 Db2 資料庫。#####

```
db2 "connect to database_name user master_username using master_password"
```

2. 尋找支援的 Db2 版本。

```
db2 "select service_level, fixpack_num from table(sysproc.env_get_inst_info()) as instanceinfo"
```

3. 檢視特定 Db2 版本的參數。

- 檢視資料庫管理員組態參數。使用 AWS Management Console 或執行下列命令來檢查連接到資料庫執行個體的參數群組：

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
      when value_flags = 'NONE' then '' else value_flags end flags,
      cast(substr(value,1,64) as varchar(64)) as current_value
      from sysibmadm.dbmcfg
      order by name asc with UR"
```

- 檢視所有資料庫組態參數。

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
      when value_flags = 'NONE' then '' else value_flags end flags,
      cast(substr(value,1,64) as varchar(64)) as current_value
      from table(db_get_cfg(null)) order by name asc, member asc with UR"
```

- 檢視目前設定的登錄變數。

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
      cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
      level from table(env_get_reg_variables(null))
      order by reg_var_name,member with UR"
```

- 檢視所有支援的登錄變數清單。

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
      cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
```

```
level from table(env_get_reg_variables(null,1))
order by reg_var_name,member with UR"
```

修改參數

您可以修改自訂參數群組中的資料庫管理員和登錄參數。首先建立自訂參數群組，然後修改該自訂參數群組中的參數。如需詳細資訊，請參閱 [在資料庫執行個體中使用資料庫參數群組](#)。

若要變更資料庫參數，請執行下列命令。

1. Connect 至 `rdsadmin` 資料庫。在下面的例子中，用您的信息替換 `MASTER ###` 和 `###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 呼叫 `rdsadmin.update_db_param` 預存程序以變更資料庫參數。如需更多詳細資訊，請參閱 [rdsadmin.update_db_param](#)。

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Amazon RDS 上 Db2 資料庫的 EBCDIC 定序

適用於 Db2 的 Amazon RDS 支援 Db2 資料庫的 EBCDIC 定序。您只能在使用 Amazon RDS [the section called "rdsadmin.create_database"](#) 預存程序建立資料庫時，指定資料庫的 EBCDIC 定序順序。

使用 AWS Management Console、AWS CLI 或 RDS API 建立適用於 Db2 資料庫執行個體的 RDS 時，您可以指定資料庫名稱。如果您指定資料庫名稱，Amazon RDS 會使用預設定序建立資料庫。SYSTEM 如果您需要使用 EBCDIC 定序建立資料庫，請勿在建立資料庫執行個體時指定資料庫名稱。

RDS for Db2 中的資料庫定序會在建立時設定，且不可變。如果您在建立資料庫執行個體時指定了資料庫名稱，而且想要使用 EBCDIC 定序的資料庫，請刪除資料庫執行個體並建立新的執行個體。

若要使用 EBCDIC 定序建立 Db2 資料庫

1. 使用、或 RDS API 建立適用於 Db2 資料庫執行個體的 RDS AWS Management Console AWS CLI，而不指定資料庫名稱。如需詳細資訊，請參閱 [建立資料庫執行個體](#)。
2. 建立 Db2 資料庫，並呼叫預存程序，將定序選項設定為 EBCDIC 值。`rdsadmin.create_database`如需詳細資訊，請參閱 [rdsadmin.create_database](#)。

Important

使用預存程序建立資料庫之後，就無法變更定序順序。如果要讓資料庫使用不同的定序順序，請呼叫 [the section called “rdsadmin.drop_database”](#) 預存程序來卸除資料庫。然後，建立具有所需定序順序的資料庫。

適用於 Db2 資料庫執行個體的 Amazon RDS 的當地時區

依預設會設定執行 Db2 之 Amazon RDS 資料庫執行個體的時區。預設值是國際標準時間 (UTC)。若要符合應用程式的時區，您可以改為將資料庫執行個體的時區設定為本地時區。

您可以在第一次建立資料庫執行個體時設定時區。您可以使用 AWS Management Console RDS API 或 AWS CLI。如需詳細資訊，請參閱 [建立資料庫執行個體](#)。

如果您的資料庫執行個體屬於異地同步備份部署的一部分，則當容錯移轉時，其時區仍然是您設定的本地時區。

您可以將資料庫執行個體還原到您指定的時間點。時間會以您當地的時區顯示。如需詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

在資料庫執行個體上設定本地時區有下列限制：

- 您無法修改現有 Amazon RDS for Db2 資料庫執行個體的時區。
- 您無法將快照從某個時區的資料庫執行個體還原至另一個時區的資料庫執行個體。
- 強烈建議您不要將備份檔案從某個時區還原至不同時區。如果您將備份檔案從一個時區還原到另一個時區，則必須稽核查詢和應用程式，以瞭解時區變更的影響。

可用時區

您可以使用下列值做為時區設定。

區域	時區
非洲	非洲/開羅，非洲/卡薩布蘭卡，非洲/哈拉雷，非洲/拉哥斯，非洲/魯安達，非洲/蒙羅維亞，非洲/奈洛比，非洲/的黎波里，非洲/溫吐克
美洲	美洲/阿拉瓜伊納，美洲/阿根廷/布宜諾斯艾利斯，美洲/亞松森，美洲/波哥大，美洲/卡拉卡斯，美洲/芝加哥，美洲/赤瓦瓦州，美洲/古雅巴，美洲/丹佛，美洲/底特律，美洲/福塔力沙，美洲/哥特哈布，美洲/瓜地馬拉，美洲/哈利法克斯，美洲/利馬，美洲/洛杉磯，美洲/瑪瑙斯，美洲/馬塔莫羅斯，美洲/墨西哥市，美洲/蒙特利，美洲/蒙特維多，美洲/紐約州，美洲/鳳凰城，美洲/聖地牙哥，美洲/聖保羅，美洲/提華納，美洲/多倫多
亞洲	亞洲/安曼，亞洲/阿什哈巴德，亞洲/巴格達，亞洲/巴庫，亞洲/曼谷，亞洲/貝魯特，亞洲/加爾各答，亞洲/大馬士革，亞洲/達卡，亞洲/香港，亞洲/伊爾庫次克，亞洲/雅加達，亞洲/耶路撒冷，亞洲/喀布爾，亞洲/喀拉蚩，亞洲/加德滿都，亞洲/加爾各答，亞洲/克拉斯諾雅斯克，亞洲/馬加丹，亞洲/馬尼拉，亞洲/馬斯喀特，亞洲/新西伯利亞，亞洲/仰光，亞洲/利雅德，亞洲/首爾，亞洲/上海，亞洲/新加坡，亞洲/台北，亞洲/德黑蘭，亞洲/東京，亞洲/庫倫，亞洲/海參崴，亞洲/雅庫次克，亞洲/葉里溫
大西洋	大西洋/亞速爾群島，大西洋/維德角群島
澳大利亞	澳洲/阿德萊德，澳洲/布里斯班，澳洲/達爾文，澳洲/尤克拉，澳洲/荷巴特，澳洲/羅豪，澳洲/伯斯，澳洲/雪梨
巴西	巴西/，巴西/東DeNoronha部
加拿大	加拿大/紐芬蘭，加拿大/薩克其萬省
等等	等等/GMT-3
歐洲	歐洲/阿姆斯特丹，歐洲/雅典，歐洲/柏林，歐洲/都柏林，歐洲/赫爾辛基，歐洲/加里寧格勒，歐洲/倫敦，歐洲/馬德里，歐洲/莫斯科，歐洲/布拉格，歐洲/羅馬，歐洲/薩拉熱窩，歐洲/斯德哥爾摩
太平洋	太平洋/亞庇，太平洋/奧克蘭，太平洋/查塔姆，太平洋/斐濟，太平洋/關島，太平洋/檀香山，太平洋/刻里提瑪斯，太平洋/馬克沙斯，太平洋/薩摩亞，太平洋/東加塔普，太平洋/威克

區域	時區
美國	美國/阿拉斯加，美國/中部，美國/東印第安納州，美國/東部，美國/太平洋
UTC	UTC

建立適用於 Db2 資料庫執行個體的 Amazon RDS 的先決條件

下列項目是建立資料庫執行個體前的先決條件。

主題

- [管理員帳戶](#)
- [其他考量](#)

管理員帳戶

建立資料庫執行個體時，必須指定執行個體的管理員帳戶。Amazon RDS ACCESSCTRL 授予此本機資料庫管理員帳戶的授權。

管理員帳號具有下列特性、權能和限制：

- 是本機使用者，而不是 AWS 帳戶。
- 沒有 Db2 執行個體層級權限，例如SYSADM、SYSMAINT或。SYSCTRL
- 無法停止或啟動 Db2 執行個體。
- 如果您在建立資料庫執行個體時指定名稱，則無法刪除 Db2 資料庫。
- 具有 Db2 資料庫的完整存取權，包括目錄表格和視圖。
- 可以使用 Amazon RDS 存放程序建立本機使用者和群組。
- 可以授予和撤銷授權和權限。

管理員帳戶可以執行下列工作：

- 建立、修改或刪除資料庫執行個體。
- 建立資料庫快照。
- 啟動 point-in-time 還原。
- 建立資料庫快照的自動備份。
- 建立資料庫快照的手動備份。
- 使用其他 Amazon RDS 功能。

其他考量

建立資料庫執行個體之前，請考慮下列項目：

- 每個適用於 Db2 的 Amazon RDS 資料庫執行個體都可以託管單一 Db2 資料庫。
- 初始資料庫名稱
 - 如果您在建立資料庫執行個體時未提供資料庫名稱，Amazon RDS 不會建立資料庫。
 - 在下列情況下，請勿提供資料庫名稱：
 - 您想要使用 Amazon RDS 存放程序來[建立](#)或[刪除](#)資料庫。
 - 您想要建立使用 EBCDIC 定序順序的資料庫。如需詳細資訊，請參閱 [Amazon RDS 上 Db2 資料庫的 EBCDIC 定序](#)。
 - 您想要從 Amazon S3 還原備份。
 - 您要從 AIX 或移轉 Windows。如需詳細資訊，請參閱 [在 Linux 環境之間 Windows 進 AIX 行一次性移轉](#)。
- 在使用您自己的授權 (BYOL) 模型中，您必須先建立包含您的 IBM Customer ID 和 IBM Site ID 如需詳細資訊，請參閱 [為 Db2 帶來您自己的授權](#)。
- 在 Db2 授權透過 AWS Marketplace 模式中，您需要針對要使用的特定 IBM Db2 版本提供有效的 AWS Marketplace 訂閱。如果您還沒有，請[在中訂閱 Db2 以 AWS Marketplace](#)取得您要使用的 IBM Db2 版本。如需更多詳細資訊，請參閱 [通過 Db2 許可證 AWS Marketplace](#)。

連接到您的 Amazon RDS 用 Db2 資料庫執行個體

Amazon RDS 佈建您的 Amazon RDS for Db2 資料庫執行個體之後，您可以使用任何標準 SQL 用戶端應用程式連線到資料庫執行個體。由於 Amazon RDS 是受管服務，因此您無法以 SYSADM、SYSCTRLSECADM、或身分登入 SYSMAINT。

您可以使用 IBM Db2 CLP、IBM CLPPlus、或連線至執行 IBM Db2 資料庫引擎的資料庫執行個體 IBM Db2 Data Management Console。DBeaver

主題

- [尋找適用於 Db2 資料庫執行個體的 Amazon RDS 端點](#)
- [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 CLP](#)
- [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM CLPPlus](#)
- [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 DBeaver](#)
- [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 Data Management Console](#)
- [使用適用於 Db2 的 Amazon RDS 安全群組的注意事項](#)

尋找適用於 Db2 資料庫執行個體的 Amazon RDS 端點

每個 Amazon RDS 資料庫執行個體會有端點，並且每個端點有資料庫執行個體的 DNS 名稱和連接埠號碼。若要使用 SQL 用戶端應用程式連接到 Amazon RDS for Db2 資料庫執行個體，您需要資料庫執行個體的 DNS 名稱和連接埠號碼。

您可以使用 AWS Management Console 或來尋找資料庫執行個體的端點 AWS CLI。

主控台

尋找適用於 Db2 資料庫執行個體的 RDS 端點

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在主控台右上角，選擇資料庫執行個體 AWS 區域的。
3. 尋找適用於 Db2 資料庫執行個體之 RDS 的 DNS 名稱和連接埠號碼。
 - a. 選擇 Databases (資料庫) 來顯示資料庫執行個體的清單。
 - b. 選擇適用於 Db2 資料庫執行個體的 RDS 名稱，以顯示執行個體詳細資訊。

- c. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

The screenshot displays the AWS Management Console interface for an RDS instance. The 'Connectivity & security' tab is selected. Under the 'Endpoint & port' section, the endpoint is 'database-1. [redacted].amazonaws.com' and the port is '50000'. The 'Networking' section shows the instance is in the 'us-east-2a' Availability Zone, using a 'vpc-[redacted]' VPC and a 'default-vpc-[redacted]' subnet group. The 'Security' section indicates the instance is in a 'default [redacted] Active' VPC security group, is 'Publicly accessible' (Yes), and uses the 'rds-ca-2019' certificate authority.

AWS CLI

若要尋找適用於 Db2 資料庫執行個體的 RDS 端點，請執行命 [describe-db-instances](#) 令。在下列範例中，將 `### -1` 取代為資料庫執行個體的名稱。

對於 Linux/macOS、或 Unix：

```
aws rds describe-db-instances \
  --db-instance-identifier database-1 \
  --query 'DBInstances[]'.
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' \
  --output json
```

在 Windows 中：

```
aws rds describe-db-instances ^
  --db-instance-identifier database-1 ^
  --query 'DBInstances[]'.
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' ^
  --output json
```

此命令會產生類似下列範例的輸出。輸出中的 `Address` 行包含 DNS 名稱。

```
[
  {
    "DBInstanceIdentifier": "database-1",
    "DBName": "DB2DB",
    "Endpoint": {
      "Address": "database-1.123456789012.us-east-2.amazonaws.com",
      "Port": 50000,
      "HostedZoneId": "Z20C4A7DETW6VH"
    }
  }
]
```

使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 CLP

您可以使用命令列公用程式，例如連接 IBM Db2 CLP 到 Amazon RDS for Db2 資料庫執行個體。此公用程式是其中的一部分 IBM Data Server Runtime Client。若要從下載用戶端 IBM Fix Central，請參閱 Sup IBM port 中的 [IBM 資料伺服器用戶端套件版本 11.5 Mod 8 修正套件 0](#)。

主題

- [術語](#)
- [安裝用戶端](#)
- [連接至資料庫執行個體](#)
- [對 Db2 資料庫執行個體的 RDS 連線進行疑難排解](#)

術語

下列術語有助於說明 [連線至 RDS for Db2 資料庫執行個體時所使用的](#) 命令。

目錄節點

此命令註冊與本地 Db2 客戶端的遠程數據庫節點，這使得該節點可供客戶端應用程序訪問。若要編目節點，您需要提供伺服器的主機名稱、連接埠號碼和通訊協定等資訊。然後，已編目的節點代表一或多個遠端資料庫所在的目標伺服器。如需詳細資訊，請參閱 IBM Db2 文件中的 [CATALOG TCPIP/TCPIP4/TCPIP6 NODE 指令](#)。

目錄資料庫

這個命令註冊一個遠程資料庫與本地 Db2 客戶端，這使得該資料庫的客戶端應用程式訪問。若要編目資料庫，您需要提供資訊，例如資料庫的別名、它所在的節點，以及連線到資料庫所需的驗證類型。如需詳細資訊，請參閱IBM Db2文件中的[CATALOG DATABASE指令](#)。

安裝用戶端

之後[downloading the package for Linux](#)，請使用 root 或管理員權限安裝用戶端。

Note

若要在AIX或上安裝用戶端Windows，請遵循相同的程序，但修改作業系統的指令。

若要安裝用戶端 Linux

1. 執行 `./db2_install -f sysreq` 並選擇 **yes** 接受授權。
2. 選擇要安裝用戶端的位置。
3. 執行執行個體 `clientInstallDir/instance/db2icrt -s client###`。在上以有效的作業系統使用者取代 `#####`。Linux 在中 Linux，Db2 DB 執行個體名稱與作業系統使用者名稱繫結。

此指令會在 `sqllib` 指定使用者的主目錄中建立目錄 Linux。

連接至資料庫執行個體

若要連線至您的 RDS for Db2 資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找它們的資訊，請參閱[尋找端點](#)。您也需要知道建立 RDS for Db2 資料庫執行個體時所定義的資料庫名稱、主要使用者名稱和主要密碼。如需尋找它們的更多資訊，請參閱[建立資料庫執行個體](#)。

若要使用以下方式連線至適用於 Db2 資料庫執行個體 IBM Db2 CLP

1. 使用您在使用 IBM Db2 CLP 用戶端安裝期間指定的使用者名稱登入。
2. 編目適用於 Db2 資料庫執行個體的 RDS。在下列範例中，將 `node_name`、`dns_name` 和 `###` 取代之為本機目錄中節點的名稱、資料庫執行個體的 DNS 名稱和連接埠號碼。

```
db2 catalog TCPIP node node_name remote dns_name server port
```

範例

```
db2 catalog TCPIP node remnode remote database-1.123456789012.us-
east-1.amazonaws.com server 50000
```

3. 編目 `rdsadmin` 資料庫和資料庫。這可讓您連線到 `rdsadmin` 資料庫，以使用 Amazon RDS 存放程序執行某些管理任務。如需詳細資訊，請參閱 [管理適用於 Db2 資料庫執行個體的 RDS](#)。

在下列範例中，請使用此資料 `##### Database_alias#node_name` 和資料庫名稱。 `server_encrypt` 透過網路加密您的使用者名稱和密碼。

```
db2 catalog database rdsadmin [ as database_alias ] at node node_name
authentication server_encrypt

db2 catalog database database_name [ as database_alias ] at node node_name
authentication server_encrypt
```

範例

```
db2 catalog database rdsadmin at node remnode authentication server_encrypt

db2 catalog database testdb as rdsdb2 at node remnode authentication server_encrypt
```

4. Connect 至適用於 Db2 的 RDS 資料庫。在下列範例中，將 `rds_database_alias`、主要 `#####` `#####` 要使用者名稱和您 RDS for Db2 資料庫執行個體的主要密碼。

```
db2 connect to rds_database_alias user master_username using master_password
```

此命令會產生類似下列範例的輸出：

```
Database Connection Information

Database server          = DB2/LINUX8664 11.5.9.0
SQL authorization ID    = ADMIN
Local database alias    = TESTDB
```

5. 執行查詢並檢視結果。下列範例會顯示選取您建立之資料庫的 SQL 陳述式。

```
db2 "select current server from sysibm.dual"
```


此命令會產生類似下列範例的輸出：

```
1
-----
TESTDB

1 record(s) selected.
```

對 Db2 資料庫執行個體的 RDS 連線進行疑難排解

如果您收到下列NULLID錯誤，通常表示您的用戶端和 RDS for Db2 伺服器版本不相符。如需支援的 Db2 用戶端版本，請參閱IBM Db2說明文件中[支援的用戶端、驅動程式和伺服器層級組合](#)。

```
db2 "select * from syscat.tables"
SQL0805N Package "NULLID.SQLC2029 0X414141414141454A69" was not found.
SQLSTATE=51002
```

收到此錯誤之後，您必須將套件從舊版 Db2 用戶端繫結至 RDS for Db2 支援的 Db2 伺服器版本。

將套件從較舊的 Db2 用戶端繫結至較新的 Db2 伺服器

1. 在用戶端電腦上找到繫結檔案。通常，這些檔案位於 Db2 用戶端安裝路徑的 bnd 目錄中，副檔名為 .bnd。
2. Connect 至 Db2 伺服器。在下列範例中，將#####取代為 Db2 伺服器的名稱。用您的信息替換###和###。此使用者具有DBADM權限。

```
db2 connect to database_name user master_username using master_password
```

3. 執行bind命令以繫結套件。
 - a. 導覽至用戶端機器上連結檔案所在的目錄。
 - b. 針對每個檔案執行bind命令。

下列選項是必要的：

- blocking all— 將繫結檔案中的所有套件繫結到單一資料庫要求中。
- grant public— 授予執public行封裝的權限。
- sqlerror continue— 指定即使發生錯誤，bind程序仍會繼續執行。

若要取得有關bind指令的更多資訊，請參閱IBM Db2文件中的[BIND指令](#)。

4. 透過查詢syscat.package目錄檢視或檢查bind命令後傳回的訊息，確認繫結是否成功。

如需詳細資訊，請參閱 [Support 中IBM的 DB2 v11.5 繫結檔案和 Package 件名稱清單](#)。

使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM CLPPlus

您可以使用公用程式，例如連接IBM CLPPlus到 Amazon RDS for Db2 資料庫執行個體。此公用程式是其中的一部分IBM Data Server Runtime Client。若要從下載用戶端IBM Fix Central，請參閱 Sup IBM port 中的[IBM資料伺服器用戶端套件版本 11.5 Mod 8 修正套件 0](#)。

Important

建議您IBM CLPPlus在支援圖形化使用者介面 (例如macOS、Windows或Linux桌面) 的作業系統上執行。如果運行無頭Linux，請使用 switch -nw 與CLPPlus命令。

主題

- [安裝用戶端](#)
- [連接至資料庫執行個體](#)

安裝用戶端

下載的套件之後Linux，請安裝用戶端。

Note

若要在AIX或上安裝用戶端Windows，請遵循相同的程序，但修改作業系統的指令。

若要安裝用戶端 Linux

1. 執行 `./db2_install`。
2. 執行執行個體 `clientInstallDir/instance/db2icrt -s client###`。在上以有效的作業系統使用者取代#####。Linux在中Linux，Db2 DB 執行個體名稱與作業系統使用者名稱繫結。

此指令會在`sqllib`指定使用者的主目錄中建立目錄Linux。

連接至資料庫執行個體

若要連線至您的 RDS for Db2 資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找它們的資訊，請參閱[尋找端點](#)。您也需要知道建立 RDS for Db2 資料庫執行個體時所定義的資料庫名稱、主要使用者名稱和主要密碼。如需尋找它們的更多資訊，請參閱[建立資料庫執行個體](#)。

若要使用以下方式連線至適用於 Db2 資料庫執行個體 IBM CLPPlus

1. 檢閱命令語法。在下列範例中，將 `ClientDir` 取代為安裝用戶端的位置。

```
cd clientDir/bin
./clpplus -h
```

2. 設定您的 Db2 伺服器。在下列範例中，將 `dns_name#####`、`##`和`###`取代為適用於 Db2 資料庫執行個體的 DNS 名稱、資料庫名稱、端點和連接埠。如需詳細資訊，請參閱[尋找適用於 Db2 資料庫執行個體的 Amazon RDS 端點](#)。

```
db2cli writecfg add -dsn dns_name -database database_name -host endpoint -port port
-parameter "Authentication=SERVER_ENCRYPT"
```

3. Connect 至適用於 Db2 的 RDS 資料庫執行個體。在下列範例中，使用主要使用者名稱和 DNS 名稱取代 `MASTER_##### dns_name`。

```
./clpplus -nw master_username@dns_name
```

4. 隨即開Java Shell啟一個視窗。輸入適用於 Db2 資料庫執行個體的 RDS 主要密碼。

Note

如果視Java Shell窗未開啟，請執行`./clpplus -nw`以使用相同的命令列視窗。

```
Enter password: *****
```

建立連接並產生類似於下列範例的輸出：

```
Database Connection Information :
-----
Hostname = database-1.abcdefghij.us-east-1.rds.amazonaws.com
Database server = DB2/LINUX8664 SQL110590
SQL authorization ID = admin
Local database alias = DB2DB
Port = 50000
```

5. 執行查詢並檢視結果。下列範例會顯示選取您建立之資料庫的 SQL 陳述式。

```
SQL > select current server from sysibm.dual;
```

此命令會產生類似下列範例的輸出：

```
1
-----
DB2DB
SQL>
```

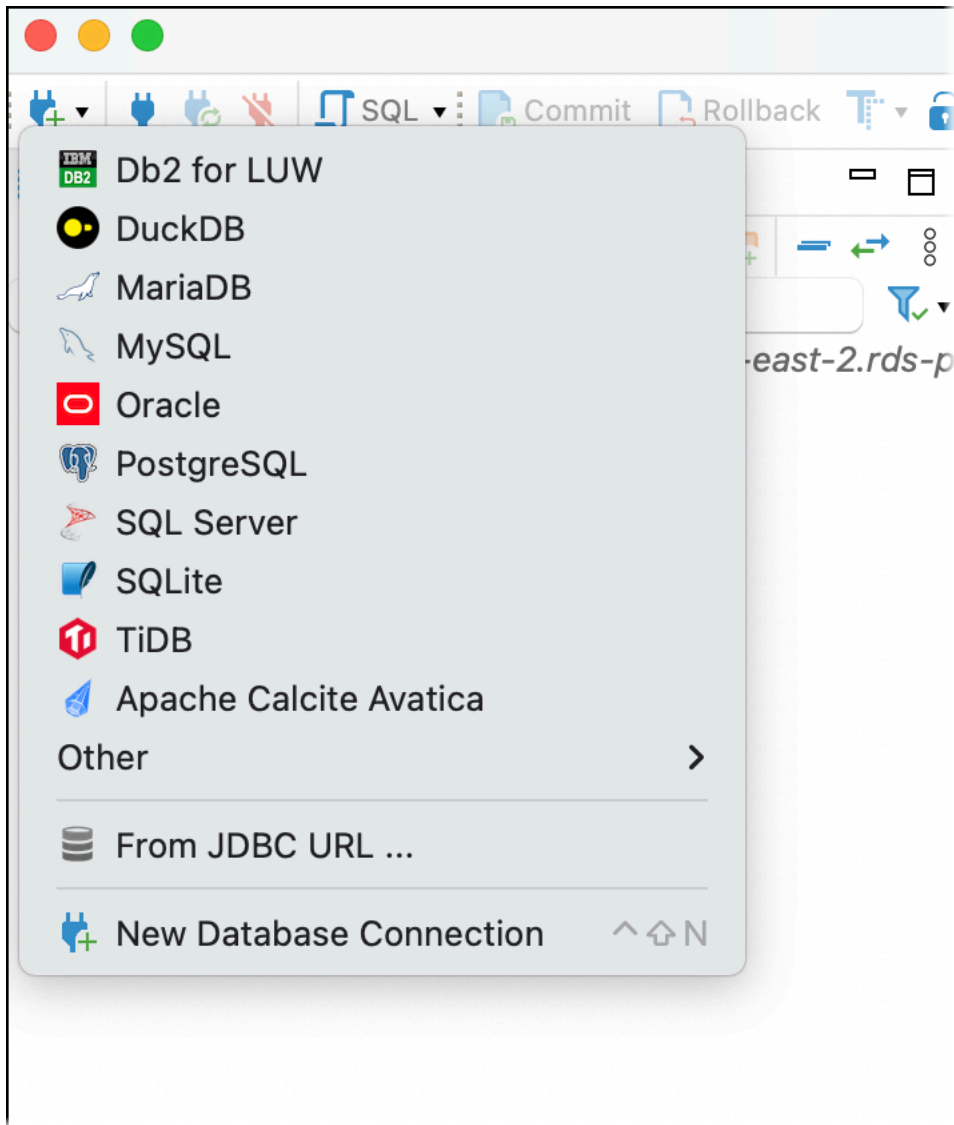
使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 DBeaver

您可以使用第三方工具，例如連接DBeaver到 Amazon RDS for Db2 資料庫執行個體。若要下載此公用程式，請參閱[DBeaver社群](#)。

若要連線至您的 RDS for Db2 資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找它們的資訊，請參閱[尋找端點](#)。您也需要知道建立 RDS for Db2 資料庫執行個體時所定義的資料庫名稱、主要使用者名稱和主要密碼。如需尋找它們的更多資訊，請參閱[建立資料庫執行個體](#)。

若要使用以下方式連線至適用於 Db2 資料庫執行個體 DBeaver

1. 啟動 DBeaver。
2. 選擇工具列中的「新增連線」圖示，然後選擇「Db2」做為 LUW。



3. 在「Connect 到資料庫」視窗中，提供 RDS for Db2 資料庫執行個體的資訊。
 - a. 輸入下列資訊：
 - 在主機中，輸入資料庫執行個體的 DNS 名稱。
 - 在連接埠中，輸入資料庫執行個體的連接埠號碼。
 - 在「資料庫」中，輸入資料庫的名稱。
 - 針對 Username (使用者名稱)，輸入資料庫執行個體之資料庫管理員的名稱。
 - 在密碼中，輸入資料庫執行個體的資料庫管理員密碼。
 - b. 選取 [儲存密碼]。
 - c. 選擇驅動程式設定。

Connect to a database

DB2 Connection Settings
Db2 for LUW connection settings

IBM DB2

Main | Trace settings | Driver properties | SSH | + Network configurations...

Database

Connect by: Host URL

URL: jdbc:db2://database-1. .amazonaws.com:50000/PERFDB

Host: database-1. .amazonaws.com Port: 50000

Database: PERFDB

Authentication (Database Native)

Username: admin

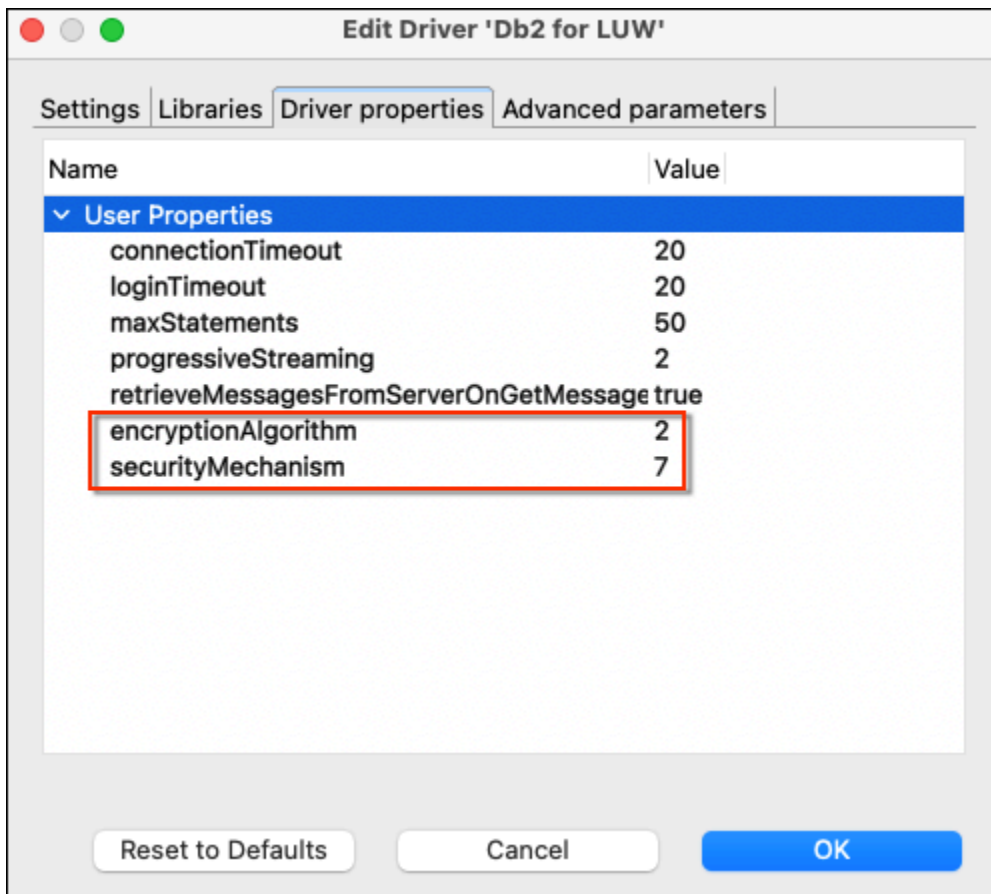
Password: Save password

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: Db2 for LUW Driver Settings

Test Connection ... < Back Next > Cancel Finish

4. 在「編輯驅動程式」視窗中，指定其他安全性內容。
 - a. 選擇「驅動程式內容」標籤。
 - b. 新增兩個使用者屬性。
 - i. 開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇 [新增屬性]。
 - ii. 對於「屬性名稱」，新增加密演算法，然後選擇「確定」。
 - iii. 選取「加密演算法」列後，選擇「值」欄並新增 2。
 - iv. 開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇 [新增屬性]。
 - v. 針對「內容名稱」，新增安全性機制，然後選擇確定。
 - vi. 選取「安全機制」列後，選擇「值」欄並新增 7。
 - c. 選擇確定。

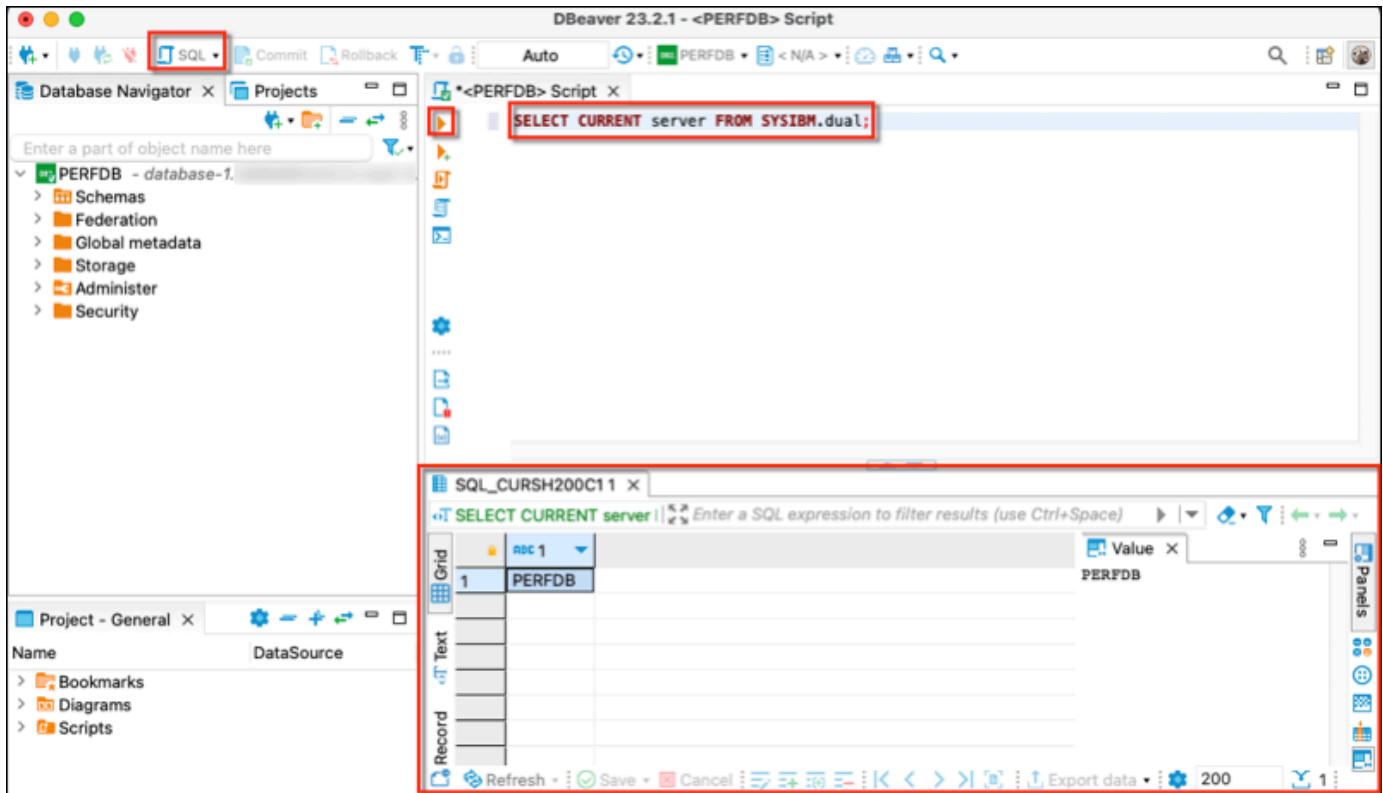


5. 在「Connect 線至資料庫」視窗中，選擇「測試連線」。如果您的電腦上沒有安裝 DB2 JDBC 驅動程式，則驅動程式會自動下載。
6. 選擇確定。
7. 選擇 Finish (完成)。
8. 在「資料庫導覽」標籤中，選擇資料庫的名稱。您現在可以探索物件。

您現在已準備好執行 SQL 命令。

若要執行 SQL 命令並檢視結果

1. 在頂端功能表中，選擇 [SQL]。這會開啟 SQL 指令碼面板。
2. 在「指令集」面板中，輸入 SQL 指令。
3. 若要執行命令，請選擇「執行 SQL 查詢」按鈕。
4. 在「SQL 結果」面板中，檢視 SQL 查詢的結果。



使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 Data Management Console

您可以使用連線到您的 Amazon RDS for Db2 資料庫執行個體 IBM Db2 Data Management Console。IBM Db2 Data Management Console 可以管理和監控 Db2 資料庫執行個體的多個 RDS。若要下載此公用程式，請參閱 IBM Support 中心中的 [3.1x IBM Db2 Data Management Console 版本](#)。

IBM Db2 Data Management Console 需要儲存庫 Db2 資料庫來儲存中繼資料和效能指標，但無法自動建立 RDS for Db2 的存放庫。

您必須先建立儲存庫資料庫，以監視一或多個適用於 Db2 資料庫執行個體的 RDS。然後使用連接到您的 RDS for Db2 資料庫執行個體 IBM Db2 Data Management Console。

主題

- [建立儲存庫資料庫以監視資料庫執行個](#)
- [使用下列方式連線至適用於 Db2 資料庫執行 IBM Db2 Data Management Console](#)

建立儲存庫資料庫以監視資料庫執行個

您可以使用現有適當大小的 RDS for Db2 資料庫執行個體做為監視其他 RDS for Db2 資料庫執行個體的儲存庫。IBM Db2 Data Management Console但是，由於 admin 使用者無SYSCTRL權建立緩衝集區和表格空間，因此使用IBM Db2 Data Management Console儲存庫建立存放庫資料庫會失敗。相反地，您必須建立儲存庫資料庫，以監視適用於 Db2 資料庫執行個體的 RDS。您可以透過兩種不同的方式建立儲存庫資料庫。您可以手動建立IBM Db2 Data Management Console儲存區域的緩衝集區、表格空間和物件。或者，您也可以建立單獨的 Amazon EC2 執行個體來託管IBM Db2 Data Management Console儲存庫。

主題

- [手動建立緩衝集區、表格空間和物件](#)
- [建立 Amazon EC2 執行個體以託管IBM Db2 Data Management Console儲存庫](#)

手動建立緩衝集區、表格空間和物件

建立要使用的緩衝區集區、表格空間及物件 IBM Db2 Data Management Console

1. 允許緩衝集區和表格空間的權限。
 - a. 變更命令檔，特別是緩衝集區和表格空間。如需詳細資訊，請參閱[IBM Db2 Data Management Console說明文件中的設定儲存庫](#)資料庫。
 - b. Connect 至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ##* 和 *###*。

```
db2 connect to rdsadmin user master_username using master_password
```

- c. 建立的緩衝集區IBM Db2 Data Management Console。在下列範例中，IBM Db2 Data Management Console將#####取代為您建立的存放庫名稱，以監控適用於 Db2 資料庫執行個體的 RDS。

```
db2 "call rdsadmin.create_bufferpool('database_name',  
    'BP4CONSOLE', 1000, 'Y', 'Y', 16384)"
```

- d. 建立的表格空間IBM Db2 Data Management Console。在下列範例中，IBM Db2 Data Management Console將#####取代為您建立的存放庫名稱，以監控適用於 Db2 資料庫執行個體的 RDS。

```
db2 "call rdsadmin.create_tablespace('database_name',
    'TS4CONSOLE', 'BP4CONSOLE', 16384)"
```

- e. 建立的暫時表格空間 IBM Db2 Data Management Console。在下列範例中，IBM Db2 Data Management Console 將 ##### 取代為您建立的存放庫名稱，以監控適用於 Db2 資料庫執行個體的 RDS。

```
db2 "call rdsadmin.create_tablespace('database_name',
    'TS4CONSOLE_TEMP', 'BP4CONSOLE', 16384, 0, 0, 'T')"
```

2. 手動建立 IBM Db2 Data Management Console 物件。如需詳細資訊，請參閱 [IBM Db2 Data Management Console 說明文件中的設定儲存庫](#) 資料庫。

建立 Amazon EC2 執行個體以託管 IBM Db2 Data Management Console 儲存庫

您可以建立單獨的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體來託管 IBM Db2 Data Management Console 儲存庫。如需建立 Amazon EC2 [Linux 執行個體的相關資訊](#)，請參閱 [Amazon EC2 使用者指南中的教學課程：開始使用 Amazon EC2 執行個體](#)。

使用下列方式連線至適用於 Db2 資料庫執行 IBM Db2 Data Management Console

若要連線至您的 RDS for Db2 資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找它們的資訊，請參閱 [尋找端點](#)。您也需要知道建立 RDS for Db2 資料庫執行個體時所定義的資料庫名稱、主要使用者名稱和主要密碼。如需尋找它們的更多資訊，請參閱 [建立資料庫執行個體](#)。如果您是透過網際網路連線，請允許資料庫連接埠的流量。如需詳細資訊，請參閱 [建立資料庫執行個體](#)。

若要使用以下方式連線至適用於 Db2 資料庫執行 IBM Db2 Data Management Console

1. 啟動 IBM Db2 Data Management Console。
2. 配置存放庫。
 - a. 在「連線和資料庫」區段中，輸入 RDS for Db2 資料庫執行個體的下列資訊：
 - 在主機中，輸入資料庫執行個體的 DNS 名稱。
 - 在連接埠中，輸入資料庫執行個體的連接埠號碼。
 - 在「資料庫」中，輸入資料庫的名稱。

Connection and database

Set up a repository on the database to enable monitoring, run SQL statements, and explore database objects. Make sure the database for the repository exists even before you start configuring the repository. You can use your own Db2 server or use the standard edition with the restricted license for this repository database. If the database is not already created, can also use the [Db2 docker](#) image and get started.

Important: For a Db2 repository database, the user must have minimum of DBADM with DATAACCESS on the database and SYSCTRL on database instance privilege. To configure the repository by a normal Db2 user, refer to this [procedure](#).

<p>Connection type</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between; align-items: center;"> IBM Db2 ▼ </div>	<p>Host</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f0f0f0;"> EXAMPLE_HOST </div>
<p>Port</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> 50000 - + </div>	<p>Database</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f0f0f0;"> SAMPLE </div>
<p>Repository schema ⓘ</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f0f0f0;"> IBMCONSOLE </div>	<p>JDBC URL attribute (optional)</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f0f0f0;"> Example: traceLevel=32;progressiveStream </div>

b. 在「安全性與認證」區段中，輸入 RDS for Db2 資料庫執行個體的下列資訊：

- 針對 [安全性類型]，選擇 [加密使用者和密碼]
- 針對 Username (使用者名稱)，輸入資料庫執行個體之資料庫管理員的名稱。
- 在密碼中，輸入資料庫執行個體的資料庫管理員密碼。

c. 選擇 Test connection (測試連線)。

i Note

如果連線失敗，請透過安全群組的輸入規則確認資料庫連接埠已開啟。如需詳細資訊，請參閱 [使用適用於 Db2 的 Amazon RDS 安全群組的注意事項](#)。

下列錯誤訊息指出連線至 RDS for Db2 資料庫執行個體的管理員使用者沒有建立緩衝集區或表格空間的權限。它還表明對於 Db2 存儲庫數據庫，用戶必須在數據庫DATAACCESS上具有DBADM和。使用者也必須擁有資料庫執SYSCTRL行個體權限。

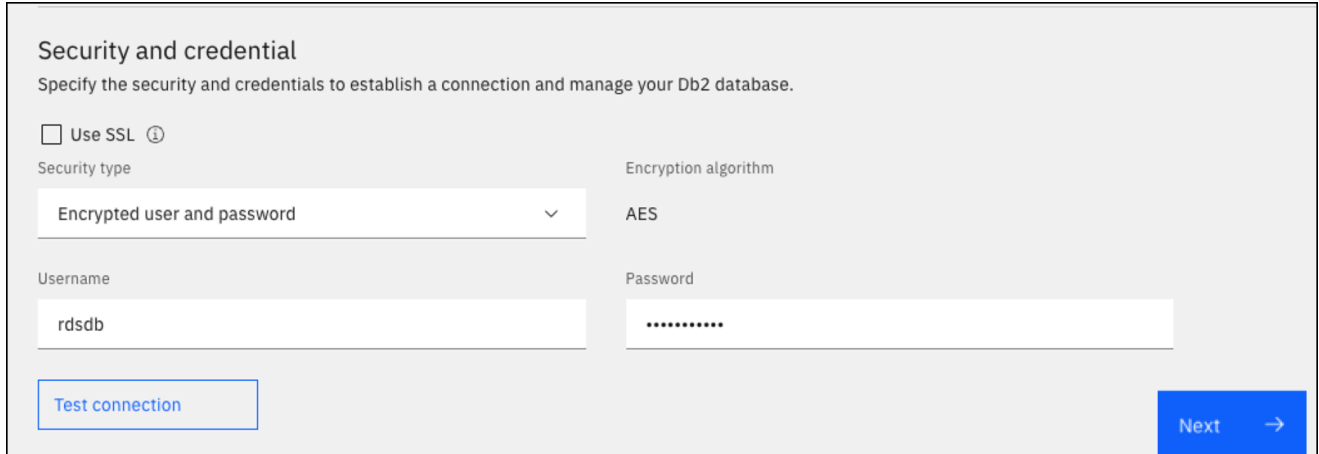


Error:
"ADMIN" does not have the privilege to perform operation "CREATE BUFFERPOOL".. SQLCODE=-552, SQLSTATE=42502

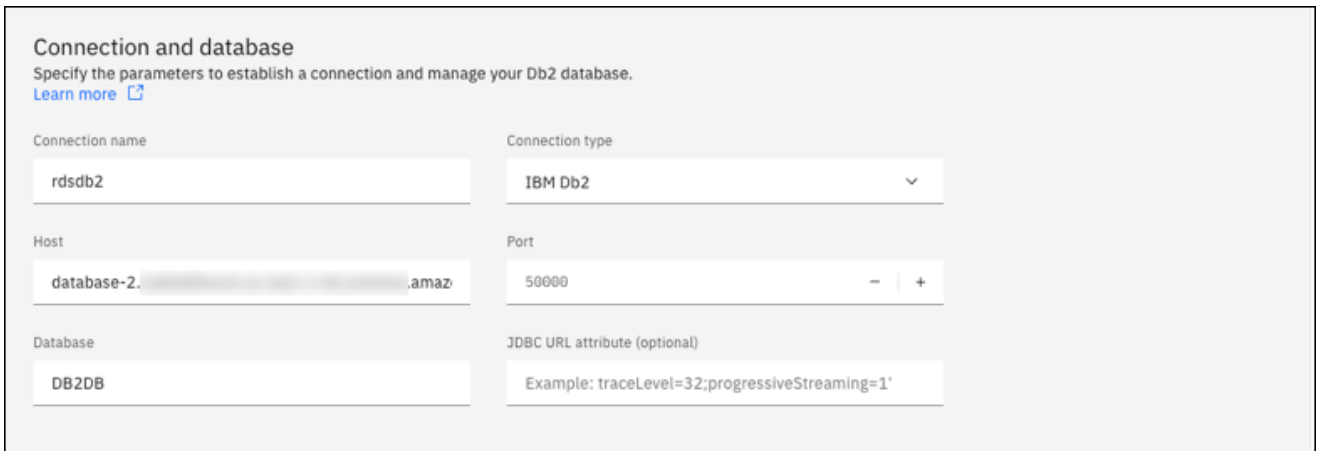
For a Db2 repository database, the user must have minimum of DBADM with DATAACCESS on the database and SYSCTRL on database instance privilege. To configure the repository by a normal Db2 user, refer to this [procedure](#)

請確定您已為儲存庫建立緩衝區表格、表格空間和物件，以監視 RDS for Db2 資料庫執行個體。或者，您也可以使用 Amazon EC2 Db2 資料庫執行個體託管 IBM Db2 Data Management Console 儲存庫，以監控適用於 Db2 資料庫的 RDS 執行個體。如需詳細資訊，請參閱 [建立儲存庫資料庫以監視資料庫執行個體](#)。

- d. 成功測試連線後，請選擇 [下一步]。



3. 在設定統計資料事件監視器選擇加入視窗中，選擇下一步。
4. (選擇性) 新增連線。如果您想要使用不同的 RDS for Db2 資料庫執行個體進行管理和監控，請將連線新增至非儲存庫 RDS for Db2 資料庫執行個體。
 - a. 在「連線和資料庫」區段中，輸入要用於管理和監督之 RDS for Db2 資料庫執行個體的下列資訊：
 - 在「連線名稱」中，輸入 Db2 資料庫識別碼。
 - 在主機中，輸入資料庫執行個體的 DNS 名稱。
 - 在連接埠中，輸入資料庫執行個體的連接埠號碼。
 - 在「資料庫」中，輸入資料庫的名稱。



Connection and database
Specify the parameters to establish a connection and manage your Db2 database.
[Learn more](#)

Connection name: rdsdb2

Connection type: IBM Db2

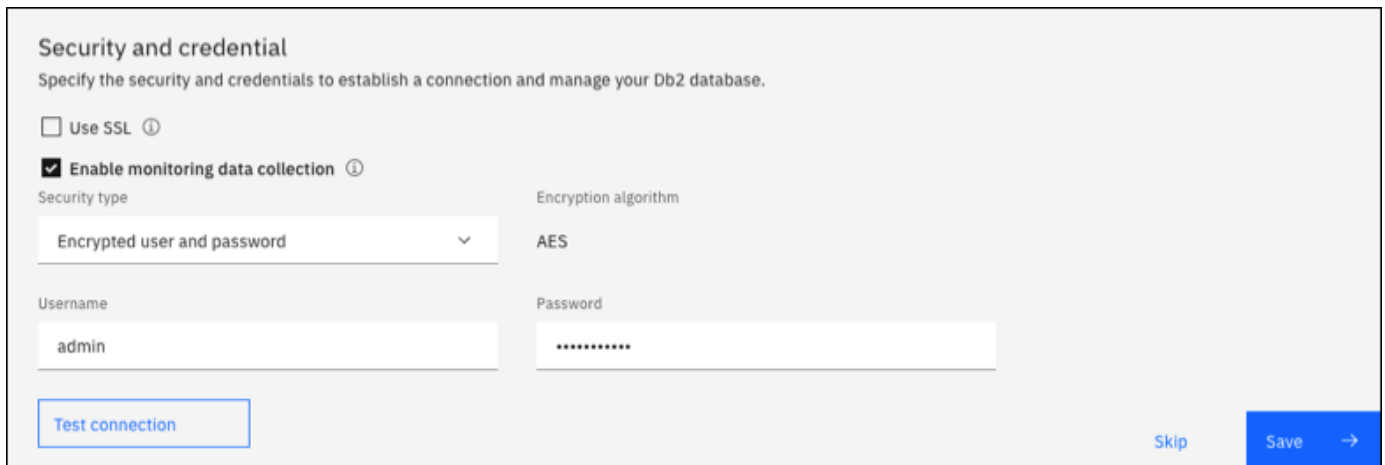
Host: database-2. .amaz

Port: 50000

Database: DB2DB

JDBC URL attribute (optional): Example: traceLevel=32;progressiveStreaming=1

- b. 在 [安全性和認證] 區段中，選取 [啟用監視資料收集]。
- c. 針對 Db2 資料庫執行個體的 RDS 輸入下列資訊：
 - 針對 Username (使用者名稱)，輸入資料庫執行個體之資料庫管理員的名稱。
 - 在密碼中，輸入資料庫執行個體的資料庫管理員密碼。
- d. 選擇 Test connection (測試連線)。
- e. 成功測試連線後，請選擇 [儲存]。



Security and credential
Specify the security and credentials to establish a connection and manage your Db2 database.

Use SSL

Enable monitoring data collection

Security type: Encrypted user and password

Encryption algorithm: AES

Username: admin

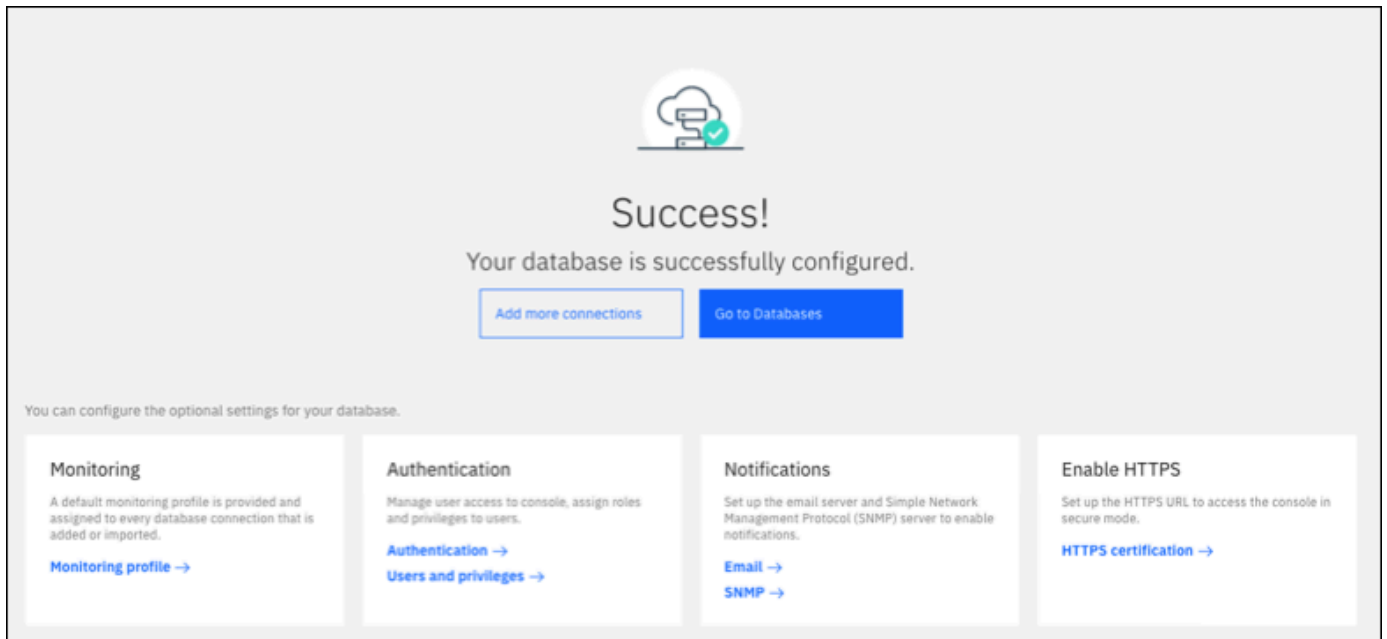
Password:

Test connection

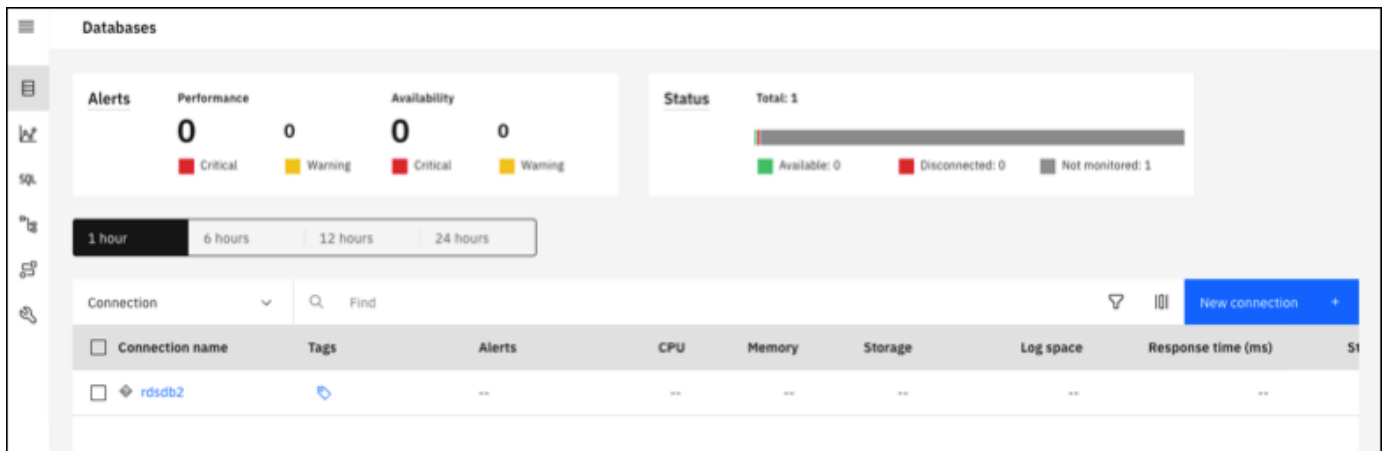
Skip

Save

新增連線之後，會出現類似下列內容的視窗。此視窗指出您的資料庫已成功設定。



5. 選擇「移至資料庫」。將會顯示類似下列內容的「資料庫」視窗。此視窗是顯示量度、狀態和連線的儀表板。

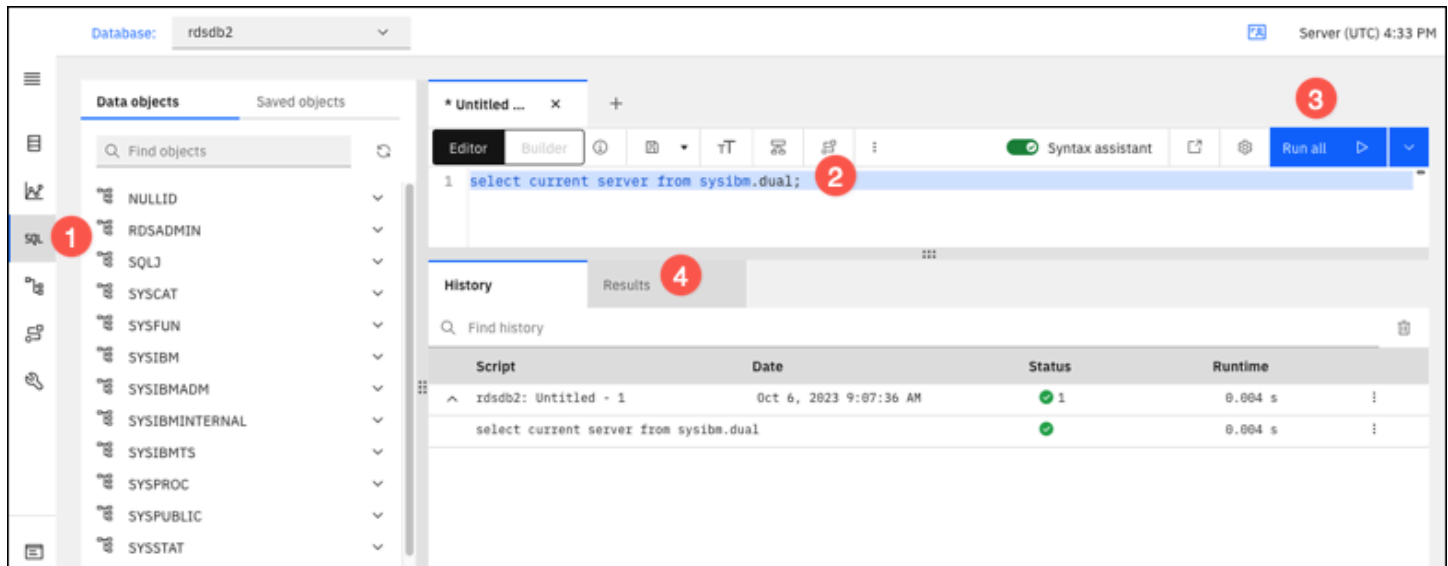


您現在可以開始使用IBM Db2 Data Management Console來執行下列類型的工作：

- 管理適用於 Db2 資料庫執行個體的多個 RDS。
- 執行 SQL 命令。
- 探索、建立或變更資料和資料庫物件。
- 在 EXPLAIN PLAN SQL 中創建語句。
- 調整查詢。

若要執行 SQL 命令並檢視結果

1. 在左側導覽列中，選擇 [SQL]。
2. 輸入 SQL 指令。
3. 選擇 [全部執行]。
4. 若要檢視結果，請選擇「結果」標籤。



使用適用於 Db2 的 Amazon RDS 安全群組的注意事項

若要連線到 Amazon RDS for Db2 資料庫執行個體，它必須與包含必要 IP 地址和網路組態的安全群組產生關聯。您的 RDS for Db2 資料庫執行個體可能會使用預設安全性群組。如果您在建立 RDS for Db2 資料庫執行個體時指派了預設的非設定安全群組，則防火牆會阻止網際網路連線。如需建立新安全群組的詳細資訊，請參閱[使用安全群組控制存取](#)。

建立新安全群組之後，您可以修改資料庫執行個體，讓它與該安全群組建立關聯。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

您可以使用 SSL 來加密對資料庫執行個體的連線，藉此增強安全性。如需更多詳細資訊，請參閱[將 SSL/TLS 與 Amazon RDS 用於 Db2 資料庫執行個體搭配使用](#)。

保護 Amazon RDS for Db2 資料庫執行個體連線

適用於 Db2 的 Amazon RDS 支援改善適用於 Db2 資料庫執行個體之 RDS 安全性的方法。

主題

- [將 SSL/TLS 與 Amazon RDS 用於 Db2 資料庫執行個體搭配使用](#)
- [使用適用於 Db2 的 Amazon RDS 的 Kerberos 身份驗證](#)

將 SSL/TLS 與 Amazon RDS 用於 Db2 資料庫執行個體搭配使用

SSL 是業界標準的通訊協定，用於保護用戶端與伺服器之間的網路連線。在 SSL 3.0 版之後，名稱已更改為 TLS，但我們仍然經常將該協議稱為 SSL。Amazon RDS 支援適用於 Db2 資料庫執行個體的 SSL 加密。您可以使用 SSL/TLS 來加密應用程式用戶端和 RDS for Db2 資料庫執行個體之間的連線。適用於 Db2 版 RDS AWS 區域的所有支援均提供 SSL/TLS 支援。

若要啟用 RDS for Db2 資料庫執行個體的 SSL/TLS 加密，請將 Db2 SSL 選項新增至與資料庫執行個體相關聯的參數群組。Amazon RDS 會根據 Db2 的要求，使用第二個連接埠進行 SSL/TLS 連線。這樣做可讓資料庫執行個體和 Db2 用戶端之間同時進行純文字和 SSL 加密的通訊。例如，您可以使用連接埠搭配純文字通訊與 VPC 內的其他資源通訊，同時使用連接埠搭配 SSL 加密的通訊與 VPC 外部的資源通訊。

主題

- [建立一個 SSL/TLS 連線](#)
- [Connect 至您的 Db2 資料庫伺服器](#)

建立一個 SSL/TLS 連線

若要建立 SSL/TLS 連線，請選擇憑證授權單位 (CA)，為所有人下載憑證服務包 AWS 區域，然後將參數新增至自訂參數群組。

步驟 1：選擇 CA 並下載憑證

選擇一個憑證授權單位 (CA)，並為所有人下載憑證套裝軟體 AWS 區域。如需詳細資訊，請參閱。

步驟 2：更新自訂參數群組中的參數

Important

如果您正在使用 RDS for Db2 的自帶授權 (BYOL) 模型，請修改為您和您 IBM Customer ID 的 IBM Site ID。如果您為 RDS for Db2 使用不同的授權模式，請依照程序將參數新增至自訂參數群組。如需詳細資訊，請參閱 [Amazon 適用於 Db2 的授權選項](#)。

您無法修改 Db2 資料庫執行個體之 RDS 的預設參數群組。因此，您必須建立自訂參數群組、對其進行修改，然後將其附加到 RDS for Db2 資料庫執行個體。如需參數群組的詳細資訊，請參閱 [在資料庫執行個體中使用資料庫參數群組](#)。

使用下表中的參數設定。

參數	Value
DB2COMM	TCPIP,SSL
SSL_SVCENAME	<any port number except the number used for the non-SSL port>

更新自訂參數群組中參數的步驟

1. 透過執行 [create-db-parameter-group](#) 指令建立自訂參數群組。

包含下列必要選項：

- `--db-parameter-group-name`— 您要建立之參數群組的名稱。
- `--db-parameter-group-family`— Db2 引擎版本和主要版本。有效值：db2-se-11-5、db2-ae-11.5。
- `--description`— 此參數群組的描述。

如需建立資料庫參數群組的詳細資訊，請參閱 [建立資料庫參數群組](#)。

2. 修改透過執行 [modify-db-parameter-group](#) 指令建立的自訂參數群組中的參數。

包含下列必要選項：

- `--db-parameter-group-name`— 您建立的參數群組名稱。
- `--parameters`— 參數名稱、值以及參數更新的應用程式方法陣列。

如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

3. 將參數群組與您的 RDS for Db2 資料庫執行個體建立關聯。如需詳細資訊，請參閱[將資料庫參數群組與資料庫執行個體建立關聯](#)。

Connect 至您的 Db2 資料庫伺服器

連接到 Db2 資料庫伺服器的指示是特定於語言的。

Java

若要使用連線至 Db2 資料庫伺服器 Java

1. 下載 JDBC 驅動程式。如需詳細資訊，請參閱[Sup IBM port 文件中的 DB2 JDBC 驅動程式版本與下載](#)。
2. 使用下列內容建立殼層指令碼檔案。此指令碼會將套裝軟體中的所有憑證新增至 Java KeyStore。

Important

確認指令碼中的路徑上是否 `keytool` 存在，以便指令碼可以找到它。如果您使用 Db2 用戶端，您可以找到 `keytool` 下 `~sqlib/java/jdk64/jre/bin` 的。

```
#!/bin/bash
PEM_FILE=$1
PASSWORD=$2
KEYSTORE=$3
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE | wc -l)
for N in $(seq 0 $((CERTS - 1))); do
  ALIAS="${PEM_FILE%.*}-$N"
  cat $PEM_FILE |
  awk "n==$N { print }; /END CERTIFICATE/ { n++ }" |
  keytool -noprompt -import -trustcacerts -alias $ALIAS -keystore $KEYSTORE -
  storepass $PASSWORD
```

```
done
```

- 若要執行 shell 指令碼，並將具有憑證服務包的 PEM 檔案匯入到 Java KeyStore，請執行下列命令。將 `shell_file_name.sh` 取代為您的殼層指令碼檔案的名稱和 `##`，並使用您的 Java KeyStore。

```
./shell_file_name.sh global-bundle.pem password truststore.jks
```

- 若要連線至您的 Db2 伺服器，請執行下列命令。將範例中的下列預留位置取代為您的 RDS for Db2 資料庫執行個體資訊。
 - `ip_address` — 資料庫執行個體端點的 IP 位址。
 - 連接 `#` — SSL 連線的連接埠號碼。這可以是除了用於非 SSL 連接埠的號碼以外的任何通訊埠號碼。
 - 資料 `###` — 資料庫執行個體中資料庫的名稱。
 - `master #####` — 資料庫執行個體的主使用者名稱。
 - `###` — 資料庫執行個體的主要密碼。

```
export trustStorePassword=MyPassword
java -cp ~/dsdriver/jdbc_sqlj_driver/linuxamd64/db2jcc4.jar \
com.ibm.db2.jcc.DB2Jcc -url \
"jdbc:db2://ip_address:port/database_name:\
sslConnection=true;sslTrustStoreLocation=\
~/truststore.jks;\
sslTrustStorePassword=${trustStorePassword};\
sslVersion=TLSv1.2;\
encryptionAlgorithm=2;\
securityMechanism=7;" \
-user master_username -password master_password
```

Node.js

若要使用連線至 Db2 資料庫伺服器 Node.js

- 安裝 `node-ibm_db` 驅動程式。如需詳細資訊，請參閱說明文件 [中的在 Linux 和 UNIX 系統上安裝 node-ibm_db 驅動程式](#)。IBM Db2
- 根據下列內容建立 JavaScript 檔案。將範例中的下列預留位置取代為您的 RDS for Db2 資料庫執行個體資訊。

- *ip_address* — 資料庫執行個體端點的 IP 位址。
- *master #####* — 資料庫執行個體的主使用者名稱。
- *###* — 資料庫執行個體的主要密碼。
- *資料###* — 資料庫執行個體中資料庫的名稱。
- *連接#* — SSL 連線的連接埠號碼。這可以是除了用於非 SSL 連接埠的號碼以外的任何通訊埠號碼。

```
var ibmdb = require("ibm_db");
const hostname = "ip_address";
const username = "master_username";
const password = "master_password";
const database = "database_name";
const port = "port";
const certPath = "/root/qa-bundle.pem";
ibmdb.open("DRIVER={DB2};DATABASE=" + database + ";HOSTNAME=" +
  hostname + ";UID=" + username + ";PWD=" + password + ";PORT=" + port +
  ";PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=" + certPath + ";", function
  (err, conn){
  if (err) return console.log(err);
  conn.close(function () {
  console.log('done');
  });
});
```

3. 若要執行 JavaScript 檔案，請執行下列命令。

```
node ssl-test.js
```

Python

若要使用連線至 Db2 資料庫伺服器 Python

1. 創建具有以下內容的 Python 文件。將範例中的下列預留位置取代為您的 RDS for Db2 資料庫執行個體資訊。
 - *連接#* — SSL 連線的連接埠號碼。這可以是除了用於非 SSL 連接埠的號碼以外的任何通訊埠號碼。

- *master #####* — 資料庫執行個體的主使用者名稱。
- *###* — 資料庫執行個體的主要密碼。
- *資料###* — 資料庫執行個體中資料庫的名稱。
- *ip_address* — 資料庫執行個體端點的 IP 位址。

```

import click
import ibm_db
import sys

port = port;
master_user_id = "master_username" # Master id used to create your DB instance
master_password = "master_password" # Master password used to create your DB
instance
db_name = "database_name" # If not given "db-name"
vpc_customer_private_ip = "ip_address" # Hosts end points - Customer private IP
Addressicert_path = "/root/ssl/global-bundle.pem" # cert path

@click.command()
@click.option("--path", help="certificate path")
def db2_connect(path):

    try:
        conn =
        ibm_db.connect(f"DATABASE={db_name};HOSTNAME={vpc_customer_private_ip};PORT={port};
        PROTOCOL=TCPIP;UID={master_user_id};PWD={master_password};SECURITY=ssl;SSLServerCertifi
        "", "")
        try:
            ibm_db.exec_immediate(conn, 'create table tablename (a int);')
            print("Query executed successfully")
        except Exception as e:
            print(e)
        finally:
            ibm_db.close(conn)
            sys.exit(1)
    except Exception as ex:
        print("Trying to connect...")

if __name__ == "__main__":
    db2_connect()

```

2. 創建以下 shell 腳本，該腳本運行您創建的Python文件。替換為 *python_file_name.py* Python腳本文件的名稱。

```
#!/bin/bash
PEM_FILE=$1
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)

for N in $(seq 0 $((CERTS - 1))); do
  ALIAS="{PEM_FILE%.*}-$N"
  cert=`cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }"`
  cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }" >
  $ALIAS.pem
  python3 <python_file_name.py> --path $ALIAS.pem
  output=`echo $?`
  if [ $output == 1 ]; then
    break
  fi
done
```

3. 若要使用憑證服務包匯入PEM檔案並執行 shell 指令碼，請執行下列命令。將 *shell_file_name.sh* 取代為您的殼層指令碼檔案的名稱。

```
./shell_file_name.sh global-bundle.pem
```

使用適用於 Db2 的 Amazon RDS 的Kerberos身份驗證

當使用者連線到 Amazon RDS for Db2 資料庫執行個體時，您可以使用身分Kerberos驗證來驗證使用者。您的資料庫執行個體與 AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) 搭配使用以啟用Kerberos驗證。當使用者透過加入信任網域的 RDS for Db2 資料庫執行個體進行驗證時，驗證要求會轉寄至您使用建立的目錄。AWS Directory Service如需詳細資訊，請參閱《AWS Directory Service 使用者指南》中的[什麼是 AWS Directory Service ?](#)。

首先，建立儲存使用者認證的 AWS Managed Microsoft AD 目錄。然後，將 AWS Managed Microsoft AD 目錄的網域和其他資訊新增至您的 RDS for Db2 資料庫執行個體。當使用者透過 RDS for Db2 資料庫執行個體進行驗證時，驗證要求會轉送至目 AWS Managed Microsoft AD 錄。

將您的所有登入資料保留在相同目錄可以節省您的時間和精力。透過這種方式，這樣您就有一個集中的位置來存放及管理多個資料庫執行個體的登入資料。使用目錄也可以改善您的整體安全性描述檔。

主題

- [區域和版本可用性](#)
- [適用於 Db2 資料庫執行個體的 RDS Kerberos 驗證概觀](#)
- [為 Db2 資料庫執行個體的 RDS 設定 Kerberos 驗證](#)
- [管理網域中的資料庫執行個體](#)
- [使用驗證連線至適用於 Db2 的 Kerberos RDS](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需有關具有 Kerberos 驗證之 RDS for Db2 的版本和區域可用性的詳細資訊，請參閱[Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎](#)。

Note

Kerberos 針對 Db2 資料庫執行個體已取代 RDS 的資料庫執行個體類別，不支援驗證。如需詳細資訊，請參閱 [Amazon RDS 適用於 Db2 執行個體類別](#)。

適用於 Db2 資料庫執行個體的 RDS Kerberos 驗證概觀

若要設定 RDS for Db2 資料庫執行個體的 Kerberos 驗證，請完成下列一般步驟，稍後將詳細說明：

1. 用 AWS Managed Microsoft AD 於建立 AWS Managed Microsoft AD 目錄。您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS Directory Service 來建立目錄。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[建立 AWS Managed Microsoft AD 目錄](#)。
2. 建立使用受管 IAM 政策的 AWS Identity and Access Management (IAM) 角色 AmazonRDSDirectoryServiceAccess。IAM 角色可讓 Amazon RDS 對您的目錄撥打電話。

若要讓 IAM 角色允許存取，AWS Security Token Service (AWS STS) 端點必須以正確 AWS 區域的 AWS 帳戶。AWS STS 端點預設在所有端點都處於作用中狀態 AWS 區域，您可以使用它們，而無需任何進一步的動作。如需詳細資訊，請參閱《IAM 使用者指南》[AWS STS AWS 區域中的「啟用和停用」](#)。

3. 使用、或 RDS API 搭配下列其中一種方法 AWS Management Console，建立或修改 RDS for Db2 資料庫執行個體：AWS CLI

- 使用主控台、[create-db-instance](#)命令或建立資料庫執行個體 API 作業，[建立新的 Db2 for Db2 資料庫執行個體](#)。如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[modify-db-instance](#)命令或 [ModifyDBInstance](#)API 作業修改現有的 Db2 資料庫執行個體 RDS。如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[restore-db-instance-from-db-snapshot](#)命令或 [RestoreDBInstanceFromDBSnapshot](#)API 作業，從資料庫快照還原 RDS for Db2 資料庫執行個體。如需說明，請參閱[從資料庫快照還原](#)。
- point-in-time 使用主控台、[restore-db-instance-to-point-in-time](#)命令或 [RestoreDBInstanceToPointInTime](#)API 作業將 RDS for Db2 資料庫執行個體還原至。如需說明，請參閱[將資料庫執行個體還原至指定的時間](#)。

您可以在與目錄相同的 Amazon Virtual Private Cloud (VPC) 中找到資料庫執行個體，也可以在不 AWS 帳戶 同的或 VPC 中找到該資料庫執行個體。當您建立或修改 Db2 資料庫執行個體的 RDS 時，請執行下列工作：

- 請提供您建立目錄時產生的網域識別符 (d-* 識別符)。
 - 請提供所建立的 IAM 角色名稱。
 - 確認資料庫執行個體安全群組可以接收來自目錄安全群組的輸入流量。
4. 設定您的 Db2 用戶端，並確認流量可以在用戶端主機和 AWS Directory Service 下列連接埠之間流動：
- TCP/UDP 連接埠 53 — DNS
 - 88-Kerberos 認證
 - TCP 389-LDAP
 - 464 — 驗證 Kerberos

為 Db2 資料庫執行個體的 RDS 設定 Kerberos 驗證

您可以使用 AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) 設定 RDS for Db2 資料庫執行個體的 Kerberos 驗證。若要設定 Kerberos 驗證，請依照下列步驟執行：

主題

- [第 1 步：使用創建一個目錄 AWS Managed Microsoft AD](#)
- [步驟 2：為 Amazon RDS 創建 IAM 角色以訪問 AWS Directory Service](#)
- [步驟 3：建立和設定使用者](#)
- [步驟 4：在中建立適用於 Db2 系統管理群組的 RDS AWS Managed Microsoft AD](#)

- [步驟 5：建立或修改適用於 Db2 資料庫執行個體的 RDS](#)
- [步驟 6：設定 Db2 用戶端](#)

第 1 步：使用創建一個目錄 AWS Managed Microsoft AD

AWS Directory Service 會在中建立完全 Active Directory 受管理的 AWS 雲端。當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會為您建立兩個網域控制站和 DNS 伺服器。目錄伺服器是在 VPC 的不同子網路中建立。此備援可協助確保即使失敗，仍能存取您的目錄。

建立目 AWS Managed Microsoft AD 錄時，請代表您 AWS Directory Service 執行下列工作：

- Active Directory 在您的 VPC 中設置一個。
- 建立含有使用者名稱 Admin 與指定密碼的目錄管理員帳戶。您可以使用此帳戶來管理目錄。

Important

確保保存此密碼。AWS Directory Service 不會儲存此密碼，也無法擷取或重設密碼。

- 建立目錄控制器的安全群組。安全性群組必須允許與 RDS for Db2 資料庫執行個體進行通訊。

啟動時 AWS Directory Service for Microsoft Active Directory，AWS 會建立包含所有目錄物件的組織單位 (OU)。此 OU 有您在建立目錄時所輸入的 NetBIOS 名稱，位於根網域中。網域根目錄擁有及管理 AWS。

使用您的 AWS Managed Microsoft AD 目錄建立的 Admin 帳戶具有 OU 最常見系統管理活動的權限：

- 建立、更新或刪除使用者。
- 將資源新增至您的網域 (例如檔案或列印伺服器)，然後將這些資源的權限指派給 OU 中的使用者。
- 建立額外的 OU 和容器。
- 委派授權。
- 從 Active Directory 資源回收筒還原已刪除的物件。
- Windows PowerShell 在上運行 Active Directory 和域名服務 (DNS) 模塊 AWS Directory Service。

Admin 帳戶也有權執行下列全網域活動：

- 管理 DNS 組態 (新增、移除或更新記錄、區域和轉寄站)。

- 檢視 DNS 事件日誌。
- 檢視安全事件日誌。

若要建立目錄 AWS Managed Microsoft AD

1. 請登入 AWS Management Console 並開啟 AWS Directory Service 主控台，網址為 <https://console.aws.amazon.com/directoryservicev2/>。
2. 選擇「設定目錄」。
3. 選擇AWS Managed Microsoft AD。AWS Managed Microsoft AD 是目前支援與 Amazon RDS 搭配使用的唯一選項。
4. 選擇 Next (下一步)。
5. 在 Enter directory information (輸入目錄資訊) 頁面上，提供下列資訊：
 - 版本 — 選擇符合您需求的版本。
 - 目錄 DNS 名稱 — 目錄的完整名稱，例如corp.example.com。
 - 目錄 NetBIOS 名稱 — 目錄的選擇性簡稱，例如CORP。
 - 目錄說明 — 目錄的選擇性描述。
 - 管理員密碼 — 目錄管理員的密碼。目錄建立程序會使用使用者名稱Admin和此密碼建立管理員帳戶。

目錄管理員密碼不得包含 "admin" 字組。密碼區分大小寫，長度須為 8 至 64 個字元。至少須有一位字元屬於以下四種類型中的三類：

- 小寫字母 (a-z)
- 大寫字母 (A-Z)
- 數字 (0-9)
- 非英數字元 (~!@#\$%^&* _+=`|\(){}[]:;'"<>.,?/)
- 確認密碼 — 重新輸入管理員密碼。

Important

請確定您已儲存此密碼。AWS Directory Service 不會儲存此密碼，也無法擷取或重設密碼。

6. 選擇 Next (下一步)。
7. 在 Choose VPC and subnets (選擇 VPC 和子網路) 頁面上，提供下列資訊：

- VPC — 選擇目錄的 VPC。您可以在這個相同的 VPC 或不同的 VPC 中建立適用於 Db2 資料庫執行個體的 RDS。
 - 子網路 — 選擇目錄伺服器的子網路。這兩個子網路必須位於不同的可用區域。
8. 選擇 Next (下一步)。
 9. 檢閱目錄資訊。如果需要變更，請選擇 Previous (上一步)，然後進行變更。若資訊無誤，請選擇 Create directory (建立目錄)。

Review & create [info](#)

Review

Directory type Microsoft AD	VPC vpc-0d6c7cf411cf1e4e2 ()
Operating system version Windows Server 2019	Subnets RDS-Pvt-subnet-4 subnet-0d7ee6515db17b7a4 (us-west-2d) RDS-Pvt-subnet-1 subnet-0ffff968223abe72a (us-west-2a)
Directory DNS name corp.example.com	
Directory NetBIOS name CORP	
Directory description My directory	

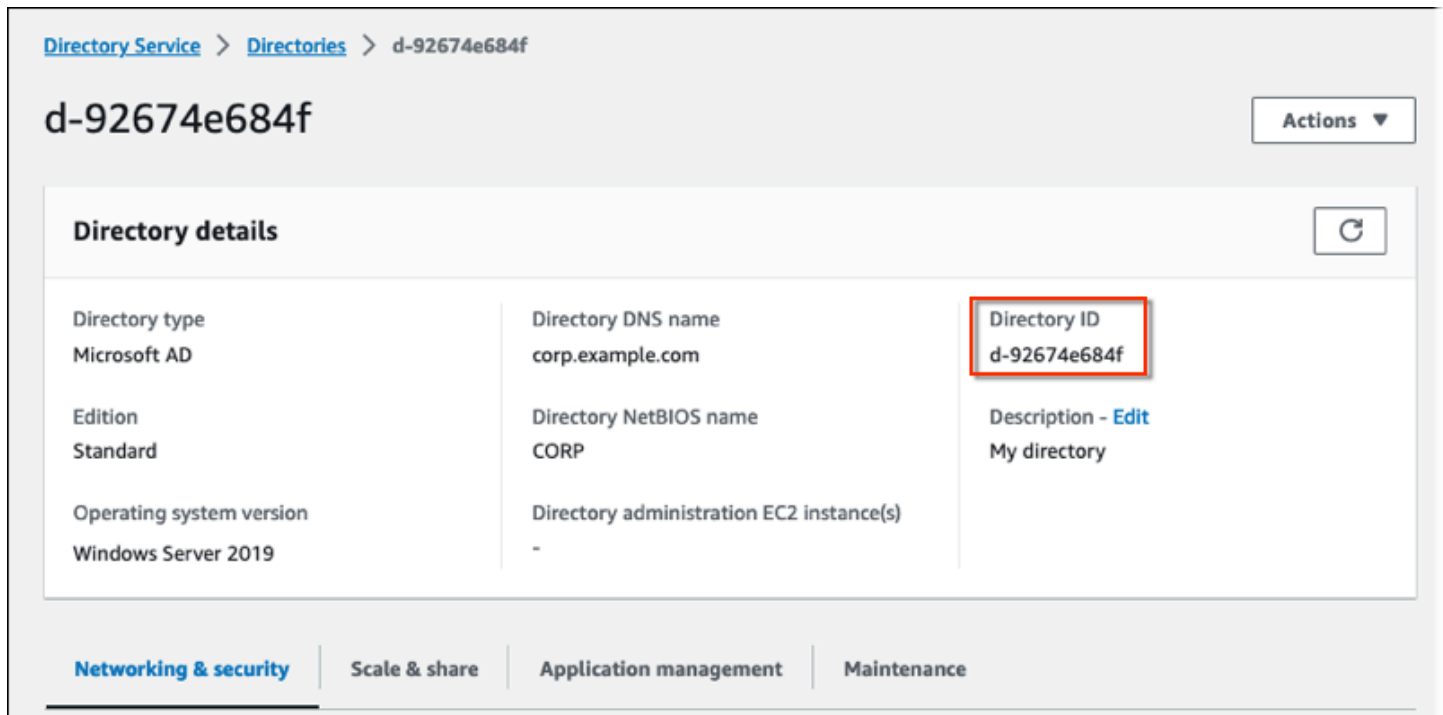
Pricing

Edition Standard	Free trial eligible Learn more ↗ 30-day limited trial
Domain controllers charge ~USD ()*	
* Includes two domain controllers, USD /mo for each additional domain controller.	

Cancel

建立目錄需要幾分鐘的時間。成功建立時，Status (狀態) 值會變更為 Active (作用中)。

若要查看目錄的相關資訊，請在「目錄 ID」下選擇目錄 ID。請記下 Directory ID (目錄 ID) 值，建立或修改 RDS for Db2 資料庫執行個體時，需要此值。



The screenshot shows the AWS Directory Service console for a directory with ID 'd-92674e684f'. The 'Directory details' section is expanded, showing the following information:

Directory type	Microsoft AD	Directory DNS name	corp.example.com	Directory ID	d-92674e684f
Edition	Standard	Directory NetBIOS name	CORP	Description - Edit	My directory
Operating system version	Windows Server 2019	Directory administration EC2 instance(s)	-		

At the bottom of the console, there are tabs for 'Networking & security', 'Scale & share', 'Application management', and 'Maintenance'. The 'Directory ID' field is highlighted with a red box.

步驟 2：為 Amazon RDS 創建 IAM 角色以訪問 AWS Directory Service

若要讓 Amazon RDS 透過 AWS Directory Service 為您撥打電話，您的 AWS 帳戶需要使用受管身分與存取權管理政策的 IAM 角色 `AmazonRDSDirectoryServiceAccess`。此角色允許 Amazon RDS 撥打電話給 AWS Directory Service。

當您使用建立資料庫執行個體，AWS Management Console 且主控台使用者帳戶具有 `iam:CreateRole` 權限時，主控台會自動建立所需的 IAM 角色。在此情況下，角色名稱為 `rds-directoryservice-kerberos-access-role`。否則，您必須手動建立 IAM 角色。建立此 IAM 角色時，請選擇 `Directory Service`，請選擇 AWS 受管政策並將其附加 `AmazonRDSDirectoryServiceAccess` 到該角色。

如需為服務建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以將許可委派給 AWS 服務](#)。

Note

適用於 RDS 的 Windows 驗證的身分與存取權管理角色 `Microsoft SQL Server` 無法用於適用於 Db2 的 RDS。

作為使用 AmazonRDSDirectoryServiceAccess 受管政策的替代方案，您可以建立具有必要許可的政策。在此情況下，IAM 角色必須具有下列 IAM 信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

該角色還必須具有以下 IAM 角色政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

步驟 3：建立和設定使用者

您可以使用 Active Directory Users and Computers 工具建立使用者。這是 Active Directory Domain Services 和 Active Directory Lightweight Directory Services 工具之一。如需詳細資訊，請參閱 Microsoft

文件中的[將使用者和電腦新增至Active Directory網域](#)。在這種情況下，使用者是個人或其他實體（例如他們的電腦），這些實體屬於網域的一部分，且其識別會維護在目錄中。

若要在 AWS Directory Service 目錄中建立使用者，您必須連線至 AWS Directory Service 目錄成員 Windows 的 Amazon EC2 執行個體。同時，您必須以具有建立使用者權限的使用者身分登入。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[建立使用者](#)。

步驟 4：在中建立適用於 Db2 系統管理群組的 RDS AWS Managed Microsoft AD

適用於 Db2 的 RDS 不支援 Kerberos 主要使用者或兩個 Amazon RDS 保留使用者 `rdsdb` 和 `rdsadmin`。相反地，您需要建立一個名為 `masterdba` 的新群組 AWS Managed Microsoft AD。如需詳細資訊，請參閱 Microsoft 文件 [Active Directory 中的建立群組帳戶](#)。您新增至此群組的任何使用者都將擁有主要使用者權限。

啟用 Kerberos 驗證之後，主要使用者會失去 `masterdba` 角色。因此，除非您停用驗證，否則主要使用者將無法存取執行個體本機使用 Kerberos 者群組成員資格。若要透過密碼登入繼續使用主要使用者，請在上建立 AWS Managed Microsoft AD 與主要使用者名稱相同的使用者。然後，將該使用者新增至群組 `masterdba`。

步驟 5：建立或修改適用於 Db2 資料庫執行個體的 RDS

建立或修改 RDS for Db2 資料庫執行個體，以搭配您的目錄使用。您可以使用 AWS Management Console AWS CLI、或 RDS API 將資料庫執行個體與目錄建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) 命令或 [CreateDBInstance](#) API 作業為 Db2 資料庫執行個體建立新的 RDS for Db2 執行個體。如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[modify-db-instance](#) 命令或修改資料庫執行個體 API 作業，修改現有的 [Db2](#) 資料庫執行個體 RDS。如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[restore-db-instance-from-db-snapshot](#) 命令或 [RestoreDBInstanceFromDBSnapshot](#) API 作業，從資料庫快照還原 RDS for Db2 資料庫執行個體。如需說明，請參閱[從資料庫 快照還原](#)。
- `point-in-time` 使用主控台、[restore-db-instance-to-point-in-time](#) 命令或 [RestoreDBInstanceToPointInTime](#) API 作業將 RDS for Db2 資料庫執行個體還原至。如需說明，請參閱[將資料庫執行個體還原至指定的時間](#)。

Kerberos 僅針對 VPC 中 Db2 資料庫執行個體的 RDS 支援驗證。資料庫執行個體可在與目錄相同的 VPC 中，或在不同 VPC 中。資料庫執行個體必須使用允許目錄 VPC 內輸入和輸出的安全群組，以便資料庫執行個體可以與目錄通訊。

主控台

使用主控台建立、修改或還原資料庫執行個體時，請在 [資料庫Kerberos驗證] 區段中選擇 [密碼和驗證]。然後選擇 Browse Directory (瀏覽目錄)。選取目錄或選擇建立目錄以使用「Directory Service」。

Database authentication

Database authentication options [Info](#)

- Password authentication
Authenticates using database passwords.
- Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Directory

corp.example.com (d-92674e684f) [Browse Directory](#)

AWS CLI

使用時 AWS CLI，資料庫執行個體需要下列參數才能使用您建立的目錄：

- 對於 `--domain` 參數，請使用建立目錄時產生的網域識別碼 (「d-* "ID)。
- 針對 `--domain-iam-role-name` 參數，使用您建立的規則，其會使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess`。

下列範例會修改資料庫執行個體以使用目錄。以您自己的值取代範例中的下列預留位置：

- 資料庫 `## - ### Db2 #####` 行個體的 RDS 名稱。
- 目 `# ID` - 您所建立之 AWS Directory Service for Microsoft Active Directory 目錄的識別碼。
- `#####` - 您建立的 IAM 角色名稱。

```
aws rds modify-db-instance --db-instance-identifier db_instance_name --domain
d-directory_id --domain-iam-role-name role_name
```

⚠ Important

如果您修改資料庫執行個體以啟用Kerberos驗證，請在進行變更後重新啟動資料庫執行個體。

步驟 6：設定 Db2 用戶端

若要設定 Db2 用戶端

1. 建立 `/etc/krb5.conf` 檔案 (或同等檔案) 以指向網域。

📘 Note

對於視窗作業系統，請建立 `C:\windows\krb5.ini` 檔案。

2. 確認流量可以在用戶端主機和之間流動 AWS Directory Service。請使用網路公用程式，Netcat例如執行下列工作：
 - a. 確認透過 DNS 傳送至連接埠 53 的流量。
 - b. 驗證透過 TCP/UDP 的通訊埠 53 和用於的流量Kerberos，其中包括 AWS Directory Service
3. 確定流量可透過資料庫連接埠在用戶端主機和資料庫執行個體之間往來。您可以使用該命令`db2`來連接和訪問數據庫。

下面的例子是 `/etc/krb5.conf` 的文件內容：AWS Managed Microsoft AD

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
kdc = example.com
admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```


管理網域中的資料庫執行個體

您可以使用 AWS Management Console、AWS CLI、或 RDS API 來管理您的資料庫執行個體及其與您的 Microsoft Active Directory。例如，您可以關聯 Active Directory 以啟用 Kerberos 驗證。您也可以移除關聯以停用 Active Directory 用 Kerberos 驗證。您也可以將資料庫執行個體移至另一個執行個體 Microsoft Active Directory 進行外部驗證。

例如，您可以使用 [modify-db-instance](#) CLI 命令執行下列動作：

- 透過為 `--domain` 選項指定目前成員資格的目錄 ID，重新嘗試啟用失敗成員資格的 Kerberos 驗證。
- 透過指定 `none` `--domain` 選項來停用資料庫執行個體的 Kerberos 驗證。
- 透過為 `--domain` 選項指定新網域的網域識別碼，將資料庫執行個體從一個網域移至另一個網域。

了解網域成員資格

在您建立或修改資料庫執行個體之後，該執行個體會成為網域的成員。您可以在主控台中或執行 [describe-db-instances](#) 命令來檢視網域成員資格的狀態。資料庫執行個體的狀態可以是下列其中一個：

- `kerberos-enabled`— 資料庫執行個體已啟用 Kerberos 驗證。
- `enabling-kerberos`— AWS 正在啟用此資料庫執行個體上的 Kerberos 驗證程序。
- `pending-enable-kerberos`— 在此資料庫執行個體上啟用 Kerberos 驗證擱置中。
- `pending-maintenance-enable-kerberos`— AWS 將嘗試在下一個排定的維護時段期間啟用資料庫執行個體 Kerberos 驗證。
- `pending-disable-kerberos`— 在此資料庫執行個體上停用 Kerberos 驗證擱置中。
- `pending-maintenance-disable-kerberos`— AWS 將嘗試在下一個排定的維護時段期間停用資料庫執行個體的 Kerberos 驗證。
- `enable-kerberos-failed`— 無法在資料庫執行個體 AWS 體上啟用 Kerberos 驗證的組態問題。在重新發出修改資料庫執行個體的指令之前，請先修正組態問題。
- `disabling-kerberos`— AWS 正在停用此資料庫執行個體上的 Kerberos 驗證程序。

啟用 Kerberos 身份驗證的請求可能會因為網路連線問題或 IAM 角色不正確而失敗。在某些情況下，當您建立或修改資料庫執行個體時，嘗試啟用 Kerberos 驗證可能會失敗。如果發生這種情況，請確認您使用的是正確的 IAM 角色，然後修改資料庫執行個體以加入網域。

使用驗證連線至適用於 Db2 的 Kerberos RDS

使用驗證連線至適用於 Db2 的 Kerberos RDS

1. 畫面出現命令提示時，執行下列命令。在下面的例子中，用您的###替換用Microsoft Active Directory戶名。

```
kinit username
```

2. 如果 RDS for Db2 資料庫執行個體使用可公開存取的 VPC，請將資料庫執行個體端點的 IP 位址新增到 Amazon EC2 用戶端上的/etc/hosts檔案。下列範例會取得 IP 位址，然後將其新增至/etc/hosts檔案。

```
% dig +short Db2-endpoint.AWS-Region.rds.amazonaws.com  
;; Truncated, retrying in TCP mode.  
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.  
34.210.197.118
```

```
% echo "34.210.197.118 Db2-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
```

3. 使用下列命令登入與相關聯的 RDS for Db2 資料庫執行個體Active Directory。將#####取代為您的 RDS for Db2 資料庫的名稱。

```
db2 connect to database_name
```

管理適用於 Db2 資料庫執行個體的 Amazon RDS

本主題涵蓋您使用 Amazon RDS for Db2 資料庫執行個體執行的常見管理任務。所有 Amazon RDS 資料庫執行個體的某些任務都相同。其他工作是適用於 Db2 的 RDS 專屬工作。

下列工作是所有 RDS 資料庫通用的工作。此外，還有適用於 Db2 的 RDS 專用工作，例如使用標準 SQL 用戶端連線至 RDS for Db2 資料庫。

任務區域	相關文件
<p>執行個體類別、儲存體和 PIOPS</p> <p>如果您要建立生產執行個體，請了解執行個體類別、儲存體類型和佈建 IOPS 在 Amazon RDS 中的運作方式。</p>	<p>資料庫執行個體類別</p> <p>Amazon RDS 儲存類型</p>
<p>異地同步備份部署</p> <p>生產資料庫執行個體應該使用多個可用區部署。「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。</p>	<p>設定及管理多可用區部署</p>
<p>Amazon VPC</p> <p>如果您 AWS 帳戶有預設的虛擬私有雲 (VPC)，則會在預設 VPC 內自動建立資料庫執行個體。如果您的帳戶沒有預設的 VPC，而您想要 VPC 中有該資料庫執行個體，則必須在建立資料庫執行個體之前，先建立 VPC 和子網路群組。</p>	<p>在 VPC 中使用資料庫執行個體</p>
<p>安全群組</p> <p>根據預設，資料庫執行個體會使用防火牆來防止存取。確定您以正確的 IP 地址和網路組態建立安全群組，來存取資料庫執行個體。</p>	<p>使用安全群組控制存取</p>
<p>參數群組</p> <p>由於 RDS for Db2 資料庫執行個體需要新增 <code>rds.ibm_customer_id</code> 和 <code>rds.ibm_site_id</code> 參數，因此請先建立參數群組，然後再建立資料庫執行個體。如果您的資料庫執行個體</p>	<p>將 IBM 識別碼新增至 Db2 資料庫執行個體之 RDS 的參數群組</p> <p>使用參數群組</p>

任務區域	相關文件
<p>需要其他特定資料庫參數，請在建立資料庫執行個體之前，將它們新增至此參數群組。</p>	
<p>選項群組</p> <p>如果您的資料庫執行個體要求特定資料庫選項，則在建立資料庫執行個體之前，須先建立選項群組。</p>	<p>適用於 Db2 資料庫執行個體的 Amazon RDS 選項</p>
<p>連線到您的資料庫執行個體</p> <p>建立安全群組並將其與資料庫執行個體建立關聯後，您可以使用任何標準 SQL 用戶端應用程式 (例如) 連線至資料庫執行個體。IBM Db2 CLP</p>	<p>連接到您的 Amazon RDS 用 Db2 資料庫執行個體</p>
<p>備份和還原</p> <p>您可以將資料庫執行個體設定為進行自動化儲存備份，或建立手動儲存快照，然後從備份或快照還原執行個體。</p>	<p>備份、還原和匯出資料</p>
<p>監控</p> <p>您可以使 IBM Db2 Data Management Console 用來監控適用於 Db2 資料庫執行個體的 RDS。</p> <p>您也可以使用 CloudWatch Amazon RDS 指標、事件和增強型監控來監控 RDS for Db2 資料庫執行個體。</p>	<p>使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 Data Management Console</p> <p>在 Amazon RDS 主控台中檢視指標</p> <p>檢視 Amazon RDS 事件</p> <p>使用增強型監控來監控作業系統指標</p>
<p>日誌檔</p> <p>您可以存取 RDS for Db2 資料庫執行個體的記錄檔。</p>	<p>監控 Amazon RDS 日誌檔案</p>

主題

- [針對適用於 Db2 資料庫執行個體的 Amazon RDS 執行一般系統任務](#)

- [執行適用於 Db2 資料庫執行個體的 Amazon RDS 常見資料庫任務](#)

針對適用於 Db2 資料庫執行個體的 Amazon RDS 執行一般系統任務

您可以在執行 Db2 的 Amazon RDS 資料庫執行個體上，執行與系統相關的某些常見資料庫管理員任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的 Shell 存取權，而且會將存取權限制在某些需要進階權限的系統程序和資料表。

主題

- [建立自訂資料庫端點](#)
- [授與和撤銷權限](#)
- [附加至適用於 Db2 資料庫執行個體的遠端 RDS](#)

建立自訂資料庫端點

當您移轉到 Amazon RDS for Db2 時，您可以使用自訂資料庫端點 URL 將對應用程式的變更降到最低。例如，如果您使用 `db2.example.com` 當前的 DNS 記錄，則可以將其添加到 Amazon 路線 53。在 Route 53 中，您可以使用私有託管區域將目前的 DNS 資料庫端點對應至 RDS for Db2 資料庫端點。若要為 Amazon RDS 資料庫端點新增自訂 A 或 CNAME 記錄，請參閱 Amazon [Route 53 開發人員指南](#) 中的 [使用 Amazon Route 53 註冊和管理網域](#)。

Note

如果您無法將網域轉移到 Route 53，您可以使用 DNS 提供者建立 RDS for Db2 資料庫端點 URL 的 CNAME 記錄。請參閱 DNS 供應商說明文件。

授與和撤銷權限

使用者可透過附加至資料庫之群組中的成員資格來存取資料庫。如果您從使用者移除附加至資料庫的所有群組，則該使用者將無法連線至資料庫。

使用下列程序來授與和撤銷控制資料庫存取權的權限。

這些程序會使用在本機電腦上執行 IBM Db2 CLP 行來連線至 RDS for Db2 資料庫執行個體。請務必對 TCPIP 節點和資料庫進行分類，以連線到本機電腦上執行的 RDS for Db2 資料庫執行個體。如需詳細資訊，請參閱 [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 CLP](#)。

主題

- [授與使用者對資料庫的存取權](#)
- [變更使用者的密碼](#)
- [將群組新增至使用者](#)
- [從使用者中移除群組](#)
- [移除使用者](#)
- [列出使用者](#)
- [建立角色](#)
- [授予角色](#)
- [撤銷角色](#)
- [授予資料庫授權](#)
- [撤銷資料庫授權](#)

授與使用者對資料庫的存取權

授與使用者存取您的資料庫

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

此命令會產生類似下列範例的輸出：

```
Database Connection Information

Database server          = DB2/LINUX8664 11.5.8.0
SQL authorization ID    = ADMIN
Local database alias    = RDSADMIN
```

2. 通過調用將用戶添加到您的授權列表中rdsadmin.add_user。如需詳細資訊，請參閱[rdsadmin.add_user](#)。

```
db2 "call rdsadmin.add_user(
      'username',
      'password',
```

```
'group_name,group_name')"
```

3. (選擇性) 透過呼叫將其他群組新增至使用者 `rdsadmin.add_groups`。如需詳細資訊，請參閱 [rdsadmin.add_groups](#)。

```
db2 "call rdsadmin.add_groups(
      'username',
      'group_name,group_name')"
```

4. 確認使用者可用的授權單位。##### `rds_database_alias`##### 另外，將用戶 `#` 替換為用戶的用戶名。

```
db2 terminate
db2 connect to rds_database_alias user master_user using master_password
db2 "SELECT SUBSTR(AUTHORITY,1,20) AUTHORITY, D_USER, D_GROUP, D_PUBLIC
      FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('username', 'U') ) AS
      T
      ORDER BY AUTHORITY"
```

此命令會產生類似下列範例的輸出：

AUTHORITY	D_USER	D_GROUP	D_PUBLIC
ACCESSCTRL	N	N	N
BINDADD	N	N	N
CONNECT	N	N	N
CREATETAB	N	N	N
CREATE_EXTERNAL_ROUT	N	N	N
CREATE_NOT_FENCED_RO	N	N	N
CREATE_SECURE_OBJECT	N	N	N
DATAACCESS	N	N	N
DBADM	N	N	N
EXPLAIN	N	N	N
IMPLICIT_SCHEMA	N	N	N
LOAD	N	N	N
QUIESCE_CONNECT	N	N	N
SECADM	N	N	N
SQLADM	N	N	N
SYSADM	*	N	*
SYSCTRL	*	N	*
SYSMAINT	*	N	*
SYSMON	*	N	*

WLMADM	N	N	N
--------	---	---	---

5. 將 RDS for Db2 角色授與

ROLE_NULLID_PACKAGES、ROLE_TABLESPACES，ROLE_PROCEDURES以及您新增使用者的群組。

Note

我們在RESTRICTIVE模式下為 Db2 資料庫執行個體建立 RDS。因此，適用於 Db2 角色的 RDS ROLE_NULLID_PACKAGES、ROLE_TABLESPACES，並ROLE_PROCEDURES授與和NULLID套件的執行權限。IBM Db2 CLP Dynamic SQL這些角色也會授與使用者表格空間的權限。

- a. Connect 至您的 Db2 資料庫。在下面的例子中，用您自己的信息替換#####，###和#密碼。

```
db2 connect to database_name user master_user using master_password
```

- b. ROLE_NULLID_PACKAGES將角色授與群組。在下列範例中，將 *group_name* 取代為您要新增角色之群組的名稱。

```
db2 "grant role ROLE_NULLID_PACKAGES to group group_name"
```

- c. ROLE_TABLESPACES將角色授與相同的群組。在下列範例中，將 *group_name* 取代為您要新增角色之群組的名稱。

```
db2 "grant role ROLE_TABLESPACES to group group_name"
```

- d. ROLE_PROCEDURES將角色授與相同的群組。在下列範例中，將 *group_name* 取代為您要新增角色之群組的名稱。

```
db2 "grant role ROLE_PROCEDURES to group group_name"
```

6. 授與connect、bindadd、createtab、和授IMPLICIT_SCHEMA權給您新增使用者的群組。在下列範例中，將 *group_name* 取代為您新增使用者的第二個群組的名稱。

```
db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to public"
db2 "grant connect, bindadd, createtab, implicit_schema on database to
group group_name"
```


7. 針對新增使用者的其他每個群組重複步驟 4 到 6。
8. 以使用者身分連線、建立資料表、將值插入資料表，然後從資料表傳回資料，以測試使用者的存取權限。在下列範例中，將 *rds_database_alias*、#用者名稱和##取代為資料庫名稱以及使用者的使用者名稱和密碼。

```
db2 connect to rds_database_alias user username using password
db2 "create table t1(c1 int not null)"
db2 "insert into t1 values (1),(2),(3),(4)"
db2 "select * from t1"
```

變更使用者的密碼

若要變更使用者的密碼

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和###。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 通過調用更改密碼rdsadmin.change_password。如需詳細資訊，請參閱 [rdsadmin.change_password](#)。

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

將群組新增至使用者

若要將群組新增至使用者

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和###。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 通過調用將組添加到用戶rdsadmin.add_groups。如需詳細資訊，請參閱 [rdsadmin.add_groups](#)。

```
db2 "call rdsadmin.add_groups(  
    'username',  
    'group_name,group_name')"
```

從使用者中移除群組

若要從使用者中移除群組

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 透過呼叫移除群組 `rdsadmin.remove_groups`。如需詳細資訊，請參閱 [rdsadmin.remove_groups](#)。

Warning

如果您從使用者移除附加至資料庫的所有群組，則該使用者將無法連線至資料庫。這是因為 Amazon RDS 授予該群組的授權，而不是使用者。

```
db2 "call rdsadmin.remove_groups(  
    'username',  
    'group_name,group_name')"
```

移除使用者

若要從授權清單中移除使用者

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 通過調用將用戶從授權列表中刪除 `rdsadmin.remove_user`。如需詳細資訊，請參閱 [rdsadmin.remove_user](#)。

```
db2 "call rdsadmin.remove_user('username')"
```

列出使用者

若要列出授權清單上的使用者，請呼叫 `rdsadmin.list_users` 預存程序。如需詳細資訊，請參閱 [rdsadmin.list_users](#)。

```
db2 "call rdsadmin.list_users()"
```

建立角色

您可以使用 [rdsadmin.create_role](#) 預存程序來建立角色。

若要建立角色

1. Connect 至資 `rdsadmin` 料庫。在下面的例子中，用您的信息替換 `Master_user ###` 和 `###`。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 將 Db2 設定為輸出內容。

```
db2 set serveroutput on
```

3. 建立角色。如需詳細資訊，請參閱 [the section called "rdsadmin.create_role"](#)。

```
db2 "call rdsadmin.create_role(  
    'database_name',  
    'role_name')"
```

4. 將 Db2 設定為不輸出內容。

```
db2 set serveroutput off
```

授予角色

您可以使用 [rdsadmin.grant_role](#) 預存程序將角色指派給角色、使用者或群組。

若要指派角色

1. Connect 至資rdsadmin料庫。在下面的例子中，用您的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 將 Db2 設定為輸出內容。

```
db2 set serveroutput on
```

3. 指派角色。如需詳細資訊，請參閱 [the section called “rdsadmin.grant_role”](#)。

```
db2 "call rdsadmin.grant_role(  
    'database_name',  
    'role_name',  
    'grantee',  
    'admin_option')"
```

4. 將 Db2 設定為不輸出內容。

```
db2 set serveroutput off
```

撤銷角色

您可以使用[rdsadmin.revoke_role](#)預存程序來撤銷角色、使用者或群組的角色。

若要撤銷角色

1. Connect 至資rdsadmin料庫。在下面的例子中，用您的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 撤銷角色。如需詳細資訊，請參閱 [the section called “rdsadmin.revoke_role”](#)。

```
db2 "call rdsadmin.revoke_role(  
    ?,  
    'database_name',  
    'role_name',  
    'grantee')"
```

授予資料庫授權

具有DBADM授權的主要使用者可以授與DBADM或DATAACCESS授權角色、使用者或群組。ACCESSCTRL

若要授與資料庫授權

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 通過呼叫授予用戶訪問權限rdsadmin.dbadm_grant。如需詳細資訊，請參閱 [rdsadmin.dbadm_grant](#)。

```
db2 "call rdsadmin.dbadm_grant(  
    ?,  
    'database_name',  
    'authorization',  
    'grantee')"
```

範例使用案例

下列程序會逐步引導您建立角色、DBADM授予角色授權，以及將角色指派給使用者。

若要建立角色、授與DBADM授權，並將角色指派給使用者

1. 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 建立呼叫資料庫PROD_ROLE的角色TESTDB。如需詳細資訊，請參閱 [rdsadmin.create_role](#)。

```
db2 "call rdsadmin.create_role(  
    'TESTDB',  
    'PROD_ROLE')"
```

3. 將角色指派給稱為的使用者PROD_USER。PROD_USER已授予管理員授權以指派角色。如需詳細資訊，請參閱 [rdsadmin.grant_role](#)。

```
db2 "call rdsadmin.grant_role(  
    'PROD_USER',  
    'PROD_ROLE',  
    'DBADM')
```

```
?,
'TESTDB',
'PROD_ROLE',
'USER PROD_USER',
'Y')"
```

- (選擇性) 提供其他授權或權限。下列範例會將授DBADM權授與名為的資料庫所命名PROD_ROLE的角色FUNDPROD。如需詳細資訊，請參閱 [rdsadmin.dbadm_grant](#)。

```
db2 "call rdsadmin.dbadm_grant(
?,
'FUNDPROD',
'DBADM',
'ROLE PROD_ROLE')"
```

- 終止您的工作階段。

```
db2 terminate
```

- 使用 RDS for Db2 testdb 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to testdb user master_username using master_password
```

- 為角色新增更多授權。

```
db2 "grant connect, implicit_schema on database to role PROD_ROLE"
```

撤銷資料庫授權

具有DBADM授權的主要使用者可以撤銷角色DBADMACCESSCTRL、使用者或群組的DATAACCESS授權。

撤銷資料庫授權

- 使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

- 通過呼叫撤銷用戶訪問權限rdsadmin.dbadm_revoke。如需詳細資訊，請參閱 [rdsadmin.dbadm_revoke](#)。

```
db2 "call rdsadmin.dbadm_revoke(  
    ?,  
    'database_name',  
    'authorization',  
    'grantee')"
```

附加至適用於 Db2 資料庫執行個體的遠端 RDS

若要附加至適用於 Db2 資料庫執行個體的遠端 RDS

1. 執行用戶端IBM Db2 CLP工作階段。如需為 RDS for Db2 資料庫執行個體和資料庫編目的相關資訊，請參閱。[使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 CLP](#)記下您 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼。
2. 附加至適用於 Db2 資料庫執行個體的 RDS。在下列範例中，將#####
TCPIP ##### RDS for D b2 資料庫執行個體的主要使用者名稱和主要密碼。

```
db2 attach to node_name user master_username using master_password
```

附加至遠端 RDS for Db2 資料庫執行個體後，您可以執行下列命令和其他get snapshot命令。如需詳細資訊，請參閱IBM Db2文件中的[GET SNAPSHOT指令](#)。

```
db2 list applications  
db2 get snapshot for all databases  
db2 get snapshot for database manager  
db2 get snapshot for all applications
```

執行適用於 Db2 資料庫執行個體的 Amazon RDS 常見資料庫任務

您可以在 Amazon RDS for Db2 資料庫執行個體上執行與資料庫相關的某些常見 DBA 任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。此外，主要使用者無法執行需要SYSADM、或SYSCTRL權限的命令或SYSMAINT公用程式。

主題

- [管理緩衝集區](#)
- [管理儲存體](#)
- [管理表格空間](#)

- [產生效能報告](#)
- [收集資料庫相關資訊](#)
- [強制關閉資料庫應用程式](#)

管理緩衝集區

您可以建立、變更或卸除 RDS for Db2 資料庫的緩衝集區。建立、變更或刪除緩衝集區需要更高層級的SYSADMIN授權，而主要使用者無法使用此權限。而是使用 Amazon RDS 存儲過程。

您也可以清除緩衝集區。

主題

- [建立緩衝集區](#)
- [更改緩衝集區](#)
- [刪除緩衝集區](#)
- [清除緩衝集區](#)

建立緩衝集區

若要為 RDS for Db2 資料庫建立緩衝集區，請呼叫`rdsadmin.create_bufferpool`預存程序。如需詳細資訊，請參閱IBM Db2文件中的[CREATE BUFFERPOOL陳述式](#)。

建立緩衝集區

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###`和`###`。

```
db2 "connect to rdsadmin user master_user using master_password"
```

2. 透過呼叫建立緩衝集區`rdsadmin.create_bufferpool`。如需詳細資訊，請參閱[rdsadmin.create_bufferpool](#)。

```
db2 "call rdsadmin.create_bufferpool(  
    'database_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    'immediate',  
    'automatic',
```



```
page_size,  
number_block_pages,  
block_size)"
```

更改緩衝集區

若要變更 RDS for Db2 資料庫的緩衝集區，請呼叫 `rdsadmin.alter_bufferpool` 預存程序。如需詳細資訊，請參閱 IBM Db2 文件中的 [ALTER BUFFERPOOL 陳述式](#)。

若要變更緩衝集區

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###` 和 `###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 通過調用改變緩衝池 `rdsadmin.alter_bufferpool`。如需詳細資訊，請參閱 [rdsadmin.alter_bufferpool](#)。

```
db2 "call rdsadmin.alter_bufferpool(  
    'database_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    'immediate',  
    'automatic',  
    change_number_blocks,  
    number_block_pages,  
    block_size)"
```

刪除緩衝集區

若要刪除 RDS for Db2 資料庫的緩衝集區，請呼叫 `rdsadmin.drop_bufferpool` 預存程序。如需詳細資訊，請參閱 IBM Db2 文件中的 [卸除緩衝集區](#)。

Important

請確定沒有將表格空間指派給您要刪除的緩衝集區。

刪除緩衝集區

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###`和`###`。

```
db2 "connect to rdsadmin user master_user using master_password"
```

2. 透過呼叫卸除緩衝集區`rdsadmin.drop_bufferpool`。如需詳細資訊，請參閱 [rdsadmin.drop_bufferpool](#)。

```
db2 "call rdsadmin.drop_bufferpool(  
    'database_name',  
    'buffer_pool_name')"
```

清除緩衝集區

您可以清除緩衝集區以強制檢查點，以便 RDS for Db2 將頁面從記憶體寫入儲存區。

Note

您不需要刷新緩衝池。Db2 會在提交交易之前同步寫入記錄。骯髒的分頁可能仍在緩衝集區中，但 Db2 會以非同步方式將它們寫入儲存區。即使系統意外關閉，當您重新啟動資料庫時，Db2 會自動執行損毀復原。在損毀復原期間，Db2 會將已認可的變更寫入資料庫，或回復未認可交易的變更。

若要清除緩衝集區

1. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至 Db2 資料庫。 `###
rds_database_alias#####`

```
db2 connect to rds_database_alias user master_username using master_password
```

2. 刷新緩衝池。

```
db2 flush bufferpools all
```

管理儲存體

Db2 會使用自動儲存來管理資料庫物件 (例如資料表、索引和暫存檔) 的實體儲存體。自動儲存可讓 Db2 系統視需要建立和管理儲存路徑，而不是手動配置儲存空間並追蹤正在使用的儲存路徑。這可以簡化 Db2 資料庫的管理作業，並減少因人為錯誤而發生錯誤的可能性。如需詳細資訊，請參閱 IBM Db2 文件中的 [自動儲存](#)。

使用 RDS for Db2，您可以透過自動擴充邏輯磁碟區和檔案系統來動態增加儲存區大小。如需詳細資訊，請參閱 [使用適合 Amazon RDS 資料庫執行個體的儲存體](#)。

管理表格空間

您可以建立、變更、重新命名或刪除 RDS for Db2 資料庫的表格空間。建立、變更、重新命名或刪除表格空間需要更高層級的 SYSADM 權限，而主要使用者無法使用此權限。而是使用 Amazon RDS 存儲過程。

主題

- [建立表格空間](#)
- [更改表格空間](#)
- [重新命名表空間](#)
- [刪除表格空間](#)
- [檢查表格空間的狀態](#)
- [返回有關表空間的詳細信息](#)
- [列出表格空間的狀態和儲存群組](#)
- [列出表格的表格空間](#)
- [列出表空間容器](#)

建立表格空間

若要建立 RDS for Db2 資料庫的表格空間，請呼叫 `rdsadmin.create_tablespace` 預存程序。如需詳細資訊，請參閱 IBM Db2 文件中的 [CREATE TABLESPACE 陳述式](#)。

Important

若要建立表格空間，您必須擁有相同頁面大小的緩衝區集區，才能與表格空間產生關聯。如需詳細資訊，請參閱 [管理緩衝集區](#)。

建立表格空間

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###`和`###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 透過呼叫建立表格空間`rdsadmin.create_tablespace`。如需詳細資訊，請參閱[rdsadmin.create_tablespace](#)。

```
db2 "call rdsadmin.create_tablespace(  
    'database_name',  
    'tablespace_name',  
    'buffer_pool_name',  
    tablespace_initial_size,  
    tablespace_increase_size,  
    'tablespace_type')"
```

更改表格空間

若要變更 RDS for Db2 資料庫的表格空間，請呼叫`rdsadmin.alter_tablespace`預存程序。您可以使用此預存程序來變更表格空間的緩衝區集區、降低上限標準，或使表格空間上線。如需詳細資訊，請參閱IBM Db2文件中的[ALTER TABLESPACE陳述式](#)。

變更表格空間

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###`和`###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 透過呼叫來變更表格空間`rdsadmin.alter_tablespace`。如需詳細資訊，請參閱[rdsadmin.alter_tablespace](#)。

```
db2 "call rdsadmin.alter_tablespace(  
    'database_name',  
    'tablespace_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    tablespace_increase_size,
```

```
'max_size', 'reduce_max',  
'reduce_stop',  
'reduce_value',  
'lower_high_water',  
'lower_high_water_stop',  
'switch_online')"
```

重新命名表空間

若要變更 RDS for Db2 資料庫的表格空間名稱，請呼叫 `rdsadmin.rename_tablespace` 預存程序。

重新命名表格空間

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###` 和 `###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 透過呼叫重新命名表格空間 `rdsadmin.rename_tablespace`。如需詳細資訊，包括可以命名表格空間的限制，請參閱 [rdsadmin.rename_tablespace](#)。

```
db2 "call rdsadmin.rename_tablespace(  
    'database_name',  
    'source_tablespace_name',  
    'target_tablespace_name')"
```

刪除表格空間

若要刪除 RDS for Db2 資料庫的表格空間，請呼叫 `rdsadmin.drop_tablespace` 預存程序。刪除表格空間之前，請先刪除表格空間中的任何物件，例如表格、索引或大型物件 (LOB)。如需詳細資訊，請參閱在 IBM Db2 文件中 [刪除資料表空間](#)。

刪除表格空間

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###` 和 `###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 透過呼叫刪除表格空間 `rdsadmin.drop_tablespace`。如需詳細資訊，請參閱 [rdsadmin.drop_tablespace](#)。

```
db2 "call rdsadmin.drop_tablespace(  
    'database_name',  
    'tablespace_name')"
```

檢查表格空間的狀態

您可以使用 `cast` 命令來檢查表格空間的狀態。

檢查表格空間的狀態

1. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至 Db2 資料庫。###
`rds_database_alias`#####

```
db2 connect to rds_database_alias user master_username using master_password
```

2. 返回一個摘要輸出。

對於摘要輸出：

```
db2 "select cast(tbsp_id as smallint) as tbsp_id,  
    cast(tbsp_name as varchar(35)) as tbsp_name,  
    cast(tbsp_type as varchar(3)) as tbsp_type,  
    cast(tbsp_state as varchar(10)) as state,  
    cast(tbsp_content_type as varchar(8)) as contents from  
    table(mon_get_tablespace(null,-1)) order by tbsp_id"
```

返回有關表空間的詳細信息

傳回表格空間的詳細資訊

1. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至 Db2 資料庫。###
`rds_database_alias`#####

```
db2 connect to rds_database_alias user master_username using master_password
```

2. 針對一個成員或所有成員，傳回資料庫中所有表格空間的詳細資訊。

對於一個成員：

```
db2 "select cast(member as smallint) as member,
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
table(mon_get_tablespace(null,-1)) order by member, tbsp_id "
```

對於所有會員：

```
db2 "select cast(member as smallint) as member
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
table(mon_get_tablespace(null,-2)) order by member, tbsp_id "
```

列出表格空間的狀態和儲存群組

若要列出表格空間的狀態和儲存群組，請執行下列 SQL 敘述句：

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name,
        varchar(TBSP_STATE, 30) state,
        tbsp_type,
        varchar(storage_group_name,30) storage_group
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

列出表格的表格空間

若要列出表格的表格空間，請執行下列 SQL 敘述句。在下列範例中，請以##### *SCHEMA_NAME # TABLE_NAME*：

```
db2 "SELECT
    VARCHAR(SD.TBSPACE,30) AS DATA_SPACE,
    VARCHAR(SL.TBSPACE,30) AS LONG_SPACE,
    VARCHAR(SI.TBSPACE,30) AS INDEX_SPACE
FROM
    SYSCAT.DATAPARTITIONS P
    JOIN SYSCAT.TABLESPACES SD ON SD.TBSPACEID = P.TBSPACEID
    LEFT JOIN SYSCAT.TABLESPACES SL ON SL.TBSPACEID = P.LONG_TBSPACEID
    LEFT JOIN SYSCAT.TABLESPACES SI ON SI.TBSPACEID = P.INDEX_TBSPACEID
WHERE
    TABSCHEMA = 'SCHEMA_NAME'
    AND TABNAME = 'TABLE_NAME'"
```

列出表空間容器

列出表格空間的表格空間容器

1. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至 Db2 資料庫。##### *rds_database_alias*#####

```
db2 connect to rds_database_alias user master_username using master_password
```

2. 傳回資料庫或特定表格空間容器中所有表格空間容器的清單。

對於所有表格空間容器：

```
db2 "select cast(member as smallint) as member,
```



```
cast(tbsp_name as varchar(35)) as tbsp_name,  
cast(container_id as smallint) as id,  
cast(container_name as varchar(60)) as container_path, container_type as type from  
table(mon_get_container(null,-2)) order by member,tbsp_id,container_id"
```

針對特定表格空間容器：

```
db2 "select cast(member as smallint) as member,  
cast(tbsp_name as varchar(35)) as tbsp_name,  
cast(container_id as smallint) as id,  
cast(container_name as varchar(60)) as container_path, container_type as type from  
table(mon_get_container('TBSP_1',-2)) order by member, tbsp_id,container_id"
```

產生效能報告

您可以使用程序或指令碼產生效能報告。如需有關使用程序的資訊，請參閱IBM Db2說明文件中的[DBSUMMARY程序-產生系統與應用程式效能測量結果摘要報告](#)。

Db2 在其~sqlllib/sample/perf目錄中包含一個db2mon.sh檔案。執行指令碼會產生低成本且廣泛的 SQL 測量結果報告。若要下載db2mon.sh檔案和相關指令碼檔案，請參閱 IBM db2 範例GitHub儲存庫中的[perf](#)目錄。

使用指令碼產生效能報告

1. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至 Db2 資料庫。在下面的例子中，用您自己的信息替換 *Master_user ###*和*###*。

```
db2 connect to rdsadmin user master_username using master_password
```

2. 透過呼叫rdsadmin.create_bufferpool建立頁面大小為 db2monbp 4096 的緩衝區集區。如需詳細資訊，請參閱 [rdsadmin.create_bufferpool](#)。

```
db2 "call rdsadmin.create_bufferpool('database_name', 'db2monbp', 4096)"
```

3. 透過呼叫，建立名為使db2montmptbsp用db2monbp緩衝集區的暫時表格空間rdsadmin.create_tablespace。如需詳細資訊，請參閱 [rdsadmin.create_tablespace](#)。

```
db2 "call rdsadmin.create_tablespace('database_name',\  
'db2montmptbsp', 'db2monbp', 4096, 1000, 100, 'T')"
```

4. 開啟指 `db2mon.sh` 令碼，然後修改關於連線到資料庫的行。

- a. 移除以下這行。

```
db2 -v connect to $dbName
```

- b. 使用下列行取代上一個步驟中的這一行。在下列範例中，請將 `master #####` 和主 `##### D` `b2` 資料庫執行個體 RDS 的主要使用者名稱和主密碼。

```
db2 -v connect to $dbName user master_username using master_password
```

5. 切換到腳本所在的目錄。#####

```
cd directory
```

6. 執行指 `db2mon.sh` 令碼，以指定的間隔輸出報表。在下列範例中，請將 `rds_database_alias` 和 `#` 數取代為資料庫名稱，以及產生報表之間的秒數 (0 到 3600)。

```
./db2mon.sh rds_database_alias seconds | tee -a db2mon.out
```

收集資料庫相關資訊

您可以使用 Amazon RDS 存放程序收集有關資料庫的資訊。這項資訊有助於監視資料庫或疑難排解問題。

若要收集有關資料庫的資訊

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###` 和 `###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 通過電話收集信息 `rdsadmin.db2pd`。如需詳細資訊，請參閱 [rdsadmin.db2pd_command](#)。

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

強制關閉資料庫應用程式

您可以使用 Amazon RDS 預存程序強制執行應用程式關閉 RDS for Db2 資料庫，以便維護資料庫。

強制關閉資料庫的應用程式

1. 使用 RDS for Db2 `rdsadmin` 資料庫執行個體的主要使用者名稱和主要密碼 Connect 線至資料庫。在下面的例子中，用您自己的信息替換 `Master_user ###`和`###`。

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. 透過呼叫將應用程式強制關閉資料庫`rdsadmin.force_application`。如需更多詳細資訊，請參閱 [rdsadmin.force_application](#)。

```
db2 "call rdsadmin.force_application(  
    ?,  
    'applications')"
```

將 Amazon RDS for Db2 資料庫執行個體與 Amazon S3 整合

您可以在 Amazon RDS for Db2 資料庫執行個體和使用 Amazon RDS 存放程序的 Amazon Simple Storage Service (Amazon S3) 儲存貯體之間傳輸檔案。如需詳細資訊，請參閱 [適用於 Db2 的 Amazon RDS 存儲過程參考](#)。

Note

資料庫執行個體與 Amazon S3 儲存貯體必須在相同的 AWS 區域中。

若要讓 RDS for Db2 與 Amazon S3 整合，您的資料庫執行個體必須能夠存取 RDS for Db2 所在的 Amazon S3 儲存貯體。如果您目前沒有 S3 儲存貯體，請[建立儲存貯體](#)。

主題

- [步驟 1：建立 IAM 政策](#)
- [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#)
- [步驟 3：將您的 IAM 角色新增至您的 RDS for Db2 資料庫執行個體](#)

步驟 1：建立 IAM 政策

在此步驟中，您會建立具有將檔案從 Amazon S3 儲存貯體傳輸到 RDS 資料庫執行個體所需許可的 AWS Identity and Access Management (IAM) 政策。此步驟假設您已建立 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[建立儲存貯體](#)。

在您建立政策之前，請記住下列資訊：

- 儲存貯體的 Amazon Resource Name (ARN)
- 如果您的存儲桶使用 SSE-KMS 或 SSE-S3 加密，則為您的 AWS Key Management Service (AWS KMS) 密鑰的 ARN。

建立包含下列許可的 IAM 政策：

```
"kms:GenerateDataKey",  
"kms:Decrypt",  
"s3:PutObject",  
"s3:GetObject",  
"s3:AbortMultipartUpload",
```

```
"s3:ListBucket",  
"s3:DeleteObject",  
"s3:GetObjectVersion",  
"s3:ListMultipartUploadParts"
```

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 建立 IAM 政策。

主控台

建立 IAM 政策以允許 Amazon RDS 存取您的 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇政策。
3. 選擇 [建立原則]，然後選擇 [JSON]。
4. 依服務新增動作。若要將檔案從 Amazon S3 儲存貯體傳輸到 Amazon RDS，您必須選取儲存貯體許可和物件許可。
5. 展開 Resources (資源)。您必須指定值區和物件資源。
6. 選擇下一步。
7. 在策略名稱中，輸入此策略的名稱。
8. (選擇性) 在說明中，輸入此原則的說明。
9. 選擇建立政策。

AWS CLI

建立 IAM 政策以允許 Amazon RDS 存取您的 Amazon S3 儲存貯體

1. 執行 `create-policy` 命令。在下列範例中，請將 `iam_####/s3_bucket_name ### IAM #### #`，以及您的 RDS for Db2 資料庫所在之 Amazon S3 儲存貯體的名稱。

對於LinuxmacOS、或Unix：

```
aws iam create-policy \  
  --policy-name iam_policy_name \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  

```

```

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetObjectVersion",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::s3_bucket_name/*",
        "arn:aws:s3:::s3_bucket_name"
      ]
    }
  ]
}'

```

在 Windows 中：

```

aws iam create-policy ^
--policy-name iam_policy_name ^
--policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetObjectVersion",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::s3_bucket_name/*",
        "arn:aws:s3:::s3_bucket_name"
      ]
    }
  ]
}'

```

```
    }  
  ]  
}'
```

2. 建立原則之後，請記下原則的 ARN。你需要的 ARN 為 [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#)。

如需建立 IAM 政策的相關資訊，請參閱 [IAM 使用者指南中的建立 IAM 政策](#)。

步驟 2：建立 IAM 角色並附加您的 IAM 政策

此步驟會假設您已於 [步驟 1：建立 IAM 政策](#) 中建立了 IAM 政策。在此步驟中，您會為 RDS for Db2 資料庫執行個體建立 IAM 角色，然後將您的 IAM 政策附加到該角色。

您可以使用 AWS Management Console 或為資料庫執行個體建立 IAM 角色 AWS CLI。

主控台

建立 IAM 角色並將您的 IAM 政策附加到該角色

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 針對信任的實體類型，選取 AWS 服務。
5. 對於服務或使用案例，請選取 RDS，然後選取 RDS — 將角色新增至資料庫。
6. 選擇下一步。
7. 對於許可政策，搜尋並選取您建立的 IAM 政策名稱。
8. 選擇下一步。
9. 在 Role name (角色名稱) 中，輸入角色名稱。
10. (選用) 在 Description (說明) 中，輸入新角色的說明。
11. 選擇建立角色。

AWS CLI

建立 IAM 角色並將您的 IAM 政策附加到該角色

1. 執行 `create-role` 命令。在下列範例中，將 `iam_role_name` 取代為您的身分與存取權管理角色的名稱。

對於LinuxmacOS、或Unix：

```
aws iam create-role \  
  --role-name iam_role_name \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
      }  
    ]  
  }'
```

在 Windows 中：

```
aws iam create-role ^  
  --role-name iam_role_name ^  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
      }  
    ]  
  }'
```


2. 角色建立後，請記下角色的 ARN。你需要的 ARN 為 [步驟 3：將您的 IAM 角色新增至您的 RDS for Db2 資料庫執行個體](#)。
3. 執行 `attach-role-policy` 命令。在下列範例中，將 `iam_policy_arn` 取代為您在中建立的身分與存取權管理政策的 ARN。 [步驟 1：建立 IAM 政策](#) 將 `iam_role_name` 取代為您剛建立的 IAM 角色的名稱。

對於 Linux/macOS、或 Unix：

```
aws iam attach-role-policy \  
  --policy-arn iam_policy_arn \  
  --role-name iam_role_name
```

在 Windows 中：

```
aws iam attach-role-policy ^  
  --policy-arn iam_policy_arn ^  
  --role-name iam_role_name
```

如需詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以將許可委派給 IAM 使用者](#)。

步驟 3：將您的 IAM 角色新增至您的 RDS for Db2 資料庫執行個體

在此步驟中，您將身分與存取權管理角色新增至您的 RDS for Db2 資料庫執行個體。請注意以下要求：

- 您必須能存取連接了必要 Amazon S3 許可政策的角色。
- 您一次只能將一個 IAM 角色與 RDS for Db2 資料庫執行個體建立關聯。
- 您的 RDS for Db2 資料庫執行個體必須處於 [可用] 狀態。

您可以使用 AWS Management Console 或將 IAM 角色新增至資料庫執行個體 AWS CLI。

主控台

將身分與存取權管理角色新增至您的 RDS for Db2 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。

3. 選擇您的 RDS for Db2 資料庫執行個體名稱。
4. 在 Connectivity & security (連線和安全) 索引標籤上，向下捲動至頁面底部的 Manage IAM roles (管理 IAM 角色) 區段。
5. 對於新增 IAM 角色到此執行個體，請選擇您在 [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#) 中建立的角色。
6. 針對 Feature (功能) 選擇 S3_INTEGRATION。
7. 選擇 Add role (新增角色)。

The screenshot shows the 'Manage IAM roles' interface. At the top, there's a title 'Manage IAM roles' and a refresh icon. Below that, there are two dropdown menus: 'Add IAM roles to this instance' (currently showing 'rds-s3-integration-role') and 'Feature' (currently showing 'S3_INTEGRATION'). To the right of these is an 'Add role' button. Below this is a section titled 'Current IAM roles for this instance (0)' with a 'Delete' button. At the bottom, there's a table with three columns: 'Role', 'Feature', and 'Status'. The table is currently empty.

AWS CLI

若要將 IAM 角色新增至您的 RDS for Db2 資料庫執行個體，請執行命 [add-role-to-db-instance](#) 令。在下列範例中，請將 *db_instance_name* # *iam_role_arn* ##### IAM ## # *ARN*。 [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#)

對於 Linux/macOS、或 Unix：

```
aws rds add-role-to-db-instance \
  --db-instance-identifier db_instance_name \
  --feature-name S3_INTEGRATION \
  --role-arn iam_role_arn \
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^
  --db-instance-identifier db_instance_name ^
  --feature-name S3_INTEGRATION ^
  --role-arn iam_role_arn ^
```

若要確認角色已成功新增至您的 RDS for Db2 資料庫執行個體，請執行命 [describe-db-instances](#) 令。在下列範例中，請以資料庫執行個體的名稱取代 *db_instance_name*。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-instances \  
  --filters "Name=db-instance-id,Values=db_instance_name" \  
  --query 'DBInstances[].AssociatedRoles'
```

在 Windows 中：

```
aws rds describe-db-instances ^  
  --filters "Name=db-instance-id,Values=db_instance_name" ^  
  --query 'DBInstances[].AssociatedRoles'
```

此命令會產生類似下列範例的輸出：

```
[  
  [  
    {  
      "RoleArn": "arn:aws:iam::0123456789012:role/rds-db2-s3-role",  
      "FeatureName": "S3_INTEGRATION",  
      "Status": "ACTIVE"  
    }  
  ]  
]
```

在 Amazon RDS 上將數據遷移到 Db2

您可以使用原生 Db2 工具 AWS 或原生 Db2 工具，將自我管理的 Db2 資料庫遷移到適用於 Db2 的 Amazon RDS。

主題

- [使用的遷移方法 AWS](#)
- [原生 Db2 工具](#)

使用的遷移方法 AWS

您可以將 Db2 資料庫從Linux、AIX或Windows環境一次性移轉到適用於 Db2 的 Amazon RDS。若要將停機時間降至最低，您可以執行近乎零的停機時間遷移。您也可以透過複寫或使用來執行同步移轉 AWS Database Migration Service。

對Linux於基於 Db2 資料庫的一次性遷移，Amazon RDS 僅支援離線和線上備份。Amazon RDS 不支援增量和Delta備份。對Linux於基於 Db2 的資料庫進行近乎零的遷移，Amazon RDS 需要線上備份。我們建議您使用線上備份進行近乎零的停機時間移轉，並針對可處理停機時間的移轉使用離線備份。

主題

- [一次性從Linux環境Linux遷移](#)
- [Linux基於 Db2 資料庫的近乎零停機時間遷移](#)
- [在Linux環境之間Windows進AIX行一次性移轉](#)
- [從Linux環境同步Linux遷移](#)
- [使用 AWS Database Migration Service \(AWS DMS \)](#)

一次性從Linux環境Linux遷移

使用這種遷移方法，您可以將自我管理的 Db2 資料庫備份到 Amazon S3 儲存貯體。然後，您可以使用 Amazon RDS 存放程序將 Db2 資料庫還原至適用於 Db2 資料庫的 Amazon RDS 資料庫執行個體。如需使用 Amazon S3 的詳細資訊，請參閱[將 Amazon RDS for Db2 資料庫執行個體與 Amazon S3 整合](#)。

主題

- [使用原生還原的限制和建議](#)
- [設定原生備份與還原](#)

• [還原您的 Db2 資料庫](#)

使用原生還原的限制和建議

下列限制與建議適用於使用原生還原：

- Amazon RDS 僅支援離線和線上備份以進行原生還原。Amazon RDS 不支援增量或Delta備份。
- 您無法從與 RDS for Db2 資料庫執行 AWS 區域 個體所在區域不同的 Amazon S3 儲存貯體還原。
- 如果您的 RDS for Db2 資料庫執行個體已包含資料庫，則無法還原資料庫。
- Amazon S3 將上傳到 Amazon S3 儲存貯體的檔案大小限制為 5 TB。如果您的資料庫備份檔案超過 5 TB，請將備份檔案分割成較小的檔案。
- Amazon RDS 不支援非圍欄的外部常式、增量還原或Delta還原。
- 您無法從加密的來源資料庫還原，但可以還原至加密的 Amazon RDS 資料庫執行個體。

當您還原資料庫時，會複製備份，然後在 RDS for Db2 資料庫執行個體上擷取。建議您為 RDS for Db2 DB 執行個體佈建儲存空間，該儲存空間等於或大於備份大小加上磁碟上原始資料庫大小的總和。

還原的資料庫大小上限是支援的最大資料庫大小減去備份大小。例如，如果支援的資料庫大小上限為 64 TiB，而備份的大小為 30 TiB，則還原的資料庫大小上限為 34 TiB。

$$64 \text{ TiB} - 30 \text{ TiB} = 34 \text{ TiB}$$

設定原生備份與還原

對於原生備份和還原，您需要下列 AWS 元件：

- 用於存放備份檔案的 Amazon S3 儲存貯體：上傳您要遷移到 Amazon RDS 的任何備份檔案。建議您將離線備份用於可處理停機時間的移轉。如果您已經有 S3 儲存貯體，則可以使用該儲存貯體。如果您沒有 S3 儲存貯體，請參閱 Amazon S3 使用者指南中的[建立](#)儲存貯體。

Note

如果您的資料庫很大且需要很長時間才能傳輸到 S3 儲存貯體，您可以訂購 AWS Snow Family 裝置並 AWS 要求執行備份。將檔案複製到裝置並將其歸還給 Snow Family 團隊後，團隊會將您的備份映像傳輸到 S3 儲存貯體。如需詳細資訊，請參閱 [AWS Snow Family 文件](#)。

- 存取 S3 儲存貯體的 IAM 角色：如果您已有 IAM 角色，則可以使用該角色。如果您沒有角色，請參閱 [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#)。
- 將信任關係和許可附加到 IAM 角色的 IAM 政策：如需詳細資訊，請參閱 [步驟 1：建立 IAM 政策](#)。
- 已新增至您的 RDS for Db2 資料庫執行個體的 IAM 角色：如需詳細資訊，請參閱 [步驟 3：將您的 IAM 角色新增至您的 RDS for Db2 資料庫執行個體](#)。

還原您的 Db2 資料庫

設定原生備份和還原之後，您就可以將 Db2 資料庫還原到 RDS for Db2 資料庫執行個體。

將 Db2 資料庫還原至適用於 Db2 的 RDS 資料庫執行個體

1. Connect 到適用於 Db2 的 RDS 資料庫執行個體。如需詳細資訊，請參閱 [連接到您的 Amazon RDS 用 Db2 資料庫執行個體](#)。
2. (選擇性) 若要確定您的資料庫已設定還原作業的最佳設定，您可 [the section called "rdsadmin.show_configuration"](#) 以呼叫以檢查 RESTORE_DATABASE_PARALLELISM 和的值 RESTORE_DATABASE_NUM_BUFFERS。視需 [the section called "rdsadmin.set_configuration"](#) 要呼叫變更這些值。明確設定這些值可以改善還原含有大量資料的資料庫時的效能。
3. 通過調用恢復您的數據庫 `rdsadmin.restore_database`。如需更多詳細資訊，請參閱 [rdsadmin.restore_database](#)。

Linux 基於 Db2 資料庫的近乎零停機時間遷移

使用這種遷移方法，您可以將 Linux 基於 Db2 的資料庫從一個自我管理的 Db2 資料庫 (來源) 遷移到適用於 Db2 的 Amazon RDS。這種方法會導致應用程式或使用者的中斷或停機時間降至最低。這種方法會備份您的資料庫，並使用記錄重新顯示來還原資料庫，這有助於防止中斷進行中的作業，並提供資料庫的高可用性。

為了達到接近零停機時間的移轉，RDS for Db2 會使用記錄重新顯示來實作還原。這種方法會備份您的自我管理 Linux 型 Db2 資料庫，並將其還原到 RDS for Db2 伺服器上。使用 Amazon RDS 存放程序後，您可以套用後續交易日誌，使資料庫保持最新狀態。

主題

- [近乎零停機時間的遷移限制和建議](#)
- [設定幾乎零停機時間的遷移](#)
- [移轉您的 Db2 資料庫](#)

近乎零停機時間的遷移限制和建議

下列限制適用於使用近乎零停機時間的移轉：

- Amazon RDS 需要線上備份，以實現近乎零停機時間的遷移。這是因為 Amazon RDS 會在您上傳存檔的交易日誌時，將您的資料庫保持在向前復原擱置狀態。如需詳細資訊，請參閱 [the section called “移轉您的 Db2 資料庫”](#)。
- 您無法從與 RDS for Db2 資料庫執行 AWS 區域 個體所在區域不同的 Amazon S3 儲存貯體還原。
- 如果您的 RDS for Db2 資料庫執行個體已包含資料庫，則無法還原資料庫。
- Amazon S3 將上傳到 S3 儲存貯體的檔案大小限制為 5 TB。如果您的資料庫備份檔案超過 5 TB，請將備份檔案分割成較小的檔案。
- Amazon RDS 不支援非圍欄的外部常式、增量還原或 Delta 還原。
- 您無法從加密的來源資料庫還原，但可以還原至加密的 Amazon RDS 資料庫執行個體。

當您還原資料庫時，Amazon RDS 會複製您的備份，然後將其擷取到 RDS for Db2 資料庫執行個體。建議您為 RDS for Db2 DB 執行個體佈建儲存空間，該儲存空間等於或大於備份大小加上磁碟上原始資料庫大小的總和。

還原的資料庫大小上限是支援的最大資料庫大小減去備份大小。例如，如果支援的資料庫大小上限為 64 TiB，而備份的大小為 30 TiB，則還原的資料庫大小上限為 34 TiB。

$$64 \text{ TiB} - 30 \text{ TiB} = 34 \text{ TiB}$$

設定幾乎零停機時間的遷移

對於接近零停機時間的遷移，您需要下列 AWS 元件：

- 用於存放備份檔案的 Amazon S3 儲存貯體：上傳您要遷移到 Amazon RDS 的任何備份檔案。Amazon RDS 需要線上備份，以實現近乎零停機時間的遷移。如果您已經有 S3 儲存貯體，則可以使用該儲存貯體。如果您沒有 S3 儲存貯體，請參閱 Amazon S3 使用者指南中的 [建立](#) 儲存貯體。

Note

如果您的資料庫很大且需要很長時間才能傳輸到 S3 儲存貯體，您可以訂購 AWS Snow Family 裝置並 AWS 要求執行備份。將檔案複製到裝置並將其歸還給 Snow Family 團隊之

後，團隊會將您的備份映像傳輸到 S3 儲存貯體。如需詳細資訊，請參閱 [AWS Snow Family 文件](#)。

- 存取 S3 儲存貯體的 IAM 角色：如果您已有 AWS Identity and Access Management (IAM) 角色，則可以使用該角色。如果您沒有角色，請參閱 [步驟 2：建立 IAM 角色並附加您的 IAM 政策](#)。
- 將信任關係和許可附加到 IAM 角色的 IAM 政策：如需詳細資訊，請參閱 [步驟 1：建立 IAM 政策](#)。
- 已新增至您的 RDS for Db2 資料庫執行個體的 IAM 角色：如需詳細資訊，請參閱 [步驟 3：將您的 IAM 角色新增至您的 RDS for Db2 資料庫執行個體](#)。

移轉您的 Db2 資料庫

設定幾乎零停機時間的遷移後，您就可以將 Db2 資料庫遷移到 RDS for Db2 資料庫執行個體。

若要執行近乎零的停機時間移轉

1. 執行來源資料庫的線上備份。如需詳細資訊，請參閱 IBM Db2 文件中的 [BACKUP DATABASE 指令](#)。
2. 將資料庫的備份複製到 Amazon S3 儲存貯體。如需使用 Amazon S3 的相關資訊，請參閱 [Amazon 簡易儲存服務使用者指南](#)。
3. 使用適用於 Db2 資料庫執行個體的 RDS ##### 和 ### Connect 線至 rdsadmin 伺服器。

```
db2 connect to rdsadmin user master_username using master_password
```

4. (選擇性) 若要確定您的資料庫已設定還原作業的最佳設定，您可 [the section called "rdsadmin.show_configuration"](#) 以呼叫以檢查 RESTORE_DATABASE_PARALLELISM 和的值 RESTORE_DATABASE_NUM_BUFFERS。視需 [the section called "rdsadmin.set_configuration"](#) 要呼叫變更這些值。明確設定這些值可以改善還原含有大量資料的資料庫時的效能。
5. 透過呼叫 `rdsadmin.restore_database`，在 RDS for Db2 伺服器上還原備份。將 `backup_type` 設定為 ONLINE。如需詳細資訊，請參閱 [rdsadmin.restore_database](#)。
6. 將您的存檔日誌從來源伺服器複製到 S3 儲存貯體。如需詳細資訊，請參閱文件中的 [封存記 IBM Db2 錄](#)。
7. 根據需要通過調用多次應用存檔日誌 `rdsadmin.rollforward_database`。設定 `complete_rollforward` 為 FALSE 使資料庫保持在一個 ROLL-FORWARD PENDING 狀態。如需詳細資訊，請參閱 [rdsadmin.rollforward_database](#)。
8. 套用所有存檔日誌之後，請撥打電話讓資料庫上線 `rdsadmin.complete_rollforward`。如需詳細資訊，請參閱 [rdsadmin.complete_rollforward](#)。

9. 透過更新資料庫的應用程式端點，或更新 DNS 端點以將流量重新導向至 RDS for Db2 伺服器，將應用程式連線切換至 RDS for Db2 伺服器。您也可以自我管理的 Db2 資料庫端點搭配 RDS for Db2 資料庫端點使用 Db2 自動用戶端重新路由功能。如需詳細資訊，請參閱 IBM Db2 文件中的 [自動用戶端重新路由描述和設定](#)。
10. (選擇性) 關閉來源資料庫。

在 Linux 環境之間 Windows 進 AIX 行一次性移轉

透過這種遷移方法，您可以使用原生 Db2 工具將自我管理的 Db2 資料庫備份到 Amazon S3 儲存貯體。原生 Db2 工具包括 export 公用程式、db2move 系統指令或 db2look 系統指令。您的 Db2 資料庫可以是自我管理的，也可以在亞馬遜彈性運算雲端 (Amazon EC2) 中進行。您可以將資料從您的 AIX 或 Windows 系統移至 Amazon S3 儲存貯體。然後，使用 Db2 用戶端將資料直接從 S3 儲存貯體載入到您的 Amazon RDS for Db2 資料庫。停機時間取決於資料庫的大小。如需使用 Amazon S3 的詳細資訊，請參閱 [將 Amazon RDS for Db2 資料庫執行個體與 Amazon S3 整合](#)。

若要將 Db2 資料庫移轉至適用於 Db2 的 RDS 資料庫

1. 準備備份您的資料庫。設定足夠的儲存空間以將備份保留在自我管理的 Db2 系統上。
2. 備份您的資料庫。
 - a. 執行系 [db2look 系統命令](#) 以擷取所有物件的資料定義語言 (DDL) 檔案。
 - b. 執行 [Db2 匯出公用程式](#)、[db2move 系統命令](#) 或 [CREATE EXTERNAL TABLE 陳述式](#)，將 Db2 表格資料卸載至 Db2 系統上的儲存空間。
3. 將您的備份移至 Amazon S3 儲存貯體。如需詳細資訊，請參閱 [將 Amazon RDS for Db2 資料庫執行個體與 Amazon S3 整合](#)。

Note

如果您的資料庫很大且需要很長時間才能傳輸到 S3 儲存貯體，您可以訂購 AWS Snow Family 裝置並 AWS 要求執行備份。將檔案複製到裝置並將其歸還給 Snow Family 團隊後，團隊會將您的備份映像傳輸到 S3 儲存貯體。如需詳細資訊，請參閱 [AWS Snow Family 文件](#)。

4. 使用 Db2 用戶端將資料直接從 S3 儲存貯體載入到 RDS for Db2 資料庫。

從Linux環境同步Linux遷移

使用這種遷移方法，您可以在自我管理的 Db2 資料庫和 Amazon RDS for Db2 資料庫執行個體之間設定複寫。對自我管理資料庫所做的變更會以近乎即時的速度複寫到 RDS for Db2 資料庫執行個體。此方法可提供持續的可用性，並將移轉過程中的停機時間降至最低。

使用 AWS Database Migration Service (AWS DMS)

您可以使 AWS DMS 用一次性移轉，然後從 Linux、Unix 和視窗上的 Db2 同步到適用於 Db2 的 Amazon RDS。如需詳細資訊，請參閱 [什麼是 AWS Database Migration Service ?](#)。

原生 Db2 工具

您可以使用數個原生 Db2 工具、公用程式和命令，將資料從 Db2 資料庫移至 Amazon RDS for Db2 資料庫。若要使用這些原生 Db2 工具，您必須能夠將用戶端機器連線到 RDS for Db2 資料庫執行個體。如需詳細資訊，請參閱 [將用戶端機器連接至適用於 Db2 的資料庫執行個體的 Amazon RDS](#)。

工具名稱	使用案例	限制
看	將中繼資料從自我管理的 Db2 資料庫複製到 RDS for Db2 資料庫。	<ul style="list-style-type: none"> 您必須修改建立緩衝集區、建立表格空間和建立角色的語法的語法，以符合使用的 適用於 Db2 的 RDS 預存程序語法。
IMPORT 命令	將包含大型物件 (LOB) 的小型表格和表格從用戶端機器移轉到 RDS for Db2 資料庫執行個體。	<ul style="list-style-type: none"> 由於 INSERT 和 DELETE 日誌操作，比 LOAD 實用程序慢。 網路頻寬有限，效能不佳。
INGEST 效用	持續將用戶端機器上沒有大型物件 (LOB) 的檔案和管道中的資料串流至 RDS for Db2 資料庫執行個體。支持 INSERT 和 MERGE 操作。	<ul style="list-style-type: none"> 無法串流包含 LOB 的資料檔案。請改用 IMPORT 指令。 自我管理的 Db2 資料庫與適用於 Db2 資料庫的 RDS 之間需要連線。
INSERT 命令	將小型資料表中的資料從自我管理的 Db2 資料庫複製到 RDS for Db2 資料庫。	<ul style="list-style-type: none"> 自我管理的 Db2 資料庫與適用於 Db2 資料庫的 RDS 之間需要連線。

工具名稱	使用案例	限制
LOAD 命令	將不含大型物件 (LOB) 的小型資料表從用戶端機器移轉到 RDS for Db2 資料庫執行個體。	<ul style="list-style-type: none"> 網路頻寬有限，效能不佳。 無法移轉包含 LOB 的資料檔案。請改用 IMPORT 指令。 網路頻寬有限，效能不佳。

將用戶端機器連接至適用於 Db2 的資料庫執行個體的 Amazon RDS

若要使用任何原生 Db2 工具將資料從 Db2 資料庫移至 Amazon RDS for Db2 資料庫，您必須先將用戶端機器連線到適用於 Db2 的 RDS 資料庫執行個體。

用戶端電腦可以是下列任何一種：

- Linux、Windows 或 macOS 上的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體。此執行個體應與您的 RDS for Db2 資料庫執行個體、或位於相同的虛擬私有雲端 (VPC) AWS Cloud9 中。AWS CloudShell
- Amazon EC2 執行個體中的自我管理 Db2 執行個體。執行個體應位於相同的 VPC 中。
- Amazon EC2 執行個體中的自我管理 Db2 執行個體。如果您啟用了 VPC 對等，則執行個體可以位於不同的 VPC 中。如需詳細資訊，請參閱 [《Amazon Virtual Private Cloud VPC 對等互連指南》](#) 中的建立 VPC 對等連線。
- 執行 Linux 或自我管理環境 macOS 中的本機電腦。Windows 您必須擁有 RDS for Db2 的公用連線，或啟用自我管理的 Db2 執行個體與之間的 VPN 連線。AWS

若要將用戶端機器連線到 RDS for Db2 資料庫執行個體，請使用登入您的用戶端機器 IBM Db2 Data Management Console。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#) 及 [IBM Db2 Data Management Console](#)。

您可以使用 AWS Database Migration Service (AWS DMS) 對資料庫執行查詢、執行 SQL 執行計畫，以及監視資料庫。如需詳細資訊，請參閱 [什麼是 AWS Database Migration Service?](#) 在《AWS Database Migration Service 使用者指南》中。

成功將用戶端機器連線到 RDS for Db2 資料庫執行個體之後，您就可以使用任何原生 Db2 工具複製資料。如需更多詳細資訊，請參閱 [原生 Db2 工具](#)。

db2look工具

db2look是原生 Db2 工具，可擷取資料定義語言 (DDL) 檔案、物件、授權、組態、WLM 和資料庫配置。您可以使用db2look將資料庫中繼資料從自我管理的 Db2 資料庫複製到 Amazon RDS for Db2 資料庫。如需詳細資訊，請參閱文件中的[使用 db2look 模擬資料庫](#)。IBM Db2

若要複製資料庫詮釋資料

1. 在自我管理的 Db2 系統上執行此db2look工具，以解壓縮 DDL 檔案。在下列範例中，將#### #取代為 Db2 資料庫的名稱。

```
db2look -d database_name -e -l -a -f -wlm -cor -createdb -printdbcfg -o db2look.sql
```

2. 如果您的用戶端機器可以存取來源 (自我管理的 Db2) 資料庫和 RDS for Db2 資料庫執行個體，您可以直接連結至遠端執行個體，在用戶端機器上建立db2look.sql檔案。然後編目遠端自我管理的 Db2 執行個體。
 - a. 編目節點。在下列範例中，將 *dns_ip_address* 和###取代為 DNS 名稱或 IP 位址以及自我管理 Db2 資料庫的連接埠號碼。

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

- b. 編目資料庫。在下列範例中，請以自我管理的 Db2 #####和您要用於此#####資料庫名稱和來源資料庫。

```
db2 catalog database source_database_name as source_database_alias at node  
srcnode \  
authentication server_encrypt
```

- c. 附加至來源資料庫。在下列範例中，將 *source_database_alias##### _password ##### Db2 #####* 用者識別碼和密碼。

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \  
-cor -createdb -printdbcfg -o db2look.sql
```

3. 如果您無法從用戶端機器存取遠端自我管理的 Db2 資料庫，請將db2look.sql檔案複製到用戶端電腦。然後編目適用於 Db2 資料庫執行個體的 RDS。
 - a. 編目節點。在下列範例中，將 *dns_ip_address* 和###取代為適用於 Db2 資料庫執行個體之 RDS 的 DNS 名稱或 IP 位址和連接埠號碼。

```
db2 catalog tcpip node remnode REMOTE dns_ip_address server port
```

- b. 編目資料庫。在下列範例中，請將 *rds_database_name* # *rds_database_alias* 取代為 Db2 資料庫的 RDS 名稱，以及您要用於此資料庫的別名。

```
db2 catalog database rds_database_name as rds_database_alias at node remnode \  
authentication server_encrypt
```

- c. 編目管理 Db2 RDS 的管理資料庫。您不能使用此數據庫存儲任何數據。

```
db2 catalog database rdsadmin as rdsadmin at node remnode authentication  
server_encrypt
```

4. 建立緩衝集區和表格空間。系統管理員沒有建立緩衝集區或表格空間的權限。不過，您可以使用 Amazon RDS 存放程序來建立它們。

- a. 尋找 `db2look.sql` 檔案中緩衝集區和表格空間的名稱和定義。
- b. 使用適用於 Db2 資料庫執行個體 RDS 的主要使用者名稱和主密碼 Connect 線至 Amazon RDS。在下面的例子中，用您自己的信息替換 *Master_user ###* 和 *###*。

```
db2 connect to rdsadmin user master_username using master_password
```

- c. 透過呼叫建立緩衝集區 `rdsadmin.create_bufferpool`。如需詳細資訊，請參閱 [rdsadmin.create_bufferpool](#)。

```
db2 "call rdsadmin.create_bufferpool(  
    'database_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    'immediate',  
    'automatic',  
    page_size,  
    number_block_pages,  
    block_size)"
```

- d. 透過呼叫建立表格空間 `rdsadmin.create_tablespace`。如需詳細資訊，請參閱 [rdsadmin.create_tablespace](#)。

```
db2 "call rdsadmin.create_tablespace(  
    'database_name',
```

```
'tablespace_name',  
'buffer_pool_name',  
tablespace_initial_size,  
tablespace_increase_size,  
'tablespace_type')"
```

- e. 針對您要新增的每個額外緩衝集區或表格空間，重複步驟 c 或 d。
- f. 終止您的連線。

```
db2 terminate
```

5. 創建表和對象。

- a. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主密碼，Connect 線至您的 RDS for Db2 資料庫。##### *rds_database_name*#####

```
db2 connect to rds_database_name user master_username using master_password
```

- b. 執行 db2look.sql 檔案。

```
db2 -tvf db2look.sql
```

- c. 終止您的連線。

```
db2 terminate
```

IMPORT 使用用戶端機器命令

您可以使用用戶端機器上的 IMPORT 命令，將資料匯入 Amazon RDS for Db2 伺服器。

Important

此指 IMPORT 命令方法適用於移轉包含大型物件 (LOB) 的小型資料表和資料表。由於 INSERT 和 DELETE 記錄操作，IMPORT 命令比 LOAD 公用程式慢。如果用戶端機器與 RDS for Db2 之間的網路頻寬有限，建議您使用不同的移轉方法。如需詳細資訊，請參閱 [原生 Db2 工具](#)。

若要將資料匯入至適用於 Db2 的 RDS 伺服器

1. 使用登入您的用戶端電腦IBM Db2 Data Management Console。如需詳細資訊，請參閱 [使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 Data Management Console](#)。
2. 在用戶端機器上為 Db2 資料庫編目 RDS。
 - a. 為節點建立目錄。在下列範例中，將 *dns_ip_address* 和###取代為 DNS 名稱或 IP 位址以及自我管理 Db2 資料庫的連接埠號碼。

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

- b. 編目資料庫。在下列範例中，請以自我管理的 Db2 #####和您要用於此#####資料庫名稱和來源資料庫資料庫。

```
db2 catalog database source_database_name as source_database_alias at node
srcnode \
authentication server_encrypt
```

3. 附加至來源資料庫。在下列範例中，將 *source_database_alias##### _password##### Db2 #####* 用者識別碼和密碼。

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
-cor -createdb -printdbcfg -o db2look.sql
```

4. 使用自我管理的 Db2 系統上的 EXPORT指令產生資料檔案。#####
###用數據#####和###稱替換文件名和表格名稱。

```
db2 "export to /directory/file_name.txt of del lob to /directory/lobs/ \
modified by coldel\| select * from table_name"
```

5. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主密碼，Connect 線至您的 RDS for Db2 資料庫。##### *rds_database_alias#####*

```
db2 connect to rds_database_alias user master_username using master_password
```

6. 使用此IMPORT命令將資料從用戶端機器上的檔案匯入遠端 RDS for Db2 資料庫。如需詳細資訊，請參閱IBM Db2文件中的[IMPORT指令](#)。在下列範例中，將##和 *file_name* 取代為資料檔案所在的用戶端機器上的目錄以及資料檔案的名稱。將#####取代為結構描述和資料表的名稱。

```
db2 "IMPORT from /directory/file_name.tbl OF DEL LOBS FROM /directory/lobs/ \
```



```
modified by coldel\| replace into SCHEMA_NAME.TABLE_NAME"
```

7. 終止您的連線。

```
db2 terminate
```

INGEST效用

您可以使用此公用INGEST程式，將用戶端機器上的檔案和管道中的資料持續串流到目標 Amazon RDS for Db2 資料庫執行個體。該INGEST實用程序支持INSERT和MERGE操作。如需詳細資訊，請參閱IBM Db2文件中的 [Ingest 公用程式](#)。

由於INGEST公用程式支援暱稱，因此您可以使用公用程式將資料從自我管理的 Db2 資料庫傳輸到 RDS for Db2 資料庫。只要兩個數據庫之間存在網絡連接，這種方法就可以工作。

Important

此公INGEST用程式不支援大型物件 (LOB)。請改用[IMPORT指令](#)。

若要使用公用INGEST程式的RESTARTABLE功能，請在 RDS for Db2 資料庫上執行下列命令。

```
db2 "call sysproc.sysinstallobjects('INGEST','C',NULL,NULL)"
```

INSERT從自我管理的 Db2 資料庫到 Amazon RDS for Db2 資料庫的命令

您可以使用自我管理的 Db2 伺服器上的INSERT命令，將資料插入 Amazon RDS for Db2 資料庫。透過這種移轉方法，您可以為遠端 RDS for Db2 資料庫執行個體使用暱稱。您的自我管理 Db2 資料庫 (來源) 必須能夠連線至 RDS for Db2 資料庫 (目標)。

Important

指INSERT令方法對於移轉小型資料表非常有用。如果自我管理的 Db2 資料庫與 RDS for Db2 資料庫之間的網路頻寬有限，建議您使用不同的移轉方法。如需詳細資訊，請參閱 [原生 Db2 工具](#)。

將資料從自我管理的 Db2 資料庫複製到 RDS for Db2 資料庫

1. 在自我管理的 Db2 執行個體上為 Db2 資料庫執行個體編目 RDS。

- a. 編目節點。在下列範例中，將 *dns_ip_address* 和###取代為 DNS 名稱或 IP 位址以及自我管理 Db2 資料庫的連接埠號碼。

```
db2 catalog tcpip node remnode REMOTE dns_ip_address SERVER port
```

- b. 編目資料庫。在下列範例中，將 *rds_database_name* 取代為 Db2 資料庫執行個體 RDS 上的資料庫名稱。

```
db2 catalog database rds_database_name as remdb at node remnode \  
authentication server_encrypt
```

2. 在自我管理的 Db2 執行個體上啟用聯合。在下列範例中，將##### Db2 執行個體上的資料庫名稱。

```
db2 update dbm cfg using FEDERATED YES source_database_name
```

3. 在適用於 Db2 資料庫執行個體的 RDS 上建立表格。

- a. 編目節點。在下列範例中，將 *dns_ip_address* 和###取代為 DNS 名稱或 IP 位址以及自我管理 Db2 資料庫的連接埠號碼。

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

- b. 編目資料庫。在下列範例中，請以自我管理的 Db2 #####和要用於此#####資料庫名稱和來源資料庫。

```
db2 catalog database source_database_name as source_database_alias at node  
srcnode \  
authentication server_encrypt
```

4. 附加至來源資料庫。在下列範例中，將 *source_database_alias##### _password##### Db2 #####* 用戶識別碼和密碼。

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \  
-cor -createdb -printdbcfg -o db2look.sql
```

5. 設定同盟，並為自我管理的 Db2 執行個體上 RDS for Db2 資料庫表格建立暱稱。

- a. Connect 到您的本機資料庫。在下列範例中，將 *source_database_name* 取代為您自我管理的 Db2 執行個體上的資料庫名稱。

```
db2 connect to source_database_name
```

- b. 建立包裝函式以存取 Db2 資料來源。

```
db2 create wrapper drda
```

- c. 在聯合資料庫上定義資料來源。在下列範例中，請以您自我### Db2 ##### *admin* 和 *admin_password*。將 *rds_database_name* 取代為適用於 Db2 資料庫執行個體的 RDS 上的資料庫名稱。

```
db2 "create server rdsdb2 type DB2/LUW version '11.5.9.0' \  
    wrapper drda authorization "admin" password "admin_password" \  
    options( dbname 'rds_database_name', node 'remnode')"
```

- d. 對應兩個資料庫上的使用者。在下列範例中，將 *Master_user #####*和主##### *D b2* 資料庫執行個體之 RDS 的認證。

```
db2 "create user mapping for user server rdsdb2 \  
    options (REMOTE_AUTHID 'master_username', REMOTE_PASSWORD 'master_password')"
```

- e. 驗證與 Db2 版 RDS 伺服器的連線。

```
db2 set passthru rdsdb2
```

- f. 在遠端 RDS for Db2 資料庫中建立資料表的暱稱。在下列範例中，將#稱和 *TABLE_NAME* 取代為資料表的暱稱和資料表名稱。

```
db2 create nickname REMOTE.NICKNAME for RDSDB2.TABLE_NAME.NICKNAME
```

6. 將資料插入遠端 RDS for Db2 資料庫的資料表中。在自我管理 Db2 執行個體的本機資料表上的 select 陳述式中使用暱稱。在下列範例中，將#稱和 *TABLE_NAME* 取代為資料表的暱稱和資料表名稱。

```
db2 "INSERT into REMOTE.NICKNAME select * from RDS2DB2.TABLE_NAME.NICKNAME"
```

LOAD使用用戶端機器命令

您可以使用此LOAD CLIENT命令將資料從檔案載入到 Amazon RDS for Db2 伺服器。由於 RDS for Db2 伺服器沒有 SSH 連線，因此您可以在自我管理的 Db2 伺服器或 Db2 用戶端機器上使用此LOAD CLIENT命令。

Important

指LOAD令方法對於移轉小型資料表非常有用。如果用戶端和 RDS for Db2 之間的網路頻寬有限，建議您使用不同的移轉方法。如需更多資訊，請參閱[原生 Db2 工具](#)。

如果您的資料檔包含對大型物件檔名的參考，則LOAD指令將無法運作，因為大型物件 (LOB) 需要位於 Db2 伺服器上。如果您嘗試將 LOB 從用戶端機器載入至 RDS for Db2 伺服器，您將會收到錯誤SQL3025N訊息。請改用[IMPORT指令](#)。

若要將資料載入至適用於 Db2 伺服器的 RDS

1. 使用登入您的用戶端電腦IBM Db2 Data Management Console。如需詳細資訊，請參閱[使用以下方式連接到您的 Amazon RDS 適用於 Db2 資料庫 IBM Db2 Data Management Console](#)。
2. 在用戶端機器上為 Db2 資料庫編目 RDS。
 - a. 編目節點。在下列範例中，將 *dns_ip_address* 和###取代為 DNS 名稱或 IP 位址以及自我管理 Db2 資料庫的連接埠號碼。

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

- b. 編目資料庫。在下列範例中，請以自我管理的 Db2 #####和要用於此#####資料庫名稱和來源資料庫。

```
db2 catalog database source_database_name as source_database_alias at node  
srcnode \  
authentication server_encrypt
```

3. 附加至來源資料庫。在下列範例中，將 *source_database_alias##### _password##### Db2 #####* 用者識別碼和密碼。

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \  
-cor -createdb -printdbcfg -o db2look.sql
```

4. 使用自我管理的 Db2 系統上的EXPORT指令產生資料檔案。#####
##用數據####和###稱替換文件名和表格名稱。

```
db2 "export to /directory/file_name.txt of del modified by coldel\| \  
select * from TPCH.TABLE_NAME"
```

5. 使用 RDS for Db2 資料庫執行個體的主要使用者名稱和主密碼，Connect 線至您的 RDS for Db2 資料庫。##### rds_database_alias#####

```
db2 connect to rds_database_alias user master_username using master_password
```

6. 使用此LOAD命令將資料從用戶端機器上的檔案載入到遠端 RDS for Db2 資料庫。如需詳細資訊，請參閱IBM Db2文件中的[LOAD指令](#)。#####用數據####和##
#稱替換文件名和表格名稱。

```
db2 "LOAD CLIENT from /directory/file_name.txt \  
modified by coldel\| replace into TPCH.TABLE_NAME \  
nonrecoverable without prompting"
```

7. 終止您的連線。

```
db2 terminate
```

適用於 Db2 資料庫執行個體的 Amazon RDS 選項

以下顯示執行 Db2 資料庫引擎的 Amazon RDS 執行個體可用的選項或其他功能。若要啟用這些選項，您可以將它們新增至自訂選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需使用選項群組的相關詳細資訊，請參閱[使用選項群組](#)。

Amazon RDS 支援 Db2 的下列選項：

選項	選項 ID
Db2 稽核記錄	DB2_AUDIT

Db2 稽核記錄

透過 Db2 稽核記錄，Amazon RDS 會記錄資料庫活動，包括登入資料庫的使用者以及針對資料庫執行的查詢。RDS 會使用您提供的 AWS Identity and Access Management (IAM) 角色，將完成的稽核日誌上傳到您的 Amazon S3 儲存貯體。

主題

- [設定 Db2 稽核記錄](#)
- [管理 Db2 稽核記錄](#)
- [檢視稽核日誌](#)
- [疑難排解 Db2 稽核記錄](#)

設定 Db2 稽核記錄

若要啟用 Amazon RDS for Db2 資料庫的稽核記錄，您可以在適用於 Db2 資料庫執行個體的 RDS 上啟用 DB2_AUDIT 此選項。然後，設定稽核策略以啟用特定資料庫的功能。若要啟用 RDS for Db2 資料庫執行個體上的選項，請設定選項的選項設定 DB2_AUDIT。您可以透過為您的 Amazon Amazon S3 存貯體提供 Amazon 資源名稱 (ARN)，以及具有存取儲存貯體許可的 IAM 角色。

若要設定 RDS for Db2 資料庫的 Db2 稽核記錄，請完成下列步驟。

主題

- [步驟 1：建立 Amazon S3 儲存貯體](#)
- [步驟 2：建立 IAM 政策](#)
- [步驟 3：建立 IAM 角色並附加您的 IAM 政策](#)
- [步驟 4：設定 Db2 稽核記錄的選項群組](#)
- [步驟 5：設定稽核策略](#)
- [步驟 6：檢查稽核配置](#)

步驟 1：建立 Amazon S3 儲存貯體

如果您尚未這麼做，請建立一個 Amazon S3 儲存貯體，讓 Amazon RDS 可以在其中上傳您的 RDS for Db2 資料庫的稽核日誌檔。下列限制適用於使用為稽核檔案目標的 S3 儲存貯體：

- 它必須與您的 RDS for Db2 資料庫執行個體相同 AWS 區域。

- 它不可以開放給大眾使用。
- 它不能使用 [S3 物件鎖定](#)。
- 儲存貯體擁有者也必須是 IAM 角色擁有者。

若要了解如何建立 Amazon S3 儲存貯體，請參閱 Amazon S3 使用者指南中的 [建立](#) 儲存貯體。

啟用稽核記錄後，Amazon RDS 會自動將日誌從您的資料庫執行個體傳送到下列位置：

- 資料庫執行個體層級記錄 *bucket_name/db2-audit-logs/dbi_resource_id/date_time_utc/*
- 資料庫層級記錄 — *bucket_name/db2-audit-logs/dbi_resource_id/date_time_utc/db_name/*

記下您儲存桶的 Amazon 資源名稱 (ARN)。完成後續步驟需要此資訊。

步驟 2：建立 IAM 政策

建立 IAM 政策，具有將稽核日誌檔從資料庫執行個體傳輸到 Amazon S3 儲存貯體所需的許可。此步驟假設您有 S3 儲存貯體。

建立原則之前，請先收集下列資訊：

- 您的水桶的 ARN。
- 如果您的儲存桶使用 SSE-KMS 加密，則您的 AWS Key Management Service (AWS KMS) 密鑰的 ARN。

建立包含下列許可的 IAM 政策：

```
"s3:ListBucket",
"s3:GetBucketACL",
"s3:GetBucketLocation",
"s3:PutObject",
"s3:ListMultipartUploadParts",
"s3:AbortMultipartUpload",
"s3:ListAllMyBuckets"
```

Note

Amazon RDS 需要在內部 `s3:ListAllMyBuckets` 採取動作，以確認是否同時 AWS 帳戶擁有 S3 儲存貯體和適用於 Db2 資料庫執行個體的 RDS。

如果您的儲存貯體使用 SSE-KMS 加密功能，請同時包含下列權限：

```
"kms:GenerateDataKey",  
"kms:Decrypt"
```

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 建立 IAM 政策。

主控台

建立 IAM 政策以允許 Amazon RDS 存取您的 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇政策。
3. 選擇 [建立原則]，然後選擇 [JSON]。
4. 在新增動作中，依 S3 篩選。加入存取權 `ListBucket`、`GetBucketAcl` 和 `GetBucketLocation`。
5. 對於 [新增資源]，選擇 [新增]。針對資源類型，選擇值區，然後輸入值區的名稱。然後，選擇「新增資源」。
6. 選擇 [新增陳述式]。
7. 在新增動作中，依 S3 篩選。新增存取權 `PutObjectListMultipartUploadParts`、和 `AbortMultipartUpload`。
8. 對於 [新增資源]，選擇 [新增]。針對資源類型，選擇物件，然後輸入 `#####/*`。然後，選擇「新增資源」。
9. 選擇 [新增陳述式]。
10. 在新增動作中，依 S3 篩選。新增存取權 `ListAllMyBuckets`。
11. 對於 [新增資源]，選擇 [新增]。對於資源類型，選擇所有資源。然後，選擇「新增資源」。
12. 如果您使用自己的 KMS 金鑰來加密資料：
 1. 選擇 [新增陳述式]。

2. 在 [新增動作] 中，依 KMS 篩選。新增存取GenerateData金鑰並解密。
 3. 對於 [新增資源]，選擇 [新增]。對於資源類型，選擇所有資源。然後，選擇「新增資源」。
13. 選擇下一步。
 14. 在策略名稱中，輸入此策略的名稱。
 15. (選擇性) 在說明中，輸入此原則的說明。
 16. 選擇建立政策。

AWS CLI

建立 IAM 政策以允許 Amazon RDS 存取您的 Amazon S3 儲存貯體

1. 執行 `create-policy` 命令。在下列範例中，請使用 IAM 政策的 `##### Amazon S3 #####` `iam_` 政策名稱/名稱。

對於LinuxmacOS、或Unix：

```
aws iam create-policy \  
  --policy-name iam_policy_name \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "Statement1",  
        "Effect": "Allow",  
        "Action": [  
          "s3:ListBucket",  
          "s3:GetBucketAcl",  
          "s3:GetBucketLocation"  
        ],  
        "Resource": [  
          "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
        ]  
      },  
      {  
        "Sid": "Statement2",  
        "Effect": "Allow",  
        "Action": [  
          "s3:PutObject",  
          "s3:ListMultipartUploadParts",  
          "s3:AbortMultipartUpload"
```

```

        ],
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ]
    },
    {
        "Sid": "Statement3",
        "Effect": "Allow",
        "Action": [
            "s3:ListAllMyBuckets"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "Statement4",
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": [
            "*"
        ]
    }
]
}'

```

在 Windows 中：

```

aws iam create-policy ^
  --policy-name iam_policy_name ^
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "Statement1",
        "Effect": "Allow",
        "Action": [
          "s3:ListBucket",
          "s3:GetBucketAcl",
          "s3:GetBucketLocation"

```

```
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    ]
  },
  {
    "Sid": "Statement2",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
  },
  {
    "Sid": "Statement3",
    "Effect": "Allow",
    "Action": [
      "s3:ListAllMyBuckets"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "Statement4",
    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": [
      "*"
    ]
  }
]
}'
```

2. 建立原則之後，請記下原則的 ARN。你需要的 ARN 為 [步驟 3：建立 IAM 角色並附加您的 IAM 政策](#)。

如需建立 IAM 政策的相關資訊，請參閱 [IAM 使用者指南中的建立 IAM 政策](#)。

步驟 3：建立 IAM 角色並附加您的 IAM 政策

此步驟假設您已在中建立 IAM 政策 [步驟 2：建立 IAM 政策](#)。在此步驟中，您會為 RDS for Db2 資料庫執行個體建立 IAM 角色，然後將您的 IAM 政策附加到該角色。

您可以使用主控台或為資料庫執行個體建立 IAM 角色 AWS CLI。

主控台

建立 IAM 角色並將您的 IAM 政策附加到該角色

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 針對信任的實體類型，選取 AWS 服務。
5. 對於服務或使用案例，請選取 RDS，然後選取 RDS — 將角色新增至資料庫。
6. 選擇下一步。
7. 對於許可政策，搜尋並選取您建立的 IAM 政策名稱。
8. 選擇下一步。
9. 在 Role name (角色名稱) 中，輸入角色名稱。
10. (選用) 在 Description (說明) 中，輸入新角色的說明。
11. 選擇建立角色。

AWS CLI

建立 IAM 角色並將您的 IAM 政策附加到該角色

1. 執行 [create-role](#) 命令。在下列範例中，將 *iam_role_name* 取代為您的身分與存取權管理角色的名稱。

對於 Linux macOS、或 Unix：

```
aws iam create-role \  
  --role-name iam_role_name \  
  --assume-role-policy-document '{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

在 Windows 中：

```
aws iam create-role ^
--role-name iam_role_name ^
--assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

2. 建立角色之後，請注意此角色的 ARN。你需要這個 ARN 的下一步，[步驟 4：設定 Db2 稽核記錄的選項群組](#)。
3. 執行 [attach-role-policy](#) 命令。在下列範例中，將 *iam_policy_arn* 取代為您在中建立的 IAM 政策的 ARN。[步驟 2：建立 IAM 政策](#) 將 *iam_role_name* 取代為您剛建立的 IAM 角色的名稱。

對於LinuxmacOS、或Unix：

```
aws iam attach-role-policy \  
--policy-arn iam_policy_arn \  
--role-name iam_role_name
```

在 Windows 中：

```
aws iam attach-role-policy ^  
  --policy-arn iam_policy_arn ^  
  --role-name iam_role_name
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

步驟 4：設定 Db2 稽核記錄的選項群組

將 Db2 稽核記錄選項新增至 RDS for Db2 資料庫執行個體的程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 新增和設定所有必要的選項。
3. 將選項群組與資料庫執行個體建立關聯。

新增 Db2 稽核記錄選項後，您不需要重新啟動資料庫執行個體。只要選項群組處於作用中狀態，您就可以建立稽核，並將稽核日誌儲存於 S3 儲存貯體。

在資料庫執行個體的選項群組上新增和設定 Db2 稽核記錄

1. 選擇下列其中一項：
 - 使用現有的選項群組。
 - 建立自訂資料庫選項群組，並使用該選項群組。如需詳細資訊，請參閱[建立選項群組](#)。
2. 將 DB2_AU DIT 選項新增至選項群組，並設定選項設定。如需新增選項的詳細資訊，請參閱[將選項新增至選項群組](#)。
 - 對於 IAM_ROLE_ARN，請輸入您在中建立的 IAM 角色的 ARN。[the section called “建立 IAM 角色並附加您的 IAM 政策”](#)
 - 對於 S3_BUCKET_ARN，請輸入 S3 儲存貯體的 ARN，以用於您的 Db2 稽核日誌。儲存貯體必須與您的 RDS for Db2 資料庫執行個體所在的區域相同。與您輸入的 IAM 角色相關聯的政策必須允許對此資源執行必要的操作。
3. 將選項群組套用至新的或現有的資料庫執行個體。選擇下列其中一項：
 - 如果您建立了新的資料庫執行個體，在啟動執行個體時套用選項群組。
 - 針對現有的資料庫執行個體，可以修改執行個體，並附加新的選項群組，以套用選項群組。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

步驟 5：設定稽核策略

若要設定 RDS for Db2 資料庫的稽核原則，請使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼連線至資料庫。然後，使用數據庫的數據庫名稱和適用的參數值調用 `rdsadmin.configure_db_audit` 存儲過程。

下列範例會連線至資料庫，並為 `testdb` 稽核、檢查、目標、SECTAMINT、系統管理員和驗證等類別設定稽核原則。狀態值會 BOTH 記錄成功和失敗，NORMAL 預設 ERROR TYPE 為。如需如何使用此預存程序的詳細資訊，請參閱 [the section called "rdsadmin.configure_db_audit"](#)。

```
db2 "connect to rdsadmin user master_user using master_password"
db2 "call rdsadmin.configure_db_audit('testdb', 'ALL', 'BOTH', ?)"
```

步驟 6：檢查稽核配置

若要確保您的稽核策略設定正確，請檢查稽核配置的狀態。

若要檢查組態，請使用 RDS for Db2 rdsadmin 資料庫執行個體的主要使用者名稱和主要密碼連線至資料庫。然後，使用數據庫的數據庫名稱運行以下 SQL 語句。在下面的例子中，數據庫名稱是 `testdb`。

```
db2 "select task_id, task_type, database_name, lifecycle,
      varchar(bson_to_json(task_input_params), 500) as task_params,
      cast(task_output as varchar(500)) as task_output
      from table(rdsadmin.get_task_status(null, 'testdb', 'CONFIGURE_DB_AUDIT'))"
```

Sample Output

TASK_ID	TASK_TYPE	DATABASE_NAME	LIFECYCLE
2	CONFIGURE_DB_AUDIT	DB2DB	SUCCESS

... continued ...

TASK_PARAMS

```
{ "AUDIT_CATEGORY" : "ALL", "CATEGORY_SETTING" : "BOTH" }
```

... continued ...

TASK_OUTPUT

```
2023-12-22T20:27:03.029Z Task execution has started.
```

```
2023-12-22T20:27:04.285Z Task execution has completed successfully.
```

管理 Db2 稽核記錄

設定 Db2 稽核記錄後，您可以修改特定資料庫的稽核原則，或停用資料庫層級或整個資料庫執行個體的稽核記錄。您也可以變更日誌檔案上傳到的 Amazon S3 儲存貯體。

主題

- [修改 Db2 稽核策略](#)
- [修改記錄檔的位置](#)
- [停用 Db2 稽核記錄](#)

修改 Db2 稽核策略

若要修改特定 RDS for Db2 資料庫的稽核原則，請執行 `rdsadmin.configure_db_audit` 預存程序。使用此預存程序，您可以變更稽核策略的類別、類別設定和錯誤類型組態。如需詳細資訊，請參閱 [the section called “rdsadmin.configure_db_audit”](#)。

修改記錄檔的位置

若要變更日誌檔案上傳到的 Amazon S3 儲存貯體，請執行下列其中一個動作：

- 修改連接至 RDS for Db2 DB 執行個體的目前選項群組 — 將 `DB2_AUDIT` 選項的 `S3_BUCKET_ARN` 設定更新為指向新值區。此外，請務必更新附加至由連結選項群組中的設定指 `IAM_ROLE_ARN` 定的 IAM 角色所附加的 IAM 政策。此 IAM 政策必須為您的新值區提供必要的存取權限。如需 IAM 政策中所需許可的相關資訊，請參閱 [建立 IAM 政策](#)。
- 將 RDS for Db2 資料庫執行個體連接至不同的選項群組 — 修改資料庫執行個體以變更附加的選項群組。請確定使用正確的 `S3_BUCKET_ARN` 和設定來 `IAM_ROLE_ARN` 設定新選項群組。如需有關如何為 `DB2_AUDIT` 選項設定這些設定的資訊，請參閱 [設定選項群組](#)。

修改選項群組時，請務必立即套用變更。如需詳細資訊，請參閱 [the section called “修改資料庫執行個體”](#)。

停用 Db2 稽核記錄

若要停用 Db2 稽核記錄，請執行下列其中一個動作：

- 停用 RDS for Db2 資料庫執行個體的稽核記錄 — 修改您的資料庫執行個體，並使用選項從中移除 `DB2_AUDIT` 選項群組。如需詳細資訊，請參閱 [the section called “修改資料庫執行個體”](#)。

- 停用特定資料庫的稽核記錄 — `rdsadmin.disable_db_audit` 使用資料庫的資料庫名稱呼叫，以停止稽核記錄並移除稽核策略。如需詳細資訊，請參閱 [the section called "rdsadmin.disable_db_audit"](#)。

```
db2 "call rdsadmin.disable_db_audit(
    'db_name')"
```

檢視稽核日誌

啟用 Db2 稽核記錄後，請等待至少一小時，再檢視 Amazon S3 儲存貯體中的稽核資料。Amazon RDS 會自動將日誌從您的 RDS for Db2 資料庫執行個體傳送到下列位置：

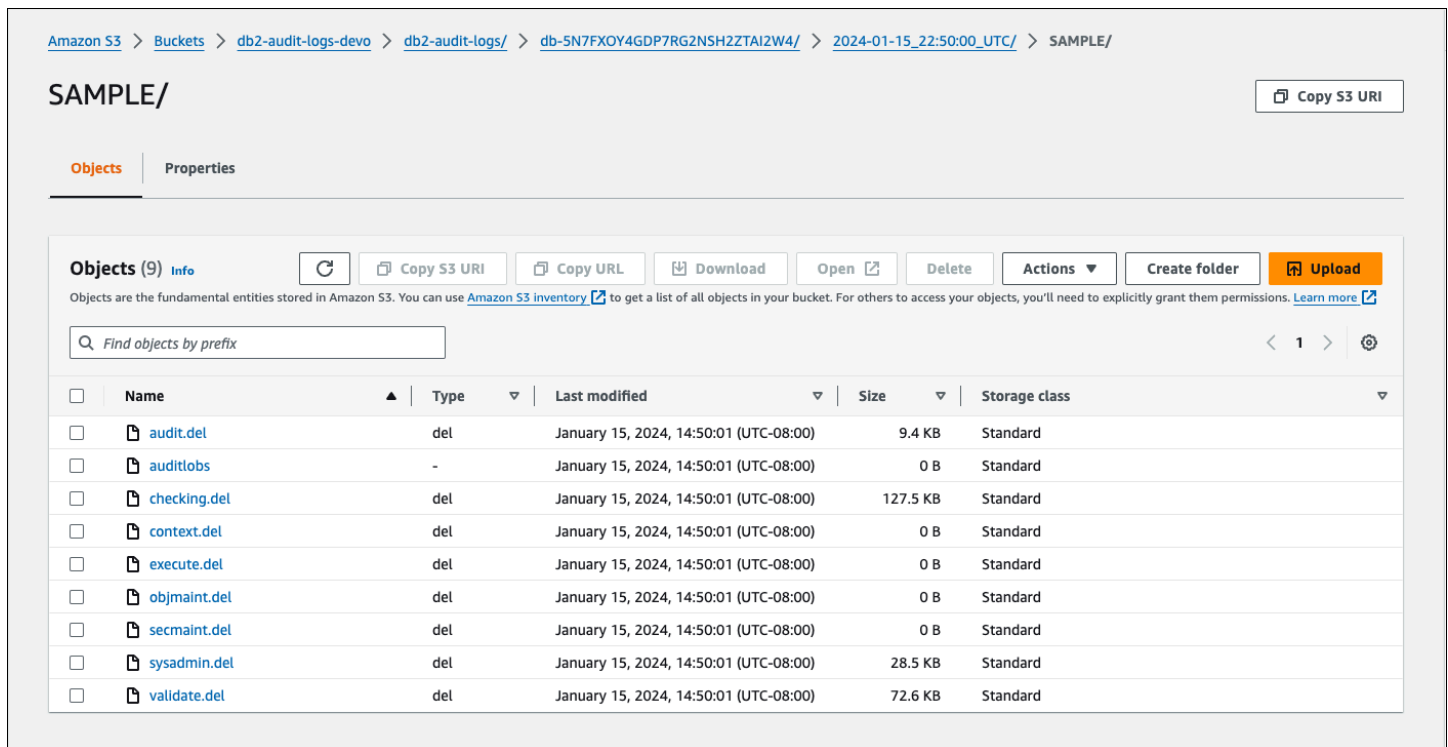
- 資料庫執行個體層級記錄 `bucket_name/db2-audit-logs/dbi_resource_id/date_time_utc/`
- 資料庫層級記錄 — `bucket_name/db2-audit-logs/dbi_resource_id/date_time_utc/db_name/`

下列 Amazon S3 主控台的螢幕擷取畫面範例顯示 RDS for Db2 資料庫執行個體層級日誌檔的資料夾清單。

The screenshot shows the Amazon S3 console interface for a bucket named '2024-01-15_22:50:00_UTC/'. The breadcrumb path is: Amazon S3 > Buckets > db2-audit-logs-dev0 > db2-audit-logs/ > db-5N7FXOY4GDP7RG2NSH2ZTAI2W4/ > 2024-01-15_22:50:00_UTC/. The console displays a list of objects with columns for Name, Type, Last modified, Size, and Storage class. The objects listed are:

Name	Type	Last modified	Size	Storage class
audit.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	9.4 KB	Standard
auditlobs	-	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
checking.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	127.5 KB	Standard
context.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
execute.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
objmaint.del	del	January 15, 2024, 14:50:02 (UTC-08:00)	0 B	Standard
SAMPLE/	Folder	-	-	-
secmaint.del	del	January 15, 2024, 14:50:02 (UTC-08:00)	0 B	Standard
sysadmin.del	del	January 15, 2024, 14:50:02 (UTC-08:00)	28.5 KB	Standard
validate.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	72.6 KB	Standard

下列 Amazon S3 主控台的螢幕擷取畫面範例顯示 RDS for Db2 資料庫執行個體的資料庫層級日誌檔。



The screenshot shows the Amazon S3 console interface for a bucket named 'SAMPLE/'. The breadcrumb navigation path is: Amazon S3 > Buckets > db2-audit-logs-dev0 > db2-audit-logs/ > db-5N7FXOY4GDP7RG2NSH2ZTAI2W4/ > 2024-01-15_22:50:00_UTC/ > SAMPLE/. The console displays a list of 9 objects, all with a 'del' type and 'Standard' storage class, all modified on January 15, 2024, at 14:50:01 (UTC-08:00). The objects are:

Name	Type	Last modified	Size	Storage class
audit.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	9.4 KB	Standard
auditlobs	-	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
checking.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	127.5 KB	Standard
context.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
execute.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
objmaint.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
secmaint.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	0 B	Standard
sysadmin.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	28.5 KB	Standard
validate.del	del	January 15, 2024, 14:50:01 (UTC-08:00)	72.6 KB	Standard

疑難排解 Db2 稽核記錄

使用下列資訊來疑難排解 Db2 稽核記錄的常見問題。

無法設定稽核策略

如果呼叫預存程序 `rdsadmin.configure_db_audit` 傳回錯誤，可能是具有選項的選項群組與 RDS for Db2 資料庫執行個體沒有關聯。DB2_AUDIT 修改資料庫執行個體以新增選項群組，然後嘗試再次呼叫預存程序。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Amazon S3 存儲桶中沒有數據

如果 Amazon S3 儲存貯體遺失記錄資料，請檢查下列各項：

- Amazon S3 儲存貯體與您的 RDS for Db2 資料庫執行個體所在的區域相同。
- 您在 IAM_ROLE_ARN 選項設定中指定的角色已設定具有將日誌上傳到 Amazon S3 儲存貯體的必要許可。如需詳細資訊，請參閱 [建立 IAM 政策](#)。
- 與 RDS for Db2 資料庫執行個體相關聯的選項群組中，IAM_ROLE_ARN 和 S3_BUCKET_ARN 選項設定的 ARN 是正確的。如需詳細資訊，請參閱 [設定選項群組](#)。

您可以連線至資料庫並執行 SQL 陳述式，以檢查稽核記錄組態的工作狀態。如需詳細資訊，請參閱 [檢查稽核配置](#)。

您也可以檢查事件，進一步瞭解可能遺失記錄檔的原因。若要取得有關如何檢視事件的資訊，請參閱 [the section called “在 Amazon RDS 主控台中檢視日誌、事件和串流”](#)。

適用於 Db2 的 Amazon RDS 的外部存儲程序

您可以建立外部常式，並將它們註冊到 Amazon RDS for Db2 資料庫做為外部存放程序。目前，RDS for Db2 僅支援外部預存程序的 Java 型常式。

基於 Java 的外部預存程序

基於 Java 的外部預存程序是您向 RDS for Db2 資料庫註冊為外部預存程序的外部 Java 常式。

主題

- [Java 型外部預存程序的限制](#)
- [設定 Java 型外部預存程序](#)

Java 型外部預存程序的限制

在開發外部例程序之前，請考慮以下限制和限制。

若要建立外部常式，請務必使用 Db2 提供的 Java 開發工具組 (JDK)。如需詳細資訊，請參閱 [Db2 資料庫產品的 Java 軟體支援](#)。

您的 Java 程式只能在 /tmp 目錄中建立檔案，而 Amazon RDS 不支援在這些檔案上啟用可執行檔或設定使用者 ID (SUID) 許可。您的 Java 程序也不能使用套接字系統調用或以下系統調用：

- _sysctl
- acct
- afs_syscall
- bpf
- capset
- chown
- chroot
- create_module
- delete_module
- fanotify_init
- fanotify_mark
- finit_module

- fsconfig
- fsopen
- fspick
- get_kernel_syms
- getpmsg
- init_module
- mount
- move_mount
- nfsservctl
- open_by_handle_at
- open_tree
- pivot_root
- putpmsg
- query_module
- quotactl
- reboot
- security
- setdomainname
- setfsuid
- sethostname
- sysfs
- tuxcall
- umount2
- uselib
- ustat
- vhangup
- vserver

如需 Db2 外部常式的其他限制，請參閱[IBM Db2文件中對外部常式的限制](#)。

設定 Java 型外部預存程序

若要設定外部預存程序，請使用外部常式建立 .jar 檔案，將其安裝在 RDS for Db2 資料庫上，然後將其註冊為外部預存程序。

主題

- [步驟 1：啟用外部預存程序](#)
- [步驟 2：使用外部例程安裝 .jar 文件](#)
- [步驟 3：註冊外部預存程序](#)
- [步驟 4：驗證外部預存程序](#)

步驟 1：啟用外部預存程序

若要啟用外部預存程序，請在與資料庫執行個體關聯的自訂參數群組中，db2_alter_authz_behaviour將參數設定為下列其中一個值：

- EXTERNAL_ROUTINE_DBADM— 隱含地授與DBADM具有CREATE_EXTERNAL_ROUTINE權限的任何使用者、群組或角色。
- EXTERNAL_ROUTINE_DBAUTH— 允許有權限的使用者將DBADMCREATE_EXTERNAL_ROUTINE權限授與任何使用者、群組或角色。在此情況下，不會以隱含方式授與使用者、群組或角色，即使是具有權限的使用者也不會被隱含DBADM授與。

如需有關此設定的詳細資訊，請參閱IBM Db2文件中的 [GRANT \(資料庫授權單位\) 陳述式](#)。

您可以使用 AWS Management Console、或 Amazon RDS API 來建立和修改自訂參數群組。AWS CLI

主控台

在自訂參數群組中規劃 db2_authz_行為參數的步驟

1. 如果您想要使用與資料庫執行個體使用的自訂資料庫參數群組不同的自訂資料庫參數群組，請建立新的資料庫參數群組。如果您使用的是自攜授權 (BYOL) 模型，請確定新的自訂參數群組包含 ID。IBM如需有關這些 ID 的資訊，請參閱[the section called “IBMDb2 使用您自己的授權的 ID”](#)。如需建立資料庫參數群組的詳細資訊，請參閱[建立資料庫參數群組](#)。
2. 在自訂db2_alter_authz_behaviour參數群組中設定參數的值。如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

AWS CLI

在自訂參數群組中規劃 db2_authz_行為參數的步驟

1. 如果您想要使用與資料庫執行個體所使用的自訂資料庫參數群組不同的自訂資料庫參數群組，請執行[create-db-parameter-group](#)指令來建立自訂參數群組。如果您使用的是自攜授權 (BYOL) 模型，請確定新的自訂參數群組包含 ID。IBM如需有關這些 ID 的資訊，請參閱[the section called “IBMDb2 使用您自己的授權的 ID”](#)。

包含下列必要選項：

- `--db-parameter-group-name`— 您要建立之參數群組的名稱。
- `--db-parameter-group-family`— Db2 引擎版本和主要版本。有效值為 `db2-se-11.5` 和 `db2-ae-11.5`。
- `--description`— 此參數群組的描述。

如需建立資料庫參數群組的詳細資訊，請參閱[建立資料庫參數群組](#)。

以下範例展示如何建立為參數群組族群命名 `MY_EXT_SP_PARAM_GROUP` 的自訂參數群組 `db2-se-11.5`。

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
--region us-east-1 \  
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP \  
--db-parameter-group-family db2-se-11.5 \  
--description "test db2 external routines"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^  
--region us-east-1 ^  
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^  
--db-parameter-group-family db2-se-11.5 ^  
--description "test db2 external routines"
```

2. 透過執行 `db2_alternate_authz_behaviour` [modify-db-parameter-group](#) 指令來修改自訂參數群組中的參數。

包含下列必要選項：

- `--db-parameter-group-name`— 您建立的參數群組名稱。
- `--parameters`— 參數名稱、值以及參數更新的應用程式方法陣列。

如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

下列範例說明如何MY_EXT_SP_PARAM_GROUP透過將值設定為db2_alternate_authz_behaviour來修改參數群組EXTERNAL_ROUTINE_DBADM。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name MY_EXT_SP_PARAM_GROUP \  
  --parameters  
  "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^  
  --parameters  
  "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

RDS API

在自訂參數群組中規劃 db2_authz_ 行為參數的步驟

1. 如果您想要使用與資料庫執行個體使用的自訂資料庫參數群組不同的自訂資料庫參數群組，請使用 Amazon RDS API [CreateDBParameterGroup](#) 操作建立新的資料庫參數群組。如果您使用的是自攜授權 (BYOL) 模型，請確定新的自訂參數群組包含 ID。IBM Db2如需有關這些 ID 的資訊，請參閱[the section called “IBMDB2 使用您自己的授權的 ID”](#)。

包含下列必要參數：

- DBParameterGroupName
- DBParameterGroupFamily
- Description

如需建立資料庫參數群組的詳細資訊，請參閱[建立資料庫參數群組](#)。

2. 修改db2_alter_authz_behaviour您使用 RDS API [ModifyDBParameterGroup](#) 作業建立的自訂參數群組中的參數。

包含下列必要參數：

- DBParameterGroupName
- Parameters

如需修改參數群組的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

步驟 2：使用外部例程安裝 .jar 文件

建立 Java 常式之後，請建立 .jar 檔案，然後執行db2 "call sqlj.install_jar('file:*file_path*',*jar_ID*)"以將其安裝在您的 RDS for Db2 資料庫上。

下列範例會示範如何建立 Java 常式，並將其安裝在 RDS for Db2 資料庫上。此範例包含簡單常式的範例程式碼，可用來測試程序。此範例會做出下列假設：

- Java 代碼是在安裝 Db2 的服務器上編譯。這是最佳做法，因為不使用 IBM 提供的 JDK 進行編譯可能會導致無法解釋的錯誤。
- 伺服器具有在本機編目的 RDS for Db2 資料庫。

如果您想要使用下列範例程式碼來嘗試此程序，請複製該程式碼，然後將其儲存到名為的檔案中MYJAVASP.java。

```
import java.sql.*;
public class MYJAVASP
{
    public static void my_JAVASP (String inparam) throws SQLException, Exception
    {
        try
        {
            // Obtain the calling context's connection details.
            Connection myConn = DriverManager.getConnection("jdbc:default:connection");
            String myQuery = "INSERT INTO TEST.TEST_TABLE VALUES (?, CURRENT DATE)";
            PreparedStatement myStmt = myConn.prepareStatement(myQuery);
```

```
myStmt.setString(1, inparam);
myStmt.executeUpdate();
}
catch (SQLException sql_ex)
{
throw sql_ex;
}
catch (Exception ex)
{
throw ex;
}
}
```

下面的命令編譯 Java 例程。

```
~/sqlllib/java/jdk64/bin/javac MYJAVASP.java
```

下面的命令創建 .jar 文件。

```
~/sqlllib/java/jdk64/bin/jar cvf MYJAVASP.jar MYJAVASP.class
```

以下命令連接到名為的數據庫MY_DB2_DATABASE並安裝 .jar 文件。

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

db2 "call sqlj.install_jar('file:/tmp/MYJAVASP.jar', 'MYJAVASP')"
db2 "call sqlj.refresh_classes()"
```

步驟 3：註冊外部預存程序

在 RDS for Db2 資料庫上安裝 .jar 檔案之後，請執行db2 CREATE PROCEDURE或db2 REPLACE PROCEDURE命令將其註冊為預存程序。

下列範例說明如何連線至資料庫，並將在上一個步驟中建立的 Java 常式註冊為預存程序。

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

create procedure TESTSP.MYJAVASP (in input char(6))
specific myjavasp
dynamic result sets 0
deterministic
```

```
language java
parameter style java
no dbinfo
fenced
threadsafe
modifies sql data
program type sub
external name 'MYJAVASP!my_JAVASP';
```

步驟 4：驗證外部預存程序

使用下列步驟來測試在上一個步驟中註冊的範例外部預存程序。

若要驗證外部預存程序

1. 創建一個表，如下面的例子。TEST.TEST_TABLE

```
db2 "create table TEST.TEST_TABLE(C1 char(6), C2 date)"
```

2. 呼叫新的外部預存程序。呼叫會傳回的狀態0。

```
db2 "call TESTSP.MYJAVASP('test')"  
Return Status = 0
```

3. 查詢您在步驟 1 中建立的資料表，以驗證預存程序呼叫的結果。

```
db2 "SELECT * from TEST.TEST_TABLE"
```

查詢會產生類似下列範例的輸出：

```
C1      C2  
-----  
test    02/05/2024
```

適用於 Db2 的 Amazon RDS 的已知問題和限制

下列項目是使用適用於 Db2 的 Amazon RDS 的已知問題和限制：

主題

- [驗證限制](#)
- [非圍欄例程](#)
- [移轉期間的非自動儲存體表格空間](#)

驗證限制

Amazon RDS 設置DB2AUTH為JCC_ENFORCE_SECMEC。由於JCC_ENFORCE_SECMEC無法修改，因此 Amazon RDS 會在 JDBC 連線強執行密碼加密。

非圍欄例程

適用於 Db2 的 RDS 不支援建立非圍欄常式。若要檢查資料庫是否包含任何非圍欄常式，請執行下列 SQL 命令：

```
SELECT 'COUNT:' || count(*) FROM SYSCAT.ROUTINES where fenced='N' and routineschema not in ('SQLJ', 'SYSCAT', 'SYSFUN', 'SYSIBM', 'SYSIBMADM', 'SYSPROC', 'SYSTOOLS')
```

移轉期間的非自動儲存體表格空間

RDS for Db2 不支援建立新的非自動儲存體表格空間。當您使用原生還原進行資料庫的一次性移轉時，RDS for Db2 會自動將非自動儲存體表格空間轉換為自動的表格空間，然後將資料庫還原至 RDS for Db2。如需有關一次性移轉的資訊，請參閱[一次性從Linux環境Linux遷移](#)和[在Linux環境之間Windows進AIX行一次性移轉](#)。

適用於 Db2 的 Amazon RDS 存儲過程參考

這些主題說明可用於執行 Db2 引擎的 Amazon RDS for Db2 資料庫執行個體的系統預存程序。若要執行這些程序，master 使用者必須先連線到資源 rdsadmin 料庫。

主題

- [授與和撤銷權限](#)
- [管理緩衝集區](#)
- [管理資料庫](#)
- [管理表格空間](#)
- [管理稽核策略](#)

授與和撤銷權限

下列預存程序授與和撤銷 Amazon RDS for Db2 資料庫的權限。若要執行這些程序，master 使用者必須先連線到資料庫。

主題

- [rdsadmin.create_role](#)
- [rdsadmin.grant_role](#)
- [rdsadmin.revoke_role](#)
- [rdsadmin.add_user](#)
- [rdsadmin.change_password](#)
- [rdsadmin.list_users](#)
- [rdsadmin.remove_user](#)
- [rdsadmin.add_groups](#)
- [rdsadmin.remove_groups](#)
- [rdsadmin.dbadm_grant](#)
- [rdsadmin.dbadm_revoke](#)

rdsadmin.create_role

建立角色。

語法

```
db2 "call rdsadmin.create_role(  
    'database_name',  
    'role_name')"
```

參數

下列是必要參數：

database_name

命令將在其上運行的數據庫的名稱。資料類型為varchar。

role_name

您要建立的角色名稱。資料類型為varchar。

使用須知

如需有關檢查建立角色狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會建立呼叫 Database MY_ROLE 的角色DB2DB。

```
db2 "call rdsadmin.create_role(  
    'DB2DB',  
    'MY_ROLE')"
```

rdsadmin.grant_role

將角色指派給角色、使用者或群組。

語法

```
db2 "call rdsadmin.grant_role(  
    ?,  
    'database_name',  
    'role_name',  
    'grantee',  
    'admin_option')"
```

參數

需要以下輸出參數：

?

輸出工作唯一識別元的參數標記。此參數僅接受?。

以下是必需的輸入參數：

database_name

命令將在其上運行的數據庫的名稱。資料類型為varchar。

role_name

您要建立的角色名稱。資料類型為varchar。

###

要接收授權的角色、使用者或群組。資料類型為varchar。有效值：ROLE、USER、GROUP、PUBLIC。

格式必須是值，後跟名稱。用逗號分隔多個值和名稱。範例：'USER *user1*, *user2*, GROUP *group1*, *group2*'。用您自己的信息替換名稱。

以下是可選的輸入參數：

####

指定受權者是否ROLE具有指派角色的DBADM授權。資料類型為char。預設值為N。

使用須知

如需有關檢查指派角色狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會將呼叫 ROLE_TEST for database 的角色指派TESTDB給稱為的角色role1、呼叫user1的使用者以及呼叫的群組group1。ROLE_TEST被賦予管理員授權來分配角色。

```
db2 "call rdsadmin.grant_role(  
    ?,  
    'TESTDB',  
    'ROLE_TEST',  
    'ROLE role1, USER user1, GROUP group1',  
    'Y')"
```

下列範例會將呼叫 Database ROLE_TEST 的角色指派TESTDB給PUBLIC。ROLE_TEST沒有授予管理員授權來分配角色。

```
db2 "call rdsadmin.grant_role(  
    ?,  
    'TESTDB',  
    'ROLE_TEST',
```



```
'PUBLIC')"
```

rdsadmin.revoke_role

撤銷角色、使用者或群組的角色。

語法

```
db2 "call rdsadmin.revoke_role(  
    ?,  
    'database_name',  
    'role_name',  
    'grantee')"
```

參數

需要以下輸出參數：

?

輸出工作唯一識別元的參數標記。此參數只接受?。

以下是必需的輸入參數：

database_name

命令將在其上運行的數據庫的名稱。資料類型為varchar。

role_name

您要撤銷的角色名稱。資料類型為varchar。

###

要失去授權的角色、使用者或群組。資料類型為varchar。有效值：ROLE、USER、GROUP、PUBLIC。

格式必須是值，後跟名稱。用逗號分隔多個值和名稱。範例：'USER *user1*, *user2*, GROUP *group1*, *group2*'。用您自己的信息替換名稱。

使用須知

如需有關檢查指派角色狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會TESTDB從名為的角色role1、呼叫ROLE_TEST的使用者以及呼叫的群組撤銷呼叫user1的Database角色。group1

```
db2 "call rdsadmin.revoke_role(  
    ?,  
    'TESTDB',  
    'ROLE_TEST',  
    'ROLE role1, USER user1, GROUP group1')"
```

下列範例會撤銷呼叫 Database ROLE_TEST TESTDB 的角色。PUBLIC

```
db2 "call rdsadmin.revoke_role(  
    ?,  
    'TESTDB',  
    'ROLE_TEST',  
    'PUBLIC')"
```

rdsadmin.add_user

將使用者新增至授權清單。

語法

```
db2 "call rdsadmin.add_user(  
    'username',  
    'password',  
    'group_name,group_name')"
```

參數

下列是必要參數：

username

使用者的使用者名稱。資料類型為varchar。

###

使用者的密碼。資料類型為varchar。

以下參數為選用：

group_name

您要新增使用者的群組名稱。資料類型為varchar。預設值為空字串或 null。

使用須知

您可以使用逗號分隔群組名稱，將使用者新增至一或多個群組。

您可以在建立新使用者或將群組新增至現有使用者時建立群組。您無法自行建立群組。

Note

您可以透過撥打電話新增的使用者數目上限rdsadmin.add_user為 5,000 位。

如需檢查新增使用者狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會建立呼叫的使用者，jorge_souza並將使用者指派給名為sales和的群組inside_sales。

```
db2 "call rdsadmin.add_user(  
    'jorge_souza',  
    '*****',  
    'sales,inside_sales')"
```

rdsadmin.change_password

變更使用者的密碼。

語法

```
db2 "call rdsadmin.change_password(  
    'username',  
    'new_password')"
```

參數

下列是必要參數：

username

使用者的使用者名稱。資料類型為varchar。

(_R)

使用者的新密碼。資料類型為varchar。

使用須知

如需檢查變更密碼狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會變更的密碼jorge_souza。

```
db2 "call rdsadmin.change_password(  
    'jorge_souza',  
    '*****')"
```

rdsadmin.list_users

列出授權清單上的使用者。

語法

```
db2 "call rdsadmin.list_users()"
```

使用須知

如需有關檢查清單使用者狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

rdsadmin.remove_user

從授權列表中刪除用戶。

語法

```
db2 "call rdsadmin.remove_user('username')"
```

參數

以下是必要參數：

username

使用者的使用者名稱。資料類型為varchar。

使用須知

如需檢查移除使用者狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會移jorge_souza除無法存取 RDS for Db2 資料庫執行個體中的資料庫。

```
db2 "call rdsadmin.remove_user('jorge_souza')"
```

rdsadmin.add_groups

將群組新增至使用者。

語法

```
db2 "call rdsadmin.add_groups(  
    'username',  
    'group_name,group_name')"
```

參數

下列是必要參數：

username

使用者的使用者名稱。資料類型為varchar。

group_name

您要新增使用者的群組名稱。資料類型為varchar。預設為空字串。

使用須知

您可以使用逗號分隔群組名稱，將一或多個群組新增至使用者。如需檢查新增群組狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會將direct_sales和b2b_sales群組新增至使用者jorge_souza。

```
db2 "call rdsadmin.add_groups(  
    'jorge_souza',  
    'direct_sales,b2b_sales')"
```

rdsadmin.remove_groups

從使用者中移除群組。

語法

```
db2 "call rdsadmin.remove_groups(  
    'username',  
    'group_name,group_name')"
```

參數

下列是必要參數：

username

使用者的使用者名稱。資料類型為varchar。

group_name

您要從中移除使用者的群組名稱。資料類型為varchar。

使用須知

您可以使用逗號分隔群組名稱，從使用者中移除一或多個群組。

如需有關檢查移除群組狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會從使用者移除direct_sales和b2b_sales群組jorge_souza。

```
db2 "call rdsadmin.remove_groups(  
    'jorge_souza',  
    'direct_sales,b2b_sales')"
```

rdsadmin.dbadm_grant

授與DBADM或DATAACCESS授權角色、使用者或群組。ACCESSCTRL

語法

```
db2 "call rdsadmin.dbadm_grant(  
    ?,  
    'database_name',  
    'authorization',  
    'grantee')"
```

參數

需要以下輸出參數：

?

輸出工作唯一識別元的參數標記。此參數僅接受?。

以下是必需的輸入參數：

database_name

命令將在其上運行的數據庫的名稱。資料類型為varchar。

authorization

要授予的授權類型。資料類型為varchar。有效值：DBADM、ACCESSCTRL、DATAACCESS。

用逗號分隔多個類型。

###

要接收授權的角色、使用者或群組。資料類型為varchar。有效值：ROLE、USER、GROUP。

格式必須是值，後跟名稱。用逗號分隔多個值和名稱。範例：'USER *user1*, *user2*, GROUP *group1*, *group2*'。用您自己的信息替換名稱。

使用須知

接收存取權的角色必須存在。

如需有關檢查授與資料庫管理員存取權的狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會將資料庫管理員存取權授與TESTDB為角色命名的資料庫ROLE_DBA。

```
db2 "call rdsadmin.dbadm_grant(  
  ?,  
  'TESTDB',  
  'DBADM',  
  'ROLE ROLE_DBA')"
```

下列範例會授user1與名TESTDB為 for 和的資料庫管理員存取權group1。

```
db2 "call rdsadmin.dbadm_grant(  
  ?,  
  'TESTDB',  
  'DBADM',  
  'USER user1, GROUP group1')"
```

下列範例會授與資料庫管理員存取權，存取名TESTDB為 for user1、user2group1、和的資料庫group2。

```
db2 "call rdsadmin.dbadm_grant(  
  ?,  
  'TESTDB',  
  'DBADM',  
  'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.dbadm_revoke

撤銷角色DBADMACCESSCTRL、使用者或群組的DATAACCESS授權。

語法

```
db2 "call rdsadmin.dbadm_revoke(  
  ?,  
  'database_name',  
  'authorization',  
  'grantee')"
```


參數

需要以下輸出參數：

?

任務的唯一識別符。此參數僅接受?。

以下是必需的輸入參數：

database_name

命令將在其上運行的數據庫的名稱。資料類型為varchar。

authorization

要撤銷的授權類型。資料類型為varchar。有效值：DBADM、ACCESSCTRL、DATAACCESS。

用逗號分隔多個類型。

###

要撤銷授權的角色、使用者或群組。資料類型為varchar。有效值：ROLE、USER、GROUP。

格式必須是值，後跟名稱。用逗號分隔多個值和名稱。範例：'USER *user1*, *user2*, GROUP *group1*, *group2*'。用您自己的信息替換名稱。

使用須知

如需檢查撤銷資料庫管理員存取權之狀態的詳細資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會撤銷資料庫管理員TESTDB對角色ROLE_DBA命名之資料庫的存取權。

```
db2 "call rdsadmin.dbadm_revoke(  
    ?,  
    'TESTDB',  
    'DBADM',  
    'ROLE ROLE_DBA')"
```

下列範例會撤銷名TESTDB為 for user1 和group1的資料庫的資料庫管理員存取權。

```
db2 "call rdsadmin.dbadm_revoke(  
  ?,  
  'TESTDB',  
  'DBADM',  
  'USER user1, GROUP group1')"
```

下列範例會撤銷名TESTDB為user1、user2和group2之資料庫的資料庫管理員存取權。group1

```
db2 "call rdsadmin.dbadm_revoke(  
  ?,  
  'TESTDB',  
  'DBADM',  
  'USER user1, user2, GROUP group1, group2')"
```

管理緩衝集區

下列預存程序會管理 Amazon RDS for Db2 資料庫的緩衝集區。若要執行這些程序，master 使用者必須先連線到資料庫。

主題

- [rdsadmin.create_bufferpool](#)
- [rdsadmin.alter_bufferpool](#)
- [rdsadmin.drop_bufferpool](#)

rdsadmin.create_bufferpool

創建一個緩衝池。

語法

```
db2 "call rdsadmin.create_bufferpool(  
    'database_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    'immediate',  
    'automatic',  
    page_size,  
    number_block_pages,  
    block_size)"
```

參數

下列是必要參數：

database_name

執行命令的資料庫名稱。資料類型為varchar。

#####

要建立的緩衝集區名稱。資料類型為varchar。

下列是選用參數：

####

緩衝集區的頁數大小。資料類型為integer。預設值為 -1。

##

指定命令是否立即執行。資料類型為char。預設值為 Y。

##

指定是否將緩衝集區設定為自動。資料類型為char。預設值為 Y。

####

緩衝集區的頁面大小。資料類型為integer。有效值：4096、8192、16384、32768。預設值為 8192。

#####

緩衝區集區中的區塊頁面數目。資料類型為integer。預設值為 0。

####

區塊頁面的區塊大小。資料類型為integer。有效值：2 到 256。預設值為 32。

使用須知

如需檢查建立緩衝集區狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會針對使用預設參數呼叫BP8的資料庫建立呼叫TESTDB的緩衝集區，因此緩衝集區會使用 8 KB 的頁面大小。

```
db2 "call rdsadmin.create_bufferpool(  
    'TESTDB',  
    BP8)"
```

下列範例會BP16針對名為的資料庫建立呼叫的緩衝集區，TESTDB該緩衝集區使用 16 KB 頁面大小，初始頁面計數為 1,000，且設定為自動。Db2 會立即執行命令。如果您使用 -1 的初始頁數，則 Db2 將使用自動分配頁面。

```
db2 "call rdsadmin.create_bufferpool(  
    'TESTDB',
```

```
'BP16',  
1000,  
'Y',  
'Y',  
16384)"
```

下列範例會建立呼叫的資料庫BP16的緩衝集區TESTDB。此緩衝集區的頁面大小為 16 KB，初始頁面計數為 10,000。Db2 會使用 500 個區塊大小為 512 的封鎖頁面，立即執行命令。

```
db2 "call rdsadmin.create_bufferpool(  
    'TESTDB',  
    'BP16',  
    10000,  
    'Y',  
    'Y',  
    16384,  
    500,  
    512)"
```

rdsadmin.alter_bufferpool

改變緩衝池。

語法

```
db2 "call rdsadmin.alter_bufferpool(  
    'database_name',  
    'buffer_pool_name',  
    buffer_pool_size,  
    'immediate',  
    'automatic',  
    change_number_blocks,  
    number_block_pages,  
    block_size)"
```

參數

下列是必要參數：

database_name

執行命令的資料庫名稱。資料類型為varchar。

#####

要變更之緩衝集區的名稱。資料類型為varchar。

#####

緩衝集區的頁數大小。資料類型為integer。

下列是選用參數：

##

指定命令是否立即執行。資料類型為char。預設值為 Y。

##

指定是否將緩衝集區設定為自動。資料類型為char。預設值為 N。

#####

指定緩衝集區中的區塊頁面數目是否有變更。資料類型為char。預設值為 N。

#####

緩衝區集區中的區塊頁面數目。資料類型為integer。預設值為 0。

####

區塊頁面的區塊大小。資料類型為integer。有效值：2 到 256。預設值為 32。

使用須知

如需檢查變更緩衝集區之狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會變更呼叫為BP16非自動資料庫的緩衝區集區，並TESTDB將大小變更為 10,000 個頁面。Db2 會立即執行此命令。

```
db2 "call rdsadmin.alter_bufferpool(  
    'TESTDB',  
    'BP16',  
    10000,  
    'Y',
```

```
'N')"
```

rdsadmin.drop_bufferpool

卸除緩衝集區。

語法

```
db2 "call rdsadmin.drop_bufferpool(  
    'database_name',  
    'buffer_pool_name'"
```

參數

下列是必要參數：

database_name

緩衝集區所屬的資料庫名稱。資料類型為varchar。

#####

要刪除的緩衝集區的名稱。資料類型為varchar。

使用須知

如需有關檢查卸除緩衝集區之狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會捨棄呼叫的資料庫BP16的緩衝集區TESTDB。

```
db2 "call rdsadmin.drop_bufferpool(  
    'TESTDB',  
    'BP16')"
```

管理資料庫

下列預存程序會管理適用於 Db2 的 Amazon RDS 資料庫。若要執行這些程序，master 使用者必須先連線到 `rdsadmin` 資料庫。

主題

- [rdsadmin.create_database](#)
- [rdsadmin.drop_database](#)
- [rdsadmin.update_db_param](#)
- [rdsadmin.set_configuration](#)
- [rdsadmin.show_configuration](#)
- [rdsadmin.restore_database](#)
- [rdsadmin.rollforward_database](#)
- [rdsadmin.complete_rollforward](#)
- [rdsadmin.db2pd_command](#)
- [rdsadmin.force_application](#)
- [rdsadmin.set_archive_log_retention](#)
- [rdsadmin.show_archive_log_retention](#)

rdsadmin.create_database

建立資料庫。

語法

```
db2 "call rdsadmin.create_database('database_name')"
```

參數

Note

此預存程序不會驗證所需參數的組合。當您調用時 [rdsadmin.get_task_status](#)，用戶定義函數可能會返回錯誤，因為 `database_codesetdatabase_territory`，的組合 `database_collation` 是無效的。如需詳細資訊，請參閱 IBM Db2 文件中的 [選擇資料庫的字碼頁、區域和定序](#)。

以下是必要參數：

database_name

要建立的資料庫名稱。資料類型為varchar。

下列是選用參數：

#####

資料庫的預設頁面大小。有效值：4096、8192、16384、32768。資料類型為integer。預設值為8192。

⚠ Important

Amazon RDS 支援 4 KiB、8 KiB 和 16 KiB 頁面的寫入原子性。相比之下，32 KiB 頁面會造成寫入撕裂或部分資料寫入桌面的風險。如果您使用 32 KiB 頁面，建議您啟用 point-in-time 復原和自動備份。否則，您會有無法從撕裂頁面復原的風險。如需詳細資訊，請參閱 [the section called “備份簡介”](#) 及 [the section called “Point-in-time 回收”](#)。

#####

為資料庫設定的程式碼。資料類型為varchar。預設值為 UTF-8。

#####

資料庫的兩個字母的國家代碼。資料類型為varchar。預設值為 US。

#####

決定儲存在資料庫中字元字串的排序與比較方式的定序順序。資料類型為varchar。

有效值：

- COMPATIBILITY— 一個 IBM Db2 第 2 版的排序規則順序。
- EBCDIC_819_037— ISO 拉丁文字碼頁，整理規則；國際化驗證碼 037 (美國英文版)。
- EBCDIC_819_500— ISO 拉丁文字碼頁，整理規則；國際化驗證碼 500 (國際版)。
- EBCDIC_850_037— ASCII 拉丁文字碼頁，定序；CCSID 037 (美國英文版)。
- EBCDIC_850_500— ASCII 拉丁文字碼頁，定序；國際化驗證碼 500

- EBCDIC_932_5026— ASCII 日文字碼頁，整理；CCSID 037 (美國英文版)。
- EBCDIC_932_5035— ASCII 日文字碼頁，整理；國際字碼編碼 500
- EBCDIC_1252_037— 視窗拉丁字碼頁，整理；CCSID 037 (美國英語)。
- EBCDIC_1252_500— 視窗拉丁文字碼頁，定序；國際化驗證碼 500 (國際版)。
- IDENTITY— 預設定序。字符串進行比較字節的字節。
- IDENTITY_16BIT— UTF-16 的相容性編碼配置：8 位元 (CESU-8) 定序順序。如需詳細資訊，請參閱[統一協會網站上的 \[Unicode 技術報告 #26\]](#)。
- NLSCHAR— 僅適用於泰文字碼頁 (CP874)。
- SYSTEM— 如果您使用SYSTEM，資料庫會自動針對database_codeset和database_territory使用定序順序。

預設值為 IDENTITY。

此外，RDS for Db2 支援下列定序群組：language-aware-collation和 locale-sensitive-collation如需詳細資訊，請參閱IBM Db2文件中的[為 Unicode 資料庫選擇定序](#)。

#####

命AUTOCONFIGURE令語法，例如，'AUTOCONFIGURE APPLY DB'。資料類型為varchar。預設值為空字符串或 null。

如需詳細資訊，請參閱IBM Db2文件中的[AUTOCONFIGURE指令](#)。

使用須知

rdsadmin.create_database如果您在建立 RDS for Db2 資料庫執行個體時未使用 Amazon RDS 主控台或指定資料庫名稱，則可以透過呼叫來建立資料庫。AWS CLI如需詳細資訊，請參閱[建立資料庫執行個體](#)。

特殊考量：

- 傳送至 Db2 執行個體的CREATE DATABASE指令會使用RESTRICTIVE此選項。
- 適用於 Db2 的 RDS 僅AUTOMATIC STORAGE使用。
- 適用於 Db2 的 RDS 會使用NUMSEGS和DFT_EXTENT_SZ的預設值。
- RDS for Db2 使用儲存區加密，不支援資料庫加密。

如需有關這些考量的詳細資訊，請參閱IBM Db2文件中的[CREATE DATABASE指令](#)。

在調用之前 `rdsadmin.create_database`，您必須連接到 `rdsadmin` 數據庫。在下列範例中，將使用您的 RDS for Db2 資料庫執行個體資訊取代 `#####`：

```
db2 connect to rdsadmin user master_username using master_password
```

如需檢查建立資料庫狀態的資訊，請參閱 [rdsadmin.get_task_status](#)。

範例

```
#####TESTJP#####
```

```
db2 "call rdsadmin.create_database('TESTJP', 4096, 'IBM-437', 'JP', 'SYSTEM')"
```

rdsadmin.drop_database

捨棄資料庫。

語法

```
db2 "call rdsadmin.drop_database('database_name')"
```

參數

以下是必要參數：

database_name

要刪除的資料庫名稱。資料類型為 `varchar`。

使用須知

只有在符合下列條件時，`rdsadmin.drop_database` 才能呼叫來刪除資料庫：

- 當您使用 Amazon RDS 主控台或建立 RDS for Db2 資料庫執行個體時，並未指定資料庫的 AWS CLI 名稱。如需詳細資訊，請參閱 [建立資料庫執行個體](#)。
- 您可以呼叫 [the section called “rdsadmin.create_database”](#) 預存程序來建立資料庫。
- 您可以呼叫 [the section called “rdsadmin.restore_database”](#) 預存程序，從離線或備份的映像還原資料庫。

在調用之前 `rdsadmin.drop_database`，您必須連接到 `rdsadmin` 數據庫。在下列範例中，將使用您的 RDS for Db2 資料庫執行個體資訊取代 `#####`：

```
db2 connect to rdsadmin user master_username using master_password
```

如需有關檢查卸除資料庫狀態的資訊，請參閱 [rdsadmin.get_task_status](#)。

範例

下面的例子刪除一個名為的數據庫 `TESTDB`：

```
db2 "call rdsadmin.drop_database('TESTDB')"
```

回應範例

如果您傳遞不正確的資料庫名稱，則預存程序會傳回下列回應範例：

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot drop database. Database with provided name does not exist". SQLSTATE=99993
```

如果您使用 Amazon RDS 主控台或建立資料庫 AWS CLI，則預存程序會傳回下列回應範例：

```
Return Status = 0
```

接收後 `Return Status = 0`，調用存 [the section called "rdsadmin.get_task_status"](#) 儲過程。類似下列範例的回應會說明狀態：

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
  2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

rdsadmin.update_db_param

更新資料庫參數。

語法

```
db2 "call rdsadmin.update_db_param(
  'database_name',
```

```
'parameter_to_modify',  
'changed_value)'"
```

參數

下列是必要參數：

database_name

要執行工作的資料庫名稱。資料類型為varchar。

_ # _

要修改的參數名稱。資料類型為varchar。如需詳細資訊，請參閱 [適用於 Db2 的 Amazon RDS 參數](#)。

()

要將參數值變更為的值。資料類型為varchar。

使用須知

如需有關檢查更新資料庫參數狀態的資訊，請參閱 [rdsadmin.get_task_status](#)。

範例

下列範例會將名為之資料庫100的archretrydelay參數更新為TESTDB：

```
db2 "call rdsadmin.update_db_param(  
    'TESTDB',  
    'archretrydelay',  
    '100')"
```

下列範例會延遲在呼叫的資料庫上建立物件的驗證，以TESTDB避免相依性檢查：

```
db2 "call rdsadmin.update_db_param(  
    'TESTDB',  
    'auto_reval',  
    'deferred_force')"
```

rdsadmin.set_configuration

設定資料庫的特定設定。

語法

```
db2 "call rdsadmin.set_configuration(  
    'name',  
    'value')"
```

參數

下列是必要參數：

name

組態設定的名稱。資料類型為varchar。

#

組態設定的值。資料類型為varchar。

使用須知

下表顯示您可以使用控制的組態設定rdsadmin.set_configuration。

名稱	描述
RESTORE_DATABASE_NUM_BUFFERS	還原作業期間要建立的緩衝區數目。此值必須小於資料庫執行個體類別的總記憶體大小。如果未設定此設定，Db2 會決定還原作業期間要使用的值。如需詳細資訊，請參閱 IBM Db2 文件 。
RESTORE_DATABASE_PARALLELISM	還原作業期間要建立的緩衝區操控點數目。此值必須小於資料庫執行個體 vCPUs 數目的兩倍。如果未設定此設定，Db2 會決定還原作業期間要使用的值。如需詳細資訊，請參閱 IBM Db2 文件 。

範例

下列範例會將組RESTORE_DATABASE_PARALLELISM態設定為8。

```
db2 "call rdsadmin.set_configuration(  
    'RESTORE_DATABASE_PARALLELISM',
```

```
'8')"
```

下列範例會將組RESTORE_DATABASE_NUM_BUFFERS態設定為150。

```
db2 "call rdsadmin.set_configuration(  
    'RESTORE_DATABASE_NUM_BUFFERS',  
    '150')"
```

rdsadmin.show_configuration

傳回您可以使用預存程序設定的目前設定rdsadmin.set_configuration。

語法

```
db2 "call rdsadmin.show_configuration(  
    'name')"
```

參數

以下參數為選用：

name

要傳回相關資訊的組態設定名稱。資料類型為varchar。

下列組態名稱有效：

- 還原作業期間要建立的緩衝區數目。
- 還原作業 — 還原作業期間要建立的緩衝區操控點數目。

使用須知

如果您未指定組態設定的名稱，會rdsadmin.show_configuration傳回您可以使用預存程序設定之所有組態設定的資訊rdsadmin.set_configuration。

範例

下列範例會傳回目前RESTORE_DATABASE_PARALLELISM組態的相關資訊。

```
db2 "call rdsadmin.show_configuration(  
    'RESTORE_DATABASE_PARALLELISM')
```

```
'RESTORE_DATABASE_PARALLELISM')"
```

rdsadmin.restore_database

還原資料庫。

語法

```
db2 "call rdsadmin.restore_database(  
    ?,  
    'database_name',  
    's3_bucket_name',  
    's3_prefix',  
    restore_timestamp,  
    'backup_type')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

database_name

要還原的資料庫名稱。此名稱必須與備份影像中的資料庫名稱相符。資料類型為varchar。

s3_ #####

備份所在之 Amazon S3 儲存貯體的名稱。資料類型為varchar。

s3_ ##

在下載期間用於檔案比對的前置詞。資料類型為varchar。

如果此參數為空，則會下載 Amazon S3 儲存貯體中的所有檔案。以下是範例前置詞：

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```


#####

資料庫備份映像檔的時間戳記。資料類型為varchar。

時間戳記包含在備份檔案名稱中。例如，20230615010101是檔案名稱的時間戳記SAMPLE.0.rdsdb.DBPART000.20230615010101.001。

###

備份的類型。資料類型為varchar。有效值：OFFLINE、ONLINE。

用ONLINE於近乎零的停機時間遷移。如需詳細資訊，請參閱 [Linux基於 Db2 資料庫的近乎零停機時間遷移](#)。

使用須知

`rdsadmin.restore_database`如果您在建立 RDS for Db2 資料庫執行個體時未使用 Amazon RDS 主控台或指定資料庫名稱，則可以呼叫來還原資料庫。AWS CLI如需詳細資訊，請參閱 [建立資料庫執行個體](#)。

還原資料庫之前，您必須為 RDS for Db2 資料庫執行個體佈建儲存空間，該儲存空間等於或大於備份大小和磁碟上原始 Db2 資料庫的總和。當您還原備份時，Amazon RDS 會在您的 RDS for Db2 資料庫執行個體上擷取備份檔案。

每個備份文件必須為 5 TB 或更小。如果備份檔案超過 5 TB，您必須將備份檔案分割為較小的檔案。

若要使用`rdsadmin.restore_database`預存程序還原所有檔案，請勿在檔案名稱中的時間戳記後加入檔案編號尾碼。例如，`s3_prefix` 會`backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101`還原下列檔案：

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.001
SAMPLE.0.rdsdb.DBPART000.20230615010101.002
SAMPLE.0.rdsdb.DBPART000.20230615010101.003
SAMPLE.0.rdsdb.DBPART000.20230615010101.004
SAMPLE.0.rdsdb.DBPART000.20230615010101.005
```

若要改善資料庫還原作業的效能，您可以設定 RDS 使用的緩衝區和緩衝區操控點數目。若要檢查目前的組態，請使用[the section called “rdsadmin.show_configuration”](#)。若要變更組態，請使用[the section called “rdsadmin.set_configuration”](#)。

如需檢查還原資料庫狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

若要在還原資料庫之後使資料庫上線並套用其他交易記錄，請參閱[rdsadmin.rollforward_database](#)。

範例

下列範例會使用單一檔案或多個具有 `s3 backupfolder/`

`SAMPLE.0.rdsdb.DBPART000.20230615010101_prefix` 的檔案來還原離線備份：

```
db2 "call rdsadmin.restore_database(  
?,  
'SAMPLE',  
'DOC-EXAMPLE-BUCKET',  
'backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101',  
20230615010101,  
'OFFLINE')"
```

rdsadmin.rollforward_database

透過呼叫還原資料庫之後，將資料庫上線並套用其他交易記錄檔[rdsadmin.restore_database](#)。

語法

```
db2 "call rdsadmin.rollforward_database(  
?,  
'database_name',  
's3_bucket_name',  
s3_prefix,  
'rollforward_to_option',  
'complete_rollforward')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

database_name

執行作業的資料庫名稱。資料類型為varchar。

`s3_ #####`

備份所在之 Amazon S3 儲存貯體的名稱。資料類型為 `varchar`。

`s3_ ##`

在下載期間用於檔案比對的前置詞。資料類型為 `varchar`。

如果此參數為空，則會下載 S3 儲存貯體中的所有檔案。下列範例為前置字元範例：

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

以下是可選的輸入參數：

`##### _ ##`

您要向前捲動的點。資料類型為 `varchar`。有效值： `END_OF_LOGS`、 `END_OF_BACKUP`。預設值為 `END OF LOGS`。

`## (_R) #####`

指定是否完成向前捲動程序。資料類型為 `varchar`。預設值為 `TRUE`。

如果 `TRUE`，那麼在完成後，數據庫是在線和可訪問的。如果 `FALSE`，則資料庫會保持 `ROLL-FORWARD PENDING` 狀態。

使用須知

通話後 [rdsadmin.restore_database](#)，您必須呼叫 `rollforward_database` 以從 S3 儲存貯體套用存檔日誌。您也可以使用這個預存程序，在呼叫後還原其他交易記錄檔 `rdsadmin.restore_database`。

如果設定 `complete_rollforward` 為 `FALSE`，則資料庫 `ROLL-FORWARD PENDING` 處於離線狀態。要使數據庫上線，您必須打電話 [rdsadmin.complete_rollforward](#)。

如需檢查向前捲動資料庫狀態的資訊，請參閱 [rdsadmin.get_task_status](#)。

範例

下列範例會向前復原至具有交易記錄檔的資料庫線上備份，然後讓資料庫上線：

```
db2 "call rdsadmin.rollforward_database(  
    ?,  
    null,
```

```
    null,  
    'END_OF_LOGS',  
    'TRUE')"
```

下列範例會向前復原至沒有交易記錄檔的資料庫線上備份，然後讓資料庫上線：

```
db2 "call rdsadmin.rollforward_database(  
    ?,  
    'TESTDB',  
    'DOC-EXAMPLE-BUCKET',  
    'logsfolder/  
    'END_OF_BACKUP',  
    'TRUE')"
```

下列範例會向前彙整至具有交易記錄檔的資料庫線上備份，然後不會讓資料庫上線：

```
db2 "call rdsadmin.rollforward_database(  
    ?,  
    'TESTDB',  
    null,  
    'onlinebackup/TESTDB',  
    'END_OF_LOGS',  
    'FALSE')"
```

下列範例會向前彙整至含有其他交易記錄檔的資料庫線上備份，然後不會讓資料庫上線：

```
db2 "call rdsadmin.rollforward_database(  
    ?,  
    'TESTDB',  
    'DOC-EXAMPLE-BUCKET',  
    'logsfolder/S0000155.LOG',  
    'END_OF_LOGS',  
    'FALSE')"
```

rdsadmin.complete_rollforward

使資料庫從ROLL-FORWARD PENDING狀態上線。

語法

```
db2 "call rdsadmin.complete_rollforward(  
    ?,  
    'TESTDB',  
    'DOC-EXAMPLE-BUCKET',  
    'logsfolder/S0000155.LOG',  
    'END_OF_LOGS',  
    'FALSE')"
```

```
?,  
'database_name')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

database_name

您要上線的資料庫名稱。資料類型為varchar。

使用須知

如果您[rdsadmin.rollforward_database](#)使用 `complete_rollforward set to` 呼叫FALSE，則您的資料庫ROLL-FORWARD PENDING處於離線狀態。若要完成向前捲動程序並使資料庫上線，請呼叫。`rdsadmin.complete_rollforward`

如需檢查完成向前捲動程序狀態的資訊，請參閱。[rdsadmin.get_task_status](#)

範例

下列範例會使資TESTDB料庫上線：

```
db2 "call rdsadmin.complete_rollfoward(  
?,  
'TESTDB')"
```

rdsadmin.db2pd_command

收集 Db2 資料庫之 RDS 的相關資訊。

語法

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

參數

需要下列輸入參數：

您要執行的db2pd命令名稱。資料類型為varchar。

參數必須以連字號開頭。如需參數清單，請參閱 IBM 說明文件中的 [db2pd-監控和疑難排解 Db2 資料庫命令](#)。

無法使用下列參數：

- -rep | -repeat
- -fil | -file
- -db | -data | -database <dbname> 沒有任何子選項，例如或 -apinfo -logs
- -inst | -instance

使用須知

此預存程序會收集可協助監視和疑難排解 RDS for Db2 資料庫的資訊。

預存程序會使用IBMdb2pd公用程式來執行各種命令。此db2pd公用程式需要SYSADM授權，Db2 主要使用者沒有授權。不過，使用 Amazon RDS 存放程序時，主要使用者可以使用公用程式執行各種命令。如需有關公用程式的詳細資訊，請參閱 IBM 說明文件中的 [db2pd-監控和疑難排解 Db2 資料庫命令](#)。

輸出限制為最大 2 MB。

如需檢查收集資料庫相關資訊之狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會傳回適用於 Db2 資料庫執行個體的 RDS 執行時間：

```
db2 "call rdsadmin.db2pd_command('-')
```

下列範例會傳回名為的資料庫的執行時間TESTDB：

```
db2 "call rdsadmin.db2pd_command('-db TESTDB -')
```

下列範例會傳回 RDS for Db2 資料庫執行個體的記憶體使用量：

```
db2 "call rdsadmin.db2pd_command('-dbptnmem')"
```

下列範例會傳回 RDS for Db2 資料庫執行個體和名為TESTDB的資料庫的記憶體集：

```
db2 "call rdsadmin.db2pd_command('-inst -db TESTDB -memsets')"
```

rdsadmin.force_application

強制使用 RDS for Db2 資料庫的應用程式。

語法

```
db2 "call rdsadmin.force_application(  
    ?,  
    'applications')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

####

您要強制關閉 RDS for Db2 資料庫的應用程式。資料類型為varchar。有效值：ALL或#####

使用逗號分隔多個應用程式的名稱。###'##### _1##### _ ## _2'##

使用須知

這個預存程序會強制所有應用程式離開資料庫，以便您可以執行維護。

預存程序會使用IBMFORCE APPLICATION命令。此命FORCE APPLICATION令需要SYSADMSYSMAINT、或SYSCTRL授權，Db2 主要使用者所沒有的 RDS。不過，使用 Amazon RDS 存放程序時，主要使用者可以使用該命令。如需詳細資訊，請參閱 IBM 說明文件中的[強制應用程式命令](#)。

如需檢查強制執行應用程式離開資料庫之狀態的資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會強制所有應用程式離開 RDS for Db2 資料庫：

```
db2 "call rdsadmin.force_application(  
    ?,  
    'ALL')"
```

下列範例會強制應用程式處理9991和1192關閉 RDS for Db2 資料庫：8891

```
db2 "call rdsadmin.force_application(  
    ?,  
    '9991, 8891, 1192')"
```

rdsadmin.set_archive_log_retention

針對指定的 RDS for Db2 資料庫設定保留封存記錄檔的時間 (以小時為單位)。

語法

```
db2 "call rdsadmin.set_archive_log_retention(  
    ?,  
    'database_name',  
    'archive_log_retention_hours')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

database_name

要設定存檔日誌保留的資料庫名稱。資料類型為varchar。

#####

保留存檔記錄檔的小時數。資料類型為smallint。預設值為0，且最大值為 168 (7 天)。

如果值為0，則 Amazon RDS 不會保留存檔日誌檔。

使用須知

您可以透過呼叫來檢視目前的封存記錄保留設定[the section called “rdsadmin.show_archive_log_retention”](#)。

您無法在rdsadmin資料庫上設定封存記錄保留設定。

範例

下列範例會將資料庫的存檔記錄保留時間設定TESTDB為 24 小時。

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '24')"
```

下列範例會停用名為的資料庫的封存記錄保留TESTDB。

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '0')"
```

rdsadmin.show_archive_log_retention

傳回指定資料庫的目前存檔記錄保留設定。

語法

```
db2 "call rdsadmin.show_archive_log_retention(
    ?,
    'database_name')"
```

參數

需要以下輸出參數：

?

輸出錯誤訊息的參數標識。此參數僅接受?。

需要下列輸入參數：

database_name

顯示存檔日誌保留設定的資料庫名稱。資料類型為varchar。

範例

下列範例顯示名為的資料庫的存檔記錄保留設定TESTDB。

```
db2 "call rdsadmin.show_archive_log_retention(  
    ?  
    'TESTDB')"
```

管理表格空間

下列預存程序會管理 Amazon RDS for Db2 資料庫的表格空間。若要執行這些程序，master 使用者必須先連線到資料庫。

主題

- [rdsadmin.create_tablespace](#)
- [rdsadmin.alter_tablespace](#)
- [rdsadmin.rename_tablespace](#)
- [rdsadmin.drop_tablespace](#)

rdsadmin.create_tablespace

建立表格空間。

語法

```
db2 "call rdsadmin.create_tablespace(  
    'database_name',  
    'tablespace_name',  
    'buffer_pool_name',  
    tablespace_page_size,  
    tablespace_initial_size,  
    tablespace_increase_size,  
    'tablespace_type')"
```

參數

下列是必要參數：

database_name

要在其中建立表格空間的資料庫名稱。資料類型為varchar。

#####

要建立之表格空間的名稱。資料類型為varchar。

表格空間名稱有下列限制：


- 它不能與此數據庫中現有表格空間的名稱相同。

- 它只能包含字符_ \$ # @ a - z A - Z 0 - 9。
- 它不能以_ 或開頭\$。
- 它不能開始SYS。

下列是選用參數：

#####


要指派表格空間的緩衝區集區名稱。資料類型為varchar。預設為空字串。

 Important

您必須擁有相同頁面大小的緩衝區集區，才能與表格空間產生關聯。

#####

表格空間的頁面大小 (以位元組為單位)。資料類型為integer。有效值：4096、8192、16384、32768。預設值是您透過呼叫建立資料庫時所使用的頁面大小 [rdsadmin.create_database](#)。


 Important

Amazon RDS 支援 4 KiB、8 KiB 和 16 KiB 頁面的寫入原子性。相比之下，32 KiB 頁面可能會遭到寫入撕裂，或將部分資料寫入桌面的風險。如果您使用 32 KiB 頁面，建議您啟用 point-in-time 復原和自動備份。否則，您會有無法從撕裂頁面復原的風險。如需詳細資訊，請參閱 [the section called “備份簡介”](#) 及 [the section called “Point-in-time 回收”](#)。

#####

表格空間的初始大小，以 KB 為單位。資料類型為integer。有效值：48或更高。預設值為空。

如果您未設定值，Db2 會為您設定適當的值。

 Note

此參數不適用於暫時表格空間，因為系統會管理暫存表格空間。

_

表格空間已滿時要增加的百分比。資料類型為integer。有效值：1—100。預設值為空。

如果您未設定值，Db2 會為您設定適當的值。

Note

此參數不適用於暫時表格空間，因為系統會管理暫存表格空間。

####

表格空間的類型。資料類型為char。有效值：U（用於使用者資料）或T（用於暫時資料）。預設值為U。

使用須知

適用於 Db2 的 RDS 永遠會為資料建立大型資料庫。

如需檢查建立表格空間之狀態的詳細資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會建立名為的表格空間，SP8並指派呼叫的資料庫BP8的緩衝集區TESTDB。表格空間的初始表格空間頁面大小為 4,096 個位元組，初始表格空間為 1,000 KB，表格大小增加設為 50%。

```
db2 "call rdsadmin.create_tablespace(  
    'TESTDB',  
    'SP8',  
    'BP8',  
    4096,  
    1000,  
    50)"
```

下列範例會建立名為的暫時表格空間SP8。它會為名為BP8的資料庫指定大小為 8 KiB 的緩衝區集區。TESTDB

```
db2 "call rdsadmin.create_tablespace(  
    'TESTDB',  
    'SP8',
```

```
'BP8',  
8192,  
NULL,  
NULL,  
'T')"
```

rdsadmin.alter_tablespace

變更表格空間。

語法

```
db2 "call rdsadmin.alter_tablespace(  
    'database_name',  
    'tablespace_name',  
    'buffer_pool_name',  
    tablespace_increase_size,  
    'max_size',  
    'reduce_max',  
    'reduce_stop',  
    'reduce_value',  
    'lower_high_water',  
    'lower_high_water_stop',  
    'switch_online')"
```

參數

下列是必要參數：

database_name

使用表格空間之資料庫的名稱。資料類型為varchar。

####

要變更之表格空間的名稱。資料類型為varchar。

下列是選用參數：

#####

要指派表格空間的緩衝區集區名稱。資料類型為varchar。預設為空字串。

⚠ Important

您必須擁有相同頁面大小的緩衝區集區，才能與表格空間產生關聯。

_

表格空間已滿時要增加的百分比。資料類型為integer。有效值：1—100。預設值為0。

####

表格空間的大小上限。資料類型為varchar。有效值：## K M |G，或NONE。預設值為NONE。

####

指定是否將高水位標記降低到其最大限制。資料類型為char。預設值為N。

_

指定是否要中斷前一個reduce_max或指reduce_value令。資料類型為char。預設值為N。

###

用來減少表格空間高浮水印的數目或百分比。資料類型為varchar。有效值：## K M |G，或1—100。預設值為N。

###

指定是否執行ALTER TABLESPACE LOWER HIGH WATER MARK命令。資料類型為char。預設值為N。

_

指定是否執行ALTER TABLESPACE LOWER HIGH WATER MARK STOP命令。資料類型為char。預設值為N。

_

指定是否執行ALTER TABLESPACE SWITCH ONLINE命令。資料類型為char。預設值為N。

使用須知**可選參**

數reduce_maxreduce_stop、reduce_value、lower_high_waterlower_high_water_stop、和switch_online是互斥的。您無法將它們與任何其他可選參數結合使用，例

如buffer_pool_name，在rdsadmin.alter_tablespace命令中。如果將這些參數與rdsadmin.alter_tablespace命令中的任何其他可選參數結合在一起rdsadmin.get_task_status，那麼當您運行時，Db2 將返回如下錯誤：

```
DB21034E The command was processed as an SQL statement because it was not a valid
Command Line Processor command. During SQL processing it returned:
SQL1763N Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason
"12"
```

如需檢查變更表格空間之狀態的詳細資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會變更名為的表格空間，SP8並指派一個緩衝集區，呼叫BP8此資料庫呼叫TESTDB以降低高浮水位標記。

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    NULL,
    NULL,
    'Y')"
```

下列範例會在資料庫TBSP_TEST中呼叫的表格空間執行REDUCE MAX命令TESTDB。

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    'Y')"
```

下列範例會在資料庫TBSP_TEST中呼叫的表格空間執行REDUCE STOP命令TESTDB。

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
```



```
NULL,  
NULL,  
'Y')"
```

rdsadmin.rename_tablespace

重新命名表格空間。

語法

```
db2 "call rdsadmin.rename_tablespace(  
?,  
'database_name',  
'source_tablespace_name',  
'target_tablespace_name')"
```

參數

下列是必要參數：

?

輸出錯誤訊息的參數標識。此參數只接受?。

database_name

表格空間所屬的資料庫名稱。資料類型為varchar。

#####

要重新命名之表格空間的名稱。資料類型為varchar。

#####

表格空間的新名稱。資料類型為varchar。

新名稱具有下列限制：

- 它不能與現有表格空間的名稱相同。
- 它只能包含字符_ \$#@a-zA-Z0-9。
- 它不能以_或開頭\$。
- 它不能開始SYS。

使用須知

如需檢查重新命名表格空間之狀態的相關資訊，請參閱[rdsadmin.get_task_status](#)。

您無法重新命名屬於rdsadmin資料庫的表格空間。

範例

下列範例會重新命名資料庫中呼叫SP8SP9的表格空間。TESTDB

```
db2 "call rdsadmin.rename_tablespace(  
    ?,  
    'TESTDB',  
    'SP8'.  
    'SP9')"
```

rdsadmin.drop_tablespace

刪除表格空間。

語法

```
db2 "call rdsadmin.drop_tablespace(  
    'database_name',  
    'tablespace_name')"
```

參數

下列是必要參數：

database_name

表格空間所屬的資料庫名稱。資料類型為varchar。

###

要刪除之表格空間的名稱。資料類型為varchar。

使用須知

如需檢查刪除表格空間之狀態的相關資訊，請參閱[rdsadmin.get_task_status](#)。

範例

下列範例會刪除SP8從名為的資料庫呼叫的表格空間TESTDB。

```
db2 "call rdsadmin.drop_tablespace(  
    'TESTDB',  
    'SP8')"
```

管理稽核策略

下列預存程序會針對使用稽核記錄的 Amazon RDS for Db2 資料庫管理稽核政策。如需詳細資訊，請參閱 [the section called “Db2 稽核記錄”](#)。若要執行這些程序，master 使用者必須先連線到 `rdsadmin` 資料庫。

主題

- [rdsadmin.configure_db_audit](#)
- [rdsadmin.disable_db_audit](#)

rdsadmin.configure_db_audit

RDS for Db2 #####如果您設定的原則不存在，呼叫這個預存程序會建立它。如果這個原則確實存在，呼叫這個預存程序會使用您提供的參數值來修改它。

語法

```
db2 "call rdsadmin.configure_db_audit(  
    'db_name',  
    'category',  
    'category_setting',  
    '?')"
```

參數

以下是必需的參數。

#####

要設定稽核原則之 RDS for Db2 資料庫的資料庫名稱。資料類型為 `varchar`。

##

要設定此稽核策略的類別名稱。資料類型為 `varchar`。以下是此參數的有效值：

- ALL— 與 ALL, Amazon RDS 不包括 `CONTEXTEXECUTE`, 或 `ERROR` 類別。
- AUDIT
- CHECKING
- CONTEXT

- ERROR
- EXECUTE— 您可以使用數據或不使用數據配置此類別。隨著 data 意味著還記錄為任何主機變量和參數標記提供的輸入數據值。預設值為無資料。如需詳細資訊，請參閱 [## _ ##](#) 參數的說明和 [the section called “範例”](#)
- OBJMAINT
- SECMAINT
- SYSADMIN
- VALIDATE

如需有關這些類別的詳細資訊，請參閱 [IBM Db2文件](#)。

_

指定稽核類別的設定。資料類型為varchar。

下表展示了每個品類的有效品類設定值。

類別	有效的類別設定
ALL	BOTH FAILURE SUCCESS NONE
AUDIT	
CHECKING	
CONTEXT	
OBJMAINT	
SECMAINT	
SYSADMIN	
VALIDATE	
ERROR	AUDIT NORMAL 。 預設值為NORMAL。
EXECUTE	BOTH, WITH BOTH, WITHOUT FAILURE, WITH FAILURE, WITHOUT SUCCESS, WITH SUCCESS, WITHOUT NONE

使用須知

在呼叫之前 `rdsadmin.configure_db_audit`，請確定 RDS for Db2 資料庫執行個體與您設定稽核原則的資料庫已與具有選項的選項群組相關聯。DB2_AUDIT 如需詳細資訊，請參閱 [the section called “設定 Db2 稽核記錄”](#)。

設定稽核策略之後，您可以依照中的步驟檢查資料庫稽核配置的狀態 [檢查稽核配置](#)。

ALL 為 `category` 參數指定不包括 `CONTEXTEXECUTE`、或 `ERROR` 類別。若要將這些類別新增至稽核策略，請 `rdsadmin.configure_db_audit` 分別呼叫您要新增的每個類別。如需詳細資訊，請參閱 [the section called “範例”](#)。

範例

下列範例會建立或修改名為的資料庫的稽核策略 `TESTDB`。在範例 1 到 5 中，如果先前未設定 `ERROR` 類別，則此類別會設定為 `NORMAL` (預設值)。若要將該設定變更為 `AUDIT`，請遵循 [Example 6: Specifying the ERROR category](#)。

範例 1：指定 ALL 品類

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ALL', 'BOTH', ?)"
```

在此範例中，呼叫會設定稽核策略中的 `AUDITCHECKINGOBJMAINTSECMAINTSYSADMIN`、`VALIDATE` 類別。指定 `BOTH` 表示每個類別都會稽核成功和失敗的事件。

範例 2：使用資料指定 EXECUTE 類別

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'SUCCESS,WITH', ?)"
```

在此範例中，呼叫會設定稽核策略中的 `EXECUTE` 類別。指定 `SUCCESS,WITH` 表示此類別的記錄只會包含成功的事件，而且會包含為主機變數和參數標記提供的輸入資料值。

範例 3：指定不含資料的 EXECUTE 類別

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'FAILURE,WITHOUT', ?)"
```

在此範例中，呼叫會設定稽核策略中的 `EXECUTE` 類別。指定 `FAILURE,WITHOUT` 表示此類別的記錄只會包含失敗事件，而且不會包含為主機變數和參數標記提供的輸入資料值。

範例 4：指定沒有狀態事件的 EXECUTE 類別

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'NONE', ?)"
```

在此範例中，呼叫會設定稽核策略中的EXECUTE類別。指定NONE表示不會稽核此類別中的任何事件。

範例 5：指定OBJMAINT品類

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'OBJMAINT', 'NONE', ?)"
```

在此範例中，呼叫會設定稽核策略中的OBJMAINT類別。指定NONE表示不會稽核此類別中的任何事件。

範例 6：指定ERROR品類

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ERROR', 'AUDIT', ?)"
```

在此範例中，呼叫會設定稽核策略中的ERROR類別。指定AUDIT表示所有錯誤 (包括稽核記錄本身發生的錯誤) 都會擷取到記錄檔中。預設錯誤類型為NORMAL。使用時NORMAL，會忽略稽核產生的錯誤，並且只會擷取與正在執行之作業相關聯的錯誤。SQLCODE

rdsadmin.disable_db_audit

停止 db *_name* 指定之 RDS for Db2 資料庫的稽核記錄，並移除為其設定的稽核原則。

Note

此預存程序只會移除透過呼叫所設定的稽核策略 [the section called "rdsadmin.configure_db_audit"](#)。

語法

```
db2 "call rdsadmin.disable_db_audit('db_name')"
```

參數

以下是必需的參數。

#####

要停用稽核記錄的 RDS for Db2 資料庫的資料庫名稱。資料類型為varchar。

使用須知

呼叫 `rdsadmin.disable_db_audit` 不會停用適用於 Db2 資料庫執行個體 RDS 的稽核記錄。若要停用資料庫執行個體層級的稽核記錄，請從資料庫執行個體移除選項群組。如需詳細資訊，請參閱 [停用 Db2 稽核記錄](#)。

範例

下列範例會停用名為的資料庫的稽核記錄TESTDB。

```
db2 "call rdsadmin.disable_db_audit('TESTDB')"
```


用於 Db2 的 Amazon RDS 用戶定義函數參考

這些主題說明執行 Db2 引擎的 Amazon RDS 資料庫執行個體可用的使用者定義函數。

主題

- [检查工作狀態](#)

檢查工作狀態

您可以使用使用 `rdsadmin.get_task_status` 者定義的函數來檢查適用於 Db2 的 Amazon RDS 的下列任務的狀態。此清單並不詳盡。

- 建立、變更或刪除緩衝集區
- 建立、變更或刪除表格空間
- 創建或刪除數據庫
- 從 Amazon S3 還原資料庫備份
- 從 Amazon S3 向前滾動資料庫日誌

`rdsadmin.get_task_status`

傳回工作的狀態。

語法

```
db2 "select task_id, task_type, database_name, lifecycle,
      varchar(bson_to_json(task_input_params), 500) as task_params,
      cast(task_output as varchar(500)) as task_output
      from table(rdsadmin.get_task_status(task_id, 'database_name', 'task_type'))"
```

參數

下列參數是選擇性的。如果您未提供任何參數，使用者定義函數會傳回所有資料庫之所有工作的狀態。Amazon RDS 會保留任務歷史記錄 35 天。

#####

正在執行之工作的識別碼。當您執行工作時，會傳回此 ID。預設：0。

database_name

正在執行工作的資料庫名稱。

####

要查詢的工作類型。有效

值：ADD_GROUPS、ADD_USER、ALTER_BUFFERPOOL、ALTER_TABLESPACE、CHANGE_PASSWOR

範例

下列範例會顯示呼叫時傳回 `rdsadmin.get_task_status` 的資料行。

```
db2 "describe select * from table(rdsadmin.get_task_status())"
```

下列範例會列出所有工作的狀態。

```
db2 "select task_id, task_type, database_name, lifecycle,  
       varchar(bson_to_json(task_input_params), 500) as task_params,  
       cast(task_output as varchar(500)) as task_output  
from table(rdsadmin.get_task_status(null,null,null))"
```

下列範例會列出特定工作的狀態。

```
db2 "select task_id, task_type, database_name,  
       varchar(bson_to_json(task_input_params), 500) as task_params  
from table(rdsadmin.get_task_status(1,null,null))"
```

下列範例會列出特定工作和資料庫的狀態。

```
db2 "select task_id, task_type, database_name,  
       varchar(bson_to_json(task_input_params), 500) as task_params  
from table(rdsadmin.get_task_status(2,'SAMPLE',null))"
```

下列範例會列出所有 ADD_GROUPS 工作的狀態。

```
db2 "select task_id, task_type, database_name,  
       varchar(bson_to_json(task_input_params), 500) as task_params  
from table(rdsadmin.get_task_status(null,null,'add_groups'))"
```

下列範例會列出特定資料庫之所有工作的狀態。

```
db2 "select task_id, task_type, database_name,  
       varchar(bson_to_json(task_input_params), 500) as task_params  
from table(rdsadmin.get_task_status(null,'testdb', null))"
```

下列範例會將 JSON 值輸出為資料行。

```
db2 "select varchar(r.task_type,25) as task_type, varchar(r.lifecycle,10) as lifecycle,
r.created_at, u.* from
    table(rdsadmin.get_task_status(null,null,'restore_db')) as r,
    json_table(r.task_input_params, 'strict $' columns(s3_prefix varchar(500)
        null on empty, s3_bucket_name varchar(500) null on empty) error on error ) as U"
```

回應

`rdsadmin.get_task_status` 使用者定義函數會傳回下列資料行：

TASK_ID

任務的 ID。

TASK_TYPE

取決於輸入參數。

- ADD_GROUPS— 新增群組。
- ADD_USER— 新增使用者。
- ALTER_BUFFERPOOL— 改變緩衝集區。
- ALTER_TABLESPACE— 變更表格空間。
- CHANGE_PASSWORD — 變更使用者的密碼。
- COMPLETE_ROLLFORWARD— 完成 `rdsadmin.rollforward_database` 工作並啟動資料庫。
- CREATE_BUFFERPOOL— 建立緩衝集區。
- CREATE_DATABASE— 建立資料庫。
- CREATE_ROLE— 為使用者建立 Db2 角色。
- CREATE_TABLESPACE— 建立表格空間。
- DROP_BUFFERPOOL— 卸除緩衝集區。
- DROP_DATABASE— 刪除資料庫。
- DROP_TABLESPACE— 卸除表格空間。
- LIST_USERS— 列出所有使用者。
- REMOVE_GROUPS— 移除群組。
- REMOVE_USER— 移除使用者。
- RESTORE_DB— 還原完整資料庫。
- ROLLFORWARD_DB_LOG— 對資料庫記錄執行 `rdsadmin.rollforward_database` 工作。

- ROLLFORWARD_STATUS — 傳回rdsadmin.rollforward_database任務的狀態。
- UPDATE_DB_PARAM— 更新資料參數。

DATABASE_NAME

與工作相關聯的資料庫名稱。

COMPLETED_WORK_BYTES

工作還原的位元組數目。

DURATION_MINS

完成任務所花費的時間。

LIFECYCLE

任務的狀態。可能的狀態：

- CREATED— 將任務提交到 Amazon RDS 後，Amazon RDS 將狀態設置為CREATED。
- IN_PROGRESS— 任務開始後，Amazon RDS 會將狀態設定為IN_PROGRESS。狀態可能需要 5 分鐘的時間才會從變更CREATED為IN_PROGRESS。
- SUCCESS— 任務完成後，Amazon RDS 會將狀態設定為SUCCESS。
- ERROR— 如果還原任務失敗，Amazon RDS 會將狀態設定為ERROR。若要取得有關錯誤的更多資訊，請參閱TASK_OUTPUT。

CREATED_BY

創authid建命令的。

CREATED_AT

建立任務的日期和時間。

LAST_UPDATED_AT

上次更新工作的資料和時間。

TASK_INPUT_PARAMS

參數會根據工作類型而有所不同。所有輸入參數都表示為 JSON 物件。例如，工RESTORE_DB作的 JSON 金鑰如下：

- DBNAME
- RESTORE_TIMESTAMP
- S3_BUCKET_NAME

- S3_PREFIX

TASK_OUTPUT

任務的其他資訊。如果在原生還原期間發生錯誤，此欄會包含錯誤的相關資訊。

回應範例

下列回應範例顯示已順利建立呼TESTJP叫的資料庫。如需詳細資訊，請參閱[the section called “rdsadmin.create_database”](#)預存程序。

```
`1 SUCCESS CREATE_DATABASE RDSDB 2023-10-24-18.32.44.962689 2023-10-24-18.34.50.038523
1 TESTJP { "CODESET" : "IBM-437", "TERRITORY" : "JP", "COLLATION" : "SYSTEM",
"AUTOCONFIGURE_CMD" : "", "PAGESIZE" : 4096 }
2023-10-24-18.33.30.079048 Task execution has started.

2023-10-24-18.34.50.038523 Task execution has completed successfully`.
```

下列回應範例說明卸除資料庫失敗的原因。如需詳細資訊，請參閱[the section called “rdsadmin.drop_database”](#)預存程序。

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

下列回應範例顯示成功還原資料庫。如需詳細資訊，請參閱[the section called “rdsadmin.restore_database”](#)預存程序。

```
1 RESTORE_DB SAMPLE SUCCESS

{ "S3_BUCKET_NAME" : "DOC-EXAMPLE-BUCKET", "S3_PREFIX" :
"SAMPLE.0.rdsdb3.DBPART000.20230413183211.001", "RESTORE_TIMESTAMP" :
"20230413183211", "BACKUP_TYPE" : "offline" }
```

```
2023-11-06-18.31.03.115795 Task execution has started.  
2023-11-06-18.31.04.300231 Preparing to download  
2023-11-06-18.31.08.368827 Download complete. Starting Restore  
2023-11-06-18.33.13.891356 Task Completed Successfully
```

Amazon RDS for MariaDB

Amazon RDS 會支援執行下列版本 MariaDB 的資料庫執行個體：

- MariaDB 10.11
- MariaDB 10.6
- MariaDB 10.5
- MariaDB 10.4
- MariaDB 10.3 (RDS 標準支援結束日期排定於 2023 年 10 月 23 日)

如需次要版本的支援詳情，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

若要建立 MariaDB 資料庫執行個體，請使用 Amazon RDS 管理工具或界面。然後，您可以使用 Amazon RDS 工具對資料庫執行個體執行管理動作。這些包括如下的動作：

- 重新設定或調整資料庫執行個體的大小
- 授權連線到資料庫執行個體
- 從備份或快照建立與還原
- 建立多可用區域次要項
- 建立僅供讀取複本
- 監控資料庫執行個體的效能

若要在資料庫執行個體中儲存與存取資料，請使用標準 MariaDB 公用程式與應用程式。

所有的 AWS 區域中都有 MariaDB。如需有關 AWS 區域的詳細資訊，請參閱[區域、可用區域和 Local Zones](#)。

您可以使用 Amazon RDS for MariaDB 資料庫來建置符合 HIPAA 規範的應用程式。依據與 AWS 簽署的商業夥伴協議 (BAA)，您可以存放醫療保健相關的資訊，包括受保護醫療資訊 (PHI)。如需更多詳細資訊，請參閱[HIPAA 合規](#)。AWS 服務範圍經過第三方稽核機構完整評估，已取得認證、合規證書或操作授權書 (ATO)。如需詳細資訊，請參閱[合規計劃的 AWS 服務範圍](#)。

建立資料庫執行個體之前，請先完成[設定 Amazon RDS](#)中的步驟。建立資料庫執行個體時，RDS 主要使用者帳戶會取得 DBA 權限 (有一些限制)。將此帳戶用於管理任務，例如建立其他資料庫帳戶。

您可以建立下列作業：

- [資料庫執行個體](#)
- [資料庫快照](#)
- [時間點還原](#)
- [自動備份](#)
- [手動備份](#)

您可以在以 Amazon 虛擬私有雲端 (VPC) 為基礎的 VPC 內使用執行 MariaDB 的資料庫執行個體。您也可以透過啟用各種選項，新增功能至 MariaDB 資料庫執行個體。Amazon RDS 支援 MariaDB 的多個可用區域部署，做為高可用性、容錯移轉解決方案。

Important

為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。而且會將存取權限制在某些需要進階權限的系統程序和資料表。您可以使用標準 SQL 用戶端 (例如 mysql 用戶端) 存取資料庫。不過，您無法使用 Telnet 或 Secure Shell (SSH) 直接存取主機。

主題

- [Amazon RDS 上的 MariaDB 功能支援](#)
- [Amazon RDS 上的 MariaDB 版本](#)
- [連接至執行 MariaDB 資料庫引擎的資料庫執行個體](#)
- [保護 MariaDB 執行個體連線安全](#)
- [使用 Amazon RDS Optimized Reads 改善 RDS for MariaDB 的查詢效能](#)
- [使用 Amazon RDS Optimized Writes 改善 MariaDB 的寫入效能](#)
- [升級 MariaDB 資料庫引擎](#)
- [將資料匯入 MariaDB 資料庫執行個體](#)
- [使用 Amazon RDS 中的 MariaDB 複寫](#)
- [MariaDB 資料庫引擎的選項](#)
- [MariaDB 的參數](#)
- [將資料從 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體](#)
- [Amazon RDS 上的 MariaDB SQL 參考](#)

- [MariaDB 資料庫執行個體的本機時區](#)
- [RDS for MariaDB 的已知問題和限制](#)

Amazon RDS 上的 MariaDB 功能支援

RDS for MariaDB 支援 MariaDB 的大多數特性與功能。部分功能的支援或權限可能有所限制。

您可以在[資料庫的最新資訊](#)頁面上篩選新的 Amazon RDS 功能。對於 Products (產品)，請選擇 Amazon RDS。然後使用關鍵字搜尋，例如 **MariaDB 2023**。

Note

以下清單並非詳盡清單。

主題

- [Amazon RDS for MariaDB 主要版本的 MariaDB 功能支援](#)
- [Amazon RDS 上的 MariaDB 支援的儲存引擎](#)
- [Amazon RDS 上 MariaDB 的快取預備](#)
- [Amazon RDS 不支援的 MariaDB 功能](#)

Amazon RDS for MariaDB 主要版本的 MariaDB 功能支援

在下面的章節中，參閱 Amazon RDS for MariaDB 主要版本的 MariaDB 功能支援相關資訊：

主題

- [Amazon RDS 上的 MariaDB 10.11 支援](#)
- [Amazon RDS 上的 MariaDB 10.6 支援](#)
- [Amazon RDS 上的 MariaDB 10.5 支援](#)
- [Amazon RDS 上的 MariaDB 10.4 支援](#)
- [Amazon RDS 上的 MariaDB 10.3 支援](#)

如需 Amazon RDS for MariaDB 支援次要版本的相關資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

Amazon RDS 上的 MariaDB 10.11 支援

對於執行 MariaDB 10.11 版或更高版本的資料庫執行個體，Amazon RDS 支援下列新功能：

- 密碼重複使用檢查外掛程式 - 您可以使用 MariaDB 密碼重複使用檢查外掛程式來防止使用者重複使用密碼，以及設定密碼的保留期。如需詳細資訊，請參閱[密碼重複使用檢查外掛程式](#)。
- GRANT TO PUBLIC 授權 - 您可以對能夠存取您的伺服器的所有使用者授予權限。如需詳細資訊，請參閱[GRANT TO PUBLIC](#)。
- 分隔 SUPER 和 READ ONLY ADMIN 權限 - 您可以移除所有使用者的 READ ONLY ADMIN 權限，即使使用者先前擁有 SUPER 權限也一樣。
- 安全性 - 您現在可以將選項 `--ssl` 設定為 MariaDB 用戶端的預設值。MariaDB 不再於組態不正確的情況下，無訊息停用 SSL。
- SQL 命令和函數 - 您現在可以使用 `SHOW ANALYZE FORMAT=JSON` 命令以及 `ROW_NUMBER`、`SFORMAT` 和 `RANDOM_BYTES` 函數。`SFORMAT` 允許設定字串格式，且預設為啟用。您可以使用單一命令進行分割區與資料表的相互轉換。`JSON_*`() 函數也有幾項改進。`DES_ENCRYPT` 和 `DES_DECRYPT` 函數在 10.10 及更高版本中已棄用。如需詳細資訊，請參閱[SFORMAT](#)。
- InnoDB 增強功能 - 這些增強功能包括下列項目：
 - 重做日誌中的效能改進，可減少寫入放大現象並改善並行。
 - 您可以變更復原資料表空間，而不需重新初始化資料目錄。此增強功能可減少控制平面額外負荷。它需要重新啟動，但在變更復原資料表空間後不需重新初始化。
 - 支援 `CHECK TABLE ... EXTENDED` 及在內部遞減排序索引。
 - 大量插入的改進。
- Binlog 變更 - 這些變更包括下列項目：
 - 分兩個階段記錄 `ALTER` 以減少複寫延遲。`binlog_alter_two_phase` 參數預設為停用，但可透過參數群組啟用。
 - 記錄 `explicit_defaults_for_timestamp`。
 - 如果交易可安全地復原，則不再記錄 `INCIDENT_EVENT`。
- 複寫改進 - 如果主要執行個體支援，則 MariaDB 10.11 版資料庫執行個體預設會使用 GTID 複寫。此外，`Seconds_Behind_Master` 更精確。
- 用戶端 - 您可以使用 `mysqlbinlog` 和 `mariadb-dump` 的新命令列選項。您可以使用 `mariadb-dump` 傾印和還原歷史資料。
- 系統版本控制 - 您可以修改歷史記錄。MariaDB 會自動建立新的分割區。

- 原子 DDL - CREATE OR REPLACE 現在為原子。陳述式成功，或是完全失敗。
- 重做日誌寫入 - 重做日誌以非同步方式寫入。
- 預存函數 - 預存函數現在可支援與預存程序中相同的 IN、OUT 及 INOUT 參數。
- 已棄用或移除的參數 - MariaDB 10.11 版資料庫執行個體已棄用或移除下列參數：
 - [innodb_change_buffering](#)
 - [innodb_disallow_writes](#)
 - [innodb_log_write_ahead_size](#)
 - [innodb_prefix_index_cluster_optimization](#)
 - [keep_files_on_create](#)
 - [old](#)
- 動態參數 - 下列參數現在為 MariaDB 10.11 版資料庫執行個體的動態參數：
 - [innodb_log_file_size](#)
 - [innodb_write_io_threads](#)
 - [innodb_read_io_threads](#)
- 參數的新預設值：下列參數在 MariaDB 10.11 版資料庫執行個體中有新的預設值：
 - [explicit_defaults_for_timestamp](#) 參數的預設值從 OFF 變更為 ON。
 - [optimizer_prune_level](#) 參數的預設值從 1 變更為 2。
- 參數的新有效值：下列參數在 MariaDB 10.11 版資料庫執行個體中有新的有效值：
 - [old](#) 參數的有效值已與 [old_mode](#) 參數的有效值合併。
 - [histogram_type](#) 參數的有效值現在包括 JSON_HB。
 - [innodb_log_buffer_size](#) 參數的有效值範圍現在為 262144 至 4294967295 (256KB 至 4096MB)。
 - [innodb_log_file_size](#) 參數的有效值範圍現在為 4194304 至 512GB (4MB 至 512GB)。
 - [optimizer_prune_level](#) 參數的有效值現在包括 2。
- 新參數 - 下列參數是 MariaDB 10.11 版資料庫執行個體的新參數：
 - [binlog_alter_two_phase](#) 參數可改善複寫效能。
 - [log_slow_min_examined_row_limit](#) 參數可改善效能。
 - [log_slow_query](#) 參數和 [log_slow_query_file](#) 參數分別是 `slow_query_log` 和 `slow_query_log_file` 的別名。
 - [optimizer_extra_pruning_depth](#)
 - [system_versioning_insert_history](#)

如需所有功能和文件清單，請參閱 MariaDB 網站上的下列資訊。

版本	變更和改進	版本備註
MariaDB 10.7	MariaDB 10.7 中的變更和改進	版本備註 - MariaDB 10.7 系列
MariaDB 10.8	MariaDB 10.8 中的變更和改進	版本備註 - MariaDB 10.8 系列
MariaDB 10.9	MariaDB 10.9 中的變更和改進	版本備註 - MariaDB 10.9 系列
MariaDB 10.10	MariaDB 10.10 中的變更和改進	版本備註 - MariaDB 10.10 系列
MariaDB 10.11	MariaDB 10.11 中的變更和改進	版本備註 - MariaDB 10.11 系列

如需未支援的功能清單，請參閱[Amazon RDS 不支援的 MariaDB 功能](#)。

Amazon RDS 上的 MariaDB 10.6 支援

對於執行 MariaDB 10.6 版或更新版本的資料庫執行個體，Amazon RDS 支援下列新功能：

- **MyRocks 儲存引擎**：您可以將 MyRocks 儲存引擎與 RDS for MariaDB 結合使用，以最佳化寫入密集型、高效能 Web 應用程序的儲存消耗量。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 支援的儲存引擎](#) 和 [MyRocks](#)。
- **AWS Identity and Access Management (IAM) 資料庫身分驗證** – 您可以使用 IAM 資料庫身分驗證來提高與 MariaDB 資料庫執行個體的連線安全性和集中管理。如需更多詳細資訊，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。
- **升級選項**：您現在可從先前的任何主要版本 (10.3、10.4、10.5) 升級至 RDS for MariaDB 10.6 版。您也可以將現有 MySQL 5.6 或 5.7 資料庫執行個體的快照還原為 MariaDB 10.6 執行個體。如需更多詳細資訊，請參閱 [升級 MariaDB 資料庫引擎](#)。
- **延遲複寫**：您現在可以設定僅供讀取複本落後於來源資料庫的可設定時間段。在標準 MariaDB 複寫配置中，來源和複本之間的複製延遲最小。藉由延遲複寫，您可以將故意延遲設定為災難復原策略。如需更多詳細資訊，請參閱 [透過 MariaDB 設定延遲複寫](#)。
- **Oracle PL/SQL 兼容性**：透過使用 RDS for MariaDB 10.6 版，您可以更輕鬆地將舊版 Oracle 應用程式遷移至 Amazon RDS。如需詳細資訊，請參閱 [SQL_MODE=ORACLE](#)。
- **不可部分完成 DDL**：若使用 RDS for MariaDB 10.6 版，您的動態資料語言 (DDL) 陳述式會具有較強的損毀安全 (Crash-safe) 功能。CREATE TABLE、ALTER TABLE、RENAME TABLE、DROP

TABLE、DROP DATABASE 和相關的 DDL 陳述式現在不可部分完成。陳述式要么成功，要么完全失敗。如需詳細資訊，請參閱[不可部分完成 DDL](#)。

- 其他增強功能：這些增強功能包括在 SQL 中將 JSON 資料轉換為關係格式的 JSON_TABLE 函數，以及使用 Innodb 更快地加載空白資料表資料。其中還包括用於分析和故障排除的新 sys_schema、忽略未使用索引的最佳化程序增強功能以及效能改進。如需詳細資訊，請參閱[JSON_TABLE](#)。
- 參數的新預設值：下列參數具有 MariaDB 10.6 版資料庫執行個體的新預設值：
 - 下列參數的預設值已從 utf8 變更為 utf8mb3：
 - [character_set_client](#)
 - [character_set_connection](#)
 - [character_set_results](#)
 - [character_set_system](#)

雖然這些參數的預設值已變更，但功能並無變化。如需詳細資訊，請參閱 MariaDB 文件中的[支援的字元集和定序](#)。

- [collation_connection](#) 參數的預設值已從 utf8_general_ci 變更為 utf8mb3_general_ci。雖然此參數的預設值已變更，但功能並無變化。
- [old_mode](#) 參數的預設值已從未設定變更為 UTF8_IS_UTF8MB3。雖然此參數的預設值已變更，但功能並無變化。

如需所有 MariaDB 10.6 功能的清單及其文件，請參閱 MariaDB 網站上的[MariaDB 10.6 的變更和改進及版本備註 - MariaDB 10.6 系列](#)。

如需未支援的功能清單，請參閱[Amazon RDS 不支援的 MariaDB 功能](#)。

Amazon RDS 上的 MariaDB 10.5 支援

對於執行 MariaDB 10.5 版或更新版本的資料庫執行個體，Amazon RDS 支援下列新功能：

- InnoDB 增強 – MariaDB 10.5 版，包括 InnoDB 增強。如需詳細資訊，請參閱 MariaDB 文件中的[InnoDB：效能改進等](#)。
- 效能結構描述更新 – MariaDB 10.5 版，包括效能結構描述更新。如需詳細資訊，請參閱 MariaDB 文件中的[效能結構描述更新以符合 MySQL 5.7 檢測和資料表](#)。
- InnoDB 重做日誌中的一個檔案 – 在 10.5 版之前的 MariaDB 版本中，innodb_log_files_in_group 參數的值被設定為 2。在 MariaDB 10.5 版中，此參數的值設定為 1。

如果您要從先前的版本升級到 MariaDB 10.5 版，並且您不修改參數，則 `innodb_log_file_size` 參數值不會變更。不過，它會套用至一個記錄檔，而不是兩個記錄檔。結果是升級後的 MariaDB 10.5 版資料庫執行個體使用升級之前所使用的重做日誌大小的一半。這項變更可能會產生明顯的效能影響。若要解決這個問題，您可以將 `innodb_log_file_size` 參數的值加倍。如需修改參數的相關資訊，請參閱 [修改資料庫參數群組中的參數](#)。

- 不支援 SHOW SLAVE STATUS 命令 – 在 10.5 版之前的 MariaDB 版本中，SHOW SLAVE STATUS 命令需要的 REPLICATION SLAVE 權限。在 MariaDB 10.5 版中，對等命令 SHOW REPLICATION STATUS 需要 REPLICATION REPLICATION ADMIN 權限。這個新權限不會授與 RDS 主要使用者。

不使用 SHOW REPLICATION STATUS 命令，而是執行新的 `mysql.rds_replica_status` 存放的程序來傳回類似的資訊。如需更多詳細資訊，請參閱 [mysql.rds_replica_status](#)。

- 不支援 SHOW RELAYLOG EVENTS 命令 – 在 10.5 版之前的 MariaDB 版本中，SHOW RELAYLOG EVENTS 命令需要 REPLICATION SLAVE 權限。在 MariaDB 10.5 版本中，這個命令需要 REPLICATION REPLICATION ADMIN 權限。這個新權限不會授與 RDS 主要使用者。
- 參數的新預設值 – 下列參數具有 MariaDB 10.5 版資料庫執行個體的新預設值：
 - [max_connections](#) 參數的預設值已變更為 `LEAST({DBInstanceClassMemory/25165760},12000)`。如需有關 LEAST 參數函數的資訊，請參閱 [資料庫參數函數](#)。
 - [innodb_adaptive_hash_index](#) 參數的預設值已變更為 OFF (0)。
 - [innodb_checksum_algorithm](#) 參數的預設值已變更為 `full_crc32`。
 - [innodb_log_file_size](#) 參數的預設值已變更為 2 GB。

如需所有 MariaDB 10.5 功能的清單及其文件，請參閱 MariaDB 網站上的 [MariaDB 10.5 的變更和改進及版本備註 - MariaDB 10.5 系列](#)。

如需未支援的功能清單，請參閱 [Amazon RDS 不支援的 MariaDB 功能](#)。

Amazon RDS 上的 MariaDB 10.4 支援

對於執行 MariaDB 10.4 版或更新版本的資料庫執行個體，Amazon RDS 支援下列新功能：

- 使用者帳戶安全性增強功能 – [密碼到期](#) 和 [帳戶鎖定](#) 改進項目
- 最佳化工具增強功能 – [最佳化工具追蹤功能](#)
- InnoDB 增強功能 – [即時 DROP COLUMN 支援](#) 和即時 `VARCHAR ROW_FORMAT=DYNAMIC` 的延伸與 `ROW_FORMAT=COMPACT`

- 新參數 – 包括 [tcp_nodedelay](#)、[tls_version](#)，和 [gtid_cleanup_batch_size](#)

如需所有 MariaDB 10.4 功能的清單及其文件，請參閱 MariaDB 網站上的 [MariaDB 10.4 的變更和改進及版本備註 - MariaDB 10.4 系列](#)。

如需未支援的功能清單，請參閱[Amazon RDS 不支援的 MariaDB 功能](#)。

Amazon RDS 上的 MariaDB 10.3 支援

對於執行 MariaDB 10.3 版或更新版本的資料庫執行個體，Amazon RDS 支援下列新功能：

- Oracle 相容性 – PL/SQL 相容性剖析器、序列、INTERSECT 和 EXCEPT 以補充 UNION、全新 TYPE OF 和 ROW TYPE OF 宣告，以及不可見欄位
- 暫存資料處理 – 系統版本控制表，用於查詢過去和現在的資料庫狀態
- 彈性 – 使用者定義的彙總、儲存獨立的欄位壓縮，以及代理通訊協定支援，可將用戶端 IP 地址轉送到伺服器
- 易管理性 – 快速 ADD COLUMN 操作，以及快速失敗資料定義語言 (DDL) 操作

如需所有 MariaDB 10.3 功能的清單及其說明文件，請參閱 MariaDB 網站上的 [MariaDB 10.3 的變更和改進及版本備註 - MariaDB 10.3 系列](#)。

如需未支援的功能清單，請參閱[Amazon RDS 不支援的 MariaDB 功能](#)。

Amazon RDS 上的 MariaDB 支援的儲存引擎

RDS for MariaDB 支援以下儲存引擎。

主題

- [InnoDB 儲存引擎](#)
- [MyRocks 儲存引擎](#)

RDS for MariaDB 目前不支援其他儲存引擎。

InnoDB 儲存引擎

雖然 MariaDB 支援多種功能不盡相同的儲存引擎，但並非所有引擎的復原能力和資料耐用性都經過最佳化設計。InnoDB 為 Amazon RDS 上 MariaDB 資料庫執行個體的推薦儲存引擎。時間點還原和快照

還原等 Amazon RDS 功能皆須搭配可復原的儲存引擎才能執行，且僅有適用於 MariaDB 版本的建議儲存引擎才支援這些功能。

如需詳細資訊，請參閱 [InnoDB](#)。

MyRocks 儲存引擎

MyRocks 儲存引擎可用於 RDS for MariaDB 10.6 版及更高版本。在生產資料庫中使用 MyRocks 儲存引擎之前，我們建議您執行徹底的基準化分析和測試，以驗證您的使用案例相對於 InnoDB 是否具有任何潛在優勢。

MariaDB 10.6 版的預設參數群組包括 MyRocks 參數。如需詳細資訊，請參閱 [MariaDB 的參數](#) 及 [使用參數群組](#)。

若要建立使用 MyRocks 儲存引擎的資料表，請在 CREATE TABLE 陳述式中指定 ENGINE=RocksDB。下列範例會建立使用 MyRocks 儲存引擎的資料表。

```
CREATE TABLE test (a INT NOT NULL, b CHAR(10)) ENGINE=RocksDB;
```

強烈建議您不要執行跨越 InnoDB 和 MyRocks 資料表的交易處理。MariaDB 不保證跨儲存引擎的交易處理 ACID (不可部分完成性、一致性、隔離性、持久性)。儘管可以在資料庫執行個體中同時包含 InnoDB 和 MyRocks 資料表，但我們不建議使用此方法，除非處於從一個儲存引擎遷移至另一個儲存引擎期間。當 InnoDB 和 MyRocks 資料表都存在於資料庫執行個體中時，每個儲存引擎都有自己的緩衝集區，這可能會導致效能下降。

MyRocks 不支援 SERIALIZABLE 隔離或間隙鎖定。所以，通常您不能將 MyRocks 用於基於陳述句的複寫。如需詳細資訊，請參閱 [MyRocks 和複寫](#)。

目前，您只能修改下列 MyRocks 參數：

- [rocksdb_block_cache_size](#)
- [rocksdb_bulk_load](#)
- [rocksdb_bulk_load_size](#)
- [rocksdb_deadlock_detect](#)
- [rocksdb_deadlock_detect_depth](#)
- [rocksdb_max_latest_deadlocks](#)

MyRocks 儲存引擎和 InnoDB 儲存引擎可根據 rocksdb_block_cache_size 和 innodb_buffer_pool_size 參數的設定來競爭記憶體。在某些情況下，您可能只打算在特定

資料庫執行個體上使用 MyRocks 儲存引擎。如果是這樣，建議將 `innodb_buffer_pool_size` `minimal` 參數設定為最小值，同時將盡力提高 `rocksdb_block_cache_size` 的值。

您可以使用 [DescribeDBLogFiles](#) 和 [DownloadDBLogFilePortion](#) 操作存取 MyRocks 日誌檔案。

如需有關 MyRocks 的詳細資訊，請參閱 MariaDB 網站上的 [MyRocks](#)。

Amazon RDS 上 MariaDB 的快取預備

InnoDB 快取預備功能可在資料庫執行個體關閉時，儲存緩衝集區的當前狀態，等到資料庫執行個體啟動時，再使用儲存的資訊重新載入緩衝集區，藉此提升 MariaDB 資料庫執行個體的效能。此方法不必在資料庫正常使用情況下將緩衝集區「熱機」，而是將已知常用查詢的頁面預先載入緩衝集區。如需快取預備功能的詳細資訊，請參閱 MariaDB 文件中的 [Dumping and restoring the buffer pool](#) (傾印和還原緩衝集區)。

在 MariaDB 10.3 版和更新版本的資料庫執行個體上，會預設啟用快取預備功能。如果要自行啟用該功能，請在資料庫執行個體的參數群組中，將 `innodb_buffer_pool_dump_at_shutdown` 和 `innodb_buffer_pool_load_at_startup` 參數設為 1。在參數群組中變更這些參數值會影響使用該參數群組的所有 MariaDB 資料庫執行個體。請先為資料庫執行個體建立新的參數群組，才能啟用特定 MariaDB 資料庫執行個體的快取預備功能。如需參數群組的相關資訊，請參閱[使用參數群組](#)。

快取預備功能主要可為使用標準儲存體的資料庫執行個體提供效能方面的好處。若您使用的是 PIOPS 儲存體，一般來說不會察覺到顯著的效能優勢。

Important

如果您的 MariaDB 資料庫執行個體沒有正常關閉 (例如容錯移轉期間)，則緩衝集區狀態不會儲存到磁碟中。這種情形下，MariaDB 會在資料庫執行個體重新啟動時，載入任何可用的緩衝集區檔案。這不會造成任何損害，只是還原的緩衝集區無法反映緩衝集區在重新啟動之前的最新狀態。為確保您有最新狀態的緩衝集區可在啟動時預備快取，建議您定期「隨需」傾印緩衝集區。您可以隨需傾印或載入緩衝集區。

您可建立事件，以定期自動傾印緩衝集區。舉例來說，下列陳述式會建立名為 `periodic_buffer_pool_dump` 的事件，每小時傾印緩衝集區一次。

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

如需詳細資訊，請參閱 MariaDB 文件中的[事件](#)。

隨需傾印與載入緩衝集區

您可以使用下列預存程序來隨需儲存和載入快取：

- 若要將緩衝集區的最新狀態傾印至磁碟，請呼叫 [mysql.rds_innodb_buffer_pool_dump_now](#) 預存程序。
- 若要從磁碟載入所儲存的緩衝集區狀態，請呼叫 [mysql.rds_innodb_buffer_pool_load_now](#) 預存程序。
- 若要取消進行中的載入操作，請呼叫 [mysql.rds_innodb_buffer_pool_load_abort](#) 預存程序。

Amazon RDS 不支援的 MariaDB 功能

Amazon RDS 上不支援下列 MariaDB 功能：

- S3 儲存引擎
- 身分驗證外掛程式 – GSSAPI
- 身分驗證外掛程式 – Unix Socket
- AWS Key Management 加密外掛程式
- 低於 MariaDB 10.6 版的延遲複寫
- InnoDB 和 Aria 的原生 MariaDB 靜態加密

您可以按照 [加密 Amazon RDS 資源](#) 中的說明啟用 MariaDB 資料庫執行個體的靜態加密。

- HandlerSocket
- 低於 MariaDB 10.6 版的 JSON 資料表類型
- MariaDB ColumnStore
- MariaDB Galera 叢集
- 多來源複寫
- 低於 MariaDB 10.6 版的 MyRocks 儲存引擎
- 密碼驗證外掛程式、`simple_password_check` 和 `cracklib_password_check`
- Spider 儲存引擎
- Sphinx 儲存引擎

- TokuDB 儲存引擎
- 儲存引擎專屬物件屬性，如 MariaDB 文件中的[引擎定義的新資料表/欄位/索引屬性](#)所述
- 資料表和資料表空間加密
- Hashicorp 金鑰管理外掛程式
- 同時執行兩項升級

為了提供受管理的服務體驗，Amazon RDS 並不提供資料庫執行個體的 Shell 存取權，而且會將存取權限制於某些需要進階權限的系統程序和資料表。Amazon RDS 可支援使用任何標準 SQL 用戶端應用程式存取資料庫執行個體上的資料庫。Amazon RDS 不允許使用者利用 Telnet、安全殼層 (SSH) 或 Windows 遠端桌面連線，直接託管資料庫執行個體的存取權。

Amazon RDS 上的 MariaDB 版本

對於 MariaDB，版本編號組織為版本 X.Y.Z。在 Amazon RDS 術語中，X.Y 代表主要版本，Z 則代表次要版本編號。進行 Amazon RDS 實作時，若主要版本編號有所變更 (例如從 10.5 版更新至 10.6 版)，即視為主要版本變更。如果只有次要版本號碼變更 (例如從 10.6.14 版變更為 10.6.16)，則版本變更會被視為次要變更。

主題

- [Amazon RDS 上支援的 MariaDB 次要版本](#)
- [Amazon RDS 上支援的 MariaDB 主要版本](#)
- [Amazon RDS for MariaDB 的已取代版本](#)

Amazon RDS 上支援的 MariaDB 次要版本

Amazon RDS 目前支援下列 MariaDB 次要版本。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。

DB engine version (資料庫引擎版本)	社群發佈日期	RDS 發佈日期	RDS 標準支援結束日期
10.11			
10.11.8	2024 年五月十六日	2024 年六月十四日	二〇二五年
10.11.7	二零二四年二月七日	2024 年二月二十六日	2025 年 3 月
10.11.6	2023 年十一月十三日	2023 年十二月十二日	2025 年 3 月
10.11.5	2023 年 8 月 14 日	2023 年 9 月 7 日	2024 年 9 月
10.11.4	2023 年 6 月 7 日	2023 年 8 月 21 日	2024 年 9 月
10.6			

DB engine version (資料庫引擎版本)	社群發佈日期	RDS 發佈日期	RDS 標準支援結束日期
10.6.18	2024 年五月十六日	2024 年六月十四日	二〇二五年
10.6.17	二零二四年二月七日	2024 年二月二十六日	2025 年 3 月
10.6.16	2023 年十一月十三日	2023 年十二月十二日	2025 年 3 月
10.6.15	2023 年 8 月 14 日	2023 年 9 月 7 日	2024 年 9 月
10.6.14	2023 年 6 月 7 日	2023 年 6 月 22 日	2024 年 9 月
10.6.13	2023 年 5 月 10 日	2023 年 6 月 15 日	2024 年 9 月
10.5			
10.5.25	2024 年五月十六日	2024 年六月十四日	二〇二五年
10.5.24	二零二四年二月七日	2024 年二月二十六日	2025 年 3 月
10.5.23	2023 年十一月十三日	2023 年十二月十二日	2025 年 3 月
10.5.22	2023 年 8 月 14 日	2023 年 9 月 7 日	2024 年 9 月
10.5.21	2023 年 6 月 7 日	2023 年 6 月 22 日	2024 年 9 月
10.5.20	2023 年 5 月 10 日	2023 年 6 月 15 日	2024 年 9 月
10.4			
10.4.34	2024 年五月十六日	2024 年六月十四日	2024 年八月
10.4.33	二零二四年二月七日	2024 年二月二十六日	2024 年八月
10.4.32	2023 年十一月十三日	2023 年十二月十二日	2024 年八月
10.4.31	2023 年 8 月 14 日	2023 年 9 月 7 日	2024 年八月
10.4.30	2023 年 6 月 7 日	2023 年 6 月 22 日	2024 年八月
10.4.29	2023 年 5 月 10 日	2023 年 6 月 15 日	2024 年八月

建立新的資料庫執行個體時，您可以指定使用目前支援的任一 MariaDB 版本。您可以指定主要版本 (例如 MariaDB 10.5)，並指定使用主要版本所支援的任一次要版本。若未指定版本，Amazon RDS 會預設使用支援的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 會預設使用您指定主要版本的最新次要版本。若要查看支援的版本清單以及新建立的資料庫執行個體的預設值，請使用 [describe-db-engine-versions](#) AWS CLI 指令。

例如，若要列出 RDS for MariaDB 支援的引擎版本，請執行下列 CLI 命令：

```
aws rds describe-db-engine-versions --engine mariadb --query "*[].[Engine:Engine,EngineVersion:EngineVersion]" --output text
```

預設的 MariaDB 版本可能會隨著 AWS 區域而有所不同。若要建立具有特定次要版本的資料庫執行個體，請在建立資料庫執行個體期間指定次要版本。您可以 AWS 區域 使用以下 AWS CLI 命令確定的默認次要版本：

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version major-engine-version --region region --query "*[].[Engine:Engine,EngineVersion:EngineVersion]" --output text
```

將 *major-engine-version* 取代為主要引擎版本，並將 *region* 取代為 AWS 區域。例如，下列 AWS CLI 命令會傳回 10.5 主要版本和美國西部 (奧勒岡) AWS 區域 (美國西部 -2) 的預設 MariaDB 次要引擎版本：

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version 10.5 --region us-west-2 --query "*[].[Engine:Engine,EngineVersion:EngineVersion]" --output text
```

Amazon RDS 上支援的 MariaDB 主要版本

RDS for MariaDB 主要版本至少在對應社群版本的生命週期結束前都會保持可用。您可以使用下列日期來規劃測試和升級週期。如果 Amazon 延長對 RDS for MariaDB 版本的支援時間超過原先聲明的時間，我們計劃更新此表格以反映後來的日期。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。

MariaDB 主要版本	社群發佈日期	RDS 發佈日期	社群生命週期結束日期	RDS 標準支援結束日期
MariaDB 10.11	2023 年 2 月 16 日	2023 年 8 月 21 日	2028 年 2 月 16 日	2028 年 2 月
MariaDB 10.6	2021 年 7 月 6 日	2022 年 2 月 3 日	2026 年 7 月 6 日	2026 年 7 月
MariaDB 10.5	2020 年 6 月 24 日	2021 年 1 月 21 日	2025 年 6 月 24 日	2025 年 6 月
MariaDB 10.4	2019 年 6 月 18 日	2020 年 4 月 6 日	2024 年 6 月 18 日	2024 年八月

Amazon RDS for MariaDB 的已取代版本

適用 Amazon RDS for MariaDB 版本 10.0、10.1、10.2 和 10.3 版已棄用。

如需適用於 MariaDB 的 Amazon RDS 淘汰政策相關資訊，請參閱 [Amazon RDS 常見問答集](#)。

連接至執行 MariaDB 資料庫引擎的資料庫執行個體

當 Amazon RDS 佈建您的資料庫執行個體後，您就可以使用任何標準 MariaDB 用戶端應用程式或公用程式來連接至執行個體。在連線字串中，指定來自主要執行個體端點的網域名稱系統 (DNS) 位址做為主機參數。您也指定來自資料庫執行個體端點的連接埠號碼做為連接埠參數。

您能夠利用 MySQL 命令列用戶端等工具，藉此連接至 Amazon RDS for MariaDB 資料庫執行個體。如需有關使用 MySQL 命令列用戶端的詳細資訊，請參閱 MariaDB 文件中的 [mysql 命令列用戶端](#)。您可以用來連線的一個 GUI 應用程式是 Heidi。如需詳細資訊，請參閱 [下載 HeidiSQL](#) 頁面。如需安裝 MySQL (包括 MySQL 命令列用戶端) 的資訊，請參閱 [安裝和升級 MySQL](#)。

大多數的 Linux 發行版均包括 MariaDB 用戶端，而不是 Oracle MySQL 用戶端。若要在 Amazon Linux 2023 上安裝 MySQL 命令列用戶端，請執行下列命令：

```
sudo dnf install mariadb105
```

若要在 Amazon Linux 2 上安裝 MySQL 命令列用戶端，請執行下列命令：

```
sudo yum install mariadb
```

若要在大多數以 Debian 為基礎的 Linux 發行版上安裝 MySQL 命令列用戶端，請執行下列命令。

```
apt-get install mariadb-client
```

若要檢查 MySQL 命令列用戶端的版本，請執行下列命令。

```
mysql --version
```

若要讀取目前用戶端版本的 MySQL 文件，請執行下列命令。

```
man mysql
```

若要基於 Amazon VPC 從 Virtual Private Cloud (VPC) 外部連線至資料庫執行個體，資料庫執行個體必須可公開存取。此外，必須使用資料庫執行個體安全群組的入站規則授予存取權，且必須符合其他需求。如需更多詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

您可以在 MariaDB 資料庫執行個體的連線上使用 SSL 加密。如需相關資訊，請參閱「[將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用](#)」。

主題

- [尋找 MariaDB 資料庫執行個體的連線資訊](#)
- [從 MySQL 命令列用戶端連線 \(未加密\)](#)
- [使用 Amazon Web Services \(AWS \) JDBC 驅動程序連接到適用於 MariaDB 的 RDS](#)
- [使用 Amazon Web Services 連接到 RDS \(AWS \) Python 驅動程序](#)
- [對 MariaDB 資料庫執行個體的連線進行故障診斷](#)

尋找 MariaDB 資料庫執行個體的連線資訊

資料庫執行個體的連線資訊包括其端點、連接埠，以及有效的資料庫使用者，例如主要使用者。例如，假設端點值為 `mydb.123456789012.us-east-1.rds.amazonaws.com`。在此情況下，連接埠值為 `3306`，資料庫使用者為 `admin`。藉由此資訊，您可以在連線字串中指定下列值：

- 對於主機或主機名稱或 DNS 名稱，請指定 `mydb.123456789012.us-east-1.rds.amazonaws.com`。
- 對於連接埠，指定 `3306`。
- 對於使用者，請指定 `admin`。

若要連線到資料庫執行個體，請針對 MariaDB 資料庫引擎使用任何用戶端。例如，您可以使用 MySQL 命令列用戶端或 MySQL 工作台。

若要尋找資料庫執行個體的連線資訊，您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) [describe-db-instances](#) 命令或 Amazon RDS API [DescribeDBInstances](#) 操作列出其詳細資訊。

主控台

若要尋找資料庫執行個體的連線資訊 AWS Management Console

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫) 以顯示資料庫執行個體的清單。
3. 選擇 MariaDB 資料庫執行個體名稱以顯示其詳細資訊。
4. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連接至資料庫執行個體。

RDS > Databases > mydb

mydb

Summary

DB identifier mydb	CPU 2.33%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port

Endpoint mydb. [redacted].us-east-1.rds.amazonaws.com	Netw
Port 3306	Availa us-eas
	VPC vpc-65
	Subne defaul

5. 如果您需要尋找主要使用者名稱，請選擇 Configuration (組態) 索引標籤並檢視 Master username (主要使用者名稱) 值。

AWS CLI

若要使用尋找 MariaDB 資料庫執行個體的連線資訊 AWS CLI，請呼叫命令。[describe-db-instances](#) 在呼叫中，查詢資料庫執行個體 ID、端點、連接埠和主要使用者名稱。

對於 Linux macOS、或 Unix：

```
aws rds describe-db-instances \  
  --filters "Name=engine,Values=mariadb" \  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

在 Windows 中：

```
aws rds describe-db-instances ^  
  --filters "Name=engine,Values=mariadb" ^  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

輸出內容應如下所示：

```
[  
  [  
    "mydb1",  
    "mydb1.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "mydb2",  
    "mydb2.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ]  
]
```

RDS API

若要使用 Amazon RDS API 尋找資料庫執行個體的連線資訊，請呼叫 [DescribeDBInstances](#) 操作。在輸出中，尋找端點地址、端點連接埠和主要使用者名稱的值。

從 MySQL 命令列用戶端連線 (未加密)

⚠ Important

僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 MySQL 連線。如需使用加密連線的詳細資訊，請參閱 [使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)。

若要使用 MySQL 命令列用戶端連線至資料庫執行個體，在用戶端電腦上的命令提示字元中輸入下列命令。如此，這就會將您連限制 MariaDB 資料庫執行個體上的資料庫。會取代您用於 *<endpoint>* 資料庫執行個體的 DNS 名稱 (端點)，以及取代您用於 *<mymasteruser>* 的主要使用者名稱。提示您輸入密碼時，請提供您使用的主要密碼。

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

輸入使用者的密碼之後，您會看到類似如下的輸出。

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

使用 Amazon Web Services (AWS) JDBC 驅動程序連接到適用於 MariaDB 的 RDS

Amazon Web Services (AWS) JDBC 驅動程序被設計為一個高級 JDBC 包裝。這個包裝是補充和擴展現有 JDBC 驅動程序的功能。該驅動程序與社區 MySQL 連接器/J 驅動程序和社區 MariaDB 連接器/J 驅動程序兼容。

若要安裝 AWS JDBC 驅動程式，請附加 AWS JDBC 驅動程式 .jar 檔案 (位於應用程式中 CLASSPATH)，並保留對個別社群驅動程式的參考。更新相應的連接 URL 前綴，如下所示：

- jdbc:mysql:// 設定為 jdbc:aws-wrapper:mysql://
- jdbc:mariadb:// 設定為 jdbc:aws-wrapper:mariadb://

如需有關 AWS JDBC 驅動程式的詳細資訊以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS\) JDBC 驅動程式 GitHub 儲存庫](#)。

使用 Amazon Web Services 連接到 RDS (AWS) Python 驅動程序

Amazon Web Services (AWS) Python 驅動程序被設計為一個先進的 Python 包裝。這個包裝器是補充並擴展了開源 Psycopg 驅動程序的功能。AWS Python 驅動程式支援 3.8 及更高版本。您可以使用 pip 命令以及 psycopg 開放原始碼套件來安裝套件。aws-advanced-python-wrapper

有關 AWS Python 驅動程序的更多信息以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS \) Python 驅動程序 GitHub 儲存庫](#)。

對 MariaDB 資料庫執行個體的連線進行故障診斷

無法連接至新的資料庫執行個體有下列兩個常見原因：

- 建立資料庫執行個體所用的安全群組，未授權從執行 MariaDB 應用程式或公用程式的裝置或 Amazon EC2 執行個體進行連線。資料庫執行個體必須擁有能授權連線的 VPC 安全群組。如需詳細資訊，請參閱 [Amazon VPC](#) 和 [Amazon RDS](#)。

您可以新增或編輯安全群組中的傳入規則。針對 Source (來源)，選擇 My IP (我的 IP)。此舉允許透過您的瀏覽器中偵測到的 IP 地址存取資料庫執行個體。

- 資料庫執行個體是使用預設連接埠 3306 來建立，而公司的防火牆規則禁止從公司網路中的裝置連接至該連接埠。若要避免此失敗情況，請以不同的連接埠重新建立執行個體。

如需連線問題的詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

保護 MariaDB 執行個體連線安全

您可以管理 MariaDB 資料庫執行個體安全。

主題

- [Amazon RDS 上的 MariaDB 安全性](#)
- [使用 SSL/TLS 加密對 MariaDB 資料庫執行個體的用戶端連線](#)
- [更新應用程式，以使用新的 SSL/TLS 憑證連線至 MariaDB 執行個體](#)

Amazon RDS 上的 MariaDB 安全性

MariaDB 資料庫執行個體的安全性有三個管理層級：

- AWS Identity and Access Management 會控管能在資料庫執行個體上執行 Amazon RDS 管理動作的對象。當您使用 IAM 登入資料連線至 AWS 時，IAM 帳戶必須具備能授予所需許可的 IAM 政策，才能執行 Amazon RDS 管理操作。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。
- 建立資料庫執行個體時，您可使用 VPC 安全群組控管哪些裝置和 Amazon EC2 執行個體可以開啟與端點和資料庫執行個體連接埠的連線。您可以使用 Secure Sockets Layer (SSL) 和 Transport Layer Security (TLS) 建立這些連線。此外，貴公司的防火牆規則可控管在公司內執行的裝置是否可開啟與資料庫執行個體的連線。
- 對 MariaDB 資料庫執行個體建立連線之後，登入和許可的驗證方式就如同在 MariaDB 獨立執行個體中一樣。CREATE USER、RENAME USER、GRANT、REVOKE 及 SET PASSWORD 等命令的運作方式如同在獨立資料庫中一樣，都會直接修改資料庫結構描述資料表。

在您建立 Amazon RDS 資料庫執行個體後，主要使用者具有下列預設權限：

- alter
- alter routine
- create
- create routine
- create temporary tables
- create user
- create view

- delete
- drop
- event
- execute
- grant option
- index
- insert
- lock tables
- process
- references
- reload

此權限在 MariaDB 資料庫執行個體上受限。不授權存取 FLUSH LOGS 或 FLUSH TABLES WITH READ LOCK 操作。

- replication client
- replication slave
- select
- show databases
- show view
- trigger
- update

如需這些權限的詳細資訊，請參閱 MariaDB 文件中的[使用者帳戶管理](#)。

Note

雖然您可刪除資料庫執行個體上的主要使用者，但不建議這麼做。若要重新建立主要使用者，請使用 ModifyDBInstance API 或 modify-db-instance AWS CLI，並以適當的參數指定新的主要使用者密碼。若執行個體中不存在此主要使用者，系統會以指定的密碼建立主要使用者。

為了提供各資料庫執行個體管理服務，建立資料庫執行個體時，系統會一併建立 rdsadmin 使用者。若企圖移除、重新命名 rdsadmin 帳戶或變更其密碼或權限，皆會導致錯誤。

請限制使用標準 kill 和 kill_query 命令，藉此允許資料庫執行個體的管理操作。此外，本服務還提供適用於 MariaDB 及 MySQL 的 Amazon RDS 命令 mysql.rds_kill、mysql.rds_kill_query 與 mysql.rds_kill_query_id，讓您在結束資料庫執行個體上的使用者工作階段或查詢。

使用 SSL/TLS 加密對 MariaDB 資料庫執行個體的用戶端連線

Secure Sockets Layer (SSL) 是一種產業標準通訊協定，可保護用戶端與伺服器之間的網路連線。SSL 3.0 版之後，名稱已變更為 Transport Layer Security (TLS)。Amazon RDS 支援 MariaDB 資料庫執行個體的 SSL/TLS 加密。使用 SSL/TLS，您可以加密應用程式用戶端與 MariaDB 資料庫執行個體之間的連接。SSL/TLS 支援在所有方面都可以使用。AWS 區域

主題

- [將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用](#)
- [MariaDB 資料庫執行個體的所有連線都需要 SSL/TLS](#)
- [使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)

將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用

當 Amazon RDS 佈建執行個體時，Amazon RDS 會建立 SSL/TLS 憑證，並將該憑證安裝在資料庫執行個體上。憑證由憑證授權機構簽署。SSL/TLS 憑證會以通用名稱 (CN) 加入資料庫執行個體端點，讓 SSL/TLS 憑證免於遭受詐騙攻擊。

Amazon RDS 建立的 SSL/TLS 憑證是可信任的根實體，應可適用於大多數情況，但若您的應用程式不接受憑證鏈結，則可能無法使用。若您的應用程式不接受憑證鏈結，您可能需要使用中繼憑證來連線至您的 AWS 區域。例如，您必須使用中繼憑證來連線到使用 SSL/TLS 的 AWS GovCloud (US) 區域。

如需有關下載憑證的詳細資訊，請參閱。如需在使用 SSL/TLS 搭配 MySQL 的詳細資訊，請參閱 [更新應用程式，以使用新的 SSL/TLS 憑證連線至 MariaDB 執行個體](#)。

Amazon RDS for MariaDB 支援傳輸層安全性 (TLS) 1.3、1.2、1.1 和 1.0 版。TLS 支持取決於 MariaDB 的次要版本。下表顯示了對 MariaDB 次要版本的 TLS 支援。

TLS 版本控制	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS 1.3	所有次要版本	所有次要版本	所有次要版本	所有次要版本
TLS 1.2	所有次要版本	所有次要版本	所有次要版本	所有次要版本
TLS 1.1	10.11.6 及更低版本	十六月十六日及更低版本	5 月 10 日及更低版本	10.4.32 及更低版本
TLS 1.0	10.11.6 及更低版本	十六月十六日及更低版本	5 月 10 日及更低版本	10.4.32 及更低版本

您可以要求特定使用者帳戶使用 SSL/TLS 連線。例如，您可以根據 MariaDB 版本，使用下列任一陳述式，要求使用者帳戶 `encrypted_user` 使用 SSL/TLS 連線。

使用以下陳述式。

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

如需使用 MariaDB 的 SSL/TLS 連線詳細資訊，請參閱 MariaDB 文件中的 [保護用戶端和伺服器的連線](#)。

MariaDB 資料庫執行個體的所有連線都需要 SSL/TLS

使用 `require_secure_transport` 參數來要求使用者與 MariaDB 資料庫執行個體的所有連線都使用 SSL/TLS。依預設，`require_secure_transport` 參數設為 OFF。您可以將 `require_secure_transport` 參數設為 ON，以要求對資料庫執行個體的連線使用 SSL/TLS。

Note

僅 MariaDB 10.5 版和更新版本支援 `require_secure_transport` 參數。

您可以更新資料庫執行個體的資料庫參數群組，以設定 `require_secure_transport` 參數值。您不需要重新啟動資料庫執行個體，變更即可生效。

當資料庫執行個體的 `require_secure_transport` 參數設為 ON 時，如果可以建立加密的連線，則資料庫用戶端即可連線到該資料庫叢集。否則，類似下列內容的錯誤訊息會傳回至用戶端：

```
ERROR 1045 (28000): Access denied for user 'USER'@'localhost' (using password: YES / NO)
```

如需有關設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

如需 `require_secure_transport` 參數的詳細資訊，請參閱 [MariaDB 文件](#)。

使用 SSL/TLS (加密) 從 MySQL 命令列用戶端連線

如果您使用的是 MySQL 5.7 版、MySQL 8.0 版或 MariaDB 版本，`mysql` 用戶端程式的參數略有不同。

若要找出您擁有的版本，請執行具有 `--version` 選項的 `mysql` 命令。在以下範例中，輸出顯示了用戶端程式是來自 MariaDB。

```
$ mysql --version
mysql Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

大多數 Linux 發行版本，例如 Amazon Linux、CentOS、SUSE 和 Debian 都用 MariaDB 取代 MySQL，而且發行版本中的 `mysql` 版是來自 MariaDB。

若要使用 SSL/TLS 連線到資料庫執行個體，請執行以下步驟：

使用 MySQL 命令列用戶端搭配 SSL/TLS 連接至資料庫執行個體

1. 下載適用於所有人的根憑證 AWS 區域。

如需有關下載憑證的詳細資訊，請參閱。

2. 使用 MySQL 命令列用戶端搭配 SSL/TLS 加密連接至資料庫執行個體。在 `-h` 參數中，將 DNS 名稱 (端點) 取代為您的資料庫執行個體。在 `--ssl-ca` 參數中，取代 SSL/TLS 憑證檔名稱。在 `-P` 參數中，取代為您的資料庫執行個體連接埠。在 `-u` 參數中，請以有效資料庫使用者取代使用者名稱，例如主要使用者。出現提示時，輸入主要使用者密碼。

下列範例顯示如何對於 MariaDB 用戶端更新的版本使用 `--ssl-ca` 參數啟動用戶端：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

若要要求 SSL/TLS 連線根據 SSL/TLS 憑證中的端點來驗證資料庫執行個體端點，請輸入下列命令：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-verify-server-cert -P 3306 -u myadmin -p
```

下列範例顯示如何對於 MySQL 5.7 用戶端更新的版本使用 `--ssl-ca` 參數啟動用戶端：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
```

3. 出現提示時，輸入主要使用者密碼。

您應該會看到類似下列的輸出。

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

更新應用程式，以使用新的 SSL/TLS 憑證連線至 MariaDB 執行個體

自 2023 年 1 月 13 日起，Amazon RDS 已發佈新的憑證認證機構 (CA) 憑證，使用 Secure Socket Layer 或 Transport Layer Security (SSL/TLS) 來連線至 RDS 資料庫執行個體。接下來，您可以找到更新應用程式使用新憑證的相關資訊。

本主題可協助您決定應用程式是否需要驗證憑證，才能連線至您的資料庫執行個體。

Note

有些應用程式設定為只有在成功驗證伺服器上的憑證時，才能連線至 MariaDB。對於這些應用程式，您必須更新用戶端應用程式信任存放區來包含新的 CA 憑證。

您可以指定下列 SSL 模式：disabled、preferred 及 required。當您使用 preferred SSL 模式且 CA 憑證不存在或不是最新版本時，連線會回復為未使用 SSL，並且仍然成功連線。

我們建議您避免使用 preferred 模式。在 preferred 模式中，如果連線遇到無效憑證，則會停止使用加密並繼續使用未加密連線。

更新用戶端應用程式信任存放區中的 CA 憑證之後，您就可以在資料庫執行個體輪換憑證。強烈建議先在開發或預備環境中測試這些步驟，再於生產環境中實作。

如需憑證輪換的詳細資訊，請參閱[輪換您的 SSL/TLS 憑證](#)。如需下載憑證的詳細資訊，請參閱。如需對 MariaDB 資料庫執行個體使用 SSL/TLS 的資訊，請參閱[將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用](#)。

主題

- [判斷用戶端是否需要驗證憑證才能連線](#)
- [更新應用程式信任存放區](#)
- [建立 SSL 連線的 Java 程式碼範例](#)

判斷用戶端是否需要驗證憑證才能連線

您可以檢查 JDBC 用戶端和 MySQL 用戶端是否需要驗證憑證才能連線。

JDBC

以下 MySQL Connector/J 8.0 範例指出一種方式來檢查應用程式的 JDBC 連線屬性，以判斷是否需要有效憑證才能成功連線。如需 MySQL 的所有 JDBC 連線選項的詳細資訊，請參閱 MySQL 文件中的[組態屬性](#)。

使用 MySQL Connector/J 8.0 時，如果連線屬性的 sslMode 設為 VERIFY_CA 或 VERIFY_IDENTITY，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

Note

如果您使用 MySQL Java Connector v5.1.38 或更高版本，或 MySQL Java Connector v8.0.9 或更高版本來連線至您的資料庫，即使您尚未明確設定應用程式在連線到資料庫時使用 SSL/TLS，這些用戶端驅動程式會預設為使用 SSL/TLS。此外，使用 SSL/TLS 時，它們會執行部分憑證驗證，如果資料庫伺服器憑證已過期，則無法連線。
指定此處所顯示提示以外的密碼，作為安全最佳實務。

MySQL

以下 MySQL 用戶端範例指出兩種方式來檢查指令碼的 MySQL 連線，以判斷是否需要有效憑證才能成功連線。如需 MySQL 用戶端所有連線選項的詳細資訊，請參閱 MySQL 文件中的[加密連線的用戶端組態](#)。

使用 MySQL 5.7 或 MySQL 8.0 用戶端時，如果您將 `--ssl-mode` 選項指定為 `VERIFY_CA` 或 `VERIFY_IDENTITY`，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem  
--ssl-mode=VERIFY_CA
```

使用 MySQL 5.6 用戶端時，如果您指定 `--ssl-verify-server-cert` 選項，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem  
--ssl-verify-server-cert
```

更新應用程式信任存放區

如需為 MySQL 應用程式更新信任存放區的資訊，請參閱 MariaDB 文件中的[搭配 MariaDB Connector/J 使用 TLS/SSL](#)。

如需下載根憑證的資訊，請參閱。

如需匯入憑證的範例指令碼，請參閱 [將憑證匯入信任存放區的範例指令碼](#)。

Note

更新信任存放區時，除了新增憑證，您還可以保留舊憑證。

如果您在應用程式中使用 MariaDB Connector/J JDBC 驅動程式，請在應用程式中設定下列屬性。

```
System.setProperty("javax.net.ssl.trustStore", certs);  
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

啟動應用程式時，設定下列屬性。

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -  
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

建立 SSL 連線的 Java 程式碼範例

下列程式碼範例示範如何使用 JDBC 來設定 SSL 連線。

```
private static final String DB_USER = "admin";  
  
private static final String DB_USER = "user name";  
private static final String DB_PASSWORD = "password";  
// This key store has only the prod root ca.  
private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";  
private static final String KEY_STORE_PASS = "keystore-password";  
  
public static void main(String[] args) throws Exception {  
    Class.forName("org.mariadb.jdbc.Driver");
```

```
System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

Properties properties = new Properties();
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);

Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-
public.cni62e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true",properties);
Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery("SELECT 1 from dual");

return;
}
```

Important

判斷資料庫連線使用 SSL/TLS 並更新應用程式信任存放區之後，您可以更新資料庫以使用 rds-ca-rsa 2048-g1 憑證。如需說明，請參閱[透過修改資料庫執行個體或叢集來更新 CA 憑證](#)中的步驟 3。

指定此處所顯示提示以外的密碼，作為安全最佳實務。

使用 Amazon RDS Optimized Reads 改善 RDS for MariaDB 的查詢效能

您可以使用 Amazon RDS Optimized Reads，為 RDS for MariaDB 實現更快的查詢處理。使用 RDS Optimized Reads 的 RDS for MariaDB 資料庫執行個體，相較於不使用它的情況，查詢處理速度最高可以快上 2 倍。

主題

- [RDS Optimized Reads 的概觀](#)
- [RDS Optimized Reads 的使用案例](#)
- [RDS Optimized Reads 的最佳實務](#)
- [使用 RDS Optimized Reads](#)
- [監控使用 RDS Optimized Reads 的資料庫執行個體](#)
- [RDS Optimized Reads 的限制](#)

RDS Optimized Reads 的概觀

當您使用已開啟 RDS Optimized Reads 的 RDS for MariaDB 資料庫執行個體時，您的資料庫執行個體可透過使用執行個體儲存體實現更快的查詢效能。執行個體存放區為資料庫執行個體提供暫時的區塊層級儲存空間。儲存體位於實際連接至主機伺服器的非揮發性記憶體儲存裝置 (NVMe) 固態硬碟 (SSD) 中。此儲存體已針對低延遲、高隨機 I/O 效能，以及高循序讀取輸送量進行最佳化。

當資料庫執行個體使用資料庫執行個體類別搭配執行個體儲存體 (例如 db.m5d 或 db.m6gd) 時，預設會開啟 RDS Optimized Reads。使用 RDS Optimized Reads，部分臨時物件會存放在執行個體儲存體上。這些暫存物件包括內部暫存檔、內部磁碟上暫存資料表、記憶體對應檔案，以及二進位日誌 (binlog) 快取檔案。如需執行個體儲存體的詳細資訊，請參閱《適用於 Linux 執行個體的 Amazon Elastic Compute Cloud 使用者指南》中的 [Amazon EC2 執行個體儲存體](#)。

在 MariaDB 中產生臨時物件以進行查詢處理的工作負載，可以利用執行個體儲存體來加快查詢處理速度。這種類型的工作負載包括涉及排序、雜湊彙總、高負載聯結、一般資料表運算式 (CTE) 的查詢，以及未編製索引之資料欄上的查詢。無論用於持久性 Amazon EBS 儲存體的儲存體組態為何，這些執行個體儲存體都會提供更高的 IOPS 和效能。因為 RDS Optimized Reads 將臨時物件上的操作卸載至執行個體儲存體，所以持久性儲存體 (Amazon EBS) 的每秒讀寫次數 (IOPS) 或輸送量現在可以用於持久性物件上的操作。這些操作包括一般資料檔案讀取和寫入，以及背景引擎操作，例如排清和插入緩衝區合併。

Note

手動和自動 RDS 快照都只包含持久性物件的引擎檔案。在執行個體儲存體中建立的臨時物件不會包含在 RDS 快照集中。

RDS Optimized Reads 的使用案例

如果您的工作負載嚴重依賴臨時物件 (例如內部資料表或檔案) 進行其查詢執行，則您可以從開啟 RDS Optimized Reads 中獲益。下列使用案例為 RDS Optimized Reads 的候選項目：

- 使用複雜的一般資料表運算式 (CTE)、衍生資料表和群組操作來執行分析查詢的應用程式
- 使用未最佳化查詢提供大量讀取流量的僅供讀取複本
- 執行涉及複雜作業之隨需或動態報告查詢的應用程式，例如搭配 GROUP BY 和 ORDER BY 子句的查詢
- 使用內部暫存資料表進行查詢處理的工作負載

您可以監控引擎狀態變數 `created_tmp_disk_tables`，以判斷在資料庫執行個體上建立的磁碟型暫存資料表數目。

- 直接或在程序中建立大型暫存資料表以儲存中繼結果的應用程式
- 對未編製索引之資料欄執行分組或排序的資料庫查詢

RDS Optimized Reads 的最佳實務

請使用 RDS Optimized Reads 的下列最佳實務：

- 針對唯讀查詢新增重試邏輯，以防這些查詢在執行期間由於執行個體儲存體已滿而失敗。
- 使用 CloudWatch 指標監視執行個體存放區上的可用儲存空間 `FreeLocalStorage`。如果執行個體儲存體由於資料庫執行個體的工作負載而達到其限制，請修改資料庫執行個體，以使用較大的資料庫執行個體類別。
- 當您的資料庫執行個體有足夠的記憶體，但仍達到執行個體儲存體的儲存限制時，請增加 `binlog_cache_size` 值以在記憶體中維護工作階段特定的 binlog 項目。此組態可防止將 binlog 項目寫入至磁碟上的暫存 binlog 快取檔案。

`binlog_cache_size` 參數是工作階段特有的。您可以變更每個新工作階段的值。此參數的設定可以在尖峰工作負載期間增加資料庫執行個體的記憶體使用率。因此，請考慮根據您應用程式的工作負載模式和資料庫執行個體上的可用記憶體來增加參數值。

- 將 MIXED 的預設值用於 `binlog_format`。根據交易的大小，將 `binlog_format` 設為 ROW 可能會在執行個體儲存體上產生大型 binlog 快取檔案。
- 避免在單一交易中執行大量變更。這些類型的交易可能會在執行個體儲存體上產生大型 binlog 快取檔案，並在執行個體儲存體已滿時產生問題。請考慮將寫入分割成多個小型交易，以將 binlog 快取檔案使用的儲存體減至最少。

使用 RDS Optimized Reads

當您在單一可用區域資料庫執行個體部署或多可用區域資料庫執行個體部署中，佈建具有下列其中一個資料庫執行個體類別的 RDS for MariaDB 資料庫執行個體時，資料庫執行個體會自動使用 RDS Optimized Reads。

若要開啟 RDS Optimized Reads，請執行下列其中一項：

- 使用其中一個資料庫執行個體類別，建立 RDS for MariaDB 資料庫執行個體。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改現有的 RDS for MariaDB 資料庫執行個體，以使用其中一個資料庫執行個體類別。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

RDS 最佳化讀取適用於 AWS 區域 所有支援具有本機 NVMe SSD 儲存體的資料庫執行個體類別。如需資料庫執行個體類別的相關資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

資料庫執行個體類別的可用性與 AWS 區域。若要判斷特定資料庫執行個體類別是否受支援 AWS 區域，請參閱 [the section called “確定資料庫執行個體類別支援 AWS 區域”](#)。

如果您不想要使用 RDS Optimized Reads，請修改資料庫執行個體，以便其不使用支援該功能的資料庫執行個體類別。

監控使用 RDS Optimized Reads 的資料庫執行個體

您可以使用下列 CloudWatch 指標監控使用 RDS 最佳化讀取的資料庫執行個體：

- FreeLocalStorage
- ReadIOPSLocalStorage

- ReadLatencyLocalStorage
- ReadThroughputLocalStorage
- WriteIOPSLocalStorage
- WriteLatencyLocalStorage
- WriteThroughputLocalStorage

這些指標提供可用執行個體儲存體、IOPS 和輸送量的相關資料。如需這些指標的詳細資訊，請參閱 [Amazon RDS 的 CloudWatch Amazon 實例級指標](#)。

RDS Optimized Reads 的限制

下列限制適用於 RDS Optimized Reads：

- 下列 RDS for MariaDB 版本支援 RDS Optimized Reads:
 - 10.11.4 和更高的 10.11 版本
 - 10.6.7 和更新的 10.6 版本
 - 10.5.16 和更新的 10.5 版本
 - 10.4.25 和更新的 10.4 版本

如需 RDS for MariaDB 版本的相關資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

- 您無法在支援 RDS Optimized Reads 的資料庫執行個體類別上，將臨時物件的位置變更為持久性儲存體 (Amazon EBS)。
- 在資料庫執行個體上啟用二進位記錄時，交易大小上限受執行個體儲存體的大小限制。在 MariaDB 中，任何需要儲存體超過 `binlog_cache_size` 寫入交易值的工作階段，都會變更為暫存 binlog 快取檔案，這些檔案是在執行個體儲存體上建立的。
- 執行個體儲存體已滿時，交易會失敗。

使用 Amazon RDS Optimized Writes 改善 MariaDB 的寫入效能

您可以使用 RDS Optimized Writes for MariaDB 來改善寫入交易的效能。當您的 RDS for MariaDB 資料庫使用 RDS Optimized Writes 時，它可以實現高達兩倍的寫入交易輸送量。

主題

- [RDS Optimized Writes 的概觀](#)
- [使用 RDS Optimized Writes](#)
- [在現有資料庫上啟用 RDS Optimized Writes](#)
- [RDS Optimized Writes 的限制](#)

RDS Optimized Writes 的概觀

當您開啟 RDS Optimized Writes 時，您的 RDS for MariaDB 資料庫只會在將資料排清至持久性儲存體時寫入一次，而不需要雙寫緩衝區。資料庫會繼續為可靠的資料庫交易提供 ACID 屬性保護，以及改善的效能。

關聯式資料庫 (例如 MariaDB) 為可靠的資料庫交易提供不可分割性、一致性、隔離和耐久性的 ACID 屬性。為了協助提供這些屬性，MariaDB 會使用稱為雙寫緩衝區的資料儲存區域，防止部分頁面寫入錯誤。當資料庫正在更新頁面時發生硬體故障 (例如停電) 時，就會發生這些錯誤。MariaDB 資料庫可以偵測部分頁面寫入，並使用雙寫緩衝區中的頁面複本進行復原。雖然這項技術可提供保護，但也會產生額外的寫入操作。如需 MariaDB 雙寫緩衝區的詳細資訊，請參閱 MariaDB 文件中的 [雙寫緩衝區](#)。

開啟 RDS Optimized Reads 後，您的 RDS for MariaDB 資料庫只會在將資料排清至持久性儲存體時寫入一次，而不需使用雙寫緩衝區。如果您在 RDS for MariaDB 資料庫上執行大量寫入工作負載，則 RDS Optimized Reads 很有用。具有大量寫入工作負載的資料庫範例包括支援數位支付、金融交易和遊戲應用程式的資料庫。

這些資料庫會在使用 AWS Nitro System 的資料庫執行個體類別上執行。由於這些系統中的硬體組態，資料庫可以在一個步驟中可靠且持久地將 16 KiB 頁面直接寫入至資料檔案。AWS Nitro System 讓 RDS Optimized Writes 變成可行。

您可以設定新的資料庫參數 `rds.optimized_writes`，以控制 RDS for MariaDB 資料庫的 RDS Optimized Writes 功能。在下列版本的 RDS for MariaDB 的資料庫參數群組中存取此參數：

- 10.11.4 和更高的 10.11 版本
- 10.6.10 和更高的 10.6 版本

請使用下列值設定參數：

- AUTO – 開啟 RDS Optimized Writes (如果資料庫支援它)。關閉 RDS Optimized Writes (如果資料庫不支援它)。此設定是預設值。
- OFF – 關閉 RDS Optimized Writes，即使資料庫支援它也一樣。

如果您將設定為使用 RDS Optimized Writes 的 RDS for MariaDB 資料庫遷移至不支援此功能的資料庫執行個體類別，RDS 會自動關閉資料庫的 RDS Optimized Writes。

當 RDS Optimized Writes 關閉時，資料庫會使用 MariaDB 雙寫緩衝區。

若要判斷 RDS for MariaDB 資料庫是否使用 RDS Optimized Writes，請檢視資料庫 `innodb_doublewrite` 參數目前的值。如果資料庫使用 RDS Optimized Writes，則此參數會設為 FALSE (0)。

使用 RDS Optimized Writes

當您使用 RDS 主控台、AWS CLI 或 RDS API，建立 RDS for MariaDB 資料庫時，您可以開啟 RDS Optimized Writes。在資料庫建立期間滿足下列兩個條件時，RDS Optimized Writes 會自動開啟：

- 您指定一個支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別。
 - 下列 RDS for MariaDB 版本支援 RDS Optimized Writes：
 - 10.11.4 和更高的 10.11 版本
 - 10.6.10 和更高的 10.6 版本

如需 RDS for MariaDB 版本的相關資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

- 使用下列資料庫執行個體類別的 RDS for MariaDB 資料庫支援 RDS Optimized Writes：
 - db.m7g
 - db.m6g
 - DB.m6gd
 - db.m6i
 - db.m5
 - db.m5d
 - db.r7g
 - db.r6g
 - db.r6gd

- db.r6i
- db.r5
- db.r5b
- db.r5d
- db.x2idn
- db.x2iedn

如需資料庫執行個體類別的相關資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

資料庫執行個體類別的可用性因 AWS 區域而異。若要判斷特定的 AWS 區域 是否支援資料庫執行個體類別，請參閱 [the section called “確定資料庫執行個體類別支援 AWS 區域”](#)。

- 在與資料庫相關聯的參數群組中，`rds.optimized_writes` 參數設為 `AUTO`。在預設參數群組中，此參數一律設為 `AUTO`。

如果您想要使用支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別，但不想要使用此功能，則請在建立資料庫時指定自訂參數群組。在此參數群組中，將 `rds.optimized_writes` 參數設為 `OFF`。如果您想要資料庫稍後使用 RDS Optimized Writes，則可以將參數設為 `AUTO` 將其開啟。如需建立自訂參數群組和設定參數的相關資訊，請參閱 [使用參數群組](#)。

如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

主控台

使用 RDS 主控台建立 RDS for MariaDB 資料庫時，您可以篩選支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別。開啟篩選條件後，您可以從可用的資料庫引擎版本和資料庫執行個體類別中進行選擇。

若要選擇支援 RDS 最佳化寫入的資料庫引擎版本，請在引擎版本中篩選支援它的 RDS for MariaDB 資料庫引擎版本，然後選擇版本。

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



MariaDB



PostgreSQL



Oracle



Microsoft SQL Server



IBM Db2



Engine version [Info](#)

View the engine versions that support the following database features.

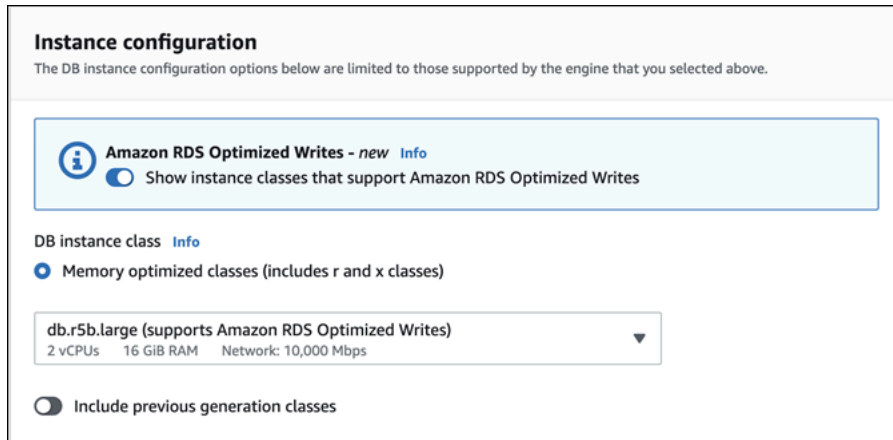
▼ Hide filters

Show versions that support the Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MariaDB 10.6.10

在 Instance configuration (執行個體組態) 區段中，篩選支援 RDS Optimized Writes 的資料庫執行個體類別，然後選擇資料庫執行個體類別。



在做出這些選擇之後，您可以選擇其他符合您需求的設定，並使用主控台完成建立 RDS for MariaDB 資料庫。

AWS CLI

若要使用建立資料庫執行個體AWS CLI，請使用[create-db-instance](#)指令。確定 `--engine-version` 和 `--db-instance-class` 值支援 RDS Optimized Writes。此外，確定與資料庫執行個體相關聯的參數群組已將 `rds.optimized_writes` 參數設為 `AUTO`：此範例會將預設參數群組與資料庫執行個體建立關聯。

Example 建立使用 RDS Optimized Writes 的資料庫執行個體

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --engine mariadb \  
  --engine-version 10.6.10 \  
  --db-instance-class db.r5b.large \  
  --manage-master-user-password \  
  --master-username admin \  
  --allocated-storage 200
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --engine mariadb ^
```

```
--engine-version 10.6.10 ^  
--db-instance-class db.r5b.large ^  
--manage-master-user-password ^  
--master-username admin ^  
--allocated-storage 200
```

RDS API

您可以使用 [CreateDBInstance](#) 操作，建立資料庫執行個體。使用此操作時，確定 `EngineVersion` 和 `DBInstanceClass` 值支援 RDS Optimized Writes。此外，確定與資料庫執行個體相關聯的參數群組已將 `rds.optimized_writes` 參數設為 `AUTO`：

在現有資料庫上啟用 RDS Optimized Writes

為了修改現有的 RDS for MariaDB，以開啟 RDS Optimized Writes，必須已使用支援的資料庫引擎版本和資料庫執行個體類別建立資料庫。此外，必須在 2023 年 3 月 7 日發行 RDS Optimized Writes 之後建立了資料庫，因為所需的基礎檔案系統組態與發行之之前所建立之資料庫的組態不相容。如果符合這些條件，您可以透過將 `rds.optimized_writes` 參數設定為 `AUTO` 來開啟 RDS Optimized Writes。

如果未使用支援的引擎版本、執行個體類別或檔案系統組態，來建立您的資料庫，您可以使用 RDS 藍/綠部署遷移至支援的組態。建立藍/綠部署時，請執行以下動作：

- 選取在綠色資料庫上啟用 Optimized Writes，然後指定支援 RDS Optimized Writes 的引擎版本和資料庫執行個體。如需受支援引擎版本和執行個體類別的清單，請參閱 [the section called “搭配新的資料庫使用”](#)。
- 在儲存體下，選擇升級儲存體檔案系統組態。此選項會將資料庫升級為相容的基礎檔案系統組態。

建立藍/綠部署時，如果將 `rds.optimized_writes` 參數設定為 `AUTO`，則會在綠色環境上自動啟用 RDS Optimized Writes。然後，您可以轉換藍/綠部署，將綠色環境提升至新的生產環境。

如需詳細資訊，請參閱 [the section called “建立藍/綠部署”](#)。

RDS Optimized Writes 的限制

當您從快照還原 RDS for MariaDB 資料庫時，只有在下列所有條件都滿足時，才能開啟資料庫的 RDS Optimized Writes：

- 快照集是從支援 RDS Optimized Writes 的資料庫建立的。
- 快照是從發佈 RDS Optimized Writes 之後所建立的資料庫中建立的。

- 快照還原至支援 RDS Optimized Writes 的資料庫。
- 還原的資料庫與將 `rds.optimized_writes` 參數設為 `AUTO` 的參數群組相關聯。

升級 MariaDB 資料庫引擎

當 Amazon RDS 支援新版本的資料庫引擎時，您可以將資料庫執行個體升級為新版本。MariaDB 資料庫執行個體有兩種升級方式：主要版本升級和次要版本升級。

主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。因此，您必須手動執行資料庫執行個體的主要版本升級。您可以修改資料庫執行個體，啟動主要版本升級。然而在您執行主要的版本升級之前，建議您遵循 [MariaDB 的主要版本升級](#) 中描述的指示。

反之，次要版本升級只包含與現有應用程式回溯相容的變更。您可以修改資料庫執行個體，手動啟動次要版本升級。或者，您也可以在建​​立或修改資料庫執行個體時，啟用 Auto minor version upgrade (自動次要版本升級) 選項。這麼做代表 Amazon RDS 測試與核准新版本後，您的資料庫執行個體將會自動升級。如需執行升級的相關資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

如果 MariaDB 資料庫執行個體使用僅供讀取複本，則升級來源執行個體之前，您必須升級所有僅供讀取複本。如果資料庫執行個體位於異地同步備份部署中，則寫入器和待命複本都會升級。在升級完成之前，您的資料庫執行個體可能無法使用。

如需 MariaDB 支援版本和版本管理的詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

資料庫引擎升級需要停機。停機持續時間依資料庫執行個體大小而有所不同。

Tip

您可以使用藍/綠部署，將資料庫執行個體升級所需的停機時間降至最低。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

主題

- [升級概觀](#)
- [MariaDB 版本號](#)
- [RDS 版本編號](#)
- [MariaDB 的主要版本升級](#)
- [升級 MariaDB 資料庫執行個體](#)
- [MariaDB 的自動次要版本升級](#)
- [使用僅供讀取複本在升級 MariaDB 資料庫時減少停機時間](#)

升級概觀

使用升級資料庫執行個體時，它會顯示資料庫執行個體的有效升級目標。AWS Management Console 您也可以使用下列 AWS CLI 命令來識別資料庫執行個體的有效升級目標：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine mariadb \  
  --engine-version version-number \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --engine mariadb ^  
  --engine-version version-number ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

例如，若要識別 MariaDB 10.5.17 版資料庫執行個體的有效升級目標，請執行下列命令：AWS CLI

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine mariadb \  
  --engine-version 10.5.17 \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --engine mariadb ^  
  --engine-version 10.5.17 ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

Amazon RDS 在升級過程中會建立兩個以上的資料庫快照。Amazon RDS 在進行任何升級變更之前，最多會建立兩個資料庫執行個體快照。如果升級對資料庫沒有用處，您可以還原其中一個快照，以建立

執行舊版本的資料庫執行個體。升級完成時，Amazon RDS 會建立資料庫執行個體的另一個快照。無論是否 AWS Backup 管理資料庫執行個體的備份，Amazon RDS 都會拍攝這些快照。

Note

只在您將資料庫執行個體的備份保留期設為大於 0 的數字時，Amazon RDS 才會建立資料庫快照。若要變更備份保留期，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

升級完成之後，您就無法回復至前一版的資料庫引擎。如果您想要回到前一版，請還原第一個建立的資料庫快照，以建立新的資料庫執行個體。

您可以控制何時將資料庫執行個體升級至 Amazon RDS 支援的新版本。這一層控制協助您維持與特定資料庫版本的相容性，並且先在您的應用程式上測試新版本，再部署於生產環境。準備好時，您就可以在最適合排程的時間執行版本升級。

如果資料庫執行個體使用讀取複寫，則在升級來源執行個體之前，您必須升級所有僅供讀取複本。

如果資料庫執行個體位於異地同步備份部署中，則主要和待命資料庫執行個體都會升級。主要和待命資料庫執行個體會同時升級，在升級完成之前會經歷停機情況。停機時間依資料庫引擎、引擎版本和資料庫執行個體大小而有所不同。

MariaDB 版本號

適用於 MariaDB 資料庫引擎的 RDS 版本編號順序採用主要次要修補程式的形式為年月修補程式，例如 10.11.5.R2.20231201 或 10.4.30。使用的格式取決於 MariaDB 引擎版本。

重大

主要版本號碼既是整數，也是版本號碼的第一個小數部分，例如 10.11。主要版本升級會增加版本號碼的主要部分。例如，從 10.5.20 升級到 10.6.12 是主要的版本升級，其中 10.5 和 10.6 是主要版本號碼。

未成年

次要版本號碼是版本號碼的第三部分，例如 10.11.5 中的 5。

補丁

修補程式是版本號碼的第四部分，例如，10.11.5.R2 中的 R2。RDS 修補程式版本中包含發行後新增到次要版本的重要錯誤修正。

年月

日期是版本號碼的第五部分，例如，在十一月五日中的 20231201。RDS 日期版本是一項安全性修補程式，其中包含在發行後新增至次要版本的重要安全性修正程式。它不包括任何可能改變引擎行為的修復程序。

主要版本	次要版本	命名方案
10.11	5 歲以上	<p>新的資料庫執行個體使用主要. 次要修補程式，例如 10.11.5.R2.20231201。</p> <p>現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 10.11.5.R2，直到您下次的主要或次要版本升級為止。</p>
	< 5	現有的資料庫執行個體使用主要的 .min. 修補程式，例如 10.11. 4.R2。
10.6	≥14	<p>新的資料庫執行個體會使用主要的. 次要修補程式，例如 10.6.14.R 2.20231201。</p> <p>現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 10.6.14.R2，直到您下次的主要或次要版本升級為止。</p>
	<	現有的資料庫執行個體使用主要的 .min. 修補程式，例如 10.6. 13.R2。
10.5	≥	<p>新的資料庫執行個體使用主要. 次要修補程式，例如 10.5.21.R2.20231201。</p> <p>現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 10.5.21.R2，直到您下次的主要或次要版本升級為止。</p>
	<	現有的資料庫執行個體使用主要的 .min. 修補程式，例如 10.5. 20.R2。

主要版本	次要版本	命名方案
10.4	三十歲以上	新的資料庫執行個體使用主要. 次要修補程式，例如 10.4.30.R2.20231201。 現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 10.4.30.R2，直到您下一次的主要或次要版本升級為止。
	< 30	現有的資料庫執行個體使用主要的 .min. 修補程式，例如 10.4. 29.R2。

RDS 版本編號

RDS 版本號碼使用 *major.minor.patch* 或 *major.minor.patch.YYYYMMDD* 命名配置。RDS 修補程式版本中包含發行後新增到次要版本的重要錯誤修正。RDS 日期版本 (*YYYYMMDD*) 是一個安全性修補程式。安全性修補程式不包含任何可能會變更引擎行為的修正程式。

若要識別資料庫的 Amazon RDS 版本編號，您必須先使用下列命令建立 `rds_tools` 延伸模組：

```
CREATE EXTENSION rds_tools;
```

您可以使用以下 SQL 查詢找出您的 RDS 對於 MariaDB 資料庫的 RDS 版本號碼：

```
mysql> select mysql.rds_version();
```

例如，查詢 RDS 為 MariaDB 10.6.14 資料庫會傳回下列輸出：

```
+-----+
| mysql.rds_version() |
+-----+
| 10.6.14.R2.20231201 |
+-----+
1 row in set (0.01 sec)
```


MariaDB 的主要版本升級

主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。因此，Amazon RDS 不會自動套用主要版本升級。您必須手動修改資料庫執行個體。我們建議您在將任何升級套用至生產執行個體之前，先進行完整測試。

對於 MariaDB 資料庫引擎的主要版本，Amazon RDS 支援以下的就地升級：

- 從任何 MariaDB 版本升級至 MariaDB 10.11
- 從任何 MariaDB 版本升級至 MariaDB 10.6
- MariaDB 10.4 至 MariaDB 10.5
- MariaDB 10.3 至 MariaDB 10.4

若要將主要版本升級到低於 10.6 版的 MariaDB，請依序升級到每個主版本。例如，若要從 10.3 版升級至 10.5 版，請依下列順序升級：10.3 到 10.4，再從 10.4 到 10.5。

如果您是使用自訂參數群組，而且欲執行主要版本升級，則必須對新的資料庫引擎版本指定預設參數群組，或對新的資料庫引擎版本建立您自己的自訂參數群組。讓新的參數群組與資料庫執行個體產生關聯，需要客戶啟動的資料庫在升級完成時重新啟動。如果需要重新啟動執行個體，才能套用參數群組變更，執行個體的參數群組將顯示 pending-reboot。可在 AWS Management Console 或使用「describe」呼叫 (例如 describe-db-instances) 檢視執行個體的參數群組。

升級 MariaDB 資料庫執行個體

如需有關手動或自動升級 MariaDB 資料庫執行個體的資訊，請參閱[升級資料庫執行個體引擎版本](#)。

MariaDB 的自動次要版本升級

如果您在建立或修改資料庫執行個體時指定下列設定，則可以讓資料庫執行個體自動升級。

- 已啟用 Auto minor version upgrade (自動次要版本升級) 設定。
- Backup retention period (備份保留期間) 設定大於 0。

在中 AWS Management Console，這些設定位於 [其他組態] 下。下圖顯示 Auto minor version upgrade (自動次要版本升級) 設定。

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

Select window

No preference

Start day **Start time** **Duration**

Monday ▼ 00 ▼ : 00 ▼ UTC 0.5 ▼ hours

如需這些設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

對於某些 RDS for MariaDB 主要版本 AWS 區域，RDS 會將一個次要版本指定為自動升級版本。次要版本經過測試，並由 Amazon RDS 核准之後，在您的維護時段期間，會自動發生次要版本升級。RDS 不會自動將較新發行的次要版本設定為自動升級版本。在 RDS 指派較新的自動升級版本之前，會考慮數個準則，例如下列：

- 已知安全性問題
- MariaDB 社群版本中的錯誤
- 從發行次要版本以來的整體機群穩定性

Note

自 MariaDB 的特定次要版本開始，已移除對使用 TLS 1.0 和 1.1 版的支援。如需支援的 MariaDB 次要版本的相關資訊，請參閱 [the section called “SSL/TLS 支援”](#)

您可以使用以下 AWS CLI 命令來確定特定的指定 MariaDB 次要版本的當前自動次要升級目標版本。
AWS 區域

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \
```

```
--engine mariadb \  
--engine-version minor-version \  
--region region \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
--engine mariadb ^  
--engine-version minor-version ^  
--region region ^  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^  
--output text
```

例如，以下 AWS CLI 命令確定美國東部（俄亥俄州）（美國東部 -2）AWS 區域 MariaDB 次要版本 10.5.16 的自動次要升級目標。

對於 Linux/macOS、或 Unix：

```
aws rds describe-db-engine-versions \  
--engine mariadb \  
--engine-version 10.5.16 \  
--region us-east-2 \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output table
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
--engine mariadb ^  
--engine-version 10.5.16 ^  
--region us-east-2 ^  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^  
--output table
```

輸出類似如下。

```

-----
| DescribeDBEngineVersions |
+-----+-----+
| AutoUpgrade | EngineVersion |
+-----+-----+
| True      | 10.5.17    |
| False       | 10.5.18      |
| False       | 10.5.19      |
| False       | 10.6.5       |
| False       | 10.6.7       |
| False       | 10.6.8       |
| False       | 10.6.10      |
| False       | 10.6.11      |
| False       | 10.6.12      |
+-----+-----+

```

在此範例中，AutoUpgrade 值是 True (若為 MariaDB 10.5.17 版)。因此，自動次要升級目標是 MariaDB 10.5.17 版，其已在輸出中反白顯示。

如果符合下列準則，在維護時段期間會自動升級 MariaDB 資料庫執行個體：

- 已啟用 Auto minor version upgrade (自動次要版本升級) 設定。
- Backup retention period (備份保留期間) 設定大於 0。
- 資料庫執行個體執行的次要資料庫引擎版本小於目前的自動升級次要版本。

如需詳細資訊，請參閱 [自動升級次要引擎版本](#)。

使用僅供讀取複本在升級 MariaDB 資料庫時減少停機時間

在大多數情況下，藍/綠部署是升級 MariaDB 資料庫執行個體時減少停機時間的最佳選項。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

如果您無法使用藍/綠部署，且您的 MariaDB 資料庫執行個體目前與生產應用程式搭配使用，您可以使用下列程序來升級資料庫執行個體的資料庫版本。此程序可降低應用程式的停機時間。

透過使用僅供讀取複本，您可以提前執行大部分的維護步驟，並將實際中斷期間的必要變更降到最低。使用此技術，您可以測試和準備新的資料庫執行個體，而無需對現有的資料庫執行個體進行任何變更。

下列程序顯示從 MariaDB 10.5 版升級至 MariaDB 10.6 版的範例。您可以使用同樣的一般步驟升級到其他的主要版本。


在資料庫執行個體處於使用中的情況下升級 MariaDB 資料庫

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 建立 MariaDB 10.5 資料庫執行個體的僅供讀取複本。此程序建立資料庫的可升級副本。資料庫執行個體的其他僅供讀取複本也可能存在。
 - a. 在主控台中，選擇 Databases (資料庫)，然後選擇您要升級的資料庫執行個體。
 - b. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
 - c. 在僅供讀取複本的 DB instance identifier (資料庫執行個體識別符) 中提供一個值，並確定 DB instance class (資料庫執行個體類別) 和其他設定符合您的 MariaDB 10.5 資料庫執行個體。
 - d. 選擇 Create read replica (建立僅供讀取複本)。
3. (選擇性) 當僅供讀取複本已建立且 Status (狀態) 顯示 Available (可用) 時，請將僅供讀取複本轉換為異地同步備份部署並啟用備份。

依預設，僅供讀取複本會建立為單一可用區域部署，並停用備份。由於僅供讀取複本最終會成為生產資料庫執行個體，因此最佳實務是設定多可用區部署並立即啟用備份。

- a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛建立的僅供讀取複本。
 - b. 選擇 Modify (修改)。
 - c. 對於 Multi-AZ deployment (異地同步備份部署)，請選擇 Create a standby instance (建立待命執行個體)。
 - d. 對於 Backup Retention Period (備份保留期)，選擇非零正值 (例如 3 天)，然後選擇 Continue (繼續)。
 - e. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
 - f. 選擇 Modify DB instance (修改資料庫執行個體)。
4. 僅供讀取複本的 Status (狀態) 顯示為 Available (可用) 時，請將僅供讀取複本升級至 MariaDB 10.6。
 - a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛建立的僅供讀取複本。
 - b. 選擇 Modify (修改)。
 - c. 在 DB engine version (資料庫引擎版本) 中，選擇升級到 MariaDB 10.6 版，然後選擇 Continue (繼續)。
 - d. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
 - e. 選擇 Modify DB instance (修改資料庫執行個體) 以開始升級。

- 升級完成且狀態顯示 [可用] 時，請確認已升級的僅供讀取複本 up-to-date 與來源 MariaDB 10.5 資料庫執行個體一起使用。若要驗證，請連線至僅供讀取複本並執行 SHOW REPLICA STATUS 命令。如果 Seconds_Behind_Master 欄位是 0，則複寫為 up-to-date。

 Note

MariaDB 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MariaDB 是 10.6 之前的版本，請使用 SHOW SLAVE STATUS。

- (選擇性) 建立僅供讀取複本的僅供讀取複本。

如果您希望資料庫執行個體在提升為獨立資料庫執行個體後擁有僅供讀取複本，您可以立即建立僅供讀取複本。

- 在主控台中，選擇 Databases (資料庫)，然後選擇您剛升級的僅供讀取複本。
- 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
- 在僅供讀取複本的 DB instance identifier (資料庫執行個體識別符) 中提供一個值，並確定 DB instance class (資料庫執行個體類別) 和其他設定符合您的 MariaDB 10.5 資料庫執行個體。
- 選擇 Create read replica (建立僅供讀取複本)。

- (選擇性) 為僅供讀取複本設定自訂資料庫參數群組。

如果您希望資料庫執行個體在提升為獨立資料庫執行個體後使用自訂參數群組，您現在可以建立資料庫參數群組，並將其與僅供讀取複本建立關聯。

- 建立 MariaDB 10.6 的自訂資料庫參數群組。如需說明，請參閱「[建立資料庫參數群組](#)」。
- 修改您想要在剛才建立的資料庫參數群組中變更的參數。如需說明，請參閱「[修改資料庫參數群組中的參數](#)」。
- 在主控台中，選擇 Databases (資料庫)，然後選擇僅供讀取複本。
- 選擇 Modify (修改)。
- 對於 DB parameter group (資料庫參數群組)，請選擇您剛才建立的 MariaDB 10.6 資料庫參數群組，然後選擇 Continue (繼續)。
- 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
- 選擇 Modify DB instance (修改資料庫執行個體) 以開始升級。

- 將 MariaDB 10.6 僅供讀取複本變成獨立資料庫執行個體。

⚠ Important

當 MariaDB 10.6 僅供讀取複本提升為獨立資料庫執行個體時，就不再是 MariaDB 10.5 資料庫執行個體的複本。建議在維護時段提升 MariaDB 10.6 僅供讀取複本，因為此時來源 MariaDB 10.5 資料庫執行個體處於唯讀模式，且所有寫入操作都已暫停。提升完成時，您可以將寫入操作導向已升級的 MariaDB 10.6 資料庫執行個體，以確保不會遺漏任何寫入操作。

此外，在提升 MariaDB 10.6 僅供讀取複本之前，建議您在 MariaDB 10.6 僅供讀取複本上執行所有必要的資料定義語言 (DDL) 操作。範例是建立索引。此方法可避免 MariaDB 10.6 僅供讀取複本在提升之後效能降低。若要提升僅供讀取複本，請使用以下程序。

- a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛升級的僅供讀取複本。
 - b. 針對 Actions (動作)，選擇 Promote (提升)。
 - c. 選擇 Yes (是)，對僅供讀取複本執行個體啟用自動備份。如需詳細資訊，請參閱 [備份簡介](#)。
 - d. 選擇 Continue (繼續)。
 - e. 選擇 Promote Read Replica (提升僅供讀取複本)。
9. 您現在已升級 MariaDB 資料庫的版本。此時，您可以將應用程式導向至新的 MariaDB 10.6 資料庫執行個體。

將資料匯入 MariaDB 資料庫執行個體

您可使用多種不同技術，將資料匯入 RDS for MariaDB 資料庫執行個體。最理想的做法需視資料來源、資料量，以及作業是屬於一次性或持續性匯入等條件而定。如要連同資料一起遷移應用程式，也需考量您願意承受的停機時間。

將資料匯入 RDS for MariaDB 資料庫執行個體的各種技術如下表所示。

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
現有的 MariaDB 資料庫執行個體	任何	一次性或持續性	極小	為持續複寫建立僅供讀取複本。升級僅供讀取複本，以便一次性建立新的資料庫執行個體。	使用資料庫執行個體僅供讀取複本
現有的 MariaDB 或 MySQL 資料庫	小型	一次性	一些	使用命令列公用程式，直接將資料複製到您的 MySQL 資料庫執行個體。	將資料從 MariaDB 或 MySQL 資料庫匯入 MariaDB 或 MySQL 資料庫執行個體
未存放在現有資料庫的資料	中型	一次性	一些	創建平面文件並使用 MySQL LOAD DATA LOCAL INFILE 語句導入它們。	匯入任何 MariaDB 或 MySQL 資料庫

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
					執行個體來源的資料
內部部署或 Amazon EC2 現有的 MariaDB 或 MySQL 資料庫	任何	持續性	極小	<p>使用現有 MariaDB 或 MySQL 資料庫為複寫來源來設定複寫。</p> <p>若要設定複寫至 MariaDB 資料庫執行個體：當外部執行個體是 MariaDB 10.0.24 版或更新版本時，您可以使用 MariaDB 全域交易識別符 (GTID)；當執行個體是 10.0.24 以前的版本時，您可以使用 MySQL 執行個體或 MariaDB 執行個體的二進位日誌座標。MariaDB GTID 的實作不同於 MySQL GTID (在 Amazon RDS 中不支援)。</p>	使用外部來源執行個體設定二進位日誌檔案位置複寫 減少將資料匯入 Amazon RDS MariaDB 或 MySQL 資料庫執行個體時的停機時間

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
任何現有資料庫	任何	一次性或持續性	極小	用於 AWS Database Migration Service 以最短的停機時間移轉資料庫，並且對於許多資料庫資料引擎，繼續進行中的複寫。	《AWS Database Migration Service 使用者指南》中的 什麼是 AWS Database Migration Service 和使用與 MySQL 相容的資料庫作為 AWS DMS 的目標

Note

mysql 系統資料庫包含登入資料庫執行個體與存取資料所需的身分驗證和授權資訊。捨棄、更改、重新命名或截斷資料庫執行個體中 mysql 資料庫的資料表、資料或其他內容，可能會導致錯誤發生，造成無法存取資料庫執行個體與資料。如果發生這種情況，可以使用快照從快照還原資料庫執行個體，AWS CLI [restore-db-instance-from-db-snapshot](#) 或使用 [restore-db-instance-to-point-in-time](#) 指令復原。

將資料從 MariaDB 或 MySQL 資料庫匯入 MariaDB 或 MySQL 資料庫執行個體

您也可將資料從現有的 MariaDB 或 MySQL 資料庫匯入 MySQL 或 MariaDB 資料庫執行個體。方法是以 [mysqldump](#) 命令複製資料庫，然後直接傳輸到 MariaDB 或 MySQL 資料庫執行個體。`mysqldump` 命令列公用程式普遍用於製作備份，以及將資料從 MariaDB 或 MySQL 伺服器傳輸到另一個伺服器上。MySQL 和 MariaDB 用戶端軟體皆隨附這個程式。

Note

如果您使用 MySQL 資料庫執行個體匯入或匯出大量資料，使用 `xtrabackup` 備份檔案和 Amazon Amazon S3 將資料移入和移出 Amazon RDS 會更加可靠且更快速。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

典型的 `mysqldump` 命令會將資料從外部資料庫移入 Amazon RDS 資料庫執行個體，範例如下。

```
mysqldump -u local_user \  
  --databases database_name \  
  --single-transaction \  
  --compress \  
  --order-by-primary \  
-plocal_password | mysql -u RDS_user \  
  --port=port_number \  
  --host=host_name \  
-pRDS_password
```

Important

切勿在 `-p` 選項與輸入的密碼間插入空白。
指定此處所顯示提示以外的憑證，作為安全最佳實務。

請確認您了解下列建議和注意事項：

- 從傾印檔案中排除下列結構描述：`sys`、`performance_schema` 和 `information_schema`。`mysqldump` 公用程式已預設排除這些結構描述。
- 如果您需要移轉使用者和權限，請考慮使用可產生資料控制語言 (DCL) 的工具來重新建立使用者和權限，例如公用程式。 [pt-show-grants](#)

- 若要執行匯入，請確認執行此程序的使用者能夠存取資料庫執行個體。如需詳細資訊，請參閱 [使用安全群組控制存取](#)。

使用的參數如下：

- `-u local_user` – 用來指定使用者名稱。首次使用此參數時，需在 `--databases` 參數所識別的本機 MariaDB 或 MySQL 資料庫中指定使用者帳戶名稱。
- `--databases database_name` – 在您要匯入至 Amazon RDS 的本機 MariaDB 或 MySQL 執行個體上，指定資料庫名稱。
- `--single-transaction` – 確保從本機資料庫負載的所有資料均與單一時間點一致。如有其他程序在 `mysqldump` 讀取資料時變更資料，使用此參數有助於維持資料完整性。
- `--compress` – 在本機資料庫的資料傳送到 Amazon RDS 前先完成壓縮，以減少耗用的網路頻寬。
- `--order-by-primary` – 以資料的主索引鍵排序各資料表的資料，以減少載入時間。
- `-plocal_password` – 用來指定密碼。首次使用此參數時，需指定第一個 `-u` 參數所識別的使用者帳戶密碼。
- `-u RDS_user` – 用來指定使用者名稱。第二次使用此參數時，需在 `--host` 參數所識別的 MariaDB 或 MySQL 資料庫執行個體的預設資料庫上，指定使用者帳戶名稱。
- `--port port_number` – 用於指定 MariaDB 或 MySQL 資料庫執行個體的連接埠。埠號預設為 3306，除非您在建立執行個體時另行變更。
- `--host host_name` – 用來指定 Amazon RDS 資料庫執行個體端點的網域名稱系統 (DNS) 名稱，例如 `myinstance.123456789012.us-east-1.rds.amazonaws.com`。端點值可在 Amazon RDS 管理主控台的執行個體詳細資訊中找到。
- `-pRDS_password` – 用來指定密碼。第二次使用此參數時，需指定第二個 `-u` 參數所識別的使用者帳戶密碼。

必須在 Amazon RDS 資料庫中手動建立預存程序、觸發程序、函數或事件。如果您正在複製的資料庫中存有上述物件，則請在執行 `mysqldump` 時排除這些物件。為此，請在 `mysqldump` 命令中包含以下參數：`--routines=0 --triggers=0 --events=0`。

下列範例會將本機主機上的 `world` 範例資料庫複製到 MySQL 資料庫執行個體。

對於 Linux、macOS、或 Unix：

```
sudo mysqldump -u localuser \
```

```
--databases world \  
--single-transaction \  
--compress \  
--order-by-primary \  
--routines=0 \  
--triggers=0 \  
--events=0 \  
-plocalpassword | mysql -u rdsuser \  
  --port=3306 \  
  --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \  
  -prdspassword
```

若為 Windows 系統，請在命令提示字元中執行下列命令，方法是對 Windows 程式選單中的命令提示字元按一下滑鼠右鍵，然後選擇以系統管理員身分執行：

```
mysqldump -u localuser ^  
  --databases world ^  
  --single-transaction ^  
  --compress ^  
  --order-by-primary ^  
  --routines=0 ^  
  --triggers=0 ^  
  --events=0 ^  
-plocalpassword | mysql -u rdsuser ^  
  --port=3306 ^  
  --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^  
  -prdspassword
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

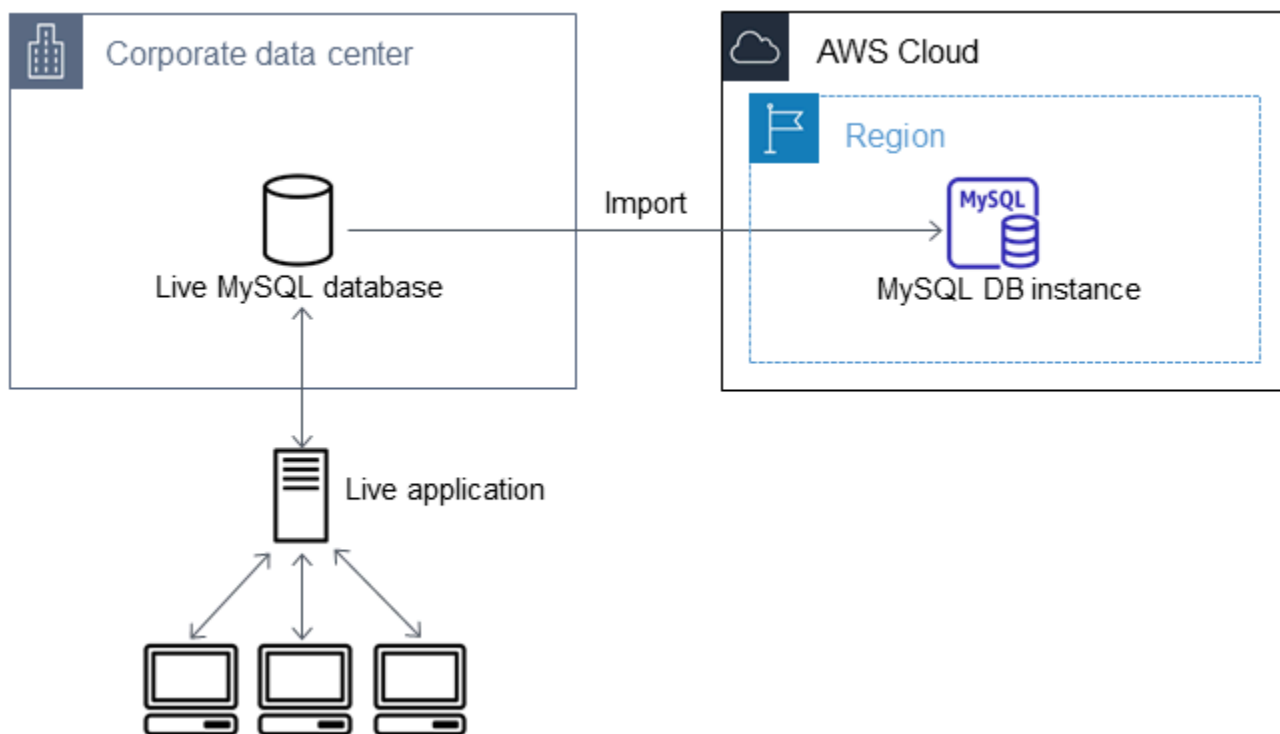
減少將資料匯入 Amazon RDS MariaDB 或 MySQL 資料庫執行個體時的停機時間

某些情況下，您可能需要將資料從支援即時應用程式的外部 MariaDB 或 MySQL 資料庫，匯入至 MariaDB 資料庫執行個體、MySQL 資料庫執行個體或 MySQL 多可用區域資料庫叢集。使用下列程序，將對應用程式可用性的影響降至最低。如果您使用的是超大型資料庫，也可使用這個程序。使用此程序，您可以減少透過網路傳送的資料量，以降低匯入成本 AWS。

此程序中，您需將資料庫資料的複本傳輸至 Amazon EC2 執行個體，並將資料匯入至新的 Amazon RDS 資料庫。然後，您可以使用複寫將 Amazon RDS 資料庫 up-to-date 與即時外部執行個體一起使用，然後再將應用程式重新導向至 Amazon RDS 資料庫。如果外部執行個體版本是 MariaDB 10.0.24 或更高版本，且目標執行個體為 RDS for MariaDB 時，請根據全域交易識別符 (GTID) 設定 MariaDB 複寫。若為其他情況，請根據二進位日誌座標設定複寫。建議 GTID 型複寫 (如果您的外部資料庫可以支援它)，因為 GTID 型複寫是更可信賴的方法。如需詳細資訊，請參閱 MariaDB 文件中的 [Global transaction ID](#) 一節。

Note

如果您想要將資料匯入至 MySQL 資料庫執行個體，且您的情況允許，建議您使用備份檔案與 Amazon S3 將資料移入和移出 Amazon RDS。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

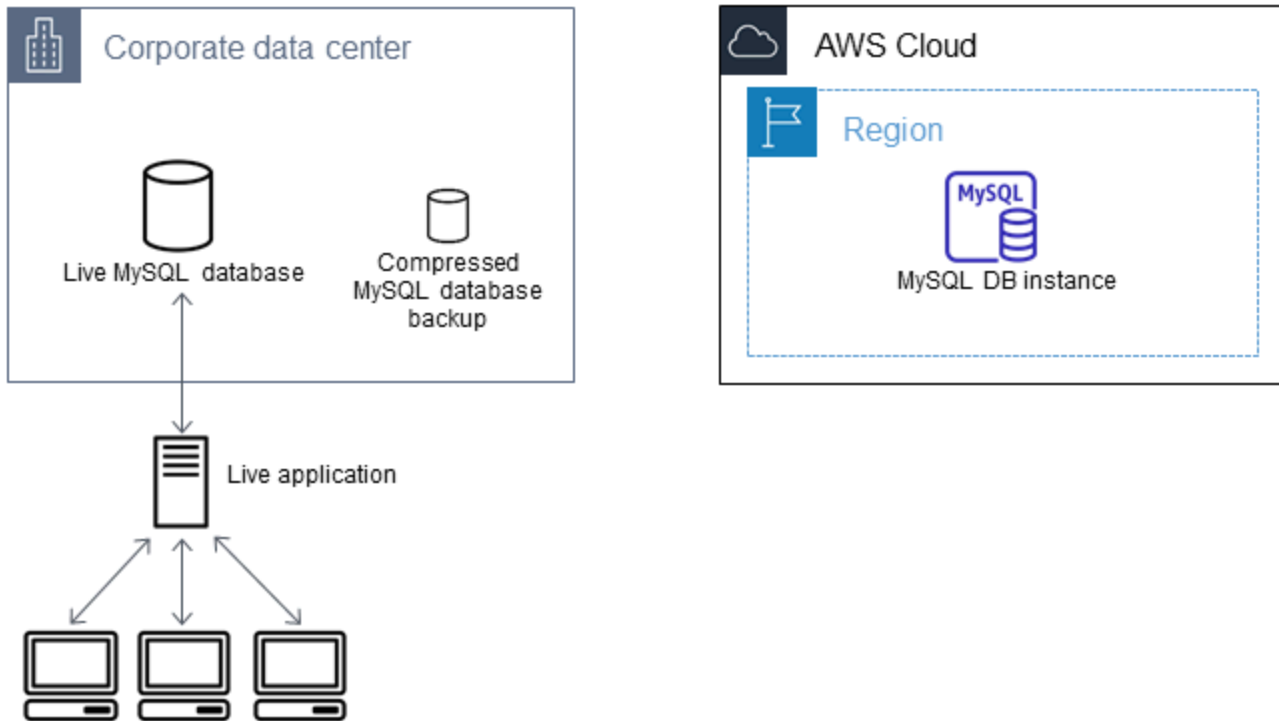


Note

由於可能發生複寫問題，我們不建議您將此程序用於比 MySQL 5.5 更早版本的來源 MySQL 資料庫。如需詳細資訊，請參閱 MySQL 文件中的 [Replication compatibility between MySQL versions](#) 一節。

建立現有資料庫的複本

要將大量資料遷移至 RDS for MariaDB 或 RDS for MySQL 資料庫，並將停機時間降至最低，第一步是建立來源資料的複本。



您可使用 `mysqldump` 公用程式，以 SQL 或分隔符號文字等任一格式建立資料庫備份。建議您在非生產環境中分別測試各種格式，以了解哪種方法能將 `mysqldump` 命令的執行時間減至最少。

亦建議您衡量 `mysqldump` 程式碼的效能與使用分隔符號文字格式載入所帶來的效益。使用分隔符號文字格式的備份會為每個傾印的資料表分別建立索引標籤分隔文字檔案。您可以使用 `LOAD DATA LOCAL INFILE` 命令平行載入這些檔案，縮短匯入資料庫所需的時間。如需選擇 `mysqldump` 格式並載入資料的相關詳細資訊，請參閱 MySQL 文件中的 [Using mysqldump For Backups](#) 一節。

開始備份作業前，請確認在要複製到 Amazon RDS 的 MariaDB 或 MySQL 資料庫上設定複寫選項。複寫選項包括開啟二進位日誌及設定獨一無二的伺服器 ID。設定這些選項後，伺服器會開始記錄資料庫交易，並調整至就緒狀態，以擔任後續程序的來源複寫執行個體。

Note

對 `mysqldump` 使用 `--single-transaction` 選項，因為它會傾印資料庫的一致狀態。若要確保有效的傾印檔案，請勿在執行 `mysqldump` 時執行資料定義語言 (DDL) 陳述式。您可以為這些作業排定維護時段。

從傾印檔案中排除下列結構描述：`sys`、`performance_schema` 和 `information_schema`。`mysqldump` 公用程式已預設排除這些結構描述。若要移轉使用者和權限，請考慮使用產生資料控制語言 (DCL) 的工具來重新建立使用者和權限，例如公用程式。[pt-show-grants](#)

設定複寫選項

1. 編輯 `my.cnf` 檔案 (此檔案通常位於 `/etc` 之下)。

```
sudo vi /etc/my.cnf
```

將 `log_bin` 和 `server_id` 選項新增至 `[mysqld]` 部分。`log_bin` 選項會提供二進位記錄檔的檔案名稱識別符。`server_id` 選項會為來源與複本關係提供伺服器唯一識別碼。

以下範例顯示 `my.cnf` 檔案的 `[mysqld]` 部分的更新結果。

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

如需詳細資訊，請參閱 [MySQL 文件](#)。

2. 如需使用多可用區域資料庫叢集進行複寫，請將 `ENFORCE_GTID_CONSISTENCY` 和 `GTID_MODE` 參數設為 `ON`。

```
mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

使用資料庫執行個體進行複製時，不需要這些設定。

3. 重新啟動 `mysql` 服務。

```
sudo service mysqld restart
```

建立現有資料庫的備份複本

1. 您可使用 `mysqldump` 公用程式，以 SQL 或分隔符號文字等任一格式建立資料備份。

指定 `--master-data=2` 來建立可用來啟動伺服器間複寫作業的備份檔案。如需詳細資訊，請參閱 [mysqldump](#) 文件。

如需提升效能並確保資料完整性，請使用 `mysqldump` 的 `--order-by-primary` 和 `--single-transaction` 選項。

為避免備份中包含 MySQL 系統資料庫，請勿以 `mysqldump` 搭配使用 `--all-databases` 選項。如需詳細資訊，請參閱 MySQL 文件中的 [Creating a Data Snapshot Using mysqldump](#) 一節。

如有必要，請使用 `chmod` 命令，確保建立備份檔的目錄可以寫入。

Important

在 Windows 上，以管理員身分執行命令視窗。

- 若要產生 SQL 輸出，請使用下列命令。

對於LinuxmacOS、或Unix：

```
sudo mysqldump \  
  --databases database_name \  
  --master-data=2 \  
  --single-transaction \  
  --order-by-primary \  
  -r backup.sql \  
  -u local_user \  
  -p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

在 Windows 中：

```
mysqldump ^  
  --databases database_name ^  
  --master-data=2 ^
```

```
--single-transaction ^
--order-by-primary ^
-r backup.sql ^
-u local_user ^
-p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

- 若要產生分隔文字輸出，請使用下列命令。

對於LinuxmacOS、或Unix：

```
sudo mysqldump \  
  --tab=target_directory \  
  --fields-terminated-by ',' \  
  --fields-enclosed-by '"' \  
  --lines-terminated-by 0x0d0a \  
  database_name \  
  --master-data=2 \  
  --single-transaction \  
  --order-by-primary \  
  -p password
```

在 Windows 中：

```
mysqldump ^  
  --tab=target_directory ^  
  --fields-terminated-by "," ^  
  --fields-enclosed-by "" ^  
  --lines-terminated-by 0x0d0a ^  
  database_name ^  
  --master-data=2 ^  
  --single-transaction ^  
  --order-by-primary ^  
  -p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

必須在 Amazon RDS 資料庫中手動建立預存程序、觸發程序、函數或事件。如果您正在複製的資料庫中存有上述物件，則請在執行 `mysqldump` 時排除這些物件。為此，請在 `mysqldump` 命令中包含以下參數：`--routines=0 --triggers=0 --events=0`。

若是使用分隔符號文字格式，執行 `mysqldump` 命令時，系統會傳回 `CHANGE MASTER TO` 註解。此註解會包含主控端日誌檔案的名稱與位置。如果外部執行個體不是 MariaDB 10.0.24 或更新版本，請注意 `MASTER_LOG_FILE` 和 `MASTER_LOG_POS` 的值。設定複寫時您需要這些值。

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
MASTER_LOG_POS=107;
```

如果您使用的是 SQL 格式，您可以在備份檔案的 `CHANGE MASTER TO` 註解中取得主控端日誌檔案的名稱與位置。如果外部執行個體是 MariaDB 10.0.24 或更新版本，您可在下一步驟中取得 GTID。

- 如果您正在使用的外部執行個體是 MariaDB 10.0.24 或更新版本，您可使用 GTID 型複寫功能。在外部 MariaDB 執行個體上執行 `SHOW MASTER STATUS` 命令，以取得二進位日誌檔案的名稱和位置，接著在外部 MariaDB 執行個體上執行 `BINLOG_GTID_POS` 命令，將其轉換為 GTID。

```
SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
```

請記下系統傳回的 GTID，設定複寫時會需要使用。

- 壓縮複製的資料，以減少複製資料到 Amazon RDS 資料庫所需的網路資源量。請記下備份檔案的大小。決定要建立多大的 Amazon EC2 執行個體時，您需要此資訊。完成後，請使用 GZIP 或您慣用的壓縮公用程式壓縮備份檔。
 - 若要壓縮 SQL 輸出，請使用下列命令。

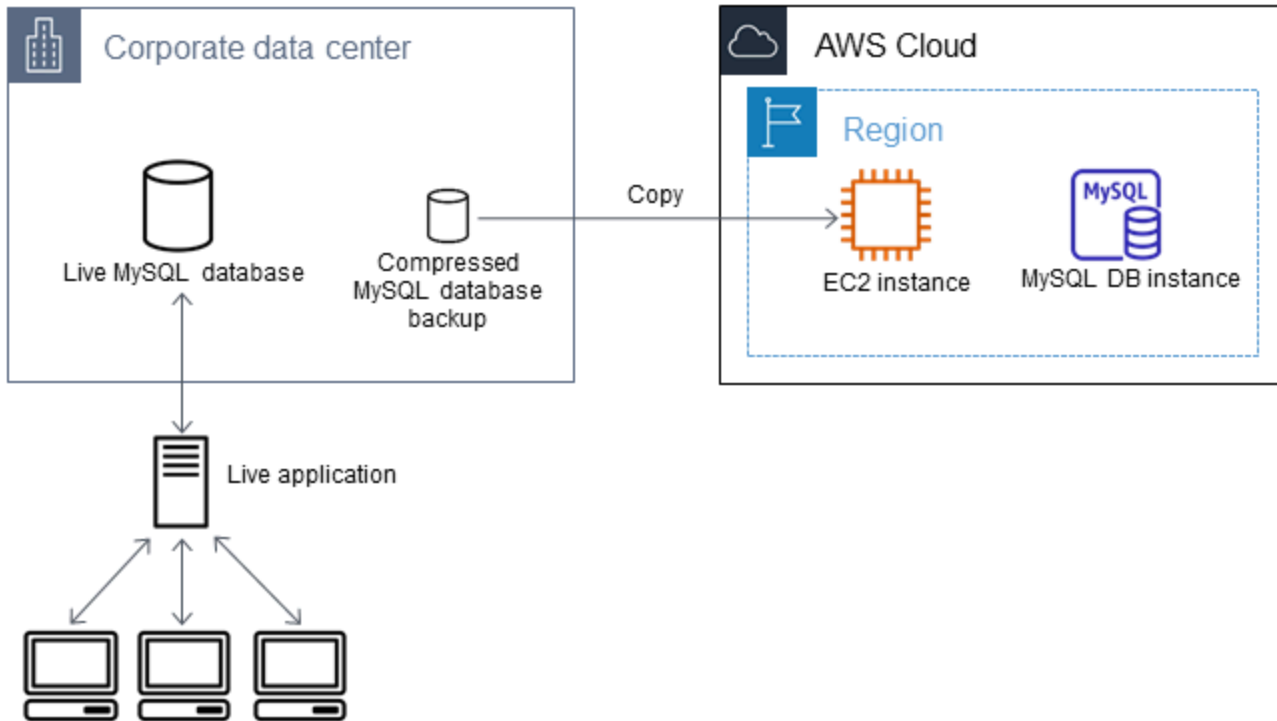
```
gzip backup.sql
```

- 若要壓縮分隔文字輸出，請使用下列命令。

```
tar -zcvf backup.tar.gz target_directory
```

建立 Amazon EC2 執行個體並複製壓縮的資料庫

比起直接在資料庫執行個體間複製未壓縮的資料，將壓縮的資料庫備份檔複製到 Amazon EC2 執行個體，需要的網路資源較少。資料放進 Amazon EC2 後，即可將資料直接複製到 MariaDB 或 MySQL 資料庫。為了節省網路資源成本，Amazon EC2 執行個體必須與 Amazon RDS 資料庫執行個體位於同一個 AWS 區域。將 Amazon EC2 執行個體放在與 Amazon RDS 資料庫相同的 AWS 區域中，也可以減少匯入期間的網路延遲。



建立 Amazon EC2 執行個體並複製資料

1. **AWS 區域** 在您打算建立 RDS 資料庫的位置中，建立虛擬私有雲端 (VPC)、VPC 安全性群組和 VPC 子網路。確認 VPC 安全群組的傳入規則會允許您應用程式連線至 AWS 所需的 IP 地址。您可以指定 IP 地址範圍 (例如 203.0.113.0/24) 或另一個 VPC 安全群組。您可以使用 [Amazon VPC 管理主控台](#) 來建立與管理 VPC 子網路和安全群組。如需詳細資訊，請參閱《Amazon Virtual Private Cloud 入門指南》中的 [Amazon VPC 入門](#)。
2. 開啟 [Amazon EC2 管理主控台](#)，然後選擇要同時包含您的 Amazon EC2 執行個體和 Amazon RDS 資料庫的 AWS 區域。使用您在步驟 1 中建立的 VPC、子網路和安全群組，啟動 Amazon EC2 執行個體。確認您選取的執行個體類型具有足夠的儲存空間，足以供未壓縮的資料庫備份檔使用。如需 Amazon EC2 執行個體的詳細資訊，請參閱《適用於 Linux 的 Amazon Elastic Compute Cloud 使用者指南》中的 [Amazon EC2 Linux 執行個體入門](#)。

- 若要從 Amazon EC2 執行個體連線至 Amazon RDS 資料庫，請編輯您的 VPC 安全群組。新增會指定您 EC2 執行個體私有 IP 地址的傳入規則。您可在 EC2 主控台視窗中，從 Instance (執行個體) 窗格的 Details (詳細資訊) 分頁標籤找到私有 IP 地址。若要編輯 VPC 安全群組並新增傳入規則，請選取 EC2 主控台導覽窗格中的 Security Groups (安全群組)，接著選擇您的安全群組，然後新增 MySQL 或 Aurora 的傳入規則，指定 EC2 執行個體的私有 IP 地址。如需了解如何將傳入規則新增至 VPC 安全群組，請參閱《Amazon VPC 使用者指南》中的[新增與移除規則](#)。
- 將壓縮的資料庫備份檔從本機系統複製到 Amazon EC2 執行個體。如有必要，請使用 `chmod` 命令，確認您擁有 Amazon EC2 執行個體的目標目錄寫入許可。您可使用 `scp` 命令或 Secure Shell (SSH) 用戶端來複製檔案。以下是範例。

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

Important

務必使用安全網路傳輸通訊協定來複製機密資料。

- 使用下列命令連線至您的 Amazon EC2 執行個體，並安裝最新更新與 MySQL 用戶端工具。

```
sudo yum update -y
sudo yum install mysql -y
```

如需詳細資訊，請參閱《Linux Amazon Elastic Compute Cloud 使用者指南》中的[連線到您的執行個體](#)。

Important

此範例會在 Amazon Linux 發行版的 Amazon Machine Image (AMI) 上安裝 MySQL 用戶端。若要在不同的發行版上安裝 MySQL 用戶端，例如 Ubuntu 或 Red Hat Enterprise Linux，則此範例將無法運作。如需安裝 MySQL 的資訊，請參閱 MySQL 文件中的[安裝和升級 MySQL](#)。

- 連線到 Amazon EC2 執行個體後，將您的資料庫備份檔解壓縮。範例如下。
 - 若要將 SQL 輸出解壓縮，請使用下列命令。

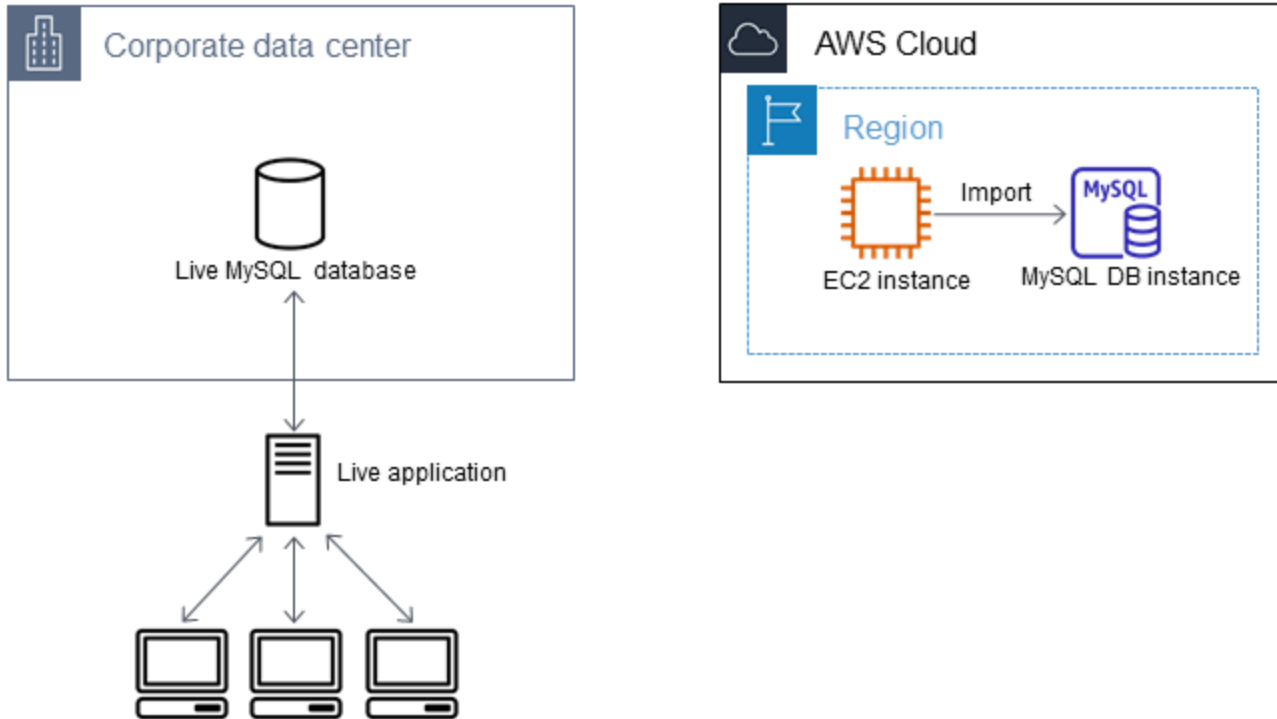
```
gzip backup.sql.gz -d
```

- 若要將分隔符號文字輸出解壓縮，請使用下列命令。

```
tar xzvf backup.tar.gz
```

建立 MySQL 或 MariaDB 資料庫，並從 Amazon EC2 執行個體匯入資料

透過在與 Amazon EC2 執行個體相同的區域中建立 MariaDB 資料庫執行個體、MySQL 資料庫執行個體或 MySQL 異 AWS 地同步備份資料庫叢集，您可以比透過網際網路更快地從 EC2 匯入資料庫備份檔案。



建立 MariaDB 或 MySQL 資料庫並匯入資料

1. 判斷需要何種資料庫執行個體類別及多大的儲存空間，才能支援這個 Amazon RDS 資料庫預計的工作負載。在此流程的過程中，決定資料載入程序需要多少空間與處理容量才夠。同時決定處理生產工作負載所需的資源。您可根據來源 MariaDB 或 MySQL 資料庫的大小與資源加以估計。如需詳細資訊，請參閱 [資料庫執行個體類別](#)。
2. 在包含 Amazon EC2 執行個體的區域中建立資料庫執行個體或異 AWS 地同步備份資料庫叢集。

若要建立 MySQL 多可用區域資料庫叢集，請遵循[建立多可用區域資料庫叢集](#)中的指示。

若要建立 MariaDB 或 MySQL 資料庫執行個體，請遵循[建立 Amazon RDS 資料庫執行個體](#)中的指示，並使用下列指導方針：

- 指定與您來源資料庫執行個體相容的資料庫引擎版本，如下所示：
 - 若您的來源執行個體為 MySQL 5.5.x 版，則 Amazon RDS 資料庫執行個體必須為 MySQL。
 - 若您的來源執行個體為 MySQL 5.6.x 或 5.7.x 版，則 Amazon RDS 資料庫執行個體必須為 MySQL 或 MariaDB。
 - 如果您的來源執行個體為 MySQL 8.0.x 版本，則 Amazon RDS 資料庫執行個體必須為 MySQL 8.0.x 版本。
 - 若您的來源執行個體為 MariaDB 5.5 或更新版本，則 Amazon RDS 資料庫執行個體必須為 MariaDB。
 - 指定與 Amazon EC2 執行個體相同的虛擬私有雲端 (VPC) 和 VPC 安全群組。這種作法可確保您的 Amazon EC2 執行個體與 Amazon RDS 執行個體能在網路上看見彼此。確保您的資料庫執行個體可以公開存取。若要如稍後說明的方式，以您的來源資料庫設定複寫，必須將資料庫執行個體設為公開存取。
 - 匯入資料庫備份前，請勿設定多個可用區域、備份保留期或僅供讀取複本。匯入完成後，您可以設定生產執行個體的多可用區域和備份保留。
3. 檢視 Amazon RDS 資料庫的預設組態選項。如果資料庫的預設參數群組沒有您想要的組態選項，請尋找其他提供您所需選項的參數群組，或者建立新的參數群組。如需建立參數群組的詳細資訊，請參閱 [使用參數群組](#) 一文。
 4. 以主要使用者身分連線至新的 Amazon RDS 資料庫。建立支援需存取執行個體的管理員、應用程式和服務所需的使用者。Amazon RDS 資料庫的主機名稱是執行個體的 Endpoint (端點) 值 (不含連接埠編號)。例如，mysamp1edb.123456789012.us-west-2.rds.amazonaws.com。端點值可在 Amazon RDS 管理主控台的資料庫詳細資訊中找到。
 5. 連線到您的 Amazon EC2 執行個體。如需詳細資訊，請參閱《Linux Amazon Elastic Compute Cloud 使用者指南》中的[連線到您的執行個體](#)。
 6. 使用 `mysql` 命令，以遠端主機的形式從 Amazon EC2 執行個體連線至 Amazon RDS 資料庫。以下是範例。

```
mysql -h host_name -P 3306 -u db_master_user -p
```

主機名稱是 Amazon RDS 資料庫端點。

7. 出現 `mysql` 提示時，請執行 `source` 命令並傳入您的資料庫傾印檔案名稱，以將資料載入 Amazon RDS 資料庫執行個體：
 - 若為 SQL 格式，請使用下列命令。

```
mysql> source backup.sql;
```

- 若為分隔符號文字格式，請先建立資料庫 (如果它不是您在設定 Amazon RDS 資料庫時建立的預設資料庫)。

```
mysql> create database database_name;  
mysql> use database_name;
```

接著建立資料表。

```
mysql> source table1.sql  
mysql> source table2.sql  
etc...
```

然後匯入資料。

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY  
' ,' ENCLOSED BY '"' LINES TERMINATED BY '\n';  
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY  
' ,' ENCLOSED BY '"' LINES TERMINATED BY '\n';  
etc...
```

若要提升效能，您可透過多個連線平行執行這些操作，如此您的所有資料表就會建立並隨後載入。

Note

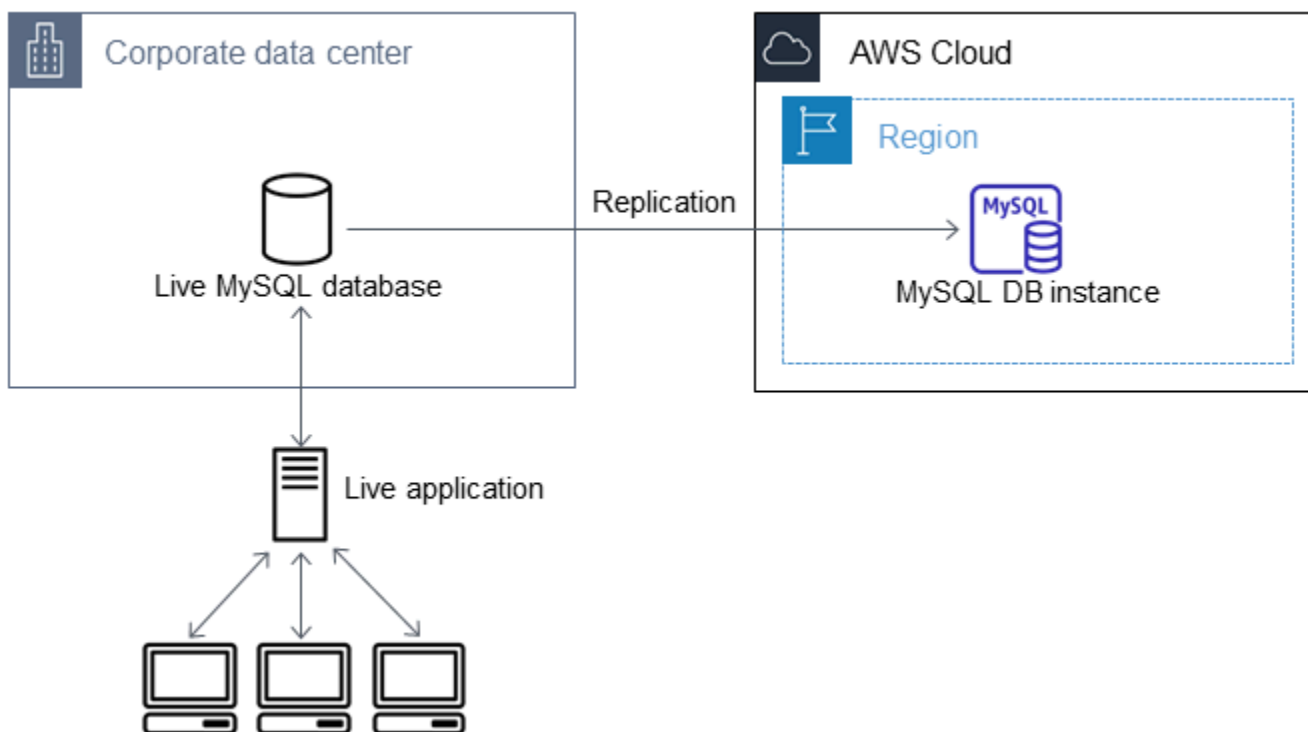
如果您在最初傾印資料表時使用 mysqldump 的任何資料格式化選項，請務必使用與 LOAD DATA LOCAL INFILE 相同的選項，以確保資料檔案內容的正確解譯。

8. 對匯入的資料庫中的一或兩個資料表執行簡單 SELECT 查詢，以確認匯入是否成功。

如果您不再需要此程序中使用的 Amazon EC2 執行個體，請終止 EC2 執行個體以減少 AWS 源使用量。若要終止 EC2 執行個體，請參閱《Amazon EC2 使用者指南》中的 [終止執行個體](#)。

在外部資料庫與新的 Amazon RDS 資料庫之間複寫資料

在複製資料並傳輸至 MariaDB 或 MySQL 資料庫期間，您的來源資料庫可能就會完成更新。因此，您可以使用複寫將複製的資料庫 up-to-date 與來源資料庫一起使用。



在 Amazon RDS 資料庫上啟動複寫功能所需的許可有其限制，不供 Amazon RDS 主要使用者使用。基於這個原因，請確認使用 Amazon RDS [mysql.rds_set_external_master](#) 命令或 [mysql.rds_set_external_master_gtid](#) 命令設定複寫功能，並使用 [mysql.rds_start_replication](#) 命令啟動在即時資料庫與 Amazon RDS 資料庫之間複寫的功能。

啟動複寫

稍早之前，您已開啟來源資料庫的二進位日誌，並設定獨一無二的伺服器 ID。現在，您可將 Amazon RDS 資料庫設為複本，並將即時資料庫設為來源複寫執行個體。

1. 在 Amazon RDS 管理主控台中，將託管來源資料庫之伺服器的 IP 地址，新增至 Amazon RDS 資料庫的 VPC 安全群組。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。

您可能還需要設定本機網路，以允許從 Amazon RDS 資料庫的 IP 地址建立連線，使其能與來源執行個體通訊。若要尋找 Amazon RDS 資料庫的 IP 地址，請使用 `host` 命令。

```
host rds_db_endpoint
```

主機名稱是來自 Amazon RDS 資料庫端點的 DNS 名稱，例如 `myinstance.123456789012.us-east-1.rds.amazonaws.com`。端點值可在 Amazon RDS 管理主控台的執行個體詳細資訊中找到。

2. 使用您選擇的用戶端，連線至來源執行個體，然後建立複寫作業所需的使用者。此帳戶只供複寫作業使用，務必限制其存取您的網域，以提升安全性。以下是範例。

MySQL 5.5、5.6 和 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY 'password';
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

3. 若為來源執行個體，請將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。舉例來說，若要將所有資料庫的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的「repl_user」使用者，請發出下列命令。

MySQL 5.5、5.6 和 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

4. 如果您使用 SQL 格式建立備份檔，且外部執行個體並非 MariaDB 10.0.24 或更新版本，請查看該檔案的內容。

```
cat backup.sql
```

該檔案包括 CHANGE MASTER TO 註解，內含主控端日誌檔案的名稱與位置。若您使用 mysqldump 命令的 --master-data 選項，備份檔內就會包含這個註解。請注意 MASTER_LOG_FILE 和 MASTER_LOG_POS 的值。

```
--  
-- Position to start replication or point-in-time recovery from  
--  
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

如果您使用分隔符號文字格式來建立備份檔，且外部執行個體並非 MariaDB 10.0.24 或更高版本，您應該已在本主題中的「建立現有資料庫的備份複本」程序的步驟 1 時就建立二進位日誌座標。

如果外部執行個體是 MariaDB 10.0.24 或更高版本，您應該已在本主題中的「建立現有資料庫的備份複本」程序的步驟 2 時就取得 GTID，以此啟動複寫作業。

5. 使 Amazon RDS 資料庫成為複本。如果外部執行個體並非 MariaDB 10.0.24 或更高版本，請以主要使用者身分連線至 Amazon RDS 資料庫，接著使用 [mysql.rds_set_external_master](#) 命令找到來源資料庫，將其視為來源複寫執行個體。如果您有 SQL 格式的備份檔，請使用您在上一步驟所確定的主控端日誌名稱與位置。或者，如果您使用的是分隔符號文字格式，請使用您在建立備份檔時確定的主控端日誌名稱與位置。以下是範例。

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,  
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

Note


指定此處所顯示提示以外的憑證，作為安全最佳實務。

如果外部執行個體並非 MariaDB 10.0.24 或更高版本，請以主要使用者身分連線至 Amazon RDS 資料庫，接著使用 [mysql.rds_set_external_master_gtid](#) 命令找到來源資料庫，將其視為來源複寫執行

個體。使用您在本主題中的「建立現有資料庫的備份複本」程序的步驟 2 中所確定的 GTID。以下是範例。

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,  
'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` 是來源複寫執行個體的 IP 地址。目前不支援 EC2 私有 DNS 地址。


 Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

6. 在 Amazon RDS 資料庫上，發出 [mysql.rds_start_replication](#) 命令來啟動複寫。

```
CALL mysql.rds_start_replication;
```

7. 在 Amazon RDS 資料庫上，執行 [顯示複本狀態](#) 命令，以判斷複本何時 up-to-date 與來源複寫執行個體一起使用。SHOW REPLICA STATUS 命令的結果包括 Seconds_Behind_Master 欄位。當 Seconds_Behind_Master 欄位傳回 0 時，則複本會 up-to-date 包含來源複寫執行個體。

 Note

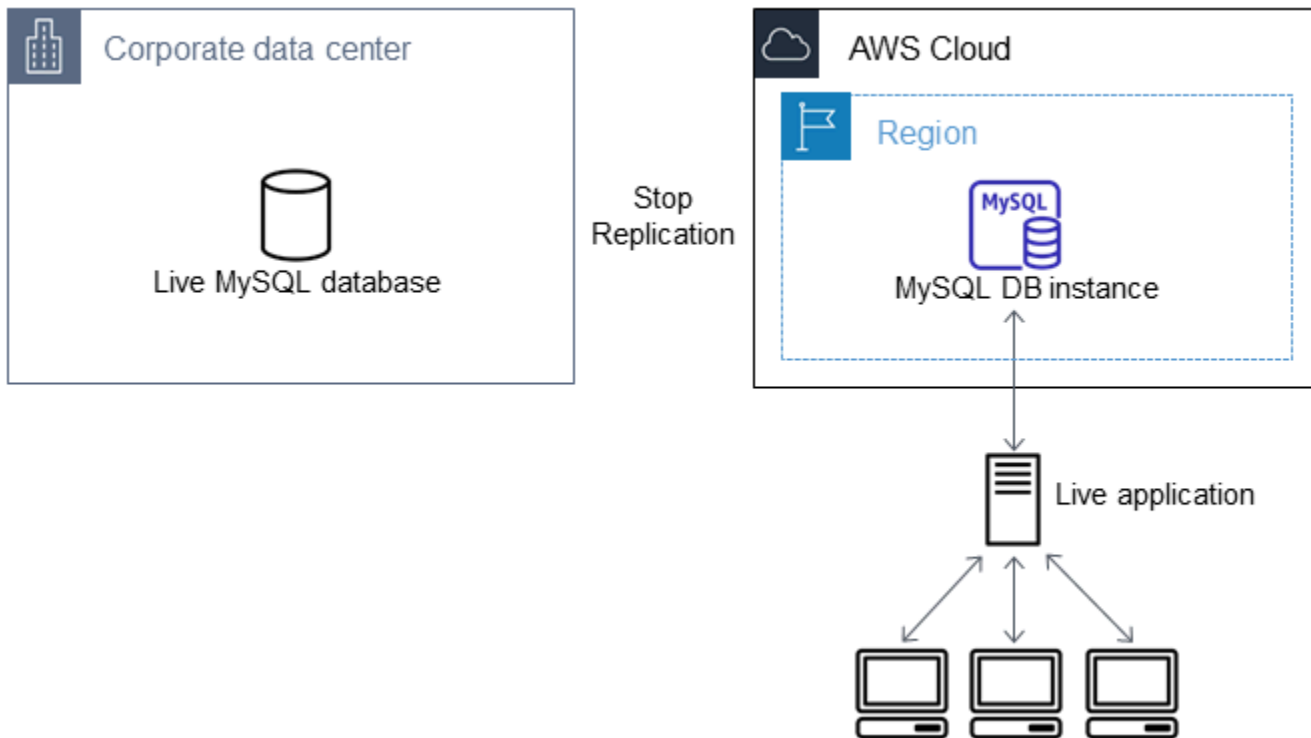
MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

若為 MariaDB 10.5、10.6 或 10.11 資料庫執行個體，請執行 [mysql.rds_replica_status](#) 程序，而非 MySQL 命令。

8. Amazon RDS 資料庫存放之後 up-to-date，請開啟自動備份，以便在需要時還原該資料庫。您可以使用 [Amazon RDS 管理主控台](#)，開啟或修改 Amazon RDS 資料庫的自動備份。如需詳細資訊，請參閱 [備份簡介](#)。

將您的即時應用程式重新導向 Amazon RDS 執行個體

在 MariaDB 或 MySQL 資料庫 up-to-date 與來源複寫執行個體一起使用之後，您現在可以更新即時應用程式以使用 Amazon RDS 執行個體。



將您的即時應用程式重新導向至 MariaDB 或 MySQL 資料庫並停止複寫

- 若要新增 Amazon RDS 資料庫的 VPC 安全群組，請新增託管應用程式之伺服器的 IP 地址。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。
- 確認 [\[顯示複本狀態\]](#) 命令結果中的 Seconds_Behind_Master 欄位是 0，表示複本 up-to-date 與來源複製執行個體一起使用。

```
SHOW REPLICA STATUS;
```

Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

若為 MariaDB 10.5、10.6 或 10.11 資料庫執行個體，請執行 [mysql.rds_replica_status](#) 程序，而非 MySQL 命令。

- 交易完成時關閉所有與此來源的連線。

- 更新您的應用程式，以使用 Amazon RDS 資料庫。這裡所謂的更新，通常包括變更連線設定，以識別 Amazon RDS 資料庫的主機名稱和連接埠、要與其連線的使用者帳戶和密碼，以及要使用的資料庫。
- 連線到資料庫執行個體。

對至多可用區域資料庫叢集，請連線至寫入器資料庫執行個體。

- 使用 [mysql.rds_stop_replication](#) 命令停止 Amazon RDS 執行個體的複寫。

```
CALL mysql.rds_stop_replication;
```

- 在您的 Amazon RDS 資料庫上執行 [mysql.rds_reset_external_master](#) 命令來重設複寫組態，使得此執行個體不再視為複本。

```
CALL mysql.rds_reset_external_master;
```

- 開啟其他 Amazon RDS 功能，例如異地同步備份支援和僅供讀取複本。如需更多詳細資訊，請參閱 [設定及管理多可用區部署](#) 及 [使用資料庫執行個體僅供讀取複本](#)。

匯入任何 MariaDB 或 MySQL 資料庫執行個體來源的資料

建議您在資料載入之前和之後建立目標 Amazon RDS 資料庫執行個體的資料庫快照。Amazon RDS 資料庫快照是您資料庫執行個體的完整備份，可將您的資料庫執行個體還原至已知狀態。啟動資料庫快照作業時，資料庫執行個體的輸入/輸出操作會在資料庫備份時暫停。

若在載入前先建立資料庫快照，一旦有需要，就能將資料庫還原至載入前的狀態。若能在載入後立即拍攝資料庫快照，萬一發生事故，就不必重新載入資料，同時也能植入新的資料庫執行個體。

下列清單顯示要採取的步驟。下文將詳細討論各步驟。

- 建立一般檔案，其中需包含要載入的資料。
- 停止任何應用程式對目標資料庫執行個體的存取作業。
- 建立資料庫快照。
- 建議關閉 Amazon RDS 自動備份功能。
- 載入資料。
- 重新啟用自動備份。

步驟 1：建立一般檔案，其中需包含要載入的資料

使用逗號分隔值 (CSV) 等通用格式存放要載入的資料。每個資料表必須擁有各自的檔案；您不可將不同資料表的資料整併成同一檔案。以對應資料表的相同名稱命名各個檔案。副檔名可隨意指定。例如，如果資料表名稱為 `sales`，則檔案名稱可能是 `sales.csv` 或 `sales.txt`，但不是 `sales_01.csv`。

可以的話，依照預計載入之資料表的主索引鍵排序資料。這樣可以大幅縮短載入時間，並將磁碟儲存空間需求降至最低。

檔案越小，此程序的速度就會越快，效率越高。如有個別檔案未壓縮前的大小大於 1 GiB，請將檔案分割成多個檔案，再分別載入。

在泛 Unix 系統 (包括 Linux) 上，使用 `split` 命令。舉例來說，下列命令會將 `sales.csv` 檔案分割成多個小於 1 GiB 的檔案，且只會於分行處分割 (`-C 1024m`)。新檔案名稱為 `sales.part_00`、`sales.part_01`，以此類推。

```
split -C 1024m -d sales.csv sales.part_
```

其他作業系統也有類似的公用程式可用。

步驟 2：停止任何應用程式對目標資料庫執行個體的存取作業

開始載入大量資料前，請先停止所有應用程式活動，使其暫停存取您預計要載入資料的目標資料庫執行個體。如果有其他工作階段會修改要載入或參考的資料表，我們尤其建議執行這項操作。這麼做可以降低載入期間違反限制條件的風險，並提升載入效能。此外，還可以將資料庫執行個體還原至載入前的時間點，以免遺失非載入作業程序所做的變更。

當然，這可能無法盡如人願，或可能不切實際。如果您無法在載入作業前停止應用程式存取資料庫執行個體，請採取因應措施，確保資料的可用性與完整性。所需的確切步驟會因特定使用案例與現場需求而異。

步驟 3：建立資料庫快照

如果您預計要載入資料的新資料庫執行個體內沒有資料，您可略過此步驟。否則，如能建立資料庫執行個體的資料庫快照，即可在必要時，將資料庫執行個體還原至載入前的時間點。如前所述，啟動資料庫快照作業時，資料庫執行個體的輸入/輸出操作會在資料庫備份時暫停幾分鐘。

下列範例使用命 AWS CLI `create-db-snapshot` 令建立 AcmeRDS 執行個體的資料庫快照，並為資料庫快照提供識別碼 "preload"。

對於LinuxmacOS、或Unix：

```
aws rds create-db-snapshot \  
  --db-instance-identifier AcmeRDS \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds create-db-snapshot ^  
  --db-instance-identifier AcmeRDS ^  
  --db-snapshot-identifier preload
```

您也可以使用從資料庫快照還原的功能，建立試轉用的測試資料庫執行個體，或復原載入期間所做的變更。

請注意，如從資料庫快照還原資料庫，系統會建立新的資料庫執行個體，且如同所有資料庫執行個體一樣，該執行個體會具備獨一無二的識別碼與端點。若要還原資料庫執行個體，且不會變更端點，請先刪除資料庫執行個體，如此才能重新使用端點。

舉例來說，若要建立試轉或其他測試所需的資料庫執行個體，您必須賦予資料庫執行個體專屬識別符。在此例中，識別符是 *AcmeRDS-2*。範例使用與 *AcmeRDS-2* 關聯的端點連線至資料庫執行個體。

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS-2 \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS-2 ^  
  --db-snapshot-identifier preload
```

若要重新使用現有端點，請先刪除資料庫執行個體，然後指定相同的識別符給還原的資料庫。

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --final-db-snapshot-identifier AcmeRDS-Final
```



```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds delete-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --final-db-snapshot-identifier AcmeRDS-Final  
  
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS ^  
  --db-snapshot-identifier preload
```

前述範例是在刪除資料庫執行個體前擷取資料庫的最終資料庫快照。此為選用操作，但建議您採用。

步驟 4：建議關閉 Amazon RDS 自動備份功能

Warning

如果您需要執行 point-in-time 復原，請勿關閉自動備份。

關閉自動備份會清除所有現有的備份，因此在關閉自動備份後無法進行 point-in-time 復原。停用自動備份功能可產生最佳化效能，但並非資料載入作業的必要功能。關閉自動備份功能不會影響手動資料庫快照。所有現有手動資料庫快照仍可用於復原作業。

關閉自動備份功能可將載入時間減少約 25%，並可降低載入期間所需的儲存空間。如果您預計載入資料的新資料庫執行個體不含任何資料，則要加快載入速度，並避免備份產生額外的儲存空間需求，關閉備份功能是最簡單的方式。不過，在某些情況下，您可能會計劃載入已包含資料的資料庫執行個體。如果是這樣，請權衡關閉備份與失去執行能力的影響所帶來的好處 point-in-time-recovery。

預設情況下，資料庫執行個體會開啟自動備份功能 (保留期為 1 天)。若要關閉自動備份，請將備份保留期設為零。載入後，您可以將備份保留期設為非零的值，以重新開啟自動備份功能。若要開啟或關閉備份功能，Amazon RDS 會關閉資料庫執行個體，然後重新啟動，以便開啟或關閉 MariaDB 或 MySQL 記錄功能。

使用指 AWS CLI `modify-db-instance` 令將備份保留設定為零，並立即套用變更。將保留期設為 0 需要重新啟動資料庫執行個體，請靜待重新啟動作業完成，再繼續操作。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --apply-immediately \  
  --backup-retention-period 0
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --apply-immediately ^  
  --backup-retention-period 0
```

您可以使用 AWS CLI `describe-db-instances` 指令檢查資料庫執行個體的狀態。以下範例顯示 `AcmeRDS` 資料庫執行個體的資料庫執行個體狀態。

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].  
{DBInstanceStatus:DBInstanceStatus}"
```

當資料庫執行個體狀態為 `available` 時，您即可繼續操作。

步驟 5：載入資料

使用 MySQL `LOAD DATA LOCAL INFILE` 陳述式將資料列從平面檔案讀取到資料庫資料表中。

下列範例說明如何將資料從名為的檔案載入 `sales.txt` 入資料庫 `Sales` 中名為的資料表。

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '  
  ENCLOSED BY '' ESCAPED BY '\\';  
Query OK, 1 row affected (0.01 sec)  
Records: 1 Deleted: 0 Skipped: 0 Warnings: 0
```

如需有關 `LOAD DATA` 陳述式的詳細資訊，請參閱 [MySQL 文件](#)。

步驟 6：重新開啟 Amazon RDS 自動備份功能

載入完成後，您可以將備份保留期設回載入前的值，以重新開啟 Amazon RDS 自動備份功能。如前所述，Amazon RDS 會重新啟動資料庫執行個體，所以短暫停機是正常現象。

下列範例會使用命 AWS CLI `modify-db-instance` 令來開啟 AcmeRDS 資料庫執行個體的自動備份，並將保留期限設定為一天。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --backup-retention-period 1 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --backup-retention-period 1 ^  
  --apply-immediately
```

使用 Amazon RDS 中的 MariaDB 複寫

您通常會使用僅供讀取複本來設定 Amazon RDS 資料庫執行個體間的複寫。如需僅供讀取複本的一般資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。如需在 Amazon RDS for MariaDB 上使用僅供讀取複本的特定資訊，請參閱 [使用 MariaDB 僅供讀取複本](#)。

您也可依據二進位日誌座標設定複寫供 MariaDB 資料庫執行個體使用。以 MariaDB 執行個體而言，您也可依據毀損安全功能提升的全域交易 ID (GTID) 來設定複寫。如需詳細資訊，請參閱 [使用外部來源執行個體設定 GTID 式複寫](#)。

下列為適用於 RDS for MariaDB 的其他複寫選項：

- 您可以在 RDS for MariaDB 資料庫執行個體，以及位於 Amazon RDS 外部的 MySQL 或 MariaDB 執行個體之間，設定複寫。如需使用外部來源設定複寫的詳細資訊，請參閱 [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。
- 您可將複寫作業設定為從 Amazon RDS 外部的 MySQL 或 MariaDB 執行個體匯入資料庫，或者設定為將資料庫匯出至這類執行個體。如需詳細資訊，請參閱 [減少將資料匯入 Amazon RDS MariaDB 或 MySQL 資料庫執行個體時的停機時間](#) 及 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

對於以上任何複寫選項，您可以使用基於列的複寫、基於陳述式或混合式複寫。基於列的複寫只會複寫 SQL 陳述式所變更的列。基於陳述式的複寫會複寫整個 SQL 陳述式。混合式複寫會盡可能使用基於陳述式的複寫，但是當基於陳述式的複寫有不安全的 SQL 陳述式執行時，則會切換到基於列的複寫。在大多數情況下，建議使用混合式複寫。資料庫執行個體的二進位日誌格式決定複寫是基於列、基於陳述式或混合式。如需有關設定二進位日誌格式的資訊，請參閱 [二進位日誌格式](#)。

主題

- [使用 MariaDB 僅供讀取複本](#)
- [使用外部來源執行個體設定 GTID 式複寫](#)
- [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)

使用 MariaDB 僅供讀取複本

接下來，您可以找到在 Amazon RDS for MariaDB 上使用僅供讀取複本的特定資訊。如需僅供讀取複本的一般資訊及使用說明，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

主題

- [使用 MariaDB 設定僅供讀取複本](#)

- [設定與 MariaDB 的複寫篩選條件](#)
- [透過 MariaDB 設定延遲複寫](#)
- [使用 MariaDB 更新僅供讀取複本](#)
- [使用 MariaDB 處理多可用區域僅供讀取複本的部署](#)
- [搭配 RDS for MariaDB 使用階層式僅供讀取複本](#)
- [監控 MariaDB 僅供讀取複本](#)
- [透過 MariaDB 僅供讀取複本啟動與停用複寫](#)
- [對 MariaDB 僅供讀取複本的問題進行故障診斷](#)

使用 MariaDB 設定僅供讀取複本

MariaDB 資料庫執行個體做為複寫來源前，務必將備份保留期設定為 0 以外的值，藉此開啟來源資料庫執行個體的自動備份。若僅供讀取複本能讓其他僅供讀取複本做為來源資料庫執行個體，則亦適用此要求。

您可以從相同區域內的一個資料庫執行個體建立至多 15 個僅供讀取複本。若希望複寫作業順利運作，每個僅供讀取複本具備的運算和儲存資源數量應與來源資料庫執行個體相同。若您擴展來源資料庫執行個體，也要擴展僅供讀取複本。

RDS for MariaDB 支援階層式僅供讀取複本。若要了解如何設定階層式僅供讀取複本，請參閱 [搭配 RDS for MariaDB 使用階層式僅供讀取複本](#)。

您可以同時執行多個僅供讀取複本建立，並刪除參考相同來源資料庫執行個體的動作。執行這些動作時，請將每個來源執行個體的僅供讀取複本數目限制在 15 個之內。

設定與 MariaDB 的複寫篩選條件

您可以使用複寫篩選條件來指定要與僅供讀取複本一起複寫的資料庫和資料表。複寫篩選條件可以包含複寫中的資料庫和資料表，或將其排除在複寫之外。

下列是複寫篩選條件的一些應用案例：

- 要縮小僅供讀取複本的大小。使用複寫篩選，您可以排除僅供讀取複本不需要的資料庫和資料表。
- 基於安全考量，要將資料庫和資料表從僅供讀取複本中排除。
- 為不同僅供讀取複本的特定應用案例複寫不同的資料庫和資料表。例如，您可以使用特定僅供讀取複本進行分析或分區。

- 對於在不同 AWS 區域 中具有僅供讀取複本的資料庫執行個體，可在不同 AWS 區域 中複寫不同的資料庫或資料表。

Note

您也可以使用複寫篩選條件來指定要與主要 MariaDB 資料庫執行個體 (設定為輸入複寫拓撲中的複寫) 一起複寫的資料庫和資料表。如需此組態的詳細資訊，請參閱「[使用外部來源執行個體設定二進位日誌檔案位置複寫](#)」。

主題

- [設定適用於 RDS for MariaDB 的複寫篩選參數](#)
- [RDS for MariaDB 的複寫篩選限制](#)
- [RDS for MariaDB 的複寫篩選範例](#)
- [檢視僅供讀取複本的複寫篩選條件](#)

設定適用於 RDS for MariaDB 的複寫篩選參數

要設定複寫篩選條件，請在僅供讀取複本上設定下列複寫篩選參數：

- `replicate-do-db` – 將變更複寫至指定的資料庫。當您為僅供讀取複本設定此參數時，只會複寫參數中指定的資料庫。
- `replicate-ignore-db` – 請勿將變更複寫至指定的資料庫。為僅供讀取複本設定 `replicate-do-db` 參數時，不會評估此參數。
- `replicate-do-table` – 將變更複製到指定的資料表。當您為僅供讀取複本設定此參數時，只會複寫參數中指定的資料表。此外，設定 `replicate-do-db` 或 `replicate-ignore-db` 參數時，必須使用僅供讀取複本將包含指定資料表的資料庫納入複寫中。
- `replicate-ignore-table` – 請勿將變更複寫至指定的資料表。為僅供讀取複本設定 `replicate-do-table` 參數時，不會評估此參數。
- `replicate-wild-do-table` – 根據指定的資料庫和資料表名稱模式複寫資料表。支援 % 和 _ 萬用字元。設定 `replicate-do-db` 或 `replicate-ignore-db` 參數時，請務必使用僅供讀取複本將包含指定資料表的資料庫納入複寫中。
- `replicate-wild-ignore-table` – 請勿根據指定的資料庫和資料表名稱模式複寫資料表。支援 % 和 _ 萬用字元。為僅供讀取複本設定 `replicate-do-table` 或 `replicate-wild-do-table` 參數時，不會評估此參數。

系統會按照列出的順序對參數進行評估。如需有關這些參數如何運作的詳細資訊，請參閱 [MariaDB 文件](#)。

根據預設，這些參數中的每個參數都有一個空值。在每個僅供讀取複本上，您可以使用這些參數來設定、變更和刪除複寫篩選條件。當您設定其中一個參數時，請使用逗號將每個篩選條件與其他篩選條件分隔。

您可以在 % 和 _ 參數中使用 `replicate-wild-do-table` 和 `replicate-wild-ignore-table` 萬用字元。% 萬用字元等同於任意數目的字元，而 _ 萬用字元只會等同於一個字元。

來源資料庫執行個體的二進位記錄格式對複寫非常重要，因為它會決定資料變更的記錄。`binlog_format` 參數的設定會決定複寫是以資料列為基礎還是以陳述式為基礎。如需詳細資訊，請參閱 [二進位日誌格式](#)。

Note

無論來源資料庫執行個體上的 `binlog_format` 設定為何，所有資料定義語言 (DDL) 陳述式都會複寫為陳述式。

RDS for MariaDB 的複寫篩選限制

下列限制適用於 RDS for MariaDB 的複寫篩選：

- 每個複寫篩選參數都有 2,000 個字元的限制。
- 複寫篩選條件不支援逗號。
- 不支援用於二進位日誌篩選的 MariaDB `binlog_do_db` 和 `binlog_ignore_db` 選項。
- 複寫篩選不支援 XA 交易。

如需詳細資訊，請參閱 MySQL 文件中的 [XA 交易的限制](#)。

- RDS for MariaDB 10.2 版不支援複寫篩選。

RDS for MariaDB 的複寫篩選範例

若要設定僅供讀取複本的複寫篩選，請修改與僅供讀取複本關聯之參數群組中的複寫篩選參數。

Note

您無法修改預設參數群組。如果僅供讀取複本使用預設參數群組，請建立新的參數群組，並將它與僅供讀取複本建立關聯。如需資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 在參數群組中設定參數。如需有關設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。當您在參數群組中設定參數時，與參數群組關聯的所有資料庫執行個體都會使用參數設定。如果您在參數群組中設定複寫篩選參數，請確定參數群組僅與僅供讀取複本相關聯。將來源資料庫執行個體的複寫篩選參數保留空白。

下列範例會使用 AWS CLI 設定參數。這些範例將 ApplyMethod 設定為 immediate，以便在 CLI 命令完成後立即發生參數變更。如果您想要在僅供讀取複本重新啟動後套用擱置變更，請將設定 ApplyMethod 為 pending-reboot。

下列範例會設定複寫篩選條件：

- [Including databases in replication](#)
- [Including tables in replication](#)
- [Including tables in replication with wildcard characters](#)
- [Escaping wildcard characters in names](#)
- [Excluding databases from replication](#)
- [Excluding tables from replication](#)
- [Excluding tables from replication using wildcard characters](#)

Example 在複寫中包含資料庫

下列範例包含複寫中的 mydb1 和 mydb2 資料庫。當您為僅供讀取複本設定 replicate-do-db 時，只會複寫參數中指定的資料庫。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",  
  "ApplyMethod": "immediate"}]"
```


在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
  "ApplyMethod":"immediate"}]"
```

Example 在複寫中包含資料表

下列範例包含複寫資料庫 table1 中的 table2 和 mydb1 資料表。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
  "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
  "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example 使用萬用字元在複寫中包含資料表

下列範例包含複寫時在資料庫 orders 中名稱開頭為 returns 和 mydb 的資料表。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
  "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
  "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Example 逸出名稱中的萬用字元

下列範例會示範如何使用逸出字元 \ 來逸出屬於名稱一部分的萬用字元。

假設您在資料庫 mydb1 中有數個以 my_table 開頭的資料庫名稱，而且您想要在複寫中包含這些資料表。資料表名稱包含底線，也是萬用字元，因此此範例會逸出資料表名稱中的底線。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
  \_table%", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
  \_table%", "ApplyMethod":"immediate"}]"
```

Example 從複寫中排除資料庫

下列範例會從複寫中排除 mydb1 和 mydb2 資料庫。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
  "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
    "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Example 從複寫中排除資料表

下列範例會從複寫中排除資料庫 table1 中的資料表 table2 和 mydb1。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
    "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
    "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example 使用萬用字元從複寫中排除資料表

下列範例會從複寫中排除資料庫 orders 中名稱開頭為 returns 和 mydb 的資料表。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
    "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
```

```
--db-parameter-group-name myparametergroup ^  
--parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":  
"mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

檢視僅供讀取複本的複寫篩選條件

您可以使用下列方式檢視僅供讀取複本的複寫篩選條件：

- 檢查與僅供讀取複本關聯之參數群組中複寫篩選參數的設定。

如需說明，請參閱「[檢視資料庫參數群組的參數值](#)」。

- 在 MariaDB 用戶端中，連線至僅供讀取複本並執行 SHOW REPLICA STATUS 陳述式。

在輸出中，下列欄位會顯示僅供讀取複本的複寫篩選條件：

- Replicate_Do_DB
- Replicate_Ignore_DB
- Replicate_Do_Table
- Replicate_Ignore_Table
- Replicate_Wild_Do_Table
- Replicate_Wild_Ignore_Table

如需有關這些欄位的詳細資訊，請參閱 MySQL 文件中的[檢查複寫狀態](#)。

Note

MariaDB 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MariaDB 是 10.5 之前的版本，請使用 SHOW SLAVE STATUS。

透過 MariaDB 設定延遲複寫

您可以將延遲複寫做為災難復原的策略。利用延遲複寫功能來指定最短時間 (以秒為單位)，即可延遲來源到僅供讀取複本的複寫作業。在發生損毀之時 (例如不小心刪除資料表)，若要快速完成事後復原作業，則請執行下列步驟：

- 停止僅供讀取複本的複寫作業，以免系統將造成損毀的變更內容傳送到該複本。

若要停止複寫作業，請使用 [mysql.rds stop replication](#) 預存程序。

- 若要將僅供讀取複本提升為新的來源資料庫執行個體，請參照[提升僅供讀取複本為獨立的資料庫執行個體](#)中的指示。

Note

- MariaDB 10.6 及更高版本支援延遲複寫。
- 使用預存程序來設定延遲複寫。您無法透過 AWS Management Console、AWS CLI 或 Amazon RDS API 設定延遲複寫。
- 您可以在延遲複寫組態中使用基於全域交易識別符 (GTID) 的複寫。

主題

- [在僅供讀取複本建立期間設定延遲複寫](#)
- [修改現有僅供讀取複本的延遲複寫](#)
- [提升僅供讀取複本](#)

在僅供讀取複本建立期間設定延遲複寫

若要為任何未來從資料庫執行個體建立的僅供讀取複本設定延遲複寫，請利用 [mysql.rds_set_configuration](#) 參數來執行 target delay 預存程序。

在僅供讀取複本建立期間設定延遲複寫

1. 透過 MariaDB 用戶端，以主要使用者的身分連接至 MariaDB 資料庫執行個體，該執行個體將成為僅供讀取複本的來源。
2. 利用 [mysql.rds_set_configuration](#) 參數來執行 target delay 預存程序。

例如，您可以執行下列預存程序，藉此設定複寫至少會延遲一小時 (3600 秒)，並將該設定套用至目前資料庫執行個體所建立的任何僅供讀取複本。

```
call mysql.rds_set_configuration('target delay', 3600);
```

Note

一旦執行此預存程序，系統即會針對任何透過 AWS CLI 或 Amazon RDS API 建立的僅供讀取複本，將複寫延遲時間設定為指定的秒數。

修改現有僅供讀取複本的延遲複寫

若要修改現有僅供讀取複本的延遲複寫，請執行 [mysql.rds_set_source_delay](#) 預存程序。

修改現有僅供讀取複本的延遲複寫

1. 透過 MariaDB 用戶端，以主要使用者的身分連接至僅供讀取複本。
2. 使用 [mysql.rds_stop_replication](#) 預存程序來停止複寫作業。
3. 執行 [mysql.rds_set_source_delay](#) 預存程序。

例如，您可以執行下列預存程序，藉此設定複寫至少會延遲一小時 (3600 秒)，並將該設定套用至僅供讀取複本。

```
call mysql.rds_set_source_delay(3600);
```

4. 使用 [mysql.rds_start_replication](#) 預存程序來啟動複寫作業。

提升僅供讀取複本

使用災難復原功能時，您可以參照中的指示，在複寫作業停止後將僅供讀取複本提升為新的來源資料庫執行個體。如需提升僅供讀取複本的相關資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

使用 MariaDB 更新僅供讀取複本

僅供讀取複本專門用來支援讀取查詢，但建議您不定期進行更新。例如，您可能需要新增索引來加快特定查詢類型存取複本的速度。您可以在僅供讀取複本的資料庫參數群組中將 `read_only` 參數設定為 0，以啟用更新。

使用 MariaDB 處理多可用區域僅供讀取複本的部署

您可以從單一可用區域或多可用區域的資料庫執行個體部署建立僅供讀取複本。您可使用異地同步備份部署來改善重要資料的耐用性和可用性，但無法使用異地同步備份部署來處理僅供讀取的查詢。反之，您可從高流量的多可用區域資料庫執行個體建立僅供讀取複本，藉此卸載僅供讀取的查詢。若異地同步

備份部署的來源執行個體容錯移轉為次要，任何相關聯的僅供讀取複本會自動將複寫作業來源切換成次要 (現為主要)。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。

您可以建立僅供讀取複本並用作多可用區域資料庫執行個體。Amazon RDS 會在另一個可用區域中建立您的複本的待命複本，以獲得該複本的容錯移轉支援。建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。

搭配 RDS for MariaDB 使用階層式僅供讀取複本

RDS for MariaDB 支援階層式僅供讀取複本。利用階層式僅供讀取複本，您可以擴展讀取，無須增加來源 RDS for MariaDB 資料庫執行個體的額外負荷。

利用階層式僅供讀取複本，您的 RDS for MariaDB 資料庫執行個體會將資料傳送至鏈結中的第一個僅供讀取複本。之後，該僅供讀取複本會將資料傳送至鏈結中的第二個複本，依此類推。最終結果是鏈結中的所有僅供讀取複本皆具有來自 RDS for MariaDB 資料庫執行個體的變更，但並無僅在來源資料庫執行個體上的額外負荷。

您可從來源 RDS for MariaDB 資料庫執行個體的鏈結中建立一系列最多三個僅供讀取複本。例如，假設您有一個 RDS for MariaDB 資料庫執行個體 `mariadb-main`。您可以執行下列作業：

- 從 `mariadb-main` 開始，建立鏈結中的第一個僅供讀取複本 `read-replica-1`。
- 接下來，從 `read-replica-1`，建立鏈結中的下一個僅供讀取複本 `read-replica-2`。
- 最後，從 `read-replica-2`，建立鏈結中的第三個僅供讀取複本 `read-replica-3`。

除了 `mariadb-main` 系列中的第三個階層式僅供讀取複本之外，您無法建立另一個僅供讀取複本。從 RDS for MariaDB 來源資料庫執行個體至一系列階層式僅供讀取複本尾端的完整執行個體系列最多可包含四個資料庫執行個體。

若要使階層式僅供讀取複本可以運作，每個來源 RDS for MariaDB 資料庫執行個體都必須開啟自動備份。若要在僅供讀取複本上開啟自動備份，首先建立該僅供讀取複本，然後修改該複本來開啟自動備份。如需詳細資訊，請參閱 [建立僅供讀取複本](#)。

與任何僅供讀取複本一樣，您可提升作為階層式一部分的僅供讀取複本。從僅供讀取複本鏈結中提升僅供讀取複本將會從鏈結中移除該僅供讀取複本。例如，假設您想要將部分工作負載從 `mariadb-main` 資料庫執行個體移至新的執行個體，僅供會計部門使用。假設範例中的三個僅供讀取複本鏈結，您決定提升 `read-replica-2`。該鏈或受到下列影響：

- 提升 `read-replica-2` 會將其從複寫鏈結中移除。
 - 其現在是一個完整的讀取/寫入資料庫執行個體。

- 這會持續複寫至 read-replica-3，就像在提升之前所做的一樣。
- 您的 mariadb-main 會持續複寫至 read-replica-1。

如需提升僅供讀取複本的相關詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

監控 MariaDB 僅供讀取複本

對於 MariaDB 僅供讀取複本，您可以 CloudWatch 透過檢視 Amazon RDS 指標來監控 Amazon 中的複寫延遲。ReplicaLagReplicaLag 指標會回報 Seconds_Behind_Master 命令的 SHOW REPLICA STATUS 欄位值。

Note

MariaDB 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MariaDB 是 10.5 之前的版本，請使用 SHOW SLAVE STATUS。

MariaDB 複寫延遲的常見原因如下：

- 網路停機。
- 寫入僅供讀取複本上的資料表仍有索引。若未將僅供讀取複本的 read_only 參數設為 0，可能會中斷複寫。
- 使用非交易儲存引擎 (例如 MyISAM)。複寫功能僅支援 MariaDB 上的 InnoDB 儲存引擎。

當 ReplicaLag 指標到達 0，複本即已跟上來源資料庫執行個體。如果 ReplicaLag 指標傳回 -1，表示複寫目前為非作用中。ReplicaLag = -1 相當於 Seconds_Behind_Master = NULL。

透過 MariaDB 僅供讀取複本啟動與停用複寫

您可呼叫系統預存程序 [mysql.rds_stop_replication](#) 和 [mysql.rds_start_replication](#)，藉此停止並重新啟動 Amazon RDS 資料庫執行個體的複寫程序。針對長時間操作 (如建立大型索引) 在兩個 Amazon RDS 執行個體間進行複寫時，即可採取此操作。匯入或匯出資料庫時，也必須停止並啟動複寫作業。如需詳細資訊，請參閱 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 及 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

若複寫作業停止連續超過 30 天 (不論手動停止或由於複寫錯誤)，Amazon RDS 會結束來源資料庫執行個體和所有僅供讀取複本間的複寫作業。這樣做的原因在於，避免來源資料庫執行個體的儲存需求增加

以及長期容錯移轉。僅供讀取複本資料庫執行個體仍可供使用。然而，無法恢復複寫作業，因為在複寫作業結束後，僅供讀取複本所需的二進位日誌已從來源資料庫執行個體刪除。您可建立新的僅供讀取複本供來源資料庫執行個體使用，以重新建立複寫作業。

對 MariaDB 僅供讀取複本的問題進行故障診斷

MariaDB 的複寫技術採取非同步，因此，來源資料庫執行個體的 BinLogDiskUsage 和僅供讀取複本上的 ReplicaLag 預料會偶爾增加。例如，來源資料庫執行個體可同時出現大量寫入操作。相對而言，僅供讀取複本的寫入操作則使用單一 I/O 執行緒序列化，這可能導致來源執行個體和僅供讀取複本之間的延遲。如需 MariaDB 文件中的僅供讀取複本詳細資訊，請造訪[複寫概觀](#)。

您可執行下列動作，減少來源資料庫執行個體的更新與僅供讀取複本的后續更新間的延遲：

- 將僅供讀取複本的儲存體大小和資料庫執行個體類別大小，設定為等同於來源資料庫執行個體。
- 確保來源資料庫執行個體和僅供讀取複本所使用之資料庫參數群組中的參數設定相容。如需詳細資訊和範例，請參閱本節稍後討論的 `max_allowed_packet` 參數。

Amazon RDS 會監控複寫您的僅供讀取複本的狀態，並在複寫因任何原因停止時，將僅供讀取複本執行個體上的 Replication State (複寫狀態) 欄位更新為 Error (錯誤)。一個可能的範例為，僅供讀取複本上執行的 DML 查詢，與來源資料庫執行個體的更新相衝突。

您可以檢視 Replication Error 欄位，藉此檢閱 MariaDB 引擎擲出之相關聯錯誤的詳細資訊。也會產生指出僅供讀取複本之狀態的事件，包括 [RDS-EVENT-0045](#)、[RDS-EVENT-0046](#) 和 [RDS-EVENT-0047](#)。如需事件和訂閱事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。如果傳回了 MariaDB 錯誤訊息，請在 [MariaDB 錯誤訊息文件](#) 中檢閱錯誤。

其中一個常見問題就是僅供讀取複本的 `max_allowed_packet` 參數值小於來源資料庫執行個體的 `max_allowed_packet` 參數值，因而導致複寫錯誤。`max_allowed_packet` 參數是能夠在資料庫參數群組中設定的自訂參數，專用來指定可在資料庫上執行的 DML 程式碼大小上限。在部分案例中，與來源資料庫執行個體建立關聯之資料庫參數群組的 `max_allowed_packet` 參數值，會小於與來源僅供讀取複本建立關聯之資料庫參數群組的 `max_allowed_packet` 參數值。以這些情況而言，複寫程序可能會擲出錯誤 (Packet 大於 `max_allowed_packet` 位元組)，並停止複寫作業。您可以透過讓來源和僅供讀取複本使用具有相同 `max_allowed_packet` 參數值的資料庫參數群組，藉此修正錯誤。

可能造成複寫錯誤的其他常見情況包括下列：

- 寫入僅供讀取複本上的資料表。如果您在僅供讀取複本上建立索引，您需要將 `read_only` 參數設為 0 才能建立索引。如果您要寫入僅供讀取複本上的資料表，可能中斷複寫。

- 使用非交易式存放引擎 (例如 MyISAM)。僅供讀取複本需要交易式存放引擎。複寫功能僅支援 MariaDB 上的 InnoDB 儲存引擎。
- 使用不安全的非確定性查詢 (例如 SYSDATE())。如需詳細資訊，請參閱[二進位日誌中安全和不安全陳述式的判定](#)。

若您認為可安全略過錯誤，請遵循 [略過目前複寫錯誤](#) 中所述的步驟。否則，您可以刪除僅供讀取複本，並使用相同的資料庫執行個體識別符建立執行個體，讓端點與您的舊僅供讀取複本端點保持相同。如果複寫錯誤已修復，Replication State (複寫狀態) 會變更為 replicating (複寫中)。

在某些情況下，如果程序失敗的期間未排清部分二進位日誌 (binlog) 事件，便無法將 MariaDB 資料庫執行個體的僅供讀取複本切換至次要資料庫。在這種情況下，手動刪除並重新建立僅供讀取複本。如果要降低此情況發生的機率，則可設定以下參數值：sync_binlog=1 和 innodb_flush_log_at_trx_commit=1。但這些設定可能會降低效能，所以請先測試這些設定所帶來的影響，再於生產環境中實作變更內容。

使用外部來源執行個體設定 GTID 式複寫

您可以根據全域交易識別符 (GTID) 從 10.0.24 版或更新版本的外部 MariaDB 執行個體設定複寫到 RDS for MariaDB 資料庫執行個體。在 Amazon RDS 上設定外部來源執行個體和複本時，按照這些準則操作：

- 監控 RDS for MariaDB 資料庫執行個體 (您的複本) 的容錯移轉事件。若發生容錯移轉，資料庫執行個體 (您的複本) 可能會以不同的網路地址，在新主機上重新建立。如需如何監控容錯移轉事件的資訊，請參閱[使用 Amazon RDS 事件通知](#)。
- 確定已將二進位日誌 (binlogs) 套用至複本前，都要將二進位日誌保留在來源執行個體上。如此一來，發生故障時，您就可以還原來源執行個體。
- 對 Amazon RDS 上的 MariaDB 資料庫執行個體啟用自動備份。如果您需要重新同步來源執行個體與複本，開啟自動備份功能可確保您能夠將複本還原至特定時間點。如需備份和時間點還原的相關資訊，請參閱[備份、還原和匯出資料](#)。

Note

在 MariaDB 資料庫執行個體上啟動複寫所需的許可受到限制，並且不授予 Amazon RDS 主要使用者。因此，您必須使用 Amazon RDS [mysql.rds_set_external_master_gtid](#) 和 [mysql.rds_start_replication](#) 命令，以設定即時資料庫與您的 RDS for MariaDB 資料庫之間的複寫。

若要在外部來源執行個體與 Amazon RDS 上的 MariaDB 資料庫執行個體之間開始複寫，請使用下列程序。

啟動複寫

1. 將來源 MariaDB 執行個體設成唯讀：

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. 取得外部 MariaDB 執行個體的目前 GTID。在作法上可使用 mysql 或您選擇的查詢編輯器來執行 `SELECT @@gtid_current_pos;`

GTID 的格式為 `<domain-id>-<server-id>-<sequence-id>`。一般 GTID 類似於

0-1234510749-1728。如需 GTID 及其組成部分的詳細資訊，請參閱 MariaDB 文件中的[全域交易 ID](#)。

3. 使用 `mysqldump` 從外部 MariaDB 執行個體將資料庫複製到 MariaDB 資料庫執行個體。如果是非常大型的資料庫，您可能需要使用 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 中的程序。

對於LinuxmacOS、或Unix：

```
mysqldump \
  --databases database_name \
  --single-transaction \
  --compress \
  --order-by-primary \
  -u local_user \
  -plocal_password | mysql \
  --host=hostname \
  --port=3306 \
  -u RDS_user_name \
  -pRDS_password
```

在Windows中：

```
mysqldump ^
  --databases database_name ^
  --single-transaction ^
  --compress ^
  --order-by-primary \
```

```
-u local_user \  
-plocal_password | mysql ^  
  --host=hostname ^  
  --port=3306 ^  
-u RDS_user_name ^  
-pRDS_password
```

Note

請注意 -p 選項與輸入的密碼之間不能有空格。
指定此處所顯示提示以外的密碼，作為安全最佳實務。

在 --host 命令中，使用 --user (-u)、--port、-p 和 mysql 選項來指定主機名稱、使用者名稱、連接埠和密碼，以連接至 MariaDB 資料庫執行個體。主機名稱是來自 MariaDB 資料庫執行個體端點的 DNS 名稱，例如 myinstance.123456789012.us-east-1.rds.amazonaws.com。端點值可在 Amazon RDS 管理主控台的執行個體詳細資訊中找到。

4. 將來源 MariaDB 執行個體重新設為可寫入。

```
mysql> SET GLOBAL read_only = OFF;  
mysql> UNLOCK TABLES;
```

5. 在 Amazon RDS 管理主控台，將代管外部 MariaDB 資料庫之伺服器的 IP 地址，新增至 MariaDB 資料庫執行個體的 VPC 安全群組。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。

IP 地址會在滿足下列條件時變更：

- 您使用公有 IP 地址供外部來源執行個體和資料庫執行個體通訊。
- 外部來源執行個體停止後又重新啟動。

如果符合這些條件，請先驗證 IP 地址再新增它。

您可能還需要設定本機網路，以允許從 MariaDB 資料庫執行個體的 IP 地址建立的連線，使之能夠與外部 MariaDB 執行個體進行通訊。若要尋找 MariaDB 資料庫執行個體的 IP 地址，請使用 host 命令。

```
host db_instance_endpoint
```

主機名稱是來自 MariaDB 資料庫執行個體端點的 DNS 名稱。

6. 使用您選擇的用戶端，連接至外部 MariaDB 執行個體，並建立用於複寫的 MariaDB 使用者。此帳戶只供複寫作業使用，務必限制其存取您的網域，以提升安全性。以下是範例。

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

7. 若為外部 MariaDB 執行個體，請將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。舉例來說，若要將所有資料庫的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的「*repl_user*」使用者，請發出下列命令。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. 將 MariaDB 資料庫執行個體變成複本。以主要使用者身分連接至 MariaDB 資料庫執行個體，然後使用 [mysql.rds_set_external_master_gtid](#) 命令，將外部 MariaDB 資料庫識別為複寫來源執行個體。使用您在步驟 2 決定的 GTID。以下是範例。

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,  
'repl_user', 'password', 'GTID', 0);
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

9. 在 MariaDB 資料庫執行個體上，發出 [mysql.rds_start_replication](#) 命令來開始複寫。

```
CALL mysql.rds_start_replication;
```

使用外部來源執行個體設定二進位日誌檔案位置複寫

您可使用二進位日誌檔案複寫，在 RDS for MySQL 或 MariaDB 資料庫執行個體，以及位於 Amazon RDS 外部的 MySQL 或 MariaDB 執行個體之間，設定複寫。

主題

- [開始之前](#)
- [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)

開始之前

您可以使用複寫交易的二進制日誌檔案位置設定複寫。

在 Amazon RDS 資料庫執行個體上啟動複寫功能所需的許可有其限制，不供 Amazon RDS 主要使用者使用。因此，請確定使用 Amazon RDS [mysql.rds_set_external_master](#) 和 [mysql.rds_start_replication](#) 命令，設定即時資料庫與 Amazon RDS 資料庫之間的複寫作業。

若要為 MySQL 或 MariaDB 資料庫設定二進位日誌記錄格式，請更新 `binlog_format` 參數。如果資料庫執行個體使用預設的資料庫執行個體參數群組，請建立新的資料庫參數群組來修改 `binlog_format` 設定。建議您在 `binlog_format` 中使用預設設定，亦即 MIXED。不過，如果需要特定的二進位日誌 (binlog) 格式，也可以將 `binlog_format` 設定為 ROW 或 STATEMENT。重新啟動資料庫執行個體，讓變更生效。

如需有關設定 `binlog_format` 參數的資訊，請參閱[設定適用於 MySQL RDS](#)。如需各種 MySQL 複寫類型隱含意義的相關資訊，請參閱 MySQL 文件中的[基於陳述式和基於列的複寫的優缺點](#)。

Note

從適用於 MySQL 的 RDS 版本 8.0.36 開始，Amazon RDS 不會複寫資料庫。mysql 因此，如果 Amazon RDS 複本上需要外部資料庫上有使用者，請務必手動建立這些使用者。

使用外部來源執行個體設定二進位日誌檔案位置複寫

在 Amazon RDS 上設定外部來源執行個體和複本時，按照這些準則操作：

- 監控 Amazon RDS 資料庫執行個體 (您的複本) 的容錯移轉事件。若發生容錯移轉，資料庫執行個體 (您的複本) 可能會以不同的網路地址，在新主機上重新建立。如需如何監控容錯移轉事件的資訊，請參閱[使用 Amazon RDS 事件通知](#)。

- 直到您已驗證二進位日誌已套用至複本前，都要將二進位日誌保存在來源執行個體上。如此一來，發生故障時，這種維護可確保您能夠還原來源執行個體。
- 開啟 Amazon RDS 資料庫執行個體的自動備份功能。如果您需要重新同步來源執行個體與複本，開啟自動備份功能可確保您能夠將複本還原至特定時間點。如需備份與 point-in-time 還原的資訊，請參閱[備份、還原和匯出資料](#)。

使用外部來源執行個體設定二進位日誌檔案複寫

1. 將來源 MySQL 或 MariaDB 執行個體設成唯讀狀態。

```
mysql> FLUSH TABLES WITH READ LOCK;  
mysql> SET GLOBAL read_only = ON;
```

2. 在來源 MySQL 或 MariaDB 執行個體上執行 SHOW MASTER STATUS 命令，以確定二進位日誌的位置。

您會獲得類似下列範例的輸出。

```
File                Position  
-----  
mysql-bin-changelog.000031    107  
-----
```


3. 使用 mysqldump，從外部執行個體將資料庫複製到 Amazon RDS 資料庫執行個體。如果是非常大型的資料庫，您可能需要使用 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 中的程序。

對於LinuxmacOS、或Unix：

```
mysqldump --databases database_name \  
  --single-transaction \  
  --compress \  
  --order-by-primary \  
  -u local_user \  
  -plocal_password | mysql \  
  --host=hostname \  
  --port=3306 \  
  -u RDS_user_name \  
  -pRDS_password
```


在 Windows 中：

```
mysqldump --databases database_name ^  
  --single-transaction ^  
  --compress ^  
  --order-by-primary ^  
  -u local_user ^  
  -plocal_password | mysql ^  
    --host=hostname ^  
    --port=3306 ^  
  -u RDS_user_name ^  
  -pRDS_password
```

 Note

請注意 -p 選項與輸入的密碼之間不能有空格。

若要指定連線到 Amazon RDS 資料庫執行個體的主機名稱、使用者名稱、連接埠和密碼，請使用 --host 命令中的 --user (-u)、--port、-p 和 mysql 選項。主機名稱是 Amazon RDS 資料庫執行個體端點的網域名稱服務 (DNS) 名稱，例如 myinstance.123456789012.us-east-1.rds.amazonaws.com。您可在 AWS Management Console 的執行個體詳細資訊中找到端點值。

4. 將來源 MySQL 或 MariaDB 執行個體重新設為可寫入狀態。

```
mysql> SET GLOBAL read_only = OFF;  
mysql> UNLOCK TABLES;
```

如需如何製作備份以搭配複寫作業使用的詳細資訊，請參閱 [MySQL 文件](#)。

5. 在中 AWS Management Console，將託管外部資料庫的伺服器 IP 位址新增到 Amazon RDS 資料庫執行個體的虛擬私有雲端 (VPC) 安全群組。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。

IP 地址會在滿足下列條件時變更：

- 您使用公有 IP 地址供外部來源執行個體和資料庫執行個體通訊。
- 外部來源執行個體停止後又重新啟動。

如果符合這些條件，請先驗證 IP 地址再新增它。

您可能還需要設定本機網路，以允許從 Amazon RDS 資料庫執行個體的 IP 地址建立連線。如此，您的本機網路就能與外部 MySQL 或 MariaDB 執行個體進行通訊。若要尋找 Amazon RDS 資料庫執行個體的 IP 地址，請使用 `host` 命令。

```
host db_instance_endpoint
```

主機名稱是 Amazon RDS 資料庫執行個體端點的 DNS 名稱。

6. 使用您選擇的用戶端連線至外部執行個體，接著建立執行複寫作業所需的使用者。只對於複寫作業使用此帳戶，並限制其存取您的網域，以提升安全性。以下是範例。

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

7. 若為外部執行個體，請將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。舉例來說，若要將所有資料庫的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的「`repl_user`」使用者，請發出下列命令。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. 將 Amazon RDS 資料庫執行個體設為複本。若要執行此操作，請先以主要使用者身分連線至 Amazon RDS 資料庫執行個體。使用 [mysql.rds_set_external_master](#) 命令找到外部 MySQL 或 MariaDB 資料庫，將其視為來源執行個體。使用您在步驟 2 中所確定的主控端日誌檔案名稱與主控端日誌位置。以下是範例。

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,  
'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

Note

在 RDS for MySQL 上，您可以改為執行 [mysql.rds_set_external_master_with_delay](#) 預存程序，進而選擇使用延遲複寫。在 RDS for MySQL 上，使用延遲複寫的原因之一，是為

了透過 [mysql.rds_start_replication_until](#) 預存程序來開啟災難復原功能。目前，RDS for MariaDB 支援延遲複寫，但不支援 `mysql.rds_start_replication_until` 程序。

9. 在 Amazon RDS 資料庫執行個體上發出 [mysql.rds_start_replication](#) 命令，啟動複寫作業。

```
CALL mysql.rds_start_replication;
```

MariaDB 資料庫引擎的選項

以下說明可供執行 MariaDB 資料庫引擎之 Amazon RDS 執行個體使用的選項或其他功能。若要開啟這些選項，請將它們新增至自訂選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需使用選項群組的相關詳細資訊，請參閱[使用選項群組](#)。

Amazon RDS 支援 MariaDB 的下列選項：

選項 ID	引擎版本
MARIADB_AUDIT_PLUGIN	MariaDB 10.3 及更新版本

MariaDB 稽核外掛程式支援

Amazon RDS 支援在 MariaDB 資料庫執行個體上使用 MariaDB 稽核外掛程式。MariaDB 稽核外掛程式會記錄資料庫活動，例如登入資料庫的使用者、針對資料庫執行的查詢，以及其他。資料庫活動的記錄會儲存在日誌檔中。

稽核外掛程式選項設定


Amazon RDS 支援 MariaDB 稽核外掛程式的下列設定。

Note

如果您未在 RDS 主控台中設定選項設定，RDS 會使用預設設定。

選項設定	有效值	預設值	描述
SERVER_AUDIT_FILE_PATH	/rdsdbdata/log/audit/	/rdsdbdata/log/audit/	日誌檔的位置。日誌檔包含 SERVER_AUDIT_EVENTS 中指定之活動的記錄。如需更多詳細資訊，請參閱 檢視並列出資料庫日誌檔案 及 MariaDB 資料庫日誌檔案 。
SERVER_AUDIT_FILE_SIZE	1-100000000	1000000	達到時導致檔案輪換的大小 (以位元組表示)。如需更多詳細資訊，請參閱 日誌檔案大小 。

選項設定	有效值	預設值	描述
ROTATE_SIZE			
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	要在 <code>server_audit_output_type=file</code> 時儲存的日誌輪換次數。如果設定為 0，則日誌檔永遠不會輪換。如需詳細資訊，請參閱 日誌檔案大小 及 下載資料庫日誌檔案 。
SERVER_AUDIT_EVENTS	CONNECT, QUERY, TABLE, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT, QUERY_DCL	CONNECT, QUERY	<p>要記錄在日誌中之活動的類型。安裝 MariaDB 稽核外掛程式是自行記錄的。</p> <ul style="list-style-type: none"> CONNECT：記錄與資料庫的成功和不成功連線，以及與資料庫的中斷連線。 QUERY：記錄針對資料庫執行之所有查詢的文字。 TABLE：記錄當針對資料庫執行查詢時受到查詢影響的資料表。 QUERY_DDL：類似於 QUERY 事件，但只傳回資料定義語言 (DDL) 查詢 (CREATE、ALTER 等)。 QUERY_DML：類似於 QUERY 事件，但只傳回資料處理語言 (DML) 查詢 (INSERT、UPDATE 和 SELECT 等)。 QUERY_DML_NO_SELECT：類似 QUERY_DML 事件，但不會記錄 SELECT 查詢。 QUERY_DCL：類似於 QUERY 事件，但只傳回資料控制語言 (DCL) 查詢 (GRANT、REVOKE 等)。

選項設定	有效值	預設值	描述
SERVER_AUDIT_INCL_USERS	多個以逗號分隔的值	無	只包括來自所指定使用者的活動。預設情況下，會記錄所有使用者的活動。SERVER_AUDIT_INCL_USERS 和 SERVER_AUDIT_EXCL_USERS 是相互排斥的。如果您將值新增至 SERVER_AUDIT_INCL_USERS ，請確保未將任何值新增至 SERVER_AUDIT_EXCL_USERS 。
SERVER_AUDIT_EXCL_USERS	多個以逗號分隔的值	無	<p>排除來自所指定使用者的活動。預設情況下，會記錄所有使用者的活動。SERVER_AUDIT_INCL_USERS 和 SERVER_AUDIT_EXCL_USERS 是相互排斥的。如果您將值新增至 SERVER_AUDIT_EXCL_USERS ，請確保未將任何值新增至 SERVER_AUDIT_INCL_USERS 。</p> <p>rdsadmin 使用者會每次查詢資料庫，以檢查資料庫的運作狀態。根據您的其他設定，此活動可能會導致您的日誌檔大小非常快速地變得非常大。如果您不需要記錄此活動，請將 rdsadmin 使用者新增至 SERVER_AUDIT_EXCL_USERS 清單。</p> <div data-bbox="829 1283 1507 1549" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>即使針對此選項設定指定了使用者，也會一律記錄所有使用者的 CONNECT 活動。</p> </div>
SERVER_AUDIT_LOGGING	ON	ON	記錄為作用中狀態。唯一有效的值為 ON。Amazon RDS 不支援停用記錄。如果想要停用記錄，請移除 MariaDB 稽核外掛程式。如需更多詳細資訊，請參閱 移除 MariaDB 稽核外掛程式 。

選項設定	有效值	預設值	描述
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	記錄中查詢字串長度的限制。

新增 MariaDB 稽核外掛程式

將 MariaDB 稽核外掛程式新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

在您新增 MariaDB 稽核外掛程式之後，不需要重新啟動資料庫執行個體。只要選項群組作用中，稽核就會立即開始。

新增 MariaDB 稽核外掛程式

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則，建立自訂的資料庫選項群組。為 Engine (引擎)，選擇 mariadb，並為 Major engine version (主要引擎版本) 選擇 10.3 或更新版本。如需更多詳細資訊，請參閱 [建立選項群組](#)。
2. 將 MARIADB_AUDIT_PLUGIN 選項新增至選項群組，並設定選項設定。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。如需每項設定的詳細資訊，請參閱 [稽核外掛程式選項設定](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體。
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改資料庫執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

檢視和下載 MariaDB 稽核外掛程式日誌

在啟用 MariaDB 稽核外掛程式之後，您可以採取與存取任何其他文字型日誌檔相同的方式來存取日誌檔中的結果。稽核日誌檔位於 `/rdsdbdata/log/audit/`。如需在主控台中檢視日誌檔的相關資訊，請參閱[檢視並列出資料庫日誌檔案](#)。如需下載日誌檔的相關資訊，請參閱[下載資料庫日誌檔案](#)。

修改 MariaDB 稽核外掛程式設定

在啟用 MariaDB 稽核外掛程式之後，您可以修改外掛程式的設定。如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。如需每項設定的詳細資訊，請參閱[稽核外掛程式選項設定](#)。

移除 MariaDB 稽核外掛程式

Amazon RDS 不支援在 MariaDB 稽核外掛程式中關閉記錄。不過，您可以從資料庫執行個體中移除外掛程式。當您移除 MariaDB 稽核外掛程式時，資料庫執行個體會自動重新啟動來停止稽核。

若要從資料庫執行個體中移除 MariaDB 稽核外掛程式，請執行下列其中一個動作：

- 從其所屬的選項群組中移除 MariaDB 稽核外掛程式。此變更會影響使用選項群組的所有資料庫執行個體。如需詳細資訊，請參閱 [從選項群組移除選項](#)
- 修改資料庫執行個體，並指定不包括外掛程式的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

MariaDB 的參數

依預設，MariaDB 資料庫執行個體會使用 MariaDB 資料庫專用的資料庫參數群組。此參數群組包含 MySQL 資料庫引擎的 Amazon RDS 資料庫參數群組中所包含的一部分參數 (而不是全部)。也包含許多新的 MariaDB 專用參數。如需使用參數群組和設定參數的相關資訊，請參閱[使用參數群組](#)。

檢視 MariaDB 參數

RDS for MariaDB 參數會設為您已選取之儲存引擎的預設值。如需 MariaDB 參數的詳細資訊，請參閱[MariaDB 文件](#)。如需 MariaDB 儲存引擎的詳細資訊，請參閱[Amazon RDS 上的 MariaDB 支援的儲存引擎](#)。

您可以使用 RDS 主控台或 AWS CLI 查看特定 RDS for MariaDB 版本的可用參數。如需有關在 RDS 主控台中查看 MariaDB 參數群組參數的資訊，請參閱[檢視資料庫參數群組的參數值](#)。

藉由 AWS CLI，您可以執行 [describe-engine-default-parameters](#) 命令來查看 RDS for MariaDB 版本的參數。為 `--db-parameter-group-family` 選項指定下列其中一個值：

- mariadb10.11
- mariadb10.6
- mariadb10.5
- mariadb10.4
- mariadb10.3

例如，若要檢視 RDS for MariaDB 10.6 版的參數，請執行下列命令。

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6
```

您的輸出結果類似以下內容。

```
{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "alter_algorithm",
        "Description": "Specify the alter table algorithm.",
        "Source": "engine-default",
        "ApplyType": "dynamic",
        "DataType": "string",
```



```

        "AllowedValues": "DEFAULT,COPY,INPLACE,NOCOPY,INSTANT",
        "IsModifiable": true
    },
    {
        "ParameterName": "analyze_sample_percentage",
        "Description": "Percentage of rows from the table ANALYZE TABLE will
sample to collect table statistics.",
        "Source": "engine-default",
        "ApplyType": "dynamic",
        "DataType": "float",
        "AllowedValues": "0-100",
        "IsModifiable": true
    },
    {
        "ParameterName": "aria_block_size",
        "Description": "Block size to be used for Aria index pages.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "integer",
        "AllowedValues": "1024-32768",
        "IsModifiable": false
    },
    {
        "ParameterName": "aria_checkpoint_interval",
        "Description": "Interval in seconds between automatic checkpoints.",
        "Source": "engine-default",
        "ApplyType": "dynamic",
        "DataType": "integer",
        "AllowedValues": "0-4294967295",
        "IsModifiable": true
    },
    ...

```

若要檢視 RDS for MariaDB 10.6 版的可修改參數，請執行下列命令。

對於LinuxmacOS、或Unix：

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 \
--query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

在Windows中：

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 ^
```

```
--query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

無法使用的 MySQL 參數

以下是未在 MariaDB 特定的資料庫參數群組中提供的 MySQL 參數：

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed
- gtid-mode
- gtid_next
- gtid_owned
- gtid_purged
- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements

- log_throttle_queries_not_using_indexes
- master-info-repository
- optimizer_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max
- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count
- validate_password_policy
- validate_password_special_char_count

如需 MySQL 參數的詳細資訊，請參閱 [MySQL 文件](#)。

將資料從 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體

您可以使用 AWS Management Console、AWS CLI 或 Amazon RDS API，將 RDS for MySQL 資料庫快照遷移至執行 MariaDB 的新資料庫執行個體。您必須使用從執行 MySQL 5.6 或 5.7 的 Amazon RDS 資料庫執行個體建立的資料庫快照。若要了解如何建立 RDS for MySQL 資料庫快照，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

遷移快照不會影響從中擷取快照的原始資料庫執行個體。您可以先測試和驗證新資料庫執行個體，再將流量轉移到新資料庫執行個體來替代原始資料庫執行個體。

在從 MySQL 遷移至 MariaDB 之後，MariaDB 資料庫執行個體將與預設資料庫參數群組和選項群組相關聯。在還原資料庫快照之後，您可以使自訂資料庫參數群組與新的資料庫執行個體產生關聯。不過，MariaDB 參數群組具有一組不同的可設定系統變數。如需 MySQL 系統變數與 MariaDB 系統變數之間差異的相關資訊，請參閱 [MariaDB 與 MySQL 之間的系統變數差異](#)。若要了解資料庫參數群組，請參閱 [使用參數群組](#)。若要了解選項群組，請參閱 [使用選項群組](#)。

執行遷移

您可以使用 AWS Management Console、AWS CLI 或 RDS API，將 RDS for MySQL 資料庫快照遷移至新的 MariaDB 資料庫執行個體。

主控台

將 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)，然後選取您要遷移的 MySQL 資料庫快照。
3. 針對 Actions (動作)，選擇 Migrate Snapshot (遷移快照)。Migrate Database (遷移資料庫) 頁面隨即出現。
4. 針對 Migrate to DB Engine (遷移至資料庫引擎)，選擇 mariadb。

Amazon RDS 會自動選取 DB engine version (資料庫引擎版本)。您無法變更資料庫引擎版本。

RDS > Snapshots > Migrate snapshot

Migrate database

Migrate this database to a new DB engine by selecting your desired options for the migrated instance.

Instance specifications

Migrate to DB engine
Name of the database engine

mariadb ▼

DB engine version
Version number of the database engine to be used for this instance

MariaDB 10.5.12 ▼

Settings

5. 在其餘區段，指定資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。
6. 選擇 Migrate (遷移)。

AWS CLI

若要將資料從 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體，請搭配下列參數使用 AWS CLI [restore-db-instance-from-db-snapshot](#) 命令：

- `--db-instance-identifier` — 要從資料庫快照建立的資料庫執行個體名稱。
- `--db-snapshot-identifier` — 要從中還原之資料庫快照集的識別碼。
- `--engine` – 要用於新執行個體的資料庫引擎。

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier newmariadbinstance \  
  --db-snapshot-identifier mysnapshot \  
  --engine mariadb
```

```
--db-snapshot-identifier mysqlsnapshot \  
--engine mariadb
```

在Windows中：

```
aws rds restore-db-instance-from-db-snapshot ^  
--db-instance-identifier newmariadbinstance ^  
--db-snapshot-identifier mysqlsnapshot ^  
--engine mariadb
```

API

若要將資料從 MySQL 資料庫快照遷移至 MariaDB 資料庫執行個體，請呼叫 Amazon RDS API 作業 [RestoreDBInstanceFromDBSnapshot](#)。

MariaDB 與 MySQL 之間的不相容性

MySQL 與 MariaDB 之間的不相容性包括下列情況：

- 您無法將利用 MySQL 8.0 建立的資料庫快照遷移至 MariaDB。
- 如果來源 MySQL 資料庫使用 SHA256 密碼雜湊，您必須重設 SHA256 雜湊的使用者密碼，才能連線到 MariaDB 資料庫。下列程式碼展示如何重設 SHA256 雜湊的密碼。

```
SET old_passwords = 0;  
UPDATE mysql.user SET plugin = 'mysql_native_password',  
Password = PASSWORD('new_password')  
WHERE (User, Host) = ('master_user_name', %);  
FLUSH PRIVILEGES;
```

- 如果您的 RDS 主要使用者帳戶使用 SHA-256 密碼雜湊，請確認使用 AWS Management Console、[modify-db-instance](#) AWS CLI 命令或 [ModifyDBInstance](#) RDS API 操作來重設密碼。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。
- MariaDB 不支援 Memcached 外掛程式。不過，Memcached 外掛程式所使用的資料會儲存為 InnoDB 資料表。在遷移 MySQL 資料庫快照之後，您可以使用 SQL 來存取 Memcached 外掛程式所使用的資料。如需 innodb_memcache 資料庫的詳細資訊，請參閱 [InnoDB memcached 外掛程式內部](#)。

Amazon RDS 上的 MariaDB SQL 參考

在下文中，您可以找到說明執行 MariaDB 資料庫引擎的 Amazon RDS 執行個體所適用系統預存程序的描述。

您可以使用 MySQL 資料庫執行個體和 MariaDB 資料庫執行個體可用的系統預存程序。[RDS for MySQL 預存程序參考](#) 中記載這些預存程序。MariaDB 資料庫執行個體支援 `mysql.rds_start_replication_until` 和 `mysql.rds_start_replication_until_gtid` 以外的所有預存程序。

此外，僅對執行 MariaDB 的 Amazon RDS 資料庫執行個體支援下列系統預存程序：

- [mysql.rds_replica_status](#)
- [mysql.rds_set_external_master_gtid](#)
- [mysql.rds_kill_query_id](#)

mysql.rds_replica_status

顯示 MariaDB 僅供讀取複本的複寫狀態。

在僅供讀取複本上呼叫此程序，以顯示複本執行緒基本參數的狀態資訊。

語法

```
CALL mysql.rds_replica_status;
```

使用須知

僅執行 MariaDB 10.5 版本及更新版本的 MariaDB 資料庫執行個體支援此程序。

此程序相當於 `SHOW REPLICA STATUS` 命令。MariaDB 10.5 版本和更新版本的資料庫執行個體不支援此命令。

在 MariaDB 的先前版本中，此相當的 `SHOW SLAVE STATUS` 命令需要 `REPLICATION SLAVE` 權限。在 MariaDB 10.5 及更高版本中，其需要 `REPLICATION REPLICA ADMIN` 權限。為了保護 MariaDB 10.5 及更高版本的資料庫執行個體的 RDS 管理，此新權限不會授予 RDS 主要使用者。

範例

下面的範例顯示了 MariaDB 僅供讀取複本的狀態：

```
call mysql.rds_replica_status;
```

回應類似如下：

```
***** 1. row *****
      Replica_IO_State: Waiting for master to send event
      Source_Host: XX.XX.XX.XXX
      Source_User: rdsrepladmin
      Source_Port: 3306
      Connect_Retry: 60
      Source_Log_File: mysql-bin-changelog.003988
      Read_Source_Log_Pos: 405
      Relay_Log_File: relaylog.011024
      Relay_Log_Pos: 657
      Relay_Source_Log_File: mysql-bin-changelog.003988
      Replica_IO_Running: Yes
      Replica_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
      Replicate_Wild_Do_Table:
      Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
      Exec_Source_Log_Pos: 405
      Relay_Log_Space: 1016
      Until_Condition: None
      Until_Log_File:
      Until_Log_Pos: 0
      Source_SSL_Allowed: No
      Source_SSL_CA_File:
      Source_SSL_CA_Path:
      Source_SSL_Cert:
      Source_SSL_Cipher:
      Source_SSL_Key:
      Seconds_Behind_Master: 0
      Source_SSL_Verify_Server_Cert: No
      Last_IO_Errno: 0
      Last_IO_Error:
      Last_SQL_Errno: 0
```



```
      Last_SQL_Error:
Replicate_Ignore_Server_Ids:
      Source_Server_Id: 807509301
      Source_SSL_Crl:
Source_SSL_Crlpath:
      Using_Gtid: Slave_Pos
      Gtid_IO_Pos: 0-807509301-3980
Replicate_Do_Domain_Ids:
Replicate_Ignore_Domain_Ids:
      Parallel_Mode: optimistic
      SQL_Delay: 0
      SQL_Remaining_Delay: NULL
Replica_SQL_Running_State: Reading event from the relay log
      Replica_DDL_Groups: 15
Replica_Non_Transactional_Groups: 0
      Replica_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)
```

mysql.rds_set_external_master_gtid

設定從在 Amazon RDS 外部執行的 MariaDB 執行個體到 MariaDB 資料庫執行個體的 GTID 型複寫。僅在外部 MariaDB 執行個體版本為 10.0.24 或更高版本時，才支援此存放程序。設定其中的一或兩個執行個體不支援 MariaDB 全域交易識別符 (GTID) 的複寫時，請使用 [mysql.rds_set_external_master](#)。

對複寫使用 GTID 可提供二進位日誌複寫未提供的損毀安全功能，因此建議您在複寫執行個體支援時加以使用。

語法

```
CALL mysql.rds_set_external_master_gtid(
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , gtid
  , ssl_encryption
);
```

參數

host_name

字串. 要成為來源執行個體、在 Amazon RDS 外部執行的 MariaDB 執行個體的主機名稱或 IP 地址。

host_port

整數. 要設定為來源執行個體、在 Amazon RDS 外部執行的 MariaDB 執行個體所使用的連接埠。如果網路組態包含會轉換連接埠號碼的 SSH 連接埠複寫，請指定 SSH 所公開的連接埠號碼。

replication_user_name

字串. MariaDB 資料庫執行個體中具有 REPLICATION SLAVE 許可的使用者 ID 要設定為僅供讀取複本。

replication_user_password

字串. replication_user_name 中指定之使用者 ID 的密碼。

gtid

字串. 來源執行個體上複寫應該從其開始的全域交易 ID。

在設定複寫時如果來源執行個體已遭鎖定，您可以使用 @@gtid_current_pos 來取得目前的 GTID，使得二進位日誌不會在您取得 GTID 和複寫開始的時間點之間變更。

否則，如果您使用 mysqldump 版本 10.0.13 或以上版本在複寫開始之前填入複本執行個體，您可以使用 --master-data 或 --dump-slave 選項，在輸出中取得 GTID 位置。如果未使用 mysqldump 版本 10.0.13 或以上版本，您可以執行 SHOW MASTER STATUS 或使用那些相同的 mysqldump 選項來取得二進位日誌檔案名稱和位置，然後透過在外部 MariaDB 執行個體上執行 BINLOG_GTID_POS，將它們轉換為 GTID：

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

如需 GTID MariaDB 實作的詳細資訊，請前往 MariaDB 文件中的[全域交易 ID](#)。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 MASTER_SSL_VERIFY_SERVER_CERT 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

使用須知

`mysql.rds_set_external_master_gtid` 程序必須由主要使用者執行。其必須在 MariaDB 資料庫執行個體上執行，您會將該執行個體設定為在 Amazon RDS 外部執行之 MariaDB 執行個體的複本。在執行 `mysql.rds_set_external_master_gtid` 之前，您必須在將 Amazon RDS 外部執行的 MariaDB 的執行個體設定為來源執行個體。如需更多詳細資訊，請參閱 [將資料匯入 MariaDB 資料庫執行個體](#)。

Warning

請勿使用 `mysql.rds_set_external_master_gtid` 來管理兩個 Amazon RDS 資料庫執行個體之間的複寫。只有在使用在 RDS 外部執行的 MariaDB 執行個體進行複寫時才使用它。如需 Amazon RDS 資料庫執行個體之間管理複寫的相關資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

呼叫 `mysql.rds_set_external_master_gtid` 將 Amazon RDS 資料庫執行個體設定為僅供讀取複本之後，您可以呼叫複本上的 [mysql.rds_start_replication](#) 來啟動複寫程序。您可以呼叫 [mysql.rds_reset_external_master](#) 來移除僅供讀取複本組態。

呼叫 `mysql.rds_set_external_master_gtid` 時，Amazon RDS 將時間、使用者和「設定主控」的動作記錄在 `mysql.rds_history` 和 `mysql.rds_replication_status` 資料表中。

範例

在 MariaDB 資料庫執行個體上執行時，下列範例會將它設定為在 Amazon RDS 外部執行之 MariaDB 執行個體的複本。

```
call mysql.rds_set_external_master_gtid
('Sourcedb.some.com',3306,'ReplicationUser','SomePassW0rd','0-123-456',0);
```

mysql.rds_kill_query_id

結束對 MariaDB 伺服器執行的查詢。

語法

```
CALL mysql.rds_kill_query_id(queryID);
```

參數

queryID

整數. 要結束之查詢的身分。

使用須知

若要停止對 MariaDB 伺服器執行的查詢，請使用 `mysql.rds_kill_query_id` 程序並傳入該查詢的 ID。若要取得查詢 ID，請查詢 MariaDB [資訊結構描述 PROCESLIST 資料表](#)，如下所示：

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM  
INFORMATION_SCHEMA.PROCESLIST WHERE USER = '<user name>';
```

MariaDB 伺服器的連線會保留。

範例

下列範例會結束查詢 ID 為 230040 的查詢：

```
call mysql.rds_kill_query_id(230040);
```

MariaDB 資料庫執行個體的本機時區

MariaDB 資料庫執行個體的時區預設為國際標準時間 (UTC)。您也可針對應用程式，將資料庫執行個體的時區設為本機時區。

若要設定資料庫執行個體的本機時區，請將資料庫執行個體參數群組中的 `time_zone` 參數設為本節稍後所列任一支援的值。當您設定參數群組的 `time_zone` 參數後，原本使用該參數群組的所有資料庫執行個體和僅供讀取複本會改為使用新的本機時區。如需設定參數群組參數的相關資訊，請參閱[使用參數群組](#)。

設定本機時區後，所有資料庫的新連線都會反映此變更。如果在您變更本機時區時有任何已開啟的連線，在您關閉連線然後開啟新連線後，才會看見本機時區更新。

您可為資料庫執行個體以及其中一或多個僅供讀取複本，設定不同的本機時區。若要執行這項操作，請為該資料庫執行個體和僅供讀取複本使用不同的參數群組，然後將各個參數群組中的 `time_zone` 參數設定為不同的本機時區。

如果您正在執行跨 AWS 區域 複寫作業，來源資料庫執行個體和僅供讀取複本會使用不同的參數群組 (參數群組為 AWS 區域 專屬)。若要在每個執行個體使用相同的本機時區，您必須設定執行個體和僅供讀取複本之參數群組中的 `time_zone` 參數。

從資料庫快照還原資料庫執行個體時，本機時區會設為 UTC。您也可在還原作業完成後，將時區更新為本機時區。如果您將資料庫執行個體還原至某個時間點，該還原資料庫執行個體的本機時區會使用還原資料庫執行個體參數群組中的時區設定。

網際網路號碼分配局 (IANA) 每年會在 <https://www.iana.org/time-zones> 發布數次新時區。每次 RDS 發布 MariaDB 的新次要維護版本時，它都會在發布時隨附最新的時區資料。當您使用最新的 RDS for MariaDB 版本時，您會有來自 RDS 的最新時區資料。為確保您的資料庫執行個體擁有最新的時區資料，建議您升級至更高的資料庫引擎版本。或者，您也可以手動修改 MariaDB 資料庫執行個體中的時區表。若要這麼做，您可以使用 SQL 命令或在 SQL 用戶端執行 [mysql_tzinfo_to_sql tool](#)。手動更新時區資料後，請重新啟動資料庫執行個體以讓變更生效。RDS 不會修改或重設執行中資料庫執行個體的時區資料。只有在您執行資料庫引擎版本升級時，才會安裝新的時區資料。

您可以將本機時區設為下列任一值。

Africa/Cairo	Asia/Riyadh
Africa/Casablanca	Asia/Seoul
Africa/Harare	Asia/Shanghai

Africa/Monrovia	Asia/Singapore
Africa/Nairobi	Asia/Taipei
Africa/Tripoli	Asia/Tehran
Africa/Windhoek	Asia/Tokyo
America/Araguaina	Asia/Ulaanbaatar
America/Asuncion	Asia/Vladivostok
America/Bogota	Asia/Yakutsk
America/Buenos_Aires	Asia/Yerevan
America/Caracas	Atlantic/Azores
America/Chihuahua	Australia/Adelaide
America/Cuiaba	Australia/Brisbane
America/Denver	Australia/Darwin
America/Fortaleza	Australia/Hobart
America/Guatemala	Australia/Perth
America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens

America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa
Asia/Karachi	US/Alaska
Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

RDS for MariaDB 的已知問題和限制

下列項目是使用 RDS for MariaDB 時的已知問題和限制。

Note

此清單並不詳盡。

主題

- [Amazon RDS 中的 MariaDB 檔案大小限制](#)
- [InnoDB 保留字詞](#)
- [自訂連接埠](#)
- [Performance Insights \(績效詳情\)](#)

Amazon RDS 中的 MariaDB 檔案大小限制

針對 MariaDB 資料庫執行個體，使用 InnoDB file-per-table 資料表空間時，資料表大小上限為 16 TB。此限制也會將系統資料表空間限制為最大 16 TB。InnoDB file-per-table 資料表空間 (其中每個資料表都在自己的資料表空間中) 會由 MariaDB 資料庫執行個體的預設設定。此上限與 MariaDB 資料庫執行個體的儲存空間上限無關。如需儲存空間限制的詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。

使用 InnoDB file-per-table 資料表空間有優缺點，取決於您的應用程式。如要判斷您應用程式的最佳方式，請前往 MySQL 文件中的 [File-Per-Table 表格空間](#)。

不建議允許資料表成長至檔案大小上限。一般來說，更理想的做法是將資料表分割成更小的資料表，如此可以改善效能並縮短復原時間。

您可以用於將大型資料表分成更小資料表的選項為分割。「分割」會根據您指定的規則，將大型資料表的部分分散成個別檔案。例如，如果您依日期儲存交易，則可以建立分割規則，使用分割將較舊的交易分配成個別檔案。然後，您可以定期封存不需要立即可供應用程式使用的歷史交易資料。如需詳細資訊，請參閱 MySQL 文件中的 [分割](#)。

決定所有 InnoDB 資料表空間的大小

- 使用下列 SQL 命令來判斷您是否有太大的資料表，且為進行分割的候選者。

Note

若為 MariaDB 10.6 及更新版本，此查詢也會傳回 InnoDB 系統資料表空間的大小。若為早於 10.6 的 MariaDB 版本，您無法透過查詢系統資料表來決定 InnoDB 系統資料表空間的大小。建議您升級到更新的版本。

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNO_DB_SYS_TABLESPACES ORDER BY 3 DESC;
```

決定非 InnoDB 使用者資料表的大小

- 使用下列 SQL 命令，判斷您的任何非 InnoDB 使用者資料表是否太大。

```
SELECT TABLE_SCHEMA, TABLE_NAME, round((((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

啟用 InnoDB file-per-table 資料表空間

- 在資料庫執行個體的參數群組中，將 `innodb_file_per_table` 參數設為 1。

停用 InnoDB file-per-table 資料表空間

- 在資料庫執行個體的參數群組中，將 `innodb_file_per_table` 參數設為 0。

如需更新參數群組的相關資訊，請參閱[使用參數群組](#)。

已啟用或停用 InnoDB file-per-table 資料表空間時，您可以發出 ALTER TABLE 命令。您可以使用此命令，將資料表從全域資料表空間移動到其自己的資料表空間。或者，您可以將資料表從其自身的資料表空間移動到全域資料表空間。以下是範例。

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

InnoDB 保留字詞

InnoDB 是用於 RDS for MariaDB 的保留字詞。此名稱無法用於 MariaDB 資料庫。

自訂連接埠

Amazon RDS 會封鎖 MariaDB 引擎的自訂連接埠 33060 的連線。為您的 MariaDB 引擎選擇不同的連接埠。

Performance Insights (績效詳情)

在適用於 RDS for MariaDB 10.11 版的 Performance Insights 中看不見 InnoDB 計數器，因為 MariaDB 社群不再支援這些計數器。

Amazon RDS for Microsoft SQL Server

Amazon RDS 支援多個版本的 Microsoft SQL Server。下方表格展示每個主要版本最新支援的次要版本。如需支援版本及 RDS 引擎版本的完整清單，請參閱 [Amazon RDS 上的 Microsoft SQL Server 版本](#)。

主要版本	Service Pack / GDR	累積更新	次要版本	知識庫文章	版本日期
SQL 伺服器	–	CU13	16.0.4125.3	KB5036432	2024年5月23日
SQL Server 2019	–	CU26	15.0.4365.2	KB5035123	2024年4月11日
SQL Server 2017	GDR	CU31	14.0.3465.1	KB5029376	2023年10月10日
SQL Server 2016	SP3 GDR	–	13.0.6435.1	KB5029186	2023年10月10日
SQL Server 2014	SP3 GDR	CU4	12.0.6449.1	KB5029185	2023年10月10日

如需 SQL Server 授權的相關資訊，請參閱 [Amazon RDS 上的 Microsoft SQL Server 授權](#)。如需 SQL Server 組建的相關資訊，請參閱此 Microsoft 支援文章，[了解如何尋找最新 SQL Server 組建的相關資訊](#)。

使用 Amazon RDS，您可以建立資料庫執行個體和資料庫快照、point-in-time 還原以及自動或手動備份。而且，您還能夠在 VPC 內使用執行 SQL Server 的資料庫執行個體。您也可以透過 Secure Sockets Layer (SSL) 來連線至執行 SQL Server 的資料庫執行個體，並利用透明資料加密 (TDE) 加密靜態資料。Amazon RDS 目前使用 SQL Server 資料庫鏡像 (DBM) 或 Always On 可用性群組 (AG) 作為高可用性的容錯移轉解決方案，藉此支援 SQL Server 的異地同步備份部署作業。

為提供受管服務體驗，Amazon RDS 不提供資料庫執行個體的殼層存取權，而且會限制存取某些需要進階權限的系統程序和資料表。Amazon RDS 可支援使用 Microsoft SQL Server Management Studio

等任何標準 SQL 用戶端應用程式來存取資料庫執行個體上的資料庫。Amazon RDS 不允許主機透過 Telnet、安全殼層 (SSH) 或 Windows 遠端桌面連線，直接存取資料庫執行個體。當您建立資料庫執行個體時，系統會指派主要使用者擔任該執行個體所有資料庫的 db_owner 角色，使其擁有資料庫層級的所有許可 (用於備份的許可除外)。Amazon RDS 會為您管理備份。

建立第一個資料庫執行個體之前，請先完成本指南的設定章節中的步驟。如需更多詳細資訊，請參閱 [設定 Amazon RDS](#)。

主題

- [Amazon RDS 上 Microsoft SQL Server 的一般管理任務](#)
- [Microsoft SQL Server 資料庫執行個體的限制](#)
- [Microsoft SQL Server 的資料庫執行個體類別支援](#)
- [Microsoft SQL Server 安全](#)
- [Microsoft SQL Server 資料庫執行個體的合規計劃支援](#)
- [Microsoft SQL Server 資料庫執行個體的 SSL 支援](#)
- [Amazon RDS 上的 Microsoft SQL Server 版本](#)
- [Amazon RDS 中的版本管理](#)
- [Amazon RDS 上的 Microsoft SQL Server 功能](#)
- [Microsoft SQL Server 資料庫執行個體的變更資料擷取支援](#)
- [不支援的功能和具備有限支援的功能](#)
- [使用 Microsoft SQL Server 資料庫鏡像或 Always On 可用性群組的異地同步備份部署](#)
- [使用透明資料加密來加密靜態資料](#)
- [Amazon RDS for Microsoft SQL Server 的函數和預存程序](#)
- [Microsoft SQL Server 資料庫執行個體的本機時區](#)
- [Amazon RDS 上的 Microsoft SQL Server 授權](#)
- [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)
- [使用 Active Directory 搭配 RDS for SQL Server](#)
- [將應用程式更新為使用新的 SSL/TLS 憑證來連線至 Microsoft SQL Server 資料庫執行個體](#)
- [升級 Microsoft SQL Server 資料庫引擎](#)
- [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)
- [使用 Amazon RDS 中的 Microsoft SQL Server 的僅供讀取複本](#)
- [Amazon RDS for Microsoft SQL Server 的異地同步備份部署](#)

- [Microsoft SQL Server on Amazon RDS 的其他功能](#)
- [Microsoft SQL Server 資料庫引擎的選項](#)
- [Microsoft SQL Server 的常用 DBA 任務](#)

Amazon RDS 上 Microsoft SQL Server 的一般管理任務

下列是您可在執行 Amazon RDS for SQL Server 資料庫執行個體時使用的一般管理任務，並隨附各工作相關說明文件的連結。

任務區域	相關文件
<p>執行個體類別、儲存體和 PIOPS</p> <p>如果您要針對生產用途建立資料庫執行個體，您應該了解執行個體類別、儲存體類型和佈建的 IOPS 在 Amazon RDS 中的運作方式。</p>	<p>Microsoft SQL Server 的資料庫執行個體類別支援</p> <p>Amazon RDS 儲存類型</p>
<p>異地同步備份部署</p> <p>生產資料庫執行個體應該使用多個可用區部署。「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。SQL Server 的異地同步備份部署是採用 SQL Server 的原生 DBM 或 AG 技術來實作。</p>	<p>設定及管理多可用區部署</p> <p>使用 Microsoft SQL Server 資料庫鏡像或 Always On 可用性群組的異地同步備份部署</p>
<p>Amazon Virtual Private Cloud (VPC)</p> <p>如果您的 AWS 帳戶具有預設 VPC，則會在預設 VPC 中自動建立資料庫執行個體。如果您的帳戶沒有預設的 VPC，而您想要 VPC 中有該資料庫執行個體，則必須在建立資料庫執行個體之前，先建立 VPC 和子網路群組。</p>	<p>在 VPC 中使用資料庫執行個體</p>
<p>安全群組</p> <p>根據預設，建立資料庫執行個體時會同時建立防火牆，可防止執行個體遭受不適當的存取。因此，您必須使用正確的 IP 地址和網路組態建立安全群組，才能存取資料庫執行個體。</p>	<p>使用安全群組控制存取</p>
<p>參數群組</p>	<p>使用參數群組</p>

任務區域	相關文件
<p>如果您的資料庫執行個體將要求特定資料庫參數，您應該在建立資料庫執行個體之前，先建立參數群組。</p>	
<p>選項群組</p> <p>如果您的資料庫執行個體將會要求特定資料庫選項，您應該在建立資料庫執行個體之前，先建立選項群組。</p>	Microsoft SQL Server 資料庫引擎的選項
<p>連線到您的資料庫執行個體</p> <p>建立安全群組並與資料庫執行個體建立關聯後，您可以使用任何標準 SQL 用戶端應用程式 (例如 Microsoft SQL Server Management Studio) 來連線至資料庫執行個體。</p>	連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體
<p>備份和還原</p> <p>當您建立資料庫執行個體時，您可以將它設定為取得自動備份。您也可以使用完整的備份檔案 (.bak 檔案)，手動備份與還原資料庫。</p>	備份簡介 使用原生備份和還原，匯入和匯出 SQL Server 資料庫
<p>監控</p> <p>您可以使用 CloudWatch Amazon RDS 指標、事件和增強型監控來監控 SQL Server 資料庫執行個體。</p>	在 Amazon RDS 主控台中檢視指標 檢視 Amazon RDS 事件
<p>日誌檔</p> <p>您可以存取 SQL Server 資料庫執行個體の日誌檔案。</p>	監控 Amazon RDS 日誌檔案 Microsoft SQL Server 資料庫日誌檔案

也有使用 SQL Server 資料庫執行個體的進階管理任務。如需詳細資訊，請參閱下列文件：

- [Microsoft SQL Server 的常用 DBA 任務](#)
- [使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄](#)
- [存取 tempdb 資料庫](#)

Microsoft SQL Server 資料庫執行個體的限制

在資料庫執行個體上進行 Microsoft SQL Server 的 Amazon RDS 實作時，您應該注意下列限制：

- 資料庫執行個體上可支援的資料庫數量上限需視執行個體類別類型和可用性模式而定，其中包括單一可用區模式、多可用區資料庫鏡像 (DBM) 模式，以及多可用區可用性群組 (AG) 模式。Microsoft SQL Server 系統資料庫不在此限內。

下表說明每個執行個體類別類型和可用性模式能支援的資料庫數量上限。此表格可幫助您判斷是否要轉換為不同的執行個體類別類型，或是轉換為不同的可用性模式。如果來源資料庫執行個體的資料庫數量超過目標執行個體類別類型或可用性模式能支援的數量，則無法修改資料庫執行個體。您可以在 Events (事件) 窗格中查看請求的狀態。

執行個體類別類型	單一可用區	採用 DBM 的多個可用區	採用 Always On AG 的多個可用區
db.*.micro 到 db.*.medium	30	N/A	N/A
db.*.large	30	30	30
db.*.xlarge 到 db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

* 表示不同的執行個體類別類型。

例如，假設您的資料庫執行個體有 76 個資料庫，且在單一可用區域的 db.*.16xlarge 上執行。您要修改該資料庫執行個體，以將其升級為使用多個可用區 Always On AG 模式。但是，資料庫執行個體所包含的資料庫超過目標組態可支援的數量，因此這個升級作業會失敗。如果您改將執行個體類別類型升級為 db.*.24xlarge，就能成功修改。

若升級失敗，畫面上顯示的事件和訊息會如下所示：

- 無法修改資料庫執行個體類別。該執行個體擁有 76 個資料庫，但轉換後可支援的數量只有 75 個。

- 無法將資料庫執行個體轉換為多個可用區模式。該執行個體擁有 76 個資料庫，但轉換後可支援的數量只有 75 個。

如果 point-in-time 還原或快照還原失敗，您會看到類似下列內容的事件和訊息：

- 資料庫執行個體進入不相容的還原模式。該執行個體擁有 76 個資料庫，但轉換後可支援的數量只有 75 個。
- 以下是保留供 Amazon RDS 使用的連接埠，因此您無法在建立資料庫執行個體時使用這些連接埠：1234, 1434, 3260, 3343, 3389, 47001, 和 49152-49156。
- 不允許從 169.254.0.0/16 範圍內的 IP 地址連線的用戶端。這是「自動私有 IP 定址範圍」(APIPA)，它用於本機連結定址。
- 如果資料庫執行個體的處理器數量超過軟體限制 (24 核心、4 個插槽、128GB RAM)，SQL Server Standard Edition 只會使用一部分可用的處理器。例如 db.m5.24xlarge 和 db.r5.24xlarge 執行個體類別。

如需詳細資訊，請參閱 Microsoft 文件中 [SQL Server 2019 \(15.x\) 的版本和支援功能](#) 下的擴展限制表。

- Amazon RDS for SQL Server 不支援將資料匯入 msdb 資料庫。
- 您無法在 SQL Server 異地同步備份部署的資料庫執行個體上重新命名資料庫。
- 在設定下列 RDS for SQL Server 資料庫參數時，請確定您使用這些準則：
 - `max server memory (mb) >= 256 MB`
 - `max worker threads >= (邏輯 CPU 數量 * 7)`

如需設定資料庫參數的詳細資訊，請參閱[使用參數群組](#)。

- SQL Server 資料庫執行個體的最高儲存空間大小如下：
 - 一般用途 (SSD) 儲存空間 – 16 TiB，適用於所有版本
 - 佈建 IOPS 儲存空間 – 16 TiB，適用於所有版本
 - 磁帶儲存空間 – 1 TiB，適用於所有版本

如果您需要更大的儲存容量，則可在多個資料庫執行個體間使用分片，以因應此限制。此方式需要連線至分片系統的應用程式中有資料相依路由邏輯。您可以使用現有的分片架構，或撰寫自訂程式碼來啟用分片。如果您使用現有架構，則架構無法將任何元件與資料庫執行個體安裝在相同的伺服器上。

- SQL Server 資料庫執行個體的最低儲存空間大小如下：
 - 一般用途 (SSD) 儲存體 – 20 GiB (適用於 Enterprise 版、Standard 版、Web 版和 Express 版)
 - 佈建 IOPS 儲存體 – 20 GiB (適用於 Enterprise 版、Standard 版、Web 版和 Express 版)

- 磁儲存體 – 20 GiB (適用於 Enterprise 版、Standard 版、Web 版和 Express 版)
- Amazon RDS 不支援在與 RDS 資料庫執行個體相同的伺服器上執行以下服務：
 - Data Quality Services
 - Master Data Services

若要使用這些功能，建議您在 Amazon EC2 執行個體上安裝 SQL Server，或使用現場部署 SQL Server 執行個體。在這些情況下，EC2 或 SQL Server 執行個體可做為 Amazon RDS 上 SQL Server 資料庫執行個體的 Master Data Services 伺服器。您可以根據 Microsoft 授權政策，在具有 Amazon EBS 儲存體的 Amazon EC2 執行個體上安裝 SQL Server。

- 由於 Microsoft SQL Server 的限制，還原至成功執行 DROP DATABASE 前的時間點，可能不會反映資料庫在該時間點的狀態。舉例來說，捨棄的資料庫通常最多可還原至發出 DROP DATABASE 命令前 5 分鐘的狀態。此類型的還原操作表示您無法還原在那幾分鐘期間於捨棄資料庫上所做的交易。若要解決此問題，您可以在還原操作完成後，重新發出 DROP DATABASE 命令。捨棄資料庫會移除該資料庫的交易日誌。
- 針對 SQL Server，您可以在建立資料庫執行個體之後建立資料庫。資料庫名稱遵循一般的 SQL Server 命名規則，並包含以下差異：
 - 資料庫名稱不能以 rdsadmin 開頭。
 - 資料庫名稱不能以空格或 Tab 鍵開始或結束。
 - 資料庫名稱不可包含任何會建立新一行文字的字元。
 - 資料庫名稱不可包含單引號 (')。
 - 適用於 SQL 伺服器的 RDS 目前不支援自動次要版本更新。如需詳細資訊，請參閱 [Amazon RDS 中的版本管理](#)。
- SQL Server 網頁版只允許您在建立新的 SQL Server 資料庫執行個體時使用開發/測試範本。

Microsoft SQL Server 的資料庫執行個體類別支援

資料庫執行個體的運算和記憶體容量由其資料庫執行個體類別決定。您需要的資料庫執行個體類別取決於您的處理能力和記憶體需求。如需更多詳細資訊，請參閱 [資料庫執行個體類別](#)。

為了方便您使用，提供下列 Microsoft SQL Server 支援的資料庫執行個體類別清單。如需最新的清單，請參閱 RDS 主控台：<https://console.aws.amazon.com/rds/>。

並非所有資料庫執行個體類別皆可用於所有受支援的 SQL Server 次要版本上。例如，一些較新的資料庫執行個體類別 (如 db.r6i) 無法用於較舊的次要版本上。您可以使用可[描述順序的 DB 執行個體選項 AWS CLI 命令](#)，找出哪些資料庫執行個體類別適用於您的 SQL Server 版本和版本。

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
企業版本	db.t3.xla rge -db.t3.2xlarge	db.t3.xla rge -db.t3.2xlarge	db.t3.xla rge -db.t3.2xlarge	db.t3.xla rge -db.t3.2xlarge
	db.r5.large ge -db.r5.24xlarge	db.r5.xla rge -db.r5.24xlarge	db.r3.xla rge -db.r3.8xlarge	db.r3.xla rge -db.r3.8xlarge
	db.r5b.large rge -db.r5b.24xlarge	db.r5b.xlarge arge -db.r5b.24xlarge	db.r4.xla rge -db.r4.16xlarge	db.r4.xla rge -db.r4.8xlarge
	db.r5d.large rge -db.r5d.24xlarge	db.r5d.xlarge arge -db.r5d.24xlarge	db.r5.xla rge -db.r5.24xlarge	db.r5.xla rge -db.r5.24xlarge
	db.r6i.large rge -db.r6i.32xlarge	db.r6i.xlarge arge -db.r6i.32xlarge	db.r5b.xlarge arge -db.r5b.24xlarge	db.r5b.xlarge arge -db.r5b.24xlarge
	db.m5.large ge -db.m5.24xlarge	db.m5.xla rge -db.m5.24xlarge	db.r5d.xlarge arge -db.r5d.24xlarge	db.r5d.xlarge arge -db.r5d.24xlarge
	db.m5d.large rge -db.m5d.24xlarge	db.m5d.xlarge arge -db.m5d.24xlarge	db.r6i.xlarge arge -db.r6i.32xlarge	db.r6i.xlarge arge -db.r6i.32xlarge
	db.m6i.large rge -db.m6i.32xlarge	db.m6i.xlarge arge -db.m6i.32xlarge	db.m4.xla rge -db.m4.16xlarge	db.m4.xla rge -db.m4.10xlarge
	db.x2iedn.xlarge -db.x2iecd.32xlarge	db.x1.16xlarge arge -db.x1.32xlarge	db.m5.xla rge -db.m5.24xlarge	db.m5.xla rge -db.m5.24xlarge

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
	db.z1d.large -db.z1d.12xlarge	db.x1e.xlarge -db.x1e.32xlarge	db.m5d.xlarge -db.m5d.24xlarge	db.m5d.xlarge -db.m5d.24xlarge
		db.x2iedn.xlarge -db.x2iedn.32xlarge	db.m6i.xlarge -db.m6i.32xlarge	db.m6i.xlarge -db.m6i.32xlarge
		db.z1d.xlarge -db.z1d.12xlarge	db.x1.16xlarge -db.x1.32xlarge	db.x1.16xlarge -db.x1.32xlarge
			db.x1e.xlarge -db.x1e.32xlarge	db.x1e.xlarge -db.x1e.32xlarge
			db.x2iedn.xlarge -db.x2iedn.32xlarge	db.x2iedn.xlarge -db.x2iedn.32xlarge
			db.z1d.xlarge -db.z1d.12xlarge	

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
標準版本	db.t3.xla rge –db.t3.2x1 arge	db.t3.xla rge –db.t3.2x1 arge	db.t3.xla rge –db.t3.2x1 arge	db.t3.xla rge –db.t3.2x1 arge
	db.r5.lar ge –db.r5.24x large	db.r5.lar ge –db.r5.24x large	db.r4.lar ge –db.r4.16x large	db.r3.lar ge –db.r3.8x1 arge
	db.r5b.la rge –db.r5b.8x large	db.r5b.la rge –db.r5b.24 xlarge	db.r5.lar ge –db.r5.24x large	db.r4.lar ge –db.r4.8x1 arge
	db.r5d.la rge –db.r5d.24 xlarge	db.r5d.la rge –db.r5d.24 xlarge	db.r5b.la rge –db.r5b.24 xlarge	db.r5.lar ge –db.r5.24x large
	db.r6i.la rge –db.r6i.8x large	db.r6i.la rge –db.r6i.8x large	db.r5d.la rge –db.r5d.24 xlarge	db.r5b.la rge –db.r5b.24 xlarge
	db.m5.lar ge –db.m5.24x large	db.m5.lar ge –db.m5.24x large	db.r6i.la rge –db.r6i.8x large	db.r5d.la rge –db.r5d.24 xlarge
	db.m5d.la rge –db.m5d.24 xlarge	db.m5d.la rge –db.m5d.24 xlarge	db.m4.lar ge –db.m4.16x large	db.r6i.la rge –db.r6i.8x large
	db.m6i.la rge –db.m6i.8x large	db.m6i.la rge –db.m6i.8x large	db.m5.lar ge –db.m5.24x large	db.m3.med ium –db.m3.2x1 arge
	db.x2iedn .xlarge –db.x2iec .8xlarge	db.x1.16x large –db.x1.32x large	db.m5d.la rge –db.m5d.24 xlarge	db.m4.lar ge –db.m4.10x large

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
	db.z1d.large -db.z1d.12xlarge	db.x1e.xlarge -db.x1e.32xlarge	db.m6i.large -db.m6i.8xlarge	db.m5.large -db.m5.24xlarge
		db.x2iedn.xlarge -db.x2iedn.32xlarge	db.x1.16xlarge -db.x1.32xlarge	db.m5d.large -db.m5d.24xlarge
	db.z1d.large -db.z1d.12xlarge	db.x1e.xlarge -db.x1e.32xlarge	db.x1e.xlarge -db.x1e.32xlarge	db.m6i.large -db.m6i.8xlarge
			db.x2iedn.xlarge -db.x2iedn.32xlarge	db.x1.16xlarge -db.x1.32xlarge
		db.z1d.large -db.z1d.12xlarge	db.z1d.large -db.z1d.12xlarge	db.x1e.xlarge -db.x1e.32xlarge
				db.x2iedn.xlarge -db.x2iedn.32xlarge

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
Web 版本	db.t3.sma 11 -db.t3.xlarge	db.t3.sma 11 -db.t3.2xlarge	db.t2.sma 11 -db.t2.medium	db.t2.sma 11 -db.t2.medium
	db.r5.large -db.r5.4xlarge	db.r5.large -db.r5.4xlarge	db.t3.sma 11 -db.t3.2xlarge	db.t3.sma 11 -db.t3.2xlarge
	db.r5b.large -db.r5b.4xlarge	db.r5b.large -db.r5b.4xlarge	db.r4.large -db.r4.2xlarge	db.r3.large -db.r3.2xlarge
	db.r5d.large -db.r5d.4xlarge	db.r5d.large -db.r5d.4xlarge	db.r5.large -db.r5.4xlarge	db.r4.large -db.r4.2xlarge
	db.r6i.large -db.r6i.4xlarge	db.r6i.large -db.r6i.4xlarge	db.r5b.large -db.r5b.4xlarge	db.r5.large -db.r5.4xlarge
	db.m5.large -db.m5.4xlarge	db.m5.large -db.m5.4xlarge	db.r5d.large -db.r5d.4xlarge	db.r5b.large -db.r5b.4xlarge
	db.m5d.large -db.m5d.4xlarge	db.m5d.large -db.m5d.4xlarge	db.r6i.large -db.r6i.4xlarge	db.r5d.large -db.r5d.4xlarge
	db.m6i.large -db.m6i.4xlarge	db.m6i.large -db.m6i.4xlarge	db.m4.large -db.m4.4xlarge	db.r6i.large -db.r6i.4xlarge
	db.z1d.large -db.z1d.13xlarge	db.z1d.large -db.z1d.3xlarge	db.m5.large -db.m5.4xlarge	db.m3.medium -db.m3.2xlarge

SQL Server 版本	支援範圍	2019 支援範圍	2017 和 2016 支援範圍	2014 支援範圍
			db.m5d.large -db.m5d.4xlarge	db.m4.large -db.m4.4xlarge
			db.m6i.large -db.m6i.4xlarge	db.m5.large -db.m5.4xlarge
			db.z1d.large -db.z1d.3xlarge	db.m5d.large -db.m5d.4xlarge
				db.m6i.large -db.m6i.4xlarge
Express 版本	db.t3.micro -db.t3.xlarge	db.t3.micro -db.t3.xlarge	db.t2.micro -db.t2.medium db.t3.micro -db.t3.xlarge	db.t2.micro -db.t2.medium db.t3.micro -db.t3.xlarge

Microsoft SQL Server 安全

Microsoft SQL Server 資料庫引擎使用角色型安全。建立資料庫執行個體時您指定的主要使用者名稱是以 SQL Server 身分驗證登入，即 processadmin、public 和 setupadmin 固定伺服器角色的成員。

建立資料庫的任何使用者會指派給該資料庫的 db_owner 角色，並具有所有資料庫層級許可 (用於備份的許可除外)。Amazon RDS 會為您管理備份。

以下伺服器層級角色無法在 Amazon RDS for SQL Server 中使用：

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

以下伺服器層級許可無法在 RDS for SQL Server 資料庫執行個體上使用：

- ALTER ANY DATABASE
- ALTER ANY EVENT NOTIFICATION
- ALTER RESOURCES
- ALTER RESOURCES (您可以使用資料庫參數群組 API 操作來修改參數，如需詳細資訊，請參閱[使用參數群組](#))
- AUTHENTICATE SERVER
- CONTROL_SERVER
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT
- 建立伺服器角色
- CREATE TRACE EVENT NOTIFICATION
- DROP ANY DATABASE
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN (您可以改用 RDS 重新開機選項)
- UNSAFE ASSEMBLY
- 變更任何可用性群組
- 建立可用性群組

Microsoft SQL Server 資料庫執行個體的合規計劃支援

AWS 範圍內的服務已由第三方稽核人員完整評估，並取得認證、合規證明或營運授權 (ATO)。如需詳細資訊，請參閱[合規計劃的AWS 服務範圍](#)。

Microsoft SQL Server 資料庫執行個體的 HIPAA 支援

您可以使用 Amazon RDS for Microsoft SQL Server 資料庫來建置符合 HIPAA 規範的應用程式。依據與 AWS 簽署的商業夥伴協議 (BAA)，您可以存放醫療保健相關的資訊，包括受保護醫療資訊 (PHI)。如需詳細資訊，請參閱 [HIPAA 合規](#)。

Amazon RDS for SQL Server 支援下列版本的 HIPAA：

- SQL 伺服器 2022 年企業版、標準版和網頁版
- SQL Server 2019 Enterprise、Standard 和 Web Edition
- SQL Server 2017 Enterprise、Standard 和 Web Edition
- SQL Server 2016 Enterprise、Standard 和 Web Edition
- SQL Server 2014 Enterprise、Standard 和 Web Edition

若要在資料庫執行個體上啟用 HIPAA 支援，請設定下列三個元件。

元件	詳細資訊
稽核	若要設定，請將 <code>rds.sqlserver_audit</code> 參數的值設為 <code>fedramp_hipaa</code> 。如果資料庫執行個體尚未使用資料庫參數群組，則您必須建立自訂參數群組，並將它附加至資料庫執行個體，然後才能修改 <code>rds.sqlserver_audit</code> 參數。如需更多詳細資訊，請參閱 使用參數群組 。
傳輸加密	若要設定傳輸加密，請強制資料庫執行個體的所有連線皆使用 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 強制連線至資料庫執行個體以使用 SSL 。
靜態加密	若要設定靜態加密，您有兩個選項： <ol style="list-style-type: none">1. 如果您執行 SQL 伺服器 2014—2022 企業版或 2022 年標準版，您可以使用透明資料加密 (TDE) 來實現靜態加密。如需詳細資訊，請參閱 支援 SQL Server 的透明資料加密。2.

元件	詳細資訊
	您可以使用 AWS Key Management Service (AWS KMS) 加密金鑰來設定靜態加密。如需詳細資訊，請參閱 加密 Amazon RDS 資源 。

Microsoft SQL Server 資料庫執行個體的 SSL 支援

您也可以使用 SSL，來加密應用程式與執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體之間的連線。您也可以強制資料庫執行個體的所有連線都使用 SSL。如果您強制所有連線都使用 SSL，用戶端自然就會這樣做，用戶端不必有任何動作即可使用 SSL。

所有 AWS 區域和所有受支援的 SQL Server 版本都支援 SSL。如需詳細資訊，請參閱 [對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)。

Amazon RDS 上的 Microsoft SQL Server 版本

建立新的資料庫執行個體時，您可以指定使用目前支援的任一 Microsoft SQL Server 版本。此外，您也可以指定 Microsoft SQL Server 主要版本 (如 Microsoft SQL Server 14.00 版)，以及指定主要版本支援的任一次要版本。若未指定版本，Amazon RDS 會預設使用支援的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 會預設使用您指定主要版本的最新次要版本。

下表顯示所有版本和所有 AWS 區域的支援版本，除非另有說明。您也可以使用指 [describe-db-engine-versions](#) AWS CLI 令查看支援的版本清單，以及新建立的資料庫執行個體的預設值。

RDS 中支援的 SQL Server 版本

主要版本	次要版本	RDS API EngineVersion 與 CLI engine-version
SQL 伺服器	16.00.4125.3 (立方十三)	16.00.4125.3.v1
	16.00.4120.1 (克十二公克)	16.00.4120.1.v1
	16.00.4115.5 (立法十二)	16.00.4115.5.v1
	16.00.4105.2 (立方十一)	16.00.4105.2.v1
	16.00.4095.4 (立方十)	16.00.4095.4.v1

主要版本	次要版本	RDS API EngineVersion 與 CLI engine-version
	16.00.4085.2 (九點)	16.00.4085.2.v1
SQL Server 2019	15.00.4365.2 (立方二十六)	15.00.4365.2
	15.00.4355.3 (立方二十五)	15.00.4355.3.v1
	15.00.4345.5 (立即二十四)	15.00.4345.5.v1
	15.00.4335.1 (CU23)	15.00.4335.1.v1
	15.00.4322.2 (CU22)	15.00.4322.2.v1
	15.00.4316.3 (CU21)	15.00.4316.3.v1
	15.00.4312.2 (CU20)	15.00.4312.2.v1
	15.00.4236.7 (CU16)	15.00.4236.7.v1
	15.00.4198.2 (CU15)	15.00.4198.2.v1
	15.00.4153.1 (CU12)	15.00.4153.1.v1
	15.00.4073.23 (CU8)	15.00.4073.23.v1
15.00.4043.16 (CU5)	15.00.4043.16.v1	

主要版本	次要版本	RDS API EngineVersion 與 CLI engine-version
SQL Server 2017	14.00.3465.1 (CU31)	14.00.3465.1.v1
	14.00.3460.9 (CU31)	14.00.3460.9.v1
	14.00.3451.2 (CU30)	14.00.3451.2.v1
	14.00.3421.10 (CU27)	14.00.3421.10.v1
	14.00.3401.7 (CU25)	14.00.3401.7.v1
	14.00.3381.3 (CU23)	14.00.3381.3.v1
	14.00.3356.20 (CU22)	14.00.3356.20.v1
	14.00.3294.2 (CU20)	14.00.3294.2.v1
	14.00.3281.6 (CU19)	14.00.3281.6.v1
SQL Server 2016	13.00.6435.1 (GDR)	13.00.6435.1.v1
	13.00.6430.49 (GDR)	13.00.6430.49.v1
	13.00.6419.1 (SP3 + Hotfix)	13.00.6419.1.v1
	13.00.6300.2 (SP3)	13.00.6300.2.v1
SQL Server 2014	12.00.6449.1 (SP3 CU4 GDR)	12.00.6449.1.v1
	12.00.6444.4 (SP3 CU4 GDR)	12.00.6444.4.v1
	12.00.6439.10 (SP3 CU4 SU)	12.00.6439.10.v1
	12.00.6433.1 (SP3 CU4 SU)	12.00.6433.1.v1
	12.00.6329.1 (SP3 CU4)	12.00.6329.1.v1
	12.00.6293.0 (SP3 CU3)	12.00.6293.0.v1

Amazon RDS 中的版本管理

Amazon RDS 包含靈活的版本管理功能，可讓您控制資料庫執行個體修補或升級的時機與方式。透過這項功能，您可以對資料庫引擎執行以下操作：

- 維護與資料庫引擎修補程式版本的相容性。
- 測試新的修補程式版本，確定這些版本可搭配您的應用程式使用後，再將其部署至生產環境。
- 按照您的服務水準協議和時間需求，規劃並執行版本升級作業。

Amazon RDS 中的 Microsoft SQL Server 引擎修補

Amazon RDS 會定期將官方 Microsoft SQL Server 資料庫修補程式彙整到 Amazon RDS 特定的資料庫執行個體引擎版本中。如需各引擎版本中 Microsoft SQL Server 修補程式的詳細資訊，請參閱 [Amazon RDS 上的版本和功能支援](#)。

目前，您需要在資料庫執行個體上手動升級所有引擎。如需更多詳細資訊，請參閱 [升級 Microsoft SQL Server 資料庫引擎](#)。

Amazon RDS 上的 Microsoft SQL Server 主要引擎版本棄用排程

下表顯示計劃的 Microsoft SQL Server 主要引擎版本棄用排程。

日期	資訊
2024 年 7 月 9 日	Microsoft 將停止更新 SQL Server 2014 的重大修補程式。如需詳細資訊，請參閱 Microsoft SQL Server 2014 。
2024 年 6 月 1 日	Amazon RDS 計劃在 RDS for SQL Server 上終止對 Microsoft SQL Server 2014 的支援執行個體會排定遷移至 SQL Server 2016 (最新可用次要版本)。如需詳細資訊，請參閱 for SQL Server 終止對 SQL Server 2014 主要版本的支援 。 為避免系統自動從 Microsoft SQL Server 2014 升級，您可在方便時自行升級。如需詳 料庫執行個體引擎版本 。
2022 年 7 月 12 日	Microsoft 將停止 SQL Server 2012 的重大修補程式更新。如需詳細資訊，請參閱 Microsoft SQL Server 2012 。

日期	資訊
2022 年 6 月 1 日	<p>Amazon RDS 計劃在 RDS for SQL Server 上終止對 Microsoft SQL Server 2012 的支援。執行個體會排定遷移至 SQL Server 2014 (最新可用次要版本)。如需詳細資訊，請參閱 for SQL Server 終止對 SQL Server 2012 主要版本的支援。</p> <p>為避免系統自動從 Microsoft SQL Server 2012 升級，您可在方便時自行升級。如需詳細資訊，請參閱 資料庫執行個體引擎版本。</p>
2021 年 9 月 1 日	<p>Amazon RDS 開始停用使用 Microsoft SQL Server 2012 來建立新的 RDS for SQL Server。如需詳細資訊，請參閱 公告：Amazon RDS for SQL Server 終止對 SQL Server 2012 的支援。</p>
2019 年 7 月 12 日	<p>Amazon RDS 團隊已於 2019 年 6 月移除對 Microsoft SQL Server 2008 R2 的支援。Microsoft SQL Server 2008 R2 執行個體會遷移至 SQL Server 2012 (最新的可用次要版本)。</p> <p>為避免系統自動從 Microsoft SQL Server 2008 R2 升級，您可以在方便的時候自行升級。如需詳細資訊，請參閱 升級資料庫執行個體引擎版本。</p>
2019 年 4 月 25 日	<p>2019 年 4 月底前，您無法再使用 Microsoft SQL Server 2008R2 建立新的 Amazon RDS 資料庫執行個體。</p>

Amazon RDS 上的 Microsoft SQL Server 功能

Amazon RDS 上支援的 SQL Server 版本包含下列功能。一般而言，除非 Microsoft 文件中另有說明，否則版本也會包含先前版本的功能。

主題

- [Microsoft SQL 伺服器功能](#)
- [Microsoft SQL Server 2019 功能](#)
- [Microsoft SQL Server 2017 功能](#)
- [Microsoft SQL Server 2016 功能](#)
- [Microsoft SQL Server 2014 功能](#)
- [Microsoft SQL Server 2012 已終止對 Amazon RDS 的支援](#)
- [Microsoft SQL Server 2008 R2 已終止對 Amazon 的支援](#)

Microsoft SQL 伺服器功能

SQL 伺服器 2022 包含許多新功能，例如：

- 參數敏感計劃優化 — 允許單個參數化語句的多個緩存計劃，從而可能減少參數嗅探的問題。
- SQL Server 分類帳 — 提供以密碼編譯方式證明您的資料未經授權未經過變更的功能。
- 交易記錄檔成長事件的即時檔案初始化 — 可加快執行高達 64MB 的記錄成長事件，包括啟用 TDE 的資料庫。
- 系統頁面門鎖並行增強功能 — 減少頁面門鎖爭用，同時配置和取消配置資料頁面和範圍，為繁重的工作負載提供顯著的效能增強功能。tempdb

如需 SQL 伺服器 2022 功能的完整清單，請參閱 Microsoft 說明文件中 [SQL 伺服器 2022 中的新增功能 \(16.x\)](#)。

如需未支援的功能清單，請參閱 [不支援的功能和具備有限支援的功能](#)。

Microsoft SQL Server 2019 功能

SQL Server 2019 包括許多新功能，如下：

- 加速資料庫復原 (ADR) – 可減少重新啟動或長時間執行的交易復原之後的損毀復原時間。
- 智慧型查詢處理 (IQP)：
 - 資料列模式記憶體授權回饋 – 會自動更正過度授權，否則會導致記憶體浪費並減少並行性。
 - 資料列儲存區上的批次模式 – 啟用分析工作負載的批次模式執行，而不需要資料欄儲存區索引。
 - 資料表變數延遲編譯 – 改善參考資料表變數之查詢的計劃品質和整體效能。
- 智慧型效能：
 - OPTIMIZE_FOR_SEQUENTIAL_KEY 索引選項 – 改善索引中高並行插入的輸送量。
 - 改良的間接檢查點擴展性 – 有助於處理繁重 DML 工作負載的資料庫
 - 並行頁面可用空間 (PFS) 更新 – 可讓處理為共用鎖定，而非獨佔鎖定。
- 監控改善：
 - WAIT_ON_SYNC_STATISTICS_REFRESH 等待類型 – 顯示用於同步統計資料重新整理作業的累積執行處理層次時間。
 - 資料庫範圍的組態 – 包括 LIGHTWEIGHT_QUERY_PROFILING 和 LAST_QUERY_PLAN_STATS。
 - 動態管理功能 (DMF) – 包含 sys.dm_exec_query_plan_stats 和 sys.dm_db_page_info。

- 詳細截斷警告 – 資料截斷錯誤訊息預設為包含資料表和資料行名稱以及截斷的值。
- 可恢復的線上索引建立 – 在 SQL Server 2017 中，只支援可恢復的線上索引重建。

如需 SQL Server 2019 功能的完整清單，請參閱 Microsoft 說明文件中的 [SQL Server 2019 \(15.x\) 的新功能](#)。

如需未支援的功能清單，請參閱[不支援的功能和具備有限支援的功能](#)。

Microsoft SQL Server 2017 功能

SQL Server 2017 包括許多新功能，如下：

- 調適型查詢處理
- 自動計畫校正 (自動調校功能)
- GraphDB
- 可繼續重建索引

如需 SQL Server 2017 功能的完整清單，請參閱 Microsoft 說明文件中的 [SQL Server 2017 的新功能](#)。

如需未支援的功能清單，請參閱[不支援的功能和具備有限支援的功能](#)。

Microsoft SQL Server 2016 功能

Amazon RDS 支援下列 SQL Server 2016 功能：

- 一律加密
- JSON 支援
- 營運分析
- 查詢存放區
- 時態表

如需 SQL Server 2016 功能的完整清單，請參閱 Microsoft 說明文件中的 [SQL Server 2016 的新功能](#)。

Microsoft SQL Server 2014 功能

除了支援的 SQL Server 2012 RDS 功能外，Amazon RDS 還支援 SQL Server 2014 中提供的新查詢最佳化器，以及延遲的耐用性功能。

如需未支援的功能清單，請參閱[不支援的功能和具備有限支援的功能](#)。

SQL Server 2014 支援來自 SQL Server 2012 的所有參數，並使用相同的預設值。SQL Server 2014 包括一個新參數，即 backup checksum default。如需詳細資訊，請參閱 Microsoft 說明文件中的[設定備份總和檢查碼預設值 \(伺服器組態選項\)](#)。

Microsoft SQL Server 2012 已終止對 Amazon RDS 的支援

SQL Server 2012 已終止對 Amazon RDS 的支援。

RDS 正將仍在使用 SQL Server 2012 的所有現有資料庫執行個體升級至 SQL Server 2014 的最新次要版本。如需詳細資訊，請參閱[Amazon RDS 中的版本管理](#)。

Microsoft SQL Server 2008 R2 已終止對 Amazon 的支援

SQL Server 2008 R2 已達到對 Amazon RDS 終止支援。

RDS 正將仍在使用 SQL Server 2008 R2 的所有現有資料庫執行個體升級至 SQL Server 2012 的最新次要版本。如需詳細資訊，請參閱[Amazon RDS 中的版本管理](#)。

Microsoft SQL Server 資料庫執行個體的變更資料擷取支援

Amazon RDS 支援執行 Microsoft SQL Server 之資料庫執行個體的變更資料擷取支援 (CDC)。CDC 會擷取對資料表中資料所做的變更，並存放每個變更的相關中繼資料，以便您稍後存取。如需詳細資訊，請參閱 Microsoft 文件中的[變更資料擷取](#)。

Amazon RDS 支援下列 SQL Server 版本的 CDC：

- Microsoft SQL Server Enterprise Edition (所有版本)
- Microsoft SQL Server Standard Edition：
 - 2022
 - 2019
 - 2017
 - 2016 版 13.00.4422.0 SP1 CU2 和更新版本

若要搭配使用 CDC 與 Amazon RDS 資料庫執行個體，請先使用 RDS 提供的預存程序，來啟用或停用資料庫層級的 CDC。在那之後，對該資料庫具有 `db_owner` 角色的任何使用者都可以使用原生 Microsoft 預先程序，來控制該資料庫上的 CDC。如需詳細資訊，請參閱 [使用變更資料擷取](#)。

您可以使用 CDC，並啟 AWS Database Migration Service 用 SQL Server 資料庫執行個體的持續複寫。

不支援的功能和具備有限支援的功能

Amazon RDS 上不支援下列 Microsoft SQL Server 功能：

- 備份至 Microsoft Azure Blob Storage
- 緩衝集區延伸
- 自訂密碼政策
- Data Quality Services
- 資料庫日誌傳送
- 資料庫快照 (Amazon RDS 僅支援資料庫執行個體快照)
- 擴充的預存程序，包括 `xp_cmdshell`
- FILESTREAM 支援
- 檔案資料表
- Machine Learning 和 R 服務 (需要作業系統存取權才能安裝)
- 維護計劃
- 效能資料收集器
- 政策型管理
- PolyBase
- 複寫
- 資源管理員
- 伺服器層級觸發程序
- Service Broker 端點
- 延展資料庫
- 值得信賴的資料庫屬性 (需要系統管理員角色)
- T-SQL 端點 (所有使用 `CREATE ENDPOINT` 的操作皆無法使用)
- WCF 資料服務

Amazon RDS 上對於下列 Microsoft SQL Server 功能具有有限的支援：

- 分散式查詢/連結的伺服器。如需詳細資訊，請參閱 [Implementing Linked Servers with Amazon RDS for Microsoft SQL Server](#)。
- 通用語言執行階段 (CLR)。在 RDS for SQL Server 2016 和更舊版本上，僅在 SAFE 模式並使用組件位元，才支援 CLR。在 RDS for SQL Server 2017 和新高版本上不支援 CLR。如需詳細資訊，請參閱 Microsoft 文件中的 [通用語言執行階段整合](#)。

使用 SQL 伺服器 2022 的 Amazon RDS 不支援下列功能：

- 暫停快照集的資料庫
- 外部資料來源
- Backup 及還原至 S3 相容物件儲存
- 物件存放區整合
- TLS 1.3 和微秒驅動系統 8.0
- 使用 QAT 進行 Backup 壓縮卸載
- SQL 伺服器分析服務
- 含異地同步備份部署的資料庫鏡像 SQL Server 永遠開啟是異地同步備份部署的唯一支援方法。

使用 Microsoft SQL Server 資料庫鏡像或 Always On 可用性群組的異地同步備份部署

Amazon RDS 會透過使用 SQL Server Database Mirroring (DBM) 或 Always On 可用性群組 (AG)，支援在執行 Microsoft SQL Server 的資料庫執行個體上進行異地同步備份部署。「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。如果發生規劃的資料庫維護或意外服務中斷，Amazon RDS 會自動容錯移轉到 up-to-date 次要複本，因此資料庫操作可以快速恢復，無需手動介入。主要與次要執行個體會使用相同的端點，其實體網路會處理轉換至被動次要複本的作業，做為容錯移轉程序的一部分。當容錯移轉發生時，您不必重新設定應用程式。

Amazon RDS 會管理容錯移轉，方法為主動監控多個可用區部署，並在主要執行個體發生問題時啟動容錯移轉。只有當待命和主要執行個體完全同步時，容錯移轉才會發生。Amazon RDS 會主動維護異地同步備份部署，方法為自動修復狀況不良的資料庫執行個體，並重建同步複寫。您什麼都不必管理。Amazon RDS 會為您處理主要執行個體、見證，以及待命執行個體。當您設定 SQL Server 多個可用區時，RDS 會為執行個體上的所有資料庫設定被動次要執行個體。

如需更多詳細資訊，請參閱 [Amazon RDS for Microsoft SQL Server 的異地同步備份部署](#)。

使用透明資料加密來加密靜態資料

Amazon RDS 支援 Microsoft SQL Server 透明資料加密 (TDE)，其會以透明方式加密儲存的資料。Amazon RDS 使用選項群組來啟用並設定這些功能。如需 TDE 選項的詳細資訊，請參閱 [支援 SQL Server 的透明資料加密](#)。

Amazon RDS for Microsoft SQL Server 的函數和預存程序

在下文中，您可以找到可協助自動化 SQL Server 任務的 Amazon RDS 函數和預存程序清單。

任務類型	程序或函數	其使用位置
管理任務	rds_drop_database	捨棄 Microsoft SQL Server 資料庫
	rds_failover_time	判斷上次容錯移轉時間
	rds_modify_db_name	重新命名多個可用區部署中的 Microsoft SQL Server 資料庫
	rds_read_error_log	檢視錯誤和代理程式日誌
	rds_set_configuration	此操作用於設定各種資料庫執行個體組態： <ul style="list-style-type: none"> • 變更多個可用區執行個體的資料擷取 • 設定追蹤和傾印檔案的保留期間 • 壓縮備份檔案
rds_set_database_online	將 Microsoft SQL Server 資料庫從「離線」轉移至「上線」	

任務類型	程序或函數	其使用位置
	rds_set_system_database_sync_objects	開啟 SQL Server Agent 任務複寫
	rds_fn_get_system_database_sync_objects	
	rds_fn_server_object_last_sync_time	
	rds_show_configuration	若要查看使用 rds_set_configuration 設定的值，請參閱以下主題： <ul style="list-style-type: none"> 變更多個可用區執行個體的資料擷取 設定追蹤和傾印檔案的保留期間
	rds_shrink_tempdbfile	縮減 tempdb 資料庫
變更資料擷取 (CDC)	rds_cdc_disable_db	停用 CDC
	rds_cdc_enable_db	啟用 CDC

任務類型	程序或函數	其使用位置
資料庫郵件	rds_fn_sy smail_all items	檢視郵件、記錄檔和附件
	rds_fn_sy smail_eve nt_log	檢視郵件、記錄檔和附件
	rds_fn_sy smail_mai lattachme nts	檢視郵件、記錄檔和附件
	rds_sysma il_contro l	此操作用於啟動和停止郵件佇列： <ul style="list-style-type: none"> • 啟動郵件佇列 • 停用郵件佇列
	rds_sysma il_delete _mailitem s_sp	刪除訊息
原生備份與還原	rds_backu p_databas e	備份資料庫
	rds_cance l_task	取消任務
	rds_finis h_restore	完成資料庫還原

任務類型	程序或函數	其使用位置
	rds_restore_database	還原資料庫
	rds_restore_log	還原日誌
Amazon S3 檔案傳輸	rds_delete_from_filesystem	刪除 RDS 資料庫執行個體上的檔案
	rds_download_from_s3	將 Amazon S3 儲存貯體中的檔案下載至 SQL Server 資料庫執行個體
	rds_gather_file_details	列出 RDS 資料庫執行個體上的檔案
	rds_upload_to_s3	從 SQL Server 資料庫執行個體將檔案上傳至 Amazon S3 儲存貯體
Microsoft Distributed Transaction Coordinator (MSDTC)	rds_msdtc_transaction_tracing	使用交易追蹤
SQL Server 稽核	rds_fn_get_audit_file	檢視稽核日誌

任務類型	程序或函數	其使用位置
透明資料加密	<code>rds_backup_tde_certificate</code> <code>rds_drop_tde_certificate</code> <code>rds_restore_tde_certificate</code> <code>rds_fn_list_user_tde_certificates</code>	支援 SQL Server 的透明資料加密

任務類型	程序或函數	其使用位置
Microsoft Business Intelligence (MSBI)	rds_msbi_task	<p>此操作搭配 SQL Server Analysis Services (SSAS) 一起使用：</p> <ul style="list-style-type: none"> • 在 Amazon RDS 上部署 SSAS 專案 • 新增網域使用者做為資料庫管理員 • 備份 SSAS 資料庫 • 還原 SSAS 資料庫 <p>此操作也搭配 SQL Server Integration Services (SSIS) 一起使用：</p> <ul style="list-style-type: none"> • SSISDB 上的管理許可 • 部署 SSIS 專案 <p>此操作也搭配 SQL Server Reporting Services (SSRS) 一起使用：</p> <ul style="list-style-type: none"> • 授予網域使用者的存取權 • 撤銷系統層級許可
	rds_fn_task_status	<p>此操作顯示 MSBI 任務的狀態：</p> <ul style="list-style-type: none"> • SSAS：監視部署任務的狀態 • SSIS：監視部署任務的狀態 • SSRS：監控任務的狀態
	rds_drop_ssis_database	卸除 SSISDB 資料庫
SSIS	rds_sqlagent_proxy	建立 SSIS 代理

任務類型	程序或函數	其使用位置
SSRS	rds_drop_ ssrs_data bases	刪除 SSRS 資料庫

Microsoft SQL Server 資料庫執行個體的本機時區

執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體的時區是根據預設來設定的。目前預設值是世界協調時間 (UTC)。您也可以將資料庫執行個體的時區改設為本機時區，以符合應用程式的時區。

您可以在第一次建立資料庫執行個體時設定時區。[您可以使用 Amazon RDS API 建立資料庫執行個體動作或建立 DB 執行個體命令來建立資料庫執行個體。AWS Management ConsoleAWS CLI](#)

如果您的資料庫執行個體是異地同步備份部署 (使用 SQL Server DBM 或 AG) 的一部分，則在容錯移轉時，時區仍會維持在您設定的本機時區。如需詳細資訊，請參閱 [使用 Microsoft SQL Server 資料庫鏡像或 Always On 可用性群組的異地同步備份部署](#)。

當您要求 point-in-time 還原時，您可以指定還原的時間。時間以您當地的時區顯示。如需更多詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

下列是在資料庫執行個體上設定本機時區的限制：

- 您無法修改現有 SQL Server 資料庫執行個體的時區。
- 您無法將快照從某個時區的資料庫執行個體還原至另一個時區的資料庫執行個體。
- 強烈建議您不要將備份檔案從某個時區還原至不同時區。如果您將備份檔案從某個時區還原至不同時區，則必須稽核您的查詢及應用程式，是否受到時區變更的影響。如需更多詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

支援的時區

您可以將本機時區設為下表中列出的其中一個值。

SQL 伺服器上 Amazon RDS 支援的時區

時區	標準時間偏移	描述	備註
阿富汗標準時間	(UTC+04:30)	喀布爾	此時區不會遵守日光節約時間。
阿拉斯加標準時間	(UTC-09:00)	阿拉斯加州	
阿留申標準時間	(UTC-10:00)	阿留申群島	
阿爾泰標準時間	(UTC+07:00)	巴爾瑙爾，戈爾諾-阿爾泰斯克	
沙烏地阿拉伯標準時間	(UTC+03:00)	科威特，利雅德	此時區不會遵守日光節約時間。
阿拉伯標準時間	(UTC+04:00)	阿布達比、馬斯喀特	
阿拉伯標準時間	(UTC+03:00)	巴格達	此時區不會遵守日光節約時間。
阿根廷標準時間	(UTC-03:00)	布宜諾斯艾利斯市	此時區不會遵守日光節約時間。
阿斯特拉罕標準時間	(UTC+04:00)	阿斯特拉罕，烏里揚諾夫斯克	
大西洋標準時間	(UTC-04:00)	大西洋時間 (加拿大)	
AUS 中央標準時間	(UTC+09:30)	達爾文	此時區不會遵守日光節約時間。
澳洲中西部標準時間	(UTC+08:45)	尤克拉	
AUS 東部標準時間	(UTC+10:00)	坎培拉、墨爾本、雪梨	
亞塞拜然標準時間	(UTC+04:00)	巴庫	
亞速爾標準時間	(UTC-01:00)	亞速爾	

時區	標準時間偏移	描述	備註
巴伊亞標準時間	(UTC-03:00)	薩爾瓦多	
孟加拉標準時間	(UTC+06:00)	達卡	此時區不會遵守日光節約時間。
白俄羅斯標準時間	(UTC+03:00)	明斯克	此時區不會遵守日光節約時間。
布干維爾標準時間	(UTC+11:00)	布干維爾島	
加拿大中央標準時間	(UTC-06:00)	薩斯喀徹溫	此時區不會遵守日光節約時間。
維德角標準時間	(UTC-01:00)	維德角群島	此時區不會遵守日光節約時間。
高加索標準時間	(UTC+04:00)	葉里溫	
中央 澳洲標準時間	(UTC+09:30)	阿德萊德	
中美洲標準時間	(UTC-06:00)	中央美洲	此時區不會遵守日光節約時間。
中亞標準時間	(UTC+06:00)	阿斯塔納	此時區不會遵守日光節約時間。
中央巴西標準時間	(UTC-04:00)	古雅巴	
中歐標準時間	(UTC+01:00)	貝爾格勒、布拉提斯拉瓦、布達佩斯、盧布亞納、布拉格	
中歐標準時間	(UTC+01:00)	塞拉耶佛、斯高彼亞、華沙、札格雷布	
中央太平洋標準時間	(UTC+11:00)	索羅門群島、新喀里多尼亞	此時區不會遵守日光節約時間。

時區	標準時間偏移	描述	備註
中央標準時間	(UTC-06:00)	中央時間 (美國和加拿大)	
中央標準時間 (墨西哥)	(UTC-06:00)	瓜達拉哈拉、墨西哥城、蒙特利	
查塔姆群島標準時間	(UTC+12:45)	查塔姆群島	
中國標準時間	(UTC+08:00)	北京、重慶、香港、烏魯木齊	此時區不會遵守日光節約時間。
古巴標準時間	(UTC-05:00)	哈瓦那	
國際換日線標準時間	(UTC-12:00)	國際換日線以西	此時區不會遵守日光節約時間。
東部非洲標準時間	(UTC+03:00)	奈洛比	此時區不會遵守日光節約時間。
東部澳洲標準時間	(UTC+10:00)	布里斯班	此時區不會遵守日光節約時間。
東部歐洲標準時間	(UTC+02:00)	奇西瑙	
東部南美洲標準時間	(UTC-03:00)	巴西利亞	
復活節島標準時間	(UTC-06:00)	復活節島	
東部標準時間	(UTC-05:00)	東部時間 (美國和加拿大)	
東部標準時間 (墨西哥)	(UTC-05:00)	切圖馬爾	
埃及標準時間	(UTC+02:00)	開羅	
葉卡捷琳堡標準時間	(UTC+05:00)	葉卡捷琳堡	
斐濟標準時間	(UTC+12:00)	斐濟	

時區	標準時間偏移	描述	備註
FLE 標準時間	(UTC+02:00)	赫爾辛基，基輔，里加，索菲亞，塔林，維爾紐斯	
喬治亞標準時間	(UTC+04:00)	提比里斯	此時區不會遵守日光節約時間。
GMT 標準時間	(UTC)	都柏林、愛丁堡、里斯本、倫敦	此時區與格林威治標準時間不同。此時區會遵守日光節約時間。
格陵蘭標準時間	(UTC-03:00)	格陵蘭	
格林威治標準時間	(UTC)	蒙羅維亞、雷克雅維克	此時區不會遵守日光節約時間。
GTB 標準時間	(UTC+02:00)	雅典、布加勒斯特	
海地標準時間	(UTC-05:00)	海地	
夏威夷標準時間	(UTC-10:00)	夏威夷州	
印度標準時間	(UTC+05:30)	清奈、加爾各答、孟買、新德里	此時區不會遵守日光節約時間。
伊朗標準時間	(UTC+03:30)	德黑蘭	
以色列標準時間	(UTC+02:00)	耶路撒冷	
約旦標準時間	(UTC+02:00)	安曼	
卡里寧格勒標準時間	(UTC+02:00)	卡里寧格勒	
堪察加標準時間	(UTC+12:00)	彼得保羅夫斯克-堪察加斯克 – 老	

時區	標準時間偏移	描述	備註
韓國標準時間	(UTC+09:00)	首爾	此時區不會遵守日光節約時間。
利比亞標準時間	(UTC+02:00)	的黎波里	
來因群島標準時間	(UTC+14:00)	刻里提瑪斯島	
羅豪標準時間	(UTC+10:30)	羅豪島	
馬加丹標準時間	(UTC+11:00)	馬加丹	此時區不會遵守日光節約時間。
麥哲倫標準時間	(UTC-03:00)	蓬塔阿雷納斯	
馬克薩斯標準時間	(UTC-09:30)	馬克薩斯群島	
模里西斯標準時間	(UTC+04:00)	路易士港	此時區不會遵守日光節約時間。
中東標準時間	(UTC+02:00)	貝魯特	
蒙特維多標準時間	(UTC-03:00)	蒙特維多	
摩洛哥標準時間	(UTC+01:00)	卡薩布蘭卡	
山地標準時間	(UTC-07:00)	山地時間 (美國和加拿大)	
山地標準時間 (墨西哥)	(UTC-07:00)	赤瓦瓦州、拉巴斯、馬薩特蘭	
緬甸標準時間	(UTC+06:30)	仰光	此時區不會遵守日光節約時間。
北部中亞標準時間	(UTC+07:00)	新西伯利亞	
納米比亞標準時間	(UTC+02:00)	溫吐克	

時區	標準時間偏移	描述	備註
尼泊爾標準時間	(UTC+05:45)	加德滿都	此時區不會遵守日光節約時間。
紐西蘭標準時間	(UTC+12:00)	奧克蘭、威靈頓	
紐芬蘭標準時間	(UTC-03:30)	紐芬蘭	
諾福克標準時間	(UTC+11:00)	諾福克島	
東北亞標準時間	(UTC+08:00)	伊爾庫次克	
北亞標準時間	(UTC+07:00)	克拉斯諾亞爾斯克	
北韓標準時間	(UTC+09:00)	平壤	
鄂木斯克標準時間	(UTC+06:00)	鄂木斯克	
太平洋 SA 標準時間	(UTC-03:00)	聖地牙哥	
太平洋標準時間	(UTC-08:00)	太平洋時間 (美國和加拿大)	
太平洋標準時間 (墨西哥)	(UTC-08:00)	下加利福尼亞州	
巴基斯坦標準時間	(UTC+05:00)	伊斯蘭馬巴德，喀拉蚩	此時區不會遵守日光節約時間。
巴拉圭標準時間	(UTC-04:00)	亞松森	
羅馬標準時間	(UTC+01:00)	布魯塞爾、哥本哈根、馬德里、巴黎	
俄羅斯時區 10	(UTC+11:00)	查庫爾達克	
俄羅斯時區 11	(UTC+12:00)	阿納底，彼得保羅夫斯克-堪察加斯克	
俄羅斯時區 3	(UTC+04:00)	伊熱夫斯克，薩馬拉	

時區	標準時間偏移	描述	備註
俄羅斯標準時間	(UTC+03:00)	莫斯科、聖彼得堡、伏爾加格勒	此時區不會遵守日光節約時間。
南美洲東部標準時間	(UTC-03:00)	卡宴，福塔雷薩	此時區不會遵守日光節約時間。
SA 太平洋標準時間	(UTC-05:00)	波哥大、利馬、基多、里約布蘭科	此時區不會遵守日光節約時間。
南美洲西部標準時間	(UTC-04:00)	喬治城，拉帕茲，瑪瑙斯，聖胡安	此時區不會遵守日光節約時間。
聖皮爾標準時間	(UTC-03:00)	聖皮埃赫及密克隆	
薩哈林標準時間	(UTC+11:00)	薩哈林	
薩摩亞標準時間	(UTC+13:00)	薩摩亞	
聖多美標準時間	(UTC+01:00)	聖多美普林西比	
薩拉托夫標準時間	(UTC+04:00)	薩拉托夫	
SE 亞洲標準時間	(UTC+07:00)	曼谷、河內、雅加達	此時區不會遵守日光節約時間。
新加坡標準時間	(UTC+08:00)	吉隆坡、新加坡	此時區不會遵守日光節約時間。
南非標準時間	(UTC+02:00)	哈拉雷，比勒陀利亞	此時區不會遵守日光節約時間。
斯里蘭卡標準時間	(UTC+05:30)	斯里哈亞華登尼普拉	此時區不會遵守日光節約時間。
蘇丹標準時間	(UTC+02:00)	喀土穆	
敘利亞標準時間	(UTC+02:00)	大馬士革	

時區	標準時間偏移	描述	備註
台北標準時間	(UTC+08:00)	台北	此時區不會遵守日光節約時間。
塔斯馬尼亞標準時間	(UTC+10:00)	荷巴特	
托坎廷斯標準時間	(UTC-03:00)	阿拉瓜	
東京標準時間	(UTC+09:00)	大阪、札幌、東京	此時區不會遵守日光節約時間。
托木斯克標準時間	(UTC+07:00)	托木斯克	
東加標準時間	(UTC+13:00)	努瓜婁發	此時區不會遵守日光節約時間。
外貝加爾標準時間	(UTC+09:00)	赤塔	
土耳其標準時間	(UTC+03:00)	伊斯坦堡	
土克斯及開科斯群島標準時間	(UTC-05:00)	土克斯及開科斯群島	
烏蘭巴托標準時間	(UTC+08:00)	烏蘭巴托	此時區不會遵守日光節約時間。
美國東部標準時間	(UTC-05:00)	印第安那州 (東部)	
美國山地標準時間	(UTC-07:00)	亞利桑那州	此時區不會遵守日光節約時間。
UTC	UTC	國際標準時間	此時區不會遵守日光節約時間。
UTC-02	(UTC-02:00)	國際標準時間-02	此時區不會遵守日光節約時間。
UTC-08	(UTC-08:00)	國際標準時間-08	
UTC-09	(UTC-09:00)	國際標準時間-09	

時區	標準時間偏移	描述	備註
UTC-11	(UTC-11:00)	國際標準時間-11	此時區不會遵守日光節約時間。
UTC+12	(UTC+12:00)	國際標準時間+12	此時區不會遵守日光節約時間。
UTC+13	(UTC+13:00)	國際標準時間+13	
委內瑞拉標準時間	(UTC-04:00)	加拉加斯	此時區不會遵守日光節約時間。
符拉迪沃斯托克標準時間	(UTC+10:00)	符拉迪沃斯托克	
伏爾加格勒標準時間	(UTC+04:00)	伏爾加格勒	
西部澳洲標準時間	(UTC+08:00)	珀斯	此時區不會遵守日光節約時間。
西部西非標準時間	(UTC+01:00)	中西非	此時區不會遵守日光節約時間。
西部歐洲標準時間	(UTC+01:00)	阿姆斯特丹、柏林、羅馬、斯德哥爾摩、維也納	
西部蒙古標準時間	(UTC+07:00)	霍夫德	
西亞標準時間	(UTC+05:00)	阿什哈巴德，塔什干	此時區不會遵守日光節約時間。
約旦河西岸標準時間	(UTC+02:00)	加薩，希伯侖	
西太平洋標準時間	(UTC+10:00)	關島，莫爾斯貝港	此時區不會遵守日光節約時間。
亞庫茲克標準時間	(UTC+09:00)	亞庫茲克	

Amazon RDS 上的 Microsoft SQL Server 授權

當您設定 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體，軟體授權已包含在其中。

這表示您不需要另外購買 SQL Server 授權。AWS 本身已擁有 SQL Server 資料庫軟體的授權。Amazon RDS 會結合軟體授權、底層硬體資源和 Amazon RDS 管理功能，並予以定價。

Amazon RDS 支援下列 Microsoft SQL Server 版本：

- Enterprise
- 標準
- Web
- Express

Note

SQL Server Web Edition 的授權僅支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務。為了符合 Microsoft 的使用權利規定，需要此等級的支援。如需詳細資訊，請參閱 [AWS 服務條款](#)。

Amazon RDS 會透過使用 SQL Server Database Mirroring (DBM) 或 Always On 可用性群組 (AG)，支援在執行 Microsoft SQL Server 的資料庫執行個體上進行異地同步備份部署。異地同步備份部署沒有其他授權要求。如需更多詳細資訊，請參閱 [Amazon RDS for Microsoft SQL Server 的異地同步備份部署](#)。

還原授權終止的資料庫執行個體

Amazon RDS 可針對授權終止的資料庫執行個體建立快照。如果您的執行個體因授權問題而終止，您可以從快照將其還原成新的資料庫執行個體。新的資料庫執行個體會包含授權。

如需更多詳細資訊，請參閱 [還原授權終止的資料庫執行個體](#)。

開發和測試

因為授權規定，我們無法在 Amazon RDS 上提供 SQL Server 開發人員版本。您可以使用 Express 版本進行許多開發、測試、及其他非生產需求。不過，如果您需要 SQL Server 企業級安裝的完整功能以進行開發，您可以使用具備 BYOM 的 CEV 在 SQL Server 的 RDS 自訂版上下載並安裝 SQL Server

開發人員版本。如需詳細資訊，請參閱[使用自有媒體 \(BYOM\) 準備 CEV](#)。開發人員版本不需要專用基礎設施。使用您自己的主機，您也可以使用無法在 Amazon RDS 上使用的其他可程式性功能。如需有關 SQL Server 版本之間差異的詳細資訊，請參閱 Microsoft 文件中的[SQL Server 2019 版本和支援的功能](#)。

連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體

在 Amazon RDS 佈建您的資料庫執行個體之後，您就可以使用任何標準 SQL 用戶端應用程式來連接至資料庫執行個體。在此主題中，您會使用 Microsoft SQL Server Management Studio (SSMS) 或 SQL Workbench/J 來連接至您的資料庫執行個體。

如需引導您進行建立和連接至範例資料庫執行個體的程序範例，請參閱 [建立並連線至 Microsoft SQL Server 資料庫執行個體](#)。

連線之前

連線到資料庫執行個體之前，資料庫執行個體必須可用且可存取。

1. 請確定其狀態為 available。您可以在執行個體的詳細資訊頁面上，AWS Management Console 或使用 [describe-db-instances](#) AWS CLI 指令來檢查此項。

RDS > Databases > database-2

database-2

Modify Actions

Summary

DB identifier database-2	CPU 7.42%	Status Available	Class db.r4.large
Role Instance	Current activity 0 Sessions	Engine SQL Server Standard Edition	Region & AZ us-west-2d

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port Endpoint database-2. .us-west-2.rds.amazonaws. Port 1433	Networking Availability zone us-west-2d VPC vpc- Subnet group default	Security VPC security groups default (sg-) (active) Public accessibility Yes Certificate authority rds-ca-2019
---	--	--

2. 確保您的來源可以存取它。根據您的案例，可能不需要公開存取它。如需詳細資訊，請參閱 [Amazon VPC 和 Amazon RDS](#)。
3. 請確定 VPC 安全群組的傳入規則允許存取您的資料庫執行個體。如需更多詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

尋找資料庫執行個體端點和連接埠號碼

您需要同時有端點和連接埠號碼，才能連接至資料庫執行個體。

尋找端點和連接埠

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體的 AWS 區域。
3. 尋找資料庫執行個體的網域名稱系統 (DNS) 名稱 (端點) 和連接埠號碼：
 - a. 開啟 RDS 主控台，然後選擇 Databases (資料庫)，以顯示資料庫執行個體清單。
 - b. 選擇 SQL Server 資料庫執行個體名稱以顯示其詳細資訊。
 - c. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。

The screenshot displays the configuration page for an Amazon RDS database instance named 'database-2'. The page is divided into several sections:

- Summary:** A table listing key attributes:

DB identifier	database-2	CPU
Role		Current
Instance		
- Navigation:** Three tabs are visible: 'Connectivity & security' (selected), 'Monitoring', and 'Logs & metrics'.
- Connectivity & security:** A sub-section containing:
 - Endpoint & port:**
 - Endpoint:** database-2. [redacted].us-east-2.rds.amazonaws.com
 - Port:** 1433

- d. 請記下連接埠號碼。

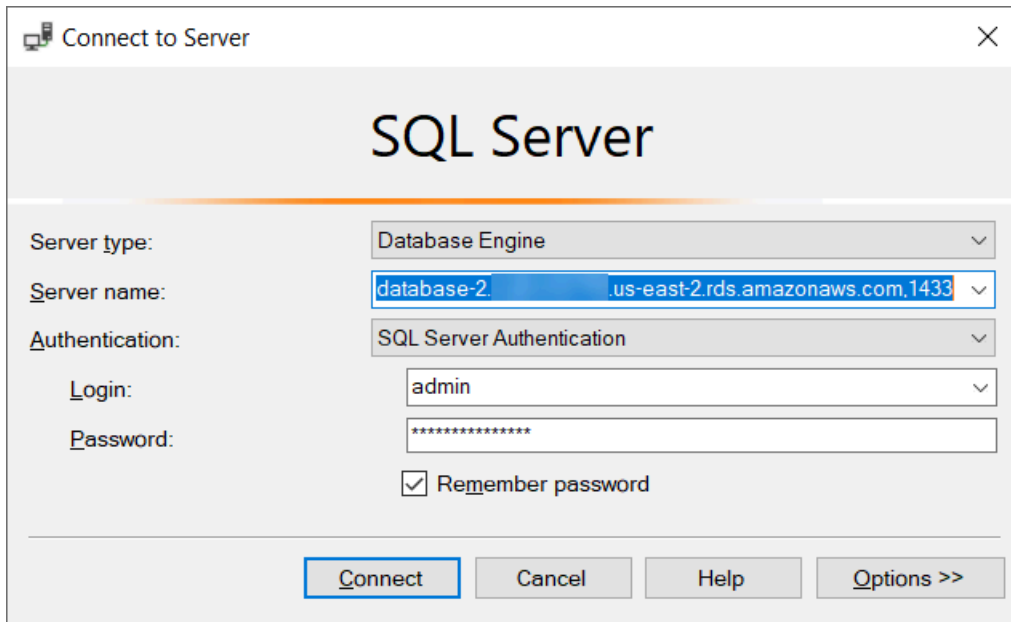
使用 Microsoft SQL Server Management Studio 連接至您的資料庫執行個體

在此程序中，您會使用 Microsoft SQL Server Management Studio (SSMS) 連接至您的範例資料庫執行個體。若要下載此公用程式的獨立版本，請參閱 Microsoft 文件中的 [下載 SQL Server Management Studio \(SSMS\)](#)。

使用 SSMS 連接至資料庫執行個體

1. 啟動 SQL Server Management Studio。

Connect to Server (連接至伺服器) 對話方塊隨即出現。



2. 提供資料庫執行個體的資訊：

- a. 針對 Server type (伺服器類型)，選擇 Database Engine (資料庫引擎)。
- b. 針對 Server name (伺服器名稱)，輸入資料庫執行個體的 DNS 名稱 (端點) 和連接埠號碼，以逗號區隔。

⚠ Important

將端點和連接埠號碼之間的冒號變更為逗號。

您的伺服器名稱應看起來與下列範例類似。

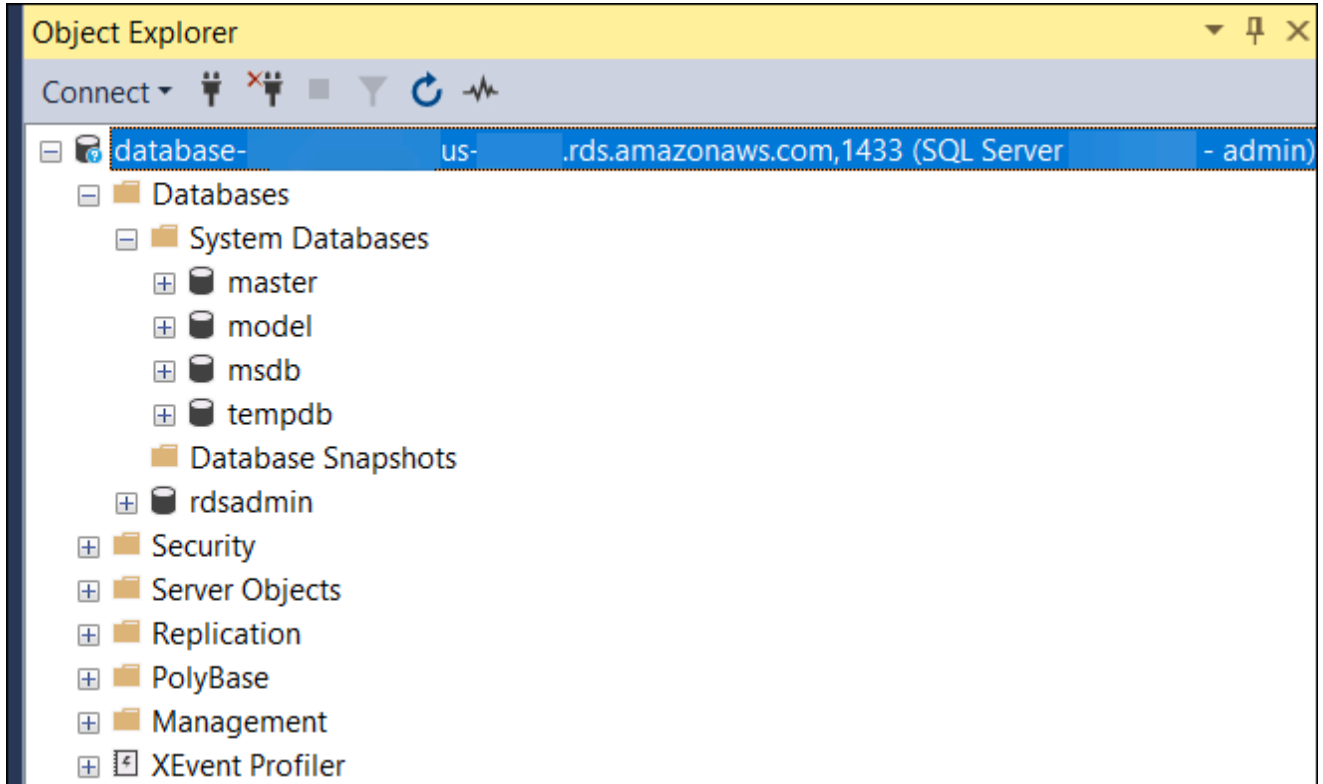
```
database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433
```

- c. 針對 Authentication (身分驗證)，選擇 SQL Server Authentication (SQL Server 身分驗證)。
 - d. 針對 Login (登入)，輸入資料庫執行個體的主要使用者名稱。
 - e. 針對 Password (密碼)，輸入資料庫執行個體的密碼。
3. 選擇 Connect (連線)。

幾分鐘後，SSMS 會連接至資料庫執行個體。

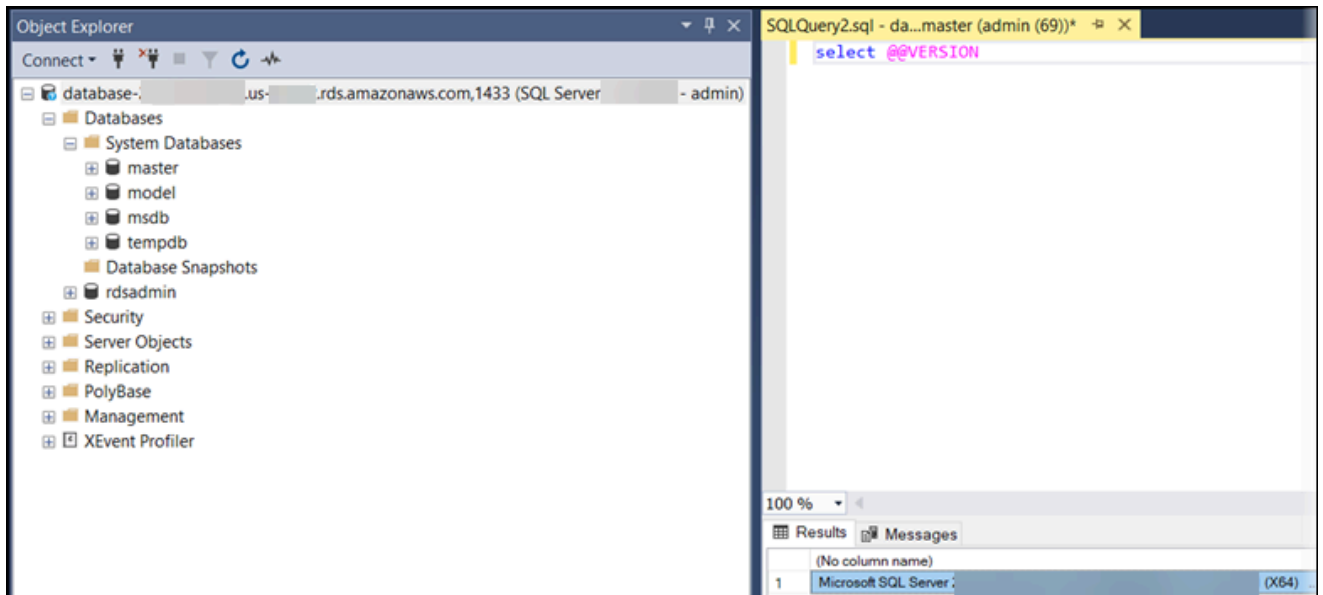
如果您無法連接至資料庫執行個體，請參閱 [安全群組考量](#) 和 [對您的 SQL Server 資料庫執行個體的連線進行故障診斷](#)。

4. 您的 SQL Server 資料庫執行個體隨附 SQL Server 的標準內建系統資料庫 (master、model、msdb 和 tempdb)。若要瀏覽系統資料庫，請執行下列動作：
 - a. 在 SSMS 的 View (檢視) 功能表中，選擇 Object Explorer (物件總管)。
 - b. 依序展開您的資料庫執行個體、資料庫和系統資料庫。



5. 您的 SQL Server 資料庫執行個體也隨附一個名為 rdsadmin 的資料庫。Amazon RDS 將使用此資料庫，以便存放用於管理資料庫的物件。rdsadmin 資料庫也包含可供您執行的預存程序，可讓您執行進階任務。如需更多詳細資訊，請參閱 [Microsoft SQL Server 的常用 DBA 任務](#)。
6. 您現在可以開始建立您自己的資料庫，並照常對您的資料庫執行個體和資料庫執行查詢。若要對您的資料庫執行個體執行測試查詢，請執行下列動作：
 - a. 在 SSMS 中，於 File (檔案) 功能表上，指向 New (新增)，然後選擇 Query with Current Connection (使用目前的連接查詢)。
 - b. 輸入下列 SQL 查詢。

```
select @@VERSION
```
 - c. 執行查詢。SSMS 會傳回 Amazon RDS 資料庫執行個體的 SQL Server 版本。



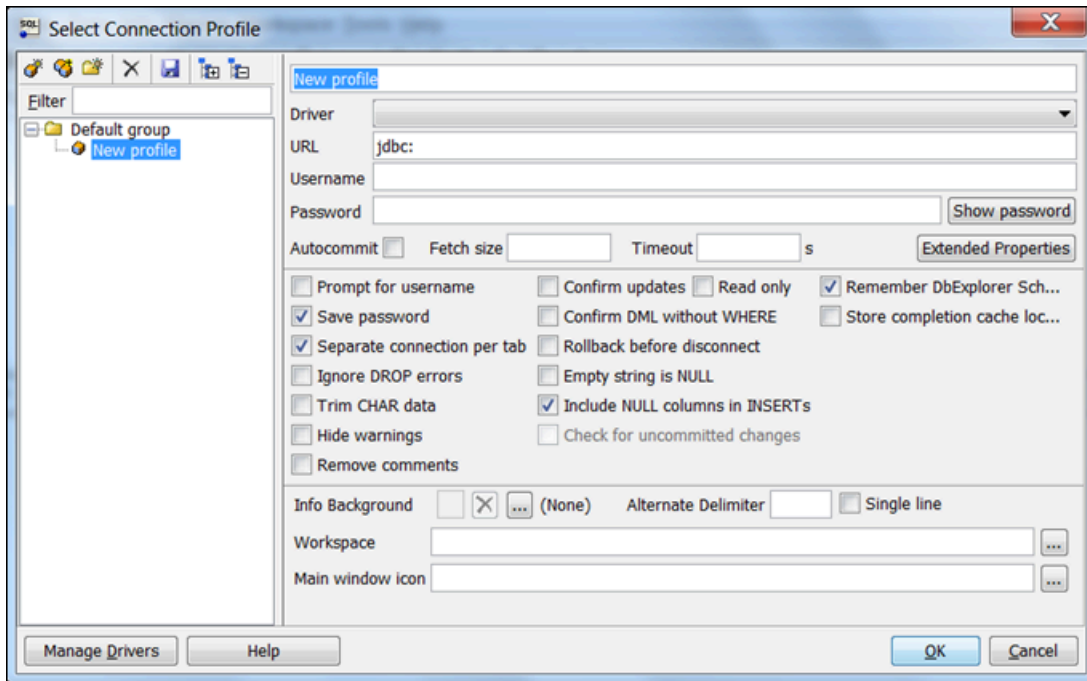
使用 SQL Workbench/J 連接至您的資料庫執行個體

此範例示範如何使用 SQL Workbench/J 資料庫工具來連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體。若要下載 SQL Workbench/J，請參閱 [SQL Workbench/J](#)。

SQL Workbench/J 使用 JDBC 來連接至您的資料庫執行個體。您也需要適用於 SQL Server 的 JDBC 驅動程式。若要下載此驅動程式，請參閱 [SQL 伺服器的 Microsoft JDBC 驅動程式 6.0](#)。

使用 SQL Workbench/J 連接至資料庫執行個體

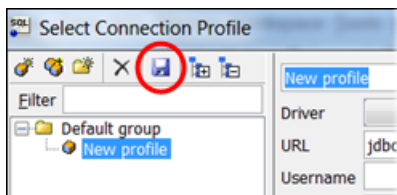
1. 開啟 SQL Workbench/J。Select Connection Profile (選取連線描述檔) 對話方塊隨即出現，如下所示。



2. 在對話方塊的第一個方塊中，輸入設定檔的名稱。
3. 針對 Driver (驅動程式)，選擇 **SQL JDBC 4.0**。
4. 針對 URL，輸入 **jdbc:sqlserver://**，接著輸入資料庫執行個體的端點。例如，URL 值可能如下。

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

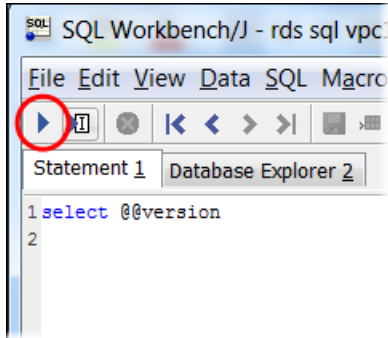
5. 針對 Username (使用者名稱)，輸入資料庫執行個體的主要使用者名稱。
6. 針對 Password (密碼)，輸入主要使用者的密碼。
7. 選擇對話方塊工具列中的儲存圖示，如下所示。



8. 選擇 OK (確定)。幾分鐘後，SQL Workbench/J 會連接至資料庫執行個體。如果您無法連接至資料庫執行個體，請參閱 [安全群組考量](#) 和 [對您的 SQL Server 資料庫執行個體的連線進行故障診斷](#)。
9. 在查詢窗格中，輸入下列 SQL 查詢。

```
select @@VERSION
```

10. 選擇工具列中的 Execute 圖示，如下所示。



查詢會傳回資料庫執行個體的版本資訊，類似如下。

```
Microsoft SQL Server 2017 (RTM-CU22) (KB4577467) - 14.0.3356.20 (X64)
```

安全群組考量

若要連線至您的資料庫執行個體，資料庫執行個體必須與安全群組相關聯。此安全性群組包含您用於存取資料庫執行個體的 IP 位址和網路組態。您可能已在建立資料庫執行個體時，將資料庫執行個體與適當的安全群組建立關聯。如果在建立資料庫執行個體時指派預設值、非設定的安全群組，您的資料庫執行個體防火牆會防止連線。

在某些情況下，您可能需要建立新的安全群組才能進行存取。如需建立新安全群組的指示，請參閱 [使用安全群組控制存取](#)。如需引導您為 VPC 安全群組設定規則之程序的主題，請參閱 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。

建立新安全群組之後，修改資料庫執行個體，讓它與該安全群組建立關聯。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

您可以使用 SSL 來加密對資料庫執行個體的連線，藉此增強安全性。如需更多詳細資訊，請參閱 [對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)。

對您的 SQL Server 資料庫執行個體的連線進行故障診斷

下表顯示當您嘗試連線到 SQL Server DB 執行個體時，可能會遇到的錯誤訊息。

問題	故障診斷建議
Could not open a connection to SQL Server –	請確認指定的伺服器名稱正確。針對 Server name (伺服器名稱)，輸入範例資料庫執行個體的 DNS 名稱和連接埠號碼，以逗號區隔。

問題	故障診斷建議
Microsoft SQL Server, Error: 53 (無法開啟 SQL Server 的連線 – Microsoft SQL Server 發生錯誤 : 53)	<p> Important</p> <p>如果 DNS 名稱和連接埠號碼之間有冒號，請將冒號變更為逗號。</p> <p>您的伺服器名稱應看起來與下列範例類似。</p> <pre data-bbox="565 569 1507 688">sample-instance.cg034itsfake.us-east-1.rds.am amazonaws.com,1433</pre>
No connection could be made because the target machine actively refused it – Microsoft SQL Server, Error: 10061 (目標機器主動拒絕，因此無法進行連線 – Microsoft SQL Server 發生錯誤 : 10061)	<p>您可以連接資料庫執行個體，但連線被拒。此問題通常是因為指定了錯誤的使用者名稱或密碼所造成。驗證使用者名稱和密碼，然後重試。</p>
建立與 SQL Server 的連線時，發生網路相關或執行個體特定的錯誤。找不到伺服器或無法存取... 等候作業已逾時 – Microsoft SQL Server，錯誤 : 258	<p>您的本機防火牆強制執行的存取規則，與您獲授權可存取資料庫執行個體的 IP 地址可能不符。問題很可能在於安全群組中的傳入規則。如需詳細資訊，請參閱 Amazon RDS 中的安全。</p> <p>您的資料庫執行個體必須可公開存取。若要從 VPC 外部連線到它，執行個體必須指派公用 IP 位址。</p>

Note

如需連線問題的詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

使用 Active Directory 搭配 RDS for SQL Server

您可以將 RDS for SQL Server 資料庫執行個體加入至 Microsoft Active Directory (AD) 網域。您的 AD 網域可在 AWS 內的 AWS 受管 AD 上或在您所選位置 (包括貴公司資料中心) 的自我管理 AD 上進行託管、在 AWS EC2 上進行託管，或是透過其他雲端提供者託管。

您可以搭配自我管理 Active Directory 使用 NTLM 驗證，來驗證網域使用者。您可以搭配 AWS 受管 Active Directory 使用 Kerberos 和 NTLM 驗證。

在以下各節中，您可以找到在 Amazon RDS 上為 Microsoft SQL Server 使用自我管理 Active Directory 和 AWS 受管 Active Directory 的相關資訊。

主題

- [搭配 Amazon RDS for SQL Server 資料庫執行個體使用自我管理 Active Directory](#)
- [使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄](#)

搭配 Amazon RDS for SQL Server 資料庫執行個體使用自我管理 Active Directory

您可以將 RDS for SQL Server 資料庫執行個體直接加入自我管理的活動目錄 (AD) 網域，無論您的 AD 託管在何處：企業資料中心、AWS EC2 或其他雲端提供者。搭配自我管理 AD，您可以使用 NTLM 身分驗證，直接控制 RDS for SQL Server 資料庫執行個體上使用者和服務的身分驗證，而無需使用中繼網域和樹系信任。當使用者透過加入自我管理 AD 網域的 RDS for SQL Server 資料庫執行個體進行驗證時，身分驗證請求會轉送至您指定的自我管理 AD 網域。

主題

- [區域和版本可用性](#)
- [要求](#)
- [限制](#)
- [設定自我管理 Active Directory 的概觀](#)
- [設定自我管理 Active Directory](#)
- [在自我管理 Active Directory 網域中管理資料庫執行個體](#)
- [了解自我管理 Active Directory 網域成員資格](#)
- [對自我管理 Active Directory 進行疑難排解](#)
- [還原 SQL Server 資料庫執行個體，然後將其新增至自我管理 Active Directory 網域](#)

區域和版本可用性

Amazon RDS 支援在所有 AWS 區域中使用 NTLM 之 SQL Server 的自我管理 AD。

要求

在將 RDS for SQL Server 資料庫執行個體加入自我管理 AD 網域之前，請確定您已符合下列需求。

主題

- [設定內部部署 AD](#)
- [設定您的網路連線能力](#)
- [設定您的 AD 網域服務帳戶](#)

設定內部部署 AD

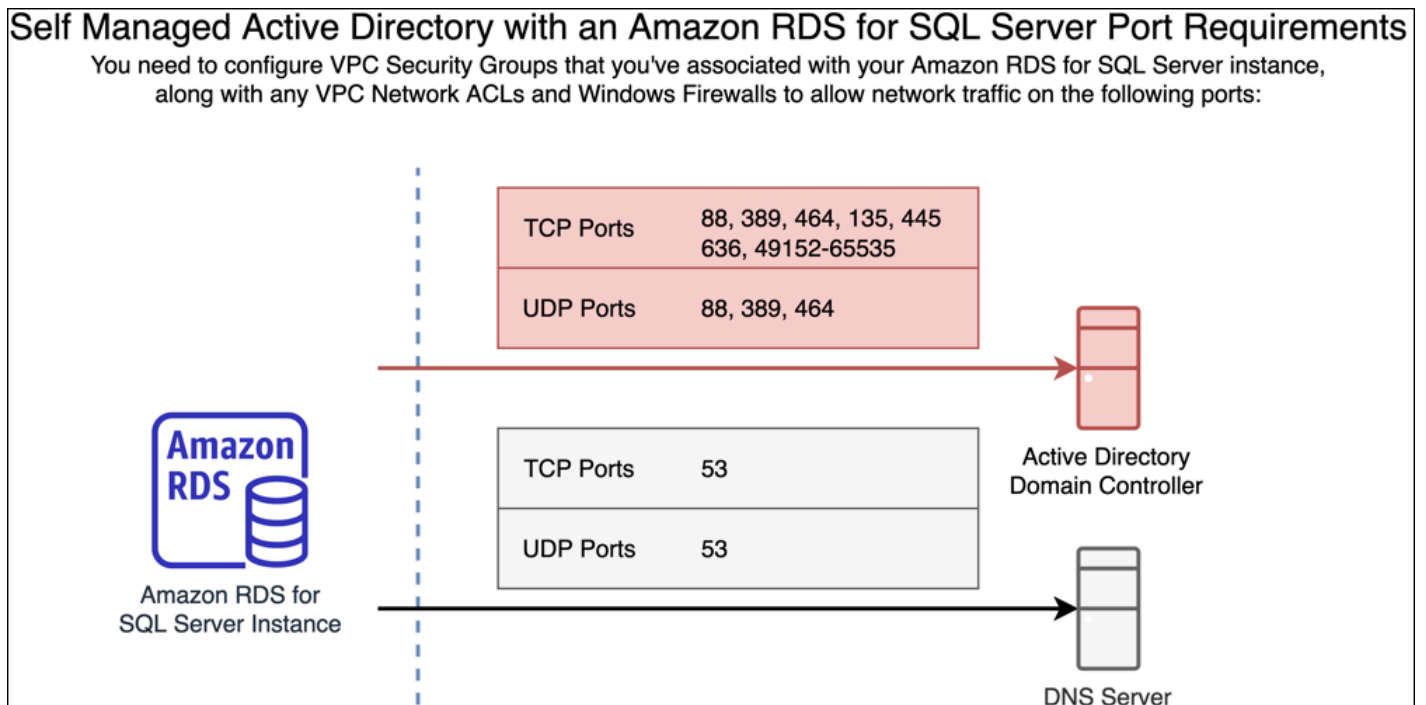
請確定您具有可以將 Amazon RDS for SQL Server 執行個體加入其中的內部部署或其他自我管理 Microsoft AD。您的內部部署 AD 應該具有下列組態：

- 如果您已定義 Active Directory 網站，請確定 VPC 中與您的 RDS for SQL Server 資料庫執行個體相關聯的子網路已定義在您的 Active Directory 網站中。確認 VPC 中的子網路與其他 AD 網站中的子網路之間沒有任何衝突。
- 您的 AD 網域控制器具有 Windows Server 2008 R2 或更新版本的網域功能層級。
- 您的 AD 網域名稱不能採用單一標籤網域 (SLD) 格式。RDS for SQL Server 不支援 SLD 網域。
- AD 的完整網域名稱 (FQDN) 不得超過 64 個字元。

設定您的網路連線能力

請確定您已符合下列網路組態：

- 在 Amazon VPC 之間設定的連線能力，而在 Amazon VPC，您想要建立 RDS for SQL Server 資料庫執行個體，以及您的自我管理 Active Directory。您可以使用 AWS 直 Connect 線、AWS VPN、VPC 對等互連或 AWS Transit Gateway 來設定連線。
- 對於 VPC 安全群組，預設 Amazon VPC 的預設安全群組已新增至主控台當中的 RDS for SQL Server 資料庫執行個體。請確定您要在其中建立 RDS for SQL Server 資料庫執行個體之子網路的安全群組和 VPC 網路 ACL 允許連接埠上的流量，並依下圖所示的方向前進。



下表識別每個連接埠的角色。

通訊協定	連接埠	角色
TCP/UDP	53	網域名稱系統 (DNS)
TCP/UDP	88	Kerberos 身分驗證
TCP/UDP	464	變更/設定密碼
TCP/UDP	389	輕量型目錄存取通訊協定 (LDAP)
TCP	135	分散式運算環境/端點映射器 (DCE/EPMAP)
TCP	445	目錄服務 SMB 檔案共用
TCP	636	透過 TLS/SSL 的輕量型目錄存取通訊協定 (LDAPS)
TCP	49152 - 65535	適用於 RPC 的暫時性連接埠

- 一般而言，網域 DNS 伺服器位於 AD 網域控制器中。您不需要設定 VPC DHCP 選項集，即可使用此功能。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [DHCP 選項集](#)。

Important

如果使用 VPC 網路 ACL，您也必須允許動態連接埠 (49152-65535) 上來自 RDS for SQL Server 資料庫執行個體的傳出流量。請確定也會在適用於每個 AD 網域控制器、DNS 伺服器和 RDS for SQL Server 資料庫執行個體的防火牆上鏡像這些流量規則。

雖然 VPC 安全群組只需要以網路流量起始的方向開啟連接埠，但大多數 Windows 防火牆和 VPC 網路 ACL 都需要雙向開啟連接埠。

設定您的 AD 網域服務帳戶

請確定您已符合 AD 網域服務帳戶的下列需求：

- 確定您在自我管理 AD 網域中具有一個服務帳戶，其具有將電腦加入網域的委派許可。網域服務帳戶是自我管理 AD 中的使用者帳戶，其已獲委派執行特定任務的許可。
- 在您正要將 RDS for SQL Server 資料庫執行個體加入其中的組織單位 (OU) 中，網域服務帳戶必須獲得委派下列許可：
 - 已驗證能夠寫入 DNS 主機名稱
 - 已驗證能夠寫入服務主體名稱
 - 建立和刪除電腦物件

這些代表將電腦物件加入自我管理 Active Directory 所需的最低許可集。如需詳細資訊，請參閱 Microsoft Windows Server 文件中的 [嘗試將電腦加入網域時發生錯誤](#)。

Important

在建立資料庫執行個體之後，請不要移動 RDS for SQL Server 在組織單位中建立的電腦物件。移動相關聯的物件會導致 RDS for SQL Server 資料庫執行個體設定錯誤。如果您需要移動 Amazon RDS 所建立的電腦物件，請使用 [ModifyDBInstance](#) RDS API 操作，透過所需的電腦物件位置修改網域參數。

限制

下列限制適用於 SQL Server 的自我管理 AD。

- NTLM 是唯一支援的身分驗證類型。不支援 Kerberos 身分驗證。如果您需要使用 kerberos 驗證，您可以使用 AWS 受管理 AD 而非自我管理 AD。
- Microsoft 分散式交易協調器 (MSDTC) 服務不受支援，因為其需要 Kerberos 身分驗證。
- 您的 RDS for SQL Server 資料庫執行個體不會使用自我管理 AD 網域的網路時間通訊協定 (NTP) 伺服器。他們改用 AWS NTP 服務。
- SQL Server 連結伺服器必須使用 SQL 身分驗證，才能連線到已加入自我管理 AD 網域的其他 RDS for SQL Server 資料庫執行個體。
- 自我管理 AD 網域中的 Microsoft 群組原則物件 (GPO) 設定不會套用至 RDS for SQL Server 資料庫執行個體。

設定自我管理 Active Directory 的概觀

若要為 RDS for SQL Server 資料庫執行個體設定自我管理 AD，請採取下列步驟，其詳細說明在[設定自我管理 Active Directory](#) 中：

在您的 AD 網域中：

- 建立組織單位 (OU)。
- 建立 AD 網域使用者。
- 將控制權委派給 AD 網域使用者。

從 AWS Management Console 或應用程式介面：

- 建立 AWS KMS 金鑰。
- 使用密碼管理員建立 AWS 密碼。
- 建立或修改 RDS for SQL Server 資料庫執行個體，並將其加入您的自我管理 AD 網域。

設定自我管理 Active Directory

若要設定自我管理 AD，請採取下列步驟。

主題

- [步驟 1：在您的 AD 中建立組織單位](#)
- [步驟 2：在您的 AD 中建立 AD 網域使用者](#)
- [步驟 3：將控制權委派給 AD 使用者](#)
- [步驟 4：建立 AWS KMS 金鑰](#)
- [步驟 5：建立 AWS 密碼](#)
- [步驟 6：建立或修改 SQL Server 資料庫執行個體](#)
- [步驟 7：建立 Windows 身分驗證 SQL Server 登入](#)

步驟 1：在您的 AD 中建立組織單位

Important

我們建議針對擁有 RDS for SQL Server 資料庫執行個體加入您自我管理 AD 網域的任何 AWS 帳戶，建立範圍為該 OU 的專用 OU 和服務認證。透過專用 OU 和服務憑證，您可以避免衝突的許可，並遵循最低權限的主體。

在您的 AD 中建立 OU

1. 以網域管理員身分連線至您的 AD 網域。
2. 開啟 Active Directory 使用者和電腦，然後選取您要在其中建立 OU 的網域。
3. 在網域上按一下滑鼠右鍵，然後選擇新增，再選擇組織單位。
4. 輸入 OU 的名稱。
5. 保持選取保護容器免遭意外刪除的方塊。
6. 按一下 OK (確定)。您的新 OU 會出現在您的網域下方。

步驟 2：在您的 AD 中建立 AD 網域使用者

網域使用者認證將用於秘 AWS 密管理員中的密碼。

在您的 AD 中建立 AD 網域使用者

1. 開啟 Active Directory 使用者和電腦，然後選取您要在其中建立使用者的網域。
2. 在使用者上按一下滑鼠右鍵，然後選擇新增，再選擇使用者。
3. 輸入使用者的名字、姓氏和登入名稱。按一下 Next (下一步)。
4. 輸入使用者的密碼。不要選取「使用者在下次登入時必須變更密碼」。不要選取「帳戶已停用」。按一下 Next (下一步)。
5. 按一下 OK (確定)。您的新使用者會出現在您的網域下方。

步驟 3：將控制權委派給 AD 使用者

將控制權委派給網域中的 AD 網域使用者

1. 開啟 Active Directory 使用者和電腦 MMC 嵌入式管理單元，然後選取您要在其中建立使用者的網域。

2. 在您先前建立的 OU 上按一下滑鼠右鍵，然後選擇委派控制權。
3. 在委派控制權精靈頁面上，按下一步。
4. 在使用者或群組區段上，按一下新增。
5. 在選取使用者、電腦或群組區段上，輸入您建立的 AD 使用者，然後按一下檢查名稱。如果 AD 使用者檢查成功，請按一下確定。
6. 在使用者或群組區段上，確認已新增 AD 使用者，然後按下一步。
7. 在要委派的任務區段上，選取要建立要委派的自訂任務，然後按下一步。
8. 在 Active Directory 物件類型區段上：
 - a. 選擇僅限資料夾中的下列物件。
 - b. 選取 [電腦物件]。
 - c. 選取在此資料夾中建立選取的物件。
 - d. 選取刪除此資料夾中選取的物件，然後按下一步。
9. 在許可區段上：
 - a. 保持選取一般。
 - b. 選取已驗證寫入 DNS 主機名稱。
 - c. 選取已驗證寫入服務主體名稱，然後按下一步。
10. 對於完成委派控制權精靈，請檢閱並確認您的設定，然後按一下完成。

步驟 4：建立 AWS KMS 金鑰

KMS 金鑰是用來加密您的 AWS 密碼。

建立 AWS KMS 金鑰的步驟

Note

對於加密金鑰，請勿使用 AWS 預設的 KMS 金鑰。請務必在包含要加入自我管理 AD 的 RDS for SQL Server 資料庫執行個體的相同 AWS 帳戶中建立 AWS KMS 金鑰。

1. 在 AWS KMS 主控台中，選擇 [建立金鑰]。
2. 對於金鑰類型，選擇對稱。

3. 對於金鑰用途，選擇加密和解密。
4. 針對 Advanced options (進階選項)：
 - a. 對於金鑰材料來源，選擇 KMS。
 - b. 對於區域性，選擇單一區域金鑰，然後按下一步。
5. 對於別名，提供 KMS 金鑰的名稱。
6. (選用) 對於描述，提供 KMS 金鑰的描述。
7. (選用) 對於標籤，提供 KMS 金鑰的標籤，然後按下一步。
8. 對於金鑰管理員，提供 IAM 使用者的名稱，然後選取該名稱。
9. 對於金鑰刪除，保持選取允許金鑰管理員刪除此金鑰的方塊，然後按下一步。
10. 對於金鑰使用者，提供上一個步驟中相同的 IAM 使用者，然後選取該使用者。按一下 Next (下一步)。
11. 檢閱組態。
12. 對於金鑰政策，在政策聲明中包含下列內容：

```
{
  "Sid": "Allow use of the KMS key on behalf of RDS",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "rds.amazonaws.com"
    ]
  },
  "Action": "kms:Decrypt",
  "Resource": "*"
}
```

13. 按一下 Finish (完成)。


步驟 5：建立 AWS 密碼

若要建立機密

Note

請務必在包含要加入自我管理 AD 的 RDS for SQL Server 資料庫執行個體的相同 AWS 帳戶中建立密碼。

1. 在 AWS Secrets Manager 中，選擇儲存新密碼。
2. 針對機密類型，選擇其他類型的機密。
3. 對於金鑰/值對，新增兩個金鑰：
 - a. 對於第一個金鑰，輸入 CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME。
 - b. 對於第一個金鑰的值，輸入您在上一個步驟中於網域上建立的 AD 使用者名稱。
 - c. 對於第二個金鑰，輸入 CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD。
 - d. 對於第二個金鑰的值，輸入您在網域上為 AD 使用者建立的密碼。
4. 對於加密金鑰，輸入您在上一個步驟中建立的 KMS 金鑰，然後按下一步。
5. 對於秘密名稱，輸入可協助您稍後尋找密碼的描述性名稱。
6. (選用) 對於描述，輸入秘密名稱的描述。
7. 對於資源許可，按一下編輯。
8. 請將下列政策新增至許可政策：

 Note

建議您使用政策中的 `aws:sourceAccount` 和 `aws:sourceArn` 條件金鑰，保護自己免受混淆代理人問題的困擾。使用您的 `aws:sourceAccount` 和 AWS 帳戶適用於 SQL 伺服器的 RDS 資料庫執行個體 ARN。 `aws:sourceArn` 如需詳細資訊，請參閱 [防止跨服務混淆代理人問題](#)。

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Effect": "Allow",
      "Principal":
      {
        "Service": "rds.amazonaws.com"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*",
      "Condition":
      {
        "StringEquals":
```



```
{
  "aws:sourceAccount": "123456789012"
},
"ArnLike":
{
  "aws:sourceArn": "arn:aws:rds:us-west-2:123456789012:db:*"
}
}
]
```

9. 按一下儲存，然後按下一步。
10. 對於設定輪換設定，保留預設值並選擇下一步。
11. 檢閱秘密的設定，然後按一下存放。
12. 選擇您建立的秘密，然後複製秘密 ARN 的值。這將在下一個步驟中用來設定自我管理 Active Directory。

步驟 6：建立或修改 SQL Server 資料庫執行個體

您可以使用主控台、CLI 或 RDS API，將 RDS for SQL Server 資料庫執行個體與自我管理 AD 網域建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) CLI 命令或建立資料庫執行個體 RDS API [作業，建立新的 SQL Server 資料庫執行個體](#)。

如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[modify-db-instance](#) CLI 命令或修改資料庫執行個體 RDS API 作業來[修改現有的 SQL Server 資料庫執行個體](#)。

如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[restore-db-instance-from-db 快照 CLI 命令或還原資料庫快照 RDS API 作業，從資料庫快照還原 SQL Server 資料庫執行個體](#)。[InstanceFrom](#)

如需說明，請參閱[從資料庫 快照還原](#)。

- point-in-time 使用主控台、[restore-db-instance-to-point-in-time](#) CLI 命令或還原 InstanceToPointInTime RDS API 作業，將 SQL Server 資料庫執行個體[還原](#)至。

如需說明，請參閱[將資料庫執行個體還原至指定的時間](#)。

使用時 AWS CLI，資料庫執行個體需要下列參數才能使用您建立的自我管理 Active Directory 網域：

- 對於 `--domain-fqdn` 參數，使用自我管理 Active Directory 的完整網域名稱 (FQDN)。
- 對於 `--domain-ou` 參數，使用您在自我管理 AD 中建立的 OU。
- 對於 `--domain-auth-secret-arn` 參數，使用您在上一個步驟中所建立之秘密 ARN 的值。
- 對於 `--domain-dns-ips` 參數，針對自我管理 AD 使用 DNS 伺服器的主要和次要 IPv4 地址。如果您沒有次要 DNS 伺服器 IP 地址，請輸入主要 IP 地址兩次。

下列範例 CLI 命令說明如何建立、修改及移除具有自我管理 AD 網域的 RDS for SQL Server 資料庫執行個體。

Important

如果您修改資料庫執行個體，以將其加入自我管理 AD 網域或從中移除，則需要重新啟動資料庫執行個體，修改才會生效。您可以選擇立即套用變更，也可以等到下一個維護時段。選擇立即套用選項會導致單一可用區域資料庫執行個體停機。多可用區域資料庫執行個體將在完成重新啟動之前執行容錯移轉。如需詳細資訊，請參閱 [排程修改設定](#)。

下列 CLI 命令會建立新的 RDS for SQL Server 資料庫執行個體，並將其加入自我管理 AD 網域。

對於 Linux macOS、或 Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier my-DB-instance \  
  --db-instance-class db.m5.xlarge \  
  --allocated-storage 50 \  
  --engine sqlserver-se \  
  --engine-version 15.00.4043.16.v1 \  
  --license-model license-included \  
  --master-username my-master-username \  
  --master-user-password my-master-password \  
  --domain-fqdn my_AD_domain.my_AD.my_domain \  
  --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \  
  --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-AD-test-secret-123456" \  
  --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

在 Windows 中：

```
aws rds create-db-instance ^
  --db-instance-identifier my-DB-instance ^
  --db-instance-class db.m5.xlarge ^
  --allocated-storage 50 ^
  --engine sqlserver-se ^
  --engine-version 15.00.4043.16.v1 ^
  --license-model license-included ^
  --master-username my-master-username ^
  --master-user-password my-master-password ^
  --domain-fqdn my-AD-test.my-AD.mydomain ^
  --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
  --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-AD-test-secret-123456" \ ^
  --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

下列 CLI 命令會修改現有的 RDS for SQL Server 資料庫執行個體，以使用自我管理 Active Directory 網域。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \
  --db-instance-identifier my-DB-instance \
  --domain-fqdn my_AD_domain.my_AD.my_domain \
  --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
  --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-AD-test-secret-123456" \
  --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier my-DBinstance ^
  --domain-fqdn my_AD_domain.my_AD.my_domain ^
  --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
  --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-AD-test-secret-123456" ^
  --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

下列 CLI 命令會從自我管理 Active Directory 網域中移除 RDS for SQL Server 資料庫執行個體。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-DB-instance \  
  --disable-domain
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-DB-instance ^  
  --disable-domain
```

步驟 7：建立 Windows 身分驗證 SQL Server 登入

使用 Amazon RDS 主要使用者憑證來連線至 SQL Server 資料庫執行個體，如同您對任何其他資料庫執行個體所做一般。因為資料庫執行個體已加入自我管理 AD 網域，所以您可以佈建 SQL Server 登入和使用者。您可以從自我管理 AD 網域中的 AD 使用者和群組公用程式執行此操作。您可透過對這些 Windows 登入授予和撤銷的標準 SQL Server 許可來管理資料庫許可。

為了讓自我管理 AD 向 SQL Server 進行身分驗證，對於自我管理 AD 使用者或使用者為其成員之自我管理 Active Directory 群組，SQL Server Windows 登入必須存在。精細定義的存取控制是透過授予和撤銷這些 SQL Server 登入的許可來處理。自我管理 AD 使用者若沒有 SQL Server 登入，或不屬於具有此類登入的自我管理 AD 群組，則無法存取 SQL Server 資料庫執行個體。

需要 ALTER ANY LOGIN 許可，才能建立自我管理 AD SQL Server 登入。如果您尚未使用此許可建立任何登入，請使用 SQL Server 身分驗證，以資料庫執行個體的主要使用者身分連線，並在主要使用者的內容下建立自我管理 AD SQL Server 登入。

您可以執行如下的資料定義語言 (DDL) 命令，為自我管理 AD 使用者或群組建立 SQL Server 登入。

Note

使用 Windows 2000 前版的登入名稱，以格式 *my_AD_domain\my_AD_domain_user* 來指定使用者和群組。您無法使用格式為 *my_AD_domain_user@my_AD_domain* 的使用者原則名稱 (UPN)。

```
USE [master]  
GO  
CREATE LOGIN [my_AD_domain\my_AD_domain_user] FROM WINDOWS WITH DEFAULT_DATABASE =  
  [master], DEFAULT_LANGUAGE = [us_english];
```

GO

如需詳細資訊，請參閱 Microsoft 開發人員網路文件中的 [建立登入 \(Transact-SQL\)](#)。

來自您網域的使用者 (人員和應用程式兩者) 現在可以使用 Windows 身分驗證，從加入自我管理 AD 網域的用戶端機器連線至 RDS for SQL Server Active 執行個體。

在自我管理 Active Directory 網域中管理資料庫執行個體

您可以使用主控台 AWS CLI 或 Amazon RDS API 來管理資料庫執行個體及其與自我管理 AD 網域的關係。例如，您可以將資料庫執行個體移入、移出或移至網域之間。

例如，使用 Amazon RDS API，您可以執行下列動作：

- 若要針對失敗的成員資格重新嘗試自我管理網域加入，請使用 [ModifyDBInstance](#) API 操作，並指定相同的參數集：
 - `--domain-fqdn`
 - `--domain-dns-ips`
 - `--domain-ou`
 - `--domain-auth-secret-arn`
- 若要從自我管理網域中移除資料庫執行個體，請使用 `ModifyDBInstance` API 操作，並針對網域參數指定 `--disable-domain`。
- 若要將資料庫執行個體從某個自我管理網域移至另一個網域，請使用 `ModifyDBInstance` API 操作，並指定新網域的網域參數。
 - `--domain-fqdn`
 - `--domain-dns-ips`
 - `--domain-ou`
 - `--domain-auth-secret-arn`
- 若要列出每個資料庫執行個體的自我管理 AD 網域成員資格，請使用 [DescribeDBInstances](#) API 操作。

了解自我管理 Active Directory 網域成員資格

在您建立或修改資料庫執行個體之後，該執行個體會成為自我管理 AD 網域的成員。主 AWS 控制台會指出資料庫執行個體之自我管理 Active Directory 網域成員資格的狀態。資料庫執行個體的狀態可以是下列其中一個：

- 已加入 – 執行個體是 AD 網域的成員。
- 加入中 – 執行個體正處於成為 AD 網域成員的過程中。
- 加入待定 – 執行個體成員資格待定。
- pending-maintenance-join— AWS 會在下一個排定的維護時段中嘗試讓執行個體成為 AD 網域的成員。
- 移除待定 – 從 AD 網域移除執行個體待定。
- pending-maintenance-removal— AWS 會在下次排定的維護時段中嘗試從 AD 網域移除執行個體。
- 失敗 – 組態問題讓執行個體無法加入 AD 網域。請在重新發出執行個體修改命令之前檢查並修正您的組態。
- 移除中 – 正在從自我管理 AD 網域移除執行個體。

因為網路連線問題，所以成為自我管理 AD 網域成員的請求可能失敗。例如，您可以建立一個資料庫執行個體或修改現有執行個體，並嘗試讓資料庫執行個體失敗，以便成為自我管理 AD 網域的成員。在此情況下，請重新發出命令以建立或修改資料庫執行個體，或修改新建立的執行個體以加入自我管理 AD 網域。

對自我管理 Active Directory 進行疑難排解

以下是您在設定或修改自我管理 AD 時可能會遇到的問題。

錯誤程式碼	描述	常見原因	故障診斷建議
錯誤 2 / 0x2	系統找不到指定的檔案。	使用 <code>-domain-ou</code> 參數指定的組織單位 (OU) 的格式或位置無效。透過 AWS Secrets Manager 指定的網域服務帳戶缺少加入 OU 所需的權限。	檢閱 <code>-domain-ou</code> 參數。確定網域服務帳戶具有 OU 的正確許可。如需詳細資訊，請參閱 設定您的 AD 網域服務帳戶 。
錯誤 5 / 0x5	存取遭拒。	網域服務帳戶的許可設定錯誤，或網域中已存在電腦帳戶。	檢閱網域中的網域服務帳戶許可，並確認網域中沒有重複的 RDS 電腦帳戶。您可以在 RDS for SQL Server 資料庫執行個體上執行 <code>SELECT @@SERVERNAME</code> ，以驗證

錯誤程式碼	描述	常見原因	故障診斷建議
			RDS 電腦帳戶的名稱。如果您使用多可用區域，請嘗試透過容錯移轉重新開機，然後再次驗證 RDS 電腦帳戶。如需詳細資訊，請參閱 重新啟動中的資料庫執行個體 。
錯誤 87 / 0x57	參數不正確。	透過 AWS Secrets Manager 指定的網域服務帳戶沒有正確的權限。使用者設定檔也可能已損毀。	檢閱網域服務帳戶的需求。如需詳細資訊，請參閱 設定您的 AD 網域服務帳戶 。
錯誤 234 / 0xEA	指定的組織單位 (OU) 不存在。	使用 <code>-domain-ou</code> 參數指定的 OU 不存在於自我管理 AD 中。	檢閱 <code>-domain-ou</code> 參數，並確保指定的 OU 存在於您的自我管理 AD 中。
錯誤 1326 / 0x52E	使用者名稱或密碼不正確。	AWS Secrets Manager 中提供的網域服務帳戶認證包含未知的使用者名稱或錯誤的密碼。也可能在您的自我管理 AD 中停用了網域帳戶。	請確定 AWS Secrets Manager 中提供的認證是正確的，而且網域帳戶已在您自我管理的 Active Directory 中啟用。
錯誤 1355 / 0x54B	指定的網域不存在或無法聯絡。	網域已關閉、指定的 DNS IP 集無法連線，或指定的 FQDN 無法連線。	檢閱 <code>-domain-dns-ips</code> 和 <code>-domain-fqdn</code> 參數以確定其正確無誤。檢閱 RDS for SQL Server 資料庫執行個體的網路組態，並確定可連線到您的自我管理 AD。如需詳細資訊，請參閱 設定您的網路連線能力 。

錯誤程式碼	描述	常見原因	故障診斷建議
錯誤	RPC 伺服器無法使用。	連線 AD 網域的 RPC 服務時發生問題。這可能是服務或網路問題。	驗證 RPC 服務是否正在您的網域控制器上執行，以及 TCP 連接埠 135 和 49152-65535 是否可在網域上，從 RDS for SQL Server 資料庫執行個體進行連線。
錯誤 2224 / 0x8B0	使用者帳戶已存在。	嘗試新增至自我管理 AD 的電腦帳戶已存在。	在 RDS for SQL Server 資料庫執行個體上執行 <code>SELECT @@SERVERNAME</code> ，來識別電腦帳戶，然後小心地將其從自我管理 AD 中移除。
錯誤 2242 / 0x8c2	此使用者的密碼已過期。	透過秘密管理員指定之網域服務帳戶的 AWS 密碼已過期。	更新網域服務帳戶的密碼，此帳戶用來將 RDS for SQL Server 資料庫執行個體加入自我管理 AD。

還原 SQL Server 資料庫執行個體，然後將其新增至自我管理 Active Directory 網域

您可以還原資料庫快照集，或為 SQL Server 資料庫執行個體執行 point-in-time 復原 (PITR)，然後將其新增至自我管理的使用中目錄網域。一旦還原了資料庫執行個體，請使用 [步驟 6：建立或修改 SQL Server 資料庫執行個體](#) 中說明的程序來修改執行個體，以將資料庫執行個體新增至自我管理 AD 網域。

使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄

當使用者連線 AWS Managed Microsoft AD 至 SQL Server 資料庫執行個體的 RDS 時，您可以使用視窗驗證來驗證使用者。資料庫執行個體搭配使 AWS Directory Service for Microsoft Active Directory (也稱為 AWS Managed Microsoft AD) 以啟用 Windows 驗證。當使用者向加入信任網域的 SQL Server 資料庫執行個體進行驗證時，身分驗證請求會轉送到您使用 AWS Directory Service 建立的網域目錄。

區域和版本可用性

Amazon RDS 僅 AWS Managed Microsoft AD 支持視窗身份驗證使用。RDS 不支援使用 AD Connector。如需詳細資訊，請參閱下列內容：

- [應用程式相容性政策 AWS Managed Microsoft AD](#)
- [AD Connector 應用程式相容性政策](#)

如需有關版本和區域可用性的詳細資訊，請參閱[透過 RDS for SQL Server 進行 Kerberos 身分驗證](#)。

設定 Windows 身分驗證概觀

Amazon RDS 對 Windows 身分驗證使用混合模式。此方法表示主要使用者 (用來建立您的 SQL Server 資料庫執行個體的名稱和密碼) 使用 SQL 身分驗證。因為主要使用者帳戶是具有特殊權限的登入資料，您應該限制對此帳戶的存取。

若要使用現場部署或自行託管的 Microsoft Active Directory 獲得 Windows 身分驗證，請建立樹系信任。信任可以是單向或雙向。如需使用設定樹系信任的詳細資訊 AWS Directory Service，請參閱《AWS Directory Service 管理指南》中的[建立信任關係的時機](#)。

若要為 SQL Server 資料庫執行個體設定 Windows 身分驗證，請執行下列步驟 (如 [為 SQL Server 資料庫執行個體設定 Windows 身分驗證](#) 所詳述)：

1. 使用 AWS Managed Microsoft AD AWS Management Console 或 AWS Directory Service API 建立 AWS Managed Microsoft AD 目錄。
2. 如果您使用 AWS CLI 或 Amazon RDS API 建立 SQL 伺服器資料庫執行個體，請建立 AWS Identity and Access Management (IAM) 角色。此角色使用受管 IAM 政策 AmazonRDSDirectoryServiceAccess，而且 Amazon RDS 允許對目錄進行呼叫。如果您使用主控台來建立您的 SQL Server 資料庫執行個體，AWS 會為您建立 IAM 角色。

若要讓角色允許存取，必須在您的 AWS 帳戶的 [AWS 地區] 中啟動 AWS Security Token Service (AWS STS) 端點。AWS STS 依預設，所有 AWS 區域中的端點都處於作用中狀態，您可以使用它們，而無需執行任何進一步 如需詳細資訊，請參閱《IAM 使用者指南》中的[管理 AWS 區域中的 AWS STS](#)。

3. 使用 Microsoft 活動目錄工具在 AWS Managed Microsoft AD 目錄中創建和配置用戶和組。如需在 Microsoft Active Directory 建立使用者的詳細資訊，請參閱 AWS Directory Service 管理指南中的[管理 AWS Managed Microsoft AD 中的使用者和群組](#)。
4. 如果您打算在不同 VPC 中尋找目錄和資料庫執行個體，請啟用跨 VPC 流量。
5. 使用 Amazon RDS 從主控台或 Amazon RDS API 建立新的 SQL 伺服器資料庫執行個體。AWS CLI 在建立請求中，您可以提供建立目錄時產生的網域識別符 ("d-*" identifier) 和您建立之角色的名稱。您也可以將現有 SQL Server 資料庫執行個體修改為使用 Windows 身分驗證，方法是設定資料庫執行個體的網域和 IAM 角色參數。
6. 如同對其他資料庫執行個體一般，使用 Amazon RDS 主要使用者登入資料來連接至 SQL Server 資料庫執行個體。由於資料庫執行個體已加入 AWS Managed Microsoft AD 網域，因此您可以從其網域中的 Active Directory 使用者和群組佈建 SQL Server 登入和使用者。(這些稱為 SQL Server "Windows" 登入。) 您可透過對這些 Windows 登入授予和撤銷的標準 SQL Server 許可來管理資料庫許可。

建立 Kerberos 身分驗證的端點

Kerberos 式身分驗證需要端點是客戶指定的主機名稱、句點，然後接著完整網域名稱 (FQDN)。例如，下列是您會對 Kerberos 式身分驗證使用的端點範例。在此範例中，SQL Server 資料庫執行個體主機名稱是 ad-test，而網域名稱是 corp-ad.company.com。

```
ad-test.corp-ad.company.com
```

如果您要確定您的連線使用的是 Kerberos，執行下列查詢：

```
SELECT net_transport, auth_scheme
FROM sys.dm_exec_connections
WHERE session_id = @@SPID;
```

為 SQL Server 資料庫執行個體設定 Windows 身分驗證

您可以使用 AWS Directory Service for Microsoft Active Directory(也稱為 AWS Managed Microsoft AD) 為 SQL Server 資料庫執行個體設定視窗驗證。若要設定 Windows 身分驗證，請執行下列步驟。

步驟 1：使用建立目錄 AWS Directory Service for Microsoft Active Directory

AWS Directory Service 在 AWS 雲中創建一個完全受管理的 Microsoft 活動目錄。當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會代表您建立兩個網域控制站和網域名稱服務 (DNS) 伺服器。目錄伺服器是在兩個子網路中建立，而這兩個子網路是在 VPC 內的兩個不同可用區域中。此備援可協助確保即使失敗，仍能存取您的目錄。

建立目 AWS Managed Microsoft AD 錄時，請代表您 AWS Directory Service 執行下列工作：

- 設定 VPC 內的 Microsoft Active Directory。
- 建立含有使用者名稱 Admin 與指定密碼的目錄管理員帳戶。您可以使用此帳戶來管理目錄。

Note

請務必儲存此密碼。AWS Directory Service 不會儲存此密碼，也無法擷取或重設密碼。

- 建立目錄控制器的安全群組。

啟動時 AWS Directory Service for Microsoft Active Directory，AWS 會建立包含所有目錄物件的組織單位 (OU)。此 OU 有您在建立目錄時所輸入的 NetBIOS 名稱，位於根網域中。網域根目錄擁有及管理 AWS。

隨著您的 AWS Managed Microsoft AD 目錄建立的管理帳戶具有您的 OU 最常見管理活動的許可：

- 建立更新或刪除使用者、群組和電腦。
- 新增資源 (例如檔案或列印伺服器) 至您的網域，然後對您 OU 中的使用者和群組指派這些資源的許可。
- 建立額外的 OU 和容器。
- 委派授權。
- 建立及連結群組政策。
- 從 Active Directory 資源回收筒還原已刪除的物件。
- 在活動目錄 Web 服務上運行 AD 和 DNS 視窗 PowerShell 模塊。

管理員帳戶也有權執行下列全網域活動：

- 管理 DNS 組態 (新增、移除或更新記錄、區域和轉寄站)。
- 檢視 DNS 事件日誌。

- 檢視安全事件日誌。

若要建立目錄 AWS Managed Microsoft AD

1. 在 [AWS Directory Service 主控台](#) 中，依序選擇 Directories (目錄) 和 Set up directory (設定目錄)。
2. 選擇 AWS Managed Microsoft AD。這是目前支援與 Amazon RDS 搭配使用的唯一選項。
3. 選擇 Next (下一步)。
4. 在 Enter directory information (輸入目錄資訊) 頁面上，提供下列資訊：

版本

選擇滿足您需求的版本。

目錄 DNS 名稱

目錄的完全合格名稱，例如 corp.example.com。SQL Server 不支援超過 47 個字元的名稱。

目錄 NetBIOS 名稱

目錄的簡短名稱，例如：CORP。

目錄描述

選擇填寫其他目錄說明。

管理員密碼

目錄管理員的密碼。目錄建立程序會建立含有使用者名稱 Admin 與這組密碼的管理員帳戶。

目錄管理員密碼不得包含 admin 一字。密碼區分大小寫，長度須為 8 至 64 個字元。至少須有一位字元屬於以下四種類型中的三類：

- 小寫字母 (a-z)
- 大寫字母 (A-Z)
- 數字 (0-9)
- 非英數字元 (~!@#\$%^&* _+=`|\(){}[]:;'"<>.,?/)

Confirm password (確認密碼)

重新輸入管理員密碼。

5. 選擇 Next (下一步)。

- 在 Choose VPC and subnets (選擇 VPC 和子網路) 頁面上，提供下列資訊：

VPC

選擇目錄的 VPC。

Note

您可以在不同 VPC 中尋找目錄和資料庫執行個體，但如果您這麼做，請務必啟用跨 VPC 流量。如需更多詳細資訊，請參閱 [步驟 4：啟用目錄和資料庫執行個體之間的跨 VPC 流量](#)。

子網路

選擇目錄伺服器的子網路。這兩個子網路必須位於不同的可用區域。

- 選擇 Next (下一步)。
- 檢閱目錄資訊。如果需要變更，請選擇 Previous (上一步)。若資訊無誤，請選擇 Create directory (建立目錄)。

Review & create

Review

Directory type Microsoft AD	VPC vpc-8b6b78e9 ()
Directory DNS name corp.example.com	Subnets subnet-75128d10 (, us-east-1a) subnet-f51665dd (, us-east-1b)
Directory NetBIOS name CORP	
Directory description My directory	

Pricing

Edition Standard	Free trial eligible Learn more 30-day limited trial
~USD () *	
* Includes two domain controllers, USD ()/mo for each additional domain controller.	

Cancel Previous **Create directory**

建立目錄需要幾分鐘的時間。成功建立時，Status (狀態) 值會變更為 Active (作用中)。

若要查看目錄的資訊，請選擇目錄清單中的目錄 ID。請記下 Directory ID (目錄 ID)。建立或修改 SQL Server 資料庫執行個體時需要此值。

Directory Service > Directories > d-90670a8d36

Directory details

[Reset user password](#)

Directory type Microsoft AD	VPC vpc-6594f31c	Status Active
Edition Standard	Subnets subnet-7d36a227 subnet-a2ab49c6	Last updated Tuesday, January 7, 2020
Directory ID d-90670a8d36	Availability zones us-east-1c, us-east-1d	Launch time Tuesday, January 7, 2020
Directory DNS name corp.example.com	DNS address 	
Directory NetBIOS name CORP		
Description - Edit My directory		

[Application management](#) | [Scale & share](#) | [Networking & security](#) | [Maintenance](#)

步驟 2：建立供 Amazon RDS 使用的 IAM 角色

如果您使用主控台來建立您的 SQL Server 資料庫執行個體，則可以略過此步驟。如果您使用 `&CLI;` 或 `&RDS;` API 來建立您的 SQL Server 資料庫執行個體，您必須建立使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess` 的 IAM 角色。此角色可讓 Amazon RDS AWS Directory Service 為您撥打電話。

如果您使用自訂原則來加入網域，而不是使用 AWS-managed `AmazonRDSDirectoryServiceAccess` 政策，請確定您允許

該 `ds:GetAuthorizedApplicationDetails` 動作。由於 AWS Directory Service API 發生變更，此要求自 2019 年 7 月起生效。

以下 IAM 政策 (`AmazonRDSDirectoryServiceAccess`) 會提供對 AWS Directory Service 的存取。

Example 提供存取權的 IAM 政策 AWS Directory Service

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

建議您在資源型信任關係中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰，將服務的許可限定於特定資來源。這是防止[混淆代理人問題](#)最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在信任關係中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 Amazon 資源名稱 (ARN)。為使用 Windows Authentication，請務必包含資料庫執行個體，如下列範例所示。

Example 與 Windows Authentication 全域條件內容索引鍵的信任關係

```
{
  "Version": "2012-10-17",
```



```
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceArn": [
          "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
        ]
      }
    }
  }
]
```

使用此 IAM 政策和信任關係建立 IAM 角色。如需建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的[建立客戶受管原則](#)。

步驟 3：建立和設定使用者和群組

您可以使用 Active Directory 使用者和運算集區來建立使用者及群組。這個工具是 Active Directory Domain Services 和 Active Directory 輕量型目錄服務工具之一。使用者代表具有目錄存取權的個人或實體。群組非常適合對使用者群組授予或拒絕權限，而無需將這些權限逐一套用到各個使用者。

若要在 AWS Directory Service 目錄中建立使用者和群組，您必須連線至屬於 AWS Directory Service 目錄成員的 Windows EC2 執行個體。您也必須以具有建立使用者及群組之許可的使用者身分來登入。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[新增使用者和群組 \(Simple AD 和 AWS Managed Microsoft AD\)](#)。

步驟 4：啟用目錄和資料庫執行個體之間的跨 VPC 流量

如果您打算在相同 VPC 中尋找目錄和資料庫執行個體，請略過本步驟，並移至[步驟 5：建立或修改 SQL Server 資料庫執行個體](#)。

如果您打算在不同 VPC 中尋找目錄和資料庫執行個體，請使用 VPC 互連或 [AWS Transit Gateway](#) 來設定跨 VPC 流量。

下列程序會使用 VPC 互連來啟用 VPC 之間的流量。請遵循《Amazon Virtual Private Cloud 互連指南》中[什麼是 VPC 互連？](#)的指示。

使用 VPC 互連以啟用跨 VPC 流量

1. 設定適當的 VPC 路由規則，以確保網路流量可以雙向對流。
2. 確保資料庫執行個體安全群組可以從目錄的安全群組接收傳入流量。
3. 確保沒有網路存取控制清單 (ACL) 規則來封鎖流量。

如果目錄擁有不同的 AWS 帳戶，您必須共用該目錄。

在 AWS 帳戶之間共用目錄

1. 按照 AWS Directory Service 《管理指南》中的[教學課程：共用目錄以進行無縫 EC2 網域加入中的說明，開始與將在其中建立資料庫執行個體的 AWS 帳戶共用 AWS Managed Microsoft AD 目錄](#)。
2. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台，並確保網域具有 SHARED 狀態，然後再繼續操作。
3. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台時，請記下目錄 ID 值。您可以使用此目錄 ID，將資料庫執行個體加入網域。

步驟 5：建立或修改 SQL Server 資料庫執行個體

建立或修改要搭配您目錄使用的 SQL Server 資料庫執行個體。您可以使用主控台、CLI 或 RDS API，將資料庫執行個體與目錄建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) CLI 命令，或 [CreateDBInstance](#) RDS API 操作，建立新的 SQL Server 資料庫執行個體。

如需說明，請參閱「[建立 Amazon RDS 資料庫執行個體](#)」。

- 使用主控台、[modify-db-instance](#) CLI 命令，或 [ModifyDBInstance](#) RDS API 操作，修改現有的 SQL Server 資料庫執行個體。

如需說明，請參閱「[修改 Amazon RDS 資料庫執行個體](#)」。

- [使用主控台、從資料庫快照恢復資料庫執行個體從 DB 快照 CLI 命令或還原資料庫快照 RDS API 作業，從資料庫快照還原 SQL Server 資料庫執行個體。InstanceFrom](#)

如需說明，請參閱[從資料庫快照還原](#)。

- [將 SQL Server 資料庫執行個體還原至 point-in-time 使用主控台、還原 DB 執行個體至時間點 CLI 命令，或恢復時間 RDS API 作業。InstanceTo PointIn](#)

如需說明，請參閱「[將資料庫執行個體還原至指定的時間](#)」。

Windows 身分驗證只支援 VPC 中的 Server 資料庫執行個體。

若要讓資料庫執行個體使用您建立的網域目錄，需要下列項目：

- 對於 Directory (目錄)，您必須選擇在建立目錄時產生的網域識別碼 (d-*ID*)。
- 確定 VPC 安全群組具有可讓資料庫執行個體與目錄通訊的輸出規則。

Microsoft SQL Server Windows Authentication

Choose a directory in which you want to allow authorized domain users to authenticate with this SQL Server instance using Windows Authentication.

Directory

corp.example.com (d-)

[Create a new directory](#)

By choosing a directory and continuing with database instance creation you authorize Amazon RDS to create the IAM role necessary for using Windows Authentication

使用時 AWS CLI，資料庫執行個體需要下列參數才能使用您建立的目錄：

- 針對 `--domain` 參數，使用您建立目錄時產生的網域識別符 (d-*ID*)。
- 針對 `--domain-iam-role-name` 參數，使用您建立的規則，其會使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess`。

例如，下列 CLI 命令會修改資料庫執行個體來使用目錄。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --domain d-ID \  
  --domain-iam-role-name role-name
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --domain d-ID ^
  --domain-iam-role-name role-name
```

Important

如果您修改資料庫執行個體，以啟用 Kerberos 身分驗證，請在進行變更後重新啟動資料庫執行個體。

步驟 6：建立 Windows 身分驗證 SQL Server 登入

如同對其他資料庫執行個體一般，使用 Amazon RDS 主要使用者登入資料來連接至 SQL Server 資料庫執行個體。由於資料庫執行個體已加入 AWS Managed Microsoft AD 網域，因此您可以佈建 SQL Server 登入和使用者。您可以從網域中的 Active Directory 使用者和群組執行這項操作。您可透過對這些 Windows 登入授予和撤銷的標準 SQL Server 許可來管理資料庫許可。

若要讓 Active Directory 使用者向 SQL Server 進行身分驗證，必須存在使用者或使用者為其成員之群組的 SQL Server Windows 登入。精細定義的存取控制是透過授予和撤銷這些 SQL Server 登入的許可來處理。沒有 SQL Server 登入或屬於具有此類登入的群組的使用者無法存取 SQL Server 資料庫執行個體。

需要 ALTER ANY LOGIN 許可，才能建立 Active Directory SQL Server 登入。如果您尚未使用此許可建立任何登入，請使用 SQL Server 身分驗證，以資料庫執行個體的主要使用者身分連線。

執行資料定義語言 (DDL) 命令來為 Active Directory 使用者或群組建立 SQL Server 登入，如下列範例。

Note

使用 Windows 2000 前版的登入名稱，以格式 *domainName\login_name* 來指定使用者和群組。您無法使用格式為 *login_name@DomainName* 的使用者原則名稱 (UPN)。您只能使用 T-SQL 陳述式，在適用於 SQL 伺服器的 RDS 執行個體上建立視窗驗證登入。您無法使用 SQL 伺服器管理工作室來建立 Windows 驗證登入。

```
USE [master]
```

```
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
  DEFAULT_LANGUAGE = [us_english];
GO
```

如需詳細資訊，請參閱 Microsoft 開發人員網路文件中的 [建立登入 \(Transact-SQL\)](#)。

來自您網域的使用者 (人員和應用程式兩者)，現在可以使用 Windows 身分驗證從加入網域的用戶端機器連接至 RDS for SQL Server 執行個體。

管理網域中的資料庫執行個體

您可以使用主控 AWS CLI 台或 Amazon RDS API 來管理資料庫執行個體及其與網域的關係。例如，您可以將資料庫執行個體移入、移出或移至網域之間。

例如，使用 Amazon RDS API，您可以執行下列動作：

- 若要對失敗的成員資格重新嘗試加入網域，請使用 [ModifyDBInstance API](#) 操作，並指定目前成員資格的目錄 ID。
- 如要更新成員資格的 IAM 角色名稱，請使用 [ModifyDBInstance API](#) 操作並指定目前成員資格的目錄 ID，以及新的 IAM 角色。
- 若要從網域移除資料庫執行個體，請使用 [ModifyDBInstance API](#) 操作，並指定 none 做為網域參數。
- 如要在網域之間移動資料庫執行個體，請使用 [ModifyDBInstance API](#) 操作，並指定新網域的網域識別符做為網域參數。
- 若要列出每個資料庫執行個體的成員資格，請使用 [DescribeDBInstances](#) API 操作。

了解網域成員資格

在您建立或修改資料庫執行個體之後，執行個體會成為網域的成員。主 AWS 控制台會指出資料庫執行個體的網域成員資格狀態。資料庫執行個體的状态可以是下列其中一個：

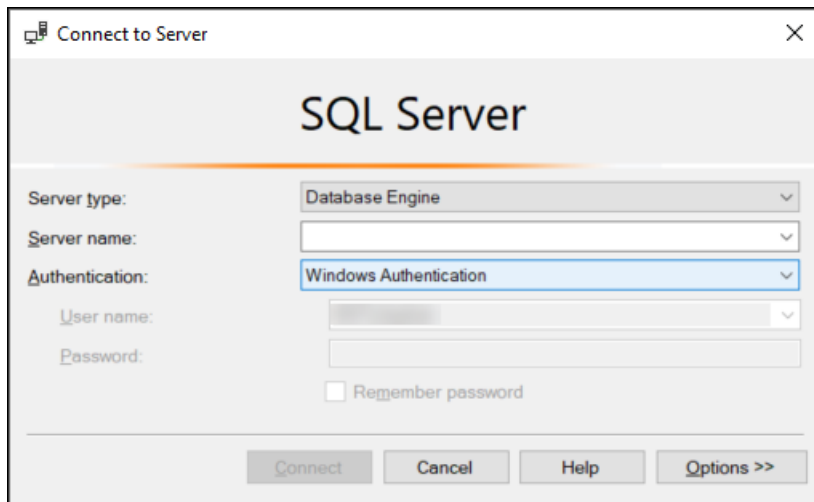
- 已加入 – 執行個體是網域的成員。
- 加入中 – 執行個體正處於成為網域成員的程序中。
- 加入待定 – 執行個體成員資格待定。
- 暫停維護加入 — AWS 會在下一個排定的維護時段中嘗試讓執行個體成為網域的成員。
- 移除待定 – 從網域移除執行個體待定。

- 暫停維護移除 — AWS 會在下一個排定的維護時段中嘗試從網域移除執行個體。
- 失敗 – 組態問題讓執行個體無法加入網域。請在重新發出執行個體修改命令之前檢查並修正您的組態。
- 移除中 – 正在從網域移除執行個體。

因為網路連線問題或 IAM 角色不正確，成為網域成員的請求可能失敗。例如，您可以建立一個資料庫執行個體或修改現有執行個體，並嘗試讓資料庫執行個體失敗，以便成為網域的成員。在此情況下，請重新發出命令以建立或修改資料庫執行個體，或修改新建立的執行個體以加入網域。

使用 Windows 身分驗證連線至 SQL Server

若要使用 Windows 身分驗證連線至 SQL Server，您必須以網域使用者的身分登入到加入網域的電腦。啟動 SQL Server Management Studio 之後，選擇 Windows 身分驗證做為身分驗證類型，如下所示。



還原 SQL Server 資料庫執行個體，並將其新增至網域

您可以還原資料庫快照集，或為 SQL Server 資料庫執行個體執行 point-in-time 復原 (PITR)，然後將其新增至網域。一旦還原資料庫執行個體，請使用 [步驟 5：建立或修改 SQL Server 資料庫執行個體](#) 中的說明程序來修改執行個體，以將資料庫執行個體新增至網域。

將應用程式更新為使用新的 SSL/TLS 憑證來連線至 Microsoft SQL Server 資料庫執行個體

自 2023 年 1 月 13 日起，Amazon RDS 已發佈新的憑證認證機構 (CA) 憑證，使用 Secure Socket Layer 或 Transport Layer Security (SSL/TLS) 來連線至 RDS 資料庫執行個體。接下來，您可以找到更新應用程式使用新憑證的相關資訊。

本主題可協助您判斷任何用戶端應用程式是否使用 SSL/TLS 連線至您的資料庫執行個體。若是如此，您可以進一步檢查那些應用程式是否需要驗證憑證才能連線。

Note

有些應用程式設定為只有在成功驗證伺服器上的憑證時，才能連線至 SQL Server 資料庫執行個體。

對於這些應用程式，您必須更新用戶端應用程式信任存放區來包含新的 CA 憑證。

更新用戶端應用程式信任存放區中的 CA 憑證之後，您就可以在資料庫執行個體輪換憑證。強烈建議先在開發或預備環境中測試這些步驟，再於生產環境中實作。

如需憑證輪換的詳細資訊，請參閱[輪換您的 SSL/TLS 憑證](#)。如需下載憑證的詳細資訊，請參閱。如需對 Microsoft SQL Server 資料庫執行個體使用 SSL/TLS 的資訊，請參閱[對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)。

主題

- [判斷任何應用程式是否使用 SSL 連線至 Microsoft SQL Server 資料庫執行個體](#)
- [判斷用戶端是否需要驗證憑證才能連線](#)
- [更新應用程式信任存放區](#)

判斷任何應用程式是否使用 SSL 連線至 Microsoft SQL Server 資料庫執行個體

在資料庫執行個體組態中檢查 `rds.force_ssl` 參數的值。依預設，`rds.force_ssl` 參數會設為 0 (關閉)。如果 `rds.force_ssl` 參數設為 1 (開啟)，則用戶端需要使用 SSL/TLS 進行連線。如需參數群組的詳細資訊，請參閱[使用參數群組](#)。

執行下列查詢，以取得資料庫執行個體的所有已開啟連線的目前加密選項。如果連線已加密，ENCRYPT_OPTION 欄會傳回 TRUE。

```
select SESSION_ID,  
       ENCRYPT_OPTION,  
       NET_TRANSPORT,  
       AUTH_SCHEME  
from SYS.DM_EXEC_CONNECTIONS
```

此查詢只顯示目前連線。不會指出過去曾經連線又中斷連線的應用程式是否使用 SSL。

判斷用戶端是否需要驗證憑證才能連線

您可以檢查不同類型的用戶端是否需要驗證憑證才能連線。

Note

如果您使用的連接器未列出，請參閱特定連接器的文件，以取得該連接器如何強制使用加密連線的相關資訊。如需詳細資訊，請參閱 Microsoft SQL Server 文件中的 [Microsoft SQL 資料庫的連線模組](#)。

SQL Server Management Studio

檢查是否強制 SQL Server Management Studio 連線使用加密：

1. 啟動 SQL Server Management Studio。
2. 在連線到伺服器中，輸入伺服器資訊、登入使用者名稱和密碼。
3. 選擇 Options (選項)。
4. 在連線頁面上，檢查是否已選取加密連線。

如需 SQL Server Management Studio 的詳細資訊，請參閱[使用 SQL Server Management Studio](#)。

Sqlcmd

以下 sqlcmd 用戶端範例會示範如何檢查指令碼的 SQL Server 連線，以判斷成功的連線是否需要有效憑證。如需詳細資訊，請參閱 Microsoft SQL Server 文件中的[使用 sqlcmd 進行連線](#)。

使用 sqlcmd 用戶端時，如果您使用 -N 命令引數來加密連線，則需要以伺服器憑證來驗證 SSL 連線，如下列範例所示。

```
$ sqlcmd -N -S dbinstance.rds.amazon.com -d ExampleDB
```

Note

如果是以 -C 選項叫用 sqlcmd，即使伺服器憑證不符合用戶端信任存放區，還是會受到信任。

ADO.NET

在下列範例中，應用程式使用 SSL 進行連線，且必須驗證伺服器憑證。

```
using SQLC = Microsoft.Data.SqlClient;

...

static public void Main()
{
    using (var connection = new SQLC.SqlConnection(
        "Server=tcp:dbinstance.rds.amazon.com;" +
        "Database=ExampleDB;User ID=LOGIN_NAME;" +
        "Password=YOUR_PASSWORD;" +
        "Encrypt=True;TrustServerCertificate=False;"
    ))
    {
        connection.Open();
        ...
    }
}
```

Java

在下列範例中，應用程式使用 SSL 進行連線，且必須驗證伺服器憑證。

```
String connectionString =  
    "jdbc:sqlserver://dbinstance.rds.amazonaws.com;" +  
    "databaseName=ExampleDB;integratedSecurity=true;" +  
    "encrypt=true;trustServerCertificate=false";
```

對於使用 JDBC 連接的用戶端，若要啟用 SSL 加密，您可能需要將 Amazon RDS 憑證新增至 Java CA 憑證存放區。如需說明，請參閱 Microsoft SQL Server 文件中的[設定用戶端的加密](#)。您也可以將 `trustStore=`*path-to-certificate-trust-store-file* 附加至連線字串，以直接提供信任的 CA 憑證檔案名稱。

Note

如果您在連線字串中使用 `TrustServerCertificate=true` (其對等設定)，連線程序會略過信任鏈驗證。在此情況下，即使無法驗證憑證，應用程式還是會連線。使用 `TrustServerCertificate=false` 可強制憑證驗證，也是最佳實務。

更新應用程式信任存放區

您可以為使用 Microsoft SQL Server 的應用程式更新信任存放區。如需說明，請參閱[加密特定連線](#)。另外，請參閱 Microsoft SQL Server 文件中的[設定用戶端的加密](#)。

如果您使用的作業系統不是 Microsoft Windows，請參閱有關 SSL/TLS 實作的軟體分發文件，以取得新增根 CA 憑證的相關資訊。例如，OpenSSL 和 GnuTLS 是常用的選項。使用實作方法將信任新增至 RDS 根 CA 憑證。Microsoft 提供在某些系統上設定憑證的相關指示。

如需下載根憑證的資訊，請參閱。

如需匯入憑證的範例指令碼，請參閱[將憑證匯入信任存放區的範例指令碼](#)。

Note

更新信任存放區時，除了新增憑證，您還可以保留舊憑證。

升級 Microsoft SQL Server 資料庫引擎

當 Amazon RDS 支援新版本的資料庫引擎時，您可以將資料庫執行個體升級為新版本。SQL Server 資料庫執行個體有兩種升級：主要版本升級和次要版本升級。

主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。因此，您必須手動執行資料庫執行個體的主要版本升級。您可以修改資料庫執行個體，啟動主要版本升級。然而，在您執行主要的版本升級之前，建議您遵循[測試升級](#)中描述的步驟來測試升級。

反之，次要版本升級只包含與現有應用程式回溯相容的變更。您可以修改資料庫執行個體，手動啟動次要版本升級。

在下列範例中，CLI 命令會傳回 AutoUpgrade 顯示為 true 的回應，指出升級為自動。

```
...  
"ValidUpgradeTarget": [  
  {  
    "Engine": "sqlserver-se",  
    "EngineVersion": "14.00.3281.6.v1",  
    "Description": "SQL Server 2017 14.00.3281.6.v1",  
    "AutoUpgrade": true,  
    "IsMajorVersionUpgrade": false  
  }  
]  
...
```

如需執行升級的詳細資訊，請參閱[升級 SQL Server 資料庫執行個體](#)。如需 Amazon RDS 上提供的 SQL Server 版本相關資訊，請參閱 [Amazon RDS for Microsoft SQL Server](#)。

主題

- [升級概觀](#)
- [主要版本升級](#)
- [多可用區域和記憶體內最佳化的考量](#)
- [僅供讀取複本考量](#)
- [選項群組考量](#)
- [參數群組考量](#)
- [測試升級](#)
- [升級 SQL Server 資料庫執行個體](#)

- [在支援結束之前升級已取代的資料庫執行個體](#)

升級概觀

Amazon RDS 在升級程序中會建立兩個資料庫快照。第一個資料庫快照是完成任何升級變更之前的資料庫執行個體。升級完成後會建立第二個資料庫快照。

Note

只在您將資料庫執行個體的備份保留期設為大於 0 的數字時，Amazon RDS 才會建立資料庫快照。若要變更備份保留期，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

升級完成之後，您就無法回復至前一版的資料庫引擎。若您想要恢復前一版本，請從升級前建立的資料庫快照進行還原，以建立新的資料庫執行個體。

在 SQL Server 的次要或主要版本升級期間，Free Storage Space (可用儲存空間) 和 Disk Queue Depth (磁碟佇列深度) 指標會顯示 -1。升級完成後，兩個指標都會恢復正常。

主要版本升級

Amazon RDS 目前支援透過以下的主要版本升級升級為 Microsoft SQL Server 資料庫執行個體。

除了 SQL Server 2008 之外，您可以將任意版本的現有資料庫執行個體升級至 SQL Server 2017 或 2019。若要升級 SQL Server 2008 的資料庫執行個體，請先將其升級為其他版本。

目前版本	支援的升級版本
SQL Server 2019	SQL 伺服器
SQL Server 2017	SQL 伺服器 SQL Server 2019
SQL Server 2016	SQL 伺服器 SQL Server 2019 SQL Server 2017

目前版本	支援的升級版本
SQL Server 2014	SQL 伺服器 SQL Server 2019 SQL Server 2017 SQL Server 2016
SQL Server 2012 (支援終止)	SQL 伺服器 SQL Server 2019 SQL Server 2017 SQL Server 2016 SQL Server 2014
SQL Server 2008 R2 (支援終止)	SQL Server 2016 SQL Server 2014 SQL Server 2012

您可以使用 AWS CLI 查詢 (例如下列範例) 來尋找特定資料庫引擎版本的可用升級。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \
  --engine sqlserver-se \
  --engine-version 14.00.3281.6.v1 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
  --output table
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^
```

```
--engine sqlserver-se ^
--engine-version 14.00.3281.6.v1 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
--output table
```

輸出顯示您可以將 14.00.3281.6 版本升級至最新可用的 SQL Server 2017 或 2019 版本。

```
-----
|DescribeDBEngineVersions|
+-----+
|      EngineVersion      |
+-----+
| 14.00.3294.2.v1         |
| 14.00.3356.20.v1        |
| 14.00.3381.3.v1         |
| 14.00.3401.7.v1         |
| 14.00.3421.10.v1        |
| 14.00.3451.2.v1         |
| 15.00.4043.16.v1        |
| 15.00.4073.23.v1        |
| 15.00.4153.1.v1         |
| 15.00.4198.2.v1         |
| 15.00.4236.7.v1         |
+-----+
```

資料庫相容性等級

您可以使用 Microsoft SQL Server 資料庫相容性等級來調整某些資料庫行為，以模仿舊版的 SQL Server。如需詳細資訊，請參閱 Microsoft 文件中的[相容性等級](#)。

升級資料庫執行個體時，所有現有的資料庫都會保持其原來的相容性等級。例如，如果從 SQL Server 2014 升級到 SQL Server 2016，則所有現有資料庫的相容性等級均為 120。升級後建立的任何新資料庫都具有相容性等級 130。

您可以使用 ALTER DATABASE 命令來變更資料庫的相容性等級。例如，若要將名為 customeracct 的資料庫變更為與 SQL Server 2014 相容，可使用以下命令：

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 120
```

多可用區域和記憶體內最佳化的考量

Amazon RDS 會透過使用 SQL Server Database Mirroring (DBM) 或 Always On 可用性群組 (AG)，支援在執行 Microsoft SQL Server 的資料庫執行個體上進行異地同步備份部署。如需更多詳細資訊，請參閱 [Amazon RDS for Microsoft SQL Server 的異地同步備份部署](#)。

如果資料庫執行個體位於異地同步備份部署中，則主要和待命執行個體都會升級。Amazon RDS 會進行滾動式升級。只有在容錯轉移期間才會停機。

SQL Server 2014 到 2019 Enterprise Edition 支援記憶體內最佳化。

僅供讀取複本考量

在資料庫版本升級期間，Amazon RDS 也會升級所有僅供讀取複本以及主要資料庫執行個體。Amazon RDS 不支援個別僅供讀取複本上的資料庫版本升級。如需僅供讀取複本的詳細資訊，請參閱 [使用 Amazon RDS 中的 Microsoft SQL Server 的僅供讀取複本](#)。

升級主要資料庫執行個體的資料庫版本時，也會自動升級所有僅供讀取複本。Amazon RDS 在升級主要資料庫執行個體之前，會先同時升級所有僅供讀取複本。在主要資料庫執行個體上的資料庫版本升級完成之前，僅供讀取複本可能無法使用。

選項群組考量

如果您的資料庫執行個體使用自訂資料庫選項群組，在某些狀況下，Amazon RDS 無法自動為資料庫執行個體指派新的選項群組。例如，升級至新的主要版本時，您必須指定新的選項群組。建議您建立新的選項群組，並在其中新增和現有自訂選項群組中相同的選項。

如需詳細資訊，請參閱「[建立選項群組](#)」或「[刪除選項群組](#)」。

參數群組考量

若您的資料庫執行個體使用自訂的資料庫參數群組：

- Amazon RDS 會在升級後自動重新啟動資料庫執行個體。
- 在某些狀況下，RDS 無法將新的參數群組自動指派至您的資料庫執行個體。

例如，升級至新的主要版本時，您必須指定新的參數群組。建議您建立新的參數群組，並在其中設定和現有自訂參數群組中相同的參數。

如需詳細資訊，請參閱「[建立資料庫參數群組](#)」或「[複製資料庫參數群組](#)」。

測試升級

對資料庫執行個體進行主要版本升級之前，請完整測試資料庫，以及會存取該資料庫的所有應用程式，了解是否與新版本相容。建議您採用下列程序。

測試主要版本升級

1. 針對新版本的資料庫引擎，檢閱 Microsoft 文件中的[升級 SQL Server](#)，了解是否有可能影響資料庫或應用程式的相容性問題。
2. 如果您的資料庫執行個體使用自訂選項群組，請建立與升級後新版本相容的新選項群組。如需更多詳細資訊，請參閱[選項群組考量](#)。
3. 如果您的資料庫執行個體使用自訂參數群組，請建立與升級後新版本相容的新參數群組。如需更多詳細資訊，請參閱[參數群組考量](#)。
4. 針對要升級的資料庫執行個體建立資料庫快照。如需詳細資訊，請參閱[為單一可用區資料庫執行個體建立資料庫快照](#)。
5. 還原資料庫快照，以建立新的測試資料庫執行個體。如需更多詳細資訊，請參閱[從資料庫快照還原](#)。
6. 使用下列其中一種方法，修改這個新的測試資料庫執行個體，以升級到新版本：
 - [主控台](#)
 - [AWS CLI](#)
 - [RDS API](#)
7. 評估升級的執行個體所使用的儲存體，以決定升級是否需要更多儲存體。
8. 對升級的資料庫執行個體儘可能進行越多次品質保證測試，以確保資料庫和應用程式搭配新版本可以正常運作。實作任何必要的新測試，以評估您在步驟 1 發現的任何相容性問題所帶來的影響。測試所有預存程序和函數。將應用程式的測試版本指向升級的資料庫執行個體。
9. 如果通過所有測試，就可以對生產資料庫執行個體進行升級。在確認一切都沒問題之前，建議您在資料庫執行個體上不要允許寫入操作。

升級 SQL Server 資料庫執行個體

如需有關手動或自動升級 SQL Server 資料庫執行個體的資訊，請參閱以下內容：

- [升級資料庫執行個體引擎版本](#)
- [在 Amazon RDS for SQL Server 上將 SQL Server 2008 R2 升級到 SQL Server 2016 的最佳實務](#)

⚠ Important

如果您有任何使用加密的快照 AWS KMS，建議您在支援結束前啟動升級。

在支援結束之前升級已取代的資料庫執行個體

主要版本已取代後，您無法將其安裝在新資料庫執行個體上。RDS 將嘗試自動升級所有現有資料庫執行個體。

如果您需要還原已淘汰的資料庫執行個體，可以執行 point-in-time 復原 (PITR) 或還原快照。這可讓您暫時存取使用已廢除版本的資料庫執行個體。但是，在主要版本完全作廢後，這些資料庫執行個體也將同時自動升級到支援的版本。

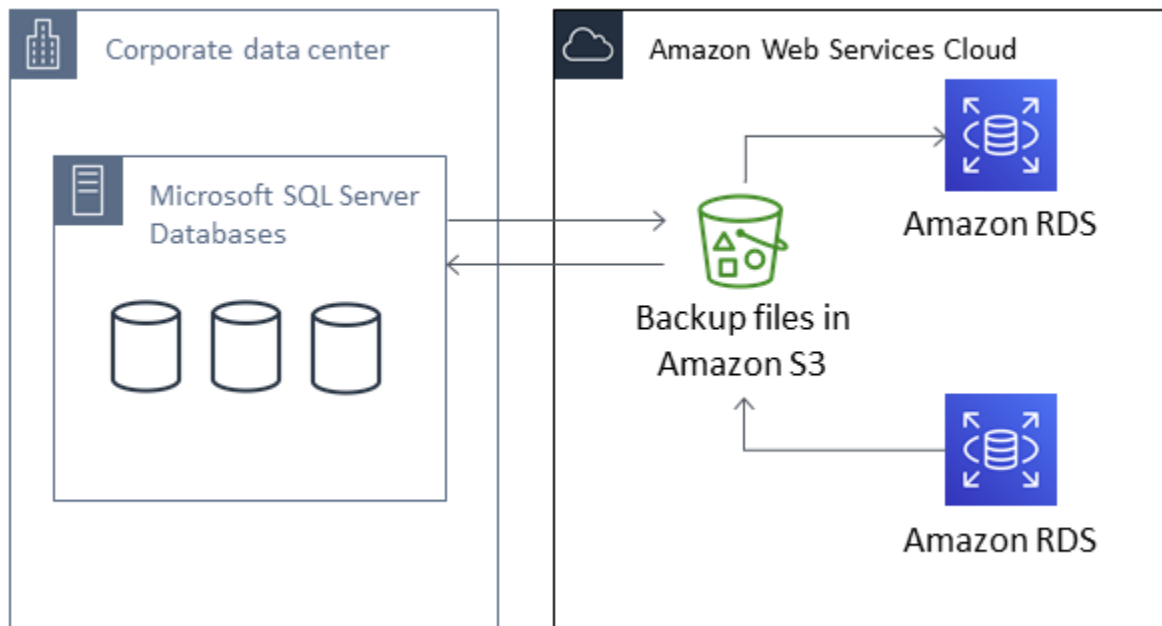
使用原生備份和還原，匯入和匯出 SQL Server 資料庫

Amazon RDS 支援使用完整備份檔案 (.bak 檔案) 進行 Microsoft SQL Server 資料庫的原生備份與還原。當您使用 RDS 時，您會存取存放在 Amazon S3 的檔案，而非使用資料庫伺服器上的本機檔案系統。

例如，您可以從本機伺服器建立完整備份，將它存放在 S3，然後還原到現有的 Amazon RDS 資料庫執行個體。您也可以從 RDS 建立備份，將備份存放在 S3，然後還原到您想要的任何地方。

單一可用 AWS 區域和異地同步備份資料庫執行個體提供原生備份和還原，包括具有僅供讀取複本的異地同步備份資料庫執行個體。對於 Amazon RDS 上支援的所有 Microsoft SQL Server 版本，都可使用原生備份與還原。

下圖顯示所支援的案例。



使用原生 .bak 檔案來備份和還原資料庫，通常是備份和還原資料庫最快的方法。使用原生備份與還原有許多其他優點。例如，您可以執行下列操作：

- 往返於 Amazon RDS 來遷移資料庫。
- 在 RDS for SQL Server 資料庫執行個體之間移動資料庫。
- 遷移 .bak 檔案內的資料結構描述、預存程序、觸發及其他資料庫程式碼。
- 備份和還原單一資料庫，而非整個資料庫執行個體。
- 建立用於開發、測試、訓練和示範的資料庫副本。

- 透過 Amazon S3 存放和傳輸備份檔案，為災難復原添加一道保護。
- 建立透明資料加密 (TDE) 已開啟的資料庫其原生備份，並將這些備份還原至內部部署資料庫。如需詳細資訊，請參閱 [支援 SQL Server 的透明資料加密](#)。
- 將 TDE 已開啟的內部部署資料庫其原生備份還原至 RDS for SQL Server 資料庫執行個體。如需詳細資訊，請參閱 [支援 SQL Server 的透明資料加密](#)。

內容

- [限制與建議](#)
- [設定原生備份與還原](#)
 - [手動建立原生備份與還原的 IAM 角色](#)
- [使用原生備份與還原](#)
 - [備份資料庫](#)
 - [用量](#)
 - [範例](#)
 - [還原資料庫](#)
 - [用量](#)
 - [範例](#)
 - [還原日誌](#)
 - [用量](#)
 - [範例](#)
 - [完成資料庫還原](#)
 - [用量](#)
 - [處理部分還原的資料庫](#)
 - [捨棄部分還原的資料庫](#)
 - [部分還原資料庫的快照還原與point-in-time 復原行為](#)
 - [取消任務](#)
 - [用量](#)
 - [追蹤任務的狀態](#)
 - [用量](#)
 - [範例](#)

- [壓縮備份檔案](#)
- [故障診斷](#)
- [使用其他方法來匯入和匯出 SQL Server 資料](#)
 - [使用快照，將資料匯入 RDS for SQL Server](#)
 - [匯入資料](#)
 - [產生和發佈指令碼精靈](#)
 - [匯入和匯出精靈](#)
 - [大量複製](#)
 - [從 RDS for SQL Server 匯出資料](#)
 - [SQL Server 匯入和匯出精靈](#)
 - [SQL Server 產生和發佈指令碼精靈及 bcp 公用程式](#)

限制與建議

以下是使用原生備份與還原時的一些限制：

- 您無法從 Amazon RDS 資料庫執行個體備份到其他 AWS 區域中的 Amazon S3 儲存貯體，或從中還原。
- 您不能還原與現有資料庫同名的資料庫。資料庫名稱是唯一的。
- 強烈建議您不要將備份從某個時區還原至不同時區。如果您將備份從某個時區還原至不同時區，則必須稽核您的查詢及應用程式，是否受到時區變更的影響。
- Amazon S3 的大小上限為每個檔案 5 TB。對於大型資料庫的原生備份，您可以使用多檔案備份。
- 可備份至 S3 的資料庫大小上限取決於資料庫執行個體上可用的記憶體、CPU、I/O 和網路資源。資料庫越大，備份代理程式所耗用的記憶體就越多。我們的測試顯示，只要有足夠的系統資源，您就可以使用我們 2xlarge 執行個體大小或更大的最新一代執行個體類型，來建立 16 TB 資料庫的壓縮備份。
- 您無法同時備份或是從 10 個備份檔案進行還原。
- 差異備份是以前次完整備份為基礎。您不能在前次完整備份和差異備份之間擷取快照，差異備份才能運作。如果您要建立差異備份，但手動或自動快照存在，請先建立另一個完整備份，再繼續建立差異備份。
- 檔案 file_guid (唯一識別符) 設定為 NULL 的資料庫不支援差異與紀錄還原。
- 您最多可以同時執行 2 個備份或還原任務。

- 您無法從 Amazon RDS 上的 SQL Server 執行原生日誌備份。
- RDS 支援高達 16 TB 資料庫的原生還原。SQL Server Express 上原生還原的資料庫不能超過 10 GB。
- 在維護時段或每當 Amazon RDS 正在擷取資料庫的快照時，您無法進行原生備份。如果原生備份工作與 RDS 每日備份時段重疊，則會取消原生備份工作。
- 在異地同步備份資料庫執行個體上，原本就只能還原以完整還原模式所備份的資料庫。
- 不支援在異地同步備份執行個體上從差異備份還原。
- 不支援在交易內呼叫 RDS 程序來進行原生備份和還原。
- 使用對稱加密 AWS KMS key 來加密備份。Amazon RDS 不支援非對稱 KMS 金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[建立對稱加密 KMS 金鑰](#)。
- 原生備份檔案是在「加密限定」加密模式下，以指定的 KMS 金鑰加密。當您還原已加密的備份檔案時，請注意它們是以「加密限定」加密模式所加密。
- 您無法還原含有 FILESTREAM 檔案群組的資料庫。

在建立、複製和還原備份檔案時，如果您的資料庫可以離線，建議您使用原生備份與還原將資料庫遷移至 RDS。如果您的現場部署資料庫無法離線，建議您使用 AWS Database Migration Service 將資料庫遷移到 Amazon RDS。如需詳細資訊，請參閱[什麼是 AWS Database Migration Service ?](#)

原生備份與還原並無法取代跨區域快照複製功能的資料復原能力。我們建議您使用快照複製將資料庫快照複製到另一個 AWS 區域，以便在 Amazon RDS 中進行跨區域災難復原。如需詳細資訊，請參閱[複製資料庫快照](#)。

設定原生備份與還原

若要設定原生備份與還原，您需要設定三個元件：

1. 用來儲存備份檔案的 Amazon S3 儲存貯體。

您須有 S3 儲存貯體用於備份檔案，然後上傳您要遷移到 RDS 的備份。如果您已經有 Amazon S3 儲存貯體，則可直接使用。如果沒有，您可以[建立儲存貯體](#)。或者，當您使用 SQLSERVER_BACKUP_RESTORE 新增 AWS Management Console 選項時，可以選擇替您建立新的儲存貯體。

如需有關使用 S3 的詳細資訊，請參閱[Amazon Simple Storage Service 使用者指南](#)。

2. 存取值區的 AWS Identity and Access Management (IAM) 角色。

如果您已經有 IAM 角色，則可直接使用。當您使用 AWS Management Console 新增 SQLSERVER_BACKUP_RESTORE 選項時，可以選擇替您建立新的 IAM 角色。或者，您也可以手動建立新的。

如果您想要手動建立新的 IAM 角色，請採取下一節所討論的方法。如果您想要將信任關係和許可政策連接到現有 IAM 角色，請執行相同的動作。

3. 在資料庫執行個體的選項群組中新增的 SQLSERVER_BACKUP_RESTORE 選項。

若要在資料庫執行個體上啟用原生備份與還原，請將 SQLSERVER_BACKUP_RESTORE 選項新增至資料庫執行個體上的選項群組。如需詳細資訊和指示，請參閱 [SQL Server 對原生備份與還原的支援](#)。

手動建立原生備份與還原的 IAM 角色

如果您要手動建立新的 IAM 角色，以用於原生備份和還原，您可以這麼做。在此情況下，您建立角色將 Amazon RDS 服務的許可委派給 Amazon S3 儲存貯體。當您建立 IAM 角色時，您會連接信任關係和許可政策。信任關係可讓 RDS 承擔此角色。許可政策定義此角色可執行的動作。如需有關建立角色的詳細資訊，請參閱 [建立角色以委派許可給 AWS 服務](#)。

對於原生備份與還原功能，請使用類似本節範例的信任關係和許可政策。在下列範例中，我們使用服務原則名稱 `rds.amazonaws.com` 做為所有服務帳戶的別名。在其他範例中，我們指定 Amazon Resource Name (ARN)，以識別我們在信任政策中授予存取的另一個帳戶、使用者或角色。

建議您在資源型信任關係中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰，將服務的許可限定於特定資來源。這是防止 [混淆代理人問題](#) 最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在信任關係中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 ARN。針對原生備份與還原，請確認同時包括資料庫選項群組和資料庫執行個體，如以下範例所示。

Example 與原生備份與還原全域條件內容索引鍵的信任關係

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
            "arn:aws:rds:Region:my_account_ID:og:option_group_name"
          ]
        }
      }
    }
  ]
}
```

下列範例使用 ARN 來指定資源。如需有關使用 ARN 的詳細資訊，請參閱 [Amazon Resource Name \(ARN\)](#)。

Example 原生備份與還原的許可政策 (無加密支援)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    },
    {
      "Effect": "Allow",
```

```

    "Action":
      [
        "s3:GetObjectAttributes",
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  }
]
}

```

Example 原生備份與還原的許可政策 (有加密支援)

如果您要將備份檔案加密，請在許可政策中包含加密金鑰。如需加密金鑰的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [入門](#)。

Note

您必須使用對稱加密 KMS 金鑰，為您的備份加密。Amazon RDS 不支援非對稱 KMS 金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [建立對稱加密 KMS 金鑰](#)。

IAM 角色也必須是 KMS 金鑰的金鑰使用者和金鑰管理員，也就是說，這必須在金鑰政策中指定。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [建立對稱加密 KMS 金鑰](#)。

```

{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Effect": "Allow",
      "Action":
      [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
    }
  ],
}

```



```
    "Resource": "arn:aws:kms:region:account-id:key/key-id"
  },
  {
    "Effect": "Allow",
    "Action":
      [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
    "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET"
  },
  {
    "Effect": "Allow",
    "Action":
      [
        "s3:GetObjectAttributes",
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
    "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
  }
]
```

使用原生備份與還原

啟用並設定原生備份與還原之後，就可以開始使用此選項。您要先連接至 Microsoft SQL Server 資料庫，然後呼叫 Amazon RDS 預存程序來執行工作。如需連接資料庫的指示，請參閱[連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。

您需要提供 Amazon S3 儲存貯體和檔案的 Amazon Resource Name (ARN)，才能使用某些預存程序。ARN 的格式為 `arn:aws:s3:::bucket_name/file_name.extension`。Amazon S3 不需要 ARN 中的帳戶號碼或 AWS 區域。

如果您也提供選用的 KMS 金鑰，金鑰 ARN 的格式為 `arn:aws:kms:region:account-id:key/key-id`。如需詳細資訊，請參閱[Amazon 資源名稱 \(ARN\) 和 AWS 服務命名空間](#)。您必須使用對稱加密 KMS 金鑰，為您的備份加密。Amazon RDS 不支援非對稱 KMS 金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[建立對稱加密 KMS 金鑰](#)。

Note

無論您是否使用 KMS 金鑰，原生備份與還原任務都會針對上傳至 S3 的檔案，預設啟用伺服器端進階加密標準 (AES) 256 位元加密。

如需如何呼叫每個預存程序的相關指示，請參閱下列主題：

- [備份資料庫](#)
- [還原資料庫](#)
- [還原日誌](#)
- [完成資料庫還原](#)
- [處理部分還原的資料庫](#)
- [取消任務](#)
- [追蹤任務的狀態](#)

備份資料庫

若要備份資料庫，請使用 `rds_backup_database` 預存程序。

Note

在維護時段或 Amazon RDS 正在擷取快照時，您無法備份資料庫。

用量

```
exec msdb.dbo.rds_backup_database
  @source_db_name='database_name',
  @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
  [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
  [@overwrite_s3_backup_file=0|1],
  [@type='DIFFERENTIAL|FULL'],
  [@number_of_files=n];
```

下列是必要參數：

- @source_db_name – 要備份之資料庫的名稱。
- @s3_arn_to_backup_to – 此 ARN 指出要用於備份的 Amazon S3 儲存貯體，再加上備份檔案的名稱。

檔案可以有任何副檔名，但通常使用 .bak。

下列是選用參數：

- @kms_master_key_arn – 對稱加密 KMS 金鑰的 ARN，用來加密項目。
 - 您無法使用預設的加密金鑰。如果您使用預設金鑰，資料庫將不會備份。
 - 如果您未指定 KMS 金鑰識別符，則不會加密備份檔案。如需詳細資訊，請參閱[加密 Amazon RDS 資源](#)。
 - 當您指定 KMS 金鑰時，會使用用戶端加密。
 - Amazon RDS 不支援非對稱 KMS 金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[建立對稱加密 KMS 金鑰](#)。
- @overwrite_s3_backup_file – 此值指出是否覆寫現有的備份檔案。
 - 0 – 不覆寫現有檔案。此為預設值。

將 @overwrite_s3_backup_file 設為 0 時，如果檔案已存在，則傳回錯誤。

- 1 – 即使現有檔案不是備份檔案，只要是指定的名稱，就覆寫檔案。

- @type – 備份的類型。
 - DIFFERENTIAL – 建立差異備份。
 - FULL – 建立完整備份。此為預設值。

差異備份是以前次完整備份為基礎。您不能在前次完整備份和差異備份之間擷取快照，差異備份才能運作。如果您要建立差異備份，但快照存在，請先建立另一個完整備份，再繼續建立差異備份。

您可以使用下列範例 SQL 查詢來尋找前次完整備份或快照：

```
select top 1
database_name
, backup_start_date
, backup_finish_date
from msdb.dbo.backupset
where database_name='mydatabase'
and type = 'D'
order by backup_start_date desc;
```

- @number_of_files – 要將備份分割 (分塊) 的檔案數目。上限為 10。
 - 完整和差異備份都支援多個備份。
 - 如果輸入值 1 或省略參數，則會建立單一備份檔案。

提供檔案共有的字首，再於末尾加上星號 (*)。星號可以位於 S3 ARN *file_name* 部分的任何位置。星號會由所產生檔案 (以 1-of-*number_of_files* 開頭) 中的一系列字母數字字串取代。

例如，如果 S3 ARN 中的檔案名稱是 backup*.bak 且您設定 @number_of_files=4，則產生的備份檔案為 backup1-of-4.bak、backup2-of-4.bak、backup3-of-4.bak 及 backup4-of-4.bak。

- 如果任一檔案名稱已存在，且 @overwrite_s3_backup_file 為 0，則會傳回錯誤。
- 多個備份在 S3 ARN 的 *file_name* 部分中只能有一個星號。
- 單一檔案備份在 S3 ARN 的 *file_name* 部分中可以有任意數量的星號。不會從所產生檔案名稱中移除星號。

範例

Example 差異備份

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
```

```
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup1.bak',  
@overwrite_s3_backup_file=1,  
@type='DIFFERENTIAL';
```

Example 有加密的完整備份

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup1.bak',  
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',  
@overwrite_s3_backup_file=1,  
@type='FULL';
```

Example 多個備份

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup*.bak',  
@number_of_files=4;
```

Example 多個差異備份

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup*.bak',  
@type='DIFFERENTIAL',  
@number_of_files=4;
```

Example 多個備份與加密

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup*.bak',  
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',  
@number_of_files=4;
```

Example 多個備份與 S3 覆寫

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3::mybucket/backup*.bak',
```

```
@overwrite_s3_backup_file=1,  
@number_of_files=4;
```

Example 單一檔案備份與 @number_of_files 參數

此範例會產生名為 backup*.bak 的備份檔案。

```
exec msdb.dbo.rds_backup_database  
@source_db_name='mydatabase',  
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',  
@number_of_files=1;
```

還原資料庫

若要還原資料庫，請呼叫 rds_restore_database 預存程序。在還原任務完成且資料庫開啟後，Amazon RDS 會建立資料庫的初始快照。

用量

```
exec msdb.dbo.rds_restore_database  
@restore_db_name='database_name',  
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',  
@with_norecovery=0|1,  
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],  
[@type='DIFFERENTIAL|FULL'];
```

下列是必要參數：

- @restore_db_name– 要還原之資料庫的名稱。資料庫名稱是唯一的。您不能還原與現有資料庫同名的資料庫。
- @s3_arn_to_restore_from – 此 ARN 指出 Amazon S3 字首和用來還原資料庫的備份檔案名稱。
 - 若為單一檔案備份，請提供整個檔案名稱。
 - 若為多檔案備份，請提供檔案共有的字首，再於末尾加上星號 (*)。
 - 如果 @s3_arn_to_restore_from 是空的，則會傳回下列錯誤訊息：S3 ARN 字首不得為空。

以下為差異還原的必要參數，但為完整還原的選用參數：

- @with_norecovery – 用於還原操作的復原子句。

- 設為 0 表示以 RECOVERY 還原。在此情況下，資料庫於還原之後會在線上。
- 設為 1 表示以 NORECOVERY 還原。在此情況下，資料庫於還原任務完成之後會停留在 RESTORING 狀態。此方法可讓您進行後續差異還原。
- 若為 DIFFERENTIAL 還原，請指定 0 或 1。
- 若為 FULL 還原，此值預設為 0。

下列是選用參數：

- @kms_master_key_arn – 如果您已將備份檔案加密，則此為用來解密檔案的 KMS 金鑰。

當您指定 KMS 金鑰時，會使用用戶端加密。

- @type – 還原的類型。有效類型為 DIFFERENTIAL 和 FULL。預設值為 FULL。

Note

若為差異還原，資料庫必須處於 RESTORING 狀態，或必須已存在一個以 NORECOVERY 還原的任務。

當資料庫在線上時，您無法還原後續差異備份。

如果資料庫已有一個使用 RECOVERY 的待定還原任務，則您無法再提交另一個還原任務。

多可用區域執行個體上不支援含 NORECOVERY 的完整還原及差異還原。

在具有僅供讀取複本的異地同步備份執行個體上還原資料庫，與在異地同步備份執行個體上還原資料庫類似。您不需要採取任何其他動作來還原複本上的資料庫。

範例

Example 單一檔案還原

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example 多檔案還原

若要避免在還原多個檔案時發生錯誤，請確保所有備份檔案具有相同的字首，以及沒有其他檔案使用該字首。

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example 含 RECOVERY 的完整資料庫還原

下列三個範例執行同樣的任務，都是含 RECOVERY 的完整還原。

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL|FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example 含加密的完整資料庫還原

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example 含 NORECOVERY 的完整資料庫還原

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=1;
```


Example 含 NORECOVERY 的差異還原

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example 含 RECOVERY 的差異還原

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=0;
```

還原日誌

若要還原日誌，請呼叫 `rds_restore_log` 預存程序。

用量

```
exec msdb.dbo.rds_restore_log
@restore_db_name='database_name',
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@with_norecovery=0/1],
[@stopat='datetime'];
```

下列是必要參數：

- `@restore_db_name` – 資料庫的名稱，表示要還原其日誌。
- `@s3_arn_to_restore_from` – 此 ARN 指出 Amazon S3 字首和用來還原日誌的日誌檔案名稱。檔案可以有任何副檔名，但通常使用 `.trn`。

如果 `@s3_arn_to_restore_from` 是空的，則會傳回下列錯誤訊息：S3 ARN 字首不得為空。

下列是選用參數：

- `@kms_master_key_arn` – 如果您已將日誌加密，則此為用來解密日誌的 KMS 金鑰。

- `@with_norecovery` – 用於還原操作的復原子句。此值預設為 1。
 - 設為 0 表示以 RECOVERY 還原。在此情況下，資料庫於還原之後會在線上。當資料庫在線上時，您無法還原更多的日誌備份。
 - 設為 1 表示以 NORECOVERY 還原。在此情況下，資料庫於還原任務完成之後會停留在 RESTORING 狀態。此方法可讓您進行後續日誌還原。
- `@stopat` – 此值指定將資料庫還原至指定日期和時間的狀態 (日期時間格式)。只有在指定日期和時間之前寫入的交易日誌記錄，才會套用至資料庫。

如果不指定此參數 (NULL)，則會還原整個日誌。

Note

若為日誌還原，資料庫必須處於還原中狀態，或必須已存在一個以 NORECOVERY 還原的任務。

當資料庫在線上時，您無法還原日誌備份。

如果資料庫已有一個使用 RECOVERY 的待定還原任務，則您無法再提交日誌還原任務。

異地同步備份執行個體上不支援日誌還原。

範例

Example 日誌還原

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example 含加密的日誌還原

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example 含 NORECOVERY 的日誌還原

下列兩個範例執行同樣的任務，都是含 NORECOVERY 的日誌還原。

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example 含 RECOVERY 的日誌還原

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example 含 STOPAT 子句的日誌還原

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

完成資料庫還原

如果資料庫的前次還原任務是使用 @with_norecovery=1 執行，則資料庫現在處於 RESTORING 狀態。使用 rds_finish_restore 預存程序開啟資料庫，以正常操作。

用量

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

Note

若要使用此方法，資料庫必須處於 RESTORING 狀態，且沒有任何待定還原任務。

異地同步備份執行個體上不支援 rds_finish_restore 程序。

若要完成還原資料庫，請使用 master 登入。或使用最近以 NORECOVERY 還原資料庫或日誌的使用者登入。

處理部分還原的資料庫

捨棄部分還原的資料庫

若要捨棄局部還原的資料庫 (停留在 RESTORING 狀態), 請使用 `rds_drop_database` 預存程序。

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

Note

如果資料庫已有待定還原任務或已完成還原任務, 則您無法提交 DROP 資料庫請求。若要捨棄資料庫, 請使用 master 登入。或使用最近以 NORECOVERY 還原資料庫或日誌的使用者登入。

部分還原資料庫的快照還原與point-in-time 復原行為

在快照 point-in-time 回復與復原期間, 來源執行處理中部分還原的資料庫 (保留在 RESTORE 狀態) 會從目標執行處理卸除。

取消任務

若要取消備份或還原任務, 請呼叫 `rds_cancel_task` 預存程序。

Note

您無法取消 FINISH_RESTORE 任務。

用量

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

以下是必要參數：

- @task_id- 要取消之任務的 ID。您可以呼叫 `rds_task_status` 來取得任務 ID。

追蹤任務的狀態

若要追蹤備份和還原任務的狀態，請呼叫 `rds_task_status` 預存程序。如果您不提供任何參數，預存程序會傳回所有任務的狀態。任務的狀態大約每兩分鐘更新一次。查詢歷程記錄會保留 36 天。

用量

```
exec msdb.dbo.rds_task_status
  [@db_name='database_name'],
  [@task_id=ID_number];
```

下列是選用參數：

- `@db_name`– 要顯示任務狀態之資料庫的名稱。
- `@task_id`– 要顯示任務狀態之任務的 ID。

範例

Example 列出特定任務的狀態

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example 列出特定資料庫和任務的狀態

```
exec msdb.dbo.rds_task_status
  @db_name='my_database',
  @task_id=5;
```

Example 列出特定資料庫上的所有任務及其狀態

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example 列出目前執行個體上的所有任務及其狀態

```
exec msdb.dbo.rds_task_status;
```

回應

`rds_task_status` 預存程序會傳回下列幾欄。

資料欄	描述
task_id	任務的 ID。
task_type	<p>任務類型取決於輸入參數，如下所示：</p> <ul style="list-style-type: none"> • 備份任務： <ul style="list-style-type: none"> • BACKUP_DB – 完整資料庫備份 • BACKUP_DB_DIFFERENTIAL – 差異資料庫備份 • 還原任務： <ul style="list-style-type: none"> • RESTORE_DB – 含 RECOVERY 的完整資料庫還原 • RESTORE_DB_NORECOVERY – 含 NORECOVERY 的完整資料庫還原 • RESTORE_DB_DIFFERENTIAL – 含 RECOVERY 的差異資料庫還原 • RESTORE_DB_DIFFERENTIAL_NORECOVERY – 含 NORECOVERY 的差異資料庫還原 • RESTORE_DB_LOG – 含 RECOVERY 的日誌還原 • RESTORE_DB_LOG_NORECOVERY – 含 NORECOVERY 的日誌還原 • 完成還原的任務： <ul style="list-style-type: none"> • FINISH_RESTORE – 完成還原並開啟資料庫 <p>當資料庫隨著下列還原任務完成而開啟之後，Amazon RDS 會建立資料庫的初始快照：</p> <ul style="list-style-type: none"> • RESTORE_DB

資料欄	描述
	<ul style="list-style-type: none"> • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
database_name	與任務相關聯之資料庫的名稱。
% complete	任務的進度 (以百分比值表示)。
duration (mins)	任務所花的時間 (以分鐘為單位)。
lifecycle	<p>任務的狀態。可能的狀態如下：</p> <ul style="list-style-type: none"> • CREATED – 呼叫 <code>rds_backup_database</code> 或 <code>rds_restore_database</code> 後，立即會建立任務，且狀態會設為 CREATED。 • IN_PROGRESS – 備份或還原任務開始之後，狀態會設為 IN_PROGRESS。最多可能需要 5 分鐘，狀態才會從 CREATED 變成 IN_PROGRESS。 • SUCCESS – 備份或還原任務完成之後，狀態會設為 SUCCESS。 • ERROR – 如果備份或還原任務失敗，狀態會設為 ERROR。關於錯誤的詳細資訊，請查看 <code>task_info</code> 欄。 • CANCEL_REQUESTED – 呼叫 <code>rds_cancel_task</code> 後，任務的狀態會立即設為 CANCEL_REQUESTED。 • CANCELLED – 成功取消任務之後，任務的狀態會設為 CANCELLED。
task_info	<p>任務的其他資訊。</p> <p>如果備份或還原資料庫時發生錯誤，此欄會包含此錯誤的相關資訊。如需可能錯誤的清單及緩解策略，請參閱故障診斷。</p>

資料欄	描述
last_updated	上次更新任務狀態的日期和時間。進度每完成 5% 後，就會更新一次狀態。
created_at	建立任務的日期和時間。
S3_object_arn	此 ARN 指出 Amazon S3 字首，以及正在備份或還原的檔案名稱。
overwrite_s3_backup_file	呼叫備份任務時所指定之 @overwrite_s3_backup_file 參數的值。如需詳細資訊，請參閱 備份資料庫 。
KMS_master_key_arn	用於加密 (備份時) 和解密 (還原時) 的 KMS 金鑰的 ARN。
filepath	不適用於原生備份與還原任務。
overwrite_file	不適用於原生備份與還原任務。

壓縮備份檔案

若要節省 Amazon S3 儲存貯體的空間，您可以壓縮備份檔案。如需有關壓縮備份檔案的詳細資訊，請參閱 Microsoft 文件中的 [備份壓縮](#)。

下列資料庫版本支援壓縮備份檔案：

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

若要對備份檔案啟用壓縮，請執行下列程式碼：

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'true';
```

若要對備份檔案停用壓縮，請執行下列程式碼：


```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'false';
```

故障診斷

下列是您使用原生備份與還原時可能遇到的問題。

問題	故障診斷建議
資料庫備份/還原選項尚未啟用，或正在進行啟用。請稍後再試。	確定您已將 SQLSERVER_BACKUP_RESTORE 選項新增至與資料庫執行個體產生關聯的資料庫選項群組。如需詳細資訊，請參閱 新增原生備份與還原選項 。
存取遭拒	<p>備份或還原程序無法存取備份檔案。這通常是如下的問題所造成：</p> <ul style="list-style-type: none"> • 參考不正確的儲存貯體。使用不正確的格式來參考儲存貯體。參考檔案名稱時未使用 ARN。 • 儲存貯體檔案上的許可不正確。例如，若是由不同帳戶建立檔案，但此帳戶現在嘗試存取檔案，請新增正確的許可。 • IAM 政策不正確或不完整。IAM 角色必須包含所有必要元素，例如包括正確版本。使用原生備份和還原，匯入和匯出 SQL Server 資料庫中詳加說明這些元素。
<edition_name> 版本不支援備份壓縮資料庫	<p>只有 Microsoft SQL Server Enterprise Edition 才支援壓縮備份檔案， Standard Edition。</p> <p>如需更多詳細資訊，請參閱 壓縮備份檔案。</p>
金鑰 <ARN> 不存在	<p>您嘗試還原已加密的備份，但未提供有效的加密金鑰。請檢查加密金鑰並重試。</p> <p>如需更多詳細資訊，請參閱 還原資料庫。</p>
請使用正確的類型重新發行任務並覆寫屬性	<p>如果您嘗試備份資料庫，且提供已存在之檔案的名稱，但將覆寫屬性設為 false，則儲存操作會失敗。若要修正此錯誤，請提供尚不存在之檔案的名稱，或將覆寫屬性設為 true。</p> <p>如需更多詳細資訊，請參閱 備份資料庫。</p>

問題	故障診斷建議
	<p>也可能是您想要還原資料庫，但不慎呼叫 <code>rds_backup_database</code> 預存程序。在此情況下，請改為呼叫 <code>rds_restore_database</code> 預存程序。</p> <p>如需更多詳細資訊，請參閱 還原資料庫。</p> <p>如果您想要還原資料庫，且呼叫 <code>rds_restore_database</code> 預存程序，請確定您提供有效備份檔案的名稱。</p> <p>如需更多詳細資訊，請參閱 使用原生備份與還原。</p>
請指定與 RDS 執行個體位於相同區域的儲存貯體	<p>您無法從 Amazon RDS 資料庫執行個體備份到其他 AWS 區域中的 Amazon S3 儲存貯體，或從中還原。您可以使用 Amazon S3 複寫將備份檔案複製到正確的 AWS 區域。</p> <p>如需詳細資訊，請參閱 Amazon S3 文件中的 跨區域複寫。</p>
指定的儲存貯體不存在	<p>請確認您以正確格式提供儲存貯體和檔案的正確 ARN。</p> <p>如需更多詳細資訊，請參閱 使用原生備份與還原。</p>
使用者 <ARN> 未獲授權，不可對資源 <ARN> 執行 <kms action>	<p>您要求加密操作，但未提供正確的 AWS KMS 權限。請確認您有正確許可，或新增許可。</p> <p>如需更多詳細資訊，請參閱 設定原生備份與還原。</p>
還原」任務無法從 10 個以上的備份檔案中還原。請減少符合的檔案數目，然後再試一次。	<p>減少您嘗試從中還原的檔案數量。必要時您可以提高每個個別檔案的大小。</p>
資料庫 ' <i>database_name</i> ' 已存在。不允許兩個僅大小寫或口音不同的資料庫。選擇不同的資料庫名稱。	<p>您不能還原與現有資料庫同名的資料庫。資料庫名稱是唯一的。</p>

使用其他方法來匯入和匯出 SQL Server 資料

接下來，您可以找到有關使用快照將 Microsoft SQL Server 資料匯入 Amazon RDS 的資訊。您還可以找到有關從執行 SQL Server 的 RDS 資料庫執行個體匯出快照的資訊。

只要您的案例有支援使用原生備份與還原功能，您即可輕鬆將資料移進和移出 Amazon RDS。如需詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

Note

Amazon RDS for Microsoft SQL Server 不支援將資料匯入 msdb 資料庫。

使用快照，將資料匯入 RDS for SQL Server

使用快照將資料匯入 SQL Server 資料庫執行個體

1. 建立資料庫執行個體。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
2. 防止應用程式存取目的地資料庫執行個體。

如果在匯入資料時防止存取您的資料庫執行個體，資料傳輸會較快。此外，您不必擔心載入資料時，其他應用程式無法同時寫入資料庫執行個體而引起的衝突。如果發生問題而必須轉返到先前的資料庫快照，唯一失去的變更只是已匯入的資料。您可以在解決問題之後重新匯入此資料。

如需有關控制存取資料庫執行個體的資訊，請參閱 [使用安全群組控制存取](#)。

3. 建立目標資料庫的快照。

如果目標資料庫中已填入資料，建議您先建立資料庫的快照，再匯入資料。如果資料匯入發生問題，或您想要捨棄變更，您可以使用快照將資料庫還原到先前的狀態。如需資料庫快照的相關資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

Note

當您建立資料庫快照時，資料庫的 I/O 操作會在備份進行時暫停一會兒 (毫秒)。

4. 在目標資料庫上停用自動備份。

在目標資料庫執行個體上停用自動備份可提升您匯入資料時的效能，因為停用自動備份時，Amazon RDS 不會記錄交易。但有一些注意事項。執行復原需要自動備 point-in-time 份。因

此，您無法在匯入資料期間將資料庫還原至特定時間點。此外，還會清除資料庫執行個體上建立的任何自動備份，除非您選擇保留它們。

選擇保留自動備份可幫助您避免資料意外刪除。Amazon RDS 也會儲存資料庫執行個體屬性以及各自動化備份，以利復原。使用此選項讓您可以在刪除後，還原備份保留期間刪除的特定時間資料庫執行個體。自動備份會在指定備份時段結束時自動刪除，就和它們在作用中資料庫執行個體一樣。

您也可以使用先前的快照來復原資料庫，您已建立的任何快照仍然可用。如需自動備份的相關資訊，請參閱 [備份簡介](#)。

5. 停用外部索引鍵限制 (如適用)。

如果需要停用外部索引鍵限制，您可以利用下列指令碼這樣做。

```
--Disable foreign keys on all tables
DECLARE @table_name SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

OPEN table_cursor;
FETCH NEXT FROM table_cursor INTO @table_name;

WHILE @@FETCH_STATUS = 0 BEGIN
    SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT
ALL';
    EXEC (@cmd);
    FETCH NEXT FROM table_cursor INTO @table_name;
END

CLOSE table_cursor;
DEALLOCATE table_cursor;

GO
```

6. 捨棄索引 (如適用)。

7. 停用觸發條件 (如適用)。

如果需要停用觸發條件，您可以利用下列指令碼這樣做。

```
--Disable triggers on all tables
DECLARE @enable BIT = 0;
```

```
DECLARE @trigger SYSNAME;
DECLARE @table SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
    table_object.name table_name
FROM sysobjects trigger_object
JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
WHERE trigger_object.type = 'TR';

OPEN trigger_cursor;
FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
        SET @cmd = 'ENABLE ';
    ELSE
        SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
END

CLOSE trigger_cursor;
DEALLOCATE trigger_cursor;

GO
```

8. 在來源 SQL Server 執行個體中，查詢您想要匯入到目的地資料庫執行個體的任何登入。

SQL Server 將登入和密碼存放在 master 資料庫中。因為 Amazon RDS 不授權存取 master 資料庫，您無法直接將登入和密碼匯入目的地資料庫執行個體。反之，您必須查詢來源 SQL Server 執行個體上的 master 資料庫，以產生資料定義語言 (DDL) 檔案。此檔案應該包含您要新增至目的地資料庫執行個體的所有登入和密碼。此檔案還應該包含您要轉移的角色成員資格和許可。

如需查詢 master 資料庫的相關資訊，請參閱 Microsoft 知識庫中的[如何在 SQL Server 2005 和 SQL Server 2008 的執行個體之間移轉登入和密碼](#)。

此指令碼會輸出另一個指令碼，供您在目的地資料庫執行個體上執行。知識庫文章中的指令碼有下列程式碼：

```
p.type IN
```

在出現 `p.type` 的每個地方，改用下列程式碼：

```
p.type = 'S'
```

9. 使用[匯入資料](#)中的方法來匯入資料。
10. 授予應用程式存取目標資料庫執行個體的權利。

資料匯入完成時，就可以授予您在匯入期間封鎖的那些應用程式存取資料庫執行個體的權利。如需有關控制存取資料庫執行個體的資訊，請參閱[使用安全群組控制存取](#)。

11. 在目標資料庫執行個體上啟用自動備份。

如需自動備份的相關資訊，請參閱[備份簡介](#)。

12. 啟用外部索引鍵限制。

如果您先前已停用外部索引鍵限制，現在可以利用下列指令碼來啟用它們。

```
--Enable foreign keys on all tables
DECLARE @table_name SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

OPEN table_cursor;
FETCH NEXT FROM table_cursor INTO @table_name;

WHILE @@FETCH_STATUS = 0 BEGIN
    SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
    EXEC (@cmd);
    FETCH NEXT FROM table_cursor INTO @table_name;
END

CLOSE table_cursor;
DEALLOCATE table_cursor;
```

13. 啟用索引 (如適用)。
14. 啟用觸發條件 (如適用)。

如果您先前已停用觸發條件，現在可以利用下列指令碼來啟用它們。

```
--Enable triggers on all tables
DECLARE @enable BIT = 1;
DECLARE @trigger SYSNAME;
DECLARE @table SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
    table_object.name table_name
FROM sysobjects trigger_object
JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
WHERE trigger_object.type = 'TR';

OPEN trigger_cursor;
FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
        SET @cmd = 'ENABLE ';
    ELSE
        SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
END

CLOSE trigger_cursor;
DEALLOCATE trigger_cursor;
```

匯入資料

Microsoft SQL Server Management Studio 是所有 Microsoft SQL Server 版本 (Express Edition 除外) 都包含的圖形化 SQL Server 用戶端。您可以從 Microsoft 免費下載 SQL Server Management Studio Express。若要尋找此下載檔，請造訪 [Microsoft 網站](#)。

Note

SQL Server Management Studio 只推出 Windows 應用程式版本。

SQL Server Management Studio 包含下列工具，您可用來將資料匯入 SQL Server 資料庫執行個體：

- 產生和發佈指令碼精靈
- 匯入和匯出精靈
- 大量複製

產生和發佈指令碼精靈

「產生和發佈指令碼精靈」會建立指令碼，其中包含資料庫的結構描述、資料庫本身或兩者都有。您可以在本機 SQL Server 部署中產生資料庫的指令碼。然後，您可以執行指令碼，將其中包含的資訊傳輸至 Amazon RDS 資料庫執行個體。

Note

如果資料庫是 1 GiB 或更大，只產生資料庫結構描述的指令碼會更有效率。然後，您使用 SQL Server 的「匯入和匯出」精靈或大量複製功能來傳輸資料。

如需「產生和發佈指令碼精靈」的詳細資訊，請參閱此 [Microsoft SQL Server 文件](#)。

在精靈中，請特別注意 Set Scripting Options (設定指令碼編寫選項) 頁面的進階選項，確保已選取您要讓指令碼包含的所有項目。例如，指令碼依預設不包含資料庫觸發條件。

產生並儲存指令碼之後，您可以使用 SQL Server Management Studio 來連接至資料庫執行個體，然後執行指令碼。

匯入和匯出精靈

「匯入和匯出精靈」會建立一個特殊 Integration Services 套件，可用來將本機 SQL Server 資料庫中的資料複製到目的地資料庫執行個體。此精靈可篩選要複製到目的地資料庫執行個體的資料表，甚至是資料表內的元組。

Note

「匯入和匯出精靈」很適合大型資料集，但從遠端匯出本機部署的資料時，可能不是最快的方法。若要使用更快的方法，請考慮採用 SQL Server 大量複製功能。

如需「匯入和匯出精靈」的詳細資訊，請參閱此 [Microsoft SQL Server 文件](#)。

在精靈的 Choose a Destination (選擇目的地) 頁面上，執行下列動作：

- 在 Server Name (伺服器名稱) 中，輸入資料庫執行個體的端點名稱。
- 對於伺服器驗證模式，請選擇 Use SQL Server Authentication (使用 SQL Server 驗證)。
- 在 User name (使用者名稱) 和 Password (密碼) 中，輸入您為資料庫執行個體所建立之主要使用者的登入資料。

大量複製

需要將來源資料庫中的資料複製到資料庫執行個體時，SQL Server 大量複製功能很有效率。大量複製可將您指定的資料寫入資料檔案，例如 ASCII 檔案。然後，您可以再次執行大量複製，將檔案的內容寫入目的地資料庫執行個體。

本節使用所有 SQL Server 版本都有的 bcp 公用程式。如需大量匯入和匯出操作的詳細資訊，請參閱此 [Microsoft SQL Server 文件](#)。

Note

使用大量複製之前，您必須先將資料庫結構描述匯入目的地資料庫執行個體。本主題稍早描述的「產生和發佈指令碼精靈」是此用途的絕佳工具。

下列命令連接至本機 SQL Server 執行個體。然後在現有 SQL Server 部署的 C:\ 根目錄中，產生指定之資料表的 Tab 分隔檔。資料表以完整名稱指定，而文字檔案與所複製的資料表同名。

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -P password -b 10000
```

上述程式碼包含下列選項：

- -n 指定大量複製使用要複製之資料的原生資料類型。
- -S 指定 bcp 公用程式要連接的 SQL Server 執行個體。
- -U 指定登入 SQL Server 執行個體的帳戶使用者名稱。
- -P 指定所指定之使用者的密碼。-U
- -b 指定每一批匯入資料的列數。

Note

根據您的匯入情況而定，可能還有其他重要的參數。例如，您可能需要有關身分值的 `-E` 參數。如需詳細資訊，請參閱此 [Microsoft SQL Server 文件](#) 中有關 `bcp` 公用程式命令列語法的完整描述。

例如，假設有一個使用預設結構描述 (store) 之名為 `dbo` 的資料庫包含名為 `customers` 的資料表。使用者帳戶 `admin` (密碼為 `insecure`) 會將 `customers` 資料表的 10,000 列複製到名為 `customers.txt` 的檔案。

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b 10000
```

產生資料檔案後，您可以使用類似的命令，將資料上傳至資料庫執行個體。事先，在目標資料庫執行個體上建立資料庫和結構描述。然後，使用 `in` 引數來指定輸入檔案，而不是使用 `out` 來指定輸出檔案。指定資料庫執行個體的端點，而不是使用 `localhost` 來指定本機 SQL Server 執行個體。如果您使用的連接埠不是 1433，則也需要指定連接埠。使用者名稱和密碼是資料庫執行個體的主要使用者和密碼。語法如下。

```
bcp dbname.schema_name.table_name
  in C:\table_name.txt -n -S endpoint,port -U master_user_name -
P master_user_password -b 10000
```

接續上一個範例，假設主要使用者名稱是 `admin`，密碼是 `insecure`。資料庫執行個體的端點是 `rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com`，您使用的連接埠是 4080。命令如下所示。

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-
east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

從 RDS for SQL Server 匯出資料

您可以選擇下列其中一個選項，從 RDS for SQL Server 資料庫執行個體匯出資料：

- 使用完整備份檔案 (.bak) 的原生資料庫備份 – 使用 .bak 檔案來備份資料庫已充分最佳化，該方式通常是匯出資料的最快方法。如需更多詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。
- SQL Server 匯入和匯出精靈 – 如需詳細資訊，請參閱 [SQL Server 匯入和匯出精靈](#)。
- SQL Server 產生和發佈指令碼精靈及 bcp 公用程式 – 如需詳細資訊，請參閱 [SQL Server 產生和發佈指令碼精靈及 bcp 公用程式](#)。

SQL Server 匯入和匯出精靈


您可以使用「SQL Server 匯入和匯出」精靈，將 RDS for SQL Server 資料庫執行個體中的一或多個資料表、檢視或查詢，複製到另一個資料存放區。如果目標資料存放區不是 SQL Server，這是最佳選擇。如需詳細資訊，請參閱 SQL Server 文件中的 [SQL Server 匯入和匯出精靈](#)。

「SQL Server 匯入和匯出」精靈隨附於 Microsoft SQL Server Management Studio。所有 Microsoft SQL Server 版本 (Express Edition 除外) 都包含此圖形化 SQL Server 用戶端。SQL Server Management Studio 只推出 Windows 應用程式版本。您可以從 Microsoft 免費下載 SQL Server Management Studio Express。若要尋找此下載檔，請造訪 [Microsoft 網站](#)。

使用 SQL Server 匯入和匯出精靈來匯出資料

1. 在 SQL Server Management Studio 中，連接至 RDS for SQL Server 資料庫執行個體。如需詳細作法，請參閱 [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。
2. 在 Object Explorer (物件總管) 中，展開 Databases (資料庫)，開啟來源資料庫的內容 (右鍵) 功能表，選擇 Tasks (任務)，然後選擇 Export Data (匯出資料)。出現精靈。
3. 在 Choose a Data Source (選擇資料來源) 頁面上，執行下列動作：
 - a. 對於 Data source (資料來源)，請選擇 **SQL Server Native Client 11.0**。
 - b. 確認 Server name (伺服器名稱) 方塊顯示 RDS for SQL Server 資料庫執行個體的端點。
 - c. 選取 Use SQL Server Authentication (使用 SQL Server 驗證)。在 User name (使用者名稱) 和 Password (密碼) 中，輸入資料庫執行個體的主要使用者名稱和密碼。
 - d. 確認 Database (資料庫) 方塊顯示您要從中匯出資料的資料庫。
 - e. 選擇下一步。
4. 在 Choose a Destination (選擇目的地) 頁面上，執行下列動作：

- a. 對於 Destination (目的地), 請選擇 **SQL Server Native Client 11.0**。

 Note

還有其他目標資料來源可用。其中包括 .NET Framework 資料提供者、OLE DB 提供者、SQL Server Native Client 提供者、ADO.NET 提供者、Microsoft Office Excel、Microsoft Office Access 及「一般檔案」來源。如果您選擇以其中一個資料來源為目標, 請略過步驟 4 的其餘部分。關於接下來需提供的連線資訊的詳情, 請參閱 SQL Server 文件中的[選擇目的地](#)。

- b. 在 Server name (伺服器名稱) 中, 輸入目標 SQL Server 資料庫執行個體的伺服器名稱。
- c. 選擇適當的驗證類型。視需要輸入使用者名稱和密碼。
- d. 在 Database (資料庫) 中, 選擇目標資料庫的名稱, 或選擇 New (新增) 來建立新資料庫, 以包含所匯出的資料。

如果您選擇 New (新增), 請參閱 SQL Server 文件中的[建立資料庫](#), 以詳細了解需要提供的資料庫詳細資訊。

- e. 選擇下一步。
5. 在 Table Copy or Query (資料表複製或查詢) 頁面上, 選擇 Copy data from one or more tables or views (從一個或多個資料表或檢視複製資料) 或 Write a query to specify the data to transfer (寫入查詢來指定要傳輸的資料)。選擇下一步。
 6. 如果您選擇 Write a query to specify the data to transfer (寫入查詢來指定要傳輸的資料), 您會看到 Provide a Source Query (提供來源查詢) 頁面。在 SQL 查詢中輸入或貼上, 然後選擇 Parse (下一步) 來進行確認。確認查詢有效後, 選擇 Next (下一步)。
 7. 在 Select Source Tables and Views (選取來源資料表和檢視) 頁面上, 執行下列動作:
 - a. 選取您要匯出的資料表和檢視, 或確認已選取您提供的查詢。
 - b. 選擇 Edit Mappings (編輯映射), 並指定資料庫和資料行映射資訊。如需詳細資訊, 請參閱 SQL Server 文件中的[資料行映射](#)。
 - c. (選用) 如需預覽要匯出的資料, 請選取資料表、檢視或查詢, 然後選擇 Preview (預覽)。
 - d. 選擇下一步。
 8. 在 Run Package (執行封裝) 頁面上, 確認已選取 Run immediately (立即執行)。選擇下一步。
 9. 在 Complete the Wizard (完成精靈) 頁面上, 確認資料匯出詳細資訊如您所預期。選擇 Finish (完成)。
 10. 在 The execution was successful (已執行成功) 頁面上, 選擇 Close (關閉)。

SQL Server 產生和發佈指令碼精靈及 bcp 公用程式

您可以使用「SQL Server 產生和發佈指令碼精靈」來為整個資料庫或只針對選取的物件建立指令碼。您可以在目標 SQL Server 資料庫執行個體上執行這些指令碼，以重新建立指令碼物件。然後，您可以使用 bcp 公用程式，將所選取之物件的資料大量匯出到目標資料庫執行個體。如果您要在兩個 SQL Server 資料庫執行個體之間移動整個資料庫 (包括資料表以外的物件) 或大量資料，這是最佳選擇。如需 bcp 命令列語法的完整描述，請參閱 Microsoft SQL Server 文件中的 [bcp 公用程式](#)。

「SQL Server 產生和發佈指令碼」精靈隨附於 Microsoft SQL Server Management Studio。所有 Microsoft SQL Server 版本 (Express Edition 除外) 都包含此圖形化 SQL Server 用戶端。SQL Server Management Studio 只推出 Windows 應用程式版本。您可以從 Microsoft [免費下載](#) SQL Server Management Studio Express。

使用 SQL Server 產生和發佈指令碼精靈及 bcp 公用程式來匯出資料

1. 在 SQL Server Management Studio 中，連接至 RDS for SQL Server 資料庫執行個體。如需詳細作法，請參閱[連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。
2. 在 Object Explorer (物件總管) 中，展開 Databases (資料庫) 節點，選取您要編寫指令碼的資料庫。
3. 遵循 SQL Server 文件的[產生和發佈指令碼精靈](#)中的說明來建立指令碼檔案。
4. 在 SQL Server Management Studio 中，連接至目標 SQL Server 資料庫執行個體。
5. 在 Object Explorer (物件總管) 中選取目標 SQL Server 資料庫執行個體後，在 File (檔案) 選單上選擇 Open (開啟)，選擇 File (檔案)，然後開啟指令碼檔案。
6. 如果您已產生整個資料庫的指令碼，請檢閱指令碼中的 CREATE DATABASE 陳述式。確定資料庫是按照您要的位置和參數建立。如需詳細資訊，請參閱 SQL Server 文件中的 [CREATE DATABASE](#)。
7. 如果您在指令碼中建立資料庫使用者，請檢查那些使用者的伺服器登入是否存在於目標資料庫執行個體上。如果不是，請建立那些使用者的登入；否則，用於建立資料庫使用者的指令碼命令會失敗。如需詳細資訊，請參閱 SQL Server 文件中的[建立登入](#)。
8. 在「SQL 編輯器」功能表上選擇 !Execute，以執行指令碼檔案和建立資料庫物件。指令碼完成時，確認所有資料庫物件都如預期地存在。
9. 使用 bcp 公用程式將 RDS for SQL Server 資料庫執行個體中的資料匯出到檔案。開啓命令提示並輸入下列命令。

```
bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -
U username -P password
```

上述程式碼包含下列選項：

- `table_name` 是您在目標資料庫中已重新建立，而現在想要將資料填入該資料表的其中一個資料表名稱。
- `data_file` 是準備建立之資料檔案的完整路徑和名稱。
- `-n` 指定大量複製使用要複製之資料的原生資料類型。
- `-S` 指定要從中匯出資料的 SQL Server 資料庫執行個體。
- `-U` 指定連接至 SQL Server 資料庫執行個體時所使用的使用者名稱。
- `-P` 指定所指定之使用者的密碼。 `-U`

以下顯示命令範例 命令。

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

重複此步驟，直到您要匯出的所有資料表都有資料檔案為止。

10. 遵循 SQL Server 文件的[大量匯入資料的基本方針](#)中的指示，以準備目標資料庫執行個體來大量匯入資料。
11. 考慮過 SQL Server 文件的[關於大量匯入和大量匯出操作](#)中所討論的效能和其他考量之後，再決定要使用的大量匯入方法。
12. 使用 `bcp` 公用程式從您建立的資料檔案中大量匯入資料。若要這麼做，請根據您在步驟 11 所做的決定，遵循 SQL Server 文件的[使用 `bcp` 公用程式匯入及匯出大量資料](#)或[使用 `BULK INSERT` 或 `OPENROWSET\(BULK...\)` 匯入大量資料](#)中的指示。

使用 Amazon RDS 中的 Microsoft SQL Server 的僅供讀取複本

您通常會使用僅供讀取複本來設定 Amazon RDS 資料庫執行個體間的複寫。如需僅供讀取複本的一般資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

在此章節，您可以找到在 Amazon RDS 上使用 SQL Server 僅供讀取複本的特定資訊。

主題

- [設定 SQL Server 僅供讀取複本](#)
- [使用 SQL Server 的僅供讀取複本限制](#)
- [RDS for SQL Server 複本的選項考量](#)
- [使用 SQL Server 僅供讀取複本同步資料庫使用者和物件](#)
- [對 SQL Server 僅供讀取複本問題進行故障診斷](#)

設定 SQL Server 僅供讀取複本

當您將資料庫執行個體作為來源執行個體以進行複寫之前，您必須在來源資料庫執行個體上啟用自動備份。若要這麼做，請將備份保留期設定為 0 以外的值。設定此類型的部署也會強制啟用自動備份。

建立 SQL Server 僅供讀取複本不需要停止使用主要資料庫執行個體。Amazon RDS 會為來源資料庫執行個體和僅供讀取複本設定必要的參數和權限，而不會中斷任何服務。將拍攝來源資料庫執行個體的快照，而此快照會變成僅供讀取複本。在您刪除僅供讀取複本不會發生中斷情況。

單一來源資料庫執行個體可建立至多 15 個僅供讀取複本。若希望複寫作業順利運作，建議您將每個僅供讀取複本的運算和儲存資源數量設定為與來源資料庫執行個體相同。若您擴展來源資料庫執行個體，也要擴展僅供讀取複本。

來源資料庫執行個體及其所有僅供讀取複本的 SQL Server 資料庫引擎版本必須相同。Amazon RDS 升級僅供讀取複本後會立即升級主要執行個體，無論維護時段為何。如需有關升級資料庫引擎版本的詳細資訊，請參閱 [升級 Microsoft SQL Server 資料庫引擎](#)。

針對僅供讀取複本接受及套用來源的變更時，應具備足夠的運算及儲存資源。如僅供讀取複本達到運算、網路或儲存資源上限時，僅供讀取複本會停止接受或套用來源的變更。您可以從來源和其他僅供讀取複本獨立修改僅供讀取複本的儲存和 CPU 資源。

使用 SQL Server 的僅供讀取複本限制

下列限制適用於上的 Amazon RDS 上的 SQL Server 僅供讀取複本：

- 僅供讀取複本只適用於 SQL Server Enterprise Edition (EE) 引擎。
- 僅供讀取複本適用於 SQL 伺服器版本
- 單一來源資料庫執行個體可建立至多 15 個僅供讀取複本。當來源資料庫執行個體具有 5 個以上僅供讀取複本時，複寫可能會延遲。
- 僅供讀取複本只適用於資料庫執行個體執行於資料庫執行個體類別具有四個或以上的虛擬 vCPU。
- 僅供讀取複本最多支援 100 個資料庫，視執行個體類別類型和可用性模式而定。您必須在來源資料庫執行個體上建立資料庫，才能自動將其複寫到僅供讀取複本。您無法選擇要複製的個別資料庫。如需詳細資訊，請參閱 [Microsoft SQL Server 資料庫執行個體的限制](#)。
- 您無法從僅供讀取複本卸除資料庫。若要刪除資料庫，請使用 `rds_drop_database` 預存程序從來源資料庫執行個體刪除該資料庫。如需詳細資訊，請參閱 [捨棄 Microsoft SQL Server 資料庫](#)。
- 如果來源資料庫執行個體使用通透資料加密 (TDE) 來加密資料，僅供讀取複本也會自動設定 TDE。

如果來源資料庫執行個體使用 KMS 金鑰來加密資料，則相同區域中的僅供讀取複本會使用相同的 KMS 金鑰。對於跨區域僅供讀取複本，您必須在建立僅供讀取複本時，從僅供讀取複本的區域指定 KMS 金鑰。您無法變更僅供讀取複本的 KMS 金鑰。

- 僅供讀取複本與來源資料庫執行個體具有相同的時區和定序，無論其建立於何種可用區域。
- 僅供讀取複本只適用於資料庫執行個體執行於資料庫執行個體類別具有四個或以上的虛擬 vCPU。
- SQL Server 的 Amazon RDS 不支援下列項目：
 - 僅供讀取複本的備份保留
 - 從僅供讀取複本 Point-in-time 復原
 - 僅供讀取複本的手動快照
 - 異地同步備份僅供讀取複本
 - 建立僅供讀取複本的僅供讀取複本
 - 使用者登入至僅供讀取複本的同步處理
- SQL Server 的 Amazon RDS 來源資料庫執行個體與僅供讀取複本之間不會相互干涉，以降低複寫時的高延遲。確保資料庫執行個體及僅供讀取複本都依照運算能力和儲存適當的調整大小，以搭配工作負載。
- 您可以在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 區域之間進行複製，但不能進出。AWS GovCloud (US) Regions

RDS for SQL Server 複本的選項考量

建立 RDS for SQL Server 複本之前，請先考慮下列要求、限制和建議：

- 如果 SQL Server 複本與其來源資料庫執行個體位於相同的區域，請確定它與來源資料庫執行個體屬於相同的選項群組。修改來源選項群組或來源選項群組成員參數到複本。套用至來源資料庫執行個體後變更會立即套用至複本，不論複本是否在維護視窗。

如需選項群組的詳細資訊，請參閱[使用選項群組](#)。

- 當您建立 SQL Server 跨區域複本時，Amazon RDS 會為此複本建立專用選項群組。

您無法從專用選項群組中移除 SQL Server 跨區域複本。其他資料庫執行個體都不能使用 SQL Server 跨區域複本的專用選項群組。

下列選項是複寫選項：若要將複寫選項新增至 SQL Server 跨區域複本，請將其新增至來源資料庫執行個體的選項群組。選項也安裝在所有來源資料庫執行個體的複本上。

- TDE

下列選項是非複寫選項：您可以從專用選項群組新增或移除下列非複寫選項：

- MSDTC
- SQLSERVER_AUDIT
- 若要在跨區域僅供讀取複本上啟用 SQLSERVER_AUDIT 選項，請在跨區域僅供讀取複本的專用選項群組和來源執行個體的選項群組上新增 SQLSERVER_AUDIT 選項。透過在 SQL Server 跨區域僅供讀取複本的來源執行個體上新增 SQLSERVER_AUDIT 選項，您可以在來源執行個體的每個跨區域僅供讀取複本上建立伺服器層級稽核物件和伺服器層級稽核規格。若要允許跨區域僅供讀取複本存取，以將完成的稽核日誌上傳到 Amazon S3 儲存貯體，請將 SQLSERVER_AUDIT 選項新增到專用選項群組，並設定選項設定。您用作稽核檔案目標的 Amazon S3 儲存貯體必須與跨區域僅供讀取複本位在同一區域。您可以獨立修改每個跨區域僅供讀取複本之 SQLSERVER_AUDIT 選項的選項設定，以便每個複本都可以存取各自區域中的 Amazon S3 儲存貯體。

跨區域僅供讀取複本不支援下列選項：

- SSRS
- SSAS
- SSIS

跨區域僅供讀取複本局部支援下列選項：

- SQLSERVER_BACKUP_RESTORE
- SQL Server 跨區域複本的來源資料庫執行個體可以具有 SQLSERVER_BACKUP_RESTORE 選項，但您無法在來源資料庫執行個體上執行原生還原，直到刪除其所有跨區域複本為止。建立跨區域複

本期間，將會取消任何現有的原生還原任務。您無法將 SQLSERVER_BACKUP_RESTORE 選項新增至專用選項群組。

如需原生備份和還原的詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

當您提升 SQL Server 跨區域僅供讀取複本時，提升的複本就像其他 SQL Server 資料庫執行個體一樣運作，包括管理其選項。如需選項群組的詳細資訊，請參閱 [使用選項群組](#)。

使用 SQL Server 僅供讀取複本同步資料庫使用者和物件

建立僅供讀取複本時，主要資料庫執行個體中存在的任何登入、自訂伺服器角色、SQL 代理程式任務或其他伺服器層級物件都應該會呈現在新建立的僅供讀取複本中。不過，建立僅供讀取複本後，在主要資料庫執行個體中建立的任何伺服器層級物件都不會自動複寫，而且您必須在僅供讀取複本中手動建立這些物件。

資料庫使用者會自動從主要資料庫執行個體複寫到僅供讀取複本。由於僅供讀取複本資料庫處於唯讀模式，因此無法在資料庫中更新資料庫使用者的安全性識別碼 (SID)。因此，在僅供讀取複本中建立 SQL 登入時，務必確保該登入的 SID 符合主要資料庫執行個體中對應 SQL 登入的 SID。如果您未同步 SQL 登入的 SID，它們將無法存取僅供讀取複本中的資料庫。Windows Active Directory (AD) 驗證的登入不會遇到這個問題，因為 SQL Server 會從 Active Directory 取得 SID。

將主要資料庫執行個體中的 SQL 登入同步至僅供讀取複本

1. 連線至主要資料庫執行個體。
2. 在主要資料庫執行個體中建立新的 SQL 登入。

```
USE [master]
GO
CREATE LOGIN TestLogin1
WITH PASSWORD = 'REPLACE WITH PASSWORD';
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

3. 為資料庫中的 SQL 登入建立新的資料庫使用者。

```
USE [REPLACE WITH YOUR DB NAME]
```

```
GO
CREATE USER TestLogin1 FOR LOGIN TestLogin1;
GO
```

4. 檢查主要資料庫執行個體中新建立之 SQL 登入的 SID。

```
SELECT name, sid FROM sys.server_principals WHERE name = TestLogin1;
```

5. 連線至僅供讀取複本。建立新的 SQL 登入。

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH sid FROM STEP #4];
```

或者，如果您有僅供讀取複本資料庫的存取權，則可以依照下列方式修正孤立的使用者：

1. 連線至僅供讀取複本。
2. 識別資料庫中孤立的使用者。

```
USE [REPLACE WITH YOUR DB NAME]
GO
EXEC sp_change_users_login 'Report';
GO
```

3. 為孤立的資料庫使用者建立新的 SQL 登入。

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH sid FROM STEP #2];
```

範例：

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'TestPa$$word#1',
SID=[0x1A2B3C4D5E6F7G8H9I0J1K2L3M4N506P];
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

對 SQL Server 僅供讀取複本問題進行故障診斷

您可以 CloudWatch 透過檢視 Amazon RDS ReplicaLag 指標來監控 Amazon 中的複寫延遲。如需複寫延遲時間的資訊，請參閱 [監控僅供讀取複本](#)。

如複寫延遲的時間過長，您可以使用下列查詢以取得延遲相關資訊。

```
SELECT AR.replica_server_name
      , DB_NAME (ARS.database_id) 'database_name'
      , AR.availability_mode_desc
      , AR.synchronization_health_desc
      , AR.last_hardened_lsn
      , AR.last_redone_lsn
      , AR.secondary_lag_seconds
FROM sys.dm_hadr_database_replica_states ARS
INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
ORDER BY AR.replica_server_name;
```

Amazon RDS for Microsoft SQL Server 的異地同步備份部署

「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。如果發生規劃的資料庫維護或意外服務中斷，Amazon RDS 會自動容錯移轉到 up-to-date 次要資料庫執行個體。此功能可讓資料庫操作迅速恢復，而不需要手動介入。主要和待命執行個體使用相同的端點，而此端點的實體網路地址在容錯移轉過程中會轉移到次要複本。當容錯移轉發生時，您不必重新設定應用程式。

Amazon RDS 會透過使用 SQL Server Database Mirroring (DBM) 或 Always On 可用性群組 (AG)，支援 Microsoft SQL Server 的異地同步備份部署。Amazon RDS 會監控和維護異地同步備份部署的運作狀態。如果發生問題，RDS 會自動修復運作不良的資料庫執行個體、重新建立同步處理，以及啟動容錯移轉。只有當待命和主要執行個體完全同步時，容錯移轉才會發生。您什麼都不必管理。

當您設定 SQL Server 多可用區域時，RDS 會自動將執行個體上的所有資料庫設定為使用 DBM 或 AG。Amazon RDS 會為您處理主要、見證及次要資料庫執行個體。因為組態是自動的，所以 RDS 會根據您部署的 SQL Server 版本選取 DBM 或 Always On AG。

Amazon RDS 支援下列 SQL Server 版本和版次使用異地同步備份搭配 Always On AG：

- SQL 伺服器
 - 標準版本
 - 企業版本
- SQL Server 2019：
 - Standard Edition 15.00.4073.23 版和更高版本
 - 企業版本
- SQL Server 2017：
 - Standard Edition 14.00.3401.7 版和更高版本
 - Enterprise Edition 14.00.3049.1 版和更高版本
- SQL Server 2016：企業版 13.00.5216.0 版和更新版本

Amazon RDS 支援下列 SQL Server 版本使用異地同步備份搭配 DBM，但先前提到的版本除外：

- SQL Server 2019：標準版 15.00.4043.16 版
- SQL Server 2017：Standard 和 Enterprise Edition
- SQL Server 2016：Standard 和 Enterprise Edition
- SQL Server 2014：Standard 和 Enterprise Edition

您可以使用下列 SQL 查詢來判斷 SQL Server 資料庫執行個體是單一可用區、採用 DBM 的多可用區域，或是採用 Always On AG 的多可用區域。

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
           WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
           ELSE 'Single-AZ'
           END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id =
dhdrs.database_id AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

輸出看起來會與以下內容相似：

```
high_availability
Multi-AZ (AlwaysOn)
```

將多可用區域新增至 Microsoft SQL Server 資料庫執行個體

使用建立新的 SQL Server 資料庫執行個體時 AWS Management Console，您可以使用資料庫鏡像 (DBM) 或永遠開啟 AGS 來新增異地同步備份。您可以從異地同步備份部署中選擇是 (鏡像 / 永遠開啟) 來執行此操作。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

在使用主控台修改現有 SQL Server 資料庫執行個體時，您可以從 Modify DB instance (修改資料庫執行個體) 頁面的 Multi-AZ deployment (多可用區部署) 中選擇 Yes (Mirroring / Always On) (是 (鏡像/永遠開啟))，以新增多個搭配了 DBM 或 AG 的多可用區域。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Note

如果您的資料庫執行個體正在執行資料庫鏡像 (DBM) —並非 Always On 可用性群組 (AG)，則—可能需要先停用記憶體內最佳化，再新增異地同步備份。如果您的資料庫執行個體執行 SQL Server 2014、2016 年或 2017 年企業版，並啟用記憶體內最佳化，請在新增異地同步備份之前停用 DBM 的記憶體內最佳化。

如果您的資料庫執行個體正在執行 AG，則不需要此步驟。

從 Microsoft SQL Server 資料庫執行個體移除多可用區域

當您使用修改現有的 SQL Server 資料庫執行個體時 AWS Management Console，您可以使用 DBM 或 AGS 移除異地同步備份。若要這麼做，請從 Modify DB instance (修改資料庫執行個體) 頁面的 Multi-AZ deployment (多可用區部署) 中選擇 No (Mirroring / Always On) (否 (鏡像/永遠開啟))。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Microsoft SQL Server 異地同步備份部署限制、備註和建議

下列是對 RDS for SQL Server 資料庫執行個體使用多可用區部署的一些限制：

- 不支援跨區域的多可用區域。
- 不支援停止多可用區部署中的 RDS for SQL Server 資料庫執行個體。
- 您無法設定次要資料庫執行個體來接受資料庫讀取活動。
- 多可用區域搭配 Always On 可用區 (AG) 支援記憶體內最佳化。
- 具有 Always On 可用性群組 (AG) 的異地同步備份不支援可用性群組接聽程式的 Kerberos 驗證。這是因為接聽程式沒有服務委託人名稱 (SPN)。
- 在 SQL Server 異地同步備份部署中的 SQL Server 資料庫執行個體上，您無法重新命名資料庫。如果您需要在這類執行個體上重新命名資料庫，請先對資料庫執行個體關閉多可用區域，再重新命名資料庫。最後，對資料庫執行個體重新開啟多可用區域。
- 您只能還原使用完整還原模型備份的多可用區域資料庫執行個體。
- 多可用區部署有 10,000 個 SQL Server Agent 任務的限制。

如果您需要更高的限制，請聯繫以申請增加 AWS Support。開啟 [AWS Support 中心](#) 頁面，如有必要請登入，然後選擇 Create case (建立案例)。選擇 Service limit increase (提高服務限制)。填妥並提交表格。

下列是對 RDS for SQL Server 資料庫執行個體使用多可用區部署的一些相關備註：

- Amazon RDS 公開 Always On AG [可用性群組接聽程式端點](#)。此端點會出現在主控台，而且由 DescribeDBInstances API 作業在端點欄位中傳回。
- Amazon RDS 支援 [可用性群組多重子網路容錯移轉](#)。
- 若要對虛擬私有雲端 (VPC) 中的 SQL Server 資料庫執行個體使用 SQL Server 多可用區域，首先請建立在至少兩個不同可用區域中有子網路的資料庫子網路群組。然後，將該資料庫子網路群組指派給 SQL Server 資料庫執行個體的主要複本。

- 當資料庫執行個體修改為異地同步備份部署時，其在修改期間的狀態為 `modifying` (修改中)。Amazon RDS 會建立待命執行個體，並建立主要資料庫執行個體的備份。在程序完成之後，主要資料庫執行個體的状态會變成 `available` (可用)。
- 異地同步備份部署會在相同節點上維護所有資料庫。如果主要主機上的資料庫容錯移轉，您的所有 SQL Server 資料庫會以一個不可分割的單位容錯移轉至您的待命主機。Amazon RDS 會佈建運作狀態良好的新主機，並取代狀況不良的主機。
- 採用 DBM 或 AG 的多可用區域支援單一待命複本。
- 使用者、登入和許可會自動複寫到次要主機。您不需要重新建立這些資料。只會針對在多可用區部署使用 Always On AG 的資料庫執行個體複寫使用者定義的伺服器角色。
- 在異地同步備份部署中，適用於 SQL Server 的 RDS 會建立 SQL Server 登入，以允許永遠開啟 AG 或資料庫鏡像。RDS 使用下列模式、`db_<dbiResourceId>_node1_login`、`db_<dbiResourceId>_node2_login`和`db_<dbiResourceId>_node_login`。
- 適用於 SQL 伺服器的 RDS 會建立 SQL 伺服器登入，以允許存取僅供讀取複本。RDS 使用下列模式建立登入`db_<readreplica_dbiResourceId>_node_login`。
- 在異地同步備份部署中，當任務複寫功能開啟時，SQL Server Agent 任務從主要主機複製至次要主機。如需詳細資訊，請參閱 [開啟 SQL Server Agent 任務複寫](#)。
- 由於同步資料複寫，相較於標準資料庫執行個體部署 (在單一可用區域中)，您可能會發現延遲增加情形。
- 容錯移轉時間會受到完成復原程序所需的時間影響。大型交易會增加此容錯移轉時間。
- 於 SQL Server 異地同步備份部署中，透過容錯移轉重新啟動只會重新啟動主要資料庫執行個體。在容錯移轉之後，主要資料庫執行個體會成為新的次要資料庫執行個體。多可用區域執行個體的參數可能無法更新。若要在無容錯移轉的情況下重新啟動，主要和次要資料庫執行個體會在重新啟動後更新參數。若資料庫執行個體沒有回應，建議您重新開機，不要進行容錯移轉。

下列是對 RDS for Microsoft SQL Server 資料庫執行個體使用異地同步備份部署的一些建議：

- 針對生產階段或生產前階段所用的資料庫，建議您使用下列選項：
 - 針對高可用性的異地同步備份部署
 - 「佈建 IOPS」可取得快速一致的績效
 - 「最佳化記憶體」而非「一般用途」
- 您無法為次要執行個體選取可用區域 (AZ)，因此，當您部署應用程式主機時，請將此納入考量。您的資料庫可能容錯移轉至其他 AZ，而且應用程式主機和資料庫可能不是在相同的 AZ 中。因此，我們建議您平衡指定 AWS 區域中所有 AZ 的應用程式主機。

- 為了獲得最佳效能，請勿在大型資料載入操作期間啟用資料庫鏡像或 Always On AG。如果要盡快載入您的資料，請在將您的資料庫執行個體轉換為異地同步備份部署之前完成載入資料。
- 存取 SQL Server 資料庫的應用程式應該具有例外處理能力可捕捉連接錯誤。下列程式碼範例示範捕捉通訊錯誤的 try/catch 區塊。在這個範例中，如果連線成功，break 陳述式會結束 while 迴圈，但如果擲回例外狀況，則重試最多 10 次。

```
int RetryMaxAttempts = 10;
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
    using (SqlConnection connection = new SqlConnection(DatabaseConnString))
    {
        using (SqlCommand command = connection.CreateCommand())
        {
            command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue')";
            try
            {
                connection.Open();
                command.ExecuteNonQuery();
                break;
            }
            catch (Exception ex)
            {
                Logger(ex.Message);
                iRetryCount++;
            }
            finally {
                connection.Close();
            }
        }
    }
    Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
}
```

- 使用多可用區域執行個體時，請勿使用 Set Partner Off 命令。例如，請勿執行以下操作。

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- 請勿將復原模式設為 simple。例如，請勿執行以下操作。

```
--Don't do this  
ALTER DATABASE db1 SET RECOVERY simple
```

- 在多可用區域資料庫執行個體上建立新登入時，請勿使用 `DEFAULT_DATABASE` 參數，因為這些設定無法套用至待命鏡像。例如，請勿執行以下操作。

```
--Don't do this  
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

也請勿執行以下操作。

```
--Don't do this  
ALTER LOGIN [test_dba] SET DEFAULT_DATABASE=[db3]
```

查明次要複本的位置

您可以使用 AWS Management Console 來查明次要複本的位置。如果是在 VPC 中設定您的主要資料庫執行個體，您需要知道次要複本的位置。

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Instance					
Configuration		Instance class		Storage	
DB instance id database-1		Instance class db.m4.large		Encryption Enabled	
Engine version 14.00.3192.2.v1		vCPU 2		KMS key aws/rds	
DB name -		RAM 8 GB		Storage type General Purpose (SSD)	
License model License Included		Availability		IOPS -	
Collation SQL_Latin1_General_CP1_CI_AS		Master username admin		Storage 20 GiB	
Option groups default:sqlserver-se-14-00		IAM db authentication Not Enabled		Storage autoscaling Enabled	
ARN arn:aws:rds:us-west-2: :db:database-1		Multi AZ Yes (Mirroring)		Maximum storage threshold 1000 GiB	
Resource id db-		Secondary Zone us-west-2c			

您也可以使用 AWS CLI 命令 `describe-db-instances` 或 RDS API 作業來檢視次要的可用區域 `DescribeDBInstances`。輸出會顯示待命鏡像所在的次要 AZ。

從資料庫鏡像遷移到 Always On 可用性群組 (AG)

Microsoft SQL Server 企業版本的 14.00.3049.1 版中，依預設可啟用 Always On 可用性群組 (AG)。

若要從資料庫鏡像 (DBM) 遷移到可用性群組 (AG)，請先檢查您的版本。如果您使用的資料庫執行個體是企業版 13.00.5216.0 以前的版本，請修改執行個體來修補為 13.00.5216.0 或更新版本。如果您使用的資料庫執行個體是企業版 14.00.3049.1 以前的版本，請修改執行個體來修補為 14.00.3049.1 或更新版本。

如果您想要升級鏡像資料庫執行個體以使用可用性群組，請先升級、修改執行個體以移除多可用區域，然後再次修改以新增多可用區域。這樣會將您的執行個體轉換為使用 Always On 可用性群組。

Microsoft SQL Server on Amazon RDS 的其他功能

在下列章節中，您可以找到執行 Microsoft SQL Server 資料庫引擎擴增 Amazon RDS 執行個體的相關資訊。

主題

- [對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)
- [設定安全性通訊協定和密碼](#)
- [整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)
- [在 Amazon RDS for SQL Server 上使用 Database Mail](#)
- [執行個體存放區支援 Amazon RDS for SQL Server 上的 tempdb 資料庫](#)
- [搭配 Amazon RDS for Microsoft SQL Server 使用擴充事件](#)
- [使用 RDS for SQL Server 存取交易日誌備份](#)

對 Microsoft SQL Server 資料庫執行個體使用 SSL

您可以使用 Secure Sockets Layer (SSL)，來加密用戶端應用程式與執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體之間的連線。SSL 支援適用於所有 AWS 區域中所有支援的 SQL Server 版本。

建立 SQL Server 資料庫執行個體時，Amazon RDS 會為執行個體建立 SSL 憑證。SSL 憑證會以一般名稱 (CN) 納入資料庫執行個體端點，讓 SSL 憑證免於詐騙攻擊。

有兩種方法可使用 SSL 來連接至 SQL Server 資料庫執行個體：

- 強制所有連線使用 SSL — 用戶端自然會這樣做，用戶端不必有任何動作即可使用 SSL。
- 加密特定的連線 — 這會設定來自特定用戶端電腦的 SSL 連線，您必須在用戶端設定才能加密連線。

如需 SQL Server 的 Transport Layer Security (TLS) 支援有關的資訊，請參閱 [Microsoft SQL Server 的 TLS 1.2 支援](#)。

強制連線至資料庫執行個體以使用 SSL

您可以強制資料庫執行個體的所有連線都使用 SSL。如果您強制所有連線都使用 SSL，用戶端自然就會這樣做，用戶端不必有任何動作即可使用 SSL。

如果您要強制使用 SSL，請使用 `rds.force_ssl` 參數。依預設，`rds.force_ssl` 參數設為 0 (off)。將 `rds.force_ssl` 參數設為 1 (on) 可強制連線使用 SSL。`rds.force_ssl` 參數是靜態，當您變更值之後，您必須重新啟動資料庫執行個體，才能使變更生效。

強制資料庫執行個體的所有連線都使用 SSL

1. 查明連接到資料庫執行個體的參數群組：
 - a. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
 - b. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體的 AWS 區域。
 - c. 在導覽窗格中選擇 Databases (資料庫)，然後選擇資料庫執行個體的名稱來顯示詳細資訊。
 - d. 選擇 Configuration (組態) 標籤。在區段中找出 Parameter group (參數群組)。
2. 如果需要，請建立新的參數群組。如果您的資料庫執行個體使用預設參數群組，則您必須建立新的參數群組。如果您的資料庫執行個體使用非預設參數群組，您可以選擇編輯現有的參數群組，或建立新的參數群組。如果您編輯現有的參數群組，則變更會影響所有使用該參數群組的資料庫執行個體。

若要建立新的參數群組，請遵循[建立資料庫參數群組](#)中的說明。

3. 編輯新的或現有的參數群組，將 `rds.force_ssl` 參數設為 `true`。若要編輯參數群組，請遵循[修改資料庫參數群組中的參數](#)中的說明。
4. 如果您已建立新的參數群組，請修改資料庫執行個體來附加新的參數群組。修改資料庫執行個體的 DB Parameter Group (資料庫參數群組) 設定。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
5. 重新啟動您的資料庫執行個體。如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

加密特定連線

您可以強制資料庫執行個體的所有連線都使用 SSL，也可以只加密來自特定用戶端電腦的連線。若要從特定用戶端使用 SSL，您必須為用戶端電腦取得憑證，在用戶端電腦匯入憑證，然後加密來自用戶端電腦的連線。

Note

所有在 2014 年 8 月 5 日之後建立的 SQL Server 執行個體，在 SSL 憑證的一般名稱 (CN) 欄位中都使用資料庫執行個體端點。在 2014 年 8 月 5 日之前，不會對 VPC 式 SQL Server 執行個體使用 SSL 憑證驗證。如果您有 2014 年 8 月 5 日之前建立的 VPC 式 SQL Server 資料庫執行個體，而且想要使用 SSL 憑證驗證，並確定將執行個體端點納入做為該資料庫執行個體之 SSL 憑證的 CN，請重新命名執行個體。重新命名資料庫執行個體會部署新的憑證，且執行個體會重新啟動，以啟用新的憑證。

為用戶端電腦取得憑證

若要將用戶端電腦到執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體的連線加密，您在用戶端電腦上需要憑證。

若要取得該憑證，請將憑證下載至用戶端電腦。您可以下載適用於所有區域的根憑證，也可以下載同時包含新舊根憑證的憑證套件。此外，您還能下載區域特定的中繼憑證。如需下載憑證的詳細資訊，請參閱。

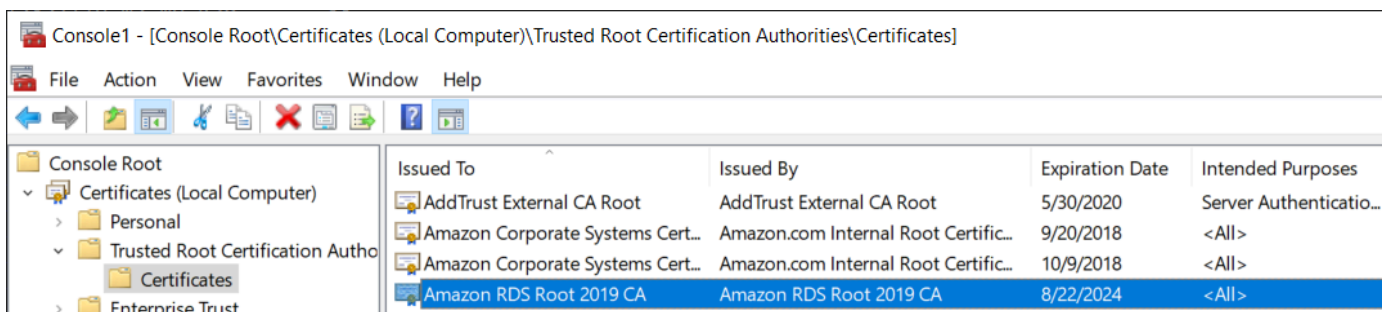
下載適當的憑證之後，請遵循下一節的程序，將該憑證匯入 Microsoft Windows 作業系統。

在用戶端電腦匯入憑證

您可以使用下列程序，將憑證匯入用戶端電腦上的 Microsoft Windows 作業系統。

將憑證匯入 Windows 作業系統：

1. 在 Start (開始) 功能表上，在搜尋方塊中輸入 **Run**，然後按 Enter (Enter)。
2. 在 Open (開啟) 方塊中，輸入 **MMC**，然後選擇 OK (確定)。
3. 在 MMC 主控台的 File (檔案) 功能表上，選擇 Add/Remove Snap-in (新增/移除嵌入式管理單元)。
4. 在 Add or Remove Snap-ins (新增或移除嵌入式管理單元) 對話方塊中，從 Available snap-ins (可用的嵌入式管理單元) 中選取 **Certificates** (憑證)，然後選擇 Add (新增)。
5. 在 Certificates snap-in (憑證嵌入式管理單元) 對話方塊中，選擇 Computer account (電腦帳戶)，然後選擇 Next (下一步)。
6. 在 Select computer (選取電腦) 對話方塊中，選擇 Finish (完成)。
7. 在 Add or Remove Snap-ins (新增或移除嵌入式管理單元) 對話方塊中，選擇 OK (確定)。
8. 在 MMC 主控台，展開 Certificates (憑證)，開啟 Trusted Root Certification Authorities (受信任的根憑證授權單位) 的內容 (右鍵) 功能表，選擇 All Tasks (所有任務)，然後選擇 Import (匯入)。
9. 在「憑證匯入精靈」的第一頁，選擇 Next (下一步)。
10. 在「憑證匯入精靈」的第二頁，選擇 Browse (瀏覽)。在瀏覽視窗中，將檔案類型變更為 All files (*.*) (所有檔案 (*.*))，因為 .pem 不是標準憑證副檔名。找出您先前下載的 .pem 檔案。
11. 選擇 Open (開啟舊檔)，以選取憑證檔案，然後選擇 Next (下一步)。
12. 在「憑證匯入精靈」的第三頁，選擇 Next (下一步)。
13. 在「憑證匯入精靈」的第四頁，選擇 Finish (完成)。隨即出現對話方塊指出匯入成功。
14. 在 MMC 主控台，展開 Certificates (憑證)，展開 Trusted Root Certification Authorities (受信任的根憑證授權單位)，然後選擇 Certificates (憑證)。找出憑證以確認其存在，如此處所示。



將執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體的連線加密

將憑證匯入用戶端電腦之後，您就可以將用戶端電腦到執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體的連線加密。

若為 SQL Server Management Studio，請使用以下程序。如需 SQL Server Management Studio 的詳細資訊，請參閱[使用 SQL Server Management Studio](#)。

從 SQL Server Management Studio 加密連線

1. 啟動 SQL Server Management Studio。
2. 在 Connect to server (連線至伺服器) 中，鍵入伺服器資訊、登入使用者名稱和密碼。
3. 選擇 Options (選項)。
4. 選取 Encrypt connection (加密連線)。
5. 選擇 Connect (連線)。
6. 執行下列查詢來確認連線已加密。確認查詢傳回的 true 為 encrypt_option。

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

若為其他任何 SQL 用戶端，請使用以下程序。

加密來自其他 SQL 用戶端的連線

1. 將 encrypt=true 附加至連線字串。此字串可能是選項，也可能是 GUI 工具中連線頁面上的屬性。

Note

對於使用 JDBC 連接的用戶端，若要啟用 SSL 加密，您可能需要將 Amazon RDS SQL 憑證新增至 Java CA 憑證 (cacert) 存放區。您可使用 [keytool](#) 公用程式做到這點。

2. 執行下列查詢來確認連線已加密。確認查詢傳回的 true 為 encrypt_option。

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```


設定安全性通訊協定和密碼

您可以使用資料庫參數開啟和關閉特定安全性通訊協定和密碼。您可以設定的安全性參數 (TLS 1.2 版除外) 如下表所示。

資料庫參數	允許的值 (預設值以粗體表示)	描述
rds.tls10	default、enabled、disabled	TLS 1.0。
rds.tls11	default、enabled、disabled	TLS 1.1。
rds.tls12	default	TLS 1.2。您無法修改此值。
rds.fips	0、1	<p>當您將參數設定為 1 時，RDS 會強制使用符合美國聯邦資訊處理標準 (FIPS) 140-2 標準的模組。</p> <p>如需詳細資訊，請參閱 Microsoft 文件中使用符合 FIPS 140-2 模式的 SQL Server 2016。</p>
rds.rc4	default、enabled、disabled	RC4 串流密碼。
rds.diffie-hellman	default、enabled、disabled	Diffie-Hellman 金鑰交換加密。
RDS。diffie-hellman-min-key-位長	default、1024、2048、4096	Diffie-Hellman 金鑰的最小位元長度。
rds.curve25519	default、enabled、disabled	Curve25519 橢圓曲線加密密碼。並非所有引擎版本都支援此參數。
rds.3des168	default、enabled、disabled	具有 168 位元金鑰長度的三重資料加密標準 (DES) 加密密碼。

Note

如果是在之後的次要引擎版本，則資料庫參數、`rds.tls10`、`rds.tls11`、`rds.rc4`、`rds.curve25519`、`rds.3des168`和的預設設定為停用。否則，會啟用預設設定。
對於之後的次要引擎版本而言，預設設定為 3072。`rds.diffie-hellman-min-key-bit-length`否則，預設設定為 2048。

請使用下列程序來設定安全性通訊協定和密碼：

1. 建立自訂資料庫參數群組。
2. 修改參數群組中的參數。
3. 將資料庫參數群組與您的資料庫執行個體建立關聯。

如需資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。

建立安全性相關參數群組

為對應到 SQL Server 版本和資料庫執行個體版本的安全性相關參數建立參數群組。

主控台

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在 Create parameter group (建立參數群組) 窗格中執行下列動作：
 - a. 對於 Parameter group family (參數群組家族)，請選擇 `sqlserver-se-13.0`。
 - b. 對於 Group name (群組名稱)，輸入參數群組的識別碼，例如 `sqlserver-ciphers-se-13`。
 - c. 對於 Description (說明)，輸入 **Parameter group for security protocols and ciphers**。
5. 選擇 Create (建立)。

CLI

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name sqlserver-ciphers-se-13 \  
  --db-parameter-group-family "sqlserver-se-13.0" \  
  --description "Parameter group for security protocols and ciphers"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name sqlserver-ciphers-se-13 ^  
  --db-parameter-group-family "sqlserver-se-13.0" ^  
  --description "Parameter group for security protocols and ciphers"
```

修改安全性相關參數

修改參數群組中對應至 SQL Server 版本和資料庫執行個體版本的安全性相關參數。

主控台

下列範例會修改您為 SQL Server Standard Edition 2016 建立的參數群組。此範例會關閉 TLS 1.0 版。

修改參數群組

- 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
- 在導覽窗格中，選擇 Parameter groups (參數群組)。
- 選擇參數群組，例如 sqlserver-ciphers-se-13。
- 在 Parameters (參數) 下，篩選 **rds** 的參數清單。

5. 選擇 Edit parameters (編輯參數)。
6. 選擇 rds.tls10。
7. 對於 Values (值) , 選擇 disabled。
8. 選擇儲存變更。

CLI

下列範例會修改您為 SQL Server Standard Edition 2016 建立的參數群組。此範例會關閉 TLS 1.0 版。

修改參數群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name sqlserver-ciphers-se-13 \  
  --parameters  
  "ParameterName=rds.tls10',ParameterValue=disabled',ApplyMethod=pending-reboot"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name sqlserver-ciphers-se-13 ^  
  --parameters  
  "ParameterName=rds.tls10',ParameterValue=disabled',ApplyMethod=pending-reboot"
```

將安全性相關參數群組與資料庫執行個體建立關聯

若要將參數群組與資料庫執行個體產生關聯，請使用 AWS Management Console 或 AWS CLI。

主控台

您可以將參數群組與新的或現有的資料庫執行個體建立關聯。

- 對於新的資料庫執行個體，請在啟動執行個體時建立關聯。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

- 對於現有的資料庫執行個體，請透過修改執行個體來建立關聯。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

CLI

您可以將參數群組與新的或現有的資料庫執行個體建立關聯。

使用參數群組建立資料庫執行個體

- 指定建立參數群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 13.00.5426.0.v1 \  
  --allocated-storage 100 \  
  --master-user-password secret123 \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --db-parameter-group-name sqlserver-ciphers-se-13
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 13.00.5426.0.v1 ^  
  --allocated-storage 100 ^  
  --master-user-password secret123 ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --db-parameter-group-name sqlserver-ciphers-se-13
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

修改資料庫執行個體並建立參數群組的關聯

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-parameter-group-name sqlserver-ciphers-se-13 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --db-parameter-group-name sqlserver-ciphers-se-13 ^  
  --apply-immediately
```

整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3

您可以在執行 Amazon RDS for SQL Server 的資料庫執行個體與 Amazon S3 儲存貯體之間傳輸檔案。這樣做可讓您搭配 SQL Server 功能來使用 Amazon S3，例如 BULK INSERT。例如，您可以從 Amazon S3 將 .csv、.xml、.txt 和其他檔案下載至資料庫執行個體主機，並將 D:\S3\ 中的資料匯入資料庫。所有檔案都存放在資料庫執行個體的 D:\S3\ 中。

有下列限制：

- 在多可用區域執行個體上進行容錯移轉後，會刪除備用複本上 D:\S3 資料夾中的檔案。如需詳細資訊，請參閱 [S3 整合的多可用區域限制](#)。
- 資料庫執行個體和 S3 儲存貯體必須位於相同 AWS 區域。
- 如果您一次執行多個 S3 整合任務，這些任務會循序執行，而不是並行執行。

Note

S3 整合任務與原生備份和還原任務有相同的佇列。不論何時，此佇列中最多只能有兩個任務在進行。因此，兩個執行中的原生備份和還原任務會封鎖任何 S3 整合任務。

- 在已還原的執行個體上，您必須重新啟用 S3 整合功能。S3 整合不會從來源執行個體傳播到還原的執行個體。D:\S3 中的檔案會在還原的執行個體上刪除。
- 下載至資料庫執行個體的檔案最多 100 個。換言之，D:\S3\ 中的檔案不能超過 100 個。
- 只有沒有副檔名或具有下列副檔名的檔案才支援下載：
載：.abf、.asdatabase、.bcp、.configsettings、.csv、.dat、.deploymentoptions、.deploymenttargets、.fn 和 .xmla。
- S3 儲存貯體必須擁有與相關 AWS Identity and Access Management (IAM) 角色相同的擁有者。因此，不支援跨帳戶 S3 整合。
- 另外，S3 儲存貯體不能對外公開。
- 從 RDS 上傳至 S3 的檔案大小限制為每個檔案 50 GB。
- 從 S3 下載至 RDS 的檔案大小限制為 S3 支援的最大值。

主題

- [整合 RDS for SQL Server 與 S3 的先決條件](#)
- [啟用 RDS for SQL Server 與 S3 的整合](#)
- [在 RDS for SQL Server 與 Amazon S3 之間傳輸檔案](#)

- [列出 RDS 資料庫執行個體上的檔案](#)
- [刪除 RDS 資料庫執行個體上的檔案](#)
- [監控檔案傳輸任務的狀態](#)
- [取消任務](#)
- [S3 整合的多可用區域限制](#)
- [停用 RDS for SQL Server 與 S3 的整合](#)

如需有關使用 Amazon S3 中檔案的詳細資訊，請參閱 [Amazon Simple Storage Service 入門](#)。

整合 RDS for SQL Server 與 S3 的先決條件

開始之前，請尋找或建立您要使用的 S3 儲存貯體。另外，新增許可，以允許 RDS DB 執行個體存取 S3 儲存貯體。若要設定此存取，請建立 IAM 政策和 IAM 角色。

主控台

建立 IAM 政策以存取 Amazon S3

1. 在 [IAM 管理主控台](#) 的導覽窗格中，選擇 Policies (政策)。
2. 建立新政策，並使用 Visual editor (視覺化編輯器) 索引標籤執行下列步驟。
3. 針對 Service (服務)，輸入 **S3**，然後選擇 S3 服務。
4. 針對 Actions (動作)，選擇下列選項，以授予資料庫執行個體所需的存取：
 - ListAllMyBuckets – 必要
 - ListBucket – 必要
 - GetBucketACL – 必要
 - GetBucketLocation – 必要
 - GetObject – 將 S3 中的檔案下載至 時需要 D:\S3\
 - PutObject – 將 D:\S3\ 中的檔案上傳至 S3 時需要
 - ListMultipartUploadParts – 將 D:\S3\ 中的檔案上傳至 S3 時需要
 - AbortMultipartUpload – 將 D:\S3\ 中的檔案上傳至 S3 時需要
5. 針對 Resources (資源)，顯示的選項取決於您在上一步選擇的動作。您看到的選項可能是 bucket (儲存貯體)、object (物件) 或兩者都有。對於每一項，新增適當的 Amazon Resource Name (ARN)。

針對 bucket (儲存貯體)，新增您要使用的儲存貯體的 ARN。例如，如果您的儲存貯體名稱為####，請將 ARN 設定為。arn:aws:s3:::DOC-EXAMPLE-BUCKET

針對 object (物件)，輸入儲存貯體的 ARN，然後選擇下列其中一個：

- 若要允許存取指定儲存貯體中的所有檔案，對於 Bucket name (儲存貯體) 和 Object name (物件名稱)，請都選擇 Any (任何)。
- 若要允許存取儲存貯體中的特定檔案或資料夾，請提供您要讓 SQL Server 存取的特定儲存貯體和物件的 ARN。

6. 遵循主控台的指示操作，直到完成建立政策。

以上為設定政策的簡略指南。如需建立 IAM 政策的詳細指示，請參閱 IAM 使用者指南中的[建立 IAM 政策](#)。

建立 IAM 角色以使用前一個程序中的 IAM 政策

1. 在 [IAM 管理主控台](#) 的導覽窗格中，選擇 Roles (角色)。
2. 建立新的 IAM 角色，並選擇以下出現在主控台的選項：
 - AWS 服務
 - RDS
 - RDS – Add Role to Database (RDS - 將角色新增至資料庫)

然後在底部選擇 Next:Permissions (下一步：許可)。

3. 針對 Attach permissions policies (連接許可政策)，輸入您先前建立的 IAM 政策的名稱。然後從清單中選擇該政策。
4. 遵循主控台的指示操作，直到完成建立角色。

以上為設定角色的簡略指南。如需有關建立角色的詳細指示，請參閱《IAM 使用者指南》中的[IAM 角色](#)。

AWS CLI

使用以下程序，為 Amazon RDS 授予 Amazon S3 儲存貯體的存取權：

1. 建立 IAM 政策以允許 Amazon RDS 存取 S3 儲存貯體。


```

        "s3:ListBucket",
        "s3:GetBucketACL",
        "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/key_prefix/*"
  }
]
}'

```

在 Windows 中：

務必將行尾改成您的界面所支援的字元 (^ 而不是 \)。另外，在 Windows 中，您必須以 \ 逸出所有雙引號。為了避免需要在 JSON 中逸出雙引號，您可以將 JSON 儲存為檔案，然後當作參數傳遞。

首先，建立包含下列許可政策的 `policy.json` 檔案：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketACL",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    }
  ]
}

```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/key_prefix/*"
    }
  ]
}
```

然後，使用以下命令建立政策：

```
aws iam create-policy ^
  --policy-name rds-s3-integration-policy ^
  --policy-document file://file_path/assume_role_policy.json
```

2. 政策建立後，請記下政策的 Amazon Resource Name (ARN)。後續步驟需要此 ARN。

建立 IAM 角色

- 下列 AWS CLI 命令會針對此目的建立 `rds-s3-integration-role` IAM 角色。

Example

對於LinuxmacOS、或Unix：

```
aws iam create-role \  
  --role-name rds-s3-integration-role \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
      }  
    ]  
  }'
```

```
]
}'
```

在 Windows 中：

務必將行尾改成您的界面所支援的字元 (^ 而不是 \)。另外，在 Windows 中，您必須以 \ 逸出所有雙引號。為了避免需要在 JSON 中逸出雙引號，您可以將 JSON 儲存為檔案，然後當作參數傳遞。

首先，建立包含下列政策的 `assume_role_policy.json` 檔案：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

使用以下命令建立 IAM 角色：

```
aws iam create-role ^
  --role-name rds-s3-integration-role ^
  --assume-role-policy-document file://file_path/assume_role_policy.json
```

Example 使用全域條件內容索引鍵建立 IAM 角色

建議您在資源型政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，將服務的許可限定於特定資源。這是防止[混淆代理人問題](#)最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在政策中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 Amazon 資源名稱 (ARN)。為進行 S3 整合，請務必包含資料庫執行個體 ARN，如下列範例所示。

對於LinuxmacOS、或Unix：

```
aws iam create-role \
  --role-name rds-s3-integration-role \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
          }
        }
      }
    ]
  }'
```

在 Windows 中：

將全域條件內容索引鍵新增至 `assume_role_policy.json`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```

        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringEquals": {
            "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
        }
    }
}

```

將 IAM 政策連接至 IAM 角色

- 下列 AWS CLI 命令會將原則附加至名為的角色 `rds-s3-integration-role`。將 *your-policy-arn* 替換為您前個步驟記下的政策 ARN。

Example

對於 Linux macOS、或 Unix：

```

aws iam attach-role-policy \
  --policy-arn your-policy-arn \
  --role-name rds-s3-integration-role

```

在 Windows 中：

```

aws iam attach-role-policy ^
  --policy-arn your-policy-arn ^
  --role-name rds-s3-integration-role

```

啟用 RDS for SQL Server 與 S3 的整合

在下一節，您可以找到如何啟用 Amazon S3 與 Amazon RDS for SQL Server 整合的資訊。若要使用 S3 整合，在使用 `S3_INTEGRATION feature-name` 參數之前，資料庫執行個體必須與您先前建立的 IAM 角色相關聯。

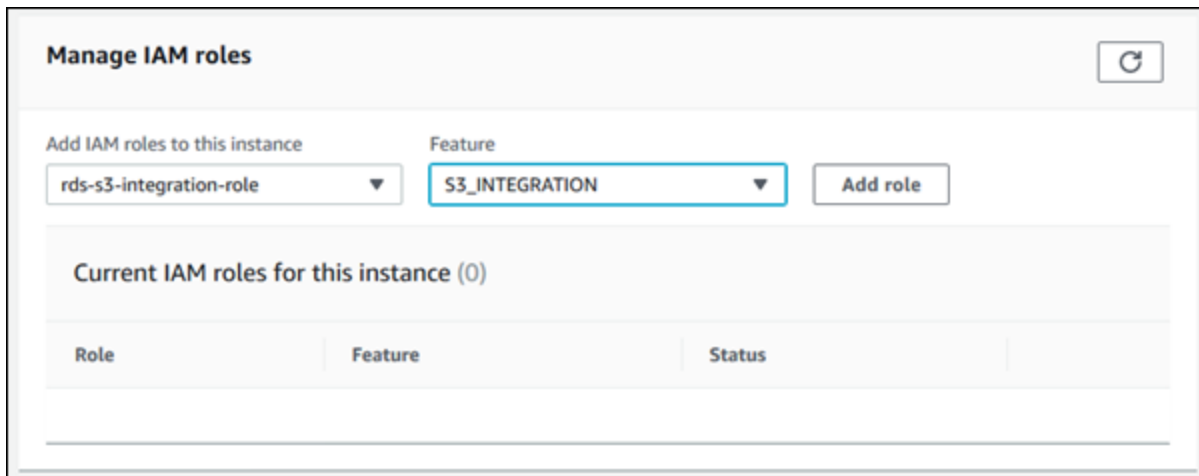
Note

若要將 IAM 角色新增至資料庫執行個體，資料庫執行個體的狀態必須為 available (可用)。

主控台

將 IAM 角色與資料庫執行個體建立關聯

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 RDS for SQL Server 資料庫執行個體名稱，以顯示其詳細資訊。
3. 在 Connectivity & security (連線能力與安全性) 索引標籤上的 Manage IAM roles (管理 IAM 角色) 區段中，選擇要在 Add IAM roles to this instance (將 IAM 角色新增至此執行個體) 下新增的 IAM 角色。
4. 針對 Feature (功能) 選擇 S3_INTEGRATION。



5. 選擇 Add role (新增角色)。

AWS CLI

將 IAM 角色新增至 RDS for SQL Server 資料庫執行個體

- 下列 AWS CLI 命令會將您的身分與存取權管理角色新增至名為的 RDS 資料庫執行個體 *mydbinstance*。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier mydbinstance \  
  --feature-name S3_INTEGRATION \  
  --role-arn your-role-arn
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --feature-name S3_INTEGRATION ^  
  --role-arn your-role-arn
```

將 *your-role-arn* 替換為您前個步驟記下的角色 ARN。S3_INTEGRATION 必須在 --feature-name 選項中指定。

在 RDS for SQL Server 與 Amazon S3 之間傳輸檔案

您可以使用 Amazon RDS 預存程序，在 Amazon S3 與 RDS 資料庫執行個體之間下載和上傳檔案。您也可以使用 Amazon RDS 預存程序，列出和刪除 RDS 執行個體上的檔案。

您自 S3 下載和上傳至 S3 的檔案會存放在 D:\S3 資料夾。您只有透過此資料夾才能存取您的檔案。您可以將檔案組織成子資料夾，在下載期間，當您包含目的地資料夾時，將為您建立這些子資料夾。

您需要提供 S3 儲存貯體和檔案的 Amazon Resource Name (ARN)，才能使用某些預存程序。ARN 的格式為 `arn:aws:s3:::DOC-EXAMPLE-BUCKET/file_name`。Amazon S3 不需要 ARN 中的帳戶號碼或 AWS 區域。

S3 整合任務循序執行，與原生備份和還原任務共同相同的佇列。不論何時，此佇列中最多只能有兩個任務在進行。可能需要五分鐘後，任務才會開始處理。

將 Amazon S3 儲存貯體中的檔案下載至 SQL Server 資料庫執行個體

若要將 S3 儲存貯體中的檔案下載至 RDS for SQL Server 資料庫執行個體，請使用 Amazon RDS 預存程序 `msdb.dbo.rds_download_from_s3` 並搭配下列參數。

參數名稱	資料類型	預設	必要	描述
@s3_arn_of_file	NVARCHAR	-	必要	要下載之檔案的 S3 ARN，例如： <code>arn:aws:s3:::DOC-EXAMPLE-BUCKET/mydata.csv</code>
@rds_file_path	NVARCHAR	-	選用	RDS 執行個體的檔案路徑。若沒有指定，則檔案路徑為 <code>D:\S3\<filename in s3></code> 。RDS 支援絕對路徑和相對路徑。如果您要建立子資料夾，請將子資料夾包含在檔案路徑中。
@overwrite_file	INT	0	選用	覆寫現有檔案： 0 = 不覆寫 1 = 覆寫

您可以下載沒有副檔名的檔案以及具有下列副檔名的檔案：`.bcp`、`.csv`、`.dat`、`.fmt`、`.info`、`.lst`、`.tbl`、`.txt` 和 `.xml`。

Note

啟用 SQL Server Integration Services 時，可支援下載具有 `.ispac` 副檔名的檔案。如需如何啟用 SSIS 的詳細資訊，請參閱 [SQL Server Integration Services](#)。

啟用 SQL Server Analysis Services 時，可支援下載具有下列副檔名的檔案：

`.abf`、`.asdatabase`、`.configsettings`、`.deploymentoptions`、`.deploymenttargets` 和 `.xmla`。如需如何啟用 SSAS 的詳細資訊，請參閱 [SQL Server Analysis Services](#)。

下列範例顯示從 S3 下載檔案的預存程序。

```
exec msdb.dbo.rds_download_from_s3
    @s3_arn_of_file='arn:aws:s3:::DOC-EXAMPLE-BUCKET/bulk_data.csv',
```

```
@rds_file_path='D:\S3\seed_data\data.csv',
@overwrite_file=1;
```

範例 `rds_download_from_s3` 操作在 `seed_data` 中建立名為 `D:\S3\` 的資料夾 (如果此資料夾尚不存在)。然後，範例從 S3 將來源檔案 `bulk_data.csv` 下載至資料庫執行個體上名為 `data.csv` 的新檔案。因為 `@overwrite_file` 參數設為 1，如果檔案已事先存在，則會覆寫檔案。

從 SQL Server 資料庫執行個體將檔案上傳至 Amazon S3 儲存貯體

若要從 RDS for SQL Server 資料庫執行個體將檔案上傳送至 S3 儲存貯體，請使用 Amazon RDS 預存程序 `msdb.dbo.rds_upload_to_s3` 並搭配下列參數。

參數名稱	資料類型	預設	必要	描述
<code>@s3_arn_of_file</code>	NVARCHAR	-	必要	要在 S3 中建立之檔案的 S3 ARN，例如： <code>arn:aws:s3:::DOC-EXAMPLE-BUCKET/mydata.csv</code>
<code>@rds_file_path</code>	NVARCHAR	-	必要	要上傳至 S3 之檔案的檔案路徑。支援絕對和相對路徑。
<code>@overwrite_file</code>	INT	-	選用	覆寫現有檔案： 0 = 不覆寫 1 = 覆寫

下列範例從指定的位置 `data.csv` 中，將名為 `D:\S3\seed_data\` 的檔案，上傳至以 ARN 指定的 S3 儲存貯體中的檔案 `new_data.csv`。

```
exec msdb.dbo.rds_upload_to_s3
  @rds_file_path='D:\S3\seed_data\data.csv',
  @s3_arn_of_file='arn:aws:s3:::DOC-EXAMPLE-BUCKET/new_data.csv',
  @overwrite_file=1;
```

因為 `@overwrite_file` 參數設為 1，如果檔案已事先存在於 S3，則會覆寫檔案。

列出 RDS 資料庫執行個體上的檔案

若要列出資料庫執行個體上存在的檔案，請使用預存程序和函數。首先，執行下列預存程序，從 D:\S3\ 中的檔案收集檔案詳細資料。

```
exec msdb.dbo.rds_gather_file_details;
```

預存程序會傳回任務的 ID。就像其他任務一樣，此預存程序以非同步方式執行。一旦任務的狀態變成 SUCCESS，您就可以在 `rds_fn_list_file_details` 函數中使用任務 ID，以列出 D:\S3\ 中現有的檔案和目錄，如下所示。

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

`rds_fn_list_file_details` 函數會傳回含有以下幾欄的表格：

輸出參數	描述
<code>filepath</code>	檔案的絕對路徑 (例如，D:\S3\mydata.csv)
<code>size_in_bytes</code>	檔案大小 (以位元組為單位)
<code>last_modified_utc</code>	上次修改日期和時間 (UTC 格式)
<code>is_directory</code>	指出項目是否為目錄的選項 (true/false)

刪除 RDS 資料庫執行個體上的檔案

若要刪除資料庫執行個體上存在的檔案，請使用 Amazon RDS 預存程序 `msdb.dbo.rds_delete_from_filesystem` 並搭配下列參數。

參數名稱	資料類型	預設	必要	描述
<code>@rds_file_path</code>	NVARCHAR	-	必要	要刪除之檔案的檔案路徑。支援絕對和相對路徑。
<code>@force_delete</code>	INT	0	選用	若要刪除目錄，必須包含此標旗並設為 1。

參數名稱	資料類型	預設	必要	描述
				1 = 刪除目錄 如果是刪除檔案，則會忽略此參數。

若要刪除目錄，@rds_file_path 的結尾必須是反斜線 (\)，@force_delete 必須設為 1。

以下範例刪除檔案 D:\S3\delete_me.txt。

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\delete_me.txt';
```

以下範例刪除目錄 D:\S3\example_folder\。

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\example_folder\' ,
    @force_delete=1;
```

監控檔案傳輸任務的狀態

若要追蹤 S3 整合任務的狀態，請呼叫 rds_fn_task_status 函數。需要兩個參數。第一個參數不適用於 S3 整合，應該一律為 NULL。第二個參數接受任務 ID。

若要查看所有任務的清單，請將第一個參數設為 NULL，將第二個參數設為 0，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

若要取得特定的任務，請將第一個參數設為 NULL，將第二個參數設為任務 ID，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

rds_fn_task_status 函數會傳回下列資訊。

輸出參數	描述
task_id	任務的 ID。

輸出參數	描述
task_type	<p>就 S3 整合而言，任務可以有下列任務類型：</p> <ul style="list-style-type: none"> • DOWNLOAD_FROM_S3 • UPLOAD_TO_S3 • LIST_FILES_ON_DISK • DELETE_FILES_ON_DISK
database_name	不適用於 S3 整合任務。
% complete	任務的進度 (以百分比表示)。
duration(mins)	任務所花的時間 (以分鐘為單位)。
lifecycle	<p>任務的狀態。可能的狀態如下：</p> <ul style="list-style-type: none"> • CREATED – 呼叫其中一個 S3 整合預存程序之後，就會建立任務，而且狀態設為 CREATED。 • IN_PROGRESS – 任務開始之後，狀態會設為 IN_PROGRESS。可能需要經過 5 分鐘，狀態才會從 CREATED 變成 IN_PROGRESS。 • SUCCESS – 任務完成之後，狀態會設為 SUCCESS。 • ERROR – 如果任務失敗，狀態會設為 ERROR。關於錯誤的詳細資訊，請查看 task_info 欄。 • CANCEL_REQUESTED – 呼叫 rds_cancel_task 之後，任務的狀態會設為 CANCEL_REQUESTED。 • CANCELLED – 成功取消任務之後，任務的狀態會設為 CANCELLED。

輸出參數	描述
task_info	任務的其他資訊。如果處理期間發生錯誤，此欄包含該錯誤的相關資訊。
last_updated	上次更新任務狀態的日期和時間。
created_at	建立任務的日期和時間。
S3_object_arn	下載或上傳時往返之 S3 物件的 ARN。
overwrite_S3_backup_file	不適用於 S3 整合任務。
KMS_master_key_arn	不適用於 S3 整合任務。
filepath	RDS 資料庫執行個體上的檔案路徑。
overwrite_file	指出是否覆寫現存檔案的選項。
task_metadata	不適用於 S3 整合任務。

取消任務

若要取消 S3 整合任務，請使用 `msdb.dbo.rds_cancel_task` 預存程序並搭配 `task_id` 參數。進行中的刪除和列出任務無法取消。以下範例顯示請求取消任務。

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

若要綜觀所有任務及其任務 ID，請使用 `rds_fn_task_status` 函數，如 [監控檔案傳輸任務的狀態](#) 中所述。

S3 整合的多可用區域限制

在多可用區域執行個體上，備用複本上 D:\S3 資料夾中的檔案會在容錯移轉後刪除。容錯移轉可能是計劃中的，例如在修改資料庫執行個體期間變更執行個體類別或升級引擎版本。容錯移轉也可能是在主要伺服器中斷期間意外發生的。

Note

我們不建議您使用 D:\S3 資料夾來儲存檔案。最佳實務是將建立的檔案上傳至 Amazon S3 使檔案能夠耐久，並在需要匯入資料時下載檔案。

若要判斷上次容錯移轉時間，您可以使用 `msdb.dbo.rds_failover_time` 預存程序。如需更多詳細資訊，請參閱 [判斷上次容錯移轉時間](#)。

Example 沒有最近的容錯移轉

此範例顯示錯誤日誌中沒有最近的容錯移轉時的輸出。自從 2020-04-29 23:59:00.01 以來，沒有發生任何容錯移轉。

因此，在那段時間之後下載且尚未使用 `rds_delete_from_filesystem` 預存程序刪除的所有檔案仍然可以在目前的主機上存取。在那段時間之前下載的檔案也可能可用。

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example 最近的容錯移轉

此範例顯示錯誤日誌中有容錯移轉時的輸出。最近一次的容錯移轉是在 2020-05-05 18:57:51.89。

在那段時間之後下載且尚未使用 `rds_delete_from_filesystem` 預存程序刪除的所有檔案仍然可以在目前的主機上存取。

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

停用 RDS for SQL Server 與 S3 的整合

接下來，您可以找到如何停用 Amazon S3 與 Amazon RDS for SQL Server 整合的資訊。停用 S3 整合時，不會刪除 D:\S3\ 中的檔案。

Note

若要從資料庫執行個體中移除 IAM 角色，資料庫執行個體的状态必須為 `available`。

主控台

取消 IAM 角色與資料庫執行個體的關聯

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 RDS for SQL Server 資料庫執行個體名稱，以顯示其詳細資訊。
3. 在 Connectivity & security (連線能力與安全性) 索引標籤上的 Manage IAM roles (管理 IAM 角色) 區段中，選擇要移除的 IAM 角色。
4. 選擇刪除。

AWS CLI

從 RDS for SQL Server 資料庫執行個體中移除 IAM 角色

- 下列 AWS CLI 命令會從名為的 RDS 版 SQL 伺服器資料庫執行個體中移除 IAM 角色 `mydbinstance`。

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-role-from-db-instance \  
  --db-instance-identifier mydbinstance \  
  --feature-name S3_INTEGRATION \  
  --role-arn your-role-arn
```

在 Windows 中：

```
aws rds remove-role-from-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --feature-name S3_INTEGRATION ^  
  --role-arn your-role-arn
```

根據 *your-role-arn* 選項，將 `--feature-name` 換成適當的 IAM 角色 ARN。

在 Amazon RDS for SQL Server 上使用 Database Mail

您可以在 SQL Server 資料庫執行個體上使用 Database Mail，從您的 Amazon RDS 傳送電子郵件訊息給使用者。訊息可包含檔案和查詢結果。Database Mail 包含下列元件：

- 設定和安全性物件 – 這些物件會建立設定檔和帳戶，並儲存在 msdb 資料庫中。
- 傳訊物件 – 這些物件包括用來傳送訊息的 [sp_send_dbmail](#) 預存程序，以及保留訊息相關資訊的資料結構。這些物件會儲存在 msdb 資料庫中。
- 記錄和稽核物件 – Database Mail 會將記錄資訊寫入 msdb 資料庫和 Microsoft Windows 應用程式事件記錄檔。
- Database Mail 可執行檔 – DatabaseMail.exe 會從 msdb 資料庫中的佇列讀取，並傳送電子郵件訊息。

RDS 支援 Web、標準和企業版上的所有 SQL Server 版本的 Database Mail。

限制

在 SQL Server 資料庫執行個體上使用 Database Mail 時將適用下列限制：

- Database Mail 不支援 SQL Server Express 版本。
- 不支援修改 Database Mail 組態參數。若要查看預先設定的值 (預設值)，請使用 [sysmail_help_configure_sp](#) 預存程序。
- 不完全支援檔案附件。如需更多詳細資訊，請參閱 [使用檔案附件](#)。
- 檔案附件大小上限為 1 MB。
- Database Mail 需要在多可用區域資料庫執行個體上設定其他組態。如需更多詳細資訊，請參閱 [異地同步備份部署的考量](#)。
- 不支援將 SQL Server 代理程式設定為傳送電子郵件訊息給預先定義的運算子。

啟用 Database Mail

請使用下列程序啟用資料庫執行個體的 Database Mail：

1. 建立新的參數群組。
2. 修改參數群組，將 database mail xps 參數設定為 1。
3. 將參數群組與資料庫執行個體建立關聯。

建立 Database Mail 的參數群組

為對應到 SQL Server 版本和資料庫執行個體版本的 database mail xps 參數建立參數群組。

Note

您也可以修改現有的參數群組。請遵循 [修改啟用 Database Mail 的參數](#) 中的程序。

主控台

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在 Create parameter group (建立參數群組) 窗格中執行下列動作：
 - a. 對於 Parameter group family (參數群組家族)，請選擇 sqlserver-se-13.0。
 - b. 對於 Group name (群組名稱)，輸入參數群組的識別碼，例如 **dbmail-sqlserver-se-13**。
 - c. 對於 Description (說明)，輸入 **Database Mail XPs**。
5. 選擇 Create (建立)。

CLI

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name dbmail-sqlserver-se-13 \  
  --db-parameter-group-family "sqlserver-se-13.0" \  
  --description "Database Mail XPs"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name dbmail-sqlserver-se-13 ^  
  --db-parameter-group-family "sqlserver-se-13.0" ^  
  --description "Database Mail XPs"
```

修改啟用 Database Mail 的參數

修改參數群組中對應至 SQL Server 版本和資料庫執行個體版本的 database mail xps 參數。

若要啟用 Database Mail，請將 database mail xps 參數設定為 1。

主控台

下列範例會修改您為 SQL Server Standard Edition 2016 所建立的參數群組。

修改參數群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇參數群組，例如 dbmail-sqlserver-se-13。
4. 在 Parameters (參數) 下，篩選 **mail** 的參數清單。
5. 選擇 database mail xps。
6. 選擇 Edit parameters (編輯參數)。
7. 輸入 **1**。
8. 選擇儲存變更。

CLI

下列範例會修改您為 SQL Server Standard Edition 2016 所建立的參數群組。

修改參數群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name dbmail-sqlserver-se-13 \  
  --parameters "ParameterName='database mail  
xps',ParameterValue=1,ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name dbmail-sqlserver-se-13 ^  
  --parameters "ParameterName='database mail  
xps',ParameterValue=1,ApplyMethod=immediate"
```

將參數群組與資料庫執行個體建立關聯

您可以使用 AWS Management Console 或將 Database Mail 參數群組與資料庫執行個體產生關聯。

AWS CLI

主控台

您可以將 Database Mail 參數群組與新的或現有的資料庫執行個體建立關聯。

- 對於新的資料庫執行個體，請在啟動執行個體時建立關聯。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請透過修改執行個體來建立關聯。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

CLI

您可以將 Database Mail 參數群組與新的或現有的資料庫執行個體建立關聯。

使用 Database Mail 參數群組建立資料庫執行個體

- 指定建立參數群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 13.00.5426.0.v1 \  
  --allocated-storage 100 \  
  --manage-master-user-password \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --db-parameter-group-name dbmail-sqlserver-se-13
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 13.00.5426.0.v1 ^  
  --allocated-storage 100 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --db-parameter-group-name dbmail-sqlserver-se-13
```

修改資料庫執行個體並關聯 Database Mail 參數群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-parameter-group-name dbmail-sqlserver-se-13 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --db-parameter-group-name dbmail-sqlserver-se-13 ^  
  --apply-immediately
```

設定 Database Mail

您可以執行下列任務來設定 Database Mail：

1. 建立 Database Mail 設定檔。
2. 建立 Database Mail 帳戶。
3. 將 Database Mail 帳戶新增至 Database Mail 設定檔。
4. 將使用者新增至 Database Mail 設定檔。

Note

若要設定 Database Mail，請確認您具有 `execute` 資料庫中預存程序的 `msdb` 許可。

建立 Database Mail 設定檔

若要建立 Database Mail 設定檔，您可以使用 [sysmail_add_sp](#) 預存程序。以下範例會建立名為 `Notifications` 的設定檔。

建立設定檔

- 使用下列 SQL 陳述式。

```
USE msdb  
GO
```



```
EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name          = 'Notifications',
    @description           = 'Profile used for sending outgoing notifications using
Amazon SES.';
GO
```

建立 Database Mail 帳戶

若要建立 Database Mail 帳戶，您可以使用 [sysmail_add_account_sp](#) 預存程序。下列範例使用 Amazon Simple Email Service，在私有 VPC 中的 RDS for SQL Server 資料庫執行個體上建立一個名為 SES 的帳戶。

使用 Amazon SES 需要下列參數：

- @email_address – 一個 Amazon SES 驗證身分。如需詳細資訊，請參閱[在 Amazon SES 中驗證身分](#)。
- @mailserver_name – 一個 Amazon SES SMTP 端點。如需詳細資訊，請參閱[連線至 Amazon SES SMTP 端點](#)。
- @username – 一個 Amazon SES SMTP 使用者名稱。如需詳細資訊，請參閱[取得 Amazon SES SMTP 憑證](#)。

請勿使用使用 AWS Identity and Access Management 者名稱。

- @password – 一個 Amazon SES SMTP 密碼。如需詳細資訊，請參閱[取得 Amazon SES SMTP 憑證](#)。

建立帳戶

- 使用下列 SQL 陳述式。

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name          = 'SES',
    @description           = 'Mail account for sending outgoing notifications.',
    @email_address         = 'nobody@example.com',
    @display_name          = 'Automated Mailer',
```

```
@mailserver_name      = 'vpce-0a1b2c3d4e5f-01234567.email-smtp.us-  
west-2.vpce.amazonaws.com',  
@port                 = 587,  
@enable_ssl           = 1,  
@username              = 'Smtp_Username',  
@password              = 'Smtp_Password';  
GO
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

將 Database Mail 帳戶新增至 Database Mail 設定檔

若要將 Database Mail 帳戶新增至 Database Mail 設定檔，您可以使用 [sysmail_add_profileaccount_sp](#) 預存程序。下列範例會將 SES 帳戶新增至 Notifications 設定檔。

將帳戶新增至設定檔

- 使用下列 SQL 陳述式。

```
USE msdb  
GO  
  
EXECUTE msdb.dbo.sysmail_add_profileaccount_sp  
    @profile_name      = 'Notifications',  
    @account_name      = 'SES',  
    @sequence_number   = 1;  
GO
```

將使用者新增至 Database Mail 設定檔

若要授與 msdb 資料庫主體使用 Database Mail 設定檔的許可，您可以使用 [sysmail_add_principalprofile_sp](#) 預存程序。主體是可以要求 SQL Server 資源的實體。資料庫主體必須對應至 SQL Server 身分驗證使用者、Windows 身分驗證使用者或 Windows 身分驗證群組。

下列範例會授與 Notifications 設定檔的公開存取權。

將使用者新增至設定檔

- 使用下列 SQL 陳述式。

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name      = 'Notifications',
    @principal_name    = 'public',
    @is_default        = 1;
GO
```

適用於 Database Mail 的 Amazon RDS 預存程序和函數

Microsoft 提供使用 Database Mail 的 [預存程序](#)，例如建立、列出、更新和刪除帳號和設定檔。此外，RDS 還提供適用於 Database Mail 的預存程序和函數，如下表所示。

程序/函數	描述
rds_fn_sysmail_allitems	顯示已傳送的訊息，包括其他使用者提交的訊息。
rds_fn_sysmail_event_log	顯示事件，包括其他使用者提交訊息的事件。
rds_fn_sysmail_mailattachments	顯示附件，包括其他使用者提交訊息的附件。
rds_sysmail_control	啟動和停止郵件佇列 (DatabaseMail.exe 處理程序)。
rds_sysmail_delete_mailitem_s_sp	從 Database Mail 內部表格刪除所有使用者傳送的電子郵件訊息。

使用 Database Mail 傳送電子郵件訊息

您可以使用 [sp_send_dbmail](#) 預存程序來利用 Database Mail 傳送電子郵件訊息。

用量

```
EXEC msdb.dbo.sp_send_dbmail
```

```
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0|1];
```

下列是必要參數：

- @profile_name – 要從中傳送訊息的 Database Mail 設定檔名稱。
- @recipients – 要傳送訊息的電子郵件地址的清單 (以分號分隔)。
- @subject – 訊息的主旨。
- @body – 訊息的內文。您也可以使用已宣告的變數作為內文。

下列是選用參數：

- @body_format – 此參數與已宣告的變數一起使用，以 HTML 格式傳送電子郵件。
- @file_attachments – 訊息附件清單 (以分號分隔)。檔案路徑必須是絕對路徑。
- @query – 要執行的 SQL 查詢。查詢結果可以作為檔案貼加，也可以包含在訊息內文中。
- @attach_query_result_as_file – 是否將查詢結果附加為檔案。設定為 0 表示否，1 表示是。預設值為 0。

範例

下列範例示範如何傳送電子郵件訊息。

Example 傳送訊息給單一收件人

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Automated DBMail message - 1',
    @body              = 'Database Mail configuration was successful.';
```

```
GO
```

Example 傳送訊息給多位收件人

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'recipient1@example.com;recipient2@example.com',
    @subject           = 'Automated DBMail message - 2',
    @body              = 'This is a message.';

GO
```

Example 將 SQL 查詢結果作為檔案附件傳送

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Test SQL query',
    @body              = 'This is a SQL query test.',
    @query             = 'SELECT * FROM abc.dbo.test',
    @attach_query_result_as_file = 1;

GO
```

Example 以 HTML 格式傳送訊息

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> </br> See the report. <b>
Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Test HTML message',
    @body              = @HTML_Body,
    @body_format       = 'HTML';
```

```
GO
```

Example 在資料庫中發生特定事件時使用觸發器傳送訊息

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
DROP TRIGGER Purchasing.iProductNotification
GO

CREATE TRIGGER iProductNotification ON Production.Product
FOR INSERT
AS
DECLARE @ProductInformation nvarchar(255);
SELECT
@ProductInformation = 'A new product, ' + Name + ', is now available for $' +
CAST(StandardCost AS nvarchar(20)) + '!
FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
@profile_name      = 'Notifications',
@recipients        = 'nobody@example.com',
@subject           = 'New product information',
@body              = @ProductInformation;
GO
```

檢視郵件、記錄檔和附件

您可以使用 RDS 預存程序來檢視訊息、事件記錄檔和附件。

檢視所有電子郵件訊息

- 使用下列 SQL 查詢：

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or
'failed' or 'unsent'
```

檢視所有電子郵件事件記錄檔

- 使用下列 SQL 查詢：

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

檢視所有電子郵件附件

- 使用下列 SQL 查詢：

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

刪除訊息

您可以使用 `rds_sysmail_delete_mailitems_sp` 預存程序來刪除訊息。

Note

當 DBMail 歷程記錄資料大小達到 1 GB，且保留期限至少為 24 小時，RDS 會自動刪除郵件表格項目。

如果您想要將郵件項目保留一段較長的時間，可以將其封存。如需詳細資訊，請參閱 Microsoft 文件中的 [建立 SQL Server Agent 作業以封存 Database Mail 訊息及事件記錄檔](#)。

刪除所有電子郵件訊息

- 使用下列 SQL 陳述式。

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;
GO
```

刪除所有具有特定狀態的電子郵件

- 使用下列 SQL 陳述式刪除所有失敗的訊息。

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';
```

```
GO
```

啟動郵件佇列

您可以使用 `rds_sysmail_control` 預存程序來啟動 Database Mail 處理程序。

Note

啟用 Database Mail 會自動啟動郵件佇列。

啟動郵件佇列

- 使用下列 SQL 陳述式。

```
EXECUTE msdb.dbo.rds_sysmail_control start;  
GO
```

停用郵件佇列

您可以使用 `rds_sysmail_control` 預存程序來停止 Database Mail 處理程序。

停止郵件佇列

- 使用下列 SQL 陳述式。

```
EXECUTE msdb.dbo.rds_sysmail_control stop;  
GO
```

使用檔案附件

在 SQL Server 上，來自 RDS 的 Database Mail 訊息不支援下列檔案附件副檔

名：.ade、.adp、.apk、.appx、.appxbundle、.bat、.bak、.cab、.chm、.cmd、.com、.cpl、.dll、.dmg、.ex
和 .wsh。

Database Mail 會使用目前使用者的 Microsoft Windows 安全性內容來控制檔案的存取。使用 SQL Server 身分驗證登入的使用者無法使用 `@file_attachments` 預存程序的 `sp_send_dbmail` 參數附

加檔案。Windows 不允許 SQL Server 將登入資料從遠端電腦提供給另一台遠端電腦。因此，從非執行 SQL Server 的電腦執行命令時，Database Mail 無法從網路共用附加檔案。

不過，您可以使用 SQL Server 代理程式任務來附加檔案。如需有關 SQL Server 代理程式的詳細資訊，請參閱 [使用 SQL Server 代理程式](#) 和 Microsoft 文件中的 [SQL Server 代理程式](#)。

異地同步備份部署的考量

當您在多可用區域資料庫執行個體上設定 Database Mail 時，組態不會自動傳播至次要可用區。建議將多可用區域執行個體轉換為單一可用區執行個體、設定 Database Mail，然後將資料庫執行個體轉換回多可用區域。然後，主節點和次要節點均設有 Database Mail 組態。

如果您從已設定 Database Mail 的多可用區域執行個體建立僅供讀取複本，則複本會繼承該組態，但不需要 SMTP 伺服器的密碼。使用密碼更新 Database Mail 帳戶。

移除 SMTP (通訊埠 25) 限制

根據預設，會針對 SQL Server 資料庫執行個體的 RDS AWS 封鎖 SMTP (連接埠 25) 上的輸出流量。這樣做是為了根據 elastic network interface 擁有者的原則來防止垃圾郵件。如果需要，您可以刪除此限制。如需詳細資訊，請參閱 [如何從 Amazon EC2 執行個體或 Lambda 函數移除連接埠 25 的限制？](#)

。

執行個體存放區支援 Amazon RDS for SQL Server 上的 tempdb 資料庫

執行個體存放區為資料庫執行個體提供暫時的區塊層級儲存空間。這個儲存空間位於實際連接到主機電腦的磁碟上。這些磁碟具有以固態硬碟 (SSD) 為基礎的非揮發性記憶體儲存裝置 (NVMe) 執行個體儲存體。此儲存體已針對低延遲、非常高的隨機 I/O 效能，以及高循序讀取輸送量進行最佳化。

將 tempdb 資料檔案和 tempdb 日誌檔案放置在執行個體存放區上，與以 Amazon EBS 為基礎的標準儲存相比，您可以達成更低的讀取和寫入延遲。

Note

SQL Server 資料庫檔案和資料庫日誌檔案不會放置在執行個體存放區。

啟用執行個體存放區

當 RDS 佈建具有下列其中一種執行個體類別的資料庫執行個體時，tempdb 資料庫會自動放置到執行個體存放區中：

- db.m5d
- db.r5d
- db.x2iedn

若要啟用執行個體存放區，請執行下列其中一項動作：

- 使用這些執行個體類型之一建立 SQL Server 資料庫執行個體。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改現有的 SQL Server 資料庫執行個體，以使用其中一個執行個體。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

執行個體存放區適用於支援一或多個這些執行個體類型的所有 AWS 區域。如需 db.m5d 和 db.r5d 執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。如需 Amazon RDS for SQL Server 支援之執行個體類別的詳細資訊，請參閱 [Microsoft SQL Server 的資料庫執行個體類別支援](#)。

檔案位置和大小考量

在沒有執行個體存放區的執行個體上，RDS 會將 tempdb 資料和日誌檔案存放在 D:\rdsdbdata\DATA 目錄中。依預設，這兩個檔案都會從 8 MB 開始。

在具有執行個體存放區的執行個體上，RDS 會將 tempdb 資料和日誌檔案存放在 T:\rdsdbdata\DATA 目錄中。

當 tempdb 只有一個資料檔案 (tempdb.mdf) 和一個日誌檔案 (templog.ldf) 時，templog.ldf 預設從 8 MB 開始，且 tempdb.mdf 從執行個體儲存體容量的 80% 或以上開始。20% 的儲存容量或 200 GB (以較少者為準) 可隨時開始。多個 tempdb 資料檔案會平均分割 80% 的磁碟空間，而日誌檔案的初始大小一律為 8 MB。

例如，如果您將資料庫執行個體類別從 db.m5.2xlarge 修改為 db.m5d.2xlarge，大小為 tempdb 的資料檔案將從每個 8 MB 增加至 234 GB。

Note

除了執行個體存放區上的 tempdb 資料和日誌檔案 (T:\rdsdbdata\DATA)，您仍然可以在資料磁碟區建立額外的 tempdb 資料和日誌檔案 (D:\rdsdbdata\DATA)。這些檔案的初始大小一律為 8 MB。

備份考量

您可能需要長時間保留備份，從而產生一段時間的成本。tempdb 資料和日誌區塊可能會經常變更，視工作負載而定。這可以大大增加資料庫快照大小。

tempdb 在執行個體存放區時，快照不包含暫存檔案。這表示相較於僅限 EBS 的儲存體，快照大小較小，而且耗用的免費備份配置較少。

磁碟已滿錯誤

如果您使用執行個體存放區中的所有可用空間，可能會收到下列錯誤：

- 資料庫 'tempdb' 的交易日誌因 'ACTIVE_TRANSACTION' 已滿。
- 無法在資料庫 'tempdb' 中為物件 'dbo.SORT temporary run storage: 140738941419520' 配置空間，因為 'PRIMARY' 檔案群組已滿。藉由刪除不需要的檔案、卸除檔案群組中的物件、將其他檔案新增至檔案群組，或為檔案群組中的現有檔案設定自動成長，以建立磁碟空間。

執行個體存放區已滿時，您可以執行下列一或多項動作：

- 調整您的工作負載或使用 tempdb 的方式。
- 擴大以使用具有更多 NVMe 儲存體的資料庫執行個體類別。

- 停止使用執行個體存放區，並使用只有 EBS 存儲體的執行個體類別。
- 在 EBS 磁碟區上新增 tempdb 次要資料或日誌檔案，以使用混合模式。

移除執行個體存放區

若要移除執行個體儲存體，請修改您的 SQL Server 資料庫執行個體，以使用不支援執行個體儲存體的執行個體類型，例如 db.m5、db.r5 或 db.x1e。

Note

當您移除執行個體存放區時，暫存檔案會移至 D:\rdsdbdata\DATA 目錄，並將大小減少為 8 MB。

搭配 Amazon RDS for Microsoft SQL Server 使用擴充事件

您可以在 Microsoft SQL Server 中使用擴充事件來擷取 Amazon RDS for SQL Server 的偵錯和疑難排解資訊。擴充事件會取代 SQL Trace 和 Server Profiler，其已被 Microsoft 取代。擴充事件類似於分析工具追蹤，但對所追蹤的事件具有更細微的控制。擴充事件在 Amazon RDS 上支援 SQL Server 2014 版及更新版本。如需詳細資訊，請參閱 Microsoft 文件中的[擴充事件概觀](#)。

對於在 Amazon RDS for SQL Server 中具有主要使用者權限的使用者，會自動開啟擴充事件。

主題

- [限制與建議](#)
- [設定 RDS for SQL Server 上的擴充事件](#)
- [異地同步備份部署的考量](#)
- [查詢擴充的事件檔案](#)

限制與建議

在 RDS for SQL Server 上使用擴充事件時，會套用下列限制：

- 擴充事件僅支援 Enterprise 和 Standard 版本。
- 您無法變更預設的擴充事件工作階段。
- 請務必將工作階段記憶體分區模式設定為 NONE。
- 工作階段事件保留模式可以是 ALLOW_SINGLE_EVENT_LOSS 或 ALLOW_MULTIPLE_EVENT_LOSS。
- 不支援 Windows 事件追蹤 (ETW) 目標。
- 請確定檔案目標位於 D:\rdsdbdata\log 目錄中。
- 對於配對相符目標，請將 respond_to_memory_pressure 屬性設定為 1。
- 信號緩衝區目標記憶體不能大於 4 MB。
- 不支援下列動作：
 - debug_break
 - create_dump_all_threads
 - create_dump_single_threads
- 下列版本及更新版本支援此 rpc_completed 事件：
15.0.4083.2、14.0.3370.1、13.0.5865.1、12.0.6433.1、11.0.7507.2。

設定 RDS for SQL Server 上的擴充事件

在 RDS for SQL Server 上，您可以設定擴充事件工作階段的特定參數值。下表描述了可設定的參數。

參數名稱	描述
<code>xe_session_max_memory</code>	指定要配置給工作階段用於事件緩衝的記憶體數量上限。
<code>xe_session_max_event_size</code>	指定大型事件允許的記憶體大小上限。此值對應於事件工作階段。
<code>xe_session_max_dispatch_latency</code>	指定傳送至擴充事件工作階段目標之前，事件在記憶體中緩衝的時間。此值對應於目標的 <code>max_dispatch_latency</code> 設定。
<code>xe_file_target_size</code>	指定檔案目標的大小上限。此值對應於檔案目標的 <code>max_file_size</code> 設定。
<code>xe_file_retention</code>	指定事件工作階段之檔案目標產生的檔案保留時間 (以天為單位)。

Note

將 `xe_file_retention` 設定為零會在 SQL Server 釋放這些檔案的鎖定之後自動移除 .xel 檔案。每當 .xel 檔案達到 `xe_file_target_size` 中設定的大小限制時，就會釋放鎖定。

您可以使用 `rdsadmin.dbo.rds_show_configuration` 存放的程序來顯示這些參數的目前值。例如，使用下列 SQL 陳述式來檢視 `xe_session_max_memory` 的目前設定。

```
exec rdsadmin.dbo.rds_show_configuration 'xe_session_max_memory'
```

您可以使用 `rdsadmin.dbo.rds_set_configuration` 存放的程序來對其進行修改。例如，使用下列 SQL 陳述式設定 `xe_session_max_memory` 為 4 MB。

```
exec rdsadmin.dbo.rds_set_configuration 'xe_session_max_memory', 4
```

異地同步備份部署的考量

當您在主要資料庫執行個體上建立擴充事件工作階段時，它不會傳播到待命複本。您可以容錯移轉並在新的主要資料庫執行個體上建立擴充事件工作階段。或者，您可以移除並重新新增異地同步備份組態，將擴充事件工作階段傳播到待命複本。RDS 會停止待命複本上的所有非預設擴充事件工作階段，讓這

些工作階段不會耗用待命的資源。因此，待命複本成為主要資料庫執行個體之後，請務必手動啟動新主要資料庫執行個體上的擴充事件工作階段。

Note

這種方法同時適用於「Always On 可用性群組」和「資料庫鏡像」。

您也可以使用 SQL Server 代理程式任務來追蹤待命複本，並在待命成為主要資料庫執行個體時啟動工作階段。例如，在 SQL Server 代理程式任務步驟中使用下列查詢，以重新啟動主要資料庫執行個體上的事件工作階段。

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
        AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
        AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
            DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
    END
END
```

如果這些工作階段處於停止狀態，此查詢會在主要資料庫執行個體上重新啟動事件工作階段 xe1 和 xe2。您也可以將具有便利間隔的排程新增至此查詢。

查詢擴充的事件檔案

您可以使用 SQL Server Management Studio 或 `sys.fn_xe_file_target_read_file` 函數來檢視使用檔案目標之擴充事件的資料。如需有關此函數的詳細資訊，請參閱 Microsoft 文件中的 [sys.fn_xe_file_target_read_file \(Transact-SQL\)](#)。

擴充事件檔案目標只能將檔案寫入 RDS for SQL Server 上的 `D:\rdsdbdata\log` 目錄。

例如，使用下列 SQL 查詢列出名稱以 `xe` 開頭之擴充事件工作階段之所有檔案的內容。

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```


使用 RDS for SQL Server 存取交易日誌備份

藉由取 RDS for SQL Server 的交易日誌備份，您可以列出資料庫的交易日誌備份檔案，並將其複製到目標 Amazon S3 儲存貯體。透過在 Amazon S3 儲存貯體中複製交易日誌備份，您可以結合完整和差異資料庫備份使用它們，以執行時間點資料庫還原。您可以使用 RDS 預存程序來設定交易日誌備份的存取、列出可用的交易日誌備份，以及將它們複製到 Amazon S3 儲存貯體。

存取交易日誌備份可提供下列功能和優點：

- 列出和檢視 RDS for SQL Server 資料庫執行個體上資料庫之可用交易日誌備份的中繼資料。
- 將可用的交易日誌備份從 RDS for SQL Server 複製到目標 Amazon S3 儲存貯體。
- 執行資料庫還 point-in-time 原，無需還原整個資料庫執行個體。如需將資料庫執行個體還原至某個時間點的詳細資訊，請參閱 [將資料庫執行個體還原至指定的時間](#)。

可用性與支援

所有 AWS 區域都支援存取交易記錄備份。Amazon RDS 上支援的所有 Microsoft SQL Server 版本都可以存取交易日誌備份。

要求

必須符合下列需求，才能啟用對交易日誌備份的存取：

- 必須在資料庫執行個體上啟用自動備份，且備份保留必須設為一或多天的值。如需啟用自動備份和設定保留政策的詳細資訊，請參閱 [啟用自動備份](#)。
- Amazon S3 儲存貯體與來源資料庫執行個體必須存在於相同的帳戶和區域中。在啟用對交易日誌備份的存取之前，請選擇現有的 Amazon S3 儲存貯體，或 [建立新的儲存貯體](#)，用於交易日誌備份檔案。
- Amazon S3 儲存貯體許可政策必須設定如下，才能允許 Amazon RDS 將交易日誌檔複製到其中：
 1. 將儲存貯體上的物件帳戶擁有權屬性設為 Bucket Owner Preferred (偏好的儲存貯體擁有者)。
 2. 新增下列政策。根據預設，將沒有任何政策，因此請使用儲存貯體存取控制清單 (ACL) 來編輯儲存貯體政策並加以新增。

下列範例使用 ARN 來指定資源。建議您在資源型信任關係中使用 SourceArn 和 SourceAccount 全域條件內容金鑰，將服務的許可限定於特定資來源。如需使用 ARN 的詳細資訊，請參閱 [Amazon Resource Name \(ARN\)](#) 和 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

Example 用於存取交易日誌備份的 Amazon S3 許可政策範例

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Only allow writes to my bucket with bucket owner full control",
      "Effect": "Allow",
      "Principal": {
        "Service": "backups.rds.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/{customer_path}/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:sourceAccount": "{customer_account}",
          "aws:sourceArn": "{db_instance_arn}"
        }
      }
    }
  ]
}
```

- 存取 Amazon S3 儲存貯體的 AWS Identity and Access Management (IAM) 角色。如果您已經有 IAM 角色，則可直接使用。當您使用 AWS Management Console 新增 SQLSERVER_BACKUP_RESTORE 選項時，可以選擇替您建立新的 IAM 角色。或者，您也可以手動建立新的。如需使用 SQLSERVER_BACKUP_RESTORE 建立和設定 IAM 角色的詳細資訊，請參閱 [手動建立原生備份與還原的 IAM 角色](#)。
- SQLSERVER_BACKUP_RESTORE 選項必須新增至資料庫執行個體上的選項群組。如需新增 SQLSERVER_BACKUP_RESTORE 選項的詳細資訊，請參閱 [SQL Server 對原生備份與還原的支援](#)。

Note

如果您的資料庫執行個體已啟用儲存加密，則必須在原生備份和還原選項群組中提供的 IAM 角色中提供 AWS KMS (KMS) 動作和金鑰。

或者，如果您打算使用 `rds_restore_log` 預存程序執行時間點資料庫還原，建議將相同的 Amazon S3 路徑用於原生備份和還原選項群組，以及存取交易日誌備份。此方法可確保 Amazon RDS 擔任選項群組中的角色來執行還原日誌功能時，可以從相同的 Amazon S3 路徑中擷取交易日誌備份。

- 如果資料庫執行個體已加密，無論加密類型為何 (AWS 受管金鑰或客戶受管金鑰)，您都必須在 IAM 角色和 `rds_tlog_backup_copy_to_S3` 預存程序中提供客戶受管 KMS 金鑰。

限制與建議

存取交易日誌備份具有下列限制和建議：

- 對於備份保留設定為 1 到 35 天的任何資料庫執行個體，您最多可以列出並複製最後七天的交易日誌備份。
- 用於存取交易日誌備份的 Amazon S3 儲存貯體與來源資料庫執行個體必須存在於相同的帳戶和區域中。不支援跨帳戶與跨區域複本。
- 只能將一個 Amazon S3 儲存貯體設定為將交易日誌備份複製到的目標。您可以使用 `rds_tlog_copy_setup` 預存程序選擇新的目標 Amazon S3 儲存貯體。如需選擇新目標 Amazon S3 儲存貯體的詳細資訊，請參閱 [設定交易日誌備份的存取權](#)。
- 如果未啟用您的 RDS 執行個體進行儲存體加密，則無法在使用 `rds_tlog_backup_copy_to_S3` 預存程序時指定 KMS 金鑰。
- 不支援多帳戶複製。用於複製的 IAM 角色只允許資料庫執行個體擁有者帳戶內 Amazon S3 儲存貯體的寫入存取權。
- 只有兩個任何類型的並行任務才能在 RDS for SQL Server 資料庫執行個體上執行。
- 在指定時間內，只能針對單一資料庫執行一個複製任務。如果您想要複製資料庫執行個體上多個資料庫的交易日誌備份，請針對每個資料庫使用個別的複製任務。
- 如果您複製已存在於 Amazon S3 儲存貯體且名稱相同的交易日誌備份，將覆寫現有的交易日誌備份。
- 您只能執行隨附的預存程序，存取主要資料庫執行個體上的交易日誌備份。您無法在 RDS for SQL Server 僅供讀取複本上，也無法在多可用區域資料庫叢集的次要執行個體上執行這些預存程序。
- 如果在 `rds_tlog_backup_copy_to_S3` 預存程序執行期間重新啟動 RDS for SQL Server 資料庫執行個體，則在資料庫執行個體再次上線時，任務將自動從頭開始重新啟動。在重新開機之前執行任務時，已複製到 Amazon S3 儲存貯體的任何交易日誌備份都會遭到覆寫。
- 無法設定 Microsoft SQL Server 系統資料庫和 RDSAdmin 資料庫，進行交易日誌備份存取。

- 不支援複製到 SSE-KMS 加密的儲存貯體。

設定交易日誌備份的存取權

若要設定交易日誌備份的存取權，請完成 [要求](#) 區段中的需求清單，然後執行 `rds_tlog_copy_setup` 預存程序。此程序將在資料庫執行個體層級啟用對交易日誌備份功能的存取。您不需要針對資料庫執行個體上的每個個別資料庫執行個體執行此程序。

Important

資料庫使用者必須在每個資料庫上獲授予 SQL Server 內的 `db_owner` 角色，才能設定和使用交易日誌備份功能的存取權。

Example 使用方式：

```
exec msdb.dbo.rds_tlog_copy_setup
@target_s3_arn='arn:aws:s3:::DOC-EXAMPLE-BUCKET/myfolder';
```

以下是必要參數：

- `@target_s3_arn` – 要將交易日誌備份檔案複製到其中之目標 Amazon S3 儲存貯體的 ARN。

Example 設定 Amazon S3 目標儲存貯體的範例：

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::DOC-EXAMPLE-LOGGING-
BUCKET/mytestdb1';
```

若要驗證組態，請呼叫 `rds_show_configuration` 預存程序。

Example 驗證組態的範例：

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

若要修改交易日誌備份的存取權，以指向不同的 Amazon S3 儲存貯體，您可以檢視目前的 Amazon S3 儲存貯體值，然後使用 @target_s3_arn 的新值重新執行預存程序 rds_tlog_copy_setup。

Example 檢視為了存取交易日誌備份而設定現有 Amazon S3 儲存貯體的範例

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Example 更新為新目標 Amazon S3 儲存貯體的範例

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3::DOC-EXAMPLE-LOGGING-BUCKET1/mynewfolder';
```

列出可用的交易日誌備份

使用 RDS for SQL Server 時，資料庫若設定為使用完整復原模式，且其資料庫執行個體備份保留設為一或多天，則會自動啟用交易日誌備份。透過啟用對交易日誌備份的存取，最多七天的這些交易日誌備份可供您複製到 Amazon S3 儲存貯體。

在啟用對交易日誌備份的存取之後，您可以開始使用它，來列出和複製可用的交易日誌備份檔案。

列出交易日誌備份

若要列出個別資料庫可用的所有交易日誌備份，請呼叫 rds_fn_list_tlog_backup_metadata 函數。您可以在呼叫函數時使用 ORDER BY 或 WHERE 子句。

Example 列出並篩選可用交易日誌備份檔案的範例

```
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename');
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
  rds_backup_seq_id = 3507;
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
  backup_file_time_utc > '2022-09-15 20:44:01' ORDER BY backup_file_time_utc DESC;
```

100 %

Results Messages

	db_name	db_id	family_guid	rds_backup_seq_id	backup_file_epoch	backup_file_time_utc	starting_lsn	ending_lsn	is_log_chain_broken	file_size_bytes	Error
1	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	43	1661846641	2022-08-30 08:04:01	5450000085730100001	5450000085731000001	0	35564	NULL
2	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	44	1661846941	2022-08-30 08:09:01	5450000085731000001	5450000085731900001	0	35473	NULL
3	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	45	1661847241	2022-08-30 08:14:01	5450000085731900001	5450000085732800001	0	35394	NULL
4	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	46	1661847541	2022-08-30 08:19:01	5450000085732800001	5450000085733700001	0	35374	NULL
5	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	47	1661847841	2022-08-30 08:24:01	5450000085733700001	5450000085734600001	0	35601	NULL
6	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	48	1661848142	2022-08-30 08:29:02	5450000085734600001	5450000085735500001	0	35470	NULL
7	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	49	1661848441	2022-08-30 08:34:01	5450000085735500001	5450000085736400001	0	35491	NULL
8	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	50	1661848741	2022-08-30 08:39:01	5450000085736400001	5450000085737300001	0	35520	NULL
9	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	51	1661849041	2022-08-30 08:44:01	5450000085737300001	5450000085738200001	0	35326	NULL
10	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	52	1661849341	2022-08-30 08:49:01	5450000085738200001	5450000085739100001	0	35407	NULL
11	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	53	1661849641	2022-08-30 08:54:01	5450000085739100001	5450000085740000001	0	35491	NULL
12	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	54	1661849941	2022-08-30 08:59:01	5450000085740000001	5450000085740900001	0	35438	NULL
13	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	55	1661850241	2022-08-30 09:04:01	5450000085740900001	5450000085741800001	0	35319	NULL
14	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	56	1661850541	2022-08-30 09:09:01	5450000085741800001	5450000085742700001	0	35270	NULL
15	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	57	1661850841	2022-08-30 09:14:01	5450000085742700001	5450000085743600001	0	35476	NULL

`rds_fn_list_tlog_backup_metadata` 函數會傳回下列輸出：

欄名稱	資料類型	描述
<code>db_name</code>	<code>sysname</code>	提供用來列出其交易日誌備份的資料庫名稱。
<code>db_id</code>	<code>int</code>	輸入參數 <code>db_name</code> 的內部資料庫識別符。
<code>family_guid</code>	<code>uniqueidentifier</code>	建立時原始資料庫的唯一 ID。此值會在資料庫還原時會保持相同，即使是不同的資料庫名稱也是如此。
<code>rds_backup_seq_id</code>	<code>int</code>	RDS 在內部用來維護每個交易日誌備份檔案之序號的 ID。
<code>backup_file_epoch</code>	<code>bigint</code>	產生交易備份檔案的 epoch 時間。
<code>backup_file_time_utc</code>	<code>datetime</code>	<code>backup_file_epoch</code> 值的 UTC 時間轉換值。
<code>starting_lsn</code>	<code>numeric(250)</code>	交易日誌備份檔案之第一個或最舊日誌記錄的日誌序號。
<code>ending_lsn</code>	<code>numeric(250)</code>	交易日誌備份檔案之最後一個或下一個日誌記錄的日誌序號。
<code>is_log_chain_broken</code>	<code>bit</code>	布林值，指出目前交易日誌備份檔案與上一個交易日誌備份檔案之間的日誌鏈是否中斷。

欄名稱	資料類型	描述
file_size_bytes	bigint	交易備份集的大小 (以位元組為單位)。
Error	varchar(4000)	錯誤訊息，如果 rds_fn_list_tlog_backup_metadata 函數擲出例外狀況的話。NULL，如果沒有例外狀況的話。

複製交易日誌備份

若要將個別資料庫的一組可用交易日誌備份複製到 Amazon S3 儲存貯體，請呼叫 rds_tlog_backup_copy_to_S3 預存程序。rds_tlog_backup_copy_to_S3 預存程序將起始新任務以複製交易日誌備份。

Note

rds_tlog_backup_copy_to_S3 預存程序將複製交易日誌備份，而不會根據 is_log_chain_broken 屬性進行驗證。因此，您應該在執行 rds_tlog_backup_copy_to_S3 預存程序之前，手動確認未中斷的日誌鏈。如需進一步說明，請參閱 [驗證交易日誌備份日誌鏈](#)。

Example rds_tlog_backup_copy_to_S3 預存程序的使用方式

```
exec msdb.dbo.rds_tlog_backup_copy_to_S3
  @db_name='mydatabasename',
  [@kms_key_arn='arn:aws:kms:region:account-id:key/key-id'],
  [@backup_file_start_time='2022-09-01 01:00:15'],
  [@backup_file_end_time='2022-09-01 21:30:45'],
  [@starting_lsn=149000000112100001],
  [@ending_lsn=149000000120400001],
  [@rds_backup_starting_seq_id=5],
  [@rds_backup_ending_seq_id=10];
```

下列是可用的輸入參數：

參數	描述
@db_name	要複製其交易日誌備份的資料庫名稱。
@kms_key_arn	客戶管理的 KMS 金鑰。如果您使用 AWS 受管 KMS 金鑰加密資料庫執行個體，則必須建立客戶受管金鑰。如果您使用客戶受管金鑰加密資料庫執行個體，則可以使用相同的 KMS 金鑰 ARN。
@backup_file_start_time	從 rds_fn_list_tlog_backup_metadata 函數的 [backup_file_time_utc] 資料欄提供的 UTC 時間戳記。
@backup_file_end_time	從 rds_fn_list_tlog_backup_metadata 函數的 [backup_file_time_utc] 資料欄提供的 UTC 時間戳記。
@starting_lsn	從 rds_fn_list_tlog_backup_metadata 函數的 [starting_lsn] 資料欄提供的日誌序號 (LSN)
@ending_lsn	從 rds_fn_list_tlog_backup_metadata 函數的 [ending_lsn] 資料欄提供的日誌序號 (LSN)。
@rds_backup_starting_seq_id	從 rds_fn_list_tlog_backup_metadata 函數的 [rds_backup_seq_id] 資料欄提供的序號。
@rds_backup_ending_seq_id	從 rds_fn_list_tlog_backup_metadata 函數的 [rds_backup_seq_id] 資料欄提供的序號。

您可以指定一組時間、LSN 或序列 ID 參數。只需要一組參數。

您也可以在任何集合中只指定單一參數。例如，透過僅針對 backup_file_end_time 參數提供一值，七天限制內在該時間之前的所有可用交易日誌備份檔案都會複製到 Amazon S3 儲存貯體。

下列是 rds_tlog_backup_copy_to_S3 預存程序的有效輸入參數組合。

提供的參數	預期的結果
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3</pre>	複製過去七天的交易日誌備份，且存在於提供的

提供的參數	預期的結果	
<pre>@db_name = 'testdb1', @backup_f ile_start _time='20 22-08-23 00:00:00', @backup_f ile_end_t ime='2022 -08-30 00:00:00';</pre>	<p>範圍 backup_fi le_start_time 與 backup_fi le_end_time 之間。在此範例 中，預存程序會複 製已在 '2022-08- 23 00:00:00' 與 '2022-08-30 00:00:00'之間產生的 交易日誌備份。</p>	
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name = 'testdb1', @backup_f ile_start _time='20 22-08-23 00:00:00';</pre>	<p>複製過去七天的交 易日誌備份，並從 提供的 backup_fi le_start_time 開始。在此範例 中，預存程序會複 製來自 '2022-08-23 00:00:00' 的交易日 誌備份，直到最新的 交易日誌備份。</p>	

提供的參數	預期的結果	
<pre>exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_end_time='2022-08-30 00:00:00';</pre>	<p>複製過去七天的交易日誌備份，直到提供的 backup_file_end_time。在此範例中，預存程序會複製來自 '2022-08-23 00:00:00' 直到 '2022-08-30 00:00:00' 的交易日誌備份。</p>	
<pre>exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name= 'testdb1', @starting_lsn =1490000000040007, @ending_lsn = 1490000000050009;</pre>	<p>複製過去七天可用且在所提供範圍 starting_lsn 與 ending_lsn 之間的交易日誌備份。在此範例中，預存程序會複製過去七天，LSN 範圍介於 1490000000040007 和 1490000000050009 之間的交易日誌備份。</p>	

提供的參數	預期的結果	
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @starting _lsn =14900000 00040007;</pre>	<p>複製過去七天可用的交易日誌備份，從提供的 starting_lsn 開始。在此範例中，預存程序會複製來自 LSN 1490000000040007 的交易日誌備份，直到最新的交易日誌備份。</p>	
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @ending_lsn =14900000 00050009;</pre>	<p>複製過去七天可用的交易日誌備份，直到提供的 ending_lsn 。在此範例中，預存程序會複製從過去七天開始直到 lsn 1490000000050009 的交易日誌備份。</p>	

提供的參數	預期的結果	
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @rds_back up_starti ng_seq_id= 2000, @rds_back up_ending _seq_id= 5000;</pre>	<p>複製過去七天可用且存在於所提供範圍 rds_backu p_startin g_seq_id 與 rds_backu p_ending_ seq_id 之間的交易日誌備份。在此範例中，預存程序會複製從過去七天開始，並在所提供 rds 備份序列 ID 範圍內的交易日誌備份，從 seq_id 2000 開始，直到 to seq_id 5000。</p>	
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @rds_back up_starti ng_seq_id= 2000;</pre>	<p>複製過去七天可用的交易日誌備份，從提供的 rds_backu p_startin g_seq_id 開始。在此範例中，預存程序會複製從 seq_id 2000 開始的交易日誌備份，直到最新的交易日誌備份。</p>	

提供的參數	預期的結果
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @rds_back up_ending _seq_id= 5000;</pre>	<p>複製過去七天可用的交易日誌備份，直到提供的 <code>rds_backup_ending_seq_id</code>。在此範例中，預存程序會複製從過去七天開始，直到 <code>seq_id 5000</code> 的交易日誌備份。</p>
<pre>exec msdb.dbo. rds_tlog_ backup_co py_to_S3 @db_name= 'testdb1', @rds_back up_starti ng_seq_id= 2000; @rds_back up_ending _seq_id= 2000;</pre>	<p>複製具有所提供 <code>rds_backup_starting_seq_id</code> 的單一交易日誌備份，如果是在過去七天內提供的話。在此範例中，預存程序會複製 <code>seq_id</code> 為 2000 的單一交易日誌備份 (如果它存在於過去七天內)。</p>

驗證交易日誌備份日誌鏈

設定用於存取交易日誌備份的資料庫必須已啟用自動備份保留。自動備份保留會將資料庫執行個體上的資料庫設為 FULL 復原模式。若要支援資料庫的時間點還原，請避免變更資料庫復原模式，這可能會導致日誌鏈中斷。建議您繼續將資料庫設為 FULL 復原模式。

若要在複製交易日誌備份之前手動驗證日誌鏈，請呼叫 `rds_fn_list_tlog_backup_metadata` 函數並檢閱 `is_log_chain_broken` 資料欄中的值。值「1」表示目前日誌備份與上一個日誌備份之間の日誌鏈已中斷。

下列範例會在 `rds_fn_list_tlog_backup_metadata` 預存程序的輸出中顯示中斷的日誌鏈。

rds_sequence_id	first_lsn	last_lsn	is_log_chain_broken
43	90023	90457	0
44	90457	90985	0
45	90987	92034	1

在一般日誌鏈中，所指定 `rds_sequence_id` 之 `first_lsn` 的日誌序號 (LSN) 值應該符合先前 `rds_sequence_id` 中 `last_lsn` 的值。在影像中，`rds_sequence_id` 45 具有 `first_lsn` 值 90987，其不符合先前 `rds_sequence_id` 44 的 `last_lsn` 值 90985。

如需 SQL Server 交易日誌架構和日誌序號的詳細資訊，請參閱 Microsoft SQL Server 文件中的[交易日誌邏輯架構](#)。

Amazon S3 儲存貯體資料夾和檔案結構

交易日誌備份在 Amazon S3 儲存貯體內具有下列標準結構和命名慣例：

- 系統會在每個資料庫的 `target_s3_arn` 路徑下建立一個新資料夾，其命名結構為 `{db_id}. {family_guid}`。
- 在資料夾內，交易日誌備份的檔案名稱結構為 `{db_id}. {family_guid}. {rds_backup_seq_id}. {backup_file_epoch}`。
- 您可以使用 `rds_fn_list_tlog_backup_metadata` 函數檢視 `family_guid`, `db_id`, `rds_backup_seq_id` and `backup_file_epoch` 的詳細資訊。

下列範例顯示 Amazon S3 儲存貯體內一組交易日誌備份的資料夾和檔案結構。

Amazon S3 > Buckets > rds-sql-server-kms-bucket > 10.36a85812-2b1e-47c6-b956-a020776fff66/

10.36a85812-2b1e-47c6-b956-a020776fff66/ Copy S3 URI

Objects Properties

Objects (87)
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
10.36a85812-2b1e-47c6-b956-a020776fff66.0.1664557862	1664557862	September 30, 2022, 14:38:23 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.1.1664558161	1664558161	September 30, 2022, 14:38:23 (UTC-07:00)	7.0 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.2.1664558461	1664558461	September 30, 2022, 14:38:24 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.3.1664558761	1664558761	September 30, 2022, 14:38:24 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.4.1664559061	1664559061	September 30, 2022, 14:38:24 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.5.1664559361	1664559361	September 30, 2022, 14:38:24 (UTC-07:00)	9.0 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.6.1664559661	1664559661	October 2, 2022, 22:27:23 (UTC-07:00)	7.0 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.7.1664559961	1664559961	October 2, 2022, 22:27:23 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.8.1664560261	1664560261	October 2, 2022, 22:27:23 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.9.1664560561	1664560561	October 2, 2022, 22:27:23 (UTC-07:00)	6.5 KB	Standard
10.36a85812-2b1e-47c6-b956-a020776fff66.10.1664560862	1664560862	October 2, 2022, 22:27:24 (UTC-07:00)	6.5 KB	Standard

追蹤任務的狀態

若要追蹤複製任務的狀態，請呼叫 `rds_task_status` 預存程序。如果您不提供任何參數，預存程序會傳回所有任務的狀態。

Example 使用方式：

```
exec msdb.dbo.rds_task_status
  @db_name='database_name',
  @task_id=ID_number;
```

下列是選用參數：

- `@db_name`– 要顯示任務狀態之資料庫的名稱。
- `@task_id`– 要顯示任務狀態之任務的 ID。

Example 列出特定任務 ID 之狀態的範例：

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example 列出特定資料庫和任務之狀態的範例：

```
exec msdb.dbo.rds_task_status@db_name='my_database',@task_id=5;
```

Example 列出特定資料庫之所有任務及其狀態的範例：

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example 列出目前執行個體上所有任務及其狀態的範例：

```
exec msdb.dbo.rds_task_status;
```

取消任務

若要取消執行中任務，請呼叫 `rds_cancel_task` 預存程序。

Example 使用方式：

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

以下是必要參數：

- `@task_id`– 要取消之任務的 ID。您可以呼叫 `rds_task_status` 預存程序來檢視任務 ID。

如需檢視和取消執行中任務的詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

針對交易日誌備份的存取進行疑難排解

下列是您使用預存程序存取交易日誌備份時可能遇到的問題。

預存程序	錯誤訊息	問題	故障診斷建議
<code>rds_tlog_copy_setup</code>	此資料庫執行個體上停	未針對資料庫執行個體啟用自動備份。	必須啟用其保留至少一天的資料庫執行個體備份保留。如需啟用

預存程序	錯誤訊息	問題	故障診斷建議
	用備份。啟用其保留至少為 "1" 的資料庫執行個體備份，然後再試一次。		自動備份和設定備份保留的詳細資訊，請參閱 Backup retention period (備份保留期間) 。
rds_tlog_copy_setup	執行 rds_tlog_copy_setup 預存程序時發生錯誤。重新連線至 RDS 端點，然後再試一次。	發生內部錯誤。	重新連線至 RDS 端點，然後重新執行 rds_tlog_copy_setup 預存程序。
rds_tlog_copy_setup	不支援在交易內執行 rds_tlog_backup_copy_setup 預存程序。驗證工作階段沒有已開啟的交易，然後再試一次。	已在交易內使用 BEGIN 和 END 嘗試預存程序。	避免在執行 rds_tlog_copy_setup 預存程序時使用 BEGIN 和 END。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_copy_setup	輸入參數 @target_s3_arn 的 S3 儲存貯體名稱應該至少包含一個空格以外的字元。	提供給輸入參數 @target_s3_arn 的值不正確。	確定輸入參數 @target_s3_arn 指定完整的 Amazon S3 儲存貯體 ARN。
rds_tlog_copy_setup	SQLSERVER_BACKUP_RESTORE 選項未啟用，或正在進行啟用。啟用此選項，或稍後再試一次。	SQLSERVER_BACKUP_RESTORE 選項未在資料庫執行個體上啟用，或者只是啟用並擱置內部啟用。	依 Requirements (需求) 區段中所指定，啟用 SQLSERVER_BACKUP_RESTORE 選項。稍候幾分鐘，然後重新執行 rds_tlog_copy_setup 預存程序。
rds_tlog_copy_setup	輸入參數 @target_s3_arn 的目標 S3 arn 不能空白或為 null。	已提供 NULL 值給輸入參數 @target_s3_arn ，或者未提供值。	確定輸入參數 @target_s3_arn 指定完整的 Amazon S3 儲存貯體 ARN。
rds_tlog_copy_setup	輸入參數 @target_s3_arn 的目標 S3 arn 必須以 arn:aws 開頭。	已提供前面沒有 arn:aws 的輸入參數 @target_s3_arn 。	確定輸入參數 @target_s3_arn 指定完整的 Amazon S3 儲存貯體 ARN。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_copy_setup	目標 S3 ARN 已設為提供的值。	先前已執行 rds_tlog_copy_setup 預存程序，並使用了 Amazon S3 儲存貯體 ARN 進行設定。	若要修改 Amazon S3 儲存貯體值以存取交易日誌備份，請提供不同的 target S3 ARN。
rds_tlog_copy_setup	無法產生憑證以啟用對交易日誌備份的存取。確認 rds_tlog_copy_setup 隨附 S3 路徑 ARN，稍後再試一次。	產生憑證以啟用對交易日誌備份的存取時發生未指定的錯誤。	檢閱您的設定組態，然後再試一次。
rds_tlog_copy_setup	當有擱置中任務時，您無法執行 rds_tlog_copy_setup 預存程序。等待擱置中任務完成，然後再試一次。	隨時只能執行兩項任務。有擱置中任務等待完成。	檢視擱置中任務並等待它們完成。如需監控任務狀態的詳細資訊，請參閱 追蹤任務的狀態 。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	已針對資料庫 %s 發出 T-log 備份檔案複製任務，任務識別符為 %d。請稍後再試一次。	隨時只能針對指定資料庫執行一個複製任務。有擱置中複製任務等待完成。	檢視擱置中任務並等待它們完成。如需監控任務狀態的詳細資訊，請參閱 追蹤任務的狀態 。
rds_tlog_backup_copy_to_S3	必須至少提供這三個參數集的其中一個。 SET-1:(@backup_file_start_time, @backup_file_end_time) SET-2:(@starting_lsn, @ending_lsn) SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	未提供這三個參數集的任一個，或提供的參數組缺少必要參數。	您可以指定時間、lsn 或序列 ID 參數。需要這三組參數中的一組。如需必要參數的詳細資訊，請參閱 複製交易日誌備份 。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	您的執行個體上停用備份。請啟用備份，一段時間後再試一次。	未針對資料庫執行個體啟用自動備份。	如需啟用自動備份和設定備份保留的詳細資訊，請參閱 Backup retention period (備份保留期間) 。
rds_tlog_backup_copy_to_S3	找不到指定的資料庫 %s。	提供給輸入參數 @db_name 的值與資料庫執行個體上的資料庫名稱不符。	使用正確的資料庫名稱。若要依名稱列出所有資料庫，請執行 <code>SELECT * from sys.databases</code>
rds_tlog_backup_copy_to_S3	無法針對 SQL Server 系統資料庫或 rdsadmin 資料庫執行 rds_tlog_backup_copy_to_S3 預存程序。	提供給輸入參數 @db_name 的值與 SQL Server 系統資料庫名稱或 RDSAdmin 資料庫相符。	存取交易日誌備份時，不允許使用下列資料庫：master, model, msdb, tempdb, RDSAdmin。
rds_tlog_backup_copy_to_S3	輸入參數 @db_name 的資料庫名稱不能空白或為 null。	提供給輸入參數 @db_name 的值空白或為 NULL。	使用正確的資料庫名稱。若要依名稱列出所有資料庫，請執行 <code>SELECT * from sys.databases</code>

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	資料庫執行個體備份保留期間必須至少設為 1，才能執行 rds_tlog_backup_copy_setup 預存程序。	未針對資料庫執行個體啟用自動備份。	如需啟用自動備份和設定備份保留的詳細資訊，請參閱 Backup retention period (備份保留期間) 。
rds_tlog_backup_copy_to_S3	執行預存程序 rds_tlog_backup_copy_to_S3 時發生錯誤。重新連線至 RDS 端點，然後再試一次。	發生內部錯誤。	重新連線至 RDS 端點，然後重新執行 rds_tlog_backup_copy_to_S3 預存程序。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	只能提供這三個參數集的其中一個。SET-1:(@backup_file_start_time, @backup_file_end_time) SET-2:(@starting_lsn, @ending_lsn) SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	提供了多個參數集。	您可以指定時間、lsn 或序列 ID 參數。需要這三組參數中的一組。如需必要參數的詳細資訊，請參閱 複製交易日誌備份 。
rds_tlog_backup_copy_to_S3	不支援在交易內執行 rds_tlog_backup_copy_to_S3 預存程序。驗證工作階段沒有已開啟的交易，然後再試一次。	已在交易內使用 BEGIN 和 END 嘗試預存程序。	避免在執行 rds_tlog_backup_copy_to_S3 預存程序時使用 BEGIN 和 END。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	提供的參數落在交易日誌記錄保留期間之外。若要列出可用的交易日誌備份檔案，請執行 rds_fn_list_tlog_backup_metadata 函數。	對於適合複本保留時段的所提供輸入參數，沒有可用的交易日誌備份。	使用有效的一組參數再試一次。如需必要參數的詳細資訊，請參閱 複製交易日誌備份 。
rds_tlog_backup_copy_to_S3	處理請求時發生許可錯誤。確保儲存貯體與資料庫執行個體位於相同的帳戶和區域中，並確認公開文件中針對範本的 S3 儲存貯體政策許可。	偵測到提供的 S3 儲存貯體或其政策許可發生問題。	確認您針對存取交易日誌備份進行的設定是正確的。如需 S3 儲存貯體設定需求的詳細資訊，請參閱 要求 。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	不允許在 RDS 僅供讀取複本執行個體上執行 rds_tlog_backup_copy_to_S3 預存程序。	已在 RDS 僅供讀取複本執行個體上嘗試預存程序。	更正至 RDS 主要資料庫執行個體以執行 rds_tlog_backup_copy_to_S3 預存程序。
rds_tlog_backup_copy_to_S3	輸入參數 @starting_lsn 的 LSN 必須小於 @ending_lsn 。	提供給輸入參數 @starting_lsn 的值大於提供給輸入參數 @ending_lsn 的值。	確定提供給輸入參數 @starting_lsn 的值小於提供給輸入參數 @ending_lsn 的值。
rds_tlog_backup_copy_to_S3	rds_tlog_backup_copy_to_S3 預存程序只能由來源資料庫中的 db_owner 角色成員執行。	對於嘗試在提供的 db_name 上執行 rds_tlog_backup_copy_to_S3 預存程序的帳戶，尚未授予 db_owner 角色。	確定執行預存程序的帳戶是透過所提供 db_name 的 db_owner 角色獲得許可。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	輸入參數的序列 ID @rds_backup_starting_seq_id 必須小於或等於或等於 @rds_backup_ending_seq_id 。	提供給輸入參數 @rds_backup_starting_seq_id 的值大於提供給輸入參數 @rds_backup_ending_seq_id 的值。	確定提供給輸入參數 @rds_backup_starting_seq_id 的值小於提供給輸入參數 @rds_backup_ending_seq_id 的值。
rds_tlog_backup_copy_to_S3	SQLSERVER_BACKUP_RESTORE 選項未啟用，或正在進行啟用。啟用此選項，或稍後再試一次。	SQLSERVER_BACKUP_RESTORE 選項未在資料庫執行個體上啟用，或者只是啟用並擱置內部啟用。	依 Requirements (需求) 區段中所指定，啟用 SQLSERVER_BACKUP_RESTORE 選項。稍候幾分鐘，然後重新執行 rds_tlog_backup_copy_to_S3 預存程序。
rds_tlog_backup_copy_to_S3	輸入參數 @backup_file_start_time 的開始時間必須小於 @backup_file_end_time 。	提供給輸入參數 @backup_file_start_time 的值大於提供給輸入參數 @backup_file_end_time 的值。	確定提供給輸入參數 @backup_file_start_time 的值小於提供給輸入參數 @backup_file_end_time 的值。

預存程序	錯誤訊息	問題	故障診斷建議
rds_tlog_backup_copy_to_S3	由於缺少存取權，我們無法處理請求。請檢查功能的設定和許可。	Amazon S3 儲存貯體許可可能有問題，或提供的 Amazon S3 儲存貯體位於另一個帳戶或區域中。	確保授權 Amazon S3 儲存貯體政策獲得允許 RDS 存取的許可。確保 Amazon S3 儲存貯體與資料庫執行個體位於相同的帳戶和區域中。
rds_tlog_backup_copy_to_S3	對於未經儲存貯體加密的執行個體，您無法提供 KMS 金鑰 ARN，作為預存程序的輸入參數。	未在資料庫執行個體上啟用儲存貯體加密時，不應提供輸入參數 @kms_key_arn 。	請不要提供輸入參數給 @kms_key_arn 。
rds_tlog_backup_copy_to_S3	對於儲存貯體加密的執行個體，您必須提供 KMS 金鑰 ARN，作為預存程序的輸入參數。	在資料庫執行個體上啟用儲存貯體加密時，必須提供輸入參數 @kms_key_arn 。	提供輸入參數給 @kms_key_arn，其中有一值符合用於交易日誌備份之 Amazon S3 儲存貯體的 ARN。

預存程序	錯誤訊息	問題	故障診斷建議
<code>rds_tlog_backup_copy_to_S3</code>	在執行 <code>rds_tlog_backup_copy_to_S3</code> 預存程序之前，您必須先執行 <code>rds_tlog_copy_setup</code> 預存程序並設定 <code>@target_s3_arn</code> 。	在嘗試執行 <code>rds_tlog_backup_copy_to_S3</code> 預存程序之前，尚未完成對交易日誌備份設定程序的存取。	執行 <code>rds_tlog_copy_setup</code> 預存程序，然後再執行 <code>rds_tlog_backup_copy_to_S3</code> 預存程序。如需執行設定程序以存取交易日誌備份的詳細資訊，請參閱 設定交易日誌備份的存取權 。

Microsoft SQL Server 資料庫引擎的選項

在本節中，您可以找到可供執行 Microsoft SQL Server 資料庫引擎之 Amazon RDS 執行個體使用的選項的說明。若要啟用這些選項，請將其新增至選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需更多詳細資訊，請參閱 [使用選項群組](#)。

如果您要尋找的選用功能不是透過 RDS 選項群組所新增，例如 SSL、Microsoft Windows 驗證和 Amazon S3 整合，請參閱 [Microsoft SQL Server on Amazon RDS 的其他功能](#)。

Amazon RDS 支援 Microsoft SQL Server 資料庫執行個體的下列選項。

選項	選項 ID	引擎版本
連結的伺服器搭配 Oracle OLEDB	OLEDB_ORACLE	SQL Server Enterprise Edition SQL Server Standard Edition
原生備份與還原	SQLSERVER_BACKUP_RESTORE	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
透明資料加密	TRANSPARENT_DATA_ENCRYPTION (RDS 主控台) TDE (AWS CLI 和 RDS API)	SQL 伺服器企業版 SQL 伺服器標準版

選項	選項 ID	引擎版本
SQL Server 稽核	SQLSERVER_AUDIT	<p>在 RDS 中，從 SQL Server 2014 開始，SQL Server 的所有版本均支援伺服器層級稽核，企業版也支援資料庫層級稽核。</p> <p>從 SQL Server SQL Server 2016 (13.x) SP1 開始，所有版本均支援伺服器層級和資料庫層級稽核。</p> <p>如需更多詳細資訊，請參閱 SQL Server 文件中的 SQL Server 稽核 (資料庫引擎)。</p>
SQL Server Analysis Services	SSAS	<p>SQL Server Enterprise Edition</p> <p>SQL Server Standard Edition</p>
SQL Server Integration Services	SSIS	<p>SQL Server Enterprise Edition</p> <p>SQL Server Standard Edition</p>
SQL Server Reporting Services	SSRS	<p>SQL Server Enterprise Edition</p> <p>SQL Server Standard Edition</p>

選項	選項 ID	引擎版本
Microsoft Distributed Transaction Coordinator	MSDTC	在 RDS 中，自 SQL Server 2014 開始，所有版本的 SQL Server 都支援分散式交易。

列出 SQL Server 版本的可用選項

您可以使用 `describe-option-group-options` AWS CLI 命令列出 SQL Server 版本的可用選項，以及這些選項的設定。

下列範例顯示了 SQL Server 2019 Enterprise Edition 的選項和選項設定。--engine-name 選項為必要項目。

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version 15.00
```

輸出看起來會與以下內容相似：

```
{
  "OptionGroupOptions": [
    {
      "Name": "MSDTC",
      "Description": "Microsoft Distributed Transaction Coordinator",
      "EngineName": "sqlserver-ee",
      "MajorEngineVersion": "15.00",
      "MinimumRequiredMinorEngineVersion": "4043.16.v1",
      "PortRequired": true,
      "DefaultPort": 5000,
      "OptionsDependedOn": [],
      "OptionsConflictsWith": [],
      "Persistent": false,
      "Permanent": false,
      "RequiresAutoMinorEngineVersionUpgrade": false,
      "VpcOnly": false,
      "OptionGroupOptionSettings": [
        {
          "SettingName": "ENABLE_SNA_LU",
          "SettingDescription": "Enable support for SNA LU protocol",

```

```
        "DefaultValue": "true",
        "ApplyType": "DYNAMIC",
        "AllowedValues": "true,false",
        "IsModifiable": true,
        "IsRequired": false,
        "MinimumEngineVersionPerAllowedValue": []
    },
    ...
    {
        "Name": "TDE",
        "Description": "SQL Server - Transparent Data Encryption",
        "EngineName": "sqlserver-ee",
        "MajorEngineVersion": "15.00",
        "MinimumRequiredMinorEngineVersion": "4043.16.v1",
        "PortRequired": false,
        "OptionsDependedOn": [],
        "OptionsConflictsWith": [],
        "Persistent": true,
        "Permanent": false,
        "RequiresAutoMinorEngineVersionUpgrade": false,
        "VpcOnly": false,
        "OptionGroupOptionSettings": []
    }
]
}
```


Amazon RDS for SQL Server 中支援連結的伺服器搭配 Oracle OLEDB

連結的伺服器搭配 RDS for SQL Server 上的 Oracle Provider for OLEDB 可讓您存取 Oracle 資料庫上的外部資料來源。您可以從遠端 Oracle 資料來源讀取資料，並針對 RDS for SQL Server 資料庫執行個體之外的遠端 Oracle 資料庫伺服器執行命令。搭配 Oracle OLEDB 使用連結的伺服器，您可以

- 直接存取 SQL Server 以外的資料來源
- 在沒有移動資料的情況下，使用相同查詢針對各種 Oracle 資料進行查詢
- 跨企業生態系統對資料來源發出分散式查詢、更新、命令和交易
- 從 Microsoft Business Intelligence 套件 (SSIS、SSRS、SSAS) 內整合與 Oracle 資料庫的連線
- 從 Oracle 資料庫遷移到 RDS for SQL Server

在現有或新的 RDS for SQL Server 資料庫執行個體上，您可以針對 Oracle 啟動一或多個連結的伺服器。然後，您可以整合外部 Oracle 資料來源與資料庫執行個體。

內容

- [支援的版本和區域](#)
- [限制與建議](#)
- [啟用連結的伺服器搭配 Oracle](#)
 - [建立 OLEDB_ORACLE 的選項群組](#)
 - [將 OLEDB_ORACLE 選項新增至選項群組](#)
 - [將選項群組關聯至您的資料庫執行個體](#)
- [修改 OLEDB 提供者屬性](#)
- [修改 OLEDB 驅動程式屬性](#)
- [停用連結的伺服器搭配 Oracle](#)

支援的版本和區域

對於下列版本上的 SQL Server 標準版和企業版，RDS for SQL Server 在所有區域都支援連結的伺服器搭配 Oracle OLEDB：

- SQL 伺服器，所有版本
- SQL Server 2019，所有版本
- SQL Server 2017，所有版本

下列 Oracle Database 版本支援連結的伺服器搭配 Oracle OLEDB：

- Oracle Database 21c，所有版本
- Oracle Database 19c，所有版本
- Oracle Database 18c，所有版本

限制與建議

請記住，連結的伺服器搭配 Oracle OLEDB 時適用下列限制和建議：

- 針對每個 RDS for SQL Server 資料庫執行個體，透過在安全群組中新增適用的 TCP 連接埠，以允許網路流量。例如，如果您要在 EC2 Oracle 資料庫執行個體與 RDS for SQL Server 資料庫執行個體之間設定連結的伺服器，則必須允許來自 EC2 Oracle 資料庫執行個體 IP 地址的流量。您也必須在 SQL Server 用來接聽資料庫通訊的連接埠上允許流量。如需安全群組的詳細資訊，請參閱 [使用安全群組控制存取](#)。
- 在開啟、關閉或修改選項群組中的 OLEDB_ORACLE 選項之後，執行 RDS for SQL Server 資料庫執行個體的重新啟動。選項群組狀態會針對這些事件顯示 pending_reboot，且為必要項目。
- 僅支援使用 Oracle 資料來源的使用者名稱和密碼進行簡易身分驗證。
- 不支援開放式資料庫連線 (ODBC) 驅動程式。僅支援最新版本的 OLEDB 驅動程式。
- 支援分散式交易 (XA)。若要啟用分散式交易，請在資料庫執行個體的選項群組中開啟 MSDTC 選項，並確定 XA 交易已開啟。如需詳細資訊，請參閱 [RDS for SQL Server 中的 Microsoft Distributed Transaction Coordinator 支援](#)。
- 不支援建立用作連接字串捷徑的資料來源名稱 (DSN)。
- 不支援 OLEDB 驅動程式追蹤。您可以使用 SQL Server 擴充事件來追蹤 OLEDB 事件。如需詳細資訊，請參閱 [設定 RDS for SQL Server 中的擴充事件](#)。
- 使用 SQL Server Management Studio (SSMS) 時，不支援存取 Oracle 連結伺服器的目錄資料夾。

啟用連結的伺服器搭配 Oracle

將 OLEDB_ORACLE 選項新增至 RDS for SQL Server 資料庫執行個體，來啟用連結的伺服器搭配 Oracle。請使用下列程序：

1. 建立新的選項群組或選擇現有的選項群組。
2. 將 OLEDB_ORACLE 選項新增至選項群組。

3. 選擇要使用的 OLEDB 驅動程式版本。
4. 將選項群組與資料庫執行個體建立關聯。
5. 重新啟動資料庫執行個體。

建立 OLEDB_ORACLE 的選項群組

若要搭配 Oracle 使用連結的伺服器，請建立選項群組，或修改對應至 SQL Server 版本和您計劃使用的資料庫執行個體版本的選項群組。若要完成此程序，請使用 AWS Management Console 或 AWS CLI。

主控台

下列程序會建立 SQL Server Standard Edition 2019 的選項群組。

建立選項群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 視窗中，執行下列動作：
 - a. 在 Name (名稱) 中，輸入您 AWS 帳戶中的唯一選項群組名稱，例如 **oracle-oledb-se-2019**。名稱僅可包含字母、數字與連字號。
 - b. 對於 Description (描述)，請輸入選項群組的簡短描述，例如 **OLEDB_ORACLE option group for SQL Server SE 2019**。用於顯示用途的說明。
 - c. 對於 Engine (引擎)，請選擇 sqlserver-se。
 - d. 對於 Major engine version (主要引擎版本)，請選擇 15.00。
5. 選擇建立。

CLI

下列程序會建立 SQL Server Standard Edition 2019 的選項群組。

建立選項群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name oracle-oledb-se-2019 \  
  --engine-name sqlserver-se \  
  --major-engine-version 15.00 \  
  --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

在Windows中：

```
aws rds create-option-group ^  
  --option-group-name oracle-oledb-se-2019 ^  
  --engine-name sqlserver-se ^  
  --major-engine-version 15.00 ^  
  --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

將 OLEDB_ORACLE 選項新增至選項群組

接下來，使用AWS Management Console或 AWS CLI 將 OLEDB_ORACLE 選項新增至選項群組。

主控台

新增 OLEDB_ORACLE 選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您剛建立的選項群組 (在此範例中為 oracle-oledb-se2019 年)。
4. 選擇 Add option (新增選項)。
5. 在 Option details (選項詳細資訊) 下，選擇 OLEDB_ORACLE 作為 Option name (選項名稱)。
6. 在 Scheduling (排程) 下，選擇要立即新增選項或是在下一個維護時段新增選項。
7. 選擇 Add option (新增選項)。

CLI

新增 OLEDB_ORACLE 選項

- 將 OLEDB_ORACLE 選項新增至選項群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name oracle-oledb-se-2019 \  
  --options OptionName=OLEDB_ORACLE \  
  --apply-immediately
```

在Windows中：

```
aws rds add-option-to-option-group ^  
  --option-group-name oracle-oledb-se-2019 ^  
  --options OptionName=OLEDB_ORACLE ^  
  --apply-immediately
```

將選項群組關聯至您的資料庫執行個體

若要將 OLEDB_ORACLE 選項群組和參數群組與資料庫執行個體建立關聯，請使用 AWS Management Console 或 AWS CLI

主控台

若要完成啟用 Oracle 連結伺服器，請將您的 OLEDB_ORACLE 選項群組與新的或現有的資料庫執行個體建立關聯

- 對於新的資料庫執行個體，請在啟動執行個體時將它們相關聯。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請透過修改執行個體來建立關聯。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

CLI

您可以將 OLEDB_ORACLE 選項群組和參數群組與新的或現有的資料庫執行個體產生關聯。

使用 **OLEDB_ORACLE** 選項群組和參數群組建立執行個體

- 指定建立選項群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mytestsqlserveroracleoledbinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 15.0.4236.7.v1 \  
  --allocated-storage 100 \  
  --manage-master-user-password \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --domain-iam-role-name my-directory-iam-role \  
  --domain my-domain-id \  
  --option-group-name oracle-oledb-se-2019 \  
  --db-parameter-group-name my-parameter-group-name
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mytestsqlserveroracleoledbinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 15.0.4236.7.v1 ^  
  --allocated-storage 100 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --domain-iam-role-name my-directory-iam-role ^  
  --domain my-domain-id ^  
  --option-group-name oracle-oledb-se-2019 ^  
  --db-parameter-group-name my-parameter-group-name
```

修改執行個體並關聯 OLEDB_ORACLE 選項群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mytestsqlserveroracleoledbinstance \  
  --option-group-name oracle-oledb-se-2019 \  
  --db-parameter-group-name my-parameter-group-name \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^\  
  --db-instance-identifier mytestsqlserveroracleoledbinstance ^\  
  --option-group-name oracle-oledb-se-2019 ^\  
  --db-parameter-group-name my-parameter-group-name ^\  
  --apply-immediately
```

修改 OLEDB 提供者屬性

您可以檢視和變更 OLEDB 提供者的屬性。只有 master 使用者可以執行此任務。在資料庫執行個體上建立的所有 Oracle 連結伺服器，都會使用與該 OLEDB 提供者的相同屬性。呼叫 `sp_MSset_oledb_prop` 預存程序來變更 OLEDB 提供者的屬性。

變更 OLEDB 提供者屬性

```
USE [master]  
GO  
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'AllowInProcess', 1  
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'DynamicParameters', 0  
GO
```

您可以修改下列屬性：

屬性名稱	建議值 (1 = 開啟、0 = 關閉)	描述
Dynamic parameter	1	允許參數化查詢中的 SQL 預留位置 (由 '?' 表示)。
Nested queries	1	允許 SELECT 子句中的巢狀 FROM 陳述式，例如子查詢。
Level zero only	0	只會針對提供者呼叫基礎層級 OLEDB 界面。
Allow inprocess	1	如果已開啟，Microsoft SQL Server 允許將提供者具體化為處理中伺服器。將此屬性設為 1 以使用 Oracle 連結伺服器。
Non transacted updates	0	若是非零，SQL Server 允許更新。
Index as access path	False	若是非零，SQL Server 會嘗試使用提供者的索引來擷取資料。
Disallow adhoc access	False	如果已設定，SQL Server 不允許針對 OLEDB 提供者執行傳遞查詢。雖然可以核取此選項，但有時候執行傳遞查詢是更好的選擇。
Supports LIKE operator	1	指出提供者支援使用 LIKE 關鍵字進行查詢。

修改 OLEDB 驅動程式屬性

建立 Oracle 連結伺服器時，您可以檢視和變更 OLEDB 驅動程式的屬性。只有 master 使用者可以執行此任務。驅動程式屬性定義 OLEDB 驅動程式在使用遠端 Oracle 資料來源時處理資料的方式。驅動程式屬性專用於在資料庫執行個體上建立的每個 Oracle 連結伺服器。呼叫 `master.dbo.sp_addlinkedserver` 預存程序來變更 OLEDB 驅動程式的屬性。

範例：建立連結的伺服器並變更 OLEDB 驅動程式 FetchSize 屬性

```
EXEC master.dbo.sp_addlinkedserver
```



```
@server = N'Oracle_link2',
@srvproduct=N'Oracle',
@provider=N'OraOLEDB.Oracle',
@datasrc=N'my-oracle-test.cnetsipka.us-west-2.rds.amazonaws.com:1521/ORCL,
@provstr='FetchSize=200'
GO
```

```
EXEC master.dbo.sp_addlinkedserver
@rmtsrvname=N'Oracle_link2',
@useself=N'False',
@locallogin=NULL,
@rmtuser=N'master',
@rmtpassword='Test#1234'
GO
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

停用連結的伺服器搭配 Oracle

若要停用連結的伺服器搭配 Oracle，請將 OLEDB_ORACLE 選項從其選項群組中移除。

Important

移除選項並不會刪除資料庫執行個體上現有的連結伺服器組態。您必須手動捨棄它們，才能將它們從資料庫執行個體中移除。

您可以在移除後重新啟用 OLEDB_ORACLE 選項，以重複使用先前在資料庫執行個體上設定的連結伺服器組態。

主控台

下列程序會移除 OLEDB_ORACLE 選項。

從選項群組中移除 OLEDB_ORACLE 選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。

2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇具有 OLEDB_ORACLE 選項的選項群組 (上述範例中的 `oracle-oledb-se-2019`)。
4. 選擇 Delete option (刪除選項)。
5. 在 Deletion options (刪除選項) 下，為 Options to delete (要刪除的選項) 選擇 OLEDB_ORACLE。
6. 在 Apply immediately (立即套用) 下，選擇 Yes (是) 立即刪除選項，或選擇 No (否) 在下一個維護時段將其刪除。
7. 選擇刪除。

CLI

下列程序會移除 OLEDB_ORACLE 選項。

從選項群組中移除 OLEDB_ORACLE 選項

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name oracle-oledb-se-2019 \  
  --options OLEDB_ORACLE \  
  --apply-immediately
```

在Windows中：

```
aws rds remove-option-from-option-group ^  
  --option-group-name oracle-oledb-se-2019 ^  
  --options OLEDB_ORACLE ^  
  --apply-immediately
```

SQL Server 對原生備份與還原的支援

您可以使用 SQL Server 資料庫的原生備份與還原功能，建立內部部署資料庫的差異或完整備份，並於 Amazon S3 上存放備份檔案。然後您可以還原到執行 SQL Server 的現有 Amazon RDS 資料庫執行個體。您還可以備份 RDS for SQL Server 資料庫，將其存放於 Amazon S3，並於其他位置將其還原。另外，您可以將備份還原至現場部署伺服器，或執行 SQL Server 的不同 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 [使用原生備份和還原，匯入和匯出 SQL Server 資料庫](#)。

Amazon RDS 支援使用差異和完整備份檔案 (.bak 檔案) 進行 Microsoft SQL Server 資料庫的原生備份與還原。

新增原生備份與還原選項

將原生備份與還原選項新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將 SQLSERVER_BACKUP_RESTORE 選項新增至選項群組。
3. 建立 AWS Identity and Access Management (IAM) 角色與此選項的關聯。IAM 角色必須具有 S3 儲存貯體的存取權限才能儲存資料庫備份。

也就是說，該選項必須設定為 `arn:aws:iam::account-id:role/role-name` 格式的有效 Amazon 資源名稱 (ARN)。如需詳細資訊，請參閱 AWS 一般參考 中的 [Amazon Resource Name \(ARN\)](#)。

IAM 角色也必須與信任關係及許可政策相連接。信任關係可允許 RDS 擔任角色，而許可政策會定義角色可以執行的動作。如需更多詳細資訊，請參閱 [手動建立原生備份與還原的 IAM 角色](#)。

4. 將選項群組與資料庫執行個體建立關聯。

在您新增原生備份與還原選項之後，您並不需要重新啟動資料庫執行個體。只要選項群組為作用中狀態，您就可以立即開始備份與還原。

主控台

新增原生備份與還原選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。

3. 建立新的選項群組或使用現有的選項群組。如需如何建立自訂資料庫選項群組的資訊，請參閱[建立選項群組](#)。

若要使用現有的選項群組，請跳到下一個步驟。

4. 將 SQLSERVER_BACKUP_RESTORE 選項新增至選項群組。如需新增選項的詳細資訊，請參閱[將選項新增至選項群組](#)。
5. 請執行下列其中一項：
 - 若要使用現有的 IAM 角色和 Amazon S3 設定，請為 IAM Role (IAM 角色) 選擇現有的 IAM 角色。如果您使用現有的 IAM 角色，RDS 會使用為此角色設定的 Amazon S3 設定。
 - 若要建立新的角色並設定新的 Amazon S3 設定，請執行以下動作：
 1. 對於 IAM 角色，選擇建立新角色。
 2. 對於 S3 bucket (S3 儲存貯體)，從清單中選擇 S3 儲存貯體。
 3. 在 S3 prefix (optional) (S3 字首 (選用)) 中，指定 Amazon S3 儲存貯體所儲存檔案要使用的字首。

該字首可以包含檔案路徑，但並非必要。如果您提供字首，RDS 會將該字首附加於所有備份檔案。然後，RDS 會在還原過程中使用該字首以識別相關檔案並忽略無關的檔案。例如，您可以將 S3 儲存貯體用於存放備份檔案以外的目的。在此情況下，您可以使用該字首讓 RDS 僅在特定資料夾及其子資料夾內執行原生備份和還原。

如果您讓字首保持空白，RDS 將不會使用字首來識別備份檔案或者要還原的檔案。因此，在多檔案還原期間，RDS 會嘗試還原 S3 儲存貯體的每個資料夾中的每個檔案。

4. 選擇 Enable encryption (啟用加密) 核取方塊以加密備份檔案。保持清除核取方塊 (預設值)，以使備份檔案未加密。

如果您選擇 Enable encryption (啟用加密)，請選擇 AWS KMS key 的加密金鑰。如需加密金鑰的詳細資訊，請參閱 AWS Key Management Service 開發人員指南 中的[入門](#)。

6. 選擇 Add option (新增選項)。
7. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

CLI

此程序會進行下列假設：

- 您正在將 `SQLSERVER_BACKUP_RESTORE` 選項新增到已存在的選項群組。如需新增選項的詳細資訊，請參閱[將選項新增至選項群組](#)。
- 您正在將選項與已存在的 IAM 角色建立關聯，並且可以存取存放備份的 S3 儲存貯體。
- 您正在將選項群組套用到已存在的資料庫執行個體。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

新增原生備份與還原選項

1. 將 `SQLSERVER_BACKUP_RESTORE` 選項新增至選項群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --apply-immediately \  
  --option-group-name mybackupgroup \  
  --options "OptionName=SQLSERVER_BACKUP_RESTORE, \  
    OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-  
name}]"
```

在Windows中：

```
aws rds add-option-to-option-group ^  
  --option-group-name mybackupgroup ^  
  --options "[{\\"OptionName\\": \\"SQLSERVER_BACKUP_RESTORE\\", ^  
  \\"OptionSettings\\": [{\\"Name\\": \\"IAM_ROLE_ARN\\", ^  
  \\"Value\\": \\"arn:aws:iam::account-id:role/role-  
  name"}]}]" ^  
  --apply-immediately
```

Note

使用 Windows 命令提示字元時，您必須在 JSON 程式碼中的雙引號 (") 開頭加上反斜線 (\)，以逸出雙引號。

2. 將選項群組套用到資料庫執行個體。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --option-group-name mybackupgroup \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --option-group-name mybackupgroup ^  
  --apply-immediately
```

修改原生備份與還原選項設定

您在啟用原生備份與還原選項之後，可以修改選項的設定。如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。

移除原生備份與還原選項

您可以關閉原生備份與還原功能，方法為從資料庫執行個體中移除該選項。在您移除原生備份與還原選項之後，您並不需要重新啟動資料庫執行個體。

若要從資料庫執行個體中移除原生備份與還原選項，請執行下列其中一個動作：

- 從其所屬的選項群組中移除選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱[從選項群組移除選項](#)。
- 修改資料庫執行個體，並指定不包括「原生備份與還原」選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

支援 SQL Server 的透明資料加密

Amazon RDS 支援使用透明資料加密 (TDE)，來加密執行 Microsoft SQL Server 之資料庫執行個體上所儲存的資料。TDE 會在資料寫入至儲存體之前自動將其加密，並在從儲存體中讀取資料時自動將其解密。

Amazon RDS 支援下列 SQL Server 版本的 TDE：

- SQL 伺服器 2022 年標準版和企業版
- SQL Server 2019 Standard 和 Enterprise Editions
- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2014 Enterprise Edition

SQL Server 的透明資料加密會使用雙層金鑰架構來提供加密金鑰管理。從資料庫主要金鑰產生的認證是用來保護資料加密金鑰。資料庫加密金鑰會在使用者資料庫上，實際執行資料加密及解密作業。Amazon RDS 會備份並管理資料庫主金鑰與 TDE 憑證。

透明資料加密係使用於需要加密敏感資料的情況。例如，您可能想要將資料檔案和備份提供給第三方，或處理與安全相關的法律合規性問題。您無法加密 SQL Server 的系統資料庫，例如 model 或 master 資料庫。

透明資料加密的詳細討論超出本指南的範圍，但請確定您了解每一個加密演算法和金鑰的安全優缺點。如需 SQL Server 之透明資料加密的相關資訊，請參閱 Microsoft 文件中的[透明資料加密 \(TDE\)](#)。

主題

- [為 RDS for SQL Server 開啟 TDE](#)
- [在 RDS for SQL Server 上加密資料](#)
- [在 RDS for SQL Server 上備份與還原 TDE 憑證](#)
- [備份與還原內部部署資料庫的 TDE 憑證](#)
- [為 RDS for SQL Server 關閉 TDE](#)

為 RDS for SQL Server 開啟 TDE

若要開啟 RDS for SQL Server 資料庫執行個體的透明資料加密，請在與該資料庫執行個體關聯的 RDS 選項群組中指定 TDE 選項。

1. 判斷資料庫執行個體是否已與具有 TDE 選項的選項群組相關聯。若要檢視與資料庫執行個體相關聯的選項群組，請使用 RDS 主控台、[describe-db-instance](#) AWS CLI 命令或 API 作業 [DescribeDBInstances](#)。
2. 如果資料庫執行個體並未與 TED 已開啟的選項群組關聯，則您有兩個選擇。您可以建立一個選項群組，並新增 TDE 選項，您也可以修改相關的選項群組，以增加該選項。

Note

在 RDS 主控台中，此選項會命名為 `TRANSPARENT_DATA_ENCRYPTION`。在 AWS CLI 和 RDS API 中，它命名為 `TDE`。

如需建立或修改選項群組的相關資訊，請參閱 [使用選項群組](#)。如需將選項新增至選項群組的相關資訊，請參閱 [將選項新增至選項群組](#)。

3. 建立資料庫執行個體與具有 TDE 選項之選項群組之間的關聯。如需使資料庫執行個體與選項群組產生關聯的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

選項群組考量

TDE 選項是一個永續性選項。除非所有資料庫執行個體和備份均不再與選項群組相關聯，否則您無法將它從該選項群組中移除。一旦您將 TDE 選項新增至選項群組，該選項群組就只能與使用 TDE 的資料庫執行個體相關聯。如需選項群組中持續選項的詳細資訊，請參閱 [選項群組概觀](#)。

因為 TDE 選項為持久性選項，所以在選項群組與相關聯的資料庫執行個體之間可能會發生衝突。在下列情況中會發生衝突：

- 目前選項群組具有 TDE 選項，而且您可將其取代為沒有 TDE 選項的選項群組。
- 您可以從資料庫快照還原至新的資料庫執行個體，其中沒有包含 TDE 選項的選項群組。如需此案例的詳細資訊，請參閱 [選項群組考量](#)。

SQL Server 效能考量

使用透明資料加密會影響 SQL Server 資料庫執行個體的效能。

如果資料庫位在至少有一個已加密資料庫的資料庫執行個體上，則未加密資料庫的效能會降低。因此，建議您將已加密和未加密資料庫保存在獨立的資料庫執行個體上。

在 RDS for SQL Server 上加密資料

一旦將 TDE 選項新增至選項群組，Amazon RDS 即會產生在加密程序中使用的憑證。然後，您可以使用憑證執行 SQL 陳述式，將資料庫執行個體上資料庫中的資料加密。

下列範例會使用 RDS 建立的憑證 (稱為 `RDSTDECertificateName`)，來加密稱為 `myDatabase` 的資料庫。

```
----- Turning on TDE -----  
  
-- Find an RDS TDE certificate to use  
USE [master]  
GO  
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'  
GO  
  
USE [myDatabase]  
GO  
-- Create a database encryption key (DEK) using one of the certificates from the  
  previous step  
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256  
  ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]  
GO  
  
-- Turn on encryption for the database  
ALTER DATABASE [myDatabase] SET ENCRYPTION ON  
GO  
  
-- Verify that the database is encrypted  
USE [master]  
GO  
SELECT name FROM sys.databases WHERE is_encrypted = 1  
GO  
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys  
GO
```

使用 TDE 加密 SQL Server 資料庫需要的時間取決於幾個因素。其中包括資料庫執行個體的大小、執行個體是否使用佈建 IOPS 儲存體、資料的量，以及其他因素。

在 RDS for SQL Server 上備份與還原 TDE 憑證

RDS for SQL Server 提供預存程序，可供備份、還原與捨棄 TDE 憑證。RDS for SQL Server 也提供一個函數可用於檢視還原的使用者 TDE 憑證。

使用者 TDE 憑證用於將資料庫還原至內部部署且 TDE 已開啟的 RDS for SQL Server。這些憑證的字首為 `UserTDECertificate_`。還原資料庫之後，及使其可供使用之前，RDS 會修改 TDE 已開啟的資料庫，以使用 RDS 產生的 TDE 憑證。這些憑證的字首為 `RDSTDECertificate`。

使用者 TDE 憑證會保留在 RDS for SQL Server 資料庫執行個體上，除非您使用 `rds_drop_tde_certificate` 預存程序放置這些憑證。如需詳細資訊，請參閱[捨棄還原的 TDE 憑證](#)。

您可以使用使用者 TDE 憑證從來源資料庫執行個體還原其他資料庫。要還原的資料庫必須使用相同的 TDE 憑證並開啟 TDE。您不必再次匯入 (還原) 同一個憑證。

主題

- [必要條件](#)
- [限制](#)
- [備份 TDE 憑證](#)
- [還原 TDE 憑證](#)
- [檢視還原的 TDE 憑證](#)
- [捨棄還原的 TDE 憑證](#)

必要條件

務必先完成下列任務，您才能在 RDS for SQL Server 上備份或還 TDE 憑證。[設定原生備份與還原](#) 中描述前三個。

1. 建立 Amazon S3 儲存貯體，以用於儲存要備份和還原的檔案。

建議您為資料庫備份與 TDE 憑證備份使用個別的儲存貯體。

2. 建立用於備份和還原檔案的 IAM 角色。

IAM 角色必須是 AWS KMS key 的使用者，也是其管理員。

除了 SQL Server 原生備份和還原所需的許可之外，IAM 角色還需要以下許可：

- S3 儲存貯體資源上的 `s3:GetBucketACL`、`s3:GetBucketLocation` 和 `s3:ListBucket`
 - * 資源上的 `s3:ListAllMyBuckets`
3. 將 `SQLSERVER_BACKUP_RESTORE` 選項新增至資料庫執行個體上的選項群組。

這是 `TRANSPARENT_DATA_ENCRYPTION` (TDE) 選項新增的部分。

4. 確定您有對稱加密 KMS 金鑰。您有下列選項：

- 如果您的帳戶中有現有的 KMS 金鑰，您可以使用它。無需採取進一步動作。
- 若您的帳戶中並無現有的對稱加密 KMS 金鑰，請藉由遵循《AWS Key Management Service 開發人員指南》中的[建立金鑰](#)來建立 KMS 金鑰。

5. 啟用 Amazon S3 整合，在資料庫執行個體與 Amazon S3 之間傳輸檔案。

如需啟用 Oracle Amazon S3 整合的詳細資訊，請參閱 [整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

限制

使用預存程序備份和還原 TDE 憑證具有以下限制：

- SQLSERVER_BACKUP_RESTORE 與 TRANSPARENT_DATA_ENCRYPTION (TDE) 選項必須新增至與資料庫執行個體產生關聯的選項群組。
- 多可用區域資料庫執行個體不支援 TDE 憑證備份和還原。
- 不支援取消 TDE 憑證備份和還原任務。
- 您不能將使用者 TDE 憑證用於 RDS for SQL Server 資料庫執行個體上任何其他資料庫的 TDE 加密。您可以將它用於從 TDE 已啟用且使用相同 TDE 憑證的來源資料庫執行個體中僅還原其他資料庫。
- 您只能刪除使用者 TDE 憑證。
- RDS 支援的使用者 TDE 憑證數量上限為 10 個。如果數量超過 10 個，請捨棄未使用的 TDE 憑證，然後再試一次。
- 憑證名稱不能為空白或為 null。
- 還原憑證時，憑證名稱不能包含關鍵字 RDSTDECERTIFICATE，且開頭必須為 UserTDECertificate_ 字首。
- @certificate_name 參數只能包含以下字元：a-z、0-9、@、\$、# 和底線 (_)。
- @certificate_file_s3_arn 的副檔名必須為 .cer (不區分大小寫)。
- @private_key_file_s3_arn 的副檔名必須為 .pvk (不區分大小寫)。
- 私有金鑰檔案的 S3 中繼資料必須包含 x-amz-meta-rds-tde-pwd 標籤。如需詳細資訊，請參閱[備份與還原內部部署資料庫的 TDE 憑證](#)。

備份 TDE 憑證

若要備份 TDE 憑證，請使用 rds_backup_tde_certificate 預存程序。具有下列語法。

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='UserTDECertificate_< i>certificate_name |
RDSTDECertificate< i>timestamp',
    @certificate_file_s3_arn='arn:aws:s3:::< i>bucket_name/< i>certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::< i>bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:< i>region:< i>account-id:key/< i>key-id',
    [@overwrite_s3_files=< i>0|1];
```

下列是必要參數：

- @certificate_name – 所要備份 TDE 憑證的名稱。
- @certificate_file_s3_arn – Amazon S3 中憑證備份檔案的目的地 Amazon Resource Name (ARN)。
- @private_key_file_s3_arn – 可保護 TDE 憑證的私有金鑰檔案的目的地 S3 ARN。
- @kms_password_key_arn – 用於加密私有金鑰密碼的對稱 KMS 金鑰其 ARN。

以下參數為選用：

- @overwrite_s3_files – 指出在 S3 中是否覆寫現有的憑證與私有金鑰檔案：
 - 0 – 不覆寫現有檔案。此為預設值。
- 將 @overwrite_s3_files 設為 0 時，如果檔案已存在，則傳回錯誤。
- 1 – 即使現有檔案不是備份檔案，只要是指定的名稱，就覆寫檔案。

Example 備份 TDE 憑證

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='RDSTDECertificate20211115T185333',
    @certificate_file_s3_arn='arn:aws:s3:::< i>TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::< i>TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:< i>us-
west-2:123456789012:key/< i>AKIAIOSFODNN7EXAMPLE',
    @overwrite_s3_files=< i>1;
```

還原 TDE 憑證

您可以使用 rds_restore_tde_certificate 預存程序來還原 (匯入) 使用者 TDE 憑證。具有下列語法。

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_< i>certificate_name',
    @certificate_file_s3_arn='arn:aws:s3:::< i>bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::< i>bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:< i>region:account-id:key/key-id';
```

下列是必要參數：

- @certificate_name – 所要還原 TDE 憑證的名稱。名稱必須以 UserTDECertificate_ 字首開頭。
- @certificate_file_s3_arn – 用於還原 TDE 憑證的備份檔案其 S3 ARN。
- @private_key_file_s3_arn – 待還原的 TDE 憑證其私有金鑰備份檔案的 S3 ARN。
- @kms_password_key_arn – 用於加密私有金鑰密碼的對稱 KMS 金鑰其 ARN。

Example 還原 TDE 憑證

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_< i>myTDEcertificate',
    @certificate_file_s3_arn='arn:aws:s3:::< i>TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::< i>TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:< i>us-west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

檢視還原的 TDE 憑證

您可以使用 rds_fn_list_user_tde_certificates 函數來檢視已還原 (已匯入) 的使用者 TDE 憑證。具有下列語法。

```
SELECT * FROM msdb.dbo.rds_fn_list_user_tde_certificates();
```

輸出結果與以下內容相似。並非所有的資料欄均會在此顯示。

name	certif	princi	pvt_ke	issuer	cert_s	thumbp	subjec	start_	expiry	pvt_key_l
	te_id	_id	ncrypt	me	al_num	t		e	te	ast_backu
			type							p_date
			c							

UserTD	343	1	ENCRYF	AnyCorr	79	0x6BB2	AnyCorr	2022-0	2023-0	NULL
rtific			_BY_MA	y	3e	341103	y	5	5	
_tde_c			R_KEY	Shippi	57	80B	Shippi	19:49:	19:49:	
					a3	FE1BA2		000000	000000	
					69	C69509				
					fd	5B5				
					1d					
					9e					
					47					
					2c					
					32					
					67					
					1d					
					9c					
					ca					
					af					

捨棄還原的 TDE 憑證

若要捨棄您未使用的已還原 (已匯入) 使用者 TDE 憑證，請使用 `rds_drop_tde_certificate` 預存程序。具有下列語法。

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
@certificate_name='UserTDECertificate_certificate_name';
```

以下是必要參數：

- `@certificate_name` – 所要捨棄 TDE 憑證的名稱。

您只能捨棄已還原 (已匯入) 的 TDE 憑證。您不能捨棄 RDS 建立的憑證。

Example 捨棄 TDE 憑證

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
@certificate_name='UserTDECertificate_myTDECertificate';
```

備份與還原內部部署資料庫的 TDE 憑證

您可以備份內部部署資料庫的 TDE 憑證，日後再將憑證還原到 RDS for SQL Server。您還可以將 RDS for SQL Server TDE 憑證還原到內部部署資料庫執行個體。

下列程序會備份 TDE 憑證和私有金鑰。系統會使用從對稱加密 KMS 金鑰產生的資料金鑰來加密私有金鑰。

備份內部部署 TDE 憑證

1. 使用 AWS CLI `generate-data-key` 指令產生資料金鑰。

```
aws kms generate-data-key \  
  --key-id my_KMS_key_ID \  
  --key-spec AES_256
```

輸出結果與以下內容相似。

```
{  
  "CiphertextBlob": "AQIDAHimL2NEoAl0Y6Bn7LJfnxi/0Ze9kTQo/  
XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B  
BwagbzBtAgEAMGgGCSqGSIB3DQEHAeBgIghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vetng  
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==",  
  "Plaintext": "U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/a0n939ZnKi/0=",  
  "KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-00ee-99ff-88dd-  
aa11bb22cc33"  
}
```

您在下一步可以使用純文字輸出作為私有金鑰密碼。

2. 備份 TDE 憑證，如下列範例所示。

```
BACKUP CERTIFICATE myOnPremTDEcertificate TO FILE = 'D:\tde-cert-backup.cer'  
WITH PRIVATE KEY (  
FILE = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\cert-  
backup-key.pvk',  
ENCRYPTION BY PASSWORD = 'U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/a0n939ZnKi/0=');
```

3. 將憑證備份檔案儲存至 Amazon S3 憑證儲存貯體。
4. 將私有金鑰備份檔案儲存到 S3 憑證儲存貯體，並在檔案的中繼資料中使用以下標籤：
 - 索引鍵 – x-amz-meta-rds-tde-pwd

- 值 – 來自產生資料金鑰的 CiphertextBlob 值，如下列範例所示。

```
AQIDAHimL2NEoA10Y6Bn7LJfnxi/0Ze9kTQo/
XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBg1ghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vet
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==
```

以下程序會將 RDS for SQL Server TDE 憑證還原至內部部署資料庫執行個體。您可以使用憑證備份、對應的私有金鑰檔案與資料金鑰，在目的地資料庫執行個體上複製與還原 TDE 憑證。還原的憑證由新伺服器的資料庫主要金鑰加密。

還原 TDE 憑證

1. 將 TDE 憑證備份檔案和私有金鑰檔案從 Amazon S3 複製到目的地執行個體。如需從 Amazon S3 複製檔案的詳細資訊，請參閱 [在 RDS for SQL Server 與 Amazon S3 之間傳輸檔案](#)。
2. 使用 KMS 金鑰解密輸出密碼文字，以擷取資料金鑰的純文字。密碼文字位於私有金鑰備份檔案的 S3 中繼資料中。

```
aws kms decrypt \
  --key-id my_KMS_key_ID \
  --ciphertext-blob fileb://exampleCiphertextFile | base64 -d \
  --output text \
  --query Plaintext
```

您在下一步可以使用純文字輸出作為私有金鑰密碼。

3. 使用下列 SQL 命令還原 TDE 憑證。

```
CREATE CERTIFICATE myOnPremTDEcertificate FROM FILE='D:\tde-cert-backup.cer'
WITH PRIVATE KEY (FILE = N'D:\tde-cert-key.pvk',
DECRYPTION BY PASSWORD = 'plain_text_output');
```

如需 KMS 解密的詳細資訊，請參閱 AWS CLI 命令參考中 KMS 一節的[解密](#)。

在目的地資料庫執行個體上還原 TDE 憑證後，您可以使用該憑證還原加密的資料庫。

Note

您可以使用相同的 TDE 憑證來加密來源資料庫執行個體上的多個 SQL Server 資料庫。若要將多個資料庫遷移至目的地執行個體，請僅將與這些資料庫關聯的 TDE 憑證複製到目的地執行個體一次。

為 RDS for SQL Server 關閉 TDE

若要關閉 RDS for SQL Server 資料庫執行個體的 TDE，首先請確定沒有任何加密的物件留在資料庫執行個體上。若要這麼做，請解密或捨棄這些物件。如果資料庫執行個體中存在加密物件，則您無法關閉資料庫執行個體的 TDE 功能。當您使用主控台從選項群組中移除 TDE 選項時，主控台會顯示它正在處理。此外，如果選項群組與加密資料庫執行個體或資料庫快照有關，則會建立一次錯誤事件。

下列範例會從稱為 `customerDatabase` 的資料庫中移除 TDE 加密。

```
----- Removing TDE -----  
  
USE [customerDatabase]  
GO  
  
-- Turn off encryption of the database  
ALTER DATABASE [customerDatabase]  
SET ENCRYPTION OFF  
GO  
  
-- Wait until the encryption state of the database becomes 1. The state is 5  
  (Decryption in progress) for a while  
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys  
GO  
  
-- Drop the DEK used for encryption  
DROP DATABASE ENCRYPTION KEY  
GO  
  
-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated  
USE [master]  
GO  
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE  
GO
```

所有物件都解密後，會有兩個選擇：

1. 您可以將資料庫執行個體修改為關聯至不具有 TDE 選項的選項群組。
2. 您可以從選項群組中移除 TDE 選項。

SQL Server 稽核

在 Amazon RDS 中，您可以使用內建的 SQL Server 稽核機制來稽核 Microsoft SQL Server 資料庫。您可以像在現場部署資料庫伺服器上一樣建立稽核與稽核規格。

RDS 會使用您提供的 IAM 角色，將完成的稽核日誌上傳到 S3 儲存貯體。如果您啟用了保留，RDS 會在設定的期間內將稽核日誌保存於資料庫執行個體上。

如需詳細資訊，請參閱 Microsoft SQL Server 文件中的 [SQL Server 稽核 \(資料庫引擎\)](#)。

使用資料庫活動串流的 SQL Server 稽核

您可以使用適用於 RDS 的資料庫活動串流，McAfee 將 SQL Server 稽核事件與 Imperva 和 IBM 的資料庫活動監視工具整合在一起。如需詳細了解以資料庫活動串流稽核 RDS SQL Server，請參閱 [在 Microsoft SQL Server 中進行稽核](#)

主題

- [對 SQL Server 稽核的支援。](#)
- [將 SQL Server 稽核新增至資料庫執行個體選項](#)
- [使用 SQL Server 稽核](#)
- [檢視稽核日誌](#)
- [將 SQL Server 稽核與多個可用區執行個體搭配使用](#)
- [設定 S3 儲存貯體](#)
- [手動建立適用於 SQL Server 稽核的 IAM 角色](#)

對 SQL Server 稽核的支援。

在 Amazon RDS 中，從 SQL Server 2014 開始，SQL Server 的所有版本均支援伺服器層級稽核，企業版還支援資料庫層級稽核。從 SQL Server 2016 (13.x) SP1 開始，所有版本均支援伺服器層級和資料庫層級稽核。如需更多詳細資訊，請參閱 SQL Server 文件中的 [SQL Server 稽核 \(資料庫引擎\)](#)。

RDS 支援對下列 SQL Server 稽核選項進行設定。

選項設定	有效值	描述
IAM_ROLE_ARN	有效的 Amazon Resource Name (ARN)，格式為	IAM 角色的 ARN，該角色授予對您想要儲存稽核日誌的 S3

選項設定	有效值	描述
	arn:aws:iam:: <i>account-id</i> :role/ <i>role-name</i> 。	儲存貯體的存取權。如需詳細資訊，請參閱 AWS 一般參考中的 Amazon Resource Name (ARN) 。
S3_BUCKET_ARN	arn:aws:s3::: <i>DOC-EXAMPLE-BUCKET</i> 或 arn:aws:s3::: <i>DOC-EXAMPLE-BUCKET</i> /key-prefix 格式的有效 ARN	您想要儲存稽核日誌的 S3 儲存貯體的 ARN。
ENABLE_COMPRESSION	true 或 false *	控制稽核日誌壓縮。預設會啟用壓縮 (設定為 true)。
RETENTION_TIME	0 至 840	SQL Server 稽核記錄保存於 RDS 執行個體上的保留時間 (以小時計)。預設會停用保留。

將 SQL Server 稽核新增至資料庫執行個體選項

啟用 SQL Server 稽核需要兩個步驟：啟用資料庫執行個體上的選項，以及啟用 SQL Server 內的功能。將 SQL Server 稽核選項新增至資料庫執行個體的過程如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 新增和設定所有必要的選項。
3. 將選項群組與資料庫執行個體建立關聯。

新增 SQL Server 稽核 選項後，並不需要重新啟動資料庫執行個體。只要選項群組處於作用中狀態，您就可以建立稽核，並將稽核日誌儲存於 S3 儲存貯體。

若要在資料庫執行個體的選項群組中新增和設定 SQL Server 稽核

1. 選擇下列其中一項：
 - 使用現有的選項群組。
 - 建立自訂資料庫選項群組，並使用該選項群組。如需更多詳細資訊，請參閱 [建立選項群組](#)。

- 將 SQLSERVER_AUDIT 選項新增至選項群組，並進行選項設定。如需新增選項的詳細資訊，請參閱[將選項新增至選項群組](#)。
 - 在 IAM role (IAM 角色) 中，如果您已經擁有具備所需原則的 IAM 角色，您可以選取該角色。若要建立新的 IAM 角色，請選擇 Create a New Role (建立新角色)。如需必要政策的資訊，請參閱[手動建立適用於 SQL Server 稽核的 IAM 角色](#)。
 - 在 Select S3 destination (選擇 S3 目的地) 中，如果您已經有想要使用的 S3 儲存貯體，請選取它。若要建立 S3 儲存貯體，請選擇 Create a New S3 Bucket (建立新的 S3 儲存貯體)。
 - 在 Enable Compression (啟用壓縮) 中，請勾選此選項，以壓縮稽核檔案。預設會啟用壓縮。若要停用壓縮，請取消勾選 Enable Compression (啟用壓縮)。
 - 在 Audit log retention (稽核日誌保留) 中，若要將稽核記錄保留在資料庫執行個體上，請選擇此選項。指定以小時計的停留時間。最長的停留時間是 35 天。
- 將選項群組套用至新的或現有的資料庫執行個體。選擇下列其中一項：
 - 如果您建立了新的資料庫執行個體，在啟動執行個體時套用選項群組。
 - 針對現有的資料庫執行個體，可以修改執行個體，並附加新的選項群組，以套用選項群組。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

修改 SQL Server 稽核選項

啟用 SQL Server 稽核選項之後，您可以修改設定。如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。

從資料庫執行個體選項中移除 SQL Server 稽核

您可以停用稽核，然後刪除該選項，以關閉 SQL Server 稽核功能。

若要移除稽核

- 停用 SQL Server 內的所有稽核設定。若要瞭解稽核執行到哪裡，請查詢 SQL Server 安全目錄檢視。如需詳細資訊，請參閱 SQL Server 文件中的[安全目錄檢視](#)。
- 從資料庫執行個體刪除 SQL Server 稽核選項 選擇下列其中一項：
 - 從資料庫執行個體使用的選項群組中刪除 SQL Server 稽核選項。此變更會影響使用同一選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱[從選項群組移除選項](#)。
 - 修改資料庫執行個體，然後選擇一個沒有 SQL Server 稽核選項的選項群組。此一變更只會影響您修改的資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

3. 從資料庫執行個體刪除 SQL Server 稽核選項後，不需要重新啟動執行個體。從 S3 儲存貯體移除不需要的稽核檔案。

使用 SQL Server 稽核

您可以使用和控制現場部署資料庫伺服器一樣的方式，控制伺服器稽核、伺服器稽核規格和資料庫稽核規格。

建立稽核

您可以使用與現場部署資料庫伺服器相同的方式建立伺服器稽核。如需如何建立伺服器稽核的資訊，請參閱 Microsoft SQL Server 文件中的 [CREATE SERVER AUDIT](#)。

為了避免錯誤，請遵守以下限制：

- 切勿超過每部執行個體所支援伺服器稽核的最大數量 – 50。
- 指示 SQL Server 將資料寫到二進位檔案。
- 伺服器稽核名稱的字首不要使用 RDS_。
- 在 FILEPATH，請指定 D:\rdsdbdata\SQLAudit。
- 在 MAXSIZE 中，指定介於 2 MB 和 50 MB 之間的大小。
- 不要設定 MAX_ROLLOVER_FILES 或 MAX_FILES。
- 如果 SQL Server 無法寫入稽核記錄，請勿設定讓它關閉資料庫執行個體。

建立稽核規格

您使用與現場部署資料庫伺服器相同的方式建立伺服器稽核規格和資料庫稽核規格。如需建立稽核規格的資訊，請參閱 Microsoft SQL Server 文件中的 [CREATE SERVER AUDIT SPECIFICATION](#) 和 [CREATE DATABASE AUDIT SPECIFICATION](#)。

為了避免錯誤，資料庫稽核規格或伺服器稽核規格名稱中的字首請不要使用 RDS_。

檢視稽核日誌

稽核日誌被儲存於 D:\rdsdbdata\SQLAudit。

SQL Server 完成稽核日誌檔案的寫入後 — 當檔案到達大小限制時 — Amazon RDS 會將檔案上傳到 S3 儲存貯體。如果啟用了保留，Amazon RDS 會將檔案移動到保留資料夾內：D:\rdsdbdata\SQLAudit\transmitted。

如需設定保留的相關資訊，請參閱 [將 SQL Server 稽核新增至資料庫執行個體選項](#)。

稽核記錄會保存在資料庫執行個體上，直到稽核日誌檔案被上傳為止。您可以執行下列命令，以查看稽核記錄。

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
      ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
      , default
      , default )
```

您可以使用相同的命令，將篩選器變更為 D:\rdsdbdata\SQLAudit\transmitted*.sqlaudit，以查看保留資料夾中的稽核記錄。

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
      ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
      , default
      , default )
```

將 SQL Server 稽核與多個可用區執行個體搭配使用

對於多可用區執行個體，將稽核日誌檔案傳送到 Amazon S3 的程序與單一可用區執行個體類似。不過，還是有一些重要的差異。

- 資料庫稽核規格物件會被複寫到所有節點。
- 伺服器稽核與伺服器稽核規格不會複寫到次要節點。反之，您必須手動建立或修改它們。

若要從兩個節點擷取伺服器稽核或伺服器稽核規格：

1. 在主要節點上建立伺服器稽核或伺服器稽核規格。
2. 容錯移轉到次要節點，同時在次要節點上建立具有相同名稱和 GUID 的伺服器稽核或伺服器稽核規格。使用 AUDIT_GUID 參數指定 GUID。

設定 S3 儲存貯體

稽核日誌檔案會自動從資料庫執行個體上傳到 S3 儲存貯體。下列限制適用於使用為稽核檔案目標的 S3 儲存貯體：

- 它必須與資料庫執行個體位於相同的 AWS 區域。
- 它不可以開放給大眾使用。
- 儲存貯體擁有者也必須是 IAM 角色擁有者。

用來儲存資料的目標索引鍵遵循此一命名結構描述：*DOC-EXAMPLE-BUCKET*/key-prefix/
instance-name/audit-name/node_file-name.ext

Note

您以 (S3_BUCKET_ARN) 選項設定來設定儲存貯體名稱和金鑰前綴的值。

結構描述由下列元素組成：

- **#####** - 您的 S3 存儲桶的名稱。
- **key-prefix** – 您想要使用於稽核日誌的自訂索引鍵字首。
- **instance-name** – Amazon RDS 執行個體的名稱。
- **audit-name** – 稽核的名稱。
- **node** – 節點的識別符，該節點是稽核日誌的來源 (node1 或 node2)。單一可用區執行個體有一個節點，多可用區執行個體則有兩個複寫節點。這些節點並不是主要和次要節點，因為主要和次要角色會隨著時間而改變。反之，節點識別符只是一個簡單的標籤。
 - **node1** – 第一個複寫節點 (單一可用區只有一個節點)。
 - **node2** – 第二個複寫節點 (多可用區有兩個節點)。
- **file-name** – 目標檔案名稱。檔案名稱係從 SQL Server 依現狀取得。
- **ext** – 檔案的副檔名 (zip 或 sqlaudit):
 - **zip** – 如果啟用了壓縮 (預設)。
 - **sqlaudit** – 如果停用了壓縮。

手動建立適用於 SQL Server 稽核的 IAM 角色

通常，當您建立新選項時，AWS Management Console 會為您建立 IAM 角色和 IAM 信任政策。然而，您可以手動建立新的 IAM 角色，並使用於 SQL Server 稽核，以便使用您可能有的其他要求自訂該角色。若要這麼做，請建立一個 IAM 角色，並委派許可，以便 Amazon RDS 服務可以使用 Amazon S3 儲存貯體。當您建立 IAM 角色時，您需要附加信任和許可政策。信任政策允許 Amazon RDS 擔

任此角色。許可政策定義此角色可以執行的動作。如需詳細資訊，請參閱 [《AWS Identity and Access Management 使用者指南》](#) 中的 [建立角色以將權限委派給 AWS 服務](#)。

您可以使用本節的範例建立您需要的信任關係和許可政策。

以下範例說明 SQL Server Audit 的信任關係。該政策會使用服務主體 `rds.amazonaws.com`，以允許 RDS 寫入 S3 儲存貯體。「服務委託人」是用來將許可授予給服務的識別符。當您允許以此種方式存取 `rds.amazonaws.com` 時，就是允許 RDS 代表您執行動作。如需服務委託人的詳細資訊，請參閱 [AWS JSON 政策元素：委託人](#)。

Example SQL Server Audit 的信任關係

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

建議您在資源型信任關係中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰，將服務的許可限定於特定資來源。這是防止 [混淆代理人問題](#) 最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在信任關係中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 Amazon 資源名稱 (ARN)。針對 SQL Server Audit，請確認同時包括資料庫選項群組和資料庫執行個體，如以下範例所示。

Example SQL Server Audit 的全域條件內容索引鍵信任關係

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
            "arn:aws:rds:Region:my_account_ID:og:option_group_name"
          ]
        }
      }
    }
  ]
}
```

在以下 SQL Server Audit 的許可政策範例中，我們指定了 Amazon S3 儲存貯體的 ARN。您可以使用 ARN 識別您想要授予存取權的特定帳戶、使用者或角色。如需使用 ARN 的詳細資訊，請參閱 [Amazon 資源名稱 \(ARN\)](#)。

Example SQL Server Audit 的許可政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketACL",
        "s3:GetBucketLocation"
      ]
    }
  ]
}
```

```
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/key_prefix/*"
  }
]
```

Note

驗證相同 AWS 帳戶是否同時擁有 S3 儲存貯體和 SQL Server 資料庫

執行 `s3:ListAllMyBuckets` 行個體時，需要執行此動作。此動作會列出帳戶中儲存貯體的名稱。

S3 儲存貯體命名空間是全域的。如果您不小心刪除了儲存貯體，其他使用者可在不同帳戶中建立相同名稱的儲存貯體。然後 SQL Server 稽核資料將會寫入新的儲存貯體。

支援 Amazon RDS for SQL Server 中的 SQL Server Analysis Services

Microsoft SQL Server Analysis Services (SSAS) 是 Microsoft 商業智慧 (MSBI) 套件的一部分。SSAS 是安裝在 SQL Server 內的線上分析處理 (OLAP) 和資料採礦工具。您可以使用 SSAS 來分析資料，以協助您做出商業決策。SSAS 與 SQL Server 關聯式資料庫不同，因為 SSAS 是針對商業智慧環境中常見的查詢和計算最佳化。

您可以在現有或新的資料庫執行個體上開啟 SSAS。它安裝在與資料庫引擎相同的資料庫執行個體上。如需 SSAS 的詳細資訊，請參閱 Microsoft [Analysis Services 文件](#)。

Amazon RDS 的下列版本支援 SSAS for SQL Server Standard 和 Enterprise Edition：

- 表格模式：
 - SQL Server 2019，15.00.4043.16.v1 版和更新版本
 - SQL Server 2017，14.00.3223.3.v1 版和更新版本
 - SQL Server 2016，13.00.5426.0.v1 版和更新版本
- 多維度模式：
 - SQL 伺服器 2019, 版本 15.00.4153.1.1 及更新版本
 - SQL Server 2017，14.00.3381.3.v1 版和更新版本
 - SQL Server 2016，13.00.5882.1.v1 版和更新版本

內容

- [限制](#)
- [開啟 SSAS](#)
 - [建立 SSAS 的選項群組。](#)
 - [將 SSAS 選項新增至選項群組](#)
 - [將選項群組關聯至您的資料庫執行個體](#)
 - [允許對 VPC 安全群組進行傳入存取](#)
 - [啟用 Amazon S3 整合](#)
- [在 Amazon RDS 上部署 SSAS 專案](#)
- [監視部署任務的狀態](#)
- [在 Amazon RDS 上使用 SSAS](#)
 - [為 SSAS 設定通過 Windows 驗證的使用者](#)
 - [新增網域使用者做為資料庫管理員](#)

- [建立 SSAS Proxy](#)
- [使用 SQL Server Agent 排程 SSAS 資料庫處理作業](#)
- [從 Proxy 撤銷 SSAS 存取權](#)
- [備份 SSAS 資料庫](#)
- [還原 SSAS 資料庫](#)
 - [將資料庫執行個體還原至指定的時間](#)
- [變更 SSAS 模式](#)
- [關閉 SSAS](#)
- [疑難排解 SSAS 問題](#)

限制

在 RDS for SQL Server 上使用 SSAS 時適用下列限制：

- RDS for SQL Server 支援在表格或多維度模式下執行 SSAS。如需詳細資訊，請參閱 Microsoft 說明文件中的[比較表格和多維度解決方案](#)在微軟檔案中。
- 一次只能使用一種 SSAS 模式。變更模式之前，請務必刪除所有 SSAS 資料庫。

如需詳細資訊，請參閱 [變更 SSAS 模式](#)。

- 不支援異地同步備份執行個體。
- 執行個體必須使用自我管理 AWS Directory Service for Microsoft Active Directory 的作用中目錄或 SSAS 驗證。如需詳細資訊，請參閱 [使用 Active Directory 搭配 RDS for SQL Server](#)。
- 使用者不會獲得 SSAS 伺服器系統管理員存取權，但可以獲得資料庫層級的系統管理員存取權。
- 唯一支援存取 SSAS 的連接埠是 2383。
- 您無法直接部署專案。我們提供 RDS 預存程序來執行此操作。如需更多詳細資訊，請參閱 [在 Amazon RDS 上部署 SSAS 專案](#)。
- 不支援在部署期間進行處理。
- 不支援使用 .xmla 檔案進行部署。
- SSAS 專案輸入檔案和資料庫備份輸出檔案只能位於資料庫執行個體的 D:\S3 資料夾中。

開啟 SSAS

使用下列程序為資料庫執行個體開啟 SSAS：

1. 建立新的選項群組或選擇現有的選項群組。
2. 將 SSAS 選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。
4. 允許傳入存取 SSAS 監聽器連接埠的 Virtual Private Cloud (VPC) 安全群組。
5. 開啟 Amazon S3 整合。

建立 SSAS 的選項群組。

使用 AWS Management Console 或建立對應 AWS CLI 至 SQL Server 引擎和您計劃使用的資料庫執行個體版本的選項群組。

Note

如果現有的選項群組適用於正確的 SQL Server 引擎和版本，則也可以使用。

主控台

下列主控台程序會建立 SQL Server Standard Edition 2017 的選項群組。

建立選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 窗格中，執行下列動作：
 - a. 在名稱中，輸入 AWS 帳戶中唯一的選項群組名稱，例如 **ssas-se-2017**。名稱僅可包含字母、數字與連字號。
 - b. 對於 Description (描述)，請輸入選項群組的簡短描述，例如 **SSAS option group for SQL Server SE 2017**。用於顯示用途的說明。
 - c. 對於 Engine (引擎)，請選擇 **sqlserver-se**。
 - d. 對於 Major engine version (主要引擎版本)，請選擇 **14.00**。
5. 選擇 Create (建立)。

CLI

下列 CLI 範例會為 SQL Server Standard Edition 2017 建立選項群組。

建立選項群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name ssas-se-2017 \  
  --engine-name sqlserver-se \  
  --major-engine-version 14.00 \  
  --option-group-description "SSAS option group for SQL Server SE 2017"
```

在 Windows 中：

```
aws rds create-option-group ^  
  --option-group-name ssas-se-2017 ^  
  --engine-name sqlserver-se ^  
  --major-engine-version 14.00 ^  
  --option-group-description "SSAS option group for SQL Server SE 2017"
```

將 SSAS 選項新增至選項群組

接下來，使用 AWS Management Console 或 AWS CLI 將SSAS選項新增至選項群組。

主控台

新增 SSAS 選項

- 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
- 在導覽窗格中，選擇 Option groups (選項群組)。
- 選擇您剛建立的選項群組。
- 選擇 Add option (新增選項)。
- 在 Option details (選項詳細資料) 下，選擇 SSAS 做為 Option name (選項名稱)。

6. 在 Option settings (選項設定) 下，執行下列動作：

- a. 為 Max memory (最大記憶體) 輸入範圍 10–80 間的值。


Max memory (最大記憶體) 指定 SSAS 開始更積極釋放記憶體的閾值上限，以便為正在執行的要求以及新的高優先順序要求騰出空間。這個數字是資料庫執行個體總記憶體的百分比。允許的值為 10–80，預設值是 45。

- b. 為 Mode (模式) 選擇 SSAS 伺服器模式 Tabular (表格) 或 Multidimensional (多維度)。

如果您沒有看到「模式」選項設定，表示您的 AWS 地區不支援多維模式。如需詳細資訊，請參閱 [限制](#)。

預設選項為 Tabular (表格)。

- c. 針對 Security groups (安全群組)，選擇要與選項產生關聯的 VPC 安全群組。

 Note

會預先填入存取 SSAS 的連接埠 (2383)。

7. 在 Scheduling (排程) 下，選擇要立即新增選項或是在下一個維護時段新增選項。

8. 選擇 Add option (新增選項)。

CLI

新增 SSAS 選項

1. 使用下列參數建立 JSON 檔案，例如 `ssas-option.json`：

- `OptionGroupName` – 您先前建立或選擇的選項群組名稱 (在下列範例中是 `ssas-se-2017`)。
- `Port` – 您用來存取 SSAS 的連接埠。唯一支援的連接埠是 2383。
- `VpcSecurityGroupMemberships` – RDS 資料庫執行個體的 VPC 安全群組成員資格。
- `MAX_MEMORY` – SSAS 應開始更積極釋放記憶體的閾值上限，以便為正在執行的要求以及新的高優先順序要求騰出空間。這個數字是資料庫執行個體總記憶體的百分比。允許的值為 10–80，預設值是 45。
- `MODE` – SSAS 伺服器模式 `Tabular` 或 `Multidimensional`。預設值為 `Tabular`。

如果您收到MODE選項設定無效的錯誤訊息，表示您的 AWS 地區不支援多維模式。如需詳細資訊，請參閱 [限制](#)。

以下是含有 SSAS 選項設定的 JSON 檔案範例。

```
{
  "OptionGroupName": "ssas-se-2017",
  "OptionsToInclude": [
    {
      "OptionName": "SSAS",
      "Port": 2383,
      "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
      "OptionSettings": [{"Name": "MAX_MEMORY", "Value": "60"},
        {"Name": "MODE", "Value": "Multidimensional"}]
    }
  ],
  "ApplyImmediately": true
}
```

2. 將 SSAS 選項新增至選項群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \
  --cli-input-json file://ssas-option.json \
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^
  --cli-input-json file://ssas-option.json ^
  --apply-immediately
```

將選項群組關聯至您的資料庫執行個體

您可以使用主控台或 CLI 為選項群組與資料庫執行個體建立關聯。

主控台

將您的選項群組關聯至新的或現有的資料庫執行個體：

- 對於新的資料庫執行個體，當您啟動執行個體時，將選項群組關聯至資料庫執行個體。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請修改執行個體並與新的選項群組建立關聯。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Note

如果您使用現有執行個體，則該執行個體必須已有與其相關聯的 Active Directory 網域和 AWS Identity and Access Management (IAM) 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和 IAM 角色。如需詳細資訊，請參閱 [使用 Active Directory 搭配 RDS for SQL Server](#)。

CLI

您可以將選項群組關聯至新的或現有的資料庫執行個體。

Note

如果您使用現有執行個體，則該執行個體必須已有與其相關聯的 Active Directory 網域和 IAM 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和 IAM 角色。如需詳細資訊，請參閱 [使用 Active Directory 搭配 RDS for SQL Server](#)。

建立使用選項群組的資料庫執行個體

- 指定建立選項群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier myssasinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --availability-zone us-east-1a \  
  --db-subnet-group mydb-subnet-group \  
  --vpc-subnet-id vpc-subnet-id \  
  --db-instance-profile db-instance-profile \  
  --db-parameters mydb-parameters \  
  --db-instance-username mydb-username \  
  --db-instance-password mydb-password \  
  --db-instance-availability-zone us-east-1a \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --db-subnet-group mydb-subnet-group \  
  --vpc-subnet-id vpc-subnet-id \  
  --db-instance-profile db-instance-profile \  
  --db-parameters mydb-parameters \  
  --db-instance-username mydb-username \  
  --db-instance-password mydb-password \  
  --db-instance-availability-zone us-east-1a
```

```
--engine-version 14.00.3223.3.v1 \  
--allocated-storage 100 \  
--manage-master-user-password \  
--master-username admin \  
--storage-type gp2 \  
--license-model li \  
--domain-iam-role-name my-directory-iam-role \  
--domain my-domain-id \  
--option-group-name ssas-se-2017
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier myssasinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 14.00.3223.3.v1 ^  
  --allocated-storage 100 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --domain-iam-role-name my-directory-iam-role ^  
  --domain my-domain-id ^  
  --option-group-name ssas-se-2017
```

修改資料庫執行個體以關聯選項群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier myssasinstance \  
  --option-group-name ssas-se-2017 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier myssasinstance ^
  --option-group-name ssas-se-2017 ^
  --apply-immediately
```

允許對 VPC 安全群組進行傳入存取

在與您的資料庫執行個體相關聯的 VPC 安全群組中，為指定的 SSAS 接聽程式連接埠建立傳入規則。如需設定安全群組的詳細資訊，請參閱[建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

啟用 Amazon S3 整合

若要將模型組態檔下載到主機以進行部署，請使用 Amazon S3 整合。如需詳細資訊，請參閱[整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

在 Amazon RDS 上部署 SSAS 專案

在 RDS 上，您無法使用 SQL Server Management Studio (SSMS) 直接部署 SSAS 專案。若要部署專案，請使用 RDS 預存程序。

Note

不支援使用 .xmla 檔案進行部署。

部署專案之前，請確定下列事項：

- Amazon S3 整合已開啟。如需詳細資訊，請參閱[整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。
- Processing Option 組態設定設為 Do Not Process。此設定表示部署後不會進行任何處理。
- 您同時擁有 *myssasproject.asdatabase* 和 *myssasproject.deploymentoptions* 檔案。它們會在您建置 SSAS 專案時自動產生。

在 RDS 上部署 SSAS 專案

1. 將 .asdatabase (SSAS 模型) 檔案從 S3 儲存貯體下載到資料庫執行個體，如下列範例所示。如需下載參數的詳細資訊，請參閱[將 Amazon S3 儲存貯體中的檔案下載至 SQL Server 資料庫執行個體](#)。

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
[@rds_file_path='D:\S3\myssasproject.asdatabase'],
[@overwrite_file=1];
```

- 將 .deploymentoptions 檔案從 S3 儲存貯體下載到您的資料庫執行個體。

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
[@rds_file_path='D:\S3\myssasproject.deploymentoptions'],
[@overwrite_file=1];
```

- 部署專案。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\myssasproject.asdatabase';
```

監視部署任務的狀態

若要追蹤部署 (或下載) 任務的狀態，請呼叫 `rds_fn_task_status` 函數。需要兩個參數。第一個參數不適用於 SSAS，所以應該一律為 NULL。第二個參數接受任務 ID。

若要查看所有任務的清單，請將第一個參數設為 NULL，將第二個參數設為 0，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

若要取得特定的任務，請將第一個參數設為 NULL，將第二個參數設為任務 ID，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

`rds_fn_task_status` 函數會傳回下列資訊。

輸出參數	描述
task_id	任務的 ID。
task_type	對於 SSAS，任務可以有如下列任務類型：

輸出參數	描述
	<ul style="list-style-type: none"> • SSAS_DEPLOY_PROJECT • SSAS_ADD_DB_ADMIN_MEMBER • SSAS_BACKUP_DB • SSAS_RESTORE_DB
database_name	不適用於 SSAS 任務。
% complete	任務的進度 (以百分比表示)。
duration (mins)	任務所花的時間 (以分鐘為單位)。
lifecycle	<p>任務的狀態。可能的狀態如下：</p> <ul style="list-style-type: none"> • CREATED – 呼叫其中一個 SSAS 預存程序之後，就會建立任務，而且狀態設為 CREATED。 • IN_PROGRESS – 任務開始之後，狀態會設為 IN_PROGRESS。可能需要經過 5 分鐘，狀態才會從 CREATED 變成 IN_PROGRESS。 • SUCCESS – 任務完成之後，狀態會設為 SUCCESS。 • ERROR – 如果任務失敗，狀態會設為 ERROR。關於錯誤的詳細資訊，請查看 <code>task_info</code> 欄。 • CANCEL_REQUESTED – 呼叫 <code>rdscancel_task</code> 之後，任務的狀態會設為 CANCEL_REQUESTED。 • CANCELLED – 成功取消任務之後，任務的狀態會設為 CANCELLED。

輸出參數	描述
task_info	任務的其他資訊。如果處理期間發生錯誤，此欄包含該錯誤的相關資訊。 如需詳細資訊，請參閱 疑難排解 SSAS 問題 。
last_updated	上次更新任務狀態的日期和時間。
created_at	建立任務的日期和時間。
S3_object_arn	不適用於 SSAS 任務。
overwrite_S3_backup_file	不適用於 SSAS 任務。
KMS_master_key_arn	不適用於 SSAS 任務。
filepath	不適用於 SSAS 任務。
overwrite_file	不適用於 SSAS 任務。
task_metadata	與 SSAS 任務相關聯的中繼資料。

在 Amazon RDS 上使用 SSAS

部署 SSAS 專案之後，您可以在 SSMS 上直接處理 OLAP 資料庫。

在 RDS 上使用 SSAS

1. 在 SSMS 中，使用 Active Directory 網域的使用者名稱和密碼連線到 SSAS。
2. 展開 Databases (資料庫)。新部署的 SSAS 資料庫隨即出現。
3. 找到連線字串，更新使用者名稱和密碼，為來源 SQL 資料庫提供存取權。這是處理 SSAS 物件的必要操作。
 - a. 若使用表格模式，請執行下列操作：
 1. 展開 Connections (連線) 索引標籤。

2. 開啟連線物件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Properties (屬性)。
 3. 檢查連線字串中的使用者名稱和密碼。
- b. 若使用多維度模式，請執行下列操作：
1. 展開 Data Sources (資料來源 索引標籤)
 2. 開啟資料來源的內容 (按一下滑鼠右鍵) 選單，然後選擇 Properties (屬性)。
 3. 檢查連線字串中的使用者名稱和密碼。
4. 開啟您所建立之 SSAS 資料庫的內容 (按一下滑鼠右鍵) 功能表，然後選擇 Process Database (處理資料庫)。

視輸入資料的大小而定，處理作業可能需要幾分鐘才能完成。

主題

- [為 SSAS 設定通過 Windows 驗證的使用者](#)
- [新增網域使用者做為資料庫管理員](#)
- [建立 SSAS Proxy](#)
- [使用 SQL Server Agent 排程 SSAS 資料庫處理作業](#)
- [從 Proxy 撤銷 SSAS 存取權](#)

為 SSAS 設定通過 Windows 驗證的使用者

主要管理員使用者 (又稱為主要使用者) 可以使用下列程式碼範例，設定通過 Windows 驗證的登入資料，並授予必要的程序許可。這樣做會為網域使用者授予許可，允許執行 SSAS 客戶任務、使用 S3 檔案傳輸程序、建立憑證以及使用 SQL Server Agent Proxy。如需詳細資訊，請參閱[登入資料 \(資料庫引擎\)](#) 和 Microsoft 文件中的[建立 SQL Server 代理程式 Proxy](#)。

您可以視需要授予部分或全部下列許可給 Windows 驗證的使用者。

Example

```
-- Create a server-level domain user login, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create domain user, if it doesn't already exist
```



```
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
GO

-- Grant necessary privileges to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO

USE [msdb]
GO
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] with grant option
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO
```

新增網域使用者做為資料庫管理員

您可以使用下列方式，將網域使用者新增為 SSAS 資料庫管理員：

- 資料庫管理員可以使用 SSMS 來建立具有 admin 權限的角色，然後將使用者新增至該角色。

- 您可以使用下列預存程序。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_ADD_DB_ADMIN_MEMBER',
@database_name='myssasdb',
@ssas_role_name='exampleRole',
@ssas_role_member='domain_name\domain_user_name';
```

下列是必要參數：

- @task_type – MSBI 任務的類型，在此案例中為 SSAS_ADD_DB_ADMIN_MEMBER。
- @database_name– 您要授與系統管理員權限的 SSAS 資料庫名稱。
- @ssas_role_name – SSAS 資料庫管理員角色名稱。如果角色不存在，則會建立該角色。
- @ssas_role_member – 您要新增至系統管理員角色的 SSAS 資料庫使用者。

建立 SSAS Proxy

需先建立 SSAS 憑證和 SSAS Proxy，才能使用 SQL Server Agent 排程 SSAS 資料庫處理作業。以 Windows 驗證的使用者身分執行這些程序。

建立 SSAS 憑證

- 建立代理的登入資料。若要執行這項操作，您可以使用 SSMS 或下列 SQL 陳述式。

```
USE [master]
GO
CREATE CREDENTIAL [SSAS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
N'mysecret'
GO
```

Note

IDENTITY 必須是網域驗證的登入。以網域驗證登入的密碼取代 *mysecret*。

建立 SSAS Proxy

1. 使用下面的 SQL 陳述式來建立代理。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
    @proxy_name=N'SSAS_Proxy',@credential_name=N'SSAS_Credential',@description=N''
GO
```

2. 使用下列 SQL 陳述式可將代理伺服器的存取權授予其他使用者。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
    @proxy_name=N'SSAS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. 使用下列 SQL 陳述式來提供 SSAS 子系統 Proxy 的存取權。

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
    @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

在代理上檢視代理和授權

1. 使用下列 SQL 陳述式來檢視代理的承授者。

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. 使用下列 SQL 陳述式來檢視子系統授予。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

使用 SQL Server Agent 排程 SSAS 資料庫處理作業

建立憑證和 Proxy，並授予 SSAS Proxy 的存取權後，就可以建立 SQL Server Agent 任務來排程 SSAS 資料庫處理作業。

排程 SSAS 資料庫處理作業

- 使用 SSMS 或 T-SQL 來建立 SQL Server Agent 任務。下列範例使用 T-SQL。您可以透過 SSMS 或 T-SQL 進一步設定其任務排程。
 - `@command` 參數概述要由 SQL Server Agent 任務執行的 XML for Analysis (XMLA) 命令。此範例設定了 SSAS 多維度資料庫處理作業。
 - `@server` 參數概述 SQL Server Agent 任務的目標 SSAS 伺服器名稱。

若要在 SQL Server Agent 任務所在的相同 RDS 資料庫執行個體中呼叫 SSAS 服務，請使用 `localhost:2383`。

若要從 RDS 資料庫執行個體外部呼叫 SSAS 服務，請使用 RDS 端點。如果 RDS 資料庫執行個體是由同一個網域加入，也可以使用 Kerberos Active Directory (AD) 端點 (*your-DB-instance-name.your-AD-domain-name*)。若使用外部資料庫執行個體，請務必正確設定與 RDS 資料庫執行個體相關聯的 VPC 安全群組，以確保安全連線。

您可以進一步編輯查詢以支援各種 XMLA 作業。SQL Server Agent 任務建立後，可直接修改 T-SQL 查詢或使用 SSMS UI 進行編輯。

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'SSAS_Job',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=0,
    @notify_level_netsend=0,
    @notify_level_page=0,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver
    @job_name=N'SSAS_Job',
    @server_name = N'(local)'
```

```

GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'SSAS_Job',
    @step_name=N'Process_SSAS_Object',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_success_step_id=0,
    @on_fail_action=2,
    @on_fail_step_id=0,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
    @command=N'<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/
engine">
    <Parallel>
        <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/
engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/
engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/
analysiservices/2008/engine/100/100" xmlns:ddl200="http://schemas.microsoft.com/
analysiservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/
analysiservices/2010/engine/200/200" xmlns:ddl300="http://schemas.microsoft.com/
analysiservices/2011/engine/300"
xmlns:ddl300_300="http://schemas.microsoft.com/
analysiservices/2011/engine/300/300" xmlns:ddl400="http://schemas.microsoft.com/
analysiservices/2012/engine/400"
xmlns:ddl400_400="http://schemas.microsoft.com/
analysiservices/2012/engine/400/400" xmlns:ddl500="http://schemas.microsoft.com/
analysiservices/2013/engine/500"
xmlns:ddl500_500="http://schemas.microsoft.com/
analysiservices/2013/engine/500/500">
        <Object>
            <DatabaseID>Your_SSAS_Database_ID</DatabaseID>
        </Object>
        <Type>ProcessFull</Type>
        <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
    </Parallel>
</Batch>',
    @server=N'localhost:2383',
    @database_name=N'master',

```

```
@flags=0,  
@proxy_name=N'SSAS_Proxy'  
GO
```

從 Proxy 撤銷 SSAS 存取權

您可以撤銷 SSAS 子系統的存取權，並使用下列預存程序刪除 SSAS Proxy。

撤銷存取權並刪除代理伺服器

1. 撤銷子系統存取權。

```
USE [msdb]  
GO  
EXEC msdb.dbo.rds_sqlagent_proxy  
@task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'  
GO
```

2. 撤銷代理上的授權。

```
USE [msdb]  
GO  
EXEC msdb.dbo.sp_revoke_login_from_proxy  
@proxy_name=N'SSAS_Proxy',@name=N'mydomain\user_name'  
GO
```

3. 刪除代理。

```
USE [msdb]  
GO  
EXEC dbo.sp_delete_proxy @proxy_name = N'SSAS_Proxy'  
GO
```

備份 SSAS 資料庫

您只能在資料庫執行個體的 D:\S3 資料夾中建立 SSAS 資料庫備份檔案。若要將備份檔案移至 S3 儲存貯體，請使用 Amazon S3。

您可以依照以下方式備份 SSAS 資料庫：

- 具有特定資料庫 admin 角色的網域使用者，可以使用 SSMS 將資料庫備份至 D:\S3 資料夾。

如需更多詳細資訊，請參閱 [新增網域使用者做為資料庫管理員](#)。

- 您可以使用下列預存程序。預存程序不支援加密。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_BACKUP_DB',
@database_name='myssasdb',
@file_path='D:\S3\ssas_db_backup.abf',
[@ssas_apply_compression=1],
[@ssas_overwrite_file=1];
```

下列是必要參數：

- @task_type – MSBI 任務的類型，在此案例中為 SSAS_BACKUP_DB。
- @database_name – 您要備份之 SSAS 資料庫的名稱。
- @file_path – SSAS 備份檔案的路徑。 .abf 副檔名為必要項目。

下列是選用參數：

- @ssas_apply_compression – 是否要套用 SSAS 備份壓縮。有效值為 1 (是) 和 0 (否)。
- @ssas_overwrite_file – 是否覆寫 SSAS 備份檔案。有效值為 1 (是) 和 0 (否)。

還原 SSAS 資料庫

使用下列預存程序從備份還原 SSAS 資料庫。

如果存在具有相同名稱的現有 SSAS 資料庫，則無法還原資料庫。還原的預存程序不支援加密的備份檔案。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

下列是必要參數：

- @task_type – MSBI 任務的類型，在此案例中為 SSAS_RESTORE_DB。
- @database_name – 您要還原的新 SSAS 資料庫名稱。
- @file_path – SSAS 備份檔案的路徑。

將資料庫執行個體還原至指定的時間

Point-in-time 復原 (PITR) 不適用於 SSAS 資料庫。如果您執行 PITR，則已還原執行個體只能使用所要求時間之前最後一個快照中的 SSAS 資料。

在還原的資料庫執行 up-to-date 行個體上安裝 SSAS 資料庫

1. 將 SSAS 資料庫備份到來源執行個體上的 D:\S3 資料夾。
2. 將備份檔案傳輸到 S3 儲存貯體。
3. 將備份檔案從 S3 儲存貯體傳輸到已還原執行個體上的 D:\S3 資料夾。
4. 執行預存程序，將 SSAS 資料庫還原到還原的執行個體。

您也可以重新處理 SSAS 專案，以還原資料庫。

變更 SSAS 模式

您可以變更 SSAS 執行的模式 (表格或多維度)。若要變更模式，請使用 AWS Management Console 或 AWS CLI 來修改 SSAS 選項中的選項設定。

Important

一次只能使用一種 SSAS 模式。變更模式之前請務必刪除所有 SSAS 資料庫，否則會收到錯誤訊息。

主控台

以下 Amazon RDS 主控台程序將 SSAS 模式變更為「表格」，並將 MAX_MEMORY 參數設為 70%。

修改 SSAS 選項

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您想修改具有 SSAS 選項的選項群組 (上述範例中的 `ssas-se-2017`)。
4. 選擇 Modify option (修改選項)。
5. 變更選項設定：
 - a. 為 Max memory (最大記憶體) 輸入 **70**。

- b. 為 Mode (模式) 選擇表格。
6. 選擇 Modify option (修改選項)。

AWS CLI

下列 AWS CLI 範例會將 SSAS 模式變更為「表格式」，並將 MAX_MEMORY 參數設定為 70%。

為了讓 CLI 命令正常運作，即使沒有要修改，也務必包含所有必要參數。

修改 SSAS 選項

- 請使用下列其中一個命令。

Example

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name ssas-se-2017 \  
  --options  
  "OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,  
{Name=MODE,Value=Tabular}]" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name ssas-se-2017 ^  
  --options  
  OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,V  
{Name=MODE,Value=Tabular}] ^  
  --apply-immediately
```

關閉 SSAS

若要關閉 SSAS，請從其選項群組中移除 SSAS 選項。

Important

移除 SSAS 選項之前，請先刪除您的 SSAS 資料庫。

我們強烈建議您先備份 SSAS 資料庫，然後再刪除它們並移除 SSAS 選項。

主控台

從選項群組中移除 SSAS 選項

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您想移除具有 SSAS 選項的選項群組 (上述範例中的 `ssas-se-2017`)。
4. 選擇 Delete option (刪除選項)。
5. 在 Deletion options (刪除選項) 下，選擇 SSAS 作為 Options to delete (要刪除的選項)。
6. 在 Apply immediately (立即套用) 下，選擇 Yes (是) 立即刪除選項，或選擇 No (否) 在下一個維護時段將其刪除。
7. 選擇刪除。

AWS CLI

從選項群組中移除 SSAS 選項

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name ssas-se-2017 \  
  --options SSAS \  
  --apply-immediately
```

在 Windows 中：

```
aws rds remove-option-from-option-group ^  
  --option-group-name ssas-se-2017 ^  
  --options SSAS ^  
  --apply-immediately
```

疑難排解 SSAS 問題

使用 SSAS 時可能會遇到下列問題。

問題	Type	故障診斷建議
無法設定 SSAS 選項。要求的 SSA 模式為 <i>new_mode</i> ，但目前的資料庫執行個體具有##個 <i>current_mode</i> 資料庫。先刪除現有資料庫，再切換到 <i>new_mode</i> 模式。若要重新取得 <i>current_mode</i> 模式的存取權，可以更新目前的資料庫選項群組，或者連接新的選項群組 (SSAS 選項的 MODE 選項設定值使用 %s)。	RDS 事件	如果仍有 SSAS 資料庫使用目前模式，就無法變更 SSAS 模式。刪除 SSAS 資料庫，然後再試一次。
無法移除 SSAS 選項，因為有##個現有的##資料庫。在刪除所有 SSAS 資料庫之前，無法移除 SSAS 選項。再次新增 SSAS 選項，刪除所有 SSAS 資料庫然後重試。	RDS 事件	如果您仍有 SSAS 資料庫，就無法關閉 SSAS。刪除 SSAS 資料庫，然後再試一次。
SSAS 選項未啟用，或正在進行啟用作業。請稍後再試。	RDS 預存程序	當選項處於關閉狀態或進行開啟作業時，無法執行 SSAS 預存程序。
SSAS 選項設定不正確。確認選項群組成員資格狀態為 "in-sync"，並檢閱 RDS 事件日誌中的相關 SSAS 組態錯誤訊息。進行這些調查之後，請再試一次。如果錯誤 AWS 持續發生，請聯絡 Support 部門。	RDS 預存程序	<p>若選項群組成員資格不處於 in-sync 狀態，就無法執行 SSAS 預存程序。這會導致 SSAS 選項處於錯誤的組態狀態。</p> <p>如果您的選項群組成員資格狀態由於 SSAS 選項修改而變更為 failed，可能有兩個原因：</p> <ol style="list-style-type: none"> 1. 移除了 SSAS 選項而沒有刪除 SSAS 資料庫。 2. SSAS 模式從表格更新為多維度，或從多維度更新為表格，而沒有刪除現有 SSAS 資料庫。

問題	Type	故障診斷建議
		<p>重新設定 SSAS 選項，因為 RDS 一次只允許使用一種 SSAS 模式，且不支援在 SSAS 資料庫存在的情況下刪除 SSAS 選項。</p> <p>檢查 RDS 事件日誌中的 SSAS 執行個體組態錯誤，並視情況解決些問題。</p>
<p>部署失敗。變更只能部署在執行 <i>deployment_file_mode</i> 模式的伺服器上。目前的伺服器模式為 <i>current_mode</i>。</p>	RDS 預存程序	<p>無法將表格模式的資料庫部署到多維度模式的伺服器，也不能將多維度模式資料庫部署到表格模式伺服器。</p> <p>請確認您使用的是模式正確的檔案，然後驗證 MODE 選項設定設為適當的值。</p>
<p>還原失敗。備份檔案只能在執行 <i>restore_file_mode</i> 模式的伺服器上還原。目前的伺服器模式為 <i>current_mode</i>。</p>	RDS 預存程序	<p>無法將表格模式的資料庫還原到多維度模式的伺服器，也不能將多維度模式資料庫還原到表格模式伺服器。</p> <p>請確認您使用的是模式正確的檔案，然後驗證 MODE 選項設定設為適當的值。</p>
<p>還原失敗。備份檔案和 RDS 資料庫執行個體版本不相容。</p>	RDS 預存程序	<p>無法還原與 SQL Server 執行個體版本不相容的 SSAS 資料庫版本。</p> <p>如需詳細資訊，請參閱 Microsoft 說明文件中的 表格模式的相容性層級 和 多維度資料庫的相容性層級。</p>
<p>還原失敗。還原作業中指定的備份檔案已損毀或非 SSAS 備份檔案。確認 @rds_file_path 的格式正確。</p>	RDS 預存程序	<p>無法還原具有損毀檔案的 SSAS 資料庫。</p> <p>請確認檔案未損毀或損壞。</p> <p>若 @rds_file_path 格式不正確 (例如含有雙反斜槓，如 D:\S3\\incorrect_format.abf 中所述)，也可能出現此錯誤。</p>

問題	Type	故障診斷建議
還原失敗。還原的資料庫名稱不能包含任何保留字詞或以下無效字元：.,;`:/* ?\" & % \$! + = () [] { } < > , 長度也不能超過 100 個字元。	RDS 預存程序	還原的資料庫名稱不能包含任何保留字詞，長度也不能超過 100 個字元。 如需 SSAS 物件命名慣例，請參閱 Microsoft 說明文件中的 物件命名規則 。
提供的角色名稱無效。角色名稱不能包含任何保留的字串。	RDS 預存程序	角色名稱不能包含任何保留的字串。 如需 SSAS 物件命名慣例，請參閱 Microsoft 說明文件中的 物件命名規則 。
提供的角色名稱無效。角色名稱不能包含以下任何保留字元：.,;`:/* ?\" & % \$! + = () [] { } < >	RDS 預存程序	角色名稱不能包含任何保留字元。 如需 SSAS 物件命名慣例，請參閱 Microsoft 說明文件中的 物件命名規則 。

支援 Amazon RDS for SQL Server 中的 SQL Server Integration Services

Microsoft SQL Server Integration Services (SSIS) 是您可以用來執行廣泛資料遷移任務的元件。SSIS 是資料整合和工作流程應用程式的平台。它具有用於資料擷取、轉換和加載 (ETL) 的資料倉儲工具。您也可以使用這個工具來自動維護 SQL Server 資料庫和更新多維度 Cube 資料。

SSIS 專案會組織成儲存為 XML .dtsx 檔案的套件。套件可以包含控制流量和資料流程。您可以使用資料流程來表示 ETL 操作。部署後，套件會儲存在 SSISDB 資料庫中的 SQL Server 中。SSISDB 是在完整復原模式下的線上交易處理 (OLTP) 資料庫。

Amazon RDS for SQL Server 會支援直接在 RDS 資料庫執行個體上執行 SSIS。您可以在現有或新的資料庫執行個體上啟用 SSIS。SSIS 會安裝在與資料庫引擎相同的資料庫執行個體上。

下列版本的 RDS 支援 SSIS for SQL Server Standard 和 Enterprise Edition：

- SQL 伺服器，所有版本
- SQL Server 2019，15.00.4043.16.v1 版和更新版本
- SQL Server 2017，14.00.3223.3.v1 版和更新版本
- SQL Server 2016，13.00.5426.0.v1 版和更新版本

內容

- [限制與建議](#)
- [啟用 SSIS](#)
 - [建立 SSIS 的選項群組](#)
 - [將 SSIS 選項新增至選項群組](#)
 - [建立 SSIS 的參數群組](#)
 - [修改 SSIS 的參數](#)
 - [將選項群組和參數群組與資料庫執行個體建立關聯](#)
 - [啟用 S3 整合](#)
- [SSISDB 上的管理許可](#)
 - [為 SSIS 設定 Windows 驗證的使用者](#)
- [部署 SSIS 專案](#)
- [監視部署任務的狀態](#)
- [使用 SSIS](#)

- [設定 SSIS 專案的資料庫連線管理員](#)
- [建立 SSIS 代理](#)
- [使用 SQL Server 代理程式排程 SSIS 套件](#)
- [從代理撤銷 SSIS 存取權](#)
- [停用 SSIS](#)
- [卸除 SSISDB 資料庫](#)

限制與建議

在 RDS for SQL Server 上執行 SSIS 時適用下列限制和建議：

- 資料庫執行個體必須具有相關的參數群組，且將 `clr enabled` 參數設定為 1。如需詳細資訊，請參閱 [修改 SSIS 的參數](#)。

Note

如果您在 SQL Server 2017 或 2019 上啟用 `clr enabled` 參數，就無法在資料庫執行個體上使用通用語言執行階段 (CLR)。如需詳細資訊，請參閱 [不支援的功能和具備有限支援的功能](#)。

- 支援下列控制流量任務：
 - 分析服務執行 DDL 任務
 - 分析服務處理任務
 - 大量插入任務
 - 檢查資料庫完整性任務
 - 資料流程任務
 - 資料挖掘查詢任務
 - 資料效能分析任務
 - 執行套件任務
 - 執行 SQL Server Agent 任務工作
 - 執行 SQL 任務
 - 執行 T-SQL 陳述式任務
- [通知運算子任務](#)

- 重新建立索引任務
- 重新組織索引任務
- 壓縮資料庫任務
- 傳輸資料庫任務
- 傳送任務工作
- 傳輸登入任務
- 傳輸 SQL Server 物件任務
- 更新統計資料任務
- 僅支援專案部署。
- 支援使用 SQL Server Agent 執行 SSIS 套件。
- SSIS 日誌記錄只能插入使用者建立的資料庫中。
- 僅使用 D:\S3 資料夾來處理檔案。放在任何其他目錄中的檔案都會被刪除。請注意其他一些檔案位置的詳細資料：
 - 將 SSIS 專案輸入和輸出檔案放在 D:\S3 資料夾。
 - 對於「資料流程任務」，變更 BLOBTempStoragePath 資料夾內 BufferTempStoragePath 和 D:\S3 檔案的位置。檔案路徑必須以 D:\S3\ 為開頭。
 - 確定用於檔案連線的所有參數、變數和運算式都指向該 D:\S3 資料夾。
 - 在異地同步備份執行個體上，由 SSIS 在 D:\S3 資料夾中建立的檔案會在容錯移轉後刪除。如需更多詳細資訊，請參閱 [S3 整合的多可用區域限制](#)。
 - 由 SSIS 在 D:\S3 資料夾中建立的檔案會上傳到您的 Amazon S3 儲存貯體，使其變得耐用。
- 不支援「匯入欄」和「匯出欄」轉換，以及「資料流程任務」上的「指令碼」元件。
- 您無法在執行中 SSIS 套件上啟用傾印，也無法在 SSIS 套件上新增資料點選。
- 不支援 SSIS 向外擴展功能。
- 您無法直接部署專案。我們提供 RDS 儲存過程來做到這一點。如需更多詳細資訊，請參閱 [部署 SSIS 專案](#)。
- 使用 DoNotSavePasswords 保護模式來建立要在 RDS 上部署的 SSIS 專案 (.ispac) 檔案。
- SSIS 不支援有讀取複本的 Always On 執行個體。
- 您無法備份與 SSIS 選項相關聯的 SSISDB 資料庫。
- 不支援從 SSIS 的其他執行個體匯入和還原 SSISDB 資料庫。

- 您可以連線至其他 SQL Server 資料庫執行個體或 Oracle 資料來源。RDS for SQL Server 上的 SSIS 不支援連線到其他資料庫引擎，例如 MySQL 或 PostgreSQL。如需連線至 Oracle 資料來源的詳細資訊，請參閱 [連結的伺服器搭配 Oracle OLEDB](#)。

啟用 SSIS

您可以透過將 SSIS 選項新增至資料庫執行個體來啟用 SSIS。請使用下列程序：

1. 建立新的選項群組或選擇現有的選項群組。
2. 將 SSIS 選項新增至選項群組。
3. 建立新參數群組，或選擇現有參數群組。
4. 修改參數群組，將 `clr enabled` 參數設定為 1。
5. 將選項群組和參數群組與資料庫執行個體建立關聯。
6. 啟用 Amazon S3 整合。

Note

如果資料庫執行個體上已存在名為 SSISDB 或預留 SSIS 登入的資料庫，則無法在執行個體上啟用 SSIS。

建立 SSIS 的選項群組

若要使用 SSIS，請建立選項群組，或修改對應至 SQL Server 版本和您計劃使用的資料庫執行個體版本的選項群組。若要執行此操作，請使用 AWS Management Console 或 AWS CLI。

主控台

下列程序會建立 SQL Server Standard Edition 2016 的選項群組。

建立選項群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 視窗中，執行下列動作：

- a. 在 Name (名稱) 中，輸入您 AWS 帳戶中的唯一選項群組名稱，例如 **ssis-se-2016**。名稱僅可包含字母、數字與連字號。
 - b. 對於 Description (描述)，請輸入選項群組的簡短描述，例如 **SSIS option group for SQL Server SE 2016**。用於顯示用途的說明。
 - c. 對於 Engine (引擎)，請選擇 **sqlserver-se**。
 - d. 對於 Major engine version (主要引擎版本)，請選擇 **13.00**。
5. 選擇 Create (建立)。

CLI

下列程序會建立 SQL Server Standard Edition 2016 的選項群組。

建立選項群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name ssis-se-2016 \  
  --engine-name sqlserver-se \  
  --major-engine-version 13.00 \  
  --option-group-description "SSIS option group for SQL Server SE 2016"
```

在Windows中：

```
aws rds create-option-group ^  
  --option-group-name ssis-se-2016 ^  
  --engine-name sqlserver-se ^  
  --major-engine-version 13.00 ^  
  --option-group-description "SSIS option group for SQL Server SE 2016"
```

將 SSIS 選項新增至選項群組

接下來，使用AWS Management Console或 AWS CLI 將 SSIS 選項新增至選項群組。

主控台

新增 SSIS 選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您剛剛建立的選項群組，在這個例子中為 `ssis-se-2016`。
4. 選擇 Add option (新增選項)。
5. 在 Option details (選項詳細資料) 下，選擇 SSIS 做為 Option name (選項名稱)。
6. 在 Scheduling (排程) 下，選擇要立即新增選項或是在下一個維護時段新增選項。
7. 選擇 Add option (新增選項)。

CLI

新增 SSIS 選項

- 將 SSIS 選項新增至選項群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name ssis-se-2016 \  
  --options OptionName=SSIS \  
  --apply-immediately
```

在Windows中：

```
aws rds add-option-to-option-group ^  
  --option-group-name ssis-se-2016 ^  
  --options OptionName=SSIS ^  
  --apply-immediately
```

建立 SSIS 的參數群組

為對應到 SQL Server 版本和您計畫用於 SSIS 的資料庫執行個體版本的 `clr enabled` 參數來建立或修改參數群組。

主控台

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在 Create parameter group (建立參數群組) 窗格中執行下列動作：
 - a. 對於 Parameter group family (參數群組家族)，請選擇 `sqlserver-se-13.0`。
 - b. 對於 Group name (群組名稱)，輸入參數群組的識別碼，例如 `ssis-sqlserver-se-13`。
 - c. 對於 Description (說明)，輸入 `clr enabled parameter group`。
5. 選擇 Create (建立)。

CLI

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name ssis-sqlserver-se-13 \  
  --db-parameter-group-family "sqlserver-se-13.0" \  
  --description "clr enabled parameter group"
```

在Windows中：

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name ssis-sqlserver-se-13 ^
  --db-parameter-group-family "sqlserver-se-13.0" ^
  --description "clr enabled parameter group"
```

修改 SSIS 的參數

修改參數群組中對應至 SQL Server 版本和資料庫執行個體版本的 `clr enabled` 參數。若為 SSIS，請將 `clr enabled` 參數設定為 1。

主控台

下列範例會修改您為 SQL Server Standard Edition 2016 建立的參數群組。

修改參數群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇參數群組，例如 `ssis-sqlserver-se-13`。
4. 在 Parameters (參數) 下，篩選 `clr` 的參數清單。
5. 選擇 `clr enabled` (clr 已啟用)。
6. 選擇 Edit parameters (編輯參數)。
7. 從 Values (值) 中選擇 1。
8. 選擇儲存變更。

CLI

下列範例會修改您為 SQL Server Standard Edition 2016 建立的參數群組。

修改參數群組

- 請執行下列其中一個命令：

Example

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name ssis-sqlserver-se-13 \  
  --parameters "ParameterName='clr  
enabled',ParameterValue=1,ApplyMethod=immediate"
```

在Windows中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name ssis-sqlserver-se-13 ^  
  --parameters "ParameterName='clr  
enabled',ParameterValue=1,ApplyMethod=immediate"
```

將選項群組和參數群組與資料庫執行個體建立關聯

若要將 SSIS 選項群組和參數群組與資料庫執行個體建立關聯，請使用 AWS Management Console 或 AWS CLI

Note

如果您使用現有執行個體，則該執行個體必須已有與其相關聯的 Active Directory 網域和 AWS Identity and Access Management (IAM) 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和 IAM 角色。如需詳細資訊，請參閱[使用 Active Directory 搭配 RDS for SQL Server](#)。

主控台

若要完成啟用 SSIS，請將 SSIS 選項群組和參數群組與新的或現有的資料庫執行個體建立關聯：

- 對於新的資料庫執行個體，請在啟動執行個體時將它們相關聯。如需更多詳細資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請透過修改執行個體來建立關聯。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

CLI

您可以將 SSIS 選項群組和參數群組與新的或現有的資料庫執行個體產生關聯。

使用 SSIS 選項群組和參數群組建立執行個體

- 指定建立選項群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier myssisinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 13.00.5426.0.v1 \  
  --allocated-storage 100 \  
  --manage-master-user-password \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --domain-iam-role-name my-directory-iam-role \  
  --domain my-domain-id \  
  --option-group-name ssis-se-2016 \  
  --db-parameter-group-name ssis-sqlserver-se-13
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier myssisinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 13.00.5426.0.v1 ^  
  --allocated-storage 100 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --domain-iam-role-name my-directory-iam-role ^  
  --domain my-domain-id ^  
  --option-group-name ssis-se-2016 ^  
  --db-parameter-group-name ssis-sqlserver-se-13
```

修改執行個體並建立 SSIS 選項群組和參數群組的關聯

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier myssisinstance \  
  --option-group-name ssis-se-2016 \  
  --db-parameter-group-name ssis-sqlserver-se-13 \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier myssisinstance ^  
  --option-group-name ssis-se-2016 ^  
  --db-parameter-group-name ssis-sqlserver-se-13 ^  
  --apply-immediately
```

啟用 S3 整合

若要將 SSIS 專案 (.ispac) 檔案下載到主機以進行部署，請使用 S3 檔案整合。如需詳細資訊，請參閱[整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

SSISDB 上的管理許可

使用 SSIS 選項建立或修改執行個體時，結果會是 SSISDB 資料庫，其中包含 `ssis_admin` 和 `ssis_logreader` 角色授予主要使用者。主要使用者在 SSISDB 中具有下列權限：

- alter on `ssis_admin` role
- alter on `ssis_logreader` role
- 變更任何使用者

因為主要使用者是 SQL 驗證的使用者，所以您無法使用主要使用者來執行 SSIS 套件。主要使用者可以使用這些權限來建立新的 SSISDB 使用者，並將它們新增至 `ssis_admin` 和 `ssis_logreader` 角色。這樣做對於授予網域使用者使用 SSIS 的存取權非常有用。

為 SSIS 設定 Windows 驗證的使用者

主要使用者可以使用下列程式碼範例，在 SSISDB 中設定 Windows 驗證的登入，並授予必要的程序許可。這樣做會授予網域使用者部署和執行 SSIS 套件、使用 S3 檔案傳輸程序、建立登入資料，以及搭配 SQL Server Agent 代理使用的許可。如需詳細資訊，請參閱[登入資料 \(資料庫引擎\)](#) 和 Microsoft 文件中的[建立 SQL Server 代理程式 Proxy](#)。

Note

您可以視需要授予部分或全部下列許可給 Windows 驗證的使用者。

Example

```
-- Create a server-level SQL login for the domain user, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create a database-level account for the domain user, if it doesn't already exist

USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Add SSIS role membership to the domain user
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

-- Add MSDB role membership to the domain user
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Grant MSDB stored procedure privileges to the domain user
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
```

```
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION

-- Add the SQLAgentUserRole privilege to the domain user
USE [msdb]
GO
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

-- Grant the ALTER ANY CREDENTIAL privilege to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

部署 SSIS 專案

在 RDS 上，您無法使用 SQL Server Management Studio (SSMS) 或 SSIS 程序直接部署 SSIS 專案。若要從 Amazon S3 中下載專案檔案，然後進行部署，請使用 RDS 儲存的程序。

若要執行預存程序，請以您授予預存程序執行許可的任何其他使用者身分登入。如需更多詳細資訊，請參閱 [為 SSIS 設定 Windows 驗證的使用者](#)。

部署 SSIS 專案

1. 下載專案 (.ispac) 檔案。

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
[@rds_file_path='D:\S3\ssisproject.ispac'],
[@overwrite_file=1];
```

2. 提交部署任務，確定下列項目：

- 該資料夾存在於 SSIS 目錄中。
- 專案名稱與您在開發 SSIS 專案時使用的專案名稱相符。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

監視部署任務的狀態

若要追蹤部署 (或下載) 任務的狀態，請呼叫 `rds_fn_task_status` 函數。需要兩個參數。第一個參數不適用於 SSIS，所以應該一律為 NULL。第二個參數接受任務 ID。

若要查看所有任務的清單，請將第一個參數設為 NULL，將第二個參數設為 0，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

若要取得特定的任務，請將第一個參數設為 NULL，將第二個參數設為任務 ID，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

`rds_fn_task_status` 函數會傳回下列資訊。

輸出參數	描述
<code>task_id</code>	任務的 ID。
<code>task_type</code>	SSIS_DEPLOY_PROJECT
<code>database_name</code>	不適用於 SSIS 任務。

輸出參數	描述
% complete	任務的進度 (以百分比表示)。
duration (mins)	任務所花的時間 (以分鐘為單位)。
lifecycle	<p>任務的狀態。可能的狀態如下：</p> <ul style="list-style-type: none"> • CREATED – 呼叫 <code>msdb.dbo.rds_msbi_task</code> 儲存的程序之後，系統就會建立任務，且狀態會設定為 CREATED。 • IN_PROGRESS – 任務開始之後，狀態會設為 IN_PROGRESS。可能需要經過 5 分鐘，狀態才會從 CREATED 變成 IN_PROGRESS。 • SUCCESS – 任務完成之後，狀態會設為 SUCCESS。 • ERROR – 如果任務失敗，狀態會設為 ERROR。關於錯誤的詳細資訊，請查看 <code>task_info</code> 欄。 • CANCEL_REQUESTED – 呼叫 <code>rds_cancel_task</code> 之後，任務的狀態會設為 CANCEL_REQUESTED。 • CANCELLED – 成功取消任務之後，任務的狀態會設為 CANCELLED。
task_info	任務的其他資訊。如果處理期間發生錯誤，此欄包含該錯誤的相關資訊。
last_updated	上次更新任務狀態的日期和時間。
created_at	建立任務的日期和時間。

輸出參數	描述
S3_object_arn	不適用於 SSIS 任務。
overwrite_S3_backup_file	不適用於 SSIS 任務。
KMS_master_key_arn	不適用於 SSIS 任務。
filepath	不適用於 SSIS 任務。
overwrite_file	不適用於 SSIS 任務。
task_metadata	與 SSIS 任務相關聯的中繼資料。

使用 SSIS

將 SSIS 專案部署到 SSIS 目錄之後，您可以直接從 SSMS 執行套件，或使用 SQL Server 代理程式來加以排程。您必須使用 Windows 驗證的登入來執行 SSIS 套件。如需更多詳細資訊，請參閱 [為 SSIS 設定 Windows 驗證的使用者](#)。

主題

- [設定 SSIS 專案的資料庫連線管理員](#)
- [建立 SSIS 代理](#)
- [使用 SQL Server 代理程式排程 SSIS 套件](#)
- [從代理撤銷 SSIS 存取權](#)

設定 SSIS 專案的資料庫連線管理員

當您使用連線管理員時，您可以使用下列類型的驗證：

- 對於使用 AWS 受管 Active Directory 的本機資料庫連線，您可以使用 SQL 驗證或 Windows 驗證。對於 Windows 驗證，請用 *DB_instance_name.fully_qualified_domain_name* 作連線字串的伺服器名稱。

例如 `myssisinstance.corp-ad.example.com`，其中 `myssisinstance` 是資料庫執行個體名稱，而且 `corp-ad.example.com` 是完整網域名稱。

- 對於遠端連線，請一律使用 SQL 驗證。
- 對於使用自我管理 Active Directory 的本機資料庫連線，您可以使用 SQL 驗證或 Windows 驗證。對於 Windows 驗證，請使用 `.` 或 `LocalHost` 作為連線字串的伺服器名稱。

建立 SSIS 代理

若要能夠使用 SQL Server 代理程式排程 SSIS 套件，請建立 SSIS 登入資料和 SSIS 代理。以 Windows 驗證的使用者身分執行這些程序。

建立 SSIS 登入資料

- 建立代理的登入資料。若要執行這項操作，您可以使用 SSMS 或下列 SQL 陳述式。

```
USE [master]
GO
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
N'mysecret'
GO
```

Note

IDENTITY 必須是網域驗證的登入。以網域驗證登入的密碼取代 *mysecret*。
每當 SSISDB 主要主機變更時，更改 SSIS 代理登入資料，以允許新的主機來存取它們。

建立 SSIS 代理

1. 使用下面的 SQL 陳述式來建立代理。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
    @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. 使用下列 SQL 陳述式可將代理伺服器的存取權授予其他使用者。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
    @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. 使用下列 SQL 陳述式來提供 SSIS 子系統存取代理。

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
    @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

在代理上檢視代理和授權

1. 使用下列 SQL 陳述式來檢視代理的承授者。

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. 使用下列 SQL 陳述式來檢視子系統授予。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

使用 SQL Server 代理程式排程 SSIS 套件

建立登入資料和代理，並授予 SSIS 存取權至代理之後，您可以建立 SQL Server Agent 任務來排程 SSIS 套件。

排程 SSIS 套件

- 您可以使用 SSMS 或 T-SQL 來建立 SQL Server Agent 任務。下列範例使用 T-SQL。

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=2,
    @notify_level_page=2,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep
    @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_fail_action=2,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0,
    @subsystem=N'SSIS',
    @command=N'/ISSERVER "\"\SSISDB\MySSISFolder\MySSISProject\MySSISPackage.dtsx\""" /
SERVER "\"my-rds-ssis-instance.corp-ad.company.com/\""
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\"";1 /Par
    "\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING
    E',
    @database_name=N'master',
    @flags=0,
    @proxy_name=N'SSIS_Proxy'
GO
```

從代理撤銷 SSIS 存取權

您可以撤銷 SSIS 子系統的存取權，並使用下列儲存的程序刪除 SSIS 代理。

撤銷存取權並刪除代理伺服器

1. 撤銷子系統存取權。


```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
    @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. 撤銷代理上的授權。

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
    @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
GO
```

3. 刪除代理。

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

停用 SSIS

若要停用 SSIS，請將 SSIS 選項從其選項群組中移除。

Important

移除選項並不會刪除 SSISDB 資料庫，因此您可以安全地移除選項，而不會遺失 SSIS 專案。您可以在移除之後重新啟用此 SSIS 選項，以重複使用先前部署至 SSIS 目錄的 SSIS 專案。

主控台

下列程序會移除 SSIS 選項。

從選項群組中移除 SSIS 選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。

2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇具有 SSIS 選項的選項群組 (上述範例中的 `ssis-se-2016`)。
4. 選擇 Delete option (刪除選項)。
5. 在 Deletion options (刪除選項) 下，為 Options to delete (要刪除的選項) 選擇 SSIS。
6. 在 Apply immediately (立即套用) 下，選擇 Yes (是) 立即刪除選項，或選擇 No (否) 在下一個維護時段將其刪除。
7. 選擇刪除。

CLI

下列程序會移除 SSIS 選項。

從選項群組中移除 SSIS 選項

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name ssis-se-2016 \  
  --options SSIS \  
  --apply-immediately
```

在Windows中：

```
aws rds remove-option-from-option-group ^  
  --option-group-name ssis-se-2016 ^  
  --options SSIS ^  
  --apply-immediately
```

卸除 SSISDB 資料庫

移除 SSIS 選項之後，系統不會刪除 SSISDB 資料庫。若要卸除 SSISDB 資料庫，請在移除 SSIS 選項之後使用 `rds_drop_ssis_database` 儲存的程序。

卸除 SSIS 資料庫

- 請使用下列預存程序。

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssis_database
GO
```

卸除 SSISDB 資料庫後，如果您重新啟用 SSIS 選項，您會得到一個新的 SSISDB 目錄。

Amazon RDS for SQL Server 中的 SQL Server Reporting Services 支援

Server Reporting Services (SSRS) 是以伺服器為基礎的應用程式，用於產生和分發報告。它是 SQL Server 服務套件的一部分，該套件也包含 SQL Server Analysis Services (SSAS) 和 SQL Server Integration Services (SSIS)。SSRS 是以 SQL Server 為基礎而建立的服務。您可以使用它來收集來自各種資料來源的資料，並以易於理解且準備好進行分析的方式呈現。

Amazon RDS for SQL Server 支援直接在 RDS 資料庫執行個體上執行 SSRS。您可以在現有或新的資料庫執行個體上使用 SSRS。

下列版本的 RDS 支援 SSRS for SQL Server Standard 和 Enterprise Edition：

- SQL 伺服器，所有版本
- SQL Server 2019，15.00.4043.16.v1 版和更新版本
- SQL Server 2017，14.00.3223.3.v1 版和更新版本
- SQL Server 2016，13.00.5820.21.v1 版和更新版本

內容

- [限制與建議](#)
- [開啟 SSRS](#)
 - [為 SSRS 建立選項群組](#)
 - [將 SSRS 選項新增到您的選項群組](#)
 - [將選項群組與資料庫執行個體建立關聯](#)
 - [允許對 VPC 安全群組進行傳入存取](#)
- [報告伺服器資料庫](#)
- [SSRS 日誌檔案](#)
- [存取 SSR Web 入口網站](#)
 - [在 RDS 上使用 SSL](#)
 - [授予網域使用者的存取權](#)
 - [存取 Web 入口網站](#)
- [將報告部署到 SSRS](#)
- [設定報告資料來源](#)
- [使用 SSRS 電子郵件傳送報告](#)

- [撤銷系統層級許可](#)
- [監控任務的狀態](#)
- [關閉 SSRS](#)
- [刪除 SSRS 資料庫](#)

限制與建議

在 RDS for SQL Server 上執行 SSRS 時適用下列限制和建議：

- 您不能在具有僅供讀取複本的資料庫執行個體上使用 SSRS。
- 執行個體必須使用自我管理的 Active Directory 或 AWS Directory Service for Microsoft Active Directory SSRS 入口網站和網頁伺服器驗證。如需詳細資訊，請參閱 [使用 Active Directory 搭配 RDS for SQL Server](#)。
- 您無法備份使用 SSRS 選項建立的報告伺服器資料庫。
- 不支援從 SSRS 的其他執行個體匯入和還原報告伺服器資料庫。如需詳細資訊，請參閱 [報告伺服器資料庫](#)。
- 您無法將 SSRS 設定為在預設 SSL 連接埠 (443) 上接聽。允許的數值是 1150–49511，但不包括 1234、1434、3260、3343、3389 和 47001。
- 不支援透過 Microsoft Windows 檔案共用的訂閱。
- 不支援使用 Reporting Services Configuration Manager。
- 不支援建立和修改角色。
- 不支援修改報表伺服器屬性。
- 不授予系統管理員和系統使用者角色。
- 您無法透過 Web 入口網站編輯系統層級的角色指派。

開啟 SSRS

使用下列程序為資料庫執行個體開啟 SSRS：

1. 建立新的選項群組或選擇現有的選項群組。
2. 將 SSRS 選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。
4. 對 SSRS 接聽程式連接埠允許 Virtual Private Cloud (VPC) 安全群組的傳入存取。

為 SSRS 建立選項群組

若要使用 SSRS，請建立對應至您打算使用之 SQL Server 引擎和資料庫執行個體版本的選項群組。若要執行這項操作，請使用 AWS Management Console 或 AWS CLI。

Note

如果現有的選項群組適用於正確的 SQL Server 引擎和版本，則也可以使用。

主控台

下列程序會建立 SQL Server Standard Edition 2017 的選項群組。

建立選項群組

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 窗格中，執行下列動作：
 - a. 在「名稱」中，輸入在您的選項群組中唯一的名稱 AWS 帳戶，例如 **ssrs-se-2017**。名稱僅可包含字母、數字與連字號。
 - b. 對於 Description (描述)，請輸入選項群組的簡短描述，例如 **SSRS option group for SQL Server SE 2017**。用於顯示用途的說明。
 - c. 對於 Engine (引擎)，請選擇 sqlserver-se。
 - d. 對於 Major engine version (主要引擎版本)，請選擇 14.00。
5. 選擇 Create (建立)。

CLI

下列程序會建立 SQL Server Standard Edition 2017 的選項群組。

建立選項群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name ssrs-se-2017 \  
  --engine-name sqlserver-se \  
  --major-engine-version 14.00 \  
  --option-group-description "SSRS option group for SQL Server SE 2017"
```

在 Windows 中：

```
aws rds create-option-group ^  
  --option-group-name ssrs-se-2017 ^  
  --engine-name sqlserver-se ^  
  --major-engine-version 14.00 ^  
  --option-group-description "SSRS option group for SQL Server SE 2017"
```

將 SSRS 選項新增到您的選項群組

接下來，使用 AWS Management Console 或將 SSRS 選項新增 AWS CLI 至您的選項群組。

主控台

新增 SSRS 選項

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您剛才建立的選項群組，然後選擇 Add option (新增選項)。
4. 在 Option details (選項詳細資訊) 下，選擇 SSRS 做為 Option name (選項名稱)。
5. 在 Option settings (選項設定) 下，執行下列動作：
 - a. 輸入 SSRS 服務要接聽的連接埠。預設值為 8443。如需允許值的清單，請參閱[限制與建議](#)。
 - b. 輸入 Max memory (最大記憶體) 的數值。

Max memory (最大記憶體) 指定沒有新的記憶體配置請求會授予報告伺服器應用程式的閾值上限。這個數字是資料庫執行個體總記憶體的百分比。允許的數值為 10–80：

- c. 針對 Security groups (安全群組)，選擇要與選項產生關聯的 VPC 安全群組。使用與資料庫執行個體相關聯的相同安全群組。

6. 若要使用 SSRS 電子郵件傳送報告，請選擇 Configure email delivery options (報告服務中的電子郵件傳送) 中的 Email delivery in reporting services (設定電子郵件傳送選項) 核取方塊，然後執行下列動作：

- a. 對於 Sender email address (寄件人電子郵件地址)，使用 From (從) SSRS 電子郵件傳送的郵件部分，輸入您要使用的電子郵件位址。

指定具有從 SMTP 伺服器傳送郵件之權限的使用者帳戶。

- b. 對於 SMTP server (SMTP 伺服器)，指定要使用的 SMTP 伺服器或閘道。

它可以是 IP 地址、公司內部網路上電腦的 NetBIOS 名稱，或是完整合格的網域名稱。

- c. 對於 SMTP port (SMTP 連接埠)，輸入用來連線至郵件伺服器的連接埠。預設為 25。

- d. 若要使用身分驗證：

- i. 請選取 Use authentication (使用身分驗證) 核取方塊。
- ii. 對於秘密 Amazon 資源名稱 (ARN)，請輸入使用者登入資料的 AWS Secrets Manager ARN。

使用下列格式：

arn:aws:secretsmanager:Region:AccountId:secret:SecretName-6RandomChara

例如：

arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3

如需建立祕密的詳細資訊，請參閱 [使用 SSRS 電子郵件傳送報告](#)。

- e. 選取 Use Secure Sockets Layer (SSL) (使用 Secure Sockets Layer (SSL)) 核取方塊，以使用 SSL 加密電子郵件訊息。

7. 在 Scheduling (排程) 下，選擇要立即新增選項或是在下一個維護時段新增選項。

8. 選擇 Add option (新增選項)。

CLI

新增 SSRS 選項

1. 建立 JSON 檔案，例如 `ssrs-option.json`。

a. 設定下列必要參數：

- `OptionGroupName` – 您先前建立或選擇的選項群組名稱 (在下列範例中是 `ssrs-se-2017`)。
- `Port` – SSRS 服務要接聽的連接埠。預設值為 8443。如需允許值的清單，請參閱[限制與建議](#)。
- `VpcSecurityGroupMemberships` – RDS 資料庫執行個體的 VPC 安全群組成員資格。
- `MAX_MEMORY` – 沒有新的記憶體配置請求會授予報告伺服器應用程式的閾值上限。這個數字是資料庫執行個體總記憶體的百分比。允許的數值為 10–80：

b. (選用) 設定使用 SSRS 電子郵件的下列參數：

- `SMTP_ENABLE_EMAIL` - 將 `true` 設為使用 SSRS 電子郵件。預設值為 `false`。
- `SMTP_SENDER_EMAIL_ADDRESS` - SSRS 電子郵件發送訊息時，From (從) 欄位使用的電子郵件地址。指定具有從 SMTP 伺服器傳送郵件之權限的使用者帳戶。
- `SMTP_SERVER` - 使用的 SMTP 伺服器或閘道。可以是 IP 地址、公司內部網路上電腦的 NetBIOS 名稱，或是完整合格的網域名稱。
- `SMTP_PORT` - 用來連線至郵件伺服器的連接埠。預設為 25。
- `SMTP_USE_SSL` - 將 `true` 設為使用 SSL 加密電子郵件。預設值為 `true`。
- `SMTP_EMAIL_CREDENTIALS_SECRET_ARN` - 保存使用者憑證的 Secrets Manager ARN。使用下列格式：

`arn:aws:secretsmanager:Region:AccountId:secret:SecretName-6RandomCharacter`

如需建立秘密的詳細資訊，請參閱 [使用 SSRS 電子郵件傳送報告](#)。

- `SMTP_USE_ANONYMOUS_AUTHENTICATION` - 如果您不想使用身分驗證，請設定為 `true` 並且不包括 `SMTP_EMAIL_CREDENTIALS_SECRET_ARN`。

預設為 `false` 當 `SMTP_ENABLE_EMAIL` 時為 `true`。

下列範例包含使用秘密 ARN 的 SSRS 電子郵件參數。

```
{
  "OptionGroupName": "ssrs-se-2017",
  "OptionsToInclude": [
    {
      "OptionName": "SSRS",
```

```

"Port": 8443,
"VpcSecurityGroupMemberships": ["sg-0abcdef123"],
"OptionSettings": [
  {"Name": "MAX_MEMORY", "Value": "60"},
  {"Name": "SMTP_ENABLE_EMAIL", "Value": "true"},
  {"Name": "SMTP_SENDER_EMAIL_ADDRESS", "Value": "nobody@example.com"},
  {"Name": "SMTP_SERVER", "Value": "email-smtp.us-west-2.amazonaws.com"},
  {"Name": "SMTP_PORT", "Value": "25"},
  {"Name": "SMTP_USE_SSL", "Value": "true"},
  {"Name": "SMTP_EMAIL_CREDENTIALS_SECRET_ARN", "Value":
"arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3"}
]
}],
"ApplyImmediately": true
}

```

2. 將 SSRS 選項新增至選項群組。

Example

對於 Linux macOS、或 Unix：

```

aws rds add-option-to-option-group \
  --cli-input-json file://ssrs-option.json \
  --apply-immediately

```

在 Windows 中：

```

aws rds add-option-to-option-group ^
  --cli-input-json file://ssrs-option.json ^
  --apply-immediately

```

將選項群組與資料庫執行個體建立關聯

使用 AWS Management Console 或將您的選項群組與資料庫執行個體建立關聯。AWS CLI

如果您使用現有資料庫執行個體，則該執行個體必須已有與其相關聯的 Active Directory 網域和 AWS Identity and Access Management (IAM) 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和 IAM 角色。如需詳細資訊，請參閱 [使用 Active Directory 搭配 RDS for SQL Server](#)。

主控台

您可以將選項群組與新的或現有的資料庫執行個體建立關聯。

- 針對新的資料庫執行個體，在啟動執行個體時與選項群組建立關聯。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請修改執行個體並與新的選項群組建立關聯。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

CLI

您可以將選項群組關聯至新的或現有的資料庫執行個體。

建立使用選項群組的資料庫執行個體

- 指定建立選項群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier myssrsinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 14.00.3223.3.v1 \  
  --allocated-storage 100 \  
  --manage-master-user-password \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --domain-iam-role-name my-directory-iam-role \  
  --domain my-domain-id \  
  --option-group-name ssrs-se-2017
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier myssrsinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^
```

```
--engine-version 14.00.3223.3.v1 ^
--allocated-storage 100 ^
--manage-master-user-password ^
--master-username admin ^
--storage-type gp2 ^
--license-model li ^
--domain-iam-role-name my-directory-iam-role ^
--domain my-domain-id ^
--option-group-name ssrs-se-2017
```

修改資料庫執行個體以使用您的選項群組

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier myssrsinstance \  
  --option-group-name ssrs-se-2017 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier myssrsinstance ^  
  --option-group-name ssrs-se-2017 ^  
  --apply-immediately
```

允許對 VPC 安全群組進行傳入存取

若要允許傳入存取與資料庫執行個體相關聯的 VPC 安全群組，請為指定的 SSRS 接聽程式連接埠建立傳入規則。如需設定安全群組的詳細資訊，請參閱[建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

報告伺服器資料庫

當您的資料庫執行個體與 SSRS 選項相關聯時，會在您的資料庫執行個體上建立兩個新的資料庫：

- rdsadmin_ReportServer
- rdsadmin_ReportServerTempDB

這些資料庫充當 ReportServer 和 ReportServerTemp DB 資料庫。SSRS 將其數據存儲在數據 ReportServer 庫中，並將其數據緩存在數據庫數據 ReportServerTemp 庫中。如需詳細資訊，請參閱 Microsoft 文件中的[報表伺服器資料庫](#)。

RDS 擁有和管理這些資料庫，因此不允許在其上進行資料庫操作，如 ALTER 和 DROP。不允許存取 rdsadmin_ReportServerTempDB 資料庫。然而，您可以在 rdsadmin_ReportServer 資料庫上執行讀取操作。

SSRS 日誌檔案

您可以列出、檢視和下載 SSRS 日誌檔案。SSRS 記錄檔遵循 ReportServerService_###記 .log 的命名慣例。這些報表伺服器日誌位於 D:\rdsdbdata\Log\SSRS 目錄中。(D:\rdsdbdata\Log 目錄也是錯誤日誌和 SQL Server Agent 日誌的父目錄)。如需詳細資訊，請參閱[檢視並列出資料庫日誌檔案](#)。

若為現有 SSRS 執行個體，可能需要重新啟動 SSRS 服務才能存取報告伺服器日誌。您可更新 SSRS 選項重新啟動服務。

如需詳細資訊，請參閱[使用 Microsoft SQL Server 日誌](#)。

存取 SSR Web 入口網站

請使用下列程序來存取 SSRS Web 入口網站：

1. 開啟 Secure Sockets Layer (SSL)。
2. 授予網域使用者的存取權。
3. 使用瀏覽器和網域使用者登入資料存取 Web 入口網站。

在 RDS 上使用 SSL

SSRS 使用 HTTPS SSL 通訊協定進行連線。若要使用此通訊協定，請將 SSL 憑證匯入至用戶端電腦上的 Microsoft Windows 作業系統中。

如需 SSL 憑證的詳細資訊，請參閱。如需在使用 SSL 搭配 MySQL 的詳細資訊，請參閱[對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)。

授予網域使用者的存取權

啟用全新的 SSRS 時，SSRS 中不會有角色指派。為了讓網域使用者或使用者群組存取 Web 入口網站，RDS 會提供預存程序。

將存取權授予 Web 入口網站上的網域使用者

- 請使用下列預存程序。

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

對於網域使用者或使用者群組，會授予 RDS_SSRS_ROLE 系統角色。此角色獲授予下列系統層級的任務：

- 執行報告
- 管理任務
- 管理共用排程
- 檢視共用排程

也會授予根資料夾上 Content Manager 的項目層級角色。

存取 Web 入口網站

SSRS_GRANT_PORTAL_PERMISSION 任務順利完成後，您就可以使用網頁瀏覽器存取入口網站。Web 入口網站 URL 具有下列格式。

```
https://rds_endpoint:port/Reports
```

在此格式中，適用以下各項：

- *rds_endpoint* – 您搭配 SSRS 使用之 RDS 資料庫執行個體的端點。

您可以在資料庫執行個體的 Connectivity & security (連線與安全) 標籤上找到端點。如需更多詳細資訊，請參閱 [連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。

- *port* – 您在 SSRS 選項中設定 SSRS 的接聽程式連接埠。

存取 Web 入口網站

1. 在瀏覽器中輸入 Web 入口網站 URL。

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
```

2. 使用您獲授予 SSRS_GRANT_PORTAL_PERMISSION 任務存取權之網域使用者的登入資料登入。

將報告部署到 SSRS

您可以存取入口網站之後，就可以將報告部署至該入口網站。您可以使用 Web 入口網站中的「上傳」工具上傳報告，或直接從 [SQL Server 資料工具 \(SSDT\)](#) 部署。從 SSDT 部署時，請確定下列事項：

- 啟動 SSDT 的使用者可以存取 SSRS 入口網站。
- SSRS 專案屬性中的 TargetServerURL 值會設定為 RDS 資料庫執行個體的 HTTPS 端點，並帶有字尾 ReportServer，例如：

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

設定報告資料來源

將報告部署至 SSRS 之後，您應該設定報告資料來源。設定報告資料來源時，請確保下列事項：

- 對於加入到的 SQL Server 資料庫執行個體的 RDS AWS Directory Service for Microsoft Active Directory，請使用完整網域名稱 (FQDN) 做為連接字串的資料來源名稱。例如 *myssrsinstance.corp-ad.example.com*，其中 *myssrsinstance* 是資料庫執行個體名稱，而且 *corp-ad.example.com* 是完整網域名稱。
- 對於加入至自我管理 Active Directory 的 RDS for SQL Server 資料庫執行個體，請使用 *.* 或 *LocalHost* 做為連線字串的資料來源名稱。

使用 SSRS 電子郵件傳送報告

SSRS 包含 SSRS 電子郵件擴充功能，您可以用來將報告傳送給使用者。

若要設定 SSRS 電子郵件，請使用 SSRS 選項設定。如需詳細資訊，請參閱 [將 SSRS 選項新增到您的選項群組](#)。

設定 SSRS 電子郵件之後，您可以訂閱報告伺服器上的報告。如需詳細資訊，請參閱 Microsoft 文件中的 [《報告服務中的電子郵件傳遞》](#)。

SSRS 電子郵件必須與 AWS Secrets Manager 整合才能在 RDS 上運作。若要與 Secrets Manager 整合，您需要建立秘密。

Note

如果您稍後變更秘密，則也必須更新 SSRS 選項群組中的選項。

若要建立 SSRS 電子郵件的秘密

1. 依照《AWS Secrets Manager 使用者指南》中 [建立秘密](#) 的步驟進行。
 - a. 針對 Select secret type (選取秘密類型)，選擇 Other type of secrets (其他秘密類型)。
 - b. 對於 ey/value pairs (鍵/值對)，輸入下列：
 - **SMTP_USERNAME** - 輸入具有從 SMTP 伺服器傳送郵件權限的使用者。
 - **SMTP_PASSWORD** - 輸入 SMTP 使用者的密碼。
 - c. 對於 Encryption key (加密金鑰)，請勿使用預設 AWS KMS key。使用您自己的現有金鑰或建立新的金鑰。

KMS 金鑰政策必須允許 kms:Decrypt 動作，例如：

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "rds.amazonaws.com"
    ]
  },
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

2. 請遵循 AWS Secrets Manager 使用者指南中 [將許可政策連接至秘密](#) 的步驟。權限政策提供 secretsmanager:GetSecretValue 動作至 rds.amazonaws.com 服務主體。

建議您使用政策中的 `aws:sourceAccount` 和 `aws:sourceArn` 條件金鑰，保護自己免受混淆代理人問題的困擾。使用您 AWS 帳戶的 `aws:sourceAccount` 和選項群組 ARN。 `aws:sourceArn` 如需詳細資訊，請參閱 [防止跨服務混淆代理人問題](#)。

以下範例示範許可政策。

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "rds.amazonaws.com"
    },
    "Action" : "secretsmanager:GetSecretValue",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "aws:sourceAccount" : "123456789012"
      },
      "ArnLike" : {
        "aws:sourceArn" : "arn:aws:rds:us-west-2:123456789012:og:ssrs-se-2017"
      }
    }
  } ]
}
```

如需更多範例，請參閱 AWS Secrets Manager 使用者指南中的 AWS Secrets Manager 的 [權限原則範例](#)。

撤銷系統層級許可

RDS_SSRS_ROLE 系統角色沒有足夠的許可，無法刪除系統層級角色指派。若要從 RDS_SSRS_ROLE 中移除使用者或使用者群組，請使用您用來授予角色的相同預存程序，但使用 SSRS_REVOKE_PORTAL_PERMISSION 任務類型。

撤銷 Web 入口網站的網域使用者存取權

- 請使用下列預存程序。

```
exec msdb.dbo.rds_msbi_task
```

```
@task_type='SSRS_REVOKE_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

這麼做會將使用者從 RDS_SSRS_ROLE 系統角色中刪除。如果使用者擁有 Content Manager 項目層級角色，也會從該項目層級角色中刪除該使用者。

監控任務的狀態

若要追蹤授予或撤銷任務的狀態，請呼叫 `rds_fn_task_status` 函數。需要兩個參數。第一個參數不適用於 SSAS，所以應該一律為 NULL。第二個參數接受任務 ID。

若要查看所有任務的清單，請將第一個參數設為 NULL，將第二個參數設為 0，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

若要取得特定的任務，請將第一個參數設為 NULL，將第二個參數設為任務 ID，如下列範例所示。

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

`rds_fn_task_status` 函數會傳回下列資訊。

輸出參數	描述
<code>task_id</code>	任務的 ID。
<code>task_type</code>	對於 SSRS，任務可以有列任務類型： <ul style="list-style-type: none"> SSRS_GRANT_PORTAL_PERMISSION SSRS_REVOKE_PORTAL_PERMISSION
<code>database_name</code>	不適用於 SSRS 任務。
<code>% complete</code>	任務的進度 (以百分比表示)。
<code>duration (mins)</code>	任務所花的時間 (以分鐘為單位)。
<code>lifecycle</code>	任務的狀態。可能的狀態如下： <ul style="list-style-type: none">

輸出參數	描述
	<p>CREATED – 呼叫其中一個 SSRS 預存程序之後，就會建立任務，而且狀態設為 CREATED。</p> <ul style="list-style-type: none"> IN_PROGRESS – 任務開始之後，狀態會設為 IN_PROGRESS 。可能需要經過 5 分鐘，狀態才會從 CREATED 變成 IN_PROGRESS 。 SUCCESS – 任務完成之後，狀態會設為 SUCCESS。 ERROR – 如果任務失敗，狀態會設為 ERROR。關於錯誤的詳細資訊，請查看 <code>task_info</code> 欄。 CANCEL_REQUESTED – 呼叫 <code>rds_cancel_task</code> 預存程序之後，任務的狀態會設定為 CANCEL_REQUESTED 。 CANCELLED – 成功取消任務之後，任務的狀態會設為 CANCELLED 。
<code>task_info</code>	任務的其他資訊。如果處理期間發生錯誤，此欄包含該錯誤的相關資訊。
<code>last_updated</code>	上次更新任務狀態的日期和時間。
<code>created_at</code>	建立任務的日期和時間。
<code>S3_object_arn</code>	不適用於 SSRS 任務。
<code>overwrite_S3_backup_file</code>	不適用於 SSRS 任務。
<code>KMS_master_key_arn</code>	不適用於 SSRS 任務。

輸出參數	描述
filepath	不適用於 SSRS 任務。
overwrite_file	不適用於 SSRS 任務。
task_metadata	與 SSRS 任務相關聯的中繼資料。

關閉 SSRS

若要關閉 SSRS，請從其選項群組中移除 SSRS 選項。移除此選項並不會刪除 SSRS 資料庫。如需詳細資訊，請參閱 [刪除 SSRS 資料庫](#)。

您可以通過新增回 SSRS 選項來再次打開 SSRS。如果您也已刪除 SSRS 資料庫，在相同的資料庫執行個體上重新新增 SSRS 會建立新的報告伺服器資料庫。

主控台

從選項群組中移除 SSRS 選項

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇具有 SSRS 選項的選項群組 (上述範例中的 `ssrs-se-2017`)。
4. 選擇 Delete option (刪除選項)。
5. 在 Deletion options (刪除選項) 下，為 Options to delete (要刪除的選項) 選擇 SSRS。
6. 在 Apply immediately (立即套用) 下，選擇 Yes (是) 立即刪除選項，或選擇 No (否) 在下一個維護時段將其刪除。
7. 選擇刪除。

CLI

從選項群組中移除 SSRS 選項

- 請執行下列其中一個命令：

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name ssrs-se-2017 \  
  --options SSRS \  
  --apply-immediately
```

在 Windows 中：

```
aws rds remove-option-from-option-group ^  
  --option-group-name ssrs-se-2017 ^  
  --options SSRS ^  
  --apply-immediately
```

刪除 SSRS 資料庫

移除 SSRS 選項並不會刪除報告伺服器資料庫。若要刪除它們，請使用下列預存程序。

若要刪除報告伺服器資料庫，請務必先移除 SSRS 選項。

刪除 SSRS 資料庫

- 請使用下列預存程序。

```
exec msdb.dbo.rds_drop_ssrs_databases
```

RDS for SQL Server 中的 Microsoft Distributed Transaction Coordinator 支援

分散式交易是涉及兩個或多個網路主機的資料庫交易。RDS for SQL Server 支援主機之間的分散式交易，其中單一主機可以是下列其中一項：

- RDS for SQL Server 資料庫執行個體
- 內部部署 SQL Server 主機
- 已安裝 SQL Server 的 Amazon EC2 主機
- 任何其他具有支援分散式交易之資料庫引擎的 EC2 主機或 RDS 資料庫執行個體

在 RDS 中，自 SQL Server 2012 (11.00.5058.0.v1 版及更新版本) 開始，RDS for SQL Server 的所有版本都支援分散式交易。支援是使用 Microsoft Distributed Transaction Coordinator (MSDTC) 提供。如需 MSDTC 的深入資訊，請參閱 Microsoft 文件中的 [分散式交易協調器](#)。

內容

- [限制](#)
- [啟用 MSDTC](#)
 - [建立 MSDTC 的選項群組](#)
 - [將 MSDTC 選項新增至選項群組](#)
 - [建立 MSDTC 的參數群組](#)
 - [修改 MSDTC 的參數](#)
 - [將選項群組和參數群組與資料庫執行個體建立關聯](#)
- [使用分散式交易](#)
- [使用 XA 交易](#)
- [使用交易追蹤](#)
- [修改 MSDTC 選項](#)
- [停用 MSDTC](#)
- [對 RDS for SQL Server 的 MSDTC 進行故障診斷](#)

限制

在 RDS for SQL Server 上使用 MSDTC 時適用下列限制：

- 使用 SQL Server 資料庫鏡像的執行個體不支援 MSDTC。如需詳細資訊，請參閱[交易 - 可用性群組和資料庫鏡像](#)。
- `in-doubt xact resolution` 參數必須設為 1 或 2。如需詳細資訊，請參閱[修改 MSDTC 的參數](#)。
- MSDTC 要求所有參與分散式交易的主機皆可使用其主機名稱進行解析。RDS 會自動為加入網域的執行個體維護此功能。不過，對於獨立執行個體，請務必手動設定 DNS 伺服器。
- SQL Server 2017 14.00.3223.3 版和更新版本以及 SQL Server 2019 支援 Java Database Connectivity (JDBC) XA 交易。
- 在 RDS 執行個體上不支援依存於用戶端動態連結程式庫 (DLL) 的分散式交易。
- 不支援使用自訂 XA 動態連結程式庫。

啟用 MSDTC

請使用下列程序為您的資料庫執行個體啟用 MSDTC：

1. 建立新的選項群組或選擇現有的選項群組。
2. 將 MSDTC 選項新增至選項群組。
3. 建立新參數群組，或選擇現有參數群組。
4. 修改參數群組，將 `in-doubt xact resolution` 參數設定為 1 或 2。
5. 將選項群組和參數群組與資料庫執行個體建立關聯。

建立 MSDTC 的選項群組

使用 AWS Management Console 或 AWS CLI 來建立對應至 SQL Server 引擎和資料庫執行個體版本的選項群組。

Note

如果現有的選項群組適用於正確的 SQL Server 引擎和版本，則也可以使用。

主控台

下列程序會建立 SQL Server Standard Edition 2016 的選項群組。

建立選項群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇 Create group (建立群組)。
4. 在 Create option group (建立選項群組) 窗格中，執行下列動作：
 - a. 在 Name (名稱) 中，輸入您 AWS 帳戶中的唯一選項群組名稱，例如 **msdtc-se-2016**。名稱僅可包含字母、數字與連字號。
 - b. 對於 Description (描述)，請輸入選項群組的簡短描述，例如 **MSDTC option group for SQL Server SE 2016**。用於顯示用途的說明。
 - c. 對於 Engine (引擎)，請選擇 **sqlserver-se**。
 - d. 對於 Major engine version (主要引擎版本)，請選擇 **13.00**。
5. 選擇 Create (建立)。

CLI

下列範例會為 SQL Server Standard Edition 2016 建立選項群組。

建立選項群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-option-group \  
  --option-group-name msdtc-se-2016 \  
  --engine-name sqlserver-se \  
  --major-engine-version 13.00 \  
  --option-group-description "MSDTC option group for SQL Server SE 2016"
```

在Windows中：

```
aws rds create-option-group ^  
  --option-group-name msdtc-se-2016 ^
```



```
--engine-name sqlserver-se ^  
--major-engine-version 13.00 ^  
--option-group-description "MSDTC option group for SQL Server SE 2016"
```

將 MSDTC 選項新增至選項群組

接下來，使用 AWS Management Console 或 AWS CLI 將 MSDTC 選項加入至選項群組。

下列選項設定是必要的：

- Port (連接埠) – 您用來存取 MSDTC 的連接埠。允許的數值是 1150–49151，但不包括 1234、1434、3260、3343、3389 和 47001。預設值為 5000。

確定您要使用的連接埠已在防火牆規則中啟用。此外，請確定在與資料庫執行個體相關聯的安全群組的傳入和傳出規則中已啟用此連接埠。如需詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

- Security groups (安全群組) – RDS 資料庫執行個體的 VPC 或安全群組成員資格。
- Authentication (身分驗證類型) – 主機之間的身分驗證模式。支援下列身分驗證類型：
 - 相互驗證 – RDS 執行個體使用整合式驗證彼此相互驗證。如果選取此選項，與此選項群組相關聯的所有執行個體都必須加入網域。
 - 無 – 主機之間不執行身分驗證。我們不建議您生產環境中使用此模式。
- 交易日誌大小 – MSDTC 交易日誌的大小。允許的數值是 4–1024 MB。預設大小為 4 MB。

下列選項設定是選擇性的：

- Enable inbound connections (啟用傳入連線) – 是否允許對與此選項群組相關聯的執行個體進行傳入 MSDTC 連線。
- Enable outbound connections (啟用輸出連線) – 是否允許從此選項群組相關聯的執行個體進行傳出 MSDTC 連線。
- Enable XA (啟用 XA) – 是否允許 XA 交易。如需 XA 通訊協定的詳細資訊，請參閱 [XA 規格](#)。
- Enable SNA LU (啟用 SNA LU) – 是否允許 SNA LU 通訊協定用於分散式交易。如需 SNA LU 通訊協定支援的詳細資訊，請參閱 Microsoft 文件中的 [管理 IBM CICS LU 6.2 交易](#)。

主控台

新增 MSDTC 選項

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇您剛建立的選項群組。
4. 選擇 Add option (新增選項)。
5. 在 Option details (選項詳細資訊) 下，選擇 MSDTC 做為 Option name (選項名稱)。
6. 在 Options settings (選項設定) 下：
 - a. 針對 Port (連接埠)，輸入存取 MSDTC 的連接埠號碼。預設值為 5000。
 - b. 針對 Security groups (安全群組)，選擇要與選項產生關聯的 VPC 安全群組。
 - c. 在 Authentication type (身分驗證類型) 中，選擇 Mutual (相互) 或 None (無)。
 - d. 針對 Transaction log size (交易日誌大小)，輸入 4–1024 之間的值。預設值為 4。
7. 在 Additional configuration (其他設定) 下，請執行下列動作：
 - a. 針對 Connections (連線)，視需要選擇 Enable inbound connections (啟用傳入連線) 和 Enable outbound connections (啟用傳出連線)。
 - b. 針對 Allowed protocols (允許的通訊協定)，視需要選擇 Enable XA (啟用 XA) 和 Enable SNA LU (啟用 SNA LU)。
8. 在 Scheduling (排程) 下，選擇要立即新增選項或是在下一個維護時段新增選項。
9. 選擇 Add option (新增選項)。

若要新增此選項，不需要重新開機。

CLI

新增 MSDTC 選項

1. 使用下列必要的參數建立 JSON 檔案，例如 `msdtc-option.json`。

```
{
  "OptionGroupName": "msdtc-se-2016",
  "OptionsToInclude": [
```

```
{
  "OptionName": "MSDTC",
  "Port": 5000,
  "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
  "OptionSettings": [{"Name": "AUTHENTICATION", "Value": "MUTUAL"},
{"Name": "TRANSACTION_LOG_SIZE", "Value": "4"}]
}],
"ApplyImmediately": true
}
```

2. 將 MSDTC 選項新增至選項群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \
  --cli-input-json file://msdtc-option.json \
  --apply-immediately
```

在Windows中：

```
aws rds add-option-to-option-group ^
  --cli-input-json file://msdtc-option.json ^
  --apply-immediately
```

不需要重新開機。

建立 MSDTC 的參數群組

為對應到 SQL Server 版本和資料庫執行個體版本的 in-doubt xact resolution 參數建立或修改參數群組。

主控台

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。

2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在 Create parameter group (建立參數群組) 窗格中執行下列動作：
 - a. 對於 Parameter group family (參數群組家族)，請選擇 `sqlserver-se-13.0`。
 - b. 對於 Group name (群組名稱)，輸入參數群組的識別碼，例如 `msdtc-sqlserver-se-13`。
 - c. 對於 Description (說明)，輸入 `in-doubt xact resolution`。
5. 選擇 Create (建立)。

CLI

下列範例會建立 SQL Server Standard Edition 2016 的參數群組。

建立參數群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name msdtc-sqlserver-se-13 \  
  --db-parameter-group-family "sqlserver-se-13.0" \  
  --description "in-doubt xact resolution"
```

在Windows中：

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name msdtc-sqlserver-se-13 ^  
  --db-parameter-group-family "sqlserver-se-13.0" ^  
  --description "in-doubt xact resolution"
```

修改 MSDTC 的參數

修改參數群組中對應至 SQL Server 版本和資料庫執行個體版本的 `in-doubt xact resolution` 參數。

對於 MSDTC，請將 `in-doubt xact resolution` 參數設定為下列其中一項：

- 1 – Presume commit。任何有疑問的 MSDTC 交易都假設已經遞交。
- 2 – Presume abort。任何有疑問的 MSDTC 交易都假設已停止。

如需詳細資訊，請參閱 Microsoft 文件中的[有疑問的 xact 解析伺服器組態選項](#)。

主控台

下列範例會修改您為 SQL Server Standard Edition 2016 所建立的參數群組。

修改參數群組

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇參數群組，例如 msdtc-sqlserver-se-13。
4. 在 Parameters (參數) 下，篩選 **xact** 的參數清單。
5. 選擇 in-doubt xact resolution (有疑問的 xact 解析)。
6. 選擇 Edit parameters (編輯參數)。
7. 輸入 **1** 或 **2**。
8. 選擇儲存變更。

CLI

下列範例會修改您為 SQL Server Standard Edition 2016 所建立的參數群組。

修改參數群組

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name msdtc-sqlserver-se-13 \  
  --parameters "ParameterName='in-doubt xact  
  resolution',ParameterValue=1,ApplyMethod=immediate"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name msdtc-sqlserver-se-13 ^
  --parameters "ParameterName='in-doubt xact
  resolution',ParameterValue=1,ApplyMethod=immediate"
```

將選項群組和參數群組與資料庫執行個體建立關聯

您可以使用AWS Management Console或 AWS CLI，將 MSDTC 選項群組和參數群組與資料庫執行個體產生關聯。

主控台

您可以將 MSDTC 選項群組和參數群組與新的或現有的資料庫執行個體產生關聯。

- 對於新的資料庫執行個體，請在啟動執行個體時將它們相關聯。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 對於現有的資料庫執行個體，請透過修改執行個體來建立關聯。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Note

如果您使用現有資料庫執行個體，則該執行個體必須已具有與其相關聯的 Active Directory 網域和 AWS Identity and Access Management IAM 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和 IAM 角色。如需詳細資訊，請參閱 [使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄](#)。

CLI

您可以將 MSDTC 選項群組和參數群組與新的或現有的資料庫執行個體產生關聯。

Note

如果您使用現有已加入網域的資料庫執行個體，則該執行個體必須已具有與其相關聯的 Active Directory 網域和 IAM 角色。如果您建立新執行個體，請指定現有的 Active Directory 網域和

IAM 角色。如需更多詳細資訊，請參閱 [使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄](#)。

使用 MSDTC 選項群組和參數群組建立資料庫執行個體

- 指定建立選項群組時所使用的相同資料庫引擎類型和主要版本。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-instance-class db.m5.2xlarge \  
  --engine sqlserver-se \  
  --engine-version 13.00.5426.0.v1 \  
  --allocated-storage 100 \  
  --manage-master-user-password \  
  --master-username admin \  
  --storage-type gp2 \  
  --license-model li \  
  --domain-iam-role-name my-directory-iam-role \  
  --domain my-domain-id \  
  --option-group-name msdtc-se-2016 \  
  --db-parameter-group-name msdtc-sqlserver-se-13
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --db-instance-class db.m5.2xlarge ^  
  --engine sqlserver-se ^  
  --engine-version 13.00.5426.0.v1 ^  
  --allocated-storage 100 ^  
  --manage-master-user-password ^  
  --master-username admin ^  
  --storage-type gp2 ^  
  --license-model li ^  
  --domain-iam-role-name my-directory-iam-role ^  
  --domain my-domain-id ^  
  --option-group-name msdtc-se-2016 ^
```

```
--db-parameter-group-name msdtc-sqlserver-se-13
```

修改資料庫執行個體並建立 MSDTC 選項群組和參數群組的關聯

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --option-group-name msdtc-se-2016 \  
  --db-parameter-group-name msdtc-sqlserver-se-13 \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --option-group-name msdtc-se-2016 ^  
  --db-parameter-group-name msdtc-sqlserver-se-13 ^  
  --apply-immediately
```

使用分散式交易

在 Amazon RDS for SQL Server 中，您執行分散式交易的方式與在內部部署執行的分散式交易相同：

- 使用 .NET Framework System.Transactions 可升級的交易，透過延遲到有需要時再建立交易，可最佳化分散式交易。

在這種情況下，升級是自動的，而不需要您進行任何干預。如果交易中只有一個資源管理員，則不會執行任何升級。如需隱含交易範圍的詳細資訊，請參閱 Microsoft 文件中的[使用交易範圍實作隱含交易](#)。

以下 .NET 實作支援可升級的交易：

- 自 ADO.NET 2.0 開始，System.Data.SqlClient 支援與 SQL Server 進行可升級的交易。如需詳細資訊，請參閱 Microsoft 文件中的[與 SQL Server 的 System.Transactions 整合](#)。

- ODP.NET 支援 System.Transactions。在 TransactionsScope 範圍中對 Oracle 資料庫 11g 版本 1 (11.1 版) 及更新版本開啟的第一個連線會建立本機交易。當開啟第二個連線時，會自動將此交易升級為分散式交易。如需 ODP.NET 中的分散式交易支援的詳細資訊，請參閱 Microsoft 文件中的 [Microsoft Distributed Transaction Coordinator 整合](#)。
- 使用 BEGIN DISTRIBUTED TRANSACTION 陳述式。如需詳細資訊，請參閱 Microsoft 文件中的 [BEGIN DISTRIBUTED TRANSACTION \(Transact-SQL\)](#)。

使用 XA 交易

從 RDS for SQL Server 2017 14.00.3223.3 版開始，您可以使用 JDBC 控制分散式交易。當您在 MSDTC 選項中，將 Enable XA 選項設定為 true 時，RDS 會自動啟用 JDBC 交易，並將 SqlJDBCXAUser 角色授與 guest 使用者。如此可允許透過 JDBC 執行分佈式交易。如需包含程式碼範例在內的詳細資訊，請參閱 Microsoft 文件中的 [認識 XA 交易](#)。

使用交易追蹤

RDS 支援控制 MSDTC 交易追蹤，以及從 RDS 資料庫執行個體下載追蹤以進行故障診斷。您可以執行下列 RDS 預存程序來控制交易追蹤工作階段。

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',
[@traceall='0/1'],
[@traceaborted='0/1'],
[@tracelong='0/1'];
```

以下是必要參數：

- trace_action – 追蹤動作。其可能是 START、STOP 或 STATUS。

下列是選用參數：

- @traceall – 設定為 1 可追蹤所有分散式交易。預設為 0。
- @traceaborted – 設定為 1 可追蹤已取消的分散式交易。預設為 0。
- @tracelong – 設定為 1 可追蹤長時間執行的分散式交易。預設為 0。

Example 開始追蹤動作

若要開始新的交易追蹤工作階段，請執行下列範例陳述式。

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

Note

一次只能使用一個交易追蹤工作階段。如果在追蹤工作階段作用中時發出新的追蹤工作階段 START 命令，則會傳回錯誤，且使用中的追蹤工作階段會維持不變。

Example 停止追蹤動作

若要停止交易追蹤工作階段，請執行下列陳述式。

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

此陳述式會停止作用中的交易追蹤工作階段，並將交易追蹤資料儲存至 RDS 資料庫執行個體上的日誌目錄。輸出的第一列包含整體結果，後續幾行指出該操作的詳細資訊。

以下是成功停止追蹤工作階段的範例。

OK: Trace session has been successfully stopped.

```
Setting log file to: D:\rdsbdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsbdbdata\MSDTC\Trace\msdtctr.mof for message formats, 8 found.
Searching for TMF files on path: (null)
Logfile D:\rdsbdbdata\MSDTC\Trace\dtctrace.log:
OS version      10.0.14393 (Currently running on 6.2.9200)
Start Time      <timestamp>
End Time        <timestamp>
Timezone is     @tzres.dll,-932 (Bias is 0mins)
BufferSize      16384 B
Maximum File Size 10 MB
Buffers Written  Not set (Logger may not have been stopped).
Logger Mode Settings (11000002) ( circular paged
ProcessorCount  1
Processing completed Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0,
Unknowns: 3
Event traces dumped to d:\rdsbdbdata\Log\msdtc_<timestamp>.log
```

您可以使用詳細資訊來查詢產生之日誌檔案的名稱。如需從 RDS 資料庫執行個體下載日誌檔案的詳細資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。

追蹤工作階段記錄會在執行個體上保留 35 天。系統會自動刪除任何較舊的追蹤工作階段日誌。

Example 狀態追蹤動作

若要追蹤交易追蹤工作階段的狀態，請執行下列陳述式。

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

此陳述式會將下列項目輸出為結果集不同的列。

```
OK
SessionStatus: <Started/Stopped>
TraceAll: <True/False>
TraceAborted: <True/False>
TraceLongLived: <True/False>
```

第一行指出該操作的整體結果：OK 或 ERROR，加上詳細資訊 (如果適用)。後續各行指出有關追蹤工作階段狀態的詳細資訊：

- SessionStatus 可為下列其中之一：
 - Started 如果追蹤工作階段正在執行中。
 - Stopped 如果沒有追蹤工作階段正在執行中。
- 追蹤工作階段旗標可以是 True 或 False，取決於其在 START 命令中的設定方式。

修改 MSDTC 選項

啟用 MSDTC 選項後，您可以修改其設定。如需如何修改選項設定的詳細資訊，請參閱 [修改選項設定](#)。

Note

MSDTC 選項設定的某些變更需要重新啟動 MSDTC 服務。此需求可能會影響執行中的分散式交易。

停用 MSDTC

若要停用 SSAS，請將 MSDTC 選項從其選項群組中移除。

主控台

將 MSDTC 選項從其選項群組中移除

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Option groups (選項群組)。
3. 選擇具有 MSDTC 選項的選項群組 (上述範例中的 `msdtc-se-2016`)。
4. 選擇 Delete option (刪除選項)。
5. 在 Deletion options (刪除選項) 下，為 Options to delete (要刪除的選項) 選擇 MSDTC。
6. 在 Apply immediately (立即套用) 下，選擇 Yes (是) 立即刪除選項，或選擇 No (否) 在下一個維護時段將其刪除。
7. 選擇刪除。

CLI

將 MSDTC 選項從其選項群組中移除

- 請使用下列其中一個命令。

Example

對於LinuxmacOS、或Unix：

```
aws rds remove-option-from-option-group \  
  --option-group-name msdtc-se-2016 \  
  --options MSDTC \  
  --apply-immediately
```

在Windows中：

```
aws rds remove-option-from-option-group ^\  
  --option-group-name msdtc-se-2016 ^\  
  --options MSDTC ^\  
  --apply-immediately
```

對 RDS for SQL Server 的 MSDTC 進行故障診斷

在某些情況下，您可能無法在用戶端電腦上執行的 MSDTC 與 RDS for SQL Server 資料庫執行個體上執行的 MSDTC 服務之間建立連線。若是如此，請確定下列事項：

- 與資料庫執行個體相關聯之安全群組的傳入規則已正確設定。如需更多詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。
- 您的用戶端電腦設定正確。
- 您的用戶端電腦上已啟用 MSDTC 防火牆規則。

設定用戶端電腦

1. 開啟 Component Services (元件服務)。


或者，在 Server Manager (伺服器管理員) 中，選擇 Tools (工具)，然後選擇 Component Services (元件服務)。

2. 展開 Component Services (元件服務)，展開 Computers (電腦)，展開 My Computer (我的電腦)，然後展開 分散式交易協調器。
3. 開啟 Local DTC (本機 DTC) 的內容 (按右鍵) 功能表，然後選擇 Properties (屬性)。
4. 選擇 Security (安全) 標籤。
5. 選擇下列所有項目：
 - Network DTC Access (網路 DTC 存取)
 - Allow Inbound (允許傳入)
 - Allow Outbound (允許傳出)
6. 請確定已選擇正確的身分驗證模式：
 - Mutual Authentication Required (需要相互驗證) – 用戶端電腦與參與分散式交易的其他節點加入相同的網域，或是網域之間已設定信任關係。
 - No Authentication Required (不需要驗證) – 所有其他情況。
7. 選擇 OK (確定) 以儲存變更。
8. 如果系統提示您重新啟動服務，請選擇 Yes (是)。

啟用 MSDTC 防火牆規則

1. 開啟 Windows 防火牆，然後選擇 Advanced settings (進階設定)。

開啟 Server Manager (伺服器管理員)，選擇 Tools (工具)，然後選取 Windows Firewall with Advanced Security (具有進階安全性的 Windows 防火牆)。

 Note

視您的作業系統而定，Windows 防火牆可能稱為 Windows Defender 防火牆。

2. 在左窗格中選擇 Inbound Rules (傳入規則)。
3. 啟用下列防火牆規則 (如果尚未啟用)：
 - Distributed Transaction Coordinator (RPC)
 - Distributed Transaction Coordinator (RPC)-EPMAP
 - Distributed Transaction Coordinator (TCP-In)
4. 關閉 Windows 防火牆。

Microsoft SQL Server 的常用 DBA 任務

本節針對執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體，說明一些一般 DBA 任務的 Amazon RDS 特定實作。為提供受管服務體驗，Amazon RDS 不提供資料庫執行個體的殼層存取權，而且會限制存取某些需要進階權限的系統程序和資料表。

Note

使用 SQL Server 資料庫執行個體時，您可以執行指令碼來修改新建立的資料庫，但無法修改 [model] 資料庫，即用做為新資料庫模型的資料庫。

主題

- [在 Amazon RDS 上存取 Microsoft SQL Server 資料庫執行個體上的 tempdb 資料庫](#)
- [使用 Database Engine Tuning Advisor 分析 Amazon RDS for SQL Server 資料庫執行個體上的資料庫工作負載](#)
- [將資料庫的 db_owner 變更為 rdsa 帳戶](#)
- [Microsoft SQL Server 的定序和字元集](#)
- [建立資料庫使用者](#)
- [判斷 Microsoft SQL Server Database 的復原模型](#)
- [判斷上次容錯移轉時間](#)
- [在大量載入期間停用快速插入](#)
- [捨棄 Microsoft SQL Server 資料庫](#)
- [重新命名多個可用區部署中的 Microsoft SQL Server 資料庫](#)
- [重新設定 db_owner 角色密碼](#)
- [還原授權終止的資料庫執行個體](#)
- [將 Microsoft SQL Server 資料庫從「離線」轉移至「上線」](#)
- [使用變更資料擷取](#)
- [使用 SQL Server 代理程式](#)
- [使用 Microsoft SQL Server 日誌](#)
- [使用追蹤和傾印檔案](#)

在 Amazon RDS 上存取 Microsoft SQL Server 資料庫執行個體上的 tempdb 資料庫

您可以在 Amazon RDS 上存取 Microsoft SQL Server 資料庫執行個體上的 tempdb 資料庫。您可以使用 Transact-SQL 透過 Microsoft SQL Server Management Studio (SSMS)，或任何其他標準 SQL 用戶端應用程式，在 tempdb 上執程式碼。如需連接至資料庫執行個體的詳細資訊，請參閱[連接至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體](#)。

資料庫執行個體的主要使用者會獲授予 CONTROL 的 tempdb 存取權，以便此使用者可以修改 tempdb 資料庫選項。主要使用者不是 tempdb 資料庫的資料庫擁有者。必要時，主要使用者可將 CONTROL 存取權授予其他使用者，以便他們也可以修改 tempdb 資料庫選項。

Note

您無法在 tempdb 資料庫上執行資料庫主控台命令 (DBCC)。

修改 tempdb 資料庫選項

您可以在 Amazon RDS 資料庫執行個體上修改 tempdb 資料庫上的資料庫選項。如需可以修改哪些選項的詳細資訊，請參閱 Microsoft 文件中的[tempdb 資料庫](#)。

在重新啟動您的資料庫執行個體之後，檔案大小上限選項之類的資料庫選項會持續存在。您可以修改資料庫選項，以在匯入資料時將效能提高到最佳化，並避免用光儲存體。

匯入資料時將效能提高到最佳化

在將大量資料匯入至資料庫執行個體時，若要將效能提高到最佳化，請將 tempdb 資料庫的 SIZE 和 FILEGROWTH 屬性設定為大數字。如需如何將 tempdb 最佳化的詳細資訊，請參閱 Microsoft 文件中的[將 tempdb 效能最佳化](#)。

下列範例示範將大小設定為 100 GB，並將檔案成長率設定為 10%。

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

防止儲存體問題

若要防止 tempdb 資料庫使用所有可用的磁碟空間，請設定 MAXSIZE 屬性。下列範例示範將屬性設定為 2048 MB。


```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

縮減 tempdb 資料庫

有兩種方式可縮減 Amazon RDS 資料庫執行個體上的 tempdb 資料庫。您可以使用 rds_shrink_tempdbfile 程序，或設定 SIZE 屬性。

使用 rds_shrink_tempdbfile 程序

您可以使用 Amazon RDS 程序 msdb.dbo.rds_shrink_tempdbfile，來縮減 tempdb 資料庫。只有在您對 rds_shrink_tempdbfile 具有 CONTROL 存取權時，才能呼叫 tempdb。當您呼叫 rds_shrink_tempdbfile 時，資料庫執行個體沒有停機時間。

rds_shrink_tempdbfile 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
@temp_filename	SYSNAME	—	必要	要縮減之檔案的邏輯名稱。
@target_size	int	null	選擇性	檔案的新大小 (以 MB 為單位)。

下列範例會取得 tempdb 資料庫的檔案名稱。

```
use tempdb;
GO

select name, * from sys.sysfiles;
GO
```

下列範例會縮減名為 tempdb 的 test_file 資料庫檔案，並要求 10 MB 的新大小：

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

設定 SIZE 屬性

您也可以設定 tempdb 屬性，然後重新啟動資料庫執行個體，來縮減 SIZE 資料庫。如需重新啟動資料庫執行個體的詳細資訊，請參閱[重新啟動中的資料庫執行個體](#)。

下列範例示範將 SIZE 屬性設定為 1024 MB。

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

異地同步備份部署的 TempDB 組態

如果您的 RDS for SQL Server 資料庫執行個體位於使用資料庫鏡像 (DBM) 或永遠開啟可用性群組 (AGG) 的異地同步備份部署中，請記住下列使用資料庫的注意事項。tempdb

您無法將主要tempdb資料庫執行個體的資料複製到次要資料庫執行個體。當您容錯移轉至次要資料庫執行個體時，該次要資料庫執行個體tempdb上將會是空白的。

您可以將tempdb資料庫選項的組態 (包括其檔案大小和自動成長設定) 從主資料庫執行個體同步到次要資料庫執行個體。SQL 伺服器版本的所有 RDS 都支援同步處理tempDB組態。您可以使用下列預存程序來開啟tempdb組態的自動同步處理：

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'TempDbFile';
```

Important

使用rds_set_system_database_sync_objects預存程序之前，請確定您已在主要資料庫執行個體 (而非次要資料庫執行個體) 上設定偏好的tempdb組態。如果您在次要資料庫執行個體上進行了組tempdb態變更，則當您開啟自動同步處理時，可能會刪除偏好的組態。

您可以使用下列功能來確認是否已開啟tempdb組態的自動同步處理：

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

開啟tempdb組態的自動同步處理時，object_class欄位會有傳回值。當它關閉時，不會返回任何值。

您可以使用以下函數來查找上次同步對象的時間，以 UTC 時間為單位：

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

例如，如果您在 01:00 修改tempdb組態，然後執行rds_fn_server_object_last_sync_time函數，則傳回的值last_sync_time應該在 01:00 之後，表示已進行自動同步處理。

如果您也使用 SQL Server 代理程式工作複寫，則可以在@object_type參數中提供 SQL 代理程式工作和tempdb組態來啟用複寫：

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =  
'SQLAgentJob,TempDbFile';
```

如需有關 SQL Server 代理程式工作複寫的詳細資訊，請參閱[開啟 SQL Server Agent 任務複寫](#)。

除了使用rds_set_system_database_sync_objects預存程序以確保tempdb組態變更會自動同步處理之外，您可以使用下列其中一種手動方法：

Note

建議您使用rds_set_system_database_sync_objects預存程序開啟tempdb組態的自動同步處理。使用自動同步處理可防止每次變更tempdb組態時執行這些手動工作。

- 首先修改您的資料庫執行個體並關閉異地同步備份，然後修改 tempdb，最後再次開啟異地同步備份。此方法不涉及任何停機時間。

如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

- 首先修改原始主要執行個體中的 tempdb，然後手動容錯移轉，最後修改新主要執行個體中的 tempdb。此方法涉及停機時間。

如需詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

使用 Database Engine Tuning Advisor 分析 Amazon RDS for SQL Server 資料庫執行個體上的資料庫工作負載

Database Engine Tuning Advisor 為 Microsoft 提供的一項用戶端應用程式，可分析資料庫工作負載並可根據您執行的查詢，為 Microsoft SQL Server 資料庫建議一組理想的索引。如同 SQL Server Management Studio，您可以從與您執行 SQL Server 的 Amazon RDS 資料庫執行個體連接的用戶端電腦執行 Tuning Advisor。用戶端電腦可以是在您自己網路內的現場部署執行的本機電腦，也可以是在與 Amazon RDS 資料庫執行個體相同區域中執行的 Amazon EC2 Windows 執行個體。

此小節顯示如何擷取工作負載供 Tuning Advisor 分析。這是擷取工作負載偏好的程序，因為 Amazon RDS 會限制主機對 SQL Server 執行個體的存取。如需詳細資訊，請參閱 Microsoft 文件中的 [Database Engine Tuning Advisor](#)。

若要使用 Tuning Advisor，您必須向建議程式提供名為工作負載的內容。工作負載是一組 Transact-SQL 陳述式，它會對您想要調校的一或多個資料庫執行。在調校資料庫時，Database Engine Tuning Advisor 使用追蹤檔案、追蹤資料表、Transact-SQL 指令碼或 XML 檔案做為工作負載輸入。使用 Amazon RDS 時，工作負載可以是用戶端電腦上的檔案，或可供用戶端電腦存取之 Amazon RDS for SQL Server 資料庫上的資料庫資料表。檔案或資料表必須包含對您要調校之資料庫的查詢 (採用適合重播的格式)。

若要讓 Tuning Advisor 更有效，工作負載應該盡可能實際。您可以透過對資料庫執行個體執行追蹤來產生工作負載檔案或資料表。執行追蹤時，您可以在資料庫執行個體上模擬負載或使用一般負載執行應用程式。

有兩個類型的追蹤：用戶端和伺服器端。您可以輕鬆設定用戶端追蹤，且可以在 SQL Server Profiler 中即時監看擷取的追蹤事件。伺服器端追蹤的設定較複雜，因此需要編寫一些 Transact-SQL 指令碼。此外，因為會將追蹤寫入 Amazon RDS 資料庫執行個體上的檔案，因此追蹤會耗用儲存空間。務必追蹤執行伺服器端追蹤所使用的儲存空間，因為資料庫執行個體可能進入儲存空間滿載的狀態，因此如果儲存空間用完將不再可用。

針對用戶端追蹤，在 SQL Server Profiler 中擷取了足夠數量的追蹤資料時，接著您可以透過將追蹤儲存在您本機電腦上的檔案或可供用戶端電腦存取的資料庫執行個體上的資料庫資料表，藉此產生工作負載檔案。使用用戶端追蹤的主要缺點是該追蹤在繁重負載的情況下，可能不會擷取所有查詢。這可能會削弱 Database Engine Tuning Advisor 所執行分析的有效性。如果需要在繁重負載下執行追蹤，並想要確保它可擷取追蹤工作階段期間的每個查詢，您應該使用伺服器端追蹤。

針對伺服器端追蹤，您必須將資料庫執行個體上的追蹤檔案放入合適的工作負載檔案中，或您可以在追蹤完成之後將追蹤儲存至資料庫執行個體上的資料表。您可以使用 SQL Server Profiler 將追蹤儲存至本機電腦上的檔案，或是從資料庫執行個體上的追蹤資料表讀取 Tuning Advisor。

在 SQL Server 資料庫執行個體上執行用戶端追蹤

在 SQL Server 資料庫執行個體上執行用戶端追蹤

1. 啟動 SQL Server Profiler。它安裝在 SQL Server 執行個體資料夾的 Performance Tools 資料夾。您必須載入或定義追蹤定義範本才能開始用戶端追蹤。
2. 在 SQL Server Profiler File (檔案) 選單中，選擇 New Trace (新增追蹤)。在 Connect to Server (連接至伺服器) 對話方塊中，輸入您想要對其執行追蹤之資料庫的資料庫執行個體端點、連接埠、主要使用者名稱和密碼。
3. 在 Trace Properties (追蹤屬性) 對話方塊中，輸入追蹤名稱並選擇追蹤定義範本。應用程式隨附了預設的範本 TSQL_Replay。您可以編輯此範本來定義追蹤。在 Trace Properties (追蹤屬性) 對話方塊的 Events Selection (事件選取) 標籤下編輯事件和事件資訊。

如需追蹤定義範本和使用 SQL Server Profiler 來指定用戶端追蹤的詳細資訊，請參閱 Microsoft 文件中的 [Database Engine Tuning Advisor](#)。

4. 開始用戶端追蹤，並在對資料庫執行個體執行追蹤時即時監看 SQL 查詢。
5. 完成追蹤時從 File (檔案) 功能表選取 Stop Trace (停止追蹤)。將結果儲存為檔案或儲存為資料庫執行個體上的追蹤資料表。

在 SQL Server 資料庫執行個體上執行伺服器端追蹤

編寫指令碼來建立伺服器端追蹤的操作可能很複雜，並且不在此文件的範圍。此小節包含您可以用作範例的範例指令碼。利用用戶端追蹤時，目標是要建立您可以使用 Database Engine Tuning Advisor 來開啟的工作負載檔案或追蹤資料表。

下列是會開始伺服器端追蹤並將詳細資料擷取至工作負載檔案之縮減的範例指令碼。此追蹤最初是將檔案儲存至 D:\RDSDBDATA\Log 目錄中的 RDSTrace.trc，並且每 100 MB 輪替，因此後續的追蹤檔案名為 RDSTrace_1.trc、RDSTrace_2.trc 等等。

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
    EXEC sp_trace_setevent @traceid, 10, 1, @on
    EXEC sp_trace_setevent @traceid, 10, 2, @on
    EXEC sp_trace_setevent @traceid, 10, 3, @on
    . . .
    EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
    EXEC sp_trace_setstatus @traceid, 1
END
```

下列範例是停止追蹤的指令碼。請注意，先前的指令碼建立的追蹤會繼續執行，直到您明確停止追蹤或該程序用完磁碟空間為止。

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1
```

```
IF @traceid IS NOT NULL BEGIN
    EXEC sp_trace_setstatus @traceid, 0
    EXEC sp_trace_setstatus @traceid, 2
END
```

您可以將伺服器端追蹤結果儲存至資料庫資料表，並透過使用 `fn_trace_gettable` 函數，將資料庫資料表用作 Tuning Advisor 的工作負載。下列命令會將 `D:\rdsdbdata\Log` 目錄中名為 `RDSTrace.trc` 之所有檔案的結果 (包括所有輪替檔案 (像是 `RDSTrace_1.trc`)) 載入目前資料庫中名為 `RDSTrace` 的資料表。

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

若要將特定輪替檔案儲存至資料表，例如 `RDSTrace_1.trc` 檔案，請指定輪替檔案的名稱並將 1 替換，而非使用預設值做為傳遞至 `fn_Trace_gettable` 的最後一個參數。

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

使用追蹤執行 Tuning Advisor

建立追蹤做為本機檔案或資料庫資料表後，接著您可以對資料庫執行個體執行 Tuning Advisor。使用 Tuning Advisor 搭配 Amazon RDS 與使用獨立式、遠端 SQL Server 執行個體時是相同的程序。您可以在用戶端機器上使用 Tuning Advisor UI 或從命令列使用 `dta.exe` 公用程式。在這兩個情況下，使用 Tuning Advisor 時，您都必須使用資料庫執行個體的端點連接至 Amazon RDS 資料庫執行個體，並提供您的主要使用者名稱和主要使用者密碼。

下列程式碼範例示範對具有 `dta.cnazcmklsdei.us-east-1.rds.amazonaws.com` 端點的 Amazon RDS 資料庫執行個體使用 `dta.exe` 命令列公用程式。此範例包括主要使用者名稱 `admin` 和主要使用者密碼 `test`，要調整的範例資料庫命名為名為 `C:\RDSTrace.trc` 的機器。範例命令列程式碼也會指定名為 `RDSTrace1` 的追蹤工作階段，並將名為 `RDSTrace.sql` 之本機機器的輸出檔案表示為 SQL 輸出指令碼，將 `RDSTrace.txt` 表示為結果檔案，以及將 `RDSTrace.xml` 表示為分析的 XML 檔案。在名為 `RDSTraceErrors` 的 `RDSDTA` 資料庫上也指定了一個錯誤資料表。

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RSDTA -
if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
RDSTrace.xml -e RSDTA.dbo.RDSTraceErrors
```


這是相同的範例命令列程式碼，除了輸入工作負載是名為 **RDSTrace** (位於 **RDSDTA** 資料庫) 的遠端 Amazon RDS 執行個體上的資料表。

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it
RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\RDSTrace.sql -or C:\RDSTrace.txt -ox C:\
RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

如需 dta 公用程式命令列參數的完整清單，請參閱 Microsoft 文件中的 [dta 公用程式](#)。

將資料庫的 **db_owner** 變更為 **rdsa** 帳戶

當您在 RDS for SQL Server 資料庫執行個體中建立或還原資料庫時，Amazon RDS 會將資料庫的擁有者設定為 **rdsa**。如果您的多可用區部署使用 SQL Server Database Mirroring (DBM) 或 Always On 可用性群組 (AGs)，則 Amazon RDS 會將次要資料庫執行個體上的資料庫擁有者設定為 **NT AUTHORITY\SYSTEM**。次要資料庫執行個體提升為主要角色之前，次要資料庫的擁有者無法變更。大多數情況下，在執行查詢時，將資料庫擁有者設定為 **NT AUTHORITY\SYSTEM** 並不成問題，但是在執行系統預存程序 (例如，需要有提高的許可才能執行的 `sys.sp_updatestats`) 時，可能會擲回錯誤。

您可以使用下列查詢來識別 **NT AUTHORITY\SYSTEM** 所擁有資料庫的擁有者：

```
SELECT name FROM sys.databases WHERE SUSER_SNAME(owner_sid) = 'NT AUTHORITY\SYSTEM';
```

您可以使用 Amazon RDS 預存程序 `rds_changedbowner_to_rdsa` 將資料庫擁有者變更為 **rdsa**。下列資料庫不允許搭配 `rds_changedbowner_to_rdsa` 使用：`master`、`model`、`msdb`、`rdsadmin`、`rdsadmin_ReportServer`、`rdsadmin_ReportServerTempDB`、`SSISDB`。

若要將資料庫擁有者變更為 **rdsa**，請呼叫 `rds_changedbowner_to_rdsa` 預存程序並提供資料庫的名稱。

Example 使用方式：

```
exec msdb.dbo.rds_changedbowner_to_rdsa 'TestDB1';
```

以下是必要參數：

- `@db_name`：要將其資料庫擁有者變更為 **rdsa** 的資料庫名稱。

Microsoft SQL Server 的定序和字元集

SQL Server 支援在多個層級的定序。您在建立資料庫執行個體時會設定預設伺服器定序。您可以在資料庫、資料表或欄層級覆寫定序。

主題

- [Microsoft SQL Server 的伺服器層級定序](#)
- [Microsoft SQL Server 的資料庫層級定序](#)

Microsoft SQL Server 的伺服器層級定序

在建立 Microsoft SQL Server 資料庫執行個體時，您可以設定要使用的伺服器定序。如果您沒有選擇不同的定序，伺服器層級定序預設為 SQL_Latin1_General_CP1_CI_AS。依預設，會將伺服器定序套用到所有資料庫和資料庫物件。

Note

從資料庫快照還原時，無法變更定序。

Amazon RDS 目前支援以下伺服器定序：

定序	描述
Arabic_CI_AS	阿拉伯文、不區分大小寫、區分重音、不區分假名、不區分寬度
Chinese_PRC_BIN2	Chinese-PRC，二進位代碼點排序順序
Chinese_PRC_CI_AS	Chinese-PRC、不區分大小寫、區分重音、不區分假名、不區分寬度
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke、不區分大小寫、區分重音、不區分假名、不區分寬度
Danish_Norwegian_CI_AS	Danish-Norwegian、不區分大小寫、區分重音、不區分假名、不區分寬度

定序	描述
Finnish_Swedish_CI_AS	芬蘭文、瑞典文和瑞典文 (芬蘭)、不區分大小寫、區分重音、卡納型不區分、不區分假名
French_CI_AS	法文、不區分大小寫、區分重音、不區分假名、不區分寬度
Hebrew_BIN	希伯來文、二進位排序
Hebrew_CI_AS	希伯來文、不區分大小寫、區分重音、不區分假名、不區分寬度
Japanese_BIN	日文、二進位排序
Japanese_CI_AS	日文、不區分大小寫、區分重音、不區分假名、不區分寬度
Japanese_CS_AS	日文、區分大小寫、區分重音、不區分假名、不區分寬度
Japanese_XJIS_140_CI_AS	日文、不區分大小寫、區分重音、不區分假名、不區分寬度、補充字元、不區分變體選擇器
Japanese_XJIS_140_CI_AS_KS_VSS	日文、不區分大小寫、區分重音、區分假名、不區分寬度、補充字元、區分變體選擇器
Japanese_XJIS_140_CI_AS_VSS	日文、不區分大小寫、區分重音、不區分假名、不區分寬度、補充字元、區分變體選擇器
Japanese_XJIS_140_CS_AS_KS_WS	日文、區分大小寫、區分重音、區分假名、區分寬度、補充字元、不區分變體選擇器
Korean_Wansung_CI_AS	Korean-Wansung、不區分大小寫、區分重音、不區分假名、不區分寬度
Latin1_General_100_BIN	Latin1-General-100、二進位排序
Latin1_General_100_BIN2	Latin1-General-100、二進位代碼點排序順序

定序	描述
Latin1_General_100_BIN2_UTF8	Latin1-General-100、二進位代碼點排序順序、UTF-8 編碼
Latin1_General_100_CI_AS	Latin1-General-100、不區分大小寫、區分重音、不區分假名、不區分寬度
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100、不區分大小寫、區分重音、補充字元、UTF-8 編碼
Latin1_General_BIN	Latin1-General、二進位排序
Latin1_General_BIN2	Latin1-General、二進位代碼點排序順序
Latin1_General_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名、不區分寬度
Latin1_General_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度
Latin1_General_CI_AS_KS	Latin1-General、不區分大小寫、區分重音、區分假名、不區分寬度
Latin1_General_CS_AS	Latin1-General、區分大小寫、區分重音、不區分假名、不區分寬度
Modern_Spanish_CI_AS	Modern-Spanish、不區分大小寫、區分重音、不區分假名、不區分寬度
Polish_CI_AS	波蘭文、不區分大小寫、區分重音、不區分假名、不區分寬度
SQL_1xCompat_CP850_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 850 上的 Unicode 資料、SQL Server 排序 49

定序	描述
SQL_Latin1_General_CP1_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 1252 上的 Unicode 資料、SQL Server 排序 54
SQL_Latin1_General_CP1_CI_AS (預設值)	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 1252 上的 Unicode 資料、SQL Server 排序 52
SQL_Latin1_General_CP1_CS_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 1252 上的 Unicode 資料、SQL Server 排序 51
SQL_Latin1_General_CP437_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 437 上的 Unicode 資料、SQL Server 排序 34
SQL_Latin1_General_CP850_BIN	Latin1-General、二進位排序順序 (適用於 Unicode 資料)、字碼頁 850 上的 SQL Server 排序 40 (適用於非 Unicode 資料)
SQL_Latin1_General_CP850_BIN2	Latin1-General、二進位代碼點排序順序，適用於非 Unicode 資料字碼頁 850 上的 Unicode 資料、SQL Server 排序 40
SQL_Latin1_General_CP850_CI_AI	Latin1-General、不區分大小寫、不區分重音、不區分假名、不區分寬度 (適用於 Unicode 資料)、字碼頁 850 上的 SQL Server 排序 44 (適用於非 Unicode 資料)

定序	描述
SQL_Latin1_General_CP850_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 850 上的 Unicode 資料、SQL Server 排序 42
SQL_Latin1_General_CP1256_CI_AS	Latin1-General、不區分大小寫、區分重音、不區分假名、不區分寬度，適用於非 Unicode 資料字碼頁 1256 上的 Unicode 資料、SQL Server 排序 146
Thai_CI_AS	泰語、不區分大小寫、區分重音、不區分假名、不區分寬度
Turkish_CI_AS	土耳其文、不區分大小寫、區分重音、不區分假名、不區分寬度

選擇定序：

- 如果您使用 Amazon RDS 主控台，在建立新的資料庫執行個體時，請選擇 Additional configuration (其他組態)，然後在 Collation (定序) 欄位中輸入定序。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 如果您使用 AWS CLI，請以 `--character-set-name` 命令使用 `create-db-instance` 選項。如需詳細資訊，請參閱 [create-db-instance](#)。
- 如果您使用 Amazon RDS API，請以 CharacterSetName 操作使用 CreateDBInstance 參數。如需詳細資訊，請參閱 [CreateDBInstance](#)。

Microsoft SQL Server 的資料庫層級定序

在建立新的資料庫或資料庫物件時，您可以透過覆寫定序來變更資料庫、資料表或資料欄層級的預設定序。例如，如果您的預設伺服器定序為 SQL_Latin1_General_CP1_CI_AS，您可以將其變更為 Mohawk_100_CI_AS 來支援 Mohawk 定序。如果必要，查詢中的每個引數可以是類型轉換以使用不同的定序。

例如，下列查詢會將 AccountName 資料欄的預設定序變更為 Mohawk_100_CI_AS

```
CREATE TABLE [dbo].[Account]
(
    [AccountID] [nvarchar](10) NOT NULL,
    [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
) ON [PRIMARY];
```

Microsoft SQL Server 資料庫引擎透過內建的 NCHAR、NVARCHAR 和 NTEXT 資料類型來支援 Unicode。例如，如果您需要 CJK 支援，在建立資料庫和資料表時，請對字元儲存使用這些 Unicode 資料類型，並覆寫預設伺服器定序。以下是 Microsoft 提供的數個連結，內容涵蓋 SQL Server 的定序和 Unicode 支援：

- [使用定序](#)
- [定序和國際術語](#)
- [使用 SQL Server 定序](#)
- [資料庫和資料庫引擎應用程式的國際考量](#)

建立資料庫使用者

您可以執行如下列範例所示的 T-SQL 指令碼，為您的 Amazon RDS for Microsoft SQL Server 資料庫執行個體建立資料庫使用者。使用 SQL Server Management Studio (SSMS) 等應用程式。您以建立資料庫執行個體時建立的主要使用者身分登入至資料庫執行個體。

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
```

如需將資料庫使用者新增至角色的範例，請參閱 [將使用者新增至 SQL AgentUser 角色](#)。

Note

若您在新增使用者時出現許可錯誤，則可透過修改資料庫執行個體主要使用者密碼來恢復權限。如需詳細資訊，請參閱 [重新設定 db_owner 角色密碼](#)。

判斷 Microsoft SQL Server Database 的復原模型

在 Amazon RDS 中，復原模型、保留期間和資料庫狀態是相連結的。

更改任何一個設定之前，了解後果是很重要的。每一個設定都會影響其他設定。例如：

- 備份保留已啟用時，如果您將資料庫的復原模型變更為 SIMPLE 或 BULK_LOGGED，Amazon RDS 會在設定變更的五分鐘內將復原模型重設為 FULL。這也將造成 RDS 取得資料庫執行個體的快照。
- 如果您將備份保留設定為 0 天，RDS 會將復原模式設定為 SIMPLE。
- 備份保留設定為 0 天時，如果您將資料庫的復原模型從 SIMPLE 變更為任何其他選項，RDS 會將復原模型重設為 SIMPLE。

Important

即使您似乎可以這麼做 — 例如使用 ALTER DATABASE，仍切勿更改多重可用區執行個體的復原模型。多重可用區必須使用備份保留以及「完整」復原模式。如果您更改了復原模型，RDS 會立即將它改回「完整」。

此一自動重設會強制 RDS 完整地重建鏡像。在此重建期間，資料庫的可用性會降低效能大約 30 到 90 分鐘，直到鏡像準備好進行容錯移轉為止。資料庫執行個體同樣會經歷從單一可用區轉換成多重可用區時效能變差的狀況。效能變差的時間取決於資料庫儲存大小 — 儲存的資料庫越大，變差的時間越久。

如需有關 SQL Server 復原模式的詳細資訊，請參閱 Microsoft 文件中的 [復原模式 \(SQL Server\)](#)。

判斷上次容錯移轉時間

若要判斷上次容錯移轉時間，請使用下列預存程序：

```
execute msdb.dbo.rds_failover_time;
```

此程序會傳回下列資訊。

輸出參數	描述
errorlog_available_from	顯示日誌目錄中可取得錯誤記錄的時間。
recent_failover_time	顯示上次容錯移轉時間 (如果可從錯誤日誌中取得)。否則顯示 null。

Note

預存程序會搜尋日誌目錄中所有可用的 SQL Server 錯誤日誌，以擷取最近的容錯移轉時間。如果 SQL Server 已覆寫容錯移轉訊息，則程序不會擷取容錯移轉時間。

Example 沒有最近的容錯移轉

此範例顯示錯誤日誌中沒有最近的容錯移轉時的輸出。自從 2020-04-29 23:59:00.01 以來，沒有發生任何容錯移轉。

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example 最近的容錯移轉

此範例顯示錯誤日誌中有容錯移轉時的輸出。最近一次的容錯移轉是在 2020-05-05 18:57:51.89。

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

在大量載入期間停用快速插入

從 SQL Server 2016 年開始，預設會啟用快速插入。快速插入會利用資料庫處於簡單或大量記錄復原模式時所發生的最小記錄，以最佳化插入效能。透過快速插入，每個大量載入批次都會取得新的擴充區塊，略過具有可用空間的現有擴充區塊的配置查詢，以最佳化插入效能。

但是，使用快速插入的批次大小的批次負載可能會導致物件消耗的未使用空間增加。如果增加批次大小不可行，啟用追蹤旗標 692 可以協助減少未使用的保留空間，但會犧牲效能。啟用此追蹤旗標會停用快速插入，同時大量載入堆積或叢集索引的資料。

您可以啟用追蹤旗標 692 做為啟動參數使用資料庫參數群組。如需更多詳細資訊，請參閱 [使用參數群組](#)。

SQL Server 2016 和更新版本的 Amazon RDS 支援追蹤旗標 692。如需有關追蹤旗標的詳細資訊，請參閱 Microsoft 說明文件中的 [DBCC TRACEON - 追蹤旗標](#)。

捨棄 Microsoft SQL Server 資料庫

您可以捨棄在單一可用區或多個可用區部署中執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體上的資料庫。若要捨棄資料庫，請使用以下命令：

```
--replace your-database-name with the name of the database you want to drop  
EXECUTE msdb.dbo.rds_drop_database N'your-database-name'
```

Note

在命令中使用直單引號。智慧引號會導致錯誤。

使用此程序捨棄資料庫時，Amazon RDS 會卸除對資料庫的所有現有連線，並移除資料庫的備份歷史記錄。

重新命名多個可用區部署中的 Microsoft SQL Server 資料庫

若要重新命名使用多個可用區域的 Microsoft SQL Server 資料庫執行個體，請使用下列程序：

1. 首先，關閉資料庫執行個體的多個可用區域。
2. 執行 `rdsadmin.dbo.rds_modify_db_name` 以重新命名資料庫。
3. 然後，開啟資料庫執行個體的多個可用區域鏡像或 Always On 可用性群組，使其返回原始狀態。

如需更多詳細資訊，請參閱 [將多可用區域新增至 Microsoft SQL Server 資料庫執行個體](#)。

Note

如果您的執行個體未使用多個可用區，執行 `rdsadmin.dbo.rds_modify_db_name` 之前或之後，不需要變更任何設定。

範例：在下列範例中，`rdsadmin.dbo.rds_modify_db_name` 預存程序會將資料庫從 **MOO** 重新命名為 **ZAR**。這類似於執行陳述式 DDL `ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`。

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'  
GO
```

重新設定 `db_owner` 角色密碼

如果您本身無法使用 Microsoft SQL Server 資料庫的 `db_owner` 角色，您可以透過修改資料庫執行個體主要密碼來重設 `db_owner` 角色的密碼。透過變更資料庫執行個體主要密碼，您可以重新取得資料庫執行個體的存取權，使用 `db_owner` 修改的密碼來存取資料庫，以及還原可能意外撤銷的 `db_owner` 角色的權限。若要變更資料庫執行個體的密碼，則可使用 Amazon RDS 主控台、AWS CLI 命令 [modify-db-instance](#)，或是使用 [ModifyDBInstance](#) 操作。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

還原授權終止的資料庫執行個體

Microsoft 已要求未回報 Microsoft 授權行動性協議資訊的 Amazon RDS 一些客戶終止其資料庫執行個體。Amazon RDS 會取得這些資料庫執行個體的快照，而您可以從快照還原為具有已包含授權模型的新資料庫執行個體。

您可以從 Standard Edition 的快照還原至 Standard Edition 或 Enterprise Edition。

您可以從 Enterprise Edition 的快照還原至 Standard Edition 或 Enterprise Edition。

在 Amazon RDS 建立執行個體的最終快照後從 SQL Server 快照還原

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇您的 SQL Server 資料庫執行個體的快照。Amazon RDS 會為您建立資料庫執行個體的最終快照。終止之執行個體快照的名稱採用格式 `instance_name-final-snapshot`。例如，如果

您的資料庫執行個體名為 `mytest.cdxcgahslksma.us-east-1.rds.com`，最終快照即名為 `mytest-final-snapshot`，並且會與原始的資料庫執行個體位於相同 AWS 區域。

- 針對 Actions (動作)，選擇 Restore Snapshot (還原快照)。

Restore DB Instance (還原資料庫執行個體) 視窗隨即出現。

- 針對 License Model (授權模型)，選擇 license-included。
- 選擇您要使用的 SQL Server 資料庫引擎。
- 對於 DB Instance Identifier (資料庫執行個體識別符)，輸入還原資料庫執行個體的名稱。
- 選擇 Restore DB Instance (還原資料庫執行個體)。

如需從快照還原的詳細資訊，請參閱[從資料庫快照還原](#)。

將 Microsoft SQL Server 資料庫從「離線」轉移至「上線」

您可以將 Amazon RDS 資料庫執行個體上的 Microsoft SQL Server 資料庫從 OFFLINE 轉移至 ONLINE。

SQL Server 方法	Amazon RDS 方法
<code>ALTER DATABASE <i>db_name</i> SET ONLINE;</code>	<code>EXEC rdsadmin.dbo.rds_set_database_online <i>db_name</i></code>

使用變更資料擷取

Amazon RDS 支援執行 Microsoft SQL Server 之資料庫執行個體的變更資料擷取支援 (CDC)。CDC 會擷取對資料表中之資料所做的變更。它會儲存與每個變更相關的中繼資料，而您稍候可存取這些中繼資料。如需 CDC 運作方式的詳細資訊，請參閱 Microsoft 文件中的[變更資料擷取](#)。

使用 CDC 搭配您的 Amazon RDS 資料庫執行個體之前，請透過執行 `msdb.dbo.rds_cdc_enable_db` 以在資料庫中啟用它。您必須有主要使用者權限才能啟用 Amazon RDS 資料庫執行個體中的 CDC。啟用 CDC 之後，該資料庫 `db_owner` 的任何使用者都可以在該資料庫的資料表上啟用或停用 CDC。

⚠ Important

還原期間，將停用 CDC。所有相關中繼資料會自動從資料庫移除。這適用於從 S3 的快照還原、point-in-time 還原和 SQL Server 原生還原。執行其中一個還原類型後，您就可以重新啟用 CDC 和重新指定要追蹤的資料表。

若要啟用資料庫執行個體的 CDC，請執行 `msdb.dbo.rds_cdc_enable_db` 預存程序。

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

若要停用資料庫執行個體的 CDC，請執行 `msdb.dbo.rds_cdc_disable_db` 預存程序。

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

主題

- [使用變更資料擷取來追蹤資料表](#)
- [變更資料擷取任務](#)
- [變更多個可用區執行個體的資料擷取](#)

使用變更資料擷取來追蹤資料表

在資料庫上啟用 CDC 之後，您就可以開始追蹤特定資料表。您可以執行 [sys.sp_cdc_enable_table](#) 來選擇要追蹤的資料表。

```
--Begin tracking a table
exec sys.sp_cdc_enable_table
    @source_schema          = N'source_schema'
    , @source_name          = N'source_name'
    , @role_name            = N'role_name'

--The following parameters are optional:

--, @capture_instance      = 'capture_instance'
--, @supports_net_changes  = supports_net_changes
--, @index_name            = 'index_name'
--, @captured_column_list  = 'captured_column_list'
--, @filegroup_name        = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
```

```
;
```

若要檢視資料表的 CDC 組態，請執行 [sys.sp_cdc_help_change_data_capture](#)。

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
-- 'schema_name', 'table_name'
;
```

如需 SQL Server 文件中 CDC 資料表、函數和預存程序的詳細資訊，請參閱下列：

- [變更資料擷取預存程序 \(Transact-SQL\)](#)
- [變更資料擷取函數 \(Transact-SQL\)](#)
- [變更資料擷取資料表 \(Transact-SQL\)](#)

變更資料擷取任務

啟用 CDC 時，SQL Server 會建立 CDC 任務。資料庫擁有者 (db_owner) 可以檢視、建立、修改和刪除 CDC 任務。不過，RDS 系統帳戶擁有這些任務。因此，任務不會在原生檢視、程序或是在 SQL Server Management Studio 中顯示。

若要控制資料庫中 CDC 的行為，請使用原生 SQL Server 程序，例如 [sp_cdc_enable_table](#) 和 [sp_cdc_start_job](#)。若要變更 CDC 任務參數，例如 maxtrans 和 maxscans，您可以使用 [sp_cdc_change_job](#)。

若要取得有關 CDC 任務的更多資訊，您可以查詢下列動態管理檢視：

- sys.dm_cdc_errors
- sys.dm_cdc_log_scan_sessions
- sysjobs
- sysjobhistory

變更多個可用區執行個體的資料擷取

如果您在多重可用區執行個體上使用 CDC，請確定鏡像的 CDC 任務組態符合委託人上的組態。CDC 任務會映射至 database_id。如果次要伺服器上的資料庫 ID 與委託人不同，那麼，任務將不會與正

確的資料庫關聯。為了嘗試在容錯移轉之後避免錯誤，RDS 會在新委託人上放置並重新建立任務。重新建立的任務會使用委託人在容錯移轉之前記錄的參數。

儘管此程序的執行速度很快，但在 RDS 可更正 CDC 任務前這些任務可能仍會執行。以下是強制讓主要和次要複本之間的參數一致的三個方式：

- 對已啟用 CDC 的所有資料庫使用相同任務參數。
- 變更 CDC 任務組態之前，將多重可用區執行個體轉換為單一可用區。
- 每當您在委託人上變更參數時，請手動傳輸參數。

容錯移轉之後，若要檢視和定義用來重新建立 CDC 任務的 CDC 參數，請使用 `rds_show_configuration` 和 `rds_set_configuration`。

下列範例會傳回 `cdc_capture_maxtrans` 的數值：針對設為 `RDS_DEFAULT` 的任何參數，RDS 會自動設定該數值。

```
-- Show configuration for each parameter on either primary and secondary replicas.  
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans';
```

若要設定次要伺服器上的組態，請執行 `rdsadmin.dbo.rds_set_configuration`。此程序會設定次要伺服器上所有資料庫的參數值。只有在容錯移轉之後才會使用這些設定。下列範例會將所有 CDC 擷取任務的 `maxtrans` 設定為 `1000`：

```
--To set values on secondary. These are used after failover.  
exec rdsadmin.dbo.rds_set_configuration 'cdc_capture_maxtrans', 1000;
```

若要設定委託人上的 CDC 任務參數，請改為使用 [sys.sp_cdc_change_job](#)。

使用 SQL Server 代理程式

借助 Amazon RDS，您可以在執行 Microsoft SQL Server 企業版、標準版或 Web 版的資料庫執行個體上使用 SQL Server 代理程式。SQL Server 代理程式是 Microsoft Windows 服務，會執行排程的管理任務 (稱為任務)。您可以使用 SQL Server 代理程式來執行 T-SQL 任務，以在 SQL Server 資料庫執行個體中重建索引、執行損毀檢查，以及彙總資料。

建立 SQL Server 資料庫執行個體時，會在 `SQLAgentUserRole` 角色中登錄主要使用者。

SQL Server 代理程式可以安排或隨需執行任務，以回應特定事件。如需詳細資訊，請參閱 Microsoft 文件中的 [SQL Server 代理程式](#)。

Note

避免排程任務在資料庫執行個體的維護和備份時段執行。由啟動的維護和備份程序 AWS 可能會中斷工作或導致工作被取消。

在異地同步備份部署中，當任務複寫功能開啟時，SQL Server Agent 任務從主要主機複製至次要主機。如需詳細資訊，請參閱 [開啟 SQL Server Agent 任務複寫](#)。

多可用區部署有 10,000 個 SQL Server Agent 任務的限制。如果您需要更高的限制，請聯繫以申請增加 AWS Support。開啟 [AWS Support 中心](#) 頁面，如有必要請登入，然後選擇 Create case (建立案例)。選擇 Service limit increase (提高服務限制)。填妥並提交表格。

若要在 SQL Server Management Studio (SSMS) 中檢視個別 SQL Server 代理程式任務的歷程紀錄，您可以開啟物件總管 (Object Explorer)，並在任務上按一下滑鼠右鍵，然後選擇 View History (檢視歷程紀錄)。

由於 SQL Server 代理程式正在資料庫執行個體中的受管理主機上執行，因此不支援某些動作：

- 不支援使用 ActiveX、Windows 命令介面或視窗來執行複寫工作和執行命令列指令碼。PowerShell
- 您無法手動開始、停止或重新啟動 SQL Server 代理程式。
- 無法從資料庫執行個體透過 SQL Server 代理程式傳送電子郵件通知。
- 不支援 SQL Server 代理程式提醒和運算子。
- 不支援使用 SQL Server 代理程式來建立備份。使用 Amazon RDS 備份您的資料庫執行個體。
- 目前，適用於 SQL 伺服器的 RDS 不支援使用 SQL 伺服器代理程式權杖。

開啟 SQL Server Agent 任務複寫

您可使用下列儲存程序開啟 SQL Server Agent 任務複寫：

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'SQLAgentJob';
```

您可在 Amazon RDS for SQL Server 支援的所有 SQL Server 版本上執行儲存程序。複製下列類別中的任務：

- [未分類 (本機)]
- [未分類 (多伺服器)]
- [未分類]

- 資料收集器
- 資料庫引擎調校 Advisor
- 資料庫維護
- 全文

僅複製使用 T-SQL 任務步驟的任務。具有 SQL 伺服器整合服務 (SSIS)、SQL 伺服器報告服務 (SSRS)、複寫等步驟類型的作業 PowerShell 不會複寫。不會複製使用 Database Mail 和伺服器層級物件的任務。

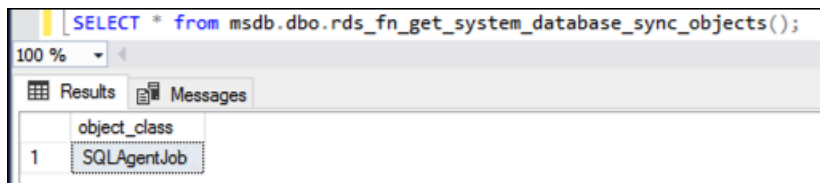
⚠ Important

主要主機是複寫的真實來源。開啟工作複寫之前，請確定您的 SQL Server 代理程式工作位於主要主機上。如果您未這麼做，則當次要主機上有較新的工作時，如果您開啟該功能，可能會導致刪除 SQL Server 代理程式工作。

您可使用下列函數來確認複製是否已開啟。

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

若 SQL Server Agent 任務正在複製，則 T-SQL 查詢傳回下列內容。若並未複製，則不會為 `object_class` 傳回任何東西。



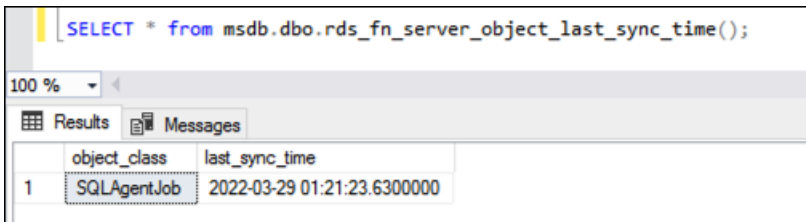
object_class
1 SQLAgentJob

您可以使用下列函數，以 UTC 時間找出物件上次同步的時間。

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

例如，假設您修改 SQL Server Agent 任務的時間為 01:00。您希望最近的同步時間在 01:00 之後，表示同步已發生。

同步化之後，針對次要節點上的 `date_created` 和 `date_modified` 傳回的值預期會相符。



	object_class	last_sync_time
1	SQLAgentJob	2022-03-29 01:21:23.6300000

如果您也使用tempdb複寫，您可以在@object_type參數中提供 SQL 代理程式工作和tempdb組態來啟用複寫：

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =  
'SQLAgentJob,TempDbFile';
```

如需tempdb複製的詳細資訊，請參閱[異地同步備份部署的 TempDB 組態](#)。

將使用者新增至 SQL AgentUser 角色

如要允許其他登入或使用者使用 SQL Server Agent，則您必須以主要使用者身分登入，並進行下列作業：

1. 透過使用 CREATE LOGIN 命令來建立另一個伺服器層級的登入。
2. 利用 msdb 命令在 CREATE USER 中建立使用者，然後將此使用者與您在前一個步驟中所建立的登入相連結。
3. 利用 SQLAgentUserRole 系統存放程序，將使用者新增至 sp_addrolemember。

例如，假設您的主要使用者名稱為 **admin**，而您想要將 SQL Server Agent 的存取權提供給名為 **theirname** 與密碼為 **theirpassword** 的使用者。於此狀況下，您可使用下列程序。

若要將使用者新增至 SQL AgentUser 角色

1. 以主要使用者身分登入。
2. 執行下列命令：

```
--Initially set context to master database  
USE [master];  
GO  
--Create a server-level login named theirname with password theirpassword  
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';  
GO
```



```
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login
theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

刪除 SQL Server 代理程式任務

您可以使用 `sp_delete_job` 存放程序來刪除 Amazon RDS for Microsoft SQL Server 上的 SQL Server 代理程式任務。

您無法使用 SSMS 來刪除 SQL Server 代理程式任務。如果這麼做，您會收到類似下列的錯誤訊息：

```
The EXECUTE permission was denied on the object 'xp_regread', database
'mssqlsystemresource', schema 'sys'.
```

RDS 屬於受管服務，其會因受限而無法執行能存取 Windows 登錄的程序。當您使用 SSMS 時，其會嘗試執行 RDS 未授權的程序 (`xp_regread`)。

Note

在 RDS for SQL Server 上，才允許 `sysadmin` 角色的成員更新或刪除不同登入所擁有的任務。

若要刪除 SQL Server 代理程式任務

- 執行下列 T-SQL 陳述式：

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

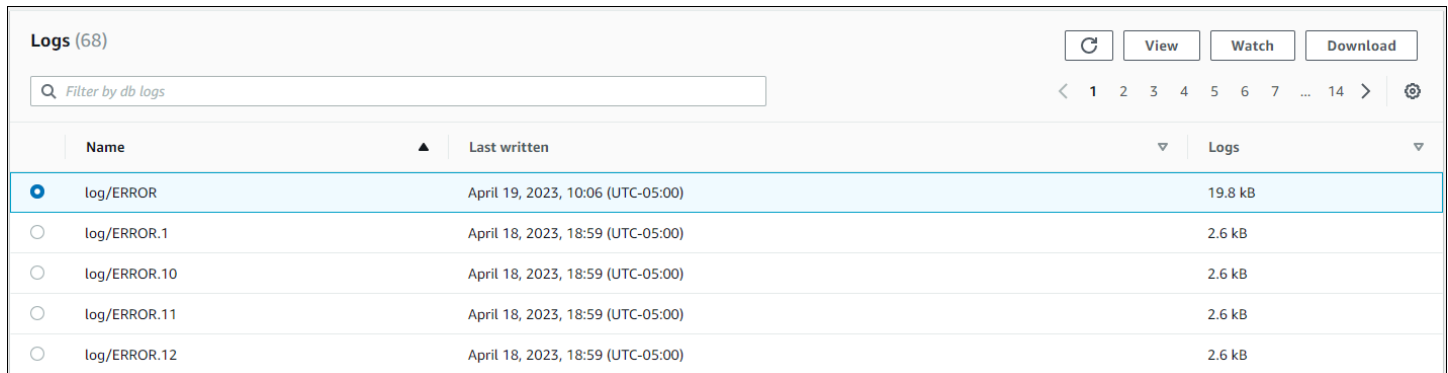
使用 Microsoft SQL Server 日誌

您可以使用 Amazon RDS 主控台來檢視、監看和下載 SQL Server Agent 日誌和 Microsoft SQL Server 錯誤日誌和 SQL Server Reporting Services (SSRS) 日誌。

監看日誌檔案

如果您檢視 Amazon RDS 主控台日的日誌，您可以查看當下日誌的內容。在主控台中監看日誌會以動態狀態開啟它，使得您可以近乎即時查看它的更新。

只有最新的日誌可供監看。例如，假設您有如下所示的日誌：



Name	Last written	Logs
<input checked="" type="radio"/> log/ERROR	April 19, 2023, 10:06 (UTC-05:00)	19.8 kB
<input type="radio"/> log/ERROR.1	April 18, 2023, 18:59 (UTC-05:00)	2.6 kB
<input type="radio"/> log/ERROR.10	April 18, 2023, 18:59 (UTC-05:00)	2.6 kB
<input type="radio"/> log/ERROR.11	April 18, 2023, 18:59 (UTC-05:00)	2.6 kB
<input type="radio"/> log/ERROR.12	April 18, 2023, 18:59 (UTC-05:00)	2.6 kB

只會主動更新做為最新日誌的 log/ERROR。您可以選擇監看其他項目，但它們是靜態並且不會更新。

封存日誌檔案

Amazon RDS 主控台會顯示上一週到當日的日誌。您可以下載和將日誌封存，以讓它們在該時間後可供參考。將日誌封存的一個方式是將它們載入 Amazon S3 儲存貯體。如需如何設定 Amazon S3 儲存貯體和上傳檔案的指示，請參閱《Amazon Simple Storage Service 入門指南》中的 [Amazon S3 基本概念](#)，並按一下 Get Started (入門)。

檢視錯誤和代理程式日誌

若要檢視 Microsoft SQL 伺服器錯誤和代理程式日誌，請使用 Amazon RDS 預存程序 `rds_read_error_log` 搭配下列參數：

- **@index**—要擷取的日誌版本。預設值為 0，它會擷取最新的錯誤日誌。指定 1 可擷取先前的日誌，指定 2 可擷取更早的日誌，依此類推。
- **@type**—要擷取的日誌類型。指定 1 可擷取錯誤日誌。指定 2 可擷取代理程式日誌。

Example

下列範例會要求最新的錯誤日誌。

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

如需 SQL Server 錯誤的詳細資訊，請參閱 Microsoft 文件中的[資料庫引擎錯誤](#)。

使用追蹤和傾印檔案

此小節說明對執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體使用追蹤檔案和傾印檔案。

產生追蹤 SQL 查詢

```
declare @rc int
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output, 0, N'D:\rdsdbdata\log\rdstest',
    @maxfilesize, NULL
```

檢視開啟的追蹤

```
select * from ::fn_trace_getinfo(default)
```

檢視追蹤內容

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

設定追蹤和傾印檔案的保留期間

追蹤和傾印檔案可能會累積和耗用磁碟空間。依預設，Amazon RDS 會清除超過七天的追蹤和傾印檔案。

若要檢視最新的追蹤和傾印檔案保留期間，請使用 `rds_show_configuration` 程序，如以下範例所示。

```
exec rdsadmin..rds_show_configuration;
```

若要修改追蹤檔案的保留期間，請使用 `rds_set_configuration` 程序，並設定 `tracefile retention` (分鐘)。下列範例會將追蹤檔案的保留期間設定為 24 小時。

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

若要修改傾印檔案的保留期間，請使用 `rds_set_configuration` 程序，並設定 `dumpfile retention` (分鐘)。下列範例會將傾印檔案的保留期間設定為 3 天。

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

基於安全原因，您無法刪除 SQL Server 資料庫執行個體上的特定追蹤或傾印檔案。若要刪除所有未使用的追蹤或傾印檔案，請將檔案的保留期間設定為 0。

Amazon RDS for MySQL

Amazon RDS 會支援執行下列版本 MySQL 的資料庫執行個體：

- MySQL 8.0
- MySQL 5.7

如需次要版本的支援詳情，請參閱[MySQL on Amazon RDS 版本](#)。

若要建立 Amazon RDS for MySQL 資料庫執行個體，請使用 Amazon RDS 管理工具或介面。然後，您可以執行下列作業：

- 調整資料庫執行個體的大小
- 授權連線至資料庫執行個體
- 從備份或快照建立和還原
- 建立多可用區域次要項
- 建立僅供讀取複本
- 監控資料庫執行個體的效能

若要在資料庫執行個體中存放與存取資料，您可以使用標準 MySQL 公用程式與應用程式。

Amazon RDS for MySQL 符合多項業界標準。例如，您可以使用 RDS for MySQL 資料庫來建置符合 HIPAA 規範的應用程式。依據與 AWS 簽署的商業夥伴協議 (BAA)，您可以使用 RDS for MySQL 資料庫存放醫療保健相關的資訊，包括受保護醫療資訊 (PHI)。Amazon RDS for MySQL 亦符合美國聯邦風險與授權管理計劃 (FedRAMP) 的安全性要求。此外，Amazon RDS for MySQL 已獲得 FedRAMP 聯合授權委員會 (JAB) 核發的 FedRAMP 高基準臨時操作授權書 (P-ATO)，可在 AWS GovCloud (US) 區域內執行。如需支援的合規標準的詳細資訊，請參閱 [AWS 雲端合規](#)。

如需關於 MySQL 各版本功能的資訊，請參閱 MySQL 文件中的 [MySQL 的主要功能](#)。

建立資料庫執行個體之前，請先完成 [設定 Amazon RDS](#) 中的步驟。建立資料庫執行個體時，RDS 主要使用者帳戶會取得 DBA 權限 (有一些限制)。將此帳戶用於管理任務，例如建立其他資料庫帳戶。

您可以建立下列作業：

- 資料庫執行個體
- 資料庫快照

- Point-in-time 還原
- 自動備份
- 手動備份

您可以在以 Amazon 虛擬私有雲端 (VPC) 為基礎的 VPC 內使用執行 MySQL 的資料庫執行個體。您也可以透過開啟各種選項，新增功能至 MySQL 資料庫執行個體。Amazon RDS 支援 MySQL 的多可用區部署，做為高可用性、容錯移轉解決方案。

Important

為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。而且會將存取權限制在某些需要進階權限的系統程序和資料表。您可以使用標準 SQL 用戶端 (例如 mysql 用戶端) 存取資料庫。不過，您無法使用 Telnet 或 Secure Shell (SSH) 直接存取主機。

主題

- [Amazon RDS 上的 MySQL 功能支援](#)
- [MySQL on Amazon RDS 版本](#)
- [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)
- [保護 MySQL 資料庫執行個體連線的安全](#)
- [使用 Amazon RDS Optimized Reads 改善 RDS for MySQL 的查詢效能](#)
- [使用 RDS Optimized Writes 改善 MySQL 的寫入效能](#)
- [升級 MySQL 資料庫引擎](#)
- [升級 MySQL 資料庫快照集引擎版本](#)
- [將資料匯入 MySQL 資料庫執行個體](#)
- [在 Amazon RDS 中使用 MySQL 複寫](#)
- [設定適用於 MySQL 的 RDS 的主動-主動式叢集](#)
- [使用複寫從 MySQL 資料庫執行個體匯出資料](#)
- [MySQL 資料庫執行個體的選項](#)
- [MySQL 的參數](#)
- [MySQL 資料庫執行個體的常用 DBA 任務](#)
- [MySQL 資料庫執行個體的本機時區](#)
- [Amazon RDS for MySQL 的已知問題和限制](#)

- [RDS for MySQL 預存程序參考](#)

Amazon RDS 上的 MySQL 功能支援

RDS for MySQL 支援 MySQL 的大多數特性與功能。部分功能的支援或權限可能有所限制。

您可以在[資料庫的最新資訊](#)頁面上篩選新的 Amazon RDS 功能。對於 Products (產品)，請選擇 Amazon RDS。然後使用關鍵字搜尋，例如 **MySQL 2022**。

Note

以下清單並非詳盡清單。

主題

- [支援的 RDS for MySQL 儲存引擎](#)
- [Amazon RDS 上的 MySQL 搭配使用 Memcached 與其他選項](#)
- [Amazon RDS 上 MySQL 的 InnoDB 快取預備](#)
- [Amazon RDS 不支援的 MySQL 功能](#)

支援的 RDS for MySQL 儲存引擎

雖然 MySQL 支援多種功能不盡相同的儲存引擎，但並非所有引擎的復原能力和資料耐用性都經過最佳化設計。Amazon RDS 可完整支援適用於 MySQL 資料庫執行個體的 InnoDB 儲存引擎。時間點還原和快照還原等 Amazon RDS 功能皆需搭配可復原的儲存引擎才能執行，且僅支援 InnoDB 儲存引擎。如需詳細資訊，請參閱 [MySQL memcached 支援](#)。

聯合儲存引擎目前不支援 Amazon RDS for MySQL。

若結構描述是由使用者建立，則 MyISAM 儲存引擎不支援可靠的復原功能，且還原作業結束後重新啟動 MySQL 時，可能發生資料遺失或毀損等問題，進而導致時間點還原或快照還原功能無法如預期般順利運作。然而，若您仍選擇將 MyISAM 與 Amazon RDS 搭配使用，在某些情況下，快照仍可派上用場。

Note

mysql 結構描述中的系統資料表可能位於 MyISAM 儲存引擎中。

如果將現有的 MyISAM 資料表轉換為 InnoDB 資料表，您可以使用 ALTER TABLE 命令 (例如，alter table TABLE_NAME engine=innodb;)。請記得，MyISAM 和 InnoDB 各有優劣之處，以應用程式執行轉換作業之前，應先完整評估相關影響。

Amazon RDS 不再支援 MySQL 5.1、5.5 和 5.6。但您仍可使用現有的 MySQL 5.1、5.5 和 5.6 快照進行還原作業。當您還原 MySQL 5.1、5.5 或 5.6 快照時，資料庫執行個體會自動升級為 MySQL 5.7。

Amazon RDS 上的 MySQL 搭配使用 Memcached 與其他選項

大部分的 Amazon RDS 資料庫引擎均支援選項群組，讓您可為資料庫執行個體選擇其他功能。RDS for MySQL 資料庫執行個體支援 memcached 選項，這是一種簡單的金鑰型快取。如需 memcached 和其他選項的詳細資訊，請參閱[MySQL 資料庫執行個體的選項](#)。如需使用選項群組的相關詳細資訊，請參閱[使用選項群組](#)。

Amazon RDS 上 MySQL 的 InnoDB 快取預備

InnoDB 快取預備功能可在資料庫執行個體關閉時，儲存緩衝集區的當前狀態，等到資料庫執行個體啟動時，再使用儲存的資訊重新載入緩衝集區，藉此提升 MySQL 資料庫執行個體的效能。如此一來，系統不必從正常的資料庫使用方式開始為緩衝集區「熱機」，而是透過已知常用查詢的頁面預先載入緩衝集區。儲存緩衝集區資訊的檔案只會存放緩衝集區中的頁面中繼資料，而非頁面本身。因此，該檔案不需要太多儲存空間。檔案大小約為快取大小的 0.2%。例如，若為 64 GiB 的快取，則快取預備檔案大小僅會有 128 MiB。如需使用 InnoDB 快取預備功能的詳細資訊，請參閱 MySQL 文件中的[儲存和還原緩衝集區狀態](#)。

RDS for MySQL 資料庫執行個體支援 InnoDB 快取預備。若要啟用 InnoDB 快取預備功能，請在資料庫執行個體的參數群組中，將 innodb_buffer_pool_dump_at_shutdown 和 innodb_buffer_pool_load_at_startup 參數設為 1。變更參數群組中的參數值會影響使用該參數群組的所有 MySQL 資料庫執行個體。若要為特定 MySQL 資料庫執行個體啟用 InnoDB 快取預備功能，您必須為該執行個體另外建立新的參數群組。如需參數群組的相關資訊，請參閱[使用參數群組](#)。

InnoDB 快取預備功能主要可為使用標準儲存體的資料庫執行個體提供效能方面的好處。若您使用的是 PIOPS 儲存體，一般來說不會察覺到顯著的效能優勢。

Important

如果您的 MySQL 資料庫執行個體沒有正常關閉 (例如容錯移轉期間)，則緩衝集區狀態不會儲存到磁碟中。這種情形下，MySQL 會在資料庫執行個體重新啟動時，載入任何可用的緩衝集區。這不會造成任何損害，只是還原的緩衝集區無法反映緩衝集區在重新啟動之前的最新狀

態。為確保您有最新狀態的緩衝集區可在啟動時預備 InnoDB 快取，建議您定期「隨需」傾印緩衝集區。

您可建立事件，以定期自動傾印緩衝集區。舉例來說，下列陳述式會建立名為 `periodic_buffer_pool_dump` 的事件，每小時傾印緩衝集區一次。

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

如需 MySQL 事件的詳細資訊，請參閱 MySQL 文件中的[事件語法](#)。

隨需傾印與載入緩衝集區


您可「隨需」儲存和載入 InnoDB 快取。

- 若要將緩衝集區的最新狀態傾印至磁碟，請呼叫 [mysql.rds_innodb_buffer_pool_dump_now](#) 預存程序。
- 若要從磁碟載入所儲存的緩衝集區狀態，請呼叫 [mysql.rds_innodb_buffer_pool_load_now](#) 預存程序。
- 若要取消進行中的載入操作，請呼叫 [mysql.rds_innodb_buffer_pool_load_abort](#) 預存程序。

Amazon RDS 不支援的 MySQL 功能

Amazon RDS 目前不支援下列 MySQL 功能：

- 身分驗證外掛程式
- 登入系統日誌時發生錯誤
- InnoDB 資料表空間加密
- 密碼強度外掛程式
- 持續的系統變數
- 重寫器查詢重寫外掛程式
- 半同步複寫
- 可傳輸的資料表空間
- X 外掛程式

 Note

所有 RDS for MySQL 5.7.版，以及 MySQL 8.0.26 和更新的 8.0 版都支援全域交易 ID。

為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。而且會將存取權限制在某些需要進階權限的系統程序和資料表。Amazon RDS 可支援使用任何標準 SQL 用戶端應用程式存取資料庫執行個體上的資料庫。Amazon RDS 不允許使用者利用 Telnet、安全殼層 (SSH) 或 Windows 遠端桌面連線，直接託管資料庫執行個體的存取權。當您建立資料庫執行個體時，系統會將您指派為該執行個體上所有資料庫的 db_owner，而且您擁有所有資料庫層級權限，但用於備份的權限除外。Amazon RDS 會為您管理備份。

MySQL on Amazon RDS 版本

若為 MySQL，版本編號組織為版本 = X.Y.Z。在 Amazon RDS 術語中，X.Y 代表主要版本，Z 則代表次要版本編號。進行 Amazon RDS 實作時，若主要版本編號有變化 — 例如從 5.7 版更新至 8.0 版，即屬於主要版本變更。如果只有次要版本號碼發生變更，則版本變更會被視為次要變更 (例如，從版本 8.0.32 到 8.0.34)。

主題

- [Amazon RDS 上支援的 MySQL 次要版本](#)
- [Amazon RDS 上支援的 MySQL 主要版本](#)
- [適用於 MySQL 版 RDS 的 Amazon RDS 擴展 Support 版本](#)
- [使用資料庫預覽環境](#)
- [資料庫預覽環境中的 MySQL 8.3 版](#)
- [資料庫預覽環境中的 MySQL 版本 8.2](#)
- [資料庫預覽環境中的 MySQL 8.1 版](#)
- [Amazon RDS for MySQL 的已取代版本](#)

Amazon RDS 上支援的 MySQL 次要版本

Amazon RDS 目前支援下列 MySQL 次要版本。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。
Amazon RDS 擴展 Support 不適用於次要版本。

MySQL 引擎版本	社群發佈日期	RDS 發佈日期	RDS 標準支援結束日期
8.0			
8.0.37	2024 年四月三十日	2024 年 6 月 18 日	二〇二五年
8.0.36	2024 年 1 月 16 日	2024 年二月十二日	2025 年 3 月

MySQL 引擎版本	社群發佈日期	RDS 發佈日期	RDS 標準支援結束日期
8.0.35	2023 年 10 月 25 日	2023 年 11 月 9 日	2025 年 3 月
8.0.34	2023 年 7 月 18 日	2023 年 8 月 9 日	2024 年 9 月
8.0.33	2023 年 4 月 18 日	2023 年 6 月 15 日	2024 年 9 月
8.0.32	2023 年 1 月 17 日	2023 年 2 月 7 日	2024 年 9 月
5.7			
5.7.44*	2023 年 10 月 25 日	2023 年 11 月 2 日	2024 年 2 月 29 日

* 當主要版本在 Amazon RDS 擴展 Support 中使用時，此次要版本將繼續可用。如需詳細資訊，請參閱 [使用 Amazon RDS 延長支援](#)。

次要版本可以在主要版本之前達到標準支持的終止。例如，次要版本 8.0.28 已於 2024 年 3 月 28 日達到標準支援日期結束，而主要版本 8.0 將於 2026 年 7 月 31 日到達此日期。RDS 將支援 MySQL 社群在這些日期之間發行的其他 8.0.* 次要版本。

建立新的資料庫執行個體時，您可以指定使用目前支援的任一 MySQL 版本。您可以指定主要版本 (如 MySQL 5.7)，並指定使用主要版本所支援的任一次要版本。若未指定版本，Amazon RDS 會預設使用支援的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 會預設使用您指定主要版本的最新次要版本。若要查看支援的版本清單以及新建立的資料庫執行個體的預設值，請使用 [describe-db-engine-versions](#) AWS CLI 指令。

例如，若要列出支援 RDS for MySQL 的引擎版本，請執行下列 CLI 命令：

```
aws rds describe-db-engine-versions --engine mysql --query "*[].[
{Engine:Engine,EngineVersion:EngineVersion}]" --output text
```

預設的 MySQL 版本可能會隨著 AWS 區域而有所不同。若要建立具有特定次要版本的資料庫執行個體，請在建立資料庫執行個體期間指定次要版本。您可以 AWS 區域 使用以下 AWS CLI 命令確定的默認次要版本：

```
aws rds describe-db-engine-versions --default-only --engine mysql
--engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

將 *major-engine-version* 取代為主要引擎版本，並將 *region* 取代為 AWS 區域。例如，下列 AWS CLI 命令會傳回 5.7 主要版本和美國西部 (奧勒岡) AWS 區域 (美國西部 -2) 的預設 MySQL 次要引擎版本：

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7
--region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output
text
```

透過 Amazon RDS，即可控制將 MySQL 執行個體升級至新主要版本的時機，而該版本受 Amazon RDS 支援。您可以維持特定 MySQL 版本的相容性，在部署至生產環境前，先利用您的應用程式測試新版本，並在合適的時間執行主要版本升級。

啟用自動化次要版本升級時，資料庫執行個體將自動升級為新的 MySQL 次要版本，因為它們受到 Amazon RDS 支援。此修補會在您排程的維護時段發生。您可以修改資料庫執行個體，以啟用或停用自動次要版本升級。

若您選擇不要自動排程升級，可以按照更新主要版本的相同程序，手動升級至支援的次要版本。如需相關資訊，請參閱「[升級資料庫執行個體引擎版本](#)」。

Amazon RDS 目前支援將主要版本從 MySQL 5.6 版升級至 5.7 版，以及從 MySQL 5.7 版升級至 8.0 版。主要版本升級可能牽涉相容性風險，因此系統不會自動執行升級，您必須提出修改資料庫執行個體的請求才行。升級生產用途的執行個體前，應先完整測試升級版本。如需升級 MySQL 資料庫執行個體的相關資訊，請參閱[升級 MySQL 資料庫引擎](#)。

您可以建立現有資料庫執行個體的資料庫快照、以資料庫快照還原並建立新的資料庫執行個體，然後啟動新資料庫執行個體的版本升級，藉此在升級前先行測試資料庫執行個體與新版本的相容情形。接著，您就可以放心在升級的複製資料庫執行個體上測試，再決定是否要升級原有的資料庫執行個體。

Amazon RDS 上的 MySQL 次要版本

次要版本

- [the section called “MySQL 本”](#)

MySQL 本

MySQL 版本 8.0.37 現在可在 Amazon RDS 上使用。此版本包含 MySQL 社群和 Amazon RDS 所新增的修正和改良功能。

新功能和增強功能

修正執行立即 DDL 陳述式後面接著 UPDATE 而導致宣告失敗的錯誤。

Amazon RDS 上支援的 MySQL 主要版本

RDS for MySQL 主要版本在標準支援下，至少在對應的社群版本生命週期結束前會保持可用。您可以在 RDS 標準支援結束日期之後繼續付費執行主要版本。如需詳細資訊，請參閱 [使用 Amazon RDS 延長支援](#) 和 [Amazon RDS for MySQL 定價](#)。

您可以使用下列日期來規劃測試和升級週期。

Note

只有月份和年份的日期是近似值，並會在已知確切日期時進行更新。

MySQL 主要版本	社群發佈日期	RDS 發佈日期	社群生命週期結束日期	RDS 標準支援結束日期	RDS 延長支援第 1 年定價開始日期	RDS 延長支援第 3 年定價開始日期	RDS 延長支援結束日期
MySQL 8.0	2018 年 4 月 19 日	2018 年 10 月 23 日	2026 年 4 月	2026 年 7 月 31 日	2026 年 8 月 1 日	2028 年 8 月 1 日	2029 年 7 月 31 日
MySQL	2015 年 10 月 21 日	2016 年 2 月 22 日	2023 年 10 月	2024 年 2 月 29 日	2024 年 3 月 1 日	2026 年 3 月 1 日	2027 年 2 月 28 日

* MySQL 5.7 現在只能在 RDS 延伸 Support 下使用。如需詳細資訊，請參閱 [使用 Amazon RDS 延長支援](#)。

適用於 MySQL 版 RDS 的 Amazon RDS 擴展 Support 版本

下列內容列出適用於 MySQL 版本的 RDS 延伸 Support 的所有版本。

推出

- [適用於 MySQL 的 RDS 延伸 Support](#)
- [適用於 MySQL 的 RDS 延伸 Support](#)

適用於 MySQL 的 RDS 延伸 Support

適用於 MySQL 的 RDS 延伸 Support 版本已提供。

修正錯誤：

- 通過實現fix_after_pullout固定field.cc斷言失敗。
- 修正針對特定 SQL 查詢將中繼資料傳回至用戶端時，空指標失敗。這些查詢在子SELECT句中包含動態參數和子查詢。
- 修正使用鬆散GROUP BY索引掃描或非連續索引範圍掃描時不正確的結果。
- 修復了在持久性期間 MySQL 崩潰時丟失的 GTID 信息。
- 修復了可能導致 InnoDB 事務無限期掛起的競爭條件。
- 修正群組複寫的憑證資訊清除中的競爭狀況。
- 修復了並發頁面操作的向後索引掃描問題。
- 修正並行案例中不一致的全文檢索搜尋 (FTS) 狀態問題。
- 修正刪除資料表時變更緩衝區的宣告問題。
- 在所有插件類型中調用deinit函數的統一行為。

CVE 固定：

- [CVE-2024-20963](#)
- [CVE-2024-20993](#)
- [CVE-2024-20998](#)
- [CVE-2024-21009](#)
- [CVE-2024-21054](#)

- [CVE-2024-21055](#)
- [CVE-2024-21057](#)
- [CVE-2024-21062](#)
- [CVE-2024-21008](#)
- [CVE-2024-21013](#)
- [CVE-2024-21047](#)
- [CVE-2024-21087](#)
- [CVE-2024-21096](#)

適用於 MySQL 的 RDS 延伸 Support

適用於 MySQL 的 RDS 延伸 Support 版本已提供。

此版本包含下列 CVE 的修補程式：

- [CVE-2024-20963](#)

使用資料庫預覽環境

在 2023 年 7 月，Oracle 宣布了 MySQL 的新版本模型。此模型包括兩種類型的版本：創新版本和 LTS 版本。Amazon RDS 使 MySQL 創新版本可在 RDS 預覽環境中使用。若要深入了解 MySQL 創新版本，請參閱 [MySQL 創新和長期支援 \(LTS\) 版本簡介](#)。

資料庫預覽環境中的 RDS for MySQL 資料庫執行個體是類似於其他 RDS for MySQL 執行個體的功能。不過，您無法將資料庫預覽環境用於生產工作負載。

預覽環境具有下列限制：

- Amazon RDS 會在您建立資料庫執行個體後 60 天將其全部刪除，也會一併刪除任何備份和快照。
- 您只能使用一般用途 SSD 和佈建 IOPS SSD 儲存裝置。
- 您無法從 AWS Support 資料庫執行個體取得協助。相反，您可以將您的問題發佈到 AWS 受管理的問答社群 [AWS Re: post](#)。
- 您無法將資料庫執行個體的快照複製到生產環境。

預覽版支援下列選項。

- 您可以使用 db.m6i、db.r6i、db.m6g、db.m5、db.t3、db.r6g 和 db.r5 資料庫執行個體類別建立資料庫執行個體。如需 RDS 執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。
- 您可以同時使用單一可用區和異地同步備份部署。
- 您可以使用標準 MySQL 傾印和載入函數，從資料庫預覽環境匯出資料庫，或將資料庫匯入至資料庫預覽環境。

資料庫預覽環境中不支援的功能

資料庫預覽環境中無法使用下列功能：

- 跨區域快照複製
- 跨區域僅供讀取複本
- RDS Proxy

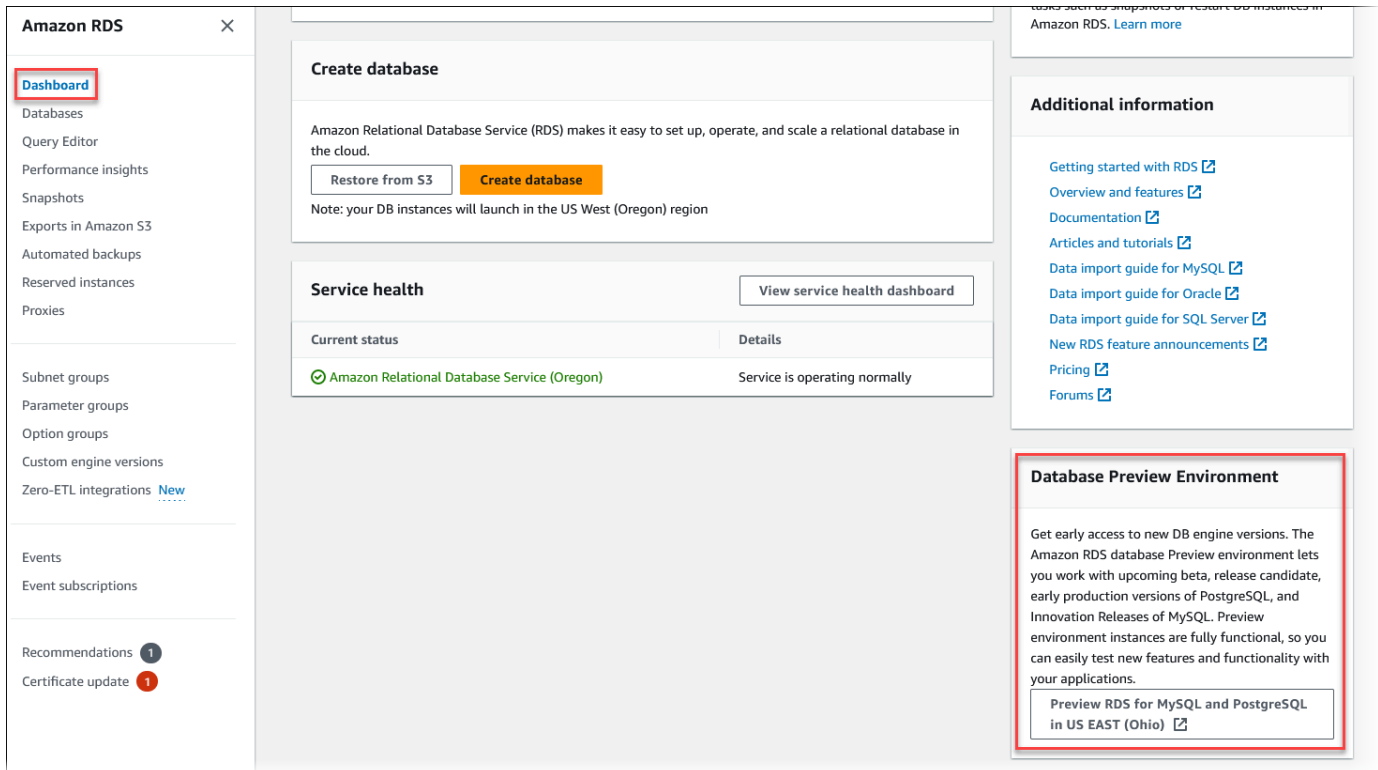
在資料庫預覽環境中建立新的資料庫執行個體

您可以使用 AWS Management Console、AWS CLI 或 RDS API 在資料庫預覽環境中建立資料庫執行個體。

主控台

在資料庫預覽環境中建立資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從導覽窗格中選擇 Dashboards (儀表板)。
3. 在儀表板頁面中，找出資料庫預覽環境區段，如下圖所示。



Amazon RDS ×

Dashboard

Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies

Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations [New](#)

Events
Event subscriptions

Recommendations **1**
Certificate update **1**

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

[Restore from S3](#) [Create database](#)

Note: your DB instances will launch in the US West (Oregon) region

Service health

[View service health dashboard](#)

Current status	Details
✔ Amazon Relational Database Service (Oregon)	Service is operating normally

Additional information

[Getting started with RDS](#)
[Overview and features](#)
[Documentation](#)
[Articles and tutorials](#)
[Data import guide for MySQL](#)
[Data import guide for Oracle](#)
[Data import guide for SQL Server](#)
[New RDS feature announcements](#)
[Pricing](#)
[Forums](#)

Database Preview Environment

Get early access to new DB engine versions. The Amazon RDS database Preview environment lets you work with upcoming beta, release candidate, early production versions of PostgreSQL, and Innovation Releases of MySQL. Preview environment instances are fully functional, so you can easily test new features and functionality with your applications.

[Preview RDS for MySQL and PostgreSQL in US EAST \(Ohio\)](#)

您可以直接導覽至[資料庫預覽環境](#)。在繼續之前，您必須確認並接受限制。

Database Preview Environment Service Agreement

The Amazon RDS Database Preview Environment is not covered by the Amazon RDS service level agreement (SLA), published at <https://aws.amazon.com/rds/sla>

Do not use the Amazon RDS Database Preview Environment for production purposes. You should only use this environment for development and testing.

Certain use cases might fail in this environment - for example, upgrading from a previous version is not supported.

I acknowledge this limited service agreement for the Amazon RDS Database Preview Environment and that I should only use this environment for development and testing.

[Cancel](#) [Accept](#)

- 若要建立 RDS for MySQL 資料庫執行個體，請遵循與建立任何 Amazon RDS 資料庫執行個體相同的程序。如需詳細資訊，請參閱 [建立資料庫執行個體](#) 中的 [主控台](#) 程序。

AWS CLI

若要使用 AWS CLI，在資料庫預覽環境中建立執行個體，請使用下列端點。

```
rds-preview.us-east-2.amazonaws.com
```

若要建立 RDS for MySQL 資料庫執行個體，請遵循與建立任何 Amazon RDS 資料庫執行個體相同的程序。如需詳細資訊，請參閱 [建立資料庫執行個體](#) 中的 [AWS CLI](#) 程序。

RDS API

若要使用 RDS API，在資料庫預覽環境中建立執行個體，請使用下列端點。

```
rds-preview.us-east-2.amazonaws.com
```

若要建立 RDS for MySQL 資料庫執行個體，請遵循與建立任何 Amazon RDS 資料庫執行個體相同的程序。如需詳細資訊，請參閱 [建立資料庫執行個體](#) 中的 [RDS API](#) 程序。

資料庫預覽環境中的 MySQL 8.3 版

MySQL 8.3 版現在可在 Amazon RDS 資料庫預覽環境中使用。MySQL 版本 8.3 包含了一些改進，這些改進在 [MySQL 8.3.0 的變化中](#) 描述。

如需資料庫預覽環境的相關資訊，請參閱 [the section called “資料庫預覽環境”](#)。若要從主控台存取預覽環境，請選取 <https://console.aws.amazon.com/rds-preview/>。

資料庫預覽環境中的 MySQL 版本 8.2

MySQL 8.2 版現在可在 Amazon RDS 資料庫預覽環境中使用。MySQL 版本 8.2 包含了一些改進，這些改進在 [MySQL 8.2.0 中所描述的變化](#)。

如需資料庫預覽環境的相關資訊，請參閱 [the section called “資料庫預覽環境”](#)。若要從主控台存取預覽環境，請選取 <https://console.aws.amazon.com/rds-preview/>。

資料庫預覽環境中的 MySQL 8.1 版

MySQL 8.1 版現可在 Amazon RDS 資料庫預覽環境中使用。MySQL 8.1 版包含 [MySQL 8.1.0 中的變更](#) 所述的數個改善項目。

如需資料庫預覽環境的相關資訊，請參閱 [the section called “ 資料庫預覽環境”](#)。若要從主控台存取預覽環境，請選取 <https://console.aws.amazon.com/rds-preview/>。

Amazon RDS for MySQL 的已取代版本

Amazon RDS for MySQL 5.1、5.5 和 5.6 版已被取代。

如需適用於 MySQL 的 Amazon RDS 淘汰政策相關資訊，請參閱 [Amazon RDS 常見問答集](#)。

連接至執行 MySQL 資料庫引擎的資料庫執行個體

您必須建立資料庫執行個體，才能連接至執行 MySQL 資料庫引擎的資料庫執行個體。如需相關資訊，請參閱「[建立 Amazon RDS 資料庫執行個體](#)」。當 Amazon RDS 佈建您的資料庫執行個體後，您就可以使用任何標準 MySQL 用戶端應用程式或公用程式來連接至執行個體。在連線字串中，請指定來自資料庫執行個體端點的 DNS 地址做為 host 參數，並指定來自資料庫執行個體端點的連接埠號碼做為 port 參數。

若要對 RDS 資料庫執行個體進行驗證，您可以使用其中一種驗證方法進行 MySQL 和 AWS Identity and Access Management (IAM) 資料庫驗證：

- 若要了解如何使用 MySQL 的其中一個身分驗證方法向 MySQL 進行驗證，請參閱 MySQL 文件中的 [身分身分驗證方法](#)。
- 若要了解如何使用 IAM 資料庫身分驗證向 MySQL 進行驗證，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

您可以使用 MySQL 命令列用戶端之類的工具，以連線至 MySQL 資料庫執行個體。如需使用 MySQL 命令列用戶端的詳細資訊，請參閱 MySQL 文件中的 [mysql – MySQL 命令列用戶端](#)。您可以用來連接的一個 GUI 應用程式是 MySQL Workbench。如需詳細資訊，請參閱 [下載 MySQL Workbench](#) 頁面。如需安裝 MySQL (包括 MySQL 命令列用戶端) 的資訊，請參閱 [安裝和升級 MySQL](#)。

若要從其 Amazon VPC 之外連線至資料庫執行個體，資料庫執行個體必須可公開存取、必須使用資料庫執行個體之安全群組的傳入規則授予存取權，且必須符合其他要求。如需詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

您可以在與 MySQL 資料庫執行個體的連線上使用 Secure Sockets Layer (SSL) 或 Transport Layer Security (TLS) 加密。如需相關資訊，請參閱 [將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)。如果您正在使用 AWS Identity and Access Management (IAM) 資料庫身份驗證，請務必使用 SSL/TLS 連線。如需相關資訊，請參閱「[適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)」。

您也可以從 Web 伺服器連接至資料庫執行個體。如需詳細資訊，請參閱 [教學：建立 Web 伺服器和 Amazon RDS 資料庫執行個體](#)。

Note

如需連接至 MariaDB 資料庫執行個體的相關資訊，請參閱 [連接至執行 MariaDB 資料庫引擎的資料庫執行個體](#)。

內容

- [尋找適用於 MySQL 資料庫執行個體的 RDS 連線資訊](#)
- [安裝 MySQL 命令列用戶端](#)
- [從 MySQL 命令列用戶端連線 \(未加密\)](#)
- [從 MySQL Workbench 連線](#)
- [使用 Amazon Web Services \(AWS \) JDBC 驅動程序連接到適用於 MySQL 的 RDS](#)
- [使用 Amazon Web Services \(AWS \) Python 驅動程序連接到適用於 MySQL 的 RDS](#)
- [對 MySQL 資料庫執行個體的連線進行故障診斷](#)

尋找適用於 MySQL 資料庫執行個體的 RDS 連線資訊

資料庫執行個體的連線資訊包括其端點、連接埠，以及有效的資料庫使用者，例如主要使用者。例如，假設端點值為 `mydb.123456789012.us-east-1.rds.amazonaws.com`。在此情況下，連接埠值為 `3306`，資料庫使用者為 `admin`。藉由此資訊，您可以在連線字串中指定下列值：

- 對於主機或主機名稱或 DNS 名稱，請指定 `mydb.123456789012.us-east-1.rds.amazonaws.com`。
- 對於連接埠，指定 `3306`。
- 對於使用者，請指定 `admin`。

若要連線到資料庫執行個體，請針對 MySQL 資料庫引擎使用任何用戶端。例如，您可以使用 MySQL 命令列用戶端或 MySQL 工作台。

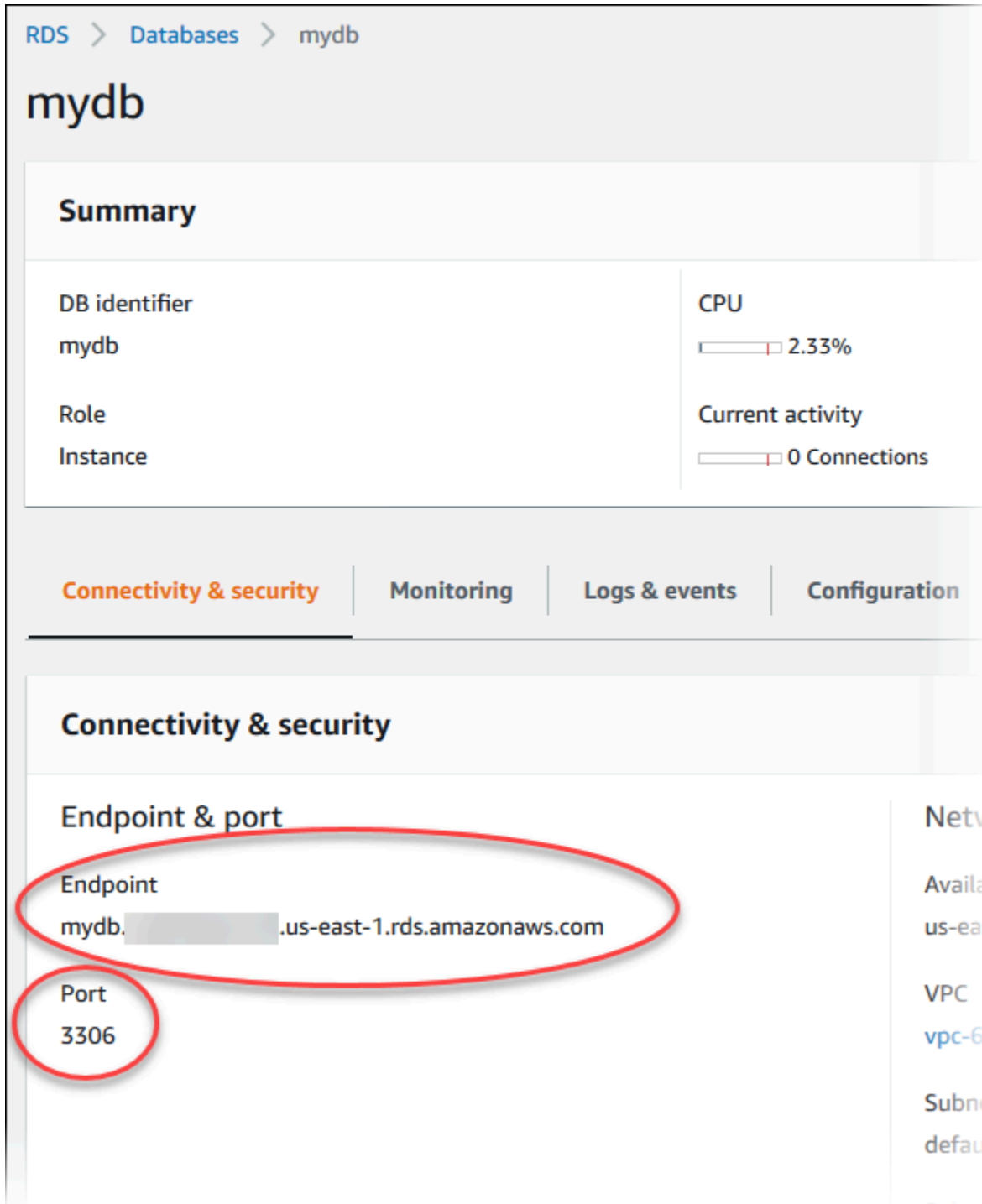
若要尋找資料庫執行個體的連線資訊，您可以使用 AWS Management Console、命 AWS CLI [describe-db-instances](#) 令或 Amazon RDS API Descri [beDBInstances](#) 操作列出其詳細資訊。

主控台

若要在中尋找資料庫執行個體的連線資訊 AWS Management Console

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫) 以顯示資料庫執行個體的清單。
3. 選擇 MySQL 資料庫執行個體名稱以顯示其詳細資訊。

- 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連接至資料庫執行個體。



RDS > Databases > mydb

mydb

Summary

DB identifier mydb	CPU 2.33%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port

Endpoint mydb. [redacted].us-east-1.rds.amazonaws.com	Network
Port 3306	Availability Zone us-east-1
	VPC vpc-65
	Subnet default

- 如果您需要尋找主要使用者名稱，請選擇 Configuration (組態) 索引標籤並檢視 Master username (主要使用者名稱) 值。

AWS CLI

若要使用尋找 MySQL 資料庫執行個體的連線資訊 AWS CLI，請呼叫命 [describe-db-instances](#) 令。在呼叫中，查詢資料庫執行個體 ID、端點、連接埠和主要使用者名稱。

對於 Linux macOS、或 Unix：

```
aws rds describe-db-instances \  
  --filters "Name=engine,Values=mysql" \  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

在 Windows 中：

```
aws rds describe-db-instances ^  
  --filters "Name=engine,Values=mysql" ^  
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

輸出內容應如下所示：

```
[  
  [  
    "mydb1",  
    "mydb1.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "mydb2",  
    "mydb2.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ]  
]
```

RDS API

若要使用 Amazon RDS API 尋找資料庫執行個體的連線資訊，請呼叫 [DescribeDBInstances](#) 操作。在輸出中，尋找端點地址、端點連接埠和主要使用者名稱的值。

安裝 MySQL 命令列用戶端

大多數的 Linux 發行版均包括 MariaDB 用戶端，而不是 Oracle MySQL 用戶端。若要在 Amazon Linux 2023 上安裝 MySQL 命令列用戶端，請執行下列命令：

```
sudo dnf install mariadb105
```

若要在 Amazon Linux 2 上安裝 MySQL 命令列用戶端，請執行下列命令：

```
sudo yum install mariadb
```

若要在大多數以 Debian 為基礎的 Linux 發行版上安裝 MySQL 命令列用戶端，請執行下列命令：

```
apt-get install mariadb-client
```

若要檢查 MySQL 命令列用戶端的版本，請執行下列命令：

```
mysql --version
```

若要讀取目前用戶端版本的 MySQL 文件，請執行下列命令：

```
man mysql
```

從 MySQL 命令列用戶端連線 (未加密)

Important

僅當用戶端和伺服器位於同一 VPC 且網路受信任時，才使用未加密的 MySQL 連線。如需使用加密連線的詳細資訊，請參閱 [使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)。

若要使用 MySQL 命令列用戶端連線至資料庫執行個體，請在命令提示字元中輸入下列命令。在 `-h` 參數中，換成您的資料庫執行個體的 DNS 名稱 (端點)。在 `-P` 參數中，換成您的資料庫執行個體的連接埠。對於 `-u` 參數，請以有效資料庫使用者取代使用者名稱，例如主要使用者。出現提示時，輸入主要使用者密碼。

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -  
u mymasteruser -p
```

輸入使用者的密碼之後，您應該會看到類似如下的輸出。

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

從 MySQL Workbench 連線

若要從 MySQL Workbench 連線

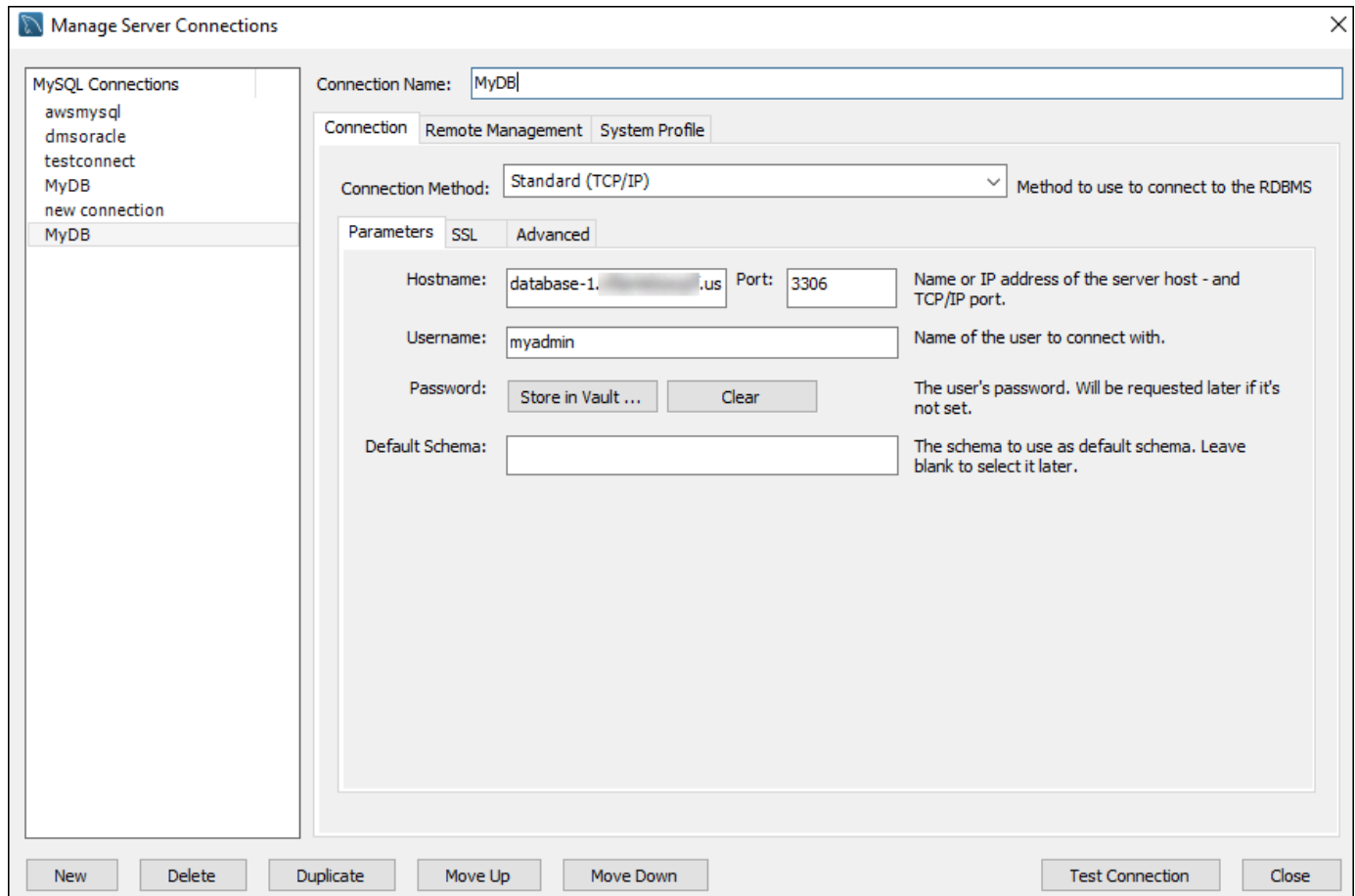
1. 在 [下載 MySQL Workbench](#) 的地方下載並安裝 MySQL Workbench。
2. 開啟 MySQL Workbench。



3. 從 Database (資料庫)，選擇 Manage Connections (管理連線)。
4. 在 Manage Server Connections (管理伺服器連線視窗) 中，選擇 New (新增)。
5. 在 Connect to Database (連線到資料庫) 的視窗中，輸入下列資訊：
 - Stored Connection (儲存的連線) – 輸入連線的名稱，例如 **MyDB**。
 - Hostname (主機名稱) – 輸入資料庫執行個體端點。
 - Port (連接埠) – 輸入資料庫執行個體使用的連接埠。
 - Username (使用者名稱) – 輸入資料庫使用者有效的使用者名稱，例如主要使用者。

- Password (密碼) – 選用，選擇 Store in Vault (儲存於保存庫) 然後輸入並儲存使用者密碼。

視窗看起來類似如下：



您可以使用 MySQL Workbench 的功能來自訂連線。例如，您可以使用 SSL 索引標籤來設定 SSL/TLS 連線。如需使用 MySQL Workbench 的詳細資訊，請參閱 [MySQL Workbench 文件](#)。使用 SSL/TLS 加密連至 MySQL 資料庫執行個體的用戶端連線。請參閱 [使用 SSL/TLS 加密對 MySQL 資料庫執行個體的用戶端連線](#)。

6. 選用，選擇 Test Connection (測試連線) 以確認成功連線到資料庫執行個體。
7. 選擇關閉。
8. 從 Database (資料庫)，選擇 Connect to Database (連線到資料庫)。
9. 從 Stored Connection (已儲存的連線)，選擇您的連線。
10. 選擇 OK (確定)。

使用 Amazon Web Services (AWS) JDBC 驅動程序連接到適用於 MySQL 的 RDS

Amazon Web Services (AWS) JDBC 驅動程序被設計為一個高級 JDBC 包裝。這個包裝是補充和擴展現有 JDBC 驅動程序的功能。該驅動程序與社區 MySQL 連接器/J 驅動程序和社區 MariaDB 連接器/J 驅動程序兼容。

若要安裝 AWS JDBC 驅動程式，請附加 AWS JDBC 驅動程式 .jar 檔案 (位於應用程式中CLASSPATH)，並保留對個別社群驅動程式的參考。更新相應的連接 URL 前綴，如下所示：

- jdbc:mysql:// 設定為 jdbc:aws-wrapper:mysql://
- jdbc:mariadb:// 設定為 jdbc:aws-wrapper:mariadb://

如需有關 AWS JDBC 驅動程式的詳細資訊以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS\) JDBC 驅動程式 GitHub 儲存庫](#)。

使用 Amazon Web Services (AWS) Python 驅動程序連接到適用於 MySQL 的 RDS

Amazon Web Services (AWS) Python 驅動程序被設計為一個先進的 Python 包裝。此包裝器與開源 Psycopg 驅動程序的功能互補並擴展。AWS Python 驅動程式支援 3.8 及更高版本。您可以使用 pip 命令以及 psycopg 開放原始碼套件來安裝套件。aws-advanced-python-wrapper

有關 AWS Python 驅動程序的更多信息以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS \) Python 驅動程序 GitHub 儲存庫](#)。

對 MySQL 資料庫執行個體的連線進行故障診斷

無法連接至新的資料庫執行個體有兩個常見原因：

- 建立資料庫執行個體所用的安全群組，未授權從執行 MySQL 應用程式或公用程式的裝置或 Amazon EC2 執行個體進行連線。資料庫執行個體必須擁能會授權連線的 VPC 安全群組。如需詳細資訊，請參閱 [Amazon VPC](#) 和 [Amazon RDS](#)。

您可以新增或編輯安全群組中的傳入規則。針對 Source (來源)，選擇 My IP (我的 IP)。此舉允許透過您的瀏覽器中偵測到的 IP 地址存取資料庫執行個體。

- 資料庫執行個體是使用預設連接埠 3306 來建立，而公司的防火牆規則禁止從公司網路中的裝置連接至該連接埠。若要避免此失敗情況，請以不同的連接埠重新建立執行個體。

如需連線問題的詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

保護 MySQL 資料庫執行個體連線的安全

您可以管理 MySQL 資料庫執行個體的安全性。

主題

- [Amazon RDS 上的 MySQL 安全性](#)
- [使用 RDS for MySQL 的密碼驗證外掛程式](#)
- [使用 SSL/TLS 加密對 MySQL 資料庫執行個體的用戶端連線](#)
- [將應用程式更新為使用新的 SSL/TLS 憑證來連線至 MySQL 資料庫執行個體](#)
- [針對 MySQL 使用 Kerberos 身分驗證](#)

Amazon RDS 上的 MySQL 安全性

MySQL 資料庫執行個體的安全性有三個受管層級：

- AWS Identity and Access Management 控制哪些人可以在資料庫執行個體上執行 Amazon RDS 管理動作。當您連線到 AWS 使用 IAM 登入資料時，您的 IAM 帳戶必須具有 IAM 政策，以授與執行 Amazon RDS 管理操作所需的許可。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。
- 建立資料庫執行個體時，您可使用 VPC 安全群組控管哪些裝置和 Amazon EC2 執行個體可以開啟與端點和資料庫執行個體連接埠的連線。您可以使用 Secure Sockets Layer (SSL) 和 Transport Layer Security (TLS) 建立這些連線。此外，貴公司的防火牆規則可控管在公司內執行的裝置是否可開啟與資料庫執行個體的連線。
- 若要驗證 MySQL 資料庫執行個體的登入資訊與權限，您可採取下列任一方式，或搭配使用多種方法。

您可採用與 MySQL 獨立執行個體相同的驗證方式。CREATE USER、RENAME USER、GRANT、REVOKE 及 SET PASSWORD 等命令的運作方式與現場部署資料庫所使用的命令相同，會直接修改資料庫結構描述資料表。不過，直接修改資料庫結構描述資料表並不是最佳作法，從 8.0.36 版開始，不支援此功能。如需詳細資訊，請參閱 MySQL 文件中的 [存取控制和帳戶管理](#)。

您也可以使用 IAM 資料庫身分驗證。透過 IAM 資料庫身分驗證，您可以使用 IAM 使用者或 IAM 角色以及身分驗證字符，驗證您的資料庫執行個體。身分驗證字符是不重複的值，由 Signature 第 4 版簽署程序所產生。透過使用 IAM 資料庫身份驗證，您可以使用相同的登入資料來控制對 AWS 源和資料庫的存取。如需詳細資訊，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

另一種選擇是 RDS for MySQL 適用的 Kerberos 身分驗證。資料庫執行個體與 AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) 搭配使用以啟用 Kerberos 驗證。使用者向加入信任網域的 MySQL 資料庫執行個體進行身分驗證時，會轉送身分驗證請求。轉寄的要求會移至您使用建立的網域目錄 AWS Directory Service。如需詳細資訊，請參閱 [針對 MySQL 使用 Kerberos 身分驗證](#)。

在您建立 Amazon RDS 資料庫執行個體後，主要使用者具有下列預設權限：

引擎版本	系統權限	資料庫角色
適用於 MySQL 的 RDS 版本 8.0.36 及更高版本	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN , SET_USER_ID , XA_RECOVER_ADMIN	rds_superuser_role 如需 rds_superuser_role 的相關資訊，請參閱 角色型權限模型 。
適用於 MySQL 的 RDS 版本低於 8.0.36	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, REPLICATION SLAVE	—

Note

雖然您可刪除資料庫執行個體上的主要使用者，但不建議這麼做。若要重新建立主要使用者，請使用 [修改資料庫執行個體](#) RDS API 作業或 [modify-db-instance](#) AWS CLI 命令，並使用適當

的參數指定新的主要使用者密碼。若執行個體中不在此主要使用者，系統會以指定的密碼建立主要使用者。

為了提供各資料庫執行個體管理服務，建立資料庫執行個體時，系統會一併建立 rdsadmin 使用者。若企圖移除、重新命名 rdsadmin 帳戶或變更其密碼或權限，皆會導致系統發生錯誤。

請限制使用標準 kill 和 kill_query 命令，藉此允許資料庫執行個體的管理操作。此外，本服務還提供 Amazon RDS 命令 rds_kill 與 rds_kill_query，讓您在結束資料庫執行個體上的使用者工作階段或查詢。

使用 RDS for MySQL 的密碼驗證外掛程式

MySQL 提供更安全的 validate_password 外掛程式。外掛程式使用您 MySQL 資料庫執行個體的資料庫參數中的參數，強制採用密碼政策。執行 MySQL 5.7 和 8.0 版本的資料庫執行個體支援該外掛程式。如需 validate_password 外掛程式詳細資訊，請參閱 MySQL 文件中的[密碼驗證外掛程式](#)。

啟用 MySQL 資料庫執行個體的 validate_password 外掛程式

1. 連接至 MySQL 資料庫執行個體，然後執行下列命令：

```
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

2. 在資料庫執行個體使用的資料庫參數群組中設定外掛程式參數。

如需參數詳細資訊，請參閱 MySQL 文件中的[密碼驗證外掛程式選項和變數](#)。

如需修改資料庫執行個體參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

安裝並啟用 password_validate 外掛程式後，請重新設定現有密碼，以符合新驗證政策。

Amazon RDS 不驗證密碼。MySQL 資料庫執行個體執行密碼驗證。若您使用 AWS Management Console、modify-db-instance AWS CLI 命令或 ModifyDBInstance RDS API 操作來設定使用者密碼，則即使新密碼不符合密碼政策亦可成功。然而，新密碼須符合密碼政策才能設定至 MySQL 資料庫執行個體。在這種情況下，Amazon RDS 紀錄下列事件。

```
"RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has failed.
```

如需 Amazon RDS 事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。

使用 SSL/TLS 加密對 MySQL 資料庫執行個體的用戶端連線

Secure Sockets Layer (SSL) 是一種產業標準通訊協定，可保護用戶端與伺服器之間的網路連線。SSL 3.0 版之後，名稱已變更為 Transport Layer Security (TLS)。Amazon RDS 支援在 MySQL 資料庫執行個體上啟用 SSL/TLS 加密。使用 SSL/TLS，您可以加密應用程式用戶端與 MySQL 資料庫執行個體之間的連接。SSL/TLS 支援適用於所有的 AWS 區域 for MySQL。

主題

- [將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)
- [MySQL 資料庫執行個體的所有連線都需要 SSL/TLS](#)
- [使用 SSL/TLS \(加密\) 從 MySQL 命令列用戶端連線](#)

將 SSL/TLS 搭配 MySQL 資料庫執行個體使用

當 Amazon RDS 佈建執行個體時，Amazon RDS 會建立 SSL/TLS 憑證，並將該憑證安裝在資料庫執行個體上。憑證由憑證授權機構簽署。SSL/TLS 憑證會以通用名稱 (CN) 加入資料庫執行個體端點，讓 SSL/TLS 憑證免於遭受詐騙攻擊。

Amazon RDS 建立的 SSL/TLS 憑證是可信任的根實體，應可適用於大多數情況，但若您的應用程式不接受憑證鏈結，則可能無法使用。若您的應用程式不接受憑證鏈結，您可能需要使用中繼憑證來連線至您的 AWS 區域。舉例來說，您必須使用中繼憑證，透過 SSL/TLS 連線至 AWS GovCloud (US) 區域。

如需有關下載憑證的詳細資訊，請參閱。如需在使用 SSL/TLS 搭配 MySQL 的詳細資訊，請參閱[將應用程式更新為使用新的 SSL/TLS 憑證來連線至 MySQL 資料庫執行個體](#)。

MySQL 使用 OpenSSL 進行安全連線。Amazon RDS for MySQL 支援 Transport Layer Security (TLS) 1.0、1.1、1.2 和 1.3 版。TLS 支援情形取決於 MySQL 版本。下表顯示 MySQL 版本的 TLS 支援。

MySQL 版本	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
MySQL 8.0	不支援	不支援	支援	支援

MySQL 版本	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
MySQL 5.7	支援	支援	支援	不支援

您可以要求特定使用者帳戶使用 SSL/TLS 連線。例如，您可以根據 MySQL 版本，使用下列任一陳述式，要求使用者帳戶 `encrypted_user` 使用 SSL/TLS 連線。

若要這麼做，請使用下列陳述式。

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

如需使用 MySQL 進行 SSL/TLS 連線的詳細資訊，請參閱 MySQL 文件中的[使用加密連線](#)。

MySQL 資料庫執行個體的所有連線都需要 SSL/TLS

使用 `require_secure_transport` 參數來要求使用者與 MySQL 資料庫執行個體的所有連線都使用 SSL/TLS。依預設，`require_secure_transport` 參數設為 OFF。您可以將 `require_secure_transport` 參數設為 ON，以要求對資料庫執行個體的連線使用 SSL/TLS。

您可以更新資料庫執行個體的資料庫參數群組，以設定 `require_secure_transport` 參數值。您不需要重新啟動資料庫執行個體，變更即可生效。

當資料庫執行個體的 `require_secure_transport` 參數設為 ON 時，如果可以建立加密的連線，則資料庫用戶端即可連線到該資料庫叢集。否則，類似下列內容的錯誤訊息會傳回至用戶端：

```
MySQL Error 3159 (HY000): Connections using insecure transport are prohibited while --require_secure_transport=ON.
```

如需有關設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

如需 `require_secure_transport` 參數的詳細資訊，請參閱[MySQL 文件](#)。

使用 SSL/TLS (加密) 從 MySQL 命令列用戶端連線

如果您使用的是 MySQL 5.7 版、MySQL 8.0 版或 MariaDB 版本，`mysql` 用戶端程式的參數略有不同。

若要找出您擁有的版本，請執行具有 `--version` 選項的 `mysql` 命令。在以下範例中，輸出顯示了用戶端程式是來自 MariaDB。

```
$ mysql --version
mysql Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

大多數 Linux 發行版本，例如 Amazon Linux、CentOS、SUSE 和 Debian 都用 MariaDB 取代 MySQL，而且發行版本中的 `mysql` 版是來自 MariaDB。

若要使用 SSL/TLS 連線到資料庫執行個體，請執行以下步驟：

使用 MySQL 命令列用戶端搭配 SSL/TLS 連接至資料庫執行個體

1. 下載適用於所有 AWS 區域 的根憑證。

如需有關下載憑證的詳細資訊，請參閱。

2. 使用 MySQL 命令列用戶端搭配 SSL/TLS 加密連接至資料庫執行個體。在 `-h` 參數中，將 DNS 名稱 (端點) 取代為您的資料庫執行個體。在 `--ssl-ca` 參數中，取代 SSL/TLS 憑證檔名稱。在 `-P` 參數中，取代為您的資料庫執行個體連接埠。在 `-u` 參數中，請以有效資料庫使用者取代使用者名稱，例如主要使用者。出現提示時，輸入主要使用者密碼。

下列範例顯示如何對於 MySQL 5.7 用戶端更新的版本使用 `--ssl-ca` 參數啟動用戶端：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
```

若要要求 SSL/TLS 連線根據 SSL/TLS 憑證中的端點來驗證資料庫執行個體端點，請輸入下列命令：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u myadmin -p
```

下列範例顯示如何對於 MariaDB 用戶端更新的版本使用 `--ssl-ca` 參數啟動用戶端：

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

3. 出現提示時，輸入主要使用者密碼。

您將會看到類似下列的輸出。

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

將應用程式更新為使用新的 SSL/TLS 憑證來連線至 MySQL 資料庫執行個體

自 2023 年 1 月 13 日起，Amazon RDS 已發佈新的憑證認證機構 (CA) 憑證，使用 Secure Socket Layer 或 Transport Layer Security (SSL/TLS) 來連線至 RDS 資料庫執行個體。接下來，您可以找到更新應用程式使用新憑證的相關資訊。

本主題可協助您判斷任何用戶端應用程式是否使用 SSL/TLS 連線至您的資料庫執行個體。若是如此，您可以進一步檢查那些應用程式是否需要驗證憑證才能連線。

Note

有些應用程式設定為只有在成功驗證伺服器上的憑證時，才能連線至 MySQL 資料庫執行個體。對於這些應用程式，您必須更新用戶端應用程式信任存放區來包含新的 CA 憑證。您可以指定下列 SSL 模式：disabled、preferred 及 required。當您使用 preferred SSL 模式且 CA 憑證不存在或不是最新版本時，連線會回退至不使用 SSL 且不加密的狀況下進行連線。

因為這些較新版本使用 OpenSSL 通訊協定，除非指定 required SSL 模式，否則過期的伺服器憑證不會阻止成功連線。

我們建議您避免使用 preferred 模式。在 preferred 模式中，如果連線遇到無效憑證，則會停止使用加密並繼續使用未加密連線。

更新用戶端應用程式信任存放區中的 CA 憑證之後，您就可以在資料庫執行個體輪換憑證。強烈建議先在開發或預備環境中測試這些步驟，再於生產環境中實作。

如需憑證輪換的詳細資訊，請參閱[輪換您的 SSL/TLS 憑證](#)。如需下載憑證的詳細資訊，請參閱。如需對 MySQL 資料庫執行個體使用 SSL/TLS 的資訊，請參閱[將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)。

主題

- [判斷任何應用程式是否使用 SSL 連線至 MySQL 資料庫執行個體](#)
- [判斷用戶端是否需要驗證憑證才能連線](#)

- [更新應用程式信任存放區](#)
- [建立 SSL 連線的 Java 程式碼範例](#)

判斷任何應用程式是否使用 SSL 連線至 MySQL 資料庫執行個體

如果您使用 Amazon RDS for MySQL 5.7 或 8.0 版，且效能結構描述已啟用，請執行下列查詢，以檢查連線是否使用 SSL/TLS。如需啟用效能結構描述的資訊，請參閱 MySQL 文件中的[效能結構描述快速入門](#)。

```
mysql> SELECT id, user, host, connection_type
        FROM performance_schema.threads pst
        INNER JOIN information_schema.processlist isp
        ON pst.processlist_id = isp.id;
```

在此輸出範例中，可看到您自己的工作階段 (admin) 和以 webapp1 登入的應用程式都使用 SSL。

```
+----+-----+-----+-----+
| id | user          | host          | connection_type |
+----+-----+-----+-----+
|  8 | admin         | 10.0.4.249:42590 | SSL/TLS         |
|  4 | event_scheduler | localhost      | NULL            |
| 10 | webapp1       | 159.28.1.1:42189 | SSL/TLS       |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

判斷用戶端是否需要驗證憑證才能連線

您可以檢查 JDBC 用戶端和 MySQL 用戶端是否需要驗證憑證才能連線。

JDBC

以下 MySQL Connector/J 8.0 範例指出一種方式來檢查應用程式的 JDBC 連線屬性，以判斷是否需要有效憑證才能成功連線。如需 MySQL 的所有 JDBC 連線選項的詳細資訊，請參閱 MySQL 文件中的[組態屬性](#)。

使用 MySQL Connector/J 8.0 時，如果連線屬性的 `sslMode` 設為 `VERIFY_CA` 或 `VERIFY_IDENTITY`，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

Note

如果您使用 MySQL Java Connector v5.1.38 或更高版本，或 MySQL Java Connector v8.0.9 或更高版本來連線至您的資料庫，即使您尚未明確設定應用程式在連線到資料庫時使用 SSL/TLS，這些用戶端驅動程式會預設為使用 SSL/TLS。此外，使用 SSL/TLS 時，它們會執行部分憑證驗證，如果資料庫伺服器憑證已過期，則無法連線。

MySQL

以下 MySQL 用戶端範例指出兩種方式來檢查指令碼的 MySQL 連線，以判斷是否需要有效憑證才能成功連線。如需 MySQL 用戶端所有連線選項的詳細資訊，請參閱 MySQL 文件中的[加密連線的用戶端組態](#)。

使用 MySQL 5.7 或 MySQL 8.0 用戶端時，如果您將 `--ssl-mode` 選項指定為 `VERIFY_CA` 或 `VERIFY_IDENTITY`，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
--ssl-mode=VERIFY_CA
```

使用 MySQL 5.6 用戶端時，如果您指定 `--ssl-verify-server-cert` 選項，則需要以伺服器 CA 憑證來驗證 SSL 連線，如下列範例所示。

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
--ssl-verify-server-cert
```

更新應用程式信任存放區

如需為 MySQL 應用程式更新信任存放區的資訊，請參閱 MySQL 文件中的[安裝 SSL 憑證](#)。

如需下載根憑證的資訊，請參閱。

如需匯入憑證的範例指令碼，請參閱 [將憑證匯入信任存放區的範例指令碼](#)。

Note

更新信任存放區時，除了新增憑證，您還可以保留舊憑證。

如果您在應用程式中使用 mysql JDBC 驅動程式，請在應用程式中設定下列屬性。

```
System.setProperty("javax.net.ssl.trustStore", certs);  
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

啟動應用程式時，設定下列屬性。

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -  
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

建立 SSL 連線的 Java 程式碼範例

下列程式碼範例示範如何使用 JDBC 設定 SSL 連線，以驗證伺服器憑證。

```
public class MySQLSSLTest {  
  
    private static final String DB_USER = "username";  
    private static final String DB_PASSWORD = "password";  
    // This key store has only the prod root ca.  
    private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";  
    private static final String KEY_STORE_PASS = "keystore-password";
```



```
public static void test(String[] args) throws Exception {
    Class.forName("com.mysql.jdbc.Driver");

    System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
    System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

    Properties properties = new Properties();
    properties.setProperty("sslMode", "VERIFY_IDENTITY");
    properties.put("user", DB_USER);
    properties.put("password", DB_PASSWORD);

    Connection connection = null;
    Statement stmt = null;
    ResultSet rs = null;
    try {
        connection =
DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
        stmt = connection.createStatement();
        rs=stmt.executeQuery("SELECT 1 from dual");
    } finally {
        if (rs != null) {
            try {
                rs.close();
            } catch (SQLException e) {
            }
        }
        if (stmt != null) {
            try {
                stmt.close();
            } catch (SQLException e) {
            }
        }
        if (connection != null) {
            try {
                connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
    return;
}
```

```
}  
}
```

Important

判斷資料庫連線使用 SSL/TLS 並更新應用程式信任存放區之後，您可以更新資料庫以使用 rds-ca-rsa 2048-g1 憑證。如需說明，請參閱[透過修改資料庫執行個體或叢集來更新 CA 憑證](#)中的步驟 3。

指定此處所顯示提示以外的密碼，作為安全最佳實務。

針對 MySQL 使用 Kerberos 身分驗證

您可以使用 Kerberos 身分驗證來在使用者連線到您的 MySQL 資料庫執行個體時對其進行身分驗證。資料庫執行個體與 AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) 搭配使用以啟用 Kerberos 驗證。使用者向加入信任網域的 MySQL 資料庫執行個體進行身分驗證時，會轉送身分驗證請求。轉寄的要求會移至您使用建立的網域目錄 AWS Directory Service。

將您的所有登入資料保留在相同目錄可以節省您的時間和精力。透過這種方式，這樣您就有一個集中的位置來存放及管理多個資料庫執行個體的登入資料。使用目錄也可以改善您的整體安全性描述檔。

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需使用 Amazon RDS 搭配 Kerberos 身分驗證的版本和區域可用性的詳細資訊，請參閱[Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎](#)。

MySQL 資料庫執行個體設定 Kerberos 身分驗證概觀

如要針對 MySQL 資料庫執行個體設定 Kerberos 身分驗證，請完成下列一般步驟，稍後會有這方面的詳細說明：

1. 用 AWS Managed Microsoft AD 於建立 AWS Managed Microsoft AD 目錄。您可以使用 AWS Management Console AWS CLI、或 AWS Directory Service 來建立目錄。如需這樣做的詳細資訊，請參閱《AWS Directory Service 管理指南》中的[建立 AWS Managed Microsoft AD 目錄](#)。
2. 建立使用受管 IAM 政策的 AWS Identity and Access Management (IAM) 角色 AmazonRDSDirectoryServiceAccess。該角色允許 Amazon RDS 對您的目錄進行呼叫。

若要讓角色允許存取，必須在您的 AWS 帳戶中啟動 AWS Security Token Service (AWS STS) 端點。AWS 區域 AWS STS 端點預設在所有端點都處於作用中狀態 AWS 區域，您可以使用它們，而無需任何進一步的動作 如需詳細資訊，請參閱《IAM 使用者指南》[AWS STS AWS 區域中的「啟用和停用」](#)。

3. 使用 Microsoft 活動目錄工具在 AWS Managed Microsoft AD 目錄中創建和配置用戶。如需有關在您的 Active Directory 中建立使用者的詳細資訊，請參閱[管理指南中的 AWS 管理 Microsoft AD 中的 AWS Directory Service 使用者和群組](#)。
4. 建立或修改 MySQL 資料庫執行個體。如果您在建立請求中使用 CLI 或 RDS API，請使用 Domain 參數指定網域識別符。使用您在建立目錄時產生的 d-* 識別符，以及您建立的角色名稱。

如果您修改了現有的 MySQL 資料庫執行個體以使用 Kerberos 身分驗證，將設定資料庫執行個體的網域和 IAM 角色參數。在與網域目錄相同的 VPC 中尋找資料庫執行個體。

5. 使用 Amazon RDS 主要使用者登入資料來連線到 MySQL 資料庫執行個體。使用 CREATE USER 子句 IDENTIFIED WITH 'auth_pam' 在 MySQL 中建立使用者。您透過這種方式建立的使用者可以使用 Kerberos 身分驗證登入 MySQL 資料庫執行個體。

為 MySQL 資料庫執行個體設定 Kerberos 身分驗證

您可 AWS Managed Microsoft AD 以用來設定 MySQL 資料庫執行個體的 Kerberos 驗證。如要設定 Kerberos 身分驗證，您可以採取下列步驟。

第 1 步：使用創建一個目錄 AWS Managed Microsoft AD

AWS Directory Service 在 AWS 雲端中建立完全受管理的作用中目錄。當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會代表您建立兩個網域控制站和網域名稱系統 (DNS) 伺服器。目錄伺服器是在 VPC 的不同子網路中建立。此備援有助於確保即使發生故障，您仍然可以存取目錄。

當您建立 AWS Managed Microsoft AD 目錄時，請代表您 AWS Directory Service 執行下列工作：

- 在 VPC 內設定 Active Directory。
- 建立含有使用者名稱 Admin 與指定密碼的目錄管理員帳戶。您可以使用此帳戶來管理目錄。

Note

請務必儲存此密碼。AWS Directory Service 不存儲它。您可以重設此密碼，但是無法擷取此密碼。

- 建立目錄控制器的安全群組。

啟動時 AWS Managed Microsoft AD，AWS 會建立包含所有目錄物件的組織單位 (OU)。此 OU 有您在建立目錄時所輸入的 NetBIOS 名稱，且位於根網域中。網域根目錄擁有及管理 AWS。

使用您 AWS Managed Microsoft AD 目錄建立的 Admin 帳戶具有 OU 最常見系統管理活動的權限：

- 建立、更新或刪除使用者
- 將資源 (例如檔案或列印伺服器) 新增至您的網域，然後對您 OU 中的使用者指派這些資源的許可
- 建立額外的 OU 和容器
- 委派授權
- 從 Active Directory 資源回收筒還原已刪除的物件
- 在活動目錄 Web 服務上運行 AD 和 DNS 視窗 PowerShell 模塊

管理員帳戶也有權執行下列全網域活動：

- 管理 DNS 組態 (新增、移除或更新記錄、區域和轉寄站)
- 檢視 DNS 事件日誌
- 檢視安全事件日誌

若要建立目錄 AWS Managed Microsoft AD

1. 請登入 AWS Management Console 並開啟 AWS Directory Service 主控台，網址為 <https://console.aws.amazon.com/directoryservicev2/>。
2. 在導覽窗格中，選擇 Directories (目錄)，然後選擇 Set up directory (設定目錄)。
3. 選擇 AWS Managed Microsoft AD。AWS Managed Microsoft AD 是您目前可以與 Amazon RDS 一起使用的唯一選項。
4. 輸入下列資訊：

目錄 DNS 名稱

目錄的完全合格名稱，例如 **corp.example.com**。

目錄 NetBIOS 名稱

目錄的簡短名稱，例如：**CORP**。

目錄描述

(選用) 目錄的描述。

管理員密碼

目錄管理員的密碼。目錄建立程序會建立含有使用者名稱 Admin 與這組密碼的管理員帳戶。

目錄管理員密碼不得包含 "admin" 字組。密碼區分大小寫，長度須為 8 至 64 個字元。至少須有一位字元屬於以下四種類型中的三類：

- 小寫字母 (a–z)
- 大寫字母 (A–Z)
- 數字 (0–9)
- 非英數字元 (~!@#\$%^&* _-+=`|\(){}[];'"<>.,?/)

Confirm password (確認密碼)

重新輸入的管理員密碼。

5. 選擇下一步。
6. 在 Networking (聯網) 區段輸入以下資訊，然後選擇 Next (下一步)。

VPC

目錄的 VPC。在相同的 VPC 中建立 MySQL 資料庫執行個體。

子網路

目錄伺服器的子網路。這兩個子網路必須位於不同的可用區域。

7. 檢閱目錄資訊，並進行必要的變更。若資訊無誤，請選擇 Create directory (建立目錄)。

Review & create

Review

Directory type Microsoft AD	VPC vpc-8b6b78e9 ()
Directory DNS name corp.example.com	Subnets subnet-75128d10 (, us-east-1a) subnet-f51665dd (, us-east-1b)
Directory NetBIOS name CORP	
Directory description My directory	

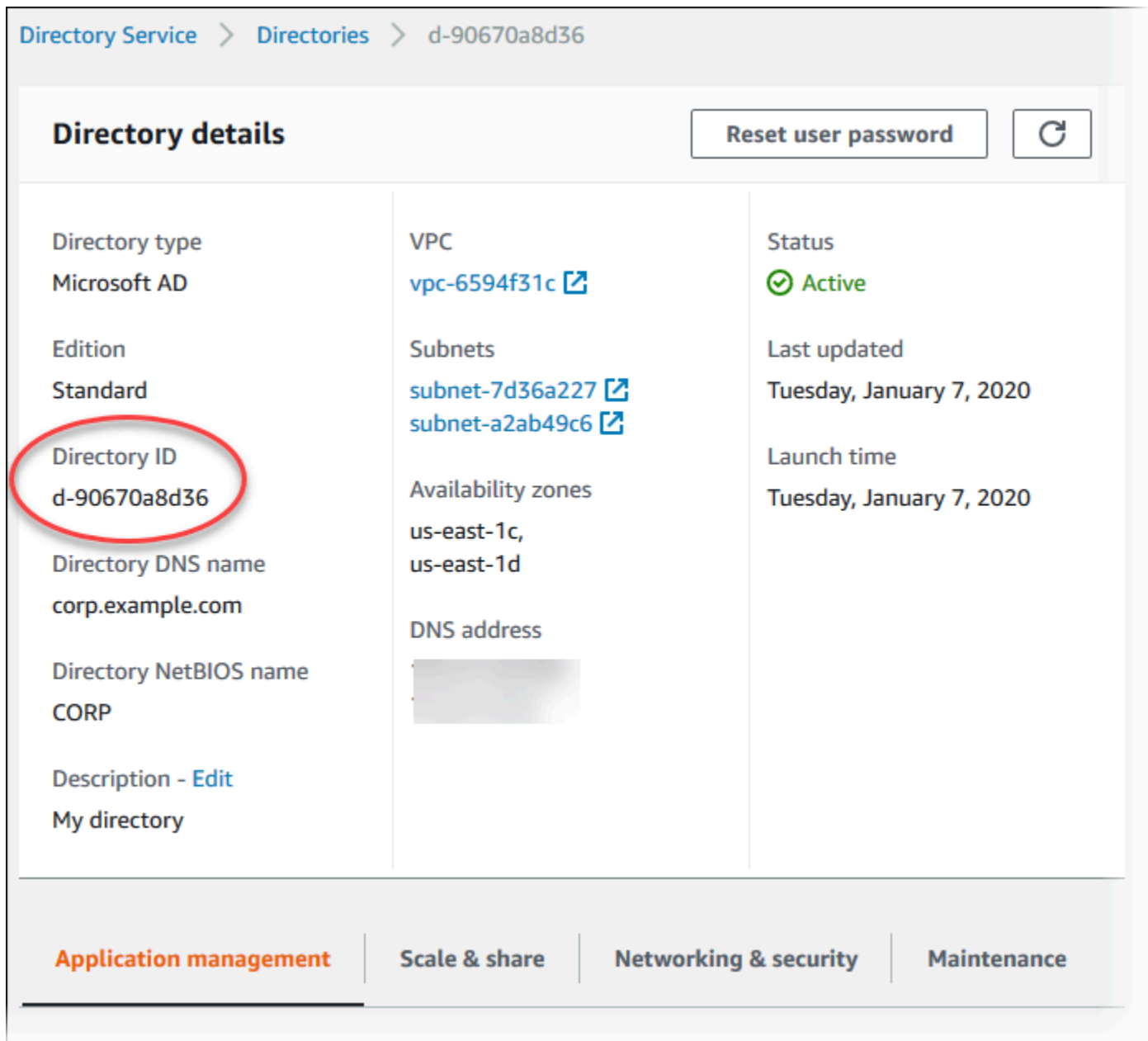
Pricing

Edition Standard	Free trial eligible Learn more 30-day limited trial
~USD () *	
* Includes two domain controllers, USD ()/mo for each additional domain controller.	

Cancel Previous **Create directory**


建立目錄需要幾分鐘的時間。成功建立時，Status (狀態) 值會變更為 Active (作用中)。


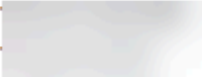
如要查看您目錄的資訊，請在目錄清單中選擇目錄名稱。請記下 Directory ID (目錄 ID)，因為您在建立或修改 MySQL 資料庫執行個體時將需要這個值。



Directory Service > Directories > d-90670a8d36

Directory details

[Reset user password](#) 

Directory type	VPC	Status
Microsoft AD	vpc-6594f31c	 Active
Edition	Subnets	Last updated
Standard	subnet-7d36a227 subnet-a2ab49c6	Tuesday, January 7, 2020
Directory ID d-90670a8d36	Availability zones	Launch time
Directory DNS name	us-east-1c, us-east-1d	Tuesday, January 7, 2020
Directory NetBIOS name	DNS address	
CORP		
Description - Edit		
My directory		


[Application management](#) | [Scale & share](#) | [Networking & security](#) | [Maintenance](#)

步驟 2：建立供 Amazon RDS 使用的 IAM 角色

若要讓 Amazon RDS AWS Directory Service 為您撥打電話，則需要使用受管身分與存取權管理政策 AmazonRDSDirectoryServiceAccess 的 IAM 角色。此角色允許 Amazon RDS 呼叫 AWS Directory Service。

使用建立資料庫執行個體 AWS Management Console 且主控台使用者具有 iam:CreateRole 權限時，主控台會自動建立此角色。在此情況下，角色名稱為 rds-directoryservice-kerberos-access-role。否則，您必須手動建立 IAM 角色。建立此 IAM 角色時 Directory Service，請選擇 AWS 受管政策並將其附加 AmazonRDSDirectoryServiceAccess 到該角色。

如需為服務建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 AWS 服務](#)。

 Note

用於 Windows Authentication for RDS for SQL Server 的 IAM 角色不可用於 RDS for MySQL。

您可以選擇性地建立具有必要許可的政策，而不是使用受管 IAM 政策 AmazonRDSDirectoryServiceAccess。在此情況下，IAM 角色必須有以下 IAM 信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

此角色也須具有下列 IAM 角色政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    }
  ]
}
```



```
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

步驟 3：建立和設定使用者

您可以使用 Active Directory 使用者和運算集區來建立使用者。這個工具是 Active Directory Domain Services 和 Active Directory 輕量型目錄服務工具的一部分。使用者代表具有目錄存取權的個人或實體。

若要在 AWS Directory Service 目錄中建立使用者，您必須連線至以 Microsoft 視窗為基礎的 Amazon EC2 執行個體。此執行處理必須是 AWS Directory Service 目錄的成員，並以具有建立使用者權限的使用者身分登入。如需詳細資訊，請參閱《AWS Managed Microsoft AD Directory Service 管理指南》中的[管理AWS 中的使用者和群組](#)。

步驟 4：建立或修改 MySQL 資料庫執行個體

建立或修改 MySQL 資料庫執行個體以搭配您的目錄使用。您可以使用主控台、CLI 或 RDS API，將資料庫執行個體與目錄建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) CLI 命令或建立資料庫執行個體 RDS API [作業](#)，[建立新的 MySQL 資料庫執行個體](#)。

如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[modify-db-instance](#) CLI 命令或修改資料庫執行個體 RDS API 作業來[修改現有的 MySQL 資料庫執行個體](#)。

如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[restore-db-instance-from-db 快照 CLI 命令](#)或還原資料庫快照 RDS API 作業，[從資料庫快照還原 MySQL 資料庫執行個體](#)。[InstanceFrom](#)

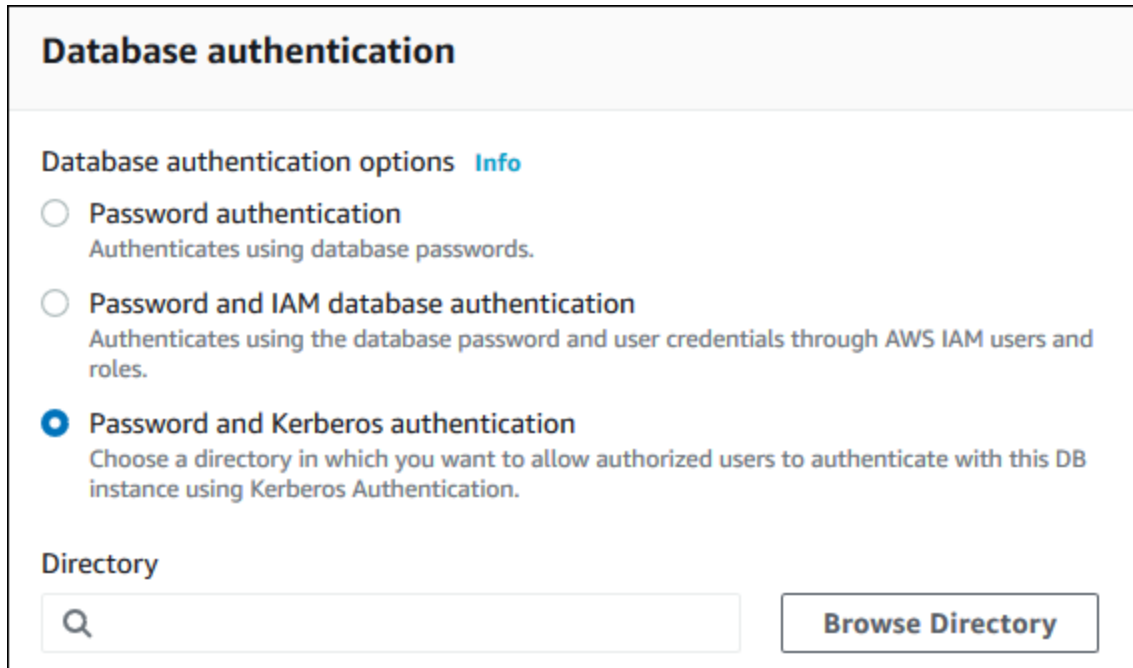
如需說明，請參閱[從資料庫 快照還原](#)。

- [point-in-time](#) 使用主控台、[restore-db-instance-to-point-in-time](#) CLI 命令或恢復 InstanceToPointInTime RDS API 作業，將 MySQL 資料庫執行個體[還原到](#)。

如需說明，請參閱「[將資料庫執行個體還原至指定的時間](#)」。

僅有 VPC 中的 MySQL 資料庫執行個體支援 Kerberos 身分驗證。資料庫執行個體可在與目錄相同的 VPC 中，或在不同 VPC 中。資料庫執行個體必須使用允許目錄 VPC 內傳入的安全群組，如此資料庫執行個體才能與目錄通訊。

使用主控台建立、修改或還原資料庫執行個體時，請在 Database authentication (資料庫身分驗證) 區段中選擇 Password and Kerberos authentication (密碼和 Kerberos 身分驗證)。選擇 Browse Directory (瀏覽目錄) 並選取目錄，或是選擇 Create a new directory (建立新目錄)。



Database authentication

Database authentication options [Info](#)

- Password authentication
Authenticates using database passwords.
- Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Directory

使用 AWS CLI 或 RDS API 時，請將資料庫執行個體與目錄建立關聯。資料庫執行個體需要下列參數，才能使用您建立的網域目錄：

- 針對 `--domain` 參數，使用您建立目錄時產生的網域識別符 ("d-*" 識別符)。
- 針對 `--domain-iam-role-name` 參數，使用您建立的規則，其會使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess`。

例如，下列 CLI 命令會修改資料庫執行個體來使用目錄。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --domain d-ID \  
  --domain-iam-role-name role-name
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --domain d-ID ^
  --domain-iam-role-name role-name
```

Important

如果您修改資料庫執行個體，以啟用 Kerberos 身分驗證，請在進行變更後重新啟動資料庫執行個體。

步驟 5：建立 Kerberos 身分驗證 MySQL 登入

使用 Amazon RDS 主要使用者登入資料來連線到 MySQL 資料庫執行個體，方法與其他任何的資料庫執行個體相同。資料庫執行個體已加入 AWS Managed Microsoft AD 網域。因此，您可以從您網域中的 Active Directory 使用者佈建 MySQL 登入和使用者。資料庫許可會透過標準 MySQL 許可管理，並從這些登入授予及撤銷。

您可以允許 Active Directory 使用者與 MySQL 進行身分驗證。若要執行此作業，請先使用 Amazon RDS 主要使用者登入資料連線到 MySQL 資料庫執行個體，方法與其他任何的資料庫執行個體相同。登入後，透過執行下列命令，在 MySQL 中使用 PAM (插入式驗證模組) 建立外部驗證的使用者。將 *testuser* 取代成使用者名稱。

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
```

來自您網域的使用者 (人員和應用程式兩者) 現在可以使用 Kerberos 身分驗證從加入網域的用戶端機器連線至資料庫執行個體。

Important

強烈建議用戶端在使用 PAM 身分驗證時使用 SSL/TLS 連線。如果他們未使用 SSL/TLS 連線，在某些情況下，密碼可能會以純文字形式傳送。若要為 AD 使用者需要 SSL/TLS 加密連線，請執行下列命令，並以使用者名稱取代 *testuser*：

```
ALTER USER 'testuser'@'%' REQUIRE SSL;
```

如需詳細資訊，請參閱 [將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)。

管理網域中的資料庫執行個體

您可以使用 CLI 或 RDS API 來管理資料庫執行個體，以及其與受管 Active Directory 的關係。例如，您可以與適用於 Kerberos 身分驗證的 Active Directory 建立關聯，也可以解除關聯 Active Directory 來停用 Kerberos 身分驗證。您也可以將要由某個 Active Directory 於外部進行身分識別的資料庫執行個體移至另一個 Active Directory。

例如，使用 Amazon RDS API，您可以執行下列動作：

- 如要重新嘗試為失敗的成員資格啟用 Kerberos 身分驗證，請使用 `ModifyDBInstance` API 操作並指定目前成員資格的目錄 ID。
- 如要更新成員資格的 IAM 角色名稱，請使用 `ModifyDBInstance` API 操作並指定目前成員資格的目錄 ID，以及新的 IAM 角色。
- 停用資料庫執行個體上的 Kerberos 身分驗證，使用 `ModifyDBInstance` API 操作並指定 `none` 做為網域參數。
- 如要在網域之間移動資料庫執行個體，請使用 `ModifyDBInstance` API 操作，並指定新網域的網域識別符做為網域參數。
- 如要列出每個資料庫執行個體的成員資格，請使用 `DescribeDBInstances` API 操作。

了解網域成員資格

在您建立或修改資料庫執行個體之後，該執行個體會成為網域的成員。您可以執行 [describe-db-instances](#) CLI 命令，檢視資料庫執行個體的網域成員資格狀態。資料庫執行個體的状态可以是下列其中一個：

- `kerberos-enabled` – 資料庫執行個體已啟用 Kerberos 身分驗證。
- `enabling-kerberos`— AWS 正在啟用此資料庫執行個體上的 Kerberos 驗證程序。
- `pending-enable-kerberos` – 啟用 Kerberos 身分驗證的過程正在此資料庫執行個體上遭到擱置。
- `pending-maintenance-enable-kerberos`— AWS 將嘗試在下一個排定的維護時段期間在資料庫執行個體上啟用 Kerberos 驗證。
- `pending-disable-kerberos` – 停用 Kerberos 身分驗證的過程正在此資料庫執行個體上遭到擱置。

- `pending-maintenance-disable-kerberos`— AWS 將嘗試在下一個排定的維護時段期間停用資料庫執行個體上的 Kerberos 驗證。
- `enable-kerberos-failed` – 有一個組態問題已禁止 AWS 在資料庫執行個體上啟用 Kerberos 身分驗證。請在重新發出資料庫執行個體修改命令之前檢查並修正您的組態。
- `disabling-kerberos`— AWS 正在停用此資料庫執行個體上的 Kerberos 驗證程序。

由於網路連線問題或 IAM 角色不正確，請求啟用 Kerberos 身分驗證可能失敗。例如，假設您建立了資料庫執行個體或修改了現有的資料庫執行個體，並且嘗試啟用 Kerberos 身分驗證失敗。如果發生這種情況，請重新發出修改命令，或修改新建立的資料庫執行個體以加入網域。

使用 Kerberos 身分驗證連線至 MySQL

如要使用 Kerberos 身分驗證連線至 MySQL，您必須使用 Kerberos 身分驗證類型登入。

如要建立您可以使用 Kerberos 身分驗證連線的資料庫使用者，請在 IDENTIFIED WITH 陳述式上使用 CREATE USER 子句。如需說明，請參閱「[步驟 5：建立 Kerberos 身分驗證 MySQL 登入](#)」。

為了避免錯誤，請使用 MariaDB mysql 用戶端。您可以在 <https://downloads.mariadb.org/> 下載 MariaDB 軟體。

在命令提示字元中，連線至與 MySQL 資料庫執行個體相關聯的其中一個端點。遵循 [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#) 中的一般程序。當系統提示您輸入密碼時，請輸入與該使用者名稱相關聯的 Kerberos 密碼。

還原 MySQL 資料庫執行個體並新增至網域

您可以還原資料庫快照或完成 MySQL 資料庫執行個體的 point-in-time 還原，然後將其新增至網域。還原資料庫執行個體後，請使用 [步驟 4：建立或修改 MySQL 資料庫執行個體](#) 中說明的程序來修改資料庫執行個體，將資料庫執行個體新增到網域。

Kerberos 身分驗證 MySQL 限制

下列限制適用於 MySQL 的 Kerberos 身分驗證：

- 僅支援。AWS Managed Microsoft AD 不過，您可以將 RDS for MySQL 資料庫執行個體加入至相同 AWS 區域中，不同帳戶所擁有的共用受管 Microsoft AD 網域。
- 啟用功能後，您必須重新啟動資料庫執行個體。
- 網域名稱長度不能超過 61 個字元。

- 您無法同時啟用 Kerberos 身分驗證和 IAM 身分驗證。為您的 MySQL 資料庫執行個體選擇一種身分驗證方法或另一種。
- 啟用功能後，請勿修改資料庫執行個體連接埠。
- 請勿搭配僅供讀取複本使用 Kerberos 身分驗證。
- 如果您已針對使用 Kerberos 驗證的 MySQL 資料庫執行個體開啟自動次要版本升級，則必須關閉 Kerberos 驗證，然後在自動升級之後重新開啟該執行個體。如需自動次要版本升級的詳細資訊，請參閱 [MySQL 的自動次要版本升級](#)。
- 如要在啟用此功能的情況下刪除資料庫執行個體，請先停用功能。若要執行此作業，請針對資料庫執行個體使用 `modify-db-instance` CLI 命令，然後為 `none` 參數指定 `--domain`。

如果您使用 CLI 或 RDS API 來刪除啟用此功能的資料庫執行個體，則預期會有延遲。

- 您無法在內部部署或自我託管的 Microsoft Active Directory 與 AWS Managed Microsoft AD 之間設定樹系信任關係。

使用 Amazon RDS Optimized Reads 改善 RDS for MySQL 的查詢效能

您可以使用 Amazon RDS Optimized Reads，為 RDS for MySQL 實現更快的查詢處理。使用 RDS Optimized Reads 的 RDS for MySQL 資料庫執行個體或多可用區域資料庫叢集，相較於不使用它的情況，查詢處理速度最高可以快上 2 倍。

主題

- [RDS Optimized Reads 的概觀](#)
- [RDS Optimized Reads 的使用案例](#)
- [RDS Optimized Reads 的最佳實務](#)
- [使用 RDS Optimized Reads](#)
- [監控使用 RDS Optimized Reads 的資料庫執行個體](#)
- [RDS Optimized Reads 的限制](#)

RDS Optimized Reads 的概觀

當您使用已開啟 RDS Optimized Reads 的 RDS for MySQL 資料庫執行個體或多可用區域資料庫叢集時，其可透過使用執行個體儲存體實現更快的查詢效能。執行個體儲存體為資料庫執行個體或多可用區域資料庫叢集提供暫時的區塊層級儲存空間。儲存體位於實際連接至主機伺服器的非揮發性記憶體儲存裝置 (NVMe) 固態硬碟 (SSD) 中。此儲存體已針對低延遲、高隨機 I/O 效能，以及高循序讀取輸送量進行最佳化。

當資料庫執行個體或多可用區域資料庫叢集使用資料庫執行個體類別搭配執行個體儲存體 (例如 db.m5d 或 db.m6gd) 時，預設會開啟 RDS Optimized Reads。使用 RDS Optimized Reads，部分臨時物件會存放在執行個體儲存體上。這些暫存物件包括內部暫存檔、內部磁碟上暫存資料表、記憶體對應檔案，以及二進位日誌 (binlog) 快取檔案。如需執行個體儲存體的詳細資訊，請參閱《適用於 Linux 執行個體的 Amazon Elastic Compute Cloud 使用者指南》中的 [Amazon EC2 執行個體儲存體](#)。

在 MySQL 中產生臨時物件以進行查詢處理的工作負載，可以利用執行個體儲存體來加快查詢處理速度。這種類型的工作負載包括涉及排序、雜湊彙總、高負載聯結、一般資料表運算式 (CTE) 的查詢，以及未編製索引之資料欄上的查詢。無論用於持久性 Amazon EBS 儲存體的儲存體組態為何，這些執行個體儲存體都會提供更高的 IOPS 和效能。因為 RDS Optimized Reads 將臨時物件上的操作卸載至執行個體儲存體，所以持久性儲存體 (Amazon EBS) 的每秒讀寫次數 (IOPS) 或輸送量現在可以用於持久性物件上的操作。這些操作包括一般資料檔案讀取和寫入，以及背景引擎操作，例如排清和插入緩衝區合併。

Note

手動和自動 RDS 快照都只包含持久性物件的引擎檔案。在執行個體儲存體中建立的臨時物件不會包含在 RDS 快照集中。

RDS Optimized Reads 的使用案例

如果您的工作負載嚴重依賴臨時物件 (例如內部資料表或檔案) 進行其查詢執行，則您可以從開啟 RDS Optimized Reads 中獲益。下列使用案例為 RDS Optimized Reads 的候選項目：

- 使用複雜的一般資料表運算式 (CTE)、衍生資料表和群組操作來執行分析查詢的應用程式
- 使用未最佳化查詢提供大量讀取流量的僅供讀取複本
- 執行涉及複雜作業之隨需或動態報告查詢的應用程式，例如搭配 GROUP BY 和 ORDER BY 子句的查詢
- 使用內部暫存資料表進行查詢處理的工作負載

您可以監控引擎狀態變數 `created_tmp_disk_tables`，以判斷在資料庫執行個體上建立的磁碟型暫存資料表數目。

- 直接或在程序中建立大型暫存資料表以儲存中繼結果的應用程式
- 對未編製索引之資料欄執行分組或排序的資料庫查詢

RDS Optimized Reads 的最佳實務

請使用 RDS Optimized Reads 的下列最佳實務：

- 針對唯讀查詢新增重試邏輯，以防這些查詢在執行期間由於執行個體儲存體已滿而失敗。
- 使用 CloudWatch 指標 `FreeLocalStorage` 監控執行個體儲存體上的可用儲存空間。如果執行個體儲存體由於資料庫執行個體的工作負載而達到其限制，請修改資料庫執行個體，以使用較大的資料庫執行個體類別。
- 當您的資料庫執行個體有足夠的記憶體，但仍達到執行個體儲存體或多可用區域資料庫叢集的儲存限制時，請增加 `binlog_cache_size` 值以在記憶體中維護工作階段特定的 binlog 項目。此組態可防止將 binlog 項目寫入至磁碟上的暫存 binlog 快照檔案。

`binlog_cache_size` 參數是工作階段特有的。您可以變更每個新工作階段的值。此參數的設定可以在尖峰工作負載期間增加資料庫執行個體的記憶體使用率。因此，請考慮根據您應用程式的工作負載模式和資料庫執行個體上的可用記憶體來增加參數值。

- 將 MIXED 的預設值用於 `binlog_format`。根據交易的大小，將 `binlog_format` 設為 ROW 可能會在執行個體儲存體上產生大型 binlog 快取檔案。
- 將 [internal tmp mem storage engine](#) 參數設為 TempTable，然後設定 [temptable_max_mmap](#) 參數，以符合執行個體儲存體上可用儲存體的大小。
- 避免在單一交易中執行大量變更。這些類型的交易可能會在執行個體儲存體上產生大型 binlog 快取檔案，並在執行個體儲存體已滿時產生問題。請考慮將寫入分割成多個小型交易，以將 binlog 快取檔案使用的儲存體減至最少。
- 將 ABORT_SERVER 的預設值用於 `binlog_error_action` 參數。這樣做可避免在啟用備份的資料庫執行個體上進行二進位記錄時發生問題。

使用 RDS Optimized Reads

當您在單一可用區域資料庫執行個體部署、多可用區域資料庫執行個體部署或多可用區域資料庫叢集部署中，佈建具有下列其中一個資料庫執行個體類別的 RDS for MySQL 資料庫執行個體時，資料庫執行個體會自動使用 RDS Optimized Reads。

若要開啟 RDS Optimized Reads，請執行下列其中一項：

- 使用其中一個資料庫執行個體類別，建立 RDS for MySQL 資料庫執行個體或多可用區域資料庫叢集。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改現有的 RDS for MySQL 資料庫執行個體或多可用區域資料庫叢集，以使用其中一個資料庫執行個體類別。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

RDS Optimized Reads 適用於支援其中一或多個具有本機 NVMe SSD 儲存體的資料庫執行個體類別之所有 AWS 區域 RDS。如需資料庫執行個體類別的相關資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

資料庫執行個體類別的可用性因 AWS 區域而異。若要判斷特定的 AWS 區域是否支援資料庫執行個體類別，請參閱 [the section called “確定資料庫執行個體類別支援 AWS 區域”](#)。

如果您不想要使用 RDS Optimized Reads，請修改資料庫執行個體或多可用區域資料庫叢集，以便其不使用支援該功能的資料庫執行個體類別。

監控使用 RDS Optimized Reads 的資料庫執行個體

您可以透過下列 CloudWatch 指標，監控使用 RDS Optimized Reads 的資料庫執行個體：

- FreeLocalStorage
- ReadIOPSLocalStorage
- ReadLatencyLocalStorage
- ReadThroughputLocalStorage
- WriteIOPSLocalStorage
- WriteLatencyLocalStorage
- WriteThroughputLocalStorage

這些指標提供可用執行個體儲存體、IOPS 和輸送量的相關資料。如需這些指標的詳細資訊，請參閱 [Amazon RDS 的 CloudWatch Amazon 實例級指標](#)。

RDS Optimized Reads 的限制

下列限制適用於 RDS Optimized Reads：

- RDS for MySQL 8.0.28 版和更新版本支援 RDS Optimized Reads。如需 RDS for MySQL 版本的相關資訊，請參閱 [MySQL on Amazon RDS 版本](#)。
- 您無法在支援 RDS Optimized Reads 的資料庫執行個體類別上，將臨時物件的位置變更為持久性儲存體 (Amazon EBS)。
- 在資料庫執行個體上啟用二進位記錄時，交易大小上限受執行個體儲存體的大小限制。在 MySQL 中，任何需要儲存體超過 `binlog_cache_size` 寫入交易值的工作階段，都會變更為暫存 binlog 快取檔案，這些檔案是在執行個體儲存體上建立的。
- 執行個體儲存體已滿時，交易會失敗。

使用 RDS Optimized Writes 改善 MySQL 的寫入效能

您可以使用 RDS Optimized Writes for MySQL 來改善寫入交易的效能。當您的 RDS for MySQL 資料庫使用 RDS Optimized Writes 時，它可以實現高達兩倍的寫入交易輸送量。

主題

- [RDS Optimized Writes 的概觀](#)
- [使用 RDS Optimized Writes](#)
- [在現有資料庫上啟用 RDS Optimized Writes](#)
- [RDS Optimized Writes 的限制](#)

RDS Optimized Writes 的概觀

當您開啟 RDS Optimized Writes 時，您的 RDS for MySQL 資料庫只會在將資料排清至持久性儲存體時寫入一次，而不需要雙寫緩衝區。資料庫會繼續為可靠的資料庫交易提供 ACID 屬性保護，以及改善的效能。

關聯式資料庫 (例如 MySQL) 為可靠的資料庫交易提供不可分割性、一致性、隔離和耐久性的 ACID 屬性。為了協助提供這些屬性，MySQL 會使用稱為雙寫緩衝區的資料儲存區域，防止部分頁面寫入錯誤。當資料庫正在更新頁面時發生硬體故障 (例如停電) 時，就會發生這些錯誤。MySQL 資料庫可以偵測部分頁面寫入，並使用雙寫緩衝區中的頁面複本進行復原。雖然這項技術可提供保護，但也會產生額外的寫入操作。如需 MySQL 雙寫緩衝區的詳細資訊，請參閱 MySQL 文件中的[雙寫緩衝區](#)。

開啟 RDS Optimized Reads 後，您的 RDS for MySQL 資料庫只會在將資料排清至持久性儲存體時寫入一次，而不需使用雙寫緩衝區。如果您在 RDS for MySQL 資料庫上執行大量寫入工作負載，則 RDS Optimized Reads 很有用。具有大量寫入工作負載的資料庫範例包括支援數位支付、金融交易和遊戲應用程式的資料庫。

這些資料庫會在使用 AWS Nitro System 的資料庫執行個體類別上執行。由於這些系統中的硬體組態，資料庫可以在一個步驟中可靠且持久地將 16 KiB 頁面直接寫入至資料檔案。AWS Nitro System 讓 RDS Optimized Writes 變成可行。

您可以設定新的資料庫參數 `rds.optimized_writes`，以控制 RDS for MySQL 資料庫的 RDS Optimized Writes 功能。在 RDS for MySQL 8.0 版的資料庫參數群組中存取此參數。請使用下列值設定參數：

- `AUTO` – 開啟 RDS Optimized Writes (如果資料庫支援它)。關閉 RDS Optimized Writes (如果資料庫不支援它)。此設定是預設值。

- OFF – 關閉 RDS Optimized Writes，即使資料庫支援它也一樣。

如果您的現有資料庫具有不支援 RDS Optimized Writes 的引擎版本、資料庫執行個體類別和/或檔案系統格式，則您可以透過建立藍/綠部署來啟用此功能。如需詳細資訊，請參閱[the section called “在現有的資料庫上啟用”](#)。

如果您將設定為使用 RDS Optimized Writes 的 RDS for MySQL 資料庫遷移至不支援此功能的資料庫執行個體類別，RDS 會自動關閉資料庫的 RDS Optimized Writes。

當 RDS Optimized Writes 關閉時，資料庫會使用 MySQL 雙寫緩衝區。

若要判斷 RDS for MySQL 資料庫是否使用 RDS Optimized Writes，請檢視資料庫 `innodb_doublewrite` 參數目前的值。如果資料庫使用 RDS Optimized Writes，則此參數會設為 FALSE (0)。

使用 RDS Optimized Writes

當您使用 RDS 主控台、AWS CLI 或 RDS API，建立 RDS for MySQL 資料庫時，您可以開啟 RDS Optimized Writes。在資料庫建立期間滿足下列兩個條件時，RDS Optimized Writes 會自動開啟：

- 您指定一個支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別。
 - RDS for MySQL 8.0.30 版和更新版本支援 RDS Optimized Writes。如需 RDS for MySQL 版本的相關資訊，請參閱 [MySQL on Amazon RDS 版本](#)。
 - 使用下列資料庫執行個體類別的 RDS for MySQL 資料庫支援 RDS Optimized Writes：
 - db.m7g
 - db.m6g
 - DB.m6gd
 - db.m6i
 - db.m5
 - db.m5d
 - db.r7g
 - db.r6g
 - db.r6gd
 - db.r6i
 - db.r5
 - db.r5b

- db.r5d
- db.x2idn
- db.x2iedn

如需資料庫執行個體類別的相關資訊，請參閱 [the section called “資料庫執行個體類別”](#)。

資料庫執行個體類別的可用性因 AWS 區域而異。若要判斷特定的 AWS 區域 是否支援資料庫執行個體類別，請參閱 [the section called “確定資料庫執行個體類別支援 AWS 區域”](#)。

若要將資料庫升級為支援 RDS Optimized Writes 的資料庫執行個體類別，您可以建立藍/綠部署。如需詳細資訊，請參閱 [the section called “在現有的資料庫上啟用”](#)。

- 在與資料庫相關聯的參數群組中，`rds.optimized_writes` 參數設為 `AUTO`。在預設參數群組中，此參數一律設為 `AUTO`。

如果您想要使用支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別，但不想要使用此功能，則請在建立資料庫時指定自訂參數群組。在此參數群組中，將 `rds.optimized_writes` 參數設為 `OFF`。如果您想要資料庫稍後使用 RDS Optimized Writes，則可以將參數設為 `AUTO` 將其開啟。如需建立自訂參數群組和設定參數的相關資訊，請參閱 [使用參數群組](#)。

如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。









主控台

使用 RDS 主控台建立 RDS for MySQL 資料庫時，您可以篩選支援 RDS Optimized Writes 的資料庫引擎版本和資料庫執行個體類別。開啟篩選條件後，您可以從可用的資料庫引擎版本和資料庫執行個體類別中進行選擇。

若要選擇支援 RDS 最佳化寫入的資料庫引擎版本，請在 Engine version (引擎版本) 中篩選支援它的 RDS for MySQL 資料庫引擎版本，然後選擇版本。

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input checked="" type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

Edition

MySQL Community

Known issues/limitations
 Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Engine version [Info](#)
 View the engine versions that support the following database features.


▼ Hide filters

Show versions that support the Multi-AZ DB cluster [Info](#)
 Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)
 Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

在 Instance configuration (執行個體組態) 區段中，篩選支援 RDS Optimized Writes 的資料庫執行個體類別，然後選擇資料庫執行個體類別。



Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

Amazon RDS Optimized Writes - new [Info](#)
 Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)
 Memory optimized classes (includes r and x classes)

db.r5b.large (supports Amazon RDS Optimized Writes)
2 vCPUs 16 GiB RAM Network: 10,000 Mbps

Include previous generation classes

在做出這些選擇之後，您可以選擇其他符合您需求的設定，並使用主控台完成建立 RDS for MySQL 資料庫。

AWS CLI

若要使用建立資料庫執行個體AWS CLI，請使用[create-db-instance](#)指令。確定 `--engine-version` 和 `--db-instance-class` 值支援 RDS Optimized Writes。此外，確定與資料庫執行個體相關聯的參數群組已將 `rds.optimized_writes` 參數設為 `AUTO`：此範例會將預設參數群組與資料庫執行個體建立關聯。

Example 建立使用 RDS Optimized Writes 的資料庫執行個體

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --engine mysql \  
  --engine-version 8.0.30 \  
  --db-instance-class db.r5b.large \  
  --manage-master-user-password \  
  --master-username admin \  
  --allocated-storage 200
```

在Windows中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --engine mysql ^  
  --engine-version 8.0.30 ^  
  --db-instance-class db.r5b.large ^
```



```
--manage-master-user-password ^  
--master-username admin ^  
--allocated-storage 200
```

RDS API

您可以使用 [CreateDBInstance](#) 操作，建立資料庫執行個體。使用此操作時，確定 `EngineVersion` 和 `DBInstanceClass` 值支援 RDS Optimized Writes。此外，確定與資料庫執行個體相關聯的參數群組已將 `rds.optimized_writes` 參數設為 `AUTO`：

在現有資料庫上啟用 RDS Optimized Writes

為了修改現有的 RDS for MySQL，以開啟 RDS Optimized Writes，必須已使用支援的資料庫引擎版本和資料庫執行個體類別建立資料庫。此外，必須在 2022 年 11 月 27 日發行 RDS Optimized Writes 之後建立了資料庫，因為所需的基礎檔案系統組態與發行之前所建立之資料庫的組態不相容。如果符合這些條件，您可以透過將 `rds.optimized_writes` 參數設定為 `AUTO` 來開啟 RDS Optimized Writes。

如果未使用支援的引擎版本、執行個體類別或檔案系統組態，來建立您的資料庫，您可以使用 RDS 藍/綠部署遷移至支援的組態。建立藍/綠部署時，請執行以下動作：

- 選取在綠色資料庫上啟用 Optimized Writes，然後指定支援 RDS Optimized Writes 的引擎版本和資料庫執行個體。如需受支援引擎版本和執行個體類別的清單，請參閱 [使用 RDS Optimized Writes](#)。
- 在儲存體下，選擇升級儲存體檔案系統組態。此選項會將資料庫升級為相容的基礎檔案系統組態。

建立藍/綠部署時，如果將 `rds.optimized_writes` 參數設定為 `AUTO`，則會在綠色環境上自動啟用 RDS Optimized Writes。然後，您可以轉換藍/綠部署，將綠色環境提升至新的生產環境。

如需詳細資訊，請參閱 [the section called “建立藍/綠部署”](#)。

RDS Optimized Writes 的限制

當您從快照還原 RDS for MySQL 資料庫時，只有在下列所有條件都滿足時，才能開啟資料庫的 RDS Optimized Writes：

- 快照集是從支援 RDS Optimized Writes 的資料庫建立的。
- 快照是從發佈 RDS Optimized Writes 之後所建立的資料庫中建立的。
- 快照還原至支援 RDS Optimized Writes 的資料庫。
- 還原的資料庫與將 `rds.optimized_writes` 參數設為 `AUTO` 的參數群組相關聯。

升級 MySQL 資料庫引擎

當 Amazon RDS 支援新版本的資料庫引擎時，您可以將資料庫執行個體升級為新版本。MySQL 資料庫有兩種升級：主要版本升級和次要版本升級。

主要版本升級

主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。因此，您必須手動執行資料庫執行個體的主要版本升級。您可以修改資料庫執行個體，啟動主要版本升級。在您執行主要版本升級之前，建議您遵循中的指示[MySQL 的主要版本升級](#)。

對於異地同步備份資料庫執行個體部署的主要版本升級，Amazon RDS 會同時升級主要和備用副本。在升級完成之前，您的資料庫執行個體將無法使用。目前，Amazon RDS 不支援異地同步備份資料庫叢集部署的主要版本升級。

Tip

您可以使用藍/綠部署，將主要版本升級所需的停機時間降至最低。如需詳細資訊，請參閱[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

次要版本升級

次要版本升級僅包含與現有應用程式向後相容的變更。您可以修改資料庫執行個體，手動啟動次要版本升級。或者，您可以在建立或修改資料庫執行個體時啟用自動次要版本升級選項。這樣做表示 Amazon RDS 會在測試和核准新版本後自動升級您的資料庫執行個體。如需執行升級的相關資訊，請參閱[升級資料庫執行個體引擎版本](#)。

當您執行異地同步備份資料庫叢集的次要版本升級時，Amazon RDS 會一次升級一個讀取器資料庫執行個體。然後，其中一個讀取器資料庫執行個體會切換為新的寫入器資料庫執行個體。然後，Amazon RDS 會升級舊的寫入器執行個體 (現在是讀取器執行個體)。

Note

異地同步備份資料庫執行個體部署次要版本升級的停機時間可能會持續數分鐘。異地同步備份資料庫叢集通常會將次要版本升級的停機時間縮短至約 35 秒。與 RDS Proxy 搭配使用時，您可以進一步將停機時間減少到一秒或更短的時間。如需詳細資訊，請參閱[使用 RDS Proxy](#)。或者，您可以使用開放原始碼資料庫代理伺服器 (例如 [ProxSQL](#)) 或 MySQL 的 [AWS JDBC 驅動程式](#)。[PgBouncer](#)

如果您的 MySQL 資料庫執行個體使用僅供讀取複本，則必須先升級所有僅供讀取複本，才能升級來源執行個體。

主題

- [升級概觀](#)
- [MySQL 本號碼](#)
- [RDS 版本編號](#)
- [MySQL 的主要版本升級](#)
- [測試升級](#)
- [升級 MySQL 資料庫執行個體](#)
- [MySQL 的自動次要版本升級](#)
- [使用僅供讀取複本在升級 MySQL 資料庫時減少停機時間](#)

升級概觀

使用升級資料庫執行個體時，它會顯示資料庫執行個體的有效升級目標。AWS Management Console 您也可以使用下列 AWS CLI 命令來識別資料庫執行個體的有效升級目標：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine mysql \  
  --engine-version version-number \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在Windows中：

```
aws rds describe-db-engine-versions ^  
  --engine mysql ^  
  --engine-version version-number ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

例如，若要識別 MySQL 版本 8.0.28 資料庫執行個體的有效升級目標，請執行下列 AWS CLI 命令：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine mysql \  
  --engine-version 8.0.28 \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在Windows中：

```
aws rds describe-db-engine-versions ^  
  --engine mysql ^  
  --engine-version 8.0.28 ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

Amazon RDS 在升級過程中會建立兩個以上的資料庫快照。Amazon RDS 在進行任何升級變更之前，最多會建立兩個資料庫執行個體快照。如果升級對資料庫沒有用處，您可以還原其中一個快照，以建立執行舊版本的資料庫執行個體。升級完成時，Amazon RDS 會建立資料庫執行個體的另一個快照。無論是否 AWS Backup 管理資料庫執行個體的備份，Amazon RDS 都會拍攝這些快照。

Note

只在您將資料庫執行個體的備份保留期設為大於 0 的數字時，Amazon RDS 才會建立資料庫快照。若要變更備份保留期，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

升級完成之後，您就無法回復至前一版的資料庫引擎。如果您想要回到前一版，請還原第一個建立的資料庫快照，以建立新的資料庫執行個體。

您可以控制何時將資料庫執行個體升級至 Amazon RDS 支援的新版本。這一層控制協助您維持與特定資料庫版本的相容性，並且先在您的應用程式上測試新版本，再部署於生產環境。準備好時，您就可以在最適合排程的時間執行版本升級。

如果您的資料庫執行個體使用讀取複寫，則必須先升級所有僅供讀取複本，才能升級來源執行個體。

MySQL 本號碼

適用於 MySQL 資料庫引擎的版本編號順序採用主要次要修補程式的形式為年月修補程式或主要次要修補程式，例如 8.0.33.R2.20231201 或 5.7.44。使用的格式取決於 MySQL 引擎版本。如需 RDS 延伸 Support 版本編號的相關資訊，請參閱[Amazon RDS 擴展 Support 版本命名](#)。

重大的

主要版本號碼既是整數，也是版本號碼的第一個小數部分，例如 8.0。主要版本升級會增加版本號碼的主要部分。例如，從 5.7 .44 升級到 8.0.33 是主要版本升級，其中 5.7 和 8.0 是主要版本號碼。

未成年

次要版本號碼是版本號碼的第三部分，例如 8.0.33 中的 33。

補丁

修補程式是版本號碼的第四部分，例如 8.0.33.R2 中的 R2。RDS 修補程式版本中包含發行後新增到次要版本的重要錯誤修正。

年月日

日期是版本編號的第五部分，例如，在 8.0.33.R2.20231201 中的 20231201。RDS 日期版本是安全性修補程式，其中包含在發行後新增至次要版本的重要安全性修正程式。它不包括任何可能改變引擎行為的修復程序。

主要版本	次要版本	命名方案
8.0	≥ 33	<p>新的資料庫執行個體使用主要的 . 次要修補程式。例如，8.0.33.R 2.20231201。</p> <p>現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 8.0.33.R2，直到您下一次的主要或次要版本升級為止。</p>
	< 33	現有的資料庫執行個體使用主要的 .min. 修補程式，例如 8.0.32.R2。
5.7	≥ 42	<p>新的資料庫執行個體使用主要 . 次要修補程式，例如 5.7.42.R2.20231201。</p> <p>現有的資料庫執行個體可能會使用主要的 .minor.patch，例如 5.7.42.R2，直到您下次的主要或次要版本升級為止。</p>

RDS 版本編號

RDS 版本號碼使用 *major.minor.patch* 或 *major.minor.patch.YYYYMMDD* 命名配置。RDS 修補程式版本中包含發行後新增到次要版本的重要錯誤修正。RDS 日期版本 (*YYYYMMDD*) 是一個安全性修補程式。安全性修補程式不包含任何可能會變更引擎行為的修正程式。如需 RDS 延伸 Support 版本編號的相關資訊，請參閱 [Amazon RDS 擴展 Support 版本命名](#)。

若要識別資料庫的 Amazon RDS 版本編號，您必須先使用下列命令建立 `rds_tools` 延伸模組：

```
CREATE EXTENSION rds_tools;
```

您可以使用以下 SQL 查詢找出您的 RDS 版本號碼為 MySQL 資料庫：

```
mysql> select mysql.rds_version();
```

例如，查詢適用於 MySQL 8.0.34 的 RDS 資料庫會傳回下列輸出：

```
+-----+
| mysql.rds_version() |
+-----+
| 8.0.34.R2.20231201  |
+-----+
1 row in set (0.01 sec)
```

MySQL 的主要版本升級

對於 MySQL 資料庫引擎的主要版本，Amazon RDS 支援以下的就地升級：

- MySQL 5.6 至 MySQL 5.7
- MySQL 5.7 至 MySQL 8.0

Note

除了 `db.m3` 前代資料庫執行個體類別，您只能以最新一代和當代資料庫執行個體類別來建立 MySQL 5.7 版和 8.0 版資料庫執行個體。

在某些情況下，您想要將在前一代資料庫執行個體類別上執行的 MySQL 5.6 版資料庫執行個體 (`db.m3` 以外) 升級至 MySQL 5.7 版資料庫執行個體。在這些情況下，先將資料庫執行個體

修改為使用最新一代的資料庫執行個體類別。在您這麼做後，您就可以將資料庫執行個體修改為使用 MySQL 5.7 版資料庫引擎。如需 Amazon RDS 資料庫執行個體類別的相關資訊，請參閱 [資料庫執行個體類別](#)。

主題

- [MySQL 主要版本升級概觀](#)
- [升級至 MySQL 5.7 版可能較慢](#)
- [從 MySQL 5.7 升級到 8.0 的前置檢查](#)
- [從 MySQL 5.7 升級至 8.0 失敗後轉返](#)

MySQL 主要版本升級概觀

主要版本升級 可能包含與現有應用程式回溯不相容的資料庫變更。因此，Amazon RDS 不會自動套用主要版本升級；您必須手動修改資料庫執行個體。我們建議您在將任何升級套用至生產執行個體之前，先進行完整測試。

若要將 Amazon RDS 上的 MySQL 5.6 版資料庫執行個體的主要版本升級至 MySQL 5.7 版或更新版，請先執行任何可用的作業系統更新。作業系統完成更新後，請依序升級至每個主要版本：5.6 版至 5.7 版，接著是 5.7 版至 8.0 版。2014 年 4 月 24 日以前所建立的 MySQL 資料庫執行個體會顯示可用的作業系統更新，直到套用更新為止。如需作業系統更新的相關資訊，請參閱 [套用資料庫執行個體的更新](#)。

在 MySQL 的主要版本升級期間，Amazon RDS 會視需求執行 MySQL 二進位檔 `mysql_upgrade`，進而升級資料表。此外，在主要版本升級期間，Amazon RDS 還會清空 `slow_log` 與 `general_log` 資料表。若要保留日誌資訊，請在主要版本升級之前儲存日誌內容。

MySQL 主要版本升級通常在大約 10 分鐘內完成。由於資料庫執行個體類別的大小，或因為執行個體未遵循 [Amazon RDS 的最佳實務](#) 中的特定操作準則，有些升級可能會較久。如果您從 Amazon RDS 主控台升級資料庫執行個體，資料庫執行個體的狀態會指出升級何時完成。如果您使用 AWS Command Line Interface (AWS CLI) 升級，請使用 [describe-db-instances](#) 指令並檢查 Status 值。

升級至 MySQL 5.7 版可能較慢

MySQL 5.6.4 版為 `datetime`、`time` 和 `timestamp` 欄引進新的日期和時間格式，這些欄允許在日期和時間值中使用小數單位。將資料庫執行個體升級至 MySQL 5.7 版時，MySQL 會將所有日期和時間欄類型強制轉換為新格式。

因為此轉換會重建資料表，資料庫執行個體升級可能需要相當長的時間才會完成。對於執行早於 MySQL 5.6.4 版的任何資料庫執行個體，就會發生強制轉換。對於從早於 MySQL 5.6.4 版升級至 5.7 版以外之版本的任何資料庫執行個體，也會發生此轉換。

如果您的資料庫執行個體執行的版本早於 MySQL 5.6.4 版，或是從早於 5.6.4 的版本升級，我們建議您額外的步驟。在這些情況下，我們建議您將資料庫中的 `datetime`、`time` 和 `timestamp` 欄進行轉換，再將資料庫執行個體升級至 MySQL 5.7 版。此轉換作業可大幅縮短資料庫執行個體升級至 MySQL 5.7 版所需的時間。您可以對包含日期或時間欄的資料表發出 `ALTER TABLE <table_name> FORCE;` 命令，進而將日期和時間欄升級為新格式。因為改變資料表會將資料表鎖定成唯讀，建議您在維護時段執行此更新。

您可以使用下列查詢，在資料庫中尋找含有 `datetime`、`time` 或 `timestamp` 欄的所有資料表，然後對每個資料表建立 `ALTER TABLE <table_name> FORCE;` 命令。

```
SET show_old_temporals = ON;
SELECT table_schema, table_name, column_name, column_type
FROM information_schema.columns
WHERE column_type LIKE '%/* 5.5 binary format */';
SET show_old_temporals = OFF;
```

從 MySQL 5.7 升級到 8.0 的前置檢查

MySQL 8.0 與 MySQL 5.7 有許多不相容的地方。這些不相容可能會在從 MySQL 5.7 升級至 MySQL 8.0 期間發生問題。因此，您可能需要為資料庫做一些準備，才能順利完成升級。以下是這些不相容問題的一般清單：

- 資料表不能使用過時的資料類型或函數。
- 不能有孤立的 `*.frm` 檔案。
- 觸發條件不能有遺漏或空白的 `DEFINER`，或是無效的建立內容。
- 分割資料表所使用的儲存引擎皆需提供原生分割支援。
- 不能違反關鍵字或保留字的規定。MySQL 8.0 中可能會保留一些先前未保留的關鍵字。

如需詳細資訊，請參閱 MySQL 文件中的 [Keywords and Reserved Words](#) (關鍵字與保留字)。

- MySQL 5.7 `mysql` 系統資料庫中的資料表名稱不得與 MySQL 8.0 資料字典所使用的資料表名稱相同。
- 不能在 `sql_mode` 系統變數設定中定義過時的 SQL 模式。
- 資料表或預存程序的個別 `ENUM` 或 `SET` 資料欄元素長度皆不得超過 255 個字元，也不能超過 1020 位元組。

- 升級至 MySQL 8.0.13 或更新版本前，共用 InnoDB 資料表空間中不能存在資料表分割區。
- MySQL 8.0.12 或較舊版本中的查詢和預存程式皆不得對 ASC 子句使用 DESC 或 GROUP BY 限定詞。
- MySQL 5.7 安裝程式不能使用 MySQL 8.0 未支援的功能。

如需詳細資訊，請參閱 MySQL 文件中的 [Features Removed in MySQL 8.0](#) (MySQL 8.0 中移除的功能)。

- 外部索引鍵的限制條件名稱不得超過 64 個字元。
- 運用 Unicode 增強支援時，請考慮將使用 utf8mb3 字元集的物件轉換成使用 utf8mb4 字元集，因為 utf8mb3 字元集已棄用。另外，utf8mb4 目前是 utf8 字元集的別名，因此請考慮使用 utf8 做為字元集參考，而不是 utf8mb3。

如需詳細資訊，請參閱 MySQL 文件中的 [The utf8mb3 character set \(3-byte UTF-8 unicode encoding\)](#)。

當您開始從 MySQL 5.7 升級到 8.0 時，Amazon RDS 會自動執行前置檢查，以偵測這些不相容問題。如需升級至 MySQL 8.0 的相關資訊，請參閱 MySQL 文件中的 [Upgrading MySQL](#) (升級 MySQL)。

系統會強制執行這些前置檢查，您無法選擇略過這些檢查。前置檢查提供以下優勢：

- 升級期間可避免非預期的停機時間。
- 出現不相容情況時，Amazon RDS 即會防止系統進行升級，並提供相關日誌讓您了解。如此，您就可以使用這些日誌來減少不相容情況，為資料庫做好升級至 MySQL 8.0 的準備。如需移除不相容問題的詳細資訊，請參閱 MySQL 文件中的 [準備您的安裝進行升級](#)，以及 MySQL Server 部落格上的 [升級至 MySQL 8.0？這裡是您需要知道的事項...](#)。

前置檢查所含的部分檢查是隨附於 MySQL 的項目，另一些則是由 Amazon RDS 團隊特別設計。如需 MySQL 提供的前置檢查相關資訊，請參閱 [升級檢查程式公用程式](#)。

前置檢查會在系統將資料庫執行個體停止以進行升級前執行，意即前置檢查執行期間不會造成任何停機時間。如果預先檢查找到不相容，Amazon RDS 會在資料庫執行個體停止前自動取消升級。Amazon RDS 也會為不相容產生事件。如需 Amazon RDS 事件的詳細資訊，請參閱 [使用 Amazon RDS 事件通知](#)。

Amazon RDS 會在日誌檔 PrePatchCompatibility.log 中記錄每個不相容的相關詳細資訊。在多數情況下，日誌項目包含修正不相容的 MySQL 文件連結。如需檢視日誌檔案的詳細資訊，請參閱 [檢視並列出資料庫日誌檔案](#)。

根據前置檢查的特性，這些檢查作業會分析資料庫中的物件。此分析會耗用資源，並增加升級完成的時間。

Note

Amazon RDS 僅會針對從 MySQL 5.7 到 MySQL 8.0 的升級操作執行所有前置檢查。對於從 MySQL 5.6 升級到 MySQL 5.7，前置檢查僅限於確認沒有孤立表格，並且有足夠的存儲空間來重建表格。不會為升級至低於 MySQL 5.7 版本執行前置檢查。

從 MySQL 5.7 升級至 8.0 失敗後轉返

當您將資料庫執行個體從 MySQL 5.7 版升級至 MySQL 8.0 版時，升級可能會失敗。特別是，如果資料字典包含預先檢查未擷取的不相容性，則可能會失敗。在此情況下，資料庫無法在新的 MySQL 8.0 版本中成功啟動。此時，Amazon RDS 會轉返升級執行的變更。轉返後，MySQL 資料庫執行個體將執行 MySQL 5.7 版。升級失敗且轉返後，Amazon RDS 會產生具有事件 ID RDS-EVENT-0188 的事件。

通常，之所以升級失敗，是因為資料庫執行個體中的資料庫與目標 MySQL 版本之間的中繼資料存在不相容。升級失敗時，您可以在 `upgradeFailure.log` 檔案中檢視這些不相容性的詳細資訊。請先解決不相容性，然後嘗試再次升級。

在升級嘗試和轉返失敗期間，您的資料庫執行個體會重新啟動。任何擱置的參數變更都會在重新啟動期間套用，並在轉返後持續存在。

如需升級至 MySQL 8.0 的詳細資訊，請參閱 MySQL 文件中的下列主題：

- [準備進行升級的裝置](#)
- [升級至 MySQL 8.0？以下是您需要知道的内容...](#)

Note

目前，升級失敗後的自動轉返僅支援 MySQL 5.7 至 8.0 主要版本升級。

測試升級

對資料庫執行個體執行主要版本升級之前，請完整測試資料庫，了解是否與新版本相容。此外，請完整測試可存取資料庫的所有應用程式，了解是否與新版本相容。建議您採用下列程序。

測試主要版本升級

1. 檢閱新版本資料庫引擎的升級文件，了解是否有可能影響資料庫或應用程式的相容性問題：
 - [MySQL 5.6 的變更](#)
 - [MySQL 5.7 的變更](#)
 - [MySQL 8.0 的變更](#)
2. 如果資料庫執行個體是自訂資料庫參數群組的成員，請以現有設定建立新的資料庫參數群組 (與新的主要版本相容)。在升級測試執行個體時，指定新的資料庫參數群組，讓升級測試確保該參數群組可順利運作。如需建立資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。
3. 針對要升級的資料庫執行個體建立資料庫快照。如需詳細資訊，請參閱[為單一可用區資料庫執行個體建立資料庫快照](#)。
4. 還原資料庫快照，以建立新的測試資料庫執行個體。如需詳細資訊，請參閱[從資料庫快照還原](#)。
5. 使用如下詳述的其中一種方法，修改這個新的測試資料庫執行個體，以升級到新版本。如果您在步驟 2 建立新的參數群組，請指定該參數群組。
6. 評估升級的執行個體所使用的儲存體，以決定升級是否需要更多儲存體。
7. 對升級的資料庫執行個體儘可能進行越多次品質保證測試，以確保資料庫和應用程式搭配新版本可以正常運作。實作任何必要的新測試，以評估您在步驟 1 發現的任何相容性問題所帶來的影響。測試所有預存程序和函數。將應用程式的測試版本指向升級的資料庫執行個體。
8. 如果通過所有測試，就可以對生產資料庫執行個體進行升級。在確認一切都沒問題之前，建議您在資料庫執行個體上不要允許寫入操作。

升級 MySQL 資料庫執行個體

如需有關手動或自動升級 MySQL 資料庫執行個體的資訊，請參閱[升級資料庫執行個體引擎版本](#)。

MySQL 的自動次要版本升級

如果您在建立或修改資料庫執行個體時指定下列設定，則可以讓資料庫執行個體自動升級。

- 已啟用 Auto minor version upgrade (自動次要版本升級) 設定。
- Backup retention period (備份保留期間) 設定大於 0。

在中 AWS Management Console，這些設定位於 [其他組態] 下。下圖顯示 Auto minor version upgrade (自動次要版本升級) 設定。

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

Select window
 No preference

Start day: Start time: : UTC Duration: hours

如需這些設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

對於某些 RDS for MySQL 主要版本 AWS 區域，RDS 會將一個次要版本指定為自動升級版本。次要版本經過測試，並由 Amazon RDS 核准之後，在您的維護時段期間，會自動發生次要版本升級。RDS 不會自動將較新發行的次要版本設定為自動升級版本。在 RDS 指派較新的自動升級版本之前，會考慮數個準則，例如下列：

- 已知安全性問題
- MySQL 社群版本中的錯誤
- 從發行次要版本以來的整體機群穩定性

您可以使用以下 AWS CLI 命令來確定特定中指定 MySQL 次要版本的當前自動次要升級目標版本 AWS 區域。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
--engine mysql \  
--engine-version minor-version \  
--region region \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output text
```

在Windows中：

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

例如，下列 AWS CLI 命令會判斷美國東部 (俄亥俄州) (美國東部 -2) AWS 區域 MySQL 次要版本 8.0.11 的自動次要升級目標。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

在Windows中：

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

輸出類似如下。

```
-----
| DescribeDBEngineVersions |
+-----+-----+
| AutoUpgrade | EngineVersion |
+-----+-----+
| False      | 8.0.15       |
```

	False		8.0.16	
	False		8.0.17	
	False		8.0.19	
	False		8.0.20	
	False		8.0.21	
	True		8.0.23	
	False		8.0.25	
+-----+-----+				

在此範例中，此 AutoUpgrade 值是 True (若為 MySQL 版本 8.0.23)。因此，自動次要升級目標是 MySQL 版本 8.0.23，其已在輸出中反白顯示。

如果符合下列準則，在維護時段期間會自動升級 MySQL 資料庫執行個體：

- 已啟用 Auto minor version upgrade (自動次要版本升級) 設定。
- Backup retention period (備份保留期間) 設定大於 0。
- 資料庫執行個體執行的次要資料庫引擎版本小於目前的自動升級次要版本。

如需詳細資訊，請參閱 [自動升級次要引擎版本](#)。

使用僅供讀取複本在升級 MySQL 資料庫時減少停機時間

在大多數情況下，藍/綠部署是升級 MySQL 資料庫執行個體時減少停機時間的最佳選項。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

如果您無法使用藍/綠部署，且您的 MySQL 資料庫執行個體目前與生產應用程式搭配使用，您可以使用下列程序來升級資料庫執行個體的資料庫版本。此程序可降低應用程式的停機時間。

透過使用僅供讀取複本，您可以提前執行大部分的維護步驟，並將實際中斷期間的必要變更降到最低。使用此技術，您可以測試和準備新的資料庫執行個體，而無需對現有的資料庫執行個體進行任何變更。

以下程序顯示從 MySQL 5.7 版升級至 MySQL 8.0 版的範例。您可以使用同樣的一般步驟升級到其他的主要版本。

Note

當您從 MySQL 5.7 版升級至 MySQL 8.0 版時，請先完成預先檢查，然後再執行升級。如需詳細資訊，請參閱 [從 MySQL 5.7 升級到 8.0 的前置檢查](#)。


在資料庫執行個體處於使用中的情況下升級 MySQL 資料庫

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 建立 MySQL 5.7 資料庫執行個體的僅供讀取複本。此程序建立資料庫的可升級副本。資料庫執行個體的其他僅供讀取複本也可能存在。
 - a. 在主控台中，選擇 Databases (資料庫)，然後選擇您要升級的資料庫執行個體。
 - b. 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
 - c. 在僅供讀取複本的 DB instance identifier (資料庫執行個體識別符) 中提供一個值，並確定 DB instance class (資料庫執行個體類別) 和其他設定符合您的 MySQL 5.7 資料庫執行個體。
 - d. 選擇 Create read replica (建立僅供讀取複本)。
3. (選擇性) 當僅供讀取複本已建立且 Status (狀態) 顯示 Available (可用) 時，請將僅供讀取複本轉換為異地同步備份部署並啟用備份。

依預設，僅供讀取複本會建立為單一可用區域部署，並停用備份。由於僅供讀取複本最終會成為生產資料庫執行個體，因此最佳實務是設定多可用區部署並立即啟用備份。

- a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛建立的僅供讀取複本。
 - b. 選擇 Modify (修改)。
 - c. 對於 Multi-AZ deployment (異地同步備份部署)，請選擇 Create a standby instance (建立待命執行個體)。
 - d. 對於 Backup Retention Period (備份保留期)，選擇非零正值 (例如 3 天)，然後選擇 Continue (繼續)。
 - e. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
 - f. 選擇 Modify DB instance (修改資料庫執行個體)。
4. 僅供讀取複本的 Status (狀態) 顯示為 Available (可用) 時，請將僅供讀取複本升級至 MySQL 8.0 :
 - a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛建立的僅供讀取複本。
 - b. 選擇 Modify (修改)。
 - c. 在 DB engine version (資料庫引擎版本) 中，選擇升級到 MySQL 8.0 版，然後選擇 Continue (繼續)。
 - d. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
 - e. 選擇 Modify DB instance (修改資料庫執行個體) 以開始升級。

- 升級完成且狀態顯示 [可用] 時，請確認已升級的僅供讀取複本 up-to-date 與來源 MySQL 5.7 資料庫執行個體一起使用。若要驗證，請連線至僅供讀取複本並執行 SHOW REPLICA STATUS 命令。如果 Seconds_Behind_Master 欄位是 0，則複寫為 up-to-date。

 Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

- (選擇性) 建立僅供讀取複本的僅供讀取複本。

如果您希望資料庫執行個體在提升為獨立資料庫執行個體後擁有僅供讀取複本，您可以立即建立僅供讀取複本。

- 在主控台中，選擇 Databases (資料庫)，然後選擇您剛升級的僅供讀取複本。
 - 在 Actions (動作) 中選擇 Create read replica (建立僅供讀取複本)。
 - 在僅供讀取複本的 DB instance identifier (資料庫執行個體識別符) 中提供一個值，並確定 DB instance class (資料庫執行個體類別) 和其他設定符合您的 MySQL 5.7 資料庫執行個體。
 - 選擇 Create read replica (建立僅供讀取複本)。
- (選擇性) 為僅供讀取複本設定自訂資料庫參數群組。

如果您希望資料庫執行個體在提升為獨立資料庫執行個體後使用自訂參數群組，您現在可以建立資料庫參數群組，並將其與僅供讀取複本建立關聯。

- 建立 MySQL 8.0 的自訂資料庫參數群組。如需說明，請參閱「[建立資料庫參數群組](#)」。
 - 修改您想要在剛才建立的資料庫參數群組中變更的參數。如需說明，請參閱「[修改資料庫參數群組中的參數](#)」。
 - 在主控台中，選擇 Databases (資料庫)，然後選擇僅供讀取複本。
 - 選擇 Modify (修改)。
 - 對於 DB parameter group (資料庫參數群組)，請選擇您剛才建立的 MySQL 8.0 資料庫參數群組，然後選擇 Continue (繼續)。
 - 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
 - 選擇 Modify DB instance (修改資料庫執行個體) 以開始升級。
- 將 MySQL 8.0 僅供讀取複本變成獨立資料庫執行個體。

⚠ Important

當 MySQL 8.0 僅供讀取複本提升為獨立資料庫執行個體時，就不再是 MySQL 5.7 資料庫執行個體的複本。建議在維護時段提升 MySQL 8.0 僅供讀取複本，因為此時來源 MySQL 5.7 資料庫執行個體處於唯讀模式，且所有寫入操作都已暫停。提升完成時，您可以將寫入操作導向已升級的 MySQL 8.0 資料庫執行個體，以確保不會遺漏任何寫入操作。

此外，在提升 MySQL 8.0 僅供讀取複本之前，建議您在 MySQL 8.0 僅供讀取複本上執行所有必要的資料定義語言 (DDL) 操作。範例是建立索引。此方法可避免 MySQL 8.0 僅供讀取複本在提升之後效能降低。若要提升僅供讀取複本，請使用以下程序。

- a. 在主控台中，選擇 Databases (資料庫)，然後選擇您剛升級的僅供讀取複本。
 - b. 針對 Actions (動作)，選擇 Promote (提升)。
 - c. 選擇 Yes (是)，對僅供讀取複本執行個體啟用自動備份。如需詳細資訊，請參閱 [備份簡介](#)。
 - d. 選擇 Continue (繼續)。
 - e. 選擇 Promote Read Replica (提升僅供讀取複本)。
9. 您現在已升級 MySQL 資料庫的版本。此時，您可以將應用程式導向至新的 MySQL 8.0 資料庫執行個體。

升級 MySQL 資料庫快照集引擎版本

利用 Amazon RDS，您可以為您的 MySQL 資料庫執行個體建立儲存體磁碟區資料庫快照。建立資料庫快照時，快照會以資料庫執行個體使用的引擎版本為基礎。除了升級您的資料庫執行個體的資料庫引擎版本，您也可以升級您的資料庫快照的引擎版本。對於適用於 MySQL 的 RDS，您可以將 5.7 版快照升級至 8.0 版。您可以升級加密或未加密的資料庫快照。

下列版本支援 MySQL 資料庫快照升級：

- 您可以從適用於 MySQL 的 RDS 快照版本 5.7.16 及更高版本 5.7 版本進行升級。
- 您可以升級至適用於 MySQL 的 RDS 快照版本 8.0.28 及更新版本，但 8.0.29、8.0.30 和 8.0.31 版除外。

您無法將 5.7.40、5.7.41 和 5.7.42 版本升級至 8.0.28 版，但您可以將這些版本升級至 8.0.32 及更高版本。

還原升級至新引擎版本的資料庫快照之後，務必測試升級已成功。如需主要版本升級的詳細資訊，請參閱 [the section called “升級 MySQL 資料庫引擎”](#)。若要了解如何還原資料庫快照，請參閱 [the section called “從資料庫快照還原”](#)。

Note

您無法升級在自動備份程序期間建立的自動資料庫快照。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 升級資料庫快照集。

主控台

升級資料庫快照

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇您要升級的快照。
4. 針對 Actions (動作)，選擇 Upgrade snapshot (升級快照)。Upgrade snapshot (升級快照) 頁面隨即出現。
5. 選擇要升級到的 New engine version (新引擎版本)。

6. 選擇 Save changes (儲存變更) 以升級快照。

升級程序期間，此資料庫快照的所有快照動作會停用。此外，資料庫快照狀態會從 [可用] 變更為 [升級]，然後在完成時變更為 [作用中]。如果資料庫快照集因為快照損毀問題而無法升級，狀態會變更為 [無法使用]。您無法從此狀態復原快照。

Note

如果資料庫升級失敗，快照會轉返至原始版本的原始狀態。

AWS CLI

若要將資料庫快照集升級為新的資料庫引擎版本，請使用 AWS CLI [modify-db-snapshot](#) 指令。

選項

- `--db-snapshot-identifier` – 要升級的資料庫快照識別符。識別符必須是唯一的 Amazon Resource Name (ARN)。如需更多詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。
- `--engine-version` – 資料庫快照會以此引擎版本為目標進行升級。

Example

對於 Linux macOS、或 Unix：

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier my_db_snapshot \  
  --engine-version new_version
```

在 Windows 中：

```
aws rds modify-db-snapshot ^  
  --db-snapshot-identifier my_db_snapshot ^  
  --engine-version new_version
```

RDS API

若要將資料庫快照集升級為新的資料庫引擎版本，請呼叫 RDS API [ModifyDB Snapshot](#) 作業。

參數

- `DBSnapshotIdentifier` – 要升級的資料庫快照識別符。識別符必須是唯一的 Amazon Resource Name (ARN)。如需更多詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。
- `EngineVersion` – 資料庫快照會以此引擎版本為目標進行升級。

將資料匯入 MySQL 資料庫執行個體

您可使用多種不同技術，將資料匯入 RDS for MySQL 資料庫執行個體。最理想的做法需視資料來源、資料量，以及作業是屬於一次性或持續性匯入等條件而定。如要連同資料一起遷移應用程式，也需考量您願意承受的停機時間。

概觀

將資料匯入 RDS for MySQL 資料庫執行個體的各種技術如下表所示。

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
現場部署或 Amazon EC2 上的現有 MySQL 資料庫	任何	一次性	一些	建立現場部署資料庫的備份、將其存放於 Amazon S3，然後將備份檔案還原至執行 MySQL 的新 Amazon RDS 資料庫執行個體。	將備份還原至 MySQL 資料庫執行個體
任何現有資料庫	任何	一次性或持續性	極小	用於 AWS Database Migration Service 以最短的停機時間移轉資料庫，並且對於許多資料庫引擎，繼續進行中的複寫。	《AWS Database Migration Service 使用者指南》中的 什麼是 AWS Database Migration Service 和使用與 MySQL

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
					相容的資料庫作為 AWS DMS 的目標
現有的 MySQL 資料庫執行個體	任何	一次性或持續性	極小	為持續複寫建立僅供讀取複本。升級僅供讀取複本，以便一次性建立新的資料庫執行個體。	使用資料庫執行個體僅供讀取複本
現有的 MariaDB 或 MySQL 資料庫	小型	一次性	一些	使用命令列公用程式，直接將資料複製到您的 MySQL 資料庫執行個體。	將資料從外部 MariaDB 或 MySQL 資料庫匯入至適用於 MariaDB 的 RDS 或適用於 MySQL 資料庫執行個體的 RDS

來源	資料量	一次性或持續性	應用程式停機時間	技術	其他資訊
未存放在現有資料庫的資料	中型	一次性	一些	創建平面文件並使用 MySQL LOAD DATA LOCAL INFILE 語句導入它們。	匯入任何 MariaDB 或 MySQL 資料庫執行個體來源的資料
內部部署或 Amazon EC2 現有的 MariaDB 或 MySQL 資料庫	任何	持續性	極小	使用現有 MariaDB 或 MySQL 資料庫為複寫來源來設定複寫。	使用外部來源執行個體設定二進位日誌檔案位置複寫 將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間

Note

'mysql' 系統資料庫包含登入資料庫執行個體與存取資料所需的身分驗證和授權資訊。卸除、更改、重新命名或截斷資料庫執行個體中 'mysql' 資料庫的資料表、資料或其他內容，可能會導致發生錯誤，造成無法存取資料庫執行個體與資料的情形。如果發生這種情況，您可以使用 AWS CLI `restore-db-instance-from-db-snapshot` 指令從快照還原資料庫執行個體。您可以使用 AWS CLI `restore-db-instance-to-point-in-time` 命令復原資料庫執行個體。

資料匯入考量

以下說明將資料載入 MySQL 的其他相關技術資訊。這些資訊適合熟悉 MySQL 伺服器架構的進階使用者參考。

二進位日誌

相較於關閉二進位日誌，一旦啟用此功能，載入資料會導致效能減損，且需要額外的可用磁碟空間 (最多可能需要原本的四倍大)。至於效能折損的程度與所需的可用磁碟空間，則與載入資料的交易大小成正比。

交易大小

交易大小在 MySQL 資料載入作業中扮演重要角色。交易大小主要會影響資源耗用情形、磁碟空間利用率、恢復程序、復原時間及輸入格式 (一般檔案或 SQL)。本節會說明交易大小如何影響二進位日誌，並解釋為何要在大型資料載入期間停用此功能。如前所述，設定 Amazon RDS 自動備份保留期可啟用及停用二進位日誌。非零的數值會啟用二進位日誌，而零則會停用此功能。我們也會說明大型交易對 InnoDB 的影響，以及為何需要盡量縮減交易大小的原因。

小型交易

若為小型交易，二進位日誌會將載入資料所需的磁碟寫入次數加倍。此效果會嚴重降低其他資料庫作業階段的效能，並增加載入資料所需的時間。效能降低的程度有一部分取決於上傳速率、載入期間發生的其他資料庫活動，以及您 Amazon RDS 資料庫執行個體的容量。

二進位日誌在完成備份及移除前，也會佔用與載入資料量約莫相等的磁碟空間。幸好，Amazon RDS 會頻繁備份並移除二進位日誌，將佔用的磁碟空間減至最少。

大型交易

若啟用二進位日誌，大型交易會耗用 IOPS 與磁碟空間，對效能造成的影響更可達 3 倍。這是因為二進位日誌快取會溢出至磁碟、佔用磁碟空間，導致每次寫入時會發生額外的 IO。在交易遞交或轉返前，快取不會寫入二進位日誌，以致於隨著載入資料量的大小佔用相應的磁碟空間。交易遞交後，快取必須複製到二進位日誌中，導致磁碟上有第三份資料複本。

正因如此，相較於停用二進位日誌，至少要有三倍大小的可用磁碟空間，才能在啟用此功能的情況下載入資料。舉例來說，單次交易有 10 GiB 的載入資料，載入期間會至少佔去 30 GiB 的磁碟空間。資料表佔用 10 GiB + 二進位日誌快取佔用 10 GiB + 二進位日誌本身佔用 10 GiB。在建立此快取檔案的工作階段終止，或另一筆交易的工作階段再次填滿其二進位日誌快取之前，快取檔案都會保留在磁碟上。由於二進位日誌在備份前必須保留在磁碟上，因此要釋放額外的 20 GiB，可能需要一些時間。

如果使用 `LOAD DATA LOCAL INFILE` 命令來載入資料，且必須從一個在本次載入前製作的備份復原資料庫，還會建立另一份資料複本。復原期間，MySQL 會擷取二進位日誌的資料，放入一般檔案中。MySQL 接著會執行 `LOAD DATA LOCAL INFILE` 命令，方式就如同原交易一樣。但是，對資料庫伺服器來說，這次的輸入檔案是本機檔案。若繼續執行前述範例，除非有至少 40 GiB 的可用磁碟空間，否則復原作業就會失敗。

停用二進位記錄

盡可能在大型資料載入期間停用二進位記錄功能，以避免資源負荷與額外的磁碟空間需求。在 Amazon RDS 中停用二進位日誌很簡單，只要將備份保留期的數值設為零即可。如要使用這個方法，建議您在載入前，先拍攝資料庫執行個體的資料庫快照。如此一來，如果有需要，您就可以快速且輕易地復原載入期間所做的變更。

載入完成後，請將備份保留期設回適當 (非零) 的數值。

如果資料庫執行個體是僅供讀取複本的來源資料庫執行個體，則不可將備份保留期設為零。

InnoDB

本節資訊會提供強而有力的論述，說明為何要在使用 InnoDB 時，盡可能縮減交易大小。

Undo (復原)

InnoDB 會產生復原，以支援交易轉返與 MVCC 之類的功能。復原會存放在 InnoDB 系統資料表空間 (通常是 `ibdata1`)，並一直保存到清除執行緒將其移除為止。清除執行緒的作用範圍無法超過最早有效交易的復原，因此在交易遞交或完成轉返前，都能有效封鎖。如果資料庫在載入期間同時處理其他交

易，即使這些交易皆已遞交，且沒有其他交易需要因為 MVCC 而復原，這些復原也會累積在系統資料表空間中，並且無法移除。這種情況下，所有交易 (包含唯讀交易) 只要存取任何交易 (不僅是載入交易) 變更的任何資料列，速度都會變慢。如果不是因為載入交易長時間執行，變慢的原因就是交易掃描了可能早已清除的復原。

復原會存放在系統資料表空間，而系統資料表空間的大小永遠不會縮減。因此，大型資料載入交易可能會造成系統資料工作表變得無比龐大，佔用磁碟空間，且若未從頭重建資料庫，將無法回收這些空間。

轉返

InnoDB 已針對遞交作業完成了最佳化。轉返大型交易需要很多時間。在某些情況下，執行 point-in-time 復原或還原資料庫快照可能會更快。

輸入資料格式

MySQL 可接受以一般檔案或 SQL 為傳入資料的格式。本節會指出兩種格式的幾項重要優點及缺點。

一般檔案

在交易盡量縮減大小的情況下，以 `LOAD DATA LOCAL INFILE` 命令載入一般檔案會是最快速且成本最低的資料載入方式。相較於以 SQL 載入相同資料，一般檔案需要的網路流量通常較少，如此不僅可以降低傳輸成本，而且因為資料庫負荷降低，載入速度也會快速許多。

整併成單一大型交易

`LOAD DATA LOCAL INFILE` 命令會將整個一般檔案視為單一交易，一次載入。這不見得是件壞事。如果可以盡量縮減個別檔案的大小，此方式就有許多優點：

- 繼續執行功能 – 輕鬆追蹤哪些檔案已載入完成。如果載入期間發生問題，您可以從中止的地方輕鬆繼續工作。有些資料可能需要重新傳輸到 Amazon RDS，但使用小檔案的話，要重新傳輸的內容可以減至最少。
- 平行載入資料 – 如果您有多餘的 IOPS 與網路頻寬可用於單檔載入，則平行載入或許可以節省時間。
- 調節載入速率 – 資料載入會對其他處理程序產生負面影響嗎？增加檔案之間的時間可調節載入作業。

請注意

交易大小增加時，`LOAD DATA LOCAL INFILE` 命令的優勢會迅速減少。如果無法將大型資料集拆分成較小的資料集，則 SQL 或許會是較好的選擇。

SQL

SQL 具備一項一般檔案所缺乏的主要優點，亦即 SQL 容易將交易大小維持在較小的狀態。然而，相較於一般檔案，SQL 需要的載入時間較長，而且一旦失敗，很難判斷要從哪裡繼續載入。舉例來說，mysqldump 檔案無法重新啟動。如果在載入 mysqldump 檔案時發生錯誤，檔案需要修改或替換才能繼續載入。替代方案是還原至載入前的時間點，然後在修正失敗原因後重新執行檔案。

使用 Amazon RDS 快照擷取檢查點

如果您的載入作業會持續數小時甚至數天，除非您可以擷取定期檢查點，否則載入時不使用二進位日誌的話，情況恐怕不太樂觀。此時，Amazon RDS 資料庫快照功能就可派上用場。資料庫快照會建立資料庫執行個體的 point-in-time 一致副本，以便在當機或其他事故後，將資料庫還原到該時間點。

若要建立檢查點，只要拍攝資料庫快照即可。您可以移除任何先前為做為檢查點而拍攝的資料庫快照，這不會影響耐用性或還原時間。

拍攝快照的速度也很快，所以經常執行檢查點作業並不會增加太多載入時間。

減少載入時間

以下額外提供幾種可減少載入時間的秘訣：

- 載入前，先建立所有次要索引。對熟悉其他資料庫的使用者來說，這麼做有點違背直覺。新增或修改次要索引會導致 MySQL 建立記錄索引變更的新資料表、從現有資料表複製資料到新資料表，並捨棄原資料表。
- 以 PK 順序載入資料。這個方法對 InnoDB 資料表特別實用，可減少 75–80% 的載入時間，且資料大小減半。
- 停用外部索引鍵限制條件 `foreign_key_checks=0`。若是以 `LOAD DATA LOCAL INFILE` 命令載入一般檔案，大部分情況都必須這麼做。執行任何載入作業時，停用 FK 檢查有助於效能大幅提升。不過，載入完成後，務必啟用限制條件並驗證資料。
- 除非已接近資源上限，否則請使用平行載入。適時使用分割資料表。
- 以 SQL 載入時，使用多值插入可將執行陳述式時的負荷降至最低。使用 `mysqldump` 命令時，系統會自動執行多值插入。
- 降低 InnoDB log IO `innodb_flush_log_at_trx_commit=0`
- 如果您將資料載入沒有僅供讀取複本的資料庫執行個體，請在載入資料時，將 `sync_binlog` 參數設為 0。資料載入後，請將 `sync_binlog` 參數設回 1。
- 在資料庫執行個體轉換為異地同步備份部署前載入資料。然而，如果資料庫執行個體已使用異地同步備份部署，則不建議您在載入資料時切換至單一可用區部署，因為這種作法的效益不高。

Note

使用 `innodb_flush_log_at_trx_commit=0` 會導致 InnoDB 每秒清空日誌，而非於每次遞交時清空。這會帶來顯著的速度優勢，但萬一發生當機，可能會導致資料遺失。請謹慎使用。

主題

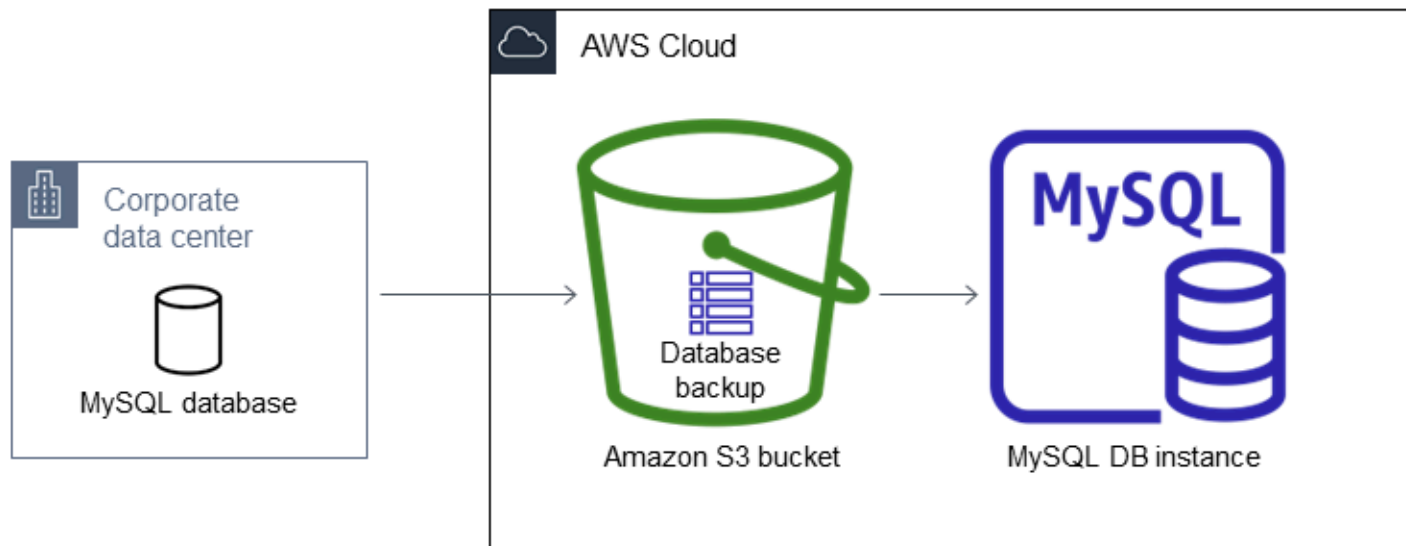
- [將備份還原至 MySQL 資料庫執行個體](#)
- [將資料從外部 MariaDB 或 MySQL 資料庫匯入至適用於 MariaDB 的 RDS 或適用於 MySQL 資料庫執行個體的 RDS](#)
- [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#)
- [匯入任何 MariaDB 或 MySQL 資料庫執行個體來源的資料](#)

將備份還原至 MySQL 資料庫執行個體

Amazon RDS 支援使用備份檔案來匯入 MySQL 資料庫。您能夠建立資料庫的備份，並將其存放在 Amazon S3，接著將備份檔案還原至執行 MySQL 的新 Amazon RDS 資料庫執行個體。

本節所述的案例會還原內部部署資料庫的備份。只要資料庫可以存取，就可以將此技術用於其他位置的資料庫，例如 Amazon EC2 或非 AWS 雲端服務。

您可以在下圖中找到支援的案例。



所有 AWS 區域中的 MySQL 都支援從 Amazon S3 匯入備份檔案。

在建立、複製和還原備份檔案時，如果您的內部部署資料庫可以離線，建議您使用備份檔案將資料庫匯入 Amazon RDS。如果資料庫不能離線，當您如本主題所述透過 Amazon S3 來移轉至 Amazon RDS 之後，您可以使用二進位制日誌 (binlog) 複寫來更新資料庫。如需詳細資訊，請參閱 [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。您也可以使用 AWS Database Migration Service 將資料庫遷移至 Amazon RDS。如需詳細資訊，請參閱 [什麼是 AWS Database Migration Service ?](#)

從 Amazon S3 將備份檔案匯入 Amazon RDS 的限制與建議

以下是從 Amazon S3 匯入備份檔案的一些限制和建議：

- 您只能將資料匯入新的資料庫執行個體，而不是現有的資料庫執行個體。
- 您必須使用 Percona 建 XtraBackup 立內部部署資料庫的備份。
- 您無法從資料庫快照將資料匯入 Amazon S3。
- 如果來源資料庫有資料表是在預設 MySQL 資料目錄外定義，您無法從這個來源資料庫遷移。

- 不支持 MySQL 的 Percona 服務器作為源數據庫，因為它可以包含mysql模式中的compression_dictionary*表。
- 您必須將資料匯入 AWS 區域中 MySQL 主要版本的預設次要版本。例如，若您的主要版本是 MySQL 8.0，而 AWS 區域的預設次要版本是 8.0.28，則您必須將資料匯入 MySQL 8.0.28 版資料庫執行個體。您可以在匯入後升級資料庫執行個體。如需決定預設次要版本的相關資訊，請參閱 [MySQL on Amazon RDS 版本](#)。
- 主要版本和次要版本都不支援回溯遷移。例如，您無法從 8.0 版遷移至 5.7 版，也無法從 8.0.32 版遷移至 8.0.31 版。
- 您無法匯入 MySQL 5.5 或 5.6 資料庫。
- 您無法將內部部署 MySQL 資料庫從一個主要版本匯入到另一個主要版本。例如，您無法將 MySQL 5.7 資料庫匯入至 RDS for MySQL 8.0 資料庫。您可以在完成匯入之後升級資料庫執行個體。
- 您無法從加密的來源資料庫還原，但可以還原至加密的 Amazon RDS 資料庫執行個體。
- 您無法從 Amazon S3 儲存貯體中加密的備份還原。
- 如果 Amazon S3 儲存貯體和 Amazon RDS 資料庫執行個體位於不同的 AWS 區域，您無法從這個儲存貯體還原。
- db.t2.micro 資料庫執行個體類別上不支援從 Amazon S3 匯入。不過，您可以先還原至不同的資料庫執行個體類別，稍後再變更資料庫執行個體類別。如需執行個體類別的詳細資訊，請參閱 [的資料庫執行個體類別的硬體規格](#)。
- Amazon S3 將上傳至 Amazon S3 儲存貯體的檔案大小限制為 5 TB。如果備份檔案超過 5 TB，您必須將備份檔案分割為較小的檔案。
- 當您還原資料庫時，會複製備份，然後在資料庫執行個體上解壓縮。因此，為資料庫執行個體佈建等於或大於備份大小總和的儲存空間，以及磁碟上原始資料庫的大小。
- Amazon RDS 將上傳至 Amazon S3 儲存貯體的檔案數量限制為 1 百萬個。如果資料庫的備份資料 (包括所有完整和增量備份) 超過 1 百萬個檔案，請使用 Gzip (.gz)、tar (.tar.gz) 或 Percona xstream (.xstream) 檔案將完整和增量備份檔案儲存在 Amazon S3 儲存貯體中。佩科納 XtraBackup 8.0 僅支持用於壓縮的帕科納 xstream。
- 不會自動匯入使用者帳戶。儲存來源資料庫中的使用者帳戶，稍後再新增至新的資料庫執行個體。
- 不會自動匯入函數。儲存來源資料庫中的函數，稍後再新增至新的資料庫執行個體。
- 不會自動匯入預存程序。儲存來源資料庫中的預存程序，稍後再新增至新的資料庫執行個體。
- 不會自動匯入時區資訊。記錄來源資料庫的時區資訊，稍後再設定新的資料庫執行個體的時區。如需詳細資訊，請參閱 [MySQL 資料庫執行個體的本機時區](#)。

- `innodb_data_file_path` 參數只能使用一個資料檔案 (預設資料檔案名稱 "ibdata1:12M:autoextend") 來設定。具有兩個資料檔或具有不同名稱之資料檔的資料庫無法使用此方法移轉。

以下是不允許的檔案名稱範例："innodb_data_file_path=ibdata1:50M; ibdata2:50M:autoextend" 和 "innodb_data_file_path=ibdata01:50M:autoextend"。

- 還原的資料庫大小上限是支援的資料庫大小上限減去備份大小。因此，如果支援的資料庫大小上限為 64 TiB，且備份的大小為 30 TiB，則還原的資料庫大小上限為 34 TiB，如下列範例所示：

$$64 \text{ TiB} - 30 \text{ TiB} = 34 \text{ TiB}$$

如需 Amazon RDS for MySQL 支援的資料庫大小上限的詳細資訊，請參閱 [一般用途 SSD 儲存體](#) 和 [佈建 IOPS SSD 儲存體](#)。

設定將備份檔案從 Amazon S3 匯入 Amazon RDS 的概觀

以下是從 Amazon S3 將備份檔案匯入 Amazon RDS 所需設定的元件：

- 用來儲存備份檔案的 Amazon S3 儲存貯體。
- 由 Percona XtraBackup 建立的內部部署資料庫備份。
- 允許 Amazon RDS 存取儲存貯體的 AWS Identity and Access Management (IAM) 角色。

如果您已經有 Amazon S3 儲存貯體，則可直接使用。如果沒有，您可以建立新的 Amazon S3 儲存貯體。如果您要建立新的儲存貯體，請參閱[建立儲存貯體](#)。

使用 Percona XtraBackup 工具創建您的備份。如需詳細資訊，請參閱 [建立資料庫備份](#)。

如果您已經有 IAM 角色，則可直接使用。如果沒有，您可以手動建立新的 IAM 角色。或者，當您使用 AWS Management Console 還原資料庫時，您可以選擇讓精靈在您的帳戶中建立新的 IAM 角色。如果您要手動建立新的 IAM 角色，或將信任和許可政策附加至現有的 IAM 角色，請參閱[手動建立 IAM 角色](#)。如果希望系統為您建立新的 IAM 角色，請遵循 [主控台](#) 中的程序。

建立資料庫備份

使用 Percona XtraBackup 軟件創建您的備份。我們建議您使用最新版本的 Percona XtraBackup。您可以 XtraBackup 從 [下載佩爾科納安裝佩爾 XtraBackup](#) 科納。

⚠ Warning

建立資料庫備份時，XtraBackup 可能會將認證儲存在 xtrabackup_info 檔案中。請務必檢查該檔案，確保其中的 tool_command 設定不包含任何敏感資訊。

ℹ Note

對於 MySQL 8.0 遷移，您必須使用佩科納 XtraBackup 8.0。佩科納 XtraBackup 8.0.12 及更高版本支持 MySQL 的所有版本的遷移。如果您要移轉至適用於 MySQL 8.0.20 或更高版本的 RDS，您必須使用佩科納 XtraBackup 8.0.12 或更高版本。

對於 MySQL 5.7 遷移，您還可以使用佩科納 XtraBackup 2.4。對於早期 MySQL 版本的遷移，您也可以使用佩科納 XtraBackup 2.3 或 2.4。

您可以使用佩科納 XtraBackup 創建 MySQL 資料庫文件的完整備份。或者，如果您已經使用 Percona XtraBackup 備份 MySQL 資料庫檔案，您可以上傳現有的完整和增量備份目錄和檔案。

有關使用 Percona 備份資料庫的更多信息 XtraBackup，請參閱 [Percona XtraBackup -文檔](#) 和 [Percona 網站上的 Xtrabackup 二進製文件](#)。

使用佩爾科納創建完整備份 XtraBackup

若要建立可從 Amazon S3 還原的 MySQL 資料庫檔案的完整備份，請使用 Percona XtraBackup 公用程式 (xtrabackup) 來備份您的資料庫。

例如，下列命令會建立 MySQL 資料庫的備份，並將檔案儲存在 /on-premises/s3-restore/backup 資料夾中。

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/s3-restore/backup>
```

如果要將備份壓縮成單一檔案 (稍後可視需要分割)，您可以將備份儲存為下列其中一種格式：

- Gzip (.gz)
- tar (.tar)
- Percona xstream (.xstream)

Note

佩科納 XtraBackup 8.0 僅支持用於壓縮的帕科納 xstream。

下列命令建立 MySQL 資料庫的備份，並分割成多個 Gzip 檔案。

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \  
--target-dir=</on-premises/s3-restore/backup> | gzip - | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.tar.gz
```

下列命令建立 MySQL 資料庫的備份，並分割成多個 tar 檔案。

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \  
--target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.tar
```

下列命令建立 MySQL 資料庫的備份，並分割成多個 xstream 檔案。

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=xstream \  
--target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.xstream
```

Note

如果您看到下列錯誤，可能是因為在命令中混合檔案格式所致：

```
ERROR:/bin/tar: This does not look like a tar archive
```

搭配佩爾科納使用增量備份 XtraBackup

如果您已使用 Percona XtraBackup 對 MySQL 資料庫檔案執行完整和增量備份，則不需要建立完整備份並將備份檔案上傳到 Amazon S3。反之，您可以將現有的備份目錄和檔案複製到 Amazon S3 儲存貯體，以節省大量時間。如需有關使用 Percona 建立增量備份的詳細資訊 XtraBackup，請參閱[增量備份](#)。

將現有的完整和增量備份檔案複製到 Amazon S3 儲存貯體時，您必須遞迴複製基本目錄的內容。這些內容包括完整備份，以及所有增量備份目錄和檔案。此副本必須保留 Amazon S3 儲存貯體中的

目錄結構。Amazon RDS 會逐一查看所有檔案和目錄。Amazon RDS 使用包含在每個增量備份中的 `xtrabackup-checkpoints` 檔案，以識別基本目錄，以及依記錄序號 (LSN) 範圍來排序增量備份。

佩爾科納的 Backup 注意事項 XtraBackup

Amazon RDS 根據檔案名稱來取用備份檔案。請根據檔案格式，以適當副檔名來命名備份檔案 — 例如，使用 Percona xstream 格式儲存的檔案應該採用 `.xstream` 副檔名。

Amazon RDS 依字母順序和自然數順序來取用備份檔案。發出 `split` 命令時，請使用 `xtrabackup` 選項，以確保依適當順序寫入和命名備份檔案。

Amazon RDS 不支持使用佩爾科納 XtraBackup 創建的部分備份。當您備份資料庫的來源檔案時，您無法使用 `--tables`、`--tables-exclude`、`--tables-file`、`--databases`、`--databases-exclude` 或 `--databases-file` 選項來建立局部備份。

Amazon RDS 支持使用佩爾科納 XtraBackup 創建的增量備份。如需有關使用 Percona 建立增量備份的詳細資訊 XtraBackup，請參閱[增量備份](#)。

手動建立 IAM 角色

如果沒有，您可以手動建立新的 IAM 角色。不過，如果您使用還原資料庫 AWS Management Console，建議您遵循中的程序，[主控台](#)並選擇讓 RDS 為您建立這個新的 IAM 角色。

若要手動建立新的 IAM 角色以便從 Amazon S3 匯入資料庫，請建立可從 Amazon RDS 將許可委派給 Amazon S3 儲存貯體的角色。當您建立 IAM 角色時，您需要附加信任和許可政策。若要從 Amazon S3 匯入備份檔案，請使用類似下列範例的信任和許可政策。如需有關建立角色的詳細資訊，請參閱[建立角色以將權限委派給 AWS 服務](#)。

信任和許可政策規定您必須提供 Amazon Resource Name (ARN)。如需 ARN 格式的詳細資訊，請參閱[Amazon 資源名稱 \(ARN\) 和 AWS 服務命名空間](#)。

Example 從 Amazon S3 匯入的信任政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {"Service": "rds.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    ]]  
  }
```

Example 從 Amazon S3 匯入的許可政策 — IAM 使用者許可

在下面的例子中，用您自己的值替換 *iam_user_id*。

```
{  
  "Version": "2012-10-17",  
  "Statement":  
  [  
    {  
      "Sid": "AllowS3AccessRole",  
      "Effect": "Allow",  
      "Action": "iam:PassRole",  
      "Resource": "arn:aws:iam::iam_user_id:role/S3Access"  
    }  
  ]  
}
```

Example 從 Amazon S3 匯入的許可政策 — 角色許可

在下面的例子中，用您自己的值替換 *## EXAMPLE-BUCKET* 和 *##*。

```
{  
  "Version": "2012-10-17",  
  "Statement":  
  [  
    {  
      "Effect": "Allow",  
      "Action":  
      [  
        "s3:ListBucket",  
        "s3:GetBucketLocation"  
      ],  
      "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET"  
    },  
    {  
      "Effect": "Allow",  
      "Action":  
      [  
        "s3:GetObject"  
      ]  
    }  
  ]  
}
```

```
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix*"
  },
  { // If your bucket is encrypted, include the following permission. This
    permission allows decryption of your AWS KMS key.
    "Effect": "Allow",
    "Action":
      [
        "kms:Decrypt"
      ],
    "Resource": [
      "arn:aws:kms:region:customer_id:key/key_id*"
    ]
  }
]
```

Note

如果包含檔案名稱字首，請在字首後面加上星號 (*)。如果您不想指定字首，則指定星號即可。

將資料從 Amazon S3 匯入新的 MySQL 資料庫執行個體

您可以使用 AWS Management Console、AWS CLI 或 RDS API 將資料從 Amazon S3 匯入新的 MySQL 資料庫執行個體。

主控台

從 Amazon S3 將資料匯入新的 MySQL 資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要 AWS 區域 在其中建立資料庫執行個體的項目。選擇與 AWS 區域 包含資料庫備份的 Amazon S3 儲存貯體相同的儲存貯體。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Restore From S3 (從 S3 還原)。

系統會顯示 Create database by restoring from S3 (從 S3 還原以建立資料庫) 頁面。

RDS > Databases > Restore from S3

Create database by restoring from S3

S3 destination


Write audit logs to S3
Enter a destination in Amazon S3 where your audit logs will be stored. Amazon S3 is object storage build to store and retrieve any amount of data from anywhere


S3 bucket
db-backup-bucket-1234.xyz

S3 prefix (optional) [Info](#)

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible) 

MySQL 

Edition

MySQL Community

Source engine version [Info](#)

8.0

Engine Version

MySQL 8.0.33

5. 在 S3 目的地下：
 - a. 選擇包含備份的 S3 bucket (S3 儲存貯體)。
 - b. (選擇性) 對於 S3 前置詞，請輸入存放在 Amazon S3 儲存貯體中之檔案的檔案路徑前置詞。

如果不指定字首，RDS 會使用 S3 儲存貯體之根資料夾中的所有檔案和資料夾來建立資料庫執行個體。如果指定字首，RDS 會使用 S3 儲存貯體中的檔案和資料夾來建立資料庫執行個體，而且檔案的路徑以指定的字首開頭。

例如，假設您將備份檔案儲存在 S3 中的一個名為 backups 的子資料夾，而且有多組備份檔案，各存放於自己的目錄中 (gzip_backup1、gzip_backup2 等等)。在此例子中，指定字首 backups/gzip_backup1，即可從 gzip_backup1 資料夾中的檔案還原。

6. 在 Engine options (引擎選項)：


- a. 針對 Engine type (引擎類型)，請選擇 MySQL。
- b. 在 Source engine version (來源引擎版本) 中，選擇來源資料庫的 MySQL 主要版本。
- c. 對於引擎版本，請在您的 AWS 區域。

在中 AWS Management Console，只有預設的次要版本可用。您可以在匯入後升級資料庫執行個體。

7. 對於 IAM 角色，建立或選擇具有允許 Amazon RDS 存取 Amazon S3 儲存貯體的必要信任政策和許可政策的 IAM 角色。執行下列動作之一：

- (建議) 選擇 [建立新角色]，然後輸入 IAM 角色名稱。使用此選項，RDS 會自動為您建立具有信任原則和權限原則的角色。
- 選擇現有的 IAM 角色。請確保此角色符合中的所有條件 [the section called “手動建立 IAM 角色”](#)。

8. 指定您的資料庫執行個體資訊。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

 Note

務必分配足夠的記憶體給新的資料庫執行個體，以確保還原操作成功。

您也可以選擇 Enable storage autoscaling (啟用儲存體自動調整規模)，以允許未來自動增長。

9. 視需要選擇其他設定。

10. 選擇 Create database (建立資料庫)。

AWS CLI

若要使用將資料從 Amazon S3 匯入新的 MySQL 資料庫執行個體 AWS CLI，請使用下列參數呼叫從 S3 [恢復-DB 執行個體命令](#)。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

Note

務必分配足夠的記憶體給新的資料庫執行個體，以確保還原操作成功。
您也可以使用 `--max-allocated-storage` 參數來啟用儲存體自動調整規模，以允許未來自動增長。

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--manage-master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-s3 \  
  --allocated-storage 250 \  
  --db-instance-identifier myidentifier \  
  --db-instance-class db.m5.large \  
  --engine mysql \  
  --master-username admin \  
  --manage-master-user-password \  
  --s3-bucket-name DOC-EXAMPLE-BUCKET \  
  --source-engine mysql \  
  --source-engine-version 5.7.33
```

```
--s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \  
--s3-prefix bucketprefix \  
--source-engine mysql \  
--source-engine-version 8.0.32 \  
--max-allocated-storage 1000
```

在 Windows 中：

```
aws rds restore-db-instance-from-s3 ^  
  --allocated-storage 250 ^  
  --db-instance-identifier myidentifier ^  
  --db-instance-class db.m5.large ^  
  --engine mysql ^  
  --master-username admin ^  
  --manage-master-user-password ^  
  --s3-bucket-name DOC-EXAMPLE-BUCKET ^  
  --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^  
  --s3-prefix bucketprefix ^  
  --source-engine mysql ^  
  --source-engine-version 8.0.32 ^  
  --max-allocated-storage 1000
```

RDS API

若要使用 Amazon RDS API 將資料從 Amazon S3 匯入新的 MySQL 資料庫執行個體，請撥打還原 B S3 操作 [InstanceFrom](#)。

將資料從外部 MariaDB 或 MySQL 資料庫匯入至適用於 MariaDB 的 RDS 或適用於 MySQL 資料庫執行個體的 RDS

您也可將資料從現有的 MariaDB 或 MySQL 資料庫匯入 MySQL 或 MariaDB 資料庫執行個體。方法是以 [mysqldump](#) 命令複製資料庫，然後直接傳輸到 MariaDB 或 MySQL 資料庫執行個體。`mysqldump` 命令列公用程式普遍用於製作備份，以及將資料從 MariaDB 或 MySQL 伺服器傳輸到另一個伺服器上。MySQL 和 MariaDB 用戶端軟體皆隨附這個程式。

Note

如果您使用 MySQL 資料庫執行個體匯入或匯出大量資料，使用 `xtrabackup` 備份檔案和 Amazon Amazon S3 將資料移入和移出 Amazon RDS 會更加可靠且更快速。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

典型的 `mysqldump` 命令會將資料從外部資料庫移入 Amazon RDS 資料庫執行個體，範例如下。

```
mysqldump -u local_user \  
  --databases database_name \  
  --single-transaction \  
  --compress \  
  --order-by-primary \  
-plocal_password | mysql -u RDS_user \  
  --port=port_number \  
  --host=host_name \  
  -pRDS_password
```

Important

切勿在 `-p` 選項與輸入的密碼間插入空白。
指定此處所顯示提示以外的憑證，作為安全最佳實務。

請確認您了解下列建議和注意事項：

- 從傾印檔案中排除下列結構描述：`sys`、`performance_schema` 和 `information_schema`。`mysqldump` 公用程式已預設排除這些結構描述。
- 如果您需要移轉使用者和權限，請考慮使用可產生資料控制語言 (DCL) 的工具來重新建立使用者和權限，例如公用程式。[pt-show-grants](#)
- 若要執行匯入，請確認執行此程序的使用者能夠存取資料庫執行個體。如需詳細資訊，請參閱 [使用安全群組控制存取](#)。

使用的參數如下：

- `-u local_user` – 用來指定使用者名稱。首次使用此參數時，需在 `--databases` 參數所識別的本機 MariaDB 或 MySQL 資料庫中指定使用者帳戶名稱。
- `--databases database_name` – 在您要匯入至 Amazon RDS 的本機 MariaDB 或 MySQL 執行個體上，指定資料庫名稱。
- `--single-transaction` – 確保從本機資料庫負載的所有資料均與單一時間點一致。如有其他程序在 `mysqldump` 讀取資料時變更資料，使用此參數有助於維持資料完整性。
- `--compress` – 在本機資料庫的資料傳送到 Amazon RDS 前先完成壓縮，以減少耗用的網路頻寬。
- `--order-by-primary` – 以資料的主索引鍵排序各資料表的資料，以減少載入時間。

- `-p`*local_password* – 用來指定密碼。首次使用此參數時，需指定第一個 `-u` 參數所識別的使用者帳戶密碼。
- `-u` *RDS_user* – 用來指定使用者名稱。第二次使用此參數時，需在 `--host` 參數所識別的 MariaDB 或 MySQL 資料庫執行個體的預設資料庫上，指定使用者帳戶名稱。
- `--port` *port_number* – 用於指定 MariaDB 或 MySQL 資料庫執行個體的連接埠。埠號預設為 3306，除非您在建立執行個體時另行變更。
- `--host` *host_name* – 用來指定 Amazon RDS 資料庫執行個體端點的網域名稱系統 (DNS) 名稱，例如 `.myinstance.123456789012.us-east-1.rds.amazonaws.com` 端點值可在 Amazon RDS 管理主控台的執行個體詳細資訊中找到。
- `-p`*RDS_password* – 用來指定密碼。第二次使用此參數時，需指定第二個 `-u` 參數所識別的使用者帳戶密碼。

必須在 Amazon RDS 資料庫中手動建立預存程序、觸發程序、函數或事件。如果您正在複製的資料庫中存有上述物件，則請在執行 `mysqldump` 時排除這些物件。為此，請在 `mysqldump` 命令中包含以下參數：`--routines=0 --triggers=0 --events=0`。

下列範例會將本機主機上的 `world` 範例資料庫複製到 MySQL 資料庫執行個體。

對於 Linux/macOS、或 Unix：

```
sudo mysqldump -u localuser \  
  --databases world \  
  --single-transaction \  
  --compress \  
  --order-by-primary \  
  --routines=0 \  
  --triggers=0 \  
  --events=0 \  
  -plocalpassword | mysql -u rdsuser \  
    --port=3306 \  
    --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \  
    -prdspassword
```

若為 Windows 系統，請在命令提示字元中執行下列命令，方法是對 Windows 程式選單中的命令提示字元按一下滑鼠右鍵，然後選擇以系統管理員身分執行：

```
mysqldump -u localuser ^  
  --databases world ^
```

```
--single-transaction ^
--compress ^
--order-by-primary ^
--routines=0 ^
--triggers=0 ^
--events=0 ^
-plocalpassword | mysql -u rdsuser ^
  --port=3306 ^
  --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
-prdpassword
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

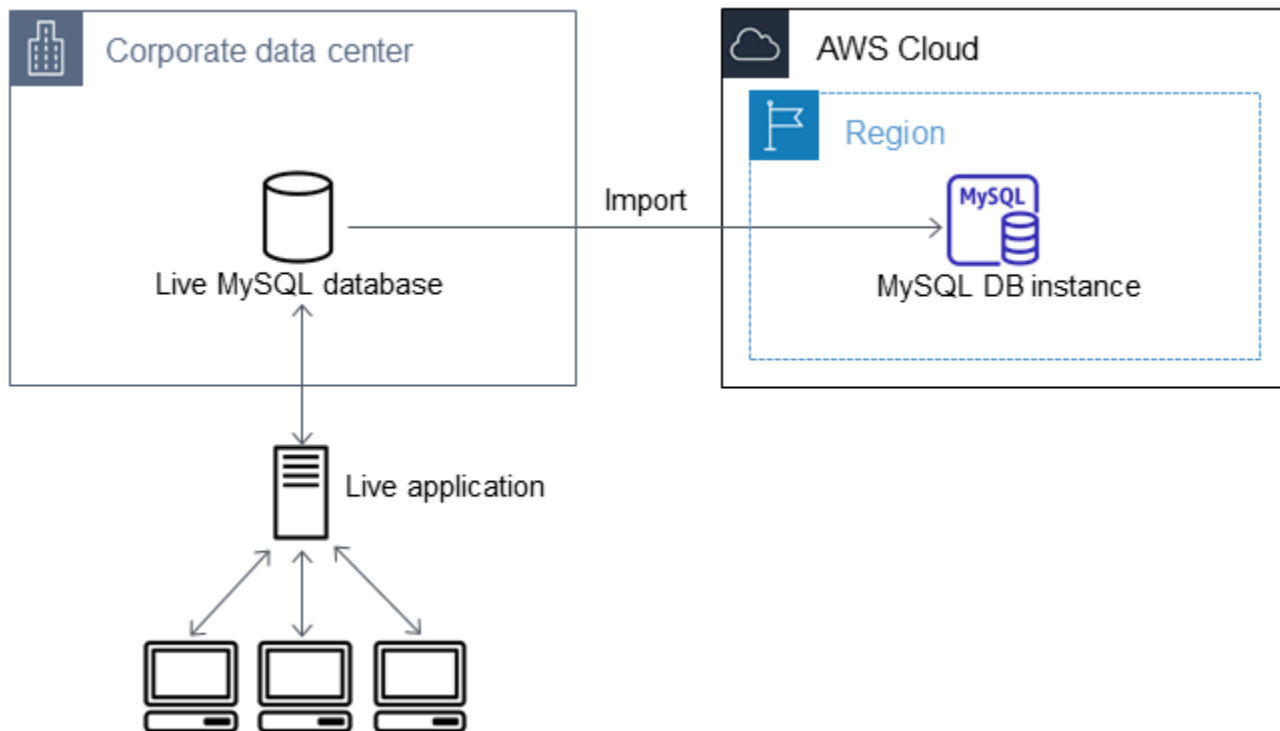
將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間

某些情況下，您可能需要將資料從支援即時應用程式的外部 MariaDB 或 MySQL 資料庫，匯入至 MariaDB 資料庫執行個體、MySQL 資料庫執行個體或 MySQL 多可用區域資料庫叢集。使用下列程序，將對應用程式可用性的影響降至最低。如果您使用的是超大型資料庫，也可使用這個程序。使用此程序，您可以減少透過網路傳送的資料量，以降低匯入成本 AWS。

此程序中，您需將資料庫資料的複本傳輸至 Amazon EC2 執行個體，並將資料匯入至新的 Amazon RDS 資料庫。然後，您可以使用複寫將 Amazon RDS 資料庫 up-to-date 與即時外部執行個體一起使用，然後再將應用程式重新導向至 Amazon RDS 資料庫。如果外部執行個體版本是 MariaDB 10.0.24 或更高版本，且目標執行個體為 RDS for MariaDB 時，請根據全域交易識別符 (GTID) 設定 MariaDB 複寫。若為其他情況，請根據二進位日誌座標設定複寫。建議 GTID 型複寫 (如果您的外部資料庫可以支援它)，因為 GTID 型複寫是更可信賴的方法。如需詳細資訊，請參閱 MariaDB 文件中的 [Global transaction ID](#) 一節。

Note

如果您想要將資料匯入至 MySQL 資料庫執行個體，且您的情況允許，建議您使用備份檔案與 Amazon S3 將資料移入和移出 Amazon RDS。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

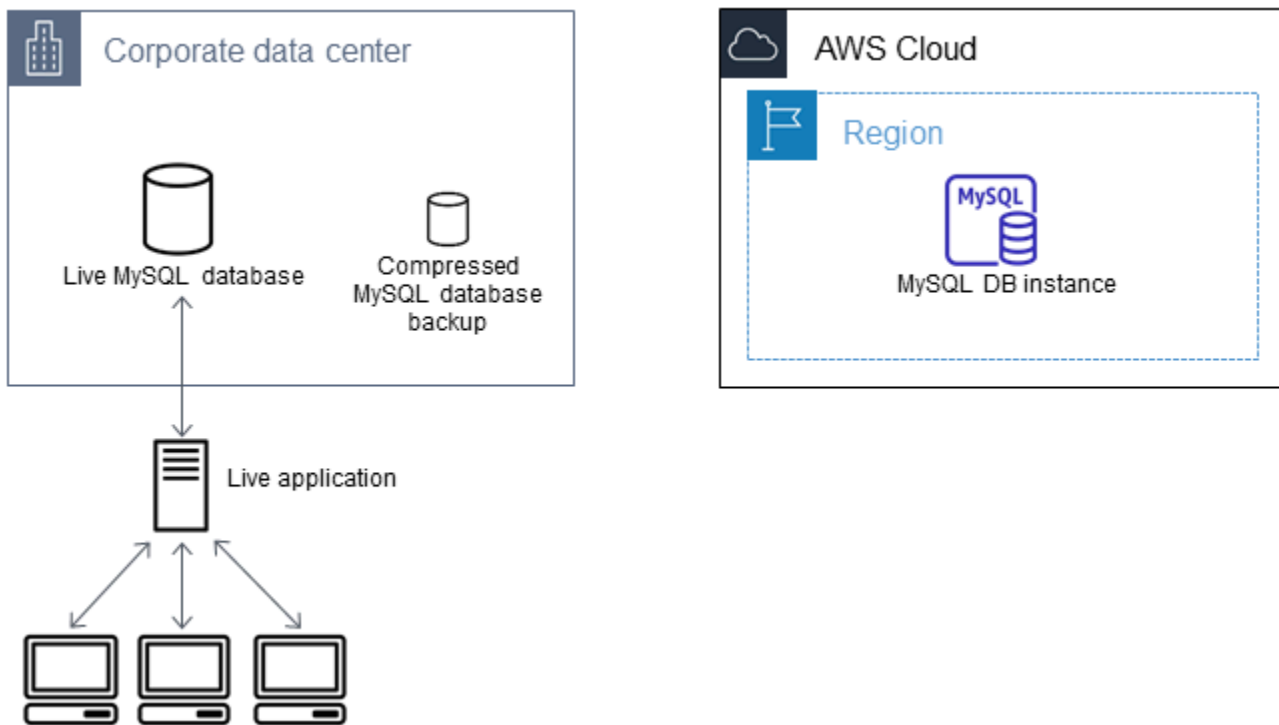


Note

由於可能發生複寫問題，我們不建議您將此程序用於比 MySQL 5.5 更早版本的來源 MySQL 資料庫。如需詳細資訊，請參閱 MySQL 文件中的 [Replication compatibility between MySQL versions](#) 一節。

建立現有資料庫的複本

要將大量資料遷移至 RDS for MariaDB 或 RDS for MySQL 資料庫，並將停機時間降至最低，第一步是建立來源資料的複本。



您可使用 `mysqldump` 公用程式，以 SQL 或分隔符號文字等任一格式建立資料庫備份。建議您在非生產環境中分別測試各種格式，以了解哪種方法能將 `mysqldump` 命令的執行時間減至最少。

亦建議您衡量 `mysqldump` 程式碼的效能與使用分隔符號文字格式載入所帶來的效益。使用分隔符號文字格式的備份會為每個傾印的資料表分別建立索引標籤分隔文字檔案。您可以使用 `LOAD DATA LOCAL INFILE` 命令平行載入這些檔案，縮短匯入資料庫所需的時間。如需選擇 `mysqldump` 格式並載入資料的相關詳細資訊，請參閱 MySQL 文件中的 [Using mysqldump For Backups](#) 一節。

開始備份作業前，請確認在要複製到 Amazon RDS 的 MariaDB 或 MySQL 資料庫上設定複寫選項。複寫選項包括開啟二進位日誌及設定獨一無二的伺服器 ID。設定這些選項後，伺服器會開始記錄資料庫交易，並調整至就緒狀態，以擔任後續程序的來源複寫執行個體。

Note

對 `mysqldump` 使用 `--single-transaction` 選項，因為它會傾印資料庫的一致狀態。若要確保有效的傾印檔案，請勿在執行 `mysqldump` 時執行資料定義語言 (DDL) 陳述式。您可以為這些作業排定維護時段。

從傾印檔案中排除下列結構描述：`sys`、`performance_schema` 和 `information_schema`。`mysqldump` 公用程式已預設排除這些結構描述。

若要移轉使用者和權限，請考慮使用產生資料控制語言 (DCL) 的工具來重新建立使用者和權限，例如公用程式。[pt-show-grants](#)

設定複寫選項

1. 編輯 `my.cnf` 檔案 (此檔案通常位於 `/etc` 之下)。

```
sudo vi /etc/my.cnf
```

將 `log_bin` 和 `server_id` 選項新增至 `[mysqld]` 部分。`log_bin` 選項會提供二進位記錄檔的檔案名稱識別符。`server_id` 選項會為來源與複本關係提供伺服器唯一識別碼。

以下範例顯示 `my.cnf` 檔案的 `[mysqld]` 部分的更新結果。

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

如需詳細資訊，請參閱 [MySQL 文件](#)。

2. 如需使用多可用區域資料庫叢集進行複寫，請將 `ENFORCE_GTID_CONSISTENCY` 和 `GTID_MODE` 參數設為 `ON`。

```
mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

使用資料庫執行個體進行複製時，不需要這些設定。

3. 重新啟動 `mysql` 服務。

```
sudo service mysqld restart
```

建立現有資料庫的備份複本

1. 您可使用 `mysqldump` 公用程式，以 `SQL` 或分隔符號文字等任一格式建立資料備份。

指定 `--master-data=2` 來建立可用來啟動伺服器間複寫作業的備份檔案。如需詳細資訊，請參閱 [mysqldump](#) 文件。

如需提升效能並確保資料完整性，請使用 `mysqldump` 的 `--order-by-primary` 和 `--single-transaction` 選項。

為避免備份中包含 MySQL 系統資料庫，請勿以 mysqldump 搭配使用 `--all-databases` 選項。如需詳細資訊，請參閱 MySQL 文件中的 [Creating a Data Snapshot Using mysqldump](#) 一節。

如有必要，請使用 `chmod` 命令，確保建立備份檔的目錄可以寫入。

Important

在 Windows 上，以管理員身分執行命令視窗。

- 若要產生 SQL 輸出，請使用下列命令。

對於LinuxmacOS、或Unix：

```
sudo mysqldump \  
  --databases database_name \  
  --master-data=2 \  
  --single-transaction \  
  --order-by-primary \  
  -r backup.sql \  
  -u local_user \  
  -p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

在 Windows 中：

```
mysqldump ^  
  --databases database_name ^  
  --master-data=2 ^  
  --single-transaction ^  
  --order-by-primary ^  
  -r backup.sql ^  
  -u local_user ^  
  -p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

- 若要產生分隔文字輸出，請使用下列命令。

對於Linux/macOS、或Unix：

```
sudo mysqldump \  
  --tab=target_directory \  
  --fields-terminated-by ',' \  
  --fields-enclosed-by '"' \  
  --lines-terminated-by 0x0d0a \  
  database_name \  
  --master-data=2 \  
  --single-transaction \  
  --order-by-primary \  
  -p password
```

在 Windows 中：

```
mysqldump ^\  
  --tab=target_directory ^\  
  --fields-terminated-by "," ^\  
  --fields-enclosed-by "" ^\  
  --lines-terminated-by 0x0d0a ^\  
  database_name ^\  
  --master-data=2 ^\  
  --single-transaction ^\  
  --order-by-primary ^\  
  -p password
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

必須在 Amazon RDS 資料庫中手動建立預存程序、觸發程序、函數或事件。如果您正在複製的資料庫中存有上述物件，則請在執行 mysqldump 時排除這些物件。為此，請在 mysqldump 命令中包含以下參數：`--routines=0 --triggers=0 --events=0`。

若是使用分隔符號文字格式，執行 `mysqldump` 命令時，系統會傳回 `CHANGE MASTER TO` 註解。此註解會包含主控端日誌檔案的名稱與位置。如果外部執行個體不是 MariaDB 10.0.24 或更新版本，請注意 `MASTER_LOG_FILE` 和 `MASTER_LOG_POS` 的值。設定複寫時您需要這些值。

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
MASTER_LOG_POS=107;
```

如果您使用的是 SQL 格式，您可以在備份檔案的 `CHANGE MASTER TO` 註解中取得主控端日誌檔案的名稱與位置。如果外部執行個體是 MariaDB 10.0.24 或更新版本，您可在下一步驟中取得 GTID。

- 如果您正在使用的外部執行個體是 MariaDB 10.0.24 或更新版本，您可使用 GTID 型複寫功能。在外部 MariaDB 執行個體上執行 `SHOW MASTER STATUS` 命令，以取得二進位日誌檔案的名稱和位置，接著在外部 MariaDB 執行個體上執行 `BINLOG_GTID_POS` 命令，將其轉換為 GTID。

```
SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
```

請記下系統傳回的 GTID，設定複寫時會需要使用。

- 壓縮複製的資料，以減少複製資料到 Amazon RDS 資料庫所需的網路資源量。請記下備份檔案的大小。決定要建立多大的 Amazon EC2 執行個體時，您需要此資訊。完成後，請使用 GZIP 或您慣用的壓縮公用程式壓縮備份檔。

- 若要壓縮 SQL 輸出，請使用下列命令。

```
gzip backup.sql
```

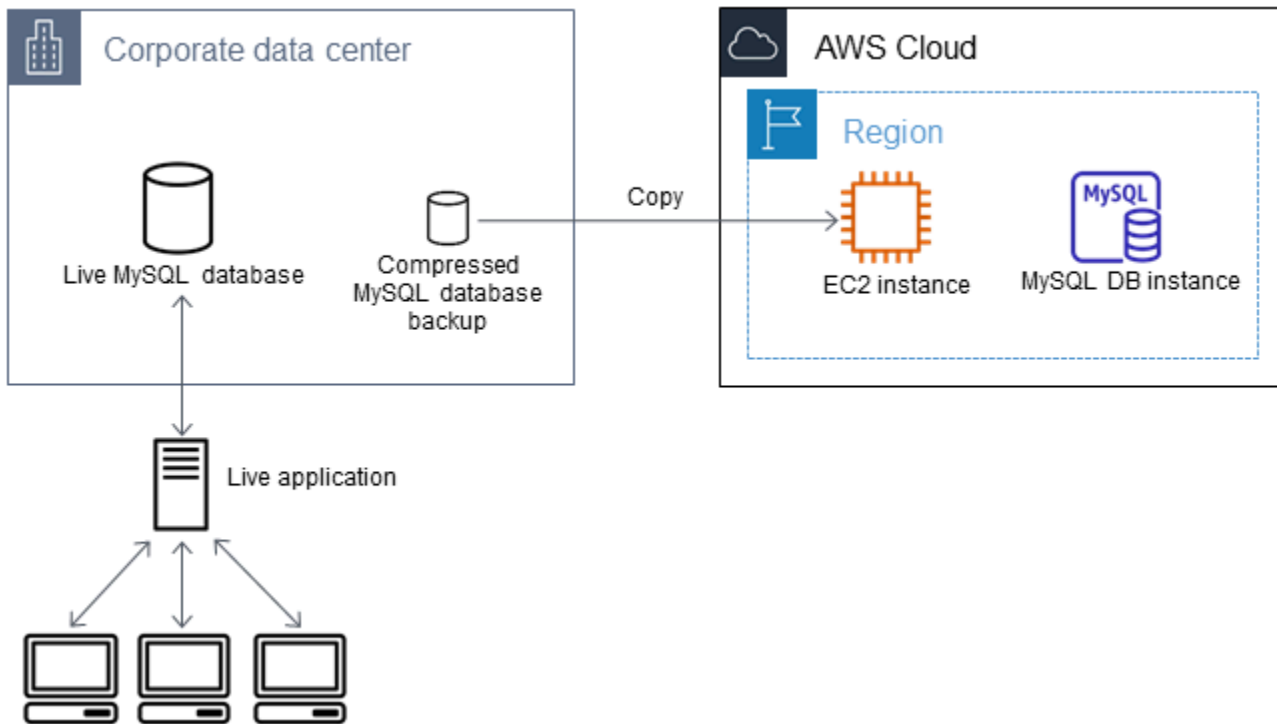
- 若要壓縮分隔文字輸出，請使用下列命令。

```
tar -zcvf backup.tar.gz target_directory
```

建立 Amazon EC2 執行個體並複製壓縮的資料庫

比起直接在資料庫執行個體間複製未壓縮的資料，將壓縮的資料庫備份檔複製到 Amazon EC2 執行個體，需要的網路資源較少。資料放進 Amazon EC2 後，即可將資料直接複製到 MariaDB 或 MySQL 資料庫。為了節省網路資源成本，Amazon EC2 執行個體必須與 Amazon RDS 資料庫執行個體位於同一

個 AWS 區域。將 Amazon EC2 執行個體放在與 Amazon RDS 資料庫相同的 AWS 區域中，也可以減少匯入期間的網路延遲。



建立 Amazon EC2 執行個體並複製資料

1. **AWS 區域** 在您打算建立 RDS 資料庫的位置中，建立虛擬私有雲端 (VPC)、VPC 安全性群組和 VPC 子網路。確認 VPC 安全群組的傳入規則會允許您應用程式連線至 AWS 所需的 IP 地址。您可以指定 IP 地址範圍 (例如 203.0.113.0/24) 或另一個 VPC 安全群組。您可以使用 [Amazon VPC 管理主控台](#) 來建立與管理 VPC 子網路和安全群組。如需詳細資訊，請參閱《Amazon Virtual Private Cloud 入門指南》中的 [Amazon VPC 入門](#)。
2. 開啟 [Amazon EC2 管理主控台](#)，然後選擇要同時包含您的 Amazon EC2 執行個體和 Amazon RDS 資料庫的 AWS 區域。使用您在步驟 1 中建立的 VPC、子網路和安全群組，啟動 Amazon EC2 執行個體。確認您選取的執行個體類型具有足夠的儲存空間，足以供未壓縮的資料庫備份檔使用。如需 Amazon EC2 執行個體的詳細資訊，請參閱《適用於 Linux 的 Amazon Elastic Compute Cloud 使用者指南》中的 [Amazon EC2 Linux 執行個體入門](#)。
3. 若要從 Amazon EC2 執行個體連線至 Amazon RDS 資料庫，請編輯您的 VPC 安全群組。新增會指定您 EC2 執行個體私有 IP 地址的傳入規則。您可在 EC2 主控台視窗中，從 Instance (執行個體) 窗格的 Details (詳細資訊) 分頁標籤找到私有 IP 地址。若要編輯 VPC 安全群組並新增傳入規則，請選取 EC2 主控台導覽窗格中的 Security Groups (安全群組)，接著選擇您的安全群組，然後新增 MySQL 或 Aurora 的傳入規則，指定 EC2 執行個體的私有 IP 地址。如需了解如何將傳入規則新增至 VPC 安全群組，請參閱《Amazon VPC 使用者指南》中的 [新增與移除規則](#)。

- 將壓縮的資料庫備份檔從本機系統複製到 Amazon EC2 執行個體。如有必要，請使用 `chmod` 命令，確認您擁有 Amazon EC2 執行個體的目標目錄寫入許可。您可使用 `scp` 命令或 Secure Shell (SSH) 用戶端來複製檔案。以下是範例。

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

⚠ Important

務必使用安全網路傳輸通訊協定來複製機密資料。

- 使用下列命令連線至您的 Amazon EC2 執行個體，並安裝最新更新與 MySQL 用戶端工具。

```
sudo yum update -y
sudo yum install mysql -y
```

如需詳細資訊，請參閱《Linux Amazon Elastic Compute Cloud 使用者指南》中的[連線到您的執行個體](#)。

⚠ Important

此範例會在 Amazon Linux 發行版的 Amazon Machine Image (AMI) 上安裝 MySQL 用戶端。若要在不同的發行版上安裝 MySQL 用戶端，例如 Ubuntu 或 Red Hat Enterprise Linux，則此範例將無法運作。如需安裝 MySQL 的資訊，請參閱 MySQL 文件中的[安裝和升級 MySQL](#)。

- 連線到 Amazon EC2 執行個體後，將您的資料庫備份檔解壓縮。範例如下。

- 若要將 SQL 輸出解壓縮，請使用下列命令。

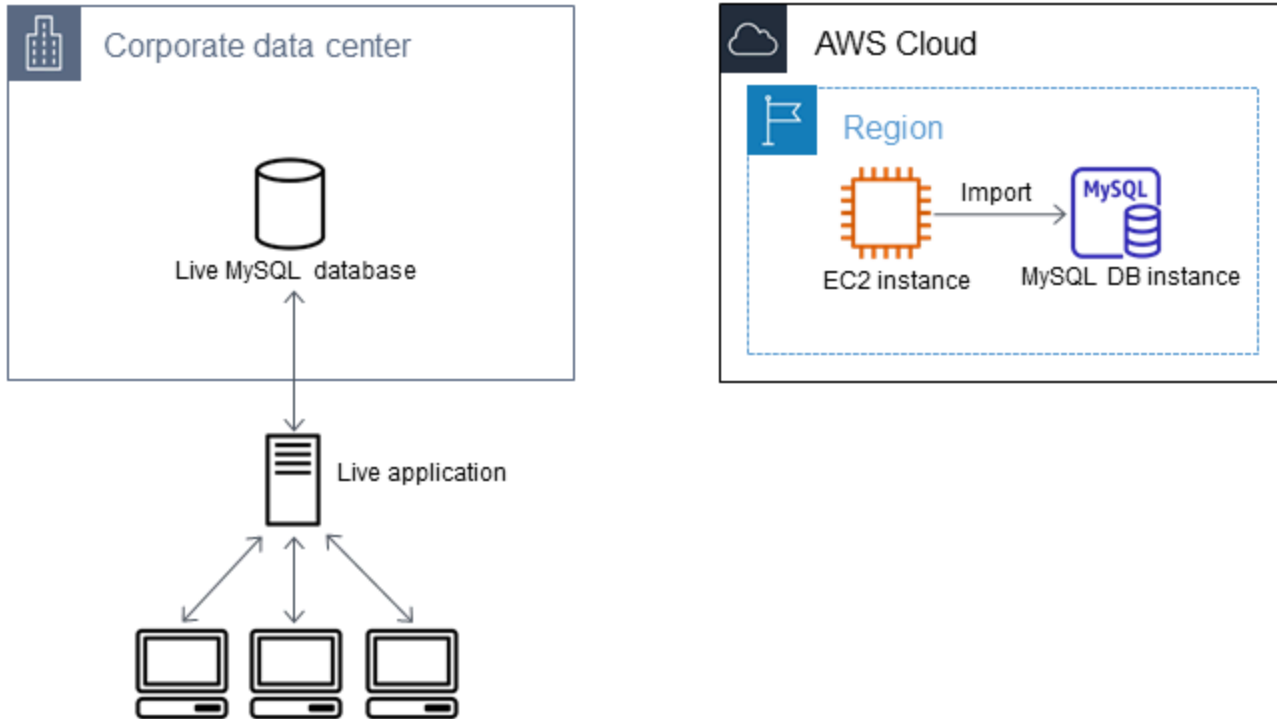
```
gzip backup.sql.gz -d
```

- 若要將分隔符號文字輸出解壓縮，請使用下列命令。

```
tar xzvf backup.tar.gz
```

建立 MySQL 或 MariaDB 資料庫，並從 Amazon EC2 執行個體匯入資料

透過在與 Amazon EC2 執行個體相同的區域中建立 MariaDB 資料庫執行個體、MySQL 資料庫執行個體或 MySQL 異 AWS 地同步備份資料庫叢集，您可以比透過網際網路更快地從 EC2 匯入資料庫備份檔案。



建立 MariaDB 或 MySQL 資料庫並匯入資料

1. 判斷需要何種資料庫執行個體類別及多大的儲存空間，才能支援這個 Amazon RDS 資料庫預計的工作負載。在此流程的過程中，決定資料載入程序需要多少空間與處理容量才夠。同時決定處理生產工作負載所需的資源。您可根據來源 MariaDB 或 MySQL 資料庫的大小與資源加以估計。如需詳細資訊，請參閱 [資料庫執行個體類別](#)。
2. 在包含 Amazon EC2 執行個體的區域中建立資料庫執行個體或異 AWS 地同步備份資料庫叢集。

若要建立 MySQL 多可用區域資料庫叢集，請遵循[建立多可用區域資料庫叢集](#)中的指示。

若要建立 MariaDB 或 MySQL 資料庫執行個體，請遵循[建立 Amazon RDS 資料庫執行個體](#)中的指示，並使用下列指導方針：

- 指定與您來源資料庫執行個體相容的資料庫引擎版本，如下所示：
 - 若您的來源執行個體為 MySQL 5.5.x 版，則 Amazon RDS 資料庫執行個體必須為 MySQL。

- 若您的來源執行個體為 MySQL 5.6.x 或 5.7.x 版，則 Amazon RDS 資料庫執行個體必須為 MySQL 或 MariaDB。
 - 如果您的來源執行個體為 MySQL 8.0.x 版本，則 Amazon RDS 資料庫執行個體必須為 MySQL 8.0.x 版本。
 - 若您的來源執行個體為 MariaDB 5.5 或更新版本，則 Amazon RDS 資料庫執行個體必須為 MariaDB。
3. 指定與 Amazon EC2 執行個體相同的虛擬私有雲端 (VPC) 和 VPC 安全群組。這種作法可確保您的 Amazon EC2 執行個體與 Amazon RDS 執行個體能在網路上看見彼此。確保您的資料庫執行個體可以公開存取。若要如稍後說明的方式，以您的來源資料庫設定複寫，必須將資料庫執行個體設為公開存取。
 4. 匯入資料庫備份前，請勿設定多個可用區域、備份保留期或僅供讀取複本。匯入完成後，您可以設定生產執行個體的多可用區域和備份保留。
 3. 檢視 Amazon RDS 資料庫的預設組態選項。如果資料庫的預設參數群組沒有您想要的組態選項，請尋找其他提供您所需選項的參數群組，或者建立新的參數群組。如需建立參數群組的詳細資訊，請參閱 [使用參數群組](#) 一文。
 4. 以主要使用者身分連線至新的 Amazon RDS 資料庫。建立支援需存取執行個體的管理員、應用程式和服務所需的使用者。Amazon RDS 資料庫的主機名稱是執行個體的 Endpoint (端點) 值 (不含連接埠編號)。例如，mysampled.123456789012.us-west-2.rds.amazonaws.com。端點值可在 Amazon RDS 管理主控台的資料庫詳細資訊中找到。
 5. 連線到您的 Amazon EC2 執行個體。如需詳細資訊，請參閱《Linux Amazon Elastic Compute Cloud 使用者指南》中的[連線到您的執行個體](#)。
 6. 使用 mysql 命令，以遠端主機的形式從 Amazon EC2 執行個體連線至 Amazon RDS 資料庫。以下是範例。

```
mysql -h host_name -P 3306 -u db_master_user -p
```

主機名稱是 Amazon RDS 資料庫端點。

7. 出現 mysql 提示時，請執行 source 命令並傳入您的資料庫傾印檔案名稱，以將資料載入 Amazon RDS 資料庫執行個體：
 - 若為 SQL 格式，請使用下列命令。

```
mysql> source backup.sql;
```

- 若為分隔符號文字格式，請先建立資料庫 (如果它不是您在設定 Amazon RDS 資料庫時建立的預設資料庫)。

```
mysql> create database database_name;  
mysql> use database_name;
```

接著建立資料表。

```
mysql> source table1.sql  
mysql> source table2.sql  
etc...
```

然後匯入資料。

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY  
' ,' ENCLOSED BY '"' LINES TERMINATED BY '\n';  
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY  
' ,' ENCLOSED BY '"' LINES TERMINATED BY '\n';  
etc...
```

若要提升效能，您可透過多個連線平行執行這些操作，如此您的所有資料表就會建立並隨後載入。

Note

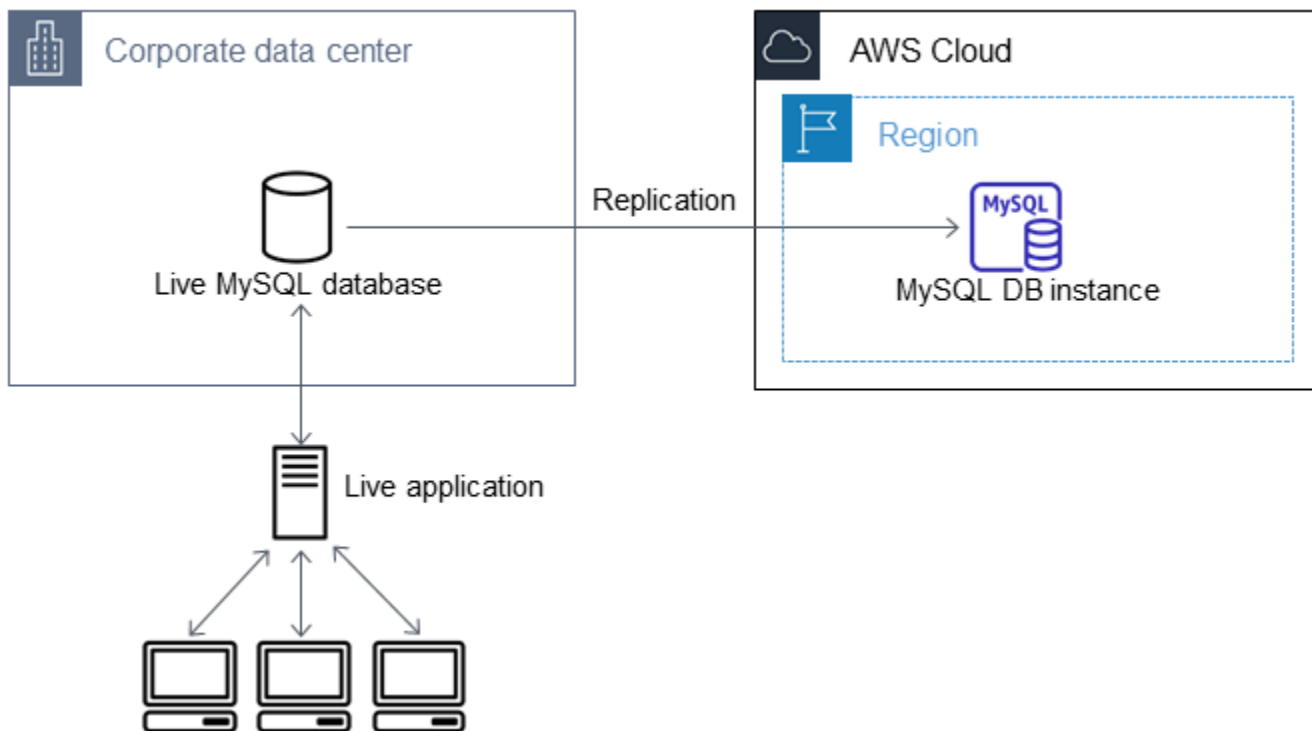
如果您在最初傾印資料表時使用 mysqldump 的任何資料格式化選項，請務必使用與 LOAD DATA LOCAL INFILE 相同的選項，以確保資料檔案內容的正確解譯。

8. 對匯入的資料庫中的一或兩個資料表執行簡單 SELECT 查詢，以確認匯入是否成功。

如果您不再需要此程序中使用的 Amazon EC2 執行個體，請終止 EC2 執行個體以減少 AWS 源使用量。若要終止 EC2 執行個體，請參閱《Amazon EC2 使用者指南》中的 [終止執行個體](#)。

在外部資料庫與新的 Amazon RDS 資料庫之間複寫資料

在複製資料並傳輸至 MariaDB 或 MySQL 資料庫期間，您的來源資料庫可能就會完成更新。因此，您可以使用複寫將複製的資料庫 up-to-date 與來源資料庫一起使用。



在 Amazon RDS 資料庫上啟動複寫功能所需的許可有其限制，不供 Amazon RDS 主要使用者使用。基於這個原因，請確認使用 Amazon RDS [mysql.rds_set_external_master](#) 命令或 [mysql.rds_set_external_master_gtid](#) 命令設定複寫功能，並使用 [mysql.rds_start_replication](#) 命令啟動在即時資料庫與 Amazon RDS 資料庫之間複寫的功能。

啟動複寫

稍早之前，您已開啟來源資料庫的二進位日誌，並設定獨一無二的伺服器 ID。現在，您可將 Amazon RDS 資料庫設為複本，並將即時資料庫設為來源複寫執行個體。

1. 在 Amazon RDS 管理主控台中，將託管來源資料庫之伺服器的 IP 地址，新增至 Amazon RDS 資料庫的 VPC 安全群組。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。

您可能還需要設定本機網路，以允許從 Amazon RDS 資料庫的 IP 地址建立連線，使其能與來源執行個體通訊。若要尋找 Amazon RDS 資料庫的 IP 地址，請使用 `host` 命令。

```
host rds_db_endpoint
```

主機名稱是來自 Amazon RDS 資料庫端點的 DNS 名稱，例如 `myinstance.123456789012.us-east-1.rds.amazonaws.com`。端點值可在 Amazon RDS 管理主控台的執行個體詳細資訊中找到。

2. 使用您選擇的用戶端，連線至來源執行個體，然後建立複寫作業所需的使用者。此帳戶只供複寫作業使用，務必限制其存取您的網域，以提升安全性。以下是範例。

MySQL 5.5、5.6 和 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY 'password';
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

3. 若為來源執行個體，請將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。舉例來說，若要將所有資料庫的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的「repl_user」使用者，請發出下列命令。

MySQL 5.5、5.6 和 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

4. 如果您使用 SQL 格式建立備份檔，且外部執行個體並非 MariaDB 10.0.24 或更新版本，請查看該檔案的內容。

```
cat backup.sql
```


該檔案包括 CHANGE MASTER TO 註解，內含主控端日誌檔案的名稱與位置。若您使用 mysqldump 命令的 --master-data 選項，備份檔內就會包含這個註解。請注意 MASTER_LOG_FILE 和 MASTER_LOG_POS 的值。

```
--  
-- Position to start replication or point-in-time recovery from  
--  
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

如果您使用分隔符號文字格式來建立備份檔，且外部執行個體並非 MariaDB 10.0.24 或更高版本，您應該已在本主題中的「建立現有資料庫的備份複本」程序的步驟 1 時就建立二進位日誌座標。

如果外部執行個體是 MariaDB 10.0.24 或更高版本，您應該已在本主題中的「建立現有資料庫的備份複本」程序的步驟 2 時就取得 GTID，以此啟動複寫作業。

5. 使 Amazon RDS 資料庫成為複本。如果外部執行個體並非 MariaDB 10.0.24 或更高版本，請以主要使用者身分連線至 Amazon RDS 資料庫，接著使用 [mysql.rds_set_external_master](#) 命令找到來源資料庫，將其視為來源複寫執行個體。如果您有 SQL 格式的備份檔，請使用您在上一步驟所確定的主控端日誌名稱與位置。或者，如果您使用的是分隔符號文字格式，請使用您在建立備份檔時確定的主控端日誌名稱與位置。以下是範例。

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,  
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```


Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

如果外部執行個體並非 MariaDB 10.0.24 或更高版本，請以主要使用者身分連線至 Amazon RDS 資料庫，接著使用 [mysql.rds_set_external_master_gtid](#) 命令找到來源資料庫，將其視為來源複寫執行個體。使用您在本主題中的「建立現有資料庫的備份複本」程序的步驟 2 中所確定的 GTID。以下是範例。

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,  
    'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` 是來源複寫執行個體的 IP 地址。目前不支援 EC2 私有 DNS 地址。


 Note

指定此處所顯示提示以外的憑證，作為安全最佳實務。

6. 在 Amazon RDS 資料庫上，發出 [mysql.rds_start_replication](#) 命令來啟動複寫。

```
CALL mysql.rds_start_replication;
```

7. 在 Amazon RDS 資料庫上，執行 [顯示複本狀態](#) 命令，以判斷複本何時 up-to-date 與來源複寫執行個體一起使用。SHOW REPLICA STATUS 命令的結果包括 Seconds_Behind_Master 欄位。當 Seconds_Behind_Master 欄位傳回 0 時，則複本會 up-to-date 包含來源複寫執行個體。

 Note

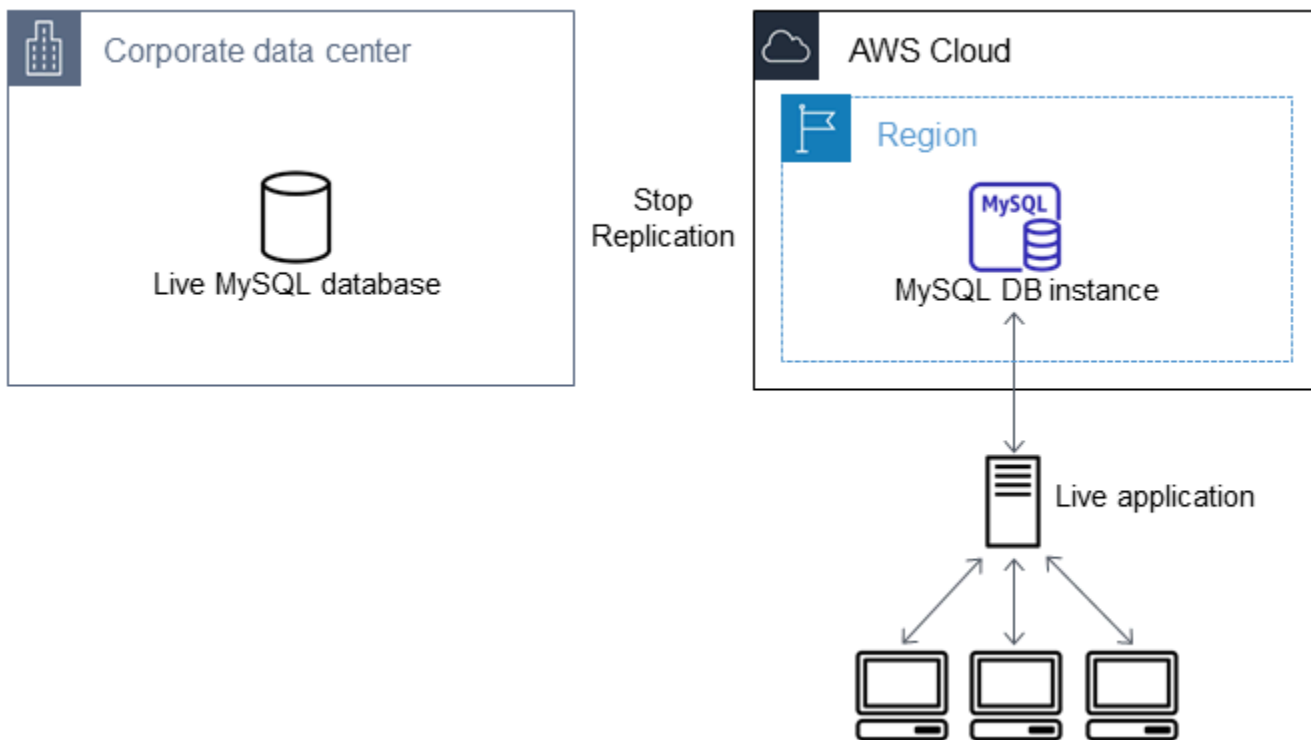
MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

若為 MariaDB 10.5、10.6 或 10.11 資料庫執行個體，請執行 [mysql.rds_replica_status](#) 程序，而非 MySQL 命令。

8. Amazon RDS 資料庫存放之後 up-to-date，請開啟自動備份，以便在需要時還原該資料庫。您可以使用 [Amazon RDS 管理主控台](#)，開啟或修改 Amazon RDS 資料庫的自動備份。如需詳細資訊，請參閱 [備份簡介](#)。

將您的即時應用程式重新導向 Amazon RDS 執行個體

在 MariaDB 或 MySQL 資料庫 up-to-date 與來源複寫執行個體一起使用之後，您現在可以更新即時應用程式以使用 Amazon RDS 執行個體。



將您的即時應用程式重新導向至 MariaDB 或 MySQL 資料庫並停止複寫

1. 若要新增 Amazon RDS 資料庫的 VPC 安全群組，請新增託管應用程式之伺服器的 IP 地址。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。
2. 確認 [\[顯示複本狀態\]](#) 命令結果中的 Seconds_Behind_Master 欄位是 0，表示複本 up-to-date 與來源複製執行個體一起使用。

```
SHOW REPLICA STATUS;
```

Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

若為 MariaDB 10.5、10.6 或 10.11 資料庫執行個體，請執行 [mysql.rds_replica_status](#) 程序，而非 MySQL 命令。

3. 交易完成時關閉所有與此來源的連線。

- 更新您的應用程式，以使用 Amazon RDS 資料庫。這裡所謂的更新，通常包括變更連線設定，以識別 Amazon RDS 資料庫的主機名稱和連接埠、要與其連線的使用者帳戶和密碼，以及要使用的資料庫。
- 連線到資料庫執行個體。

對至多可用區域資料庫叢集，請連線至寫入器資料庫執行個體。

- 使用 [mysql.rds_stop_replication](#) 命令停止 Amazon RDS 執行個體的複寫。

```
CALL mysql.rds_stop_replication;
```

- 在您的 Amazon RDS 資料庫上執行 [mysql.rds_reset_external_master](#) 命令來重設複寫組態，使得此執行個體不再視為複本。

```
CALL mysql.rds_reset_external_master;
```

- 開啟其他 Amazon RDS 功能，例如異地同步備份支援和僅供讀取複本。如需更多詳細資訊，請參閱 [設定及管理多可用區部署](#) 及 [使用資料庫執行個體僅供讀取複本](#)。

匯入任何 MariaDB 或 MySQL 資料庫執行個體來源的資料

建議您在資料載入之前和之後建立目標 Amazon RDS 資料庫執行個體的資料庫快照。Amazon RDS 資料庫快照是您資料庫執行個體的完整備份，可將您的資料庫執行個體還原至已知狀態。啟動資料庫快照作業時，資料庫執行個體的輸入/輸出操作會在資料庫備份時暫停。

若在載入前先建立資料庫快照，一旦有需要，就能將資料庫還原至載入前的狀態。若能在載入後立即拍攝資料庫快照，萬一發生事故，就不必重新載入資料，同時也能植入新的資料庫執行個體。

下列清單顯示要採取的步驟。下文將詳細討論各步驟。

- 建立一般檔案，其中需包含要載入的資料。
- 停止任何應用程式對目標資料庫執行個體的存取作業。
- 建立資料庫快照。
- 建議關閉 Amazon RDS 自動備份功能。
- 載入資料。
- 重新啟用自動備份。

步驟 1：建立一般檔案，其中需包含要載入的資料

使用逗號分隔值 (CSV) 等通用格式存放要載入的資料。每個資料表必須擁有各自的檔案；您不可將不同資料表的資料整併成同一檔案。以對應資料表的相同名稱命名各個檔案。副檔名可隨意指定。例如，如果資料表名稱為 `sales`，則檔案名稱可能是 `sales.csv` 或 `sales.txt`，但不是 `sales_01.csv`。

可以的話，依照預計載入之資料表的主索引鍵排序資料。這樣可以大幅縮短載入時間，並將磁碟儲存空間需求降至最低。

檔案越小，此程序的速度就會越快，效率越高。如有個別檔案未壓縮前的大小大於 1 GiB，請將檔案分割成多個檔案，再分別載入。

在泛 Unix 系統 (包括 Linux) 上，使用 `split` 命令。舉例來說，下列命令會將 `sales.csv` 檔案分割成多個小於 1 GiB 的檔案，且只會於分行處分割 (`-C 1024m`)。新檔案名稱為 `sales.part_00`、`sales.part_01`，以此類推。

```
split -C 1024m -d sales.csv sales.part_
```

其他作業系統也有類似的公用程式可用。

步驟 2：停止任何應用程式對目標資料庫執行個體的存取作業

開始載入大量資料前，請先停止所有應用程式活動，使其暫停存取您預計要載入資料的目標資料庫執行個體。如果有其他工作階段會修改要載入或參考的資料表，我們尤其建議執行這項操作。這麼做可以降低載入期間違反限制條件的風險，並提升載入效能。此外，還可以將資料庫執行個體還原至載入前的時間點，以免遺失非載入作業程序所做的變更。

當然，這可能無法盡如人願，或可能不切實際。如果您無法在載入作業前停止應用程式存取資料庫執行個體，請採取因應措施，確保資料的可用性與完整性。所需的確切步驟會因特定使用案例與現場需求而異。

步驟 3：建立資料庫快照

如果您預計要載入資料的新資料庫執行個體內沒有資料，您可略過此步驟。否則，如能建立資料庫執行個體的資料庫快照，即可在必要時，將資料庫執行個體還原至載入前的時間點。如前所述，啟動資料庫快照作業時，資料庫執行個體的輸入/輸出操作會在資料庫備份時暫停幾分鐘。

下列範例使用命 AWS CLI `create-db-snapshot` 令建立 AcmeRDS 執行個體的資料庫快照，並為資料庫快照提供識別碼 `"preload"`。

對於LinuxmacOS、或Unix：

```
aws rds create-db-snapshot \  
  --db-instance-identifier AcmeRDS \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds create-db-snapshot ^  
  --db-instance-identifier AcmeRDS ^  
  --db-snapshot-identifier preload
```

您也可以使用從資料庫快照還原的功能，建立試轉用的測試資料庫執行個體，或復原載入期間所做的變更。

請注意，如從資料庫快照還原資料庫，系統會建立新的資料庫執行個體，且如同所有資料庫執行個體一樣，該執行個體會具備獨一無二的識別碼與端點。若要還原資料庫執行個體，且不會變更端點，請先刪除資料庫執行個體，如此才能重新使用端點。

舉例來說，若要建立試轉或其他測試所需的資料庫執行個體，您必須賦予資料庫執行個體專屬識別符。在此例中，識別符是 *AcmeRDS-2*。範例使用與 *AcmeRDS-2* 關聯的端點連線至資料庫執行個體。

對於LinuxmacOS、或Unix：

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS-2 \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS-2 ^  
  --db-snapshot-identifier preload
```

若要重新使用現有端點，請先刪除資料庫執行個體，然後指定相同的識別符給還原的資料庫。

對於LinuxmacOS、或Unix：

```
aws rds delete-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --final-db-snapshot-identifier AcmeRDS-Final
```

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS \  
  --db-snapshot-identifier preload
```

在 Windows 中：

```
aws rds delete-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --final-db-snapshot-identifier AcmeRDS-Final  
  
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS ^  
  --db-snapshot-identifier preload
```

前述範例是在刪除資料庫執行個體前擷取資料庫的最終資料庫快照。此為選用操作，但建議您採用。

步驟 4：建議關閉 Amazon RDS 自動備份功能

Warning

如果您需要執行 point-in-time 復原，請勿關閉自動備份。

關閉自動備份會清除所有現有的備份，因此在關閉自動備份後無法進行 point-in-time 復原。停用自動備份功能可產生最佳化效能，但並非資料載入作業的必要功能。關閉自動備份功能不會影響手動資料庫快照。所有現有手動資料庫快照仍可用於復原作業。

關閉自動備份功能可將載入時間減少約 25%，並可降低載入期間所需的儲存空間。如果您預計載入資料的新資料庫執行個體不含任何資料，則要加快載入速度，並避免備份產生額外的儲存空間需求，關閉備份功能是最簡單的方式。不過，在某些情況下，您可能會計劃載入已包含資料的資料庫執行個體。如果是這樣，請權衡關閉備份與失去執行能力的影響所帶來的好處 point-in-time-recovery。

預設情況下，資料庫執行個體會開啟自動備份功能 (保留期為 1 天)。若要關閉自動備份，請將備份保留期設為零。載入後，您可以將備份保留期設為非零的值，以重新開啟自動備份功能。若要開啟或關閉備份功能，Amazon RDS 會關閉資料庫執行個體，然後重新啟動，以便開啟或關閉 MariaDB 或 MySQL 記錄功能。

使用指 AWS CLI `modify-db-instance` 令將備份保留設定為零，並立即套用變更。將保留期設為 0 需要重新啟動資料庫執行個體，請靜待重新啟動作業完成，再繼續操作。

對於Linux macOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --apply-immediately \  
  --backup-retention-period 0
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --apply-immediately ^  
  --backup-retention-period 0
```

您可以使用 AWS CLI `describe-db-instances` 指令檢查資料庫執行個體的狀態。以下範例顯示 `AcmeRDS` 資料庫執行個體的資料庫執行個體狀態。

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].  
{DBInstanceStatus:DBInstanceStatus}"
```

當資料庫執行個體狀態為 `available` 時，您即可繼續操作。

步驟 5：載入資料

使用 MySQL `LOAD DATA LOCAL INFILE` 陳述式將資料列從平面檔案讀取到資料庫資料表中。

下列範例說明如何將資料從名為的檔案載入 `sales.txt` 入資料庫 `Sales` 中名為的資料表。

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '  
  ENCLOSED BY '' ESCAPED BY '\\';  
Query OK, 1 row affected (0.01 sec)  
Records: 1 Deleted: 0 Skipped: 0 Warnings: 0
```

如需有關 `LOAD DATA` 陳述式的詳細資訊，請參閱 [MySQL 文件](#)。

步驟 6：重新開啟 Amazon RDS 自動備份功能

載入完成後，您可以將備份保留期設回載入前的值，以重新開啟 Amazon RDS 自動備份功能。如前所述，Amazon RDS 會重新啟動資料庫執行個體，所以短暫停機是正常現象。

下列範例會使用命 AWS CLI `modify-db-instance` 令來開啟 AcmeRDS 資料庫執行個體的自動備份，並將保留期限設定為一天。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --backup-retention-period 1 \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --backup-retention-period 1 ^  
  --apply-immediately
```

在 Amazon RDS 中使用 MySQL 複寫

您通常會使用僅供讀取複本來設定 Amazon RDS 資料庫執行個體間的複寫。如需僅供讀取複本的一般資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。如需在 Amazon RDS for MySQL 上使用僅供讀取複本的特定資訊，請參閱 [使用 MySQL 僅供讀取複本](#)。

對於使用 RDS for MySQL 的複寫，您可以使用全域交易識別符 (GTID)。如需詳細資訊，請參閱 [使用 GTID 式複寫](#)。

您也可以將 RDS for MySQL 資料庫執行個體及 Amazon RDS 外部的 MariaDB 或 MySQL 執行個體之間，設定複寫。如需使用外部來源設定複寫的詳細資訊，請參閱 [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。

對於以上任何複寫選項，您可以使用基於列的複寫、基於陳述式或混合式複寫。基於列的複寫只會複寫 SQL 陳述式所變更的列。基於陳述式的複寫會複寫整個 SQL 陳述式。混合式複寫會盡可能使用基於陳述式的複寫，但是當基於陳述式的複寫有不安全的 SQL 陳述式執行時，則會切換到基於列的複寫。在大多數情況下，建議使用混合式複寫。資料庫執行個體的二進位日誌格式決定複寫是基於列、基於陳述式或混合式。如需有關設定二進位日誌格式的資訊，請參閱 [設定適用於 MySQL RDS](#)。

Note

您可將複寫作業設定為從 Amazon RDS 外部的 MariaDB 或 MySQL 執行個體匯入資料庫，或者設定為將資料庫匯出至這類執行個體。如需更多詳細資訊，請參閱 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 及 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

主題

- [使用 MySQL 僅供讀取複本](#)
- [使用 GTID 式複寫](#)
- [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)
- [multi-source-replication 為適用於 MySQL 的 RDS 進行設定](#)

使用 MySQL 僅供讀取複本

接下來，您可以找到在 RDS for MySQL 上使用僅供讀取複本的特定資訊。如需僅供讀取複本的一般資訊及使用說明，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

主題

- [使用 MySQL 設定僅供讀取複本](#)
- [使用 MySQL 設定複寫篩選條件](#)
- [透過 MySQL 設定延遲複寫](#)
- [使用 MySQL 更新僅供讀取複本](#)
- [使用 MySQL 處理多可用區域僅供讀取複本的部署](#)
- [搭配 RDS for MySQL 使用階層式僅供讀取複本](#)
- [監控 MySQL 僅供讀取複本](#)
- [透過 MySQL 僅供讀取複本啟動及停用複寫作業](#)
- [對 MySQL 僅供讀取複本問題進行故障診斷](#)

使用 MySQL 設定僅供讀取複本

在 MySQL 資料庫執行個體可做為複製來源之前，請務必在來源資料庫執行個體上啟用自動備份。若要這麼做，請將備份保留期設定為 0 以外的值。若僅供讀取複本能讓其他僅供讀取複本做為來源資料庫執行個體，則亦適用此要求。執行 MySQL 任一版本的僅供讀取複本皆支援自動備份。您可以根據二進位日誌座標來設定複寫作業，以供 MySQL 資料庫執行個體使用。

在適用於 MySQL 的 RDS 版本 5.7.44 及更新版本上，您可以使用全域交易識別碼 (GTID) 來設定複寫。如需詳細資訊，請參閱 [使用 GTID 式複寫](#)。

您可以從相同區域內的一個資料庫執行個體建立至多 15 個僅供讀取複本。若希望複寫作業順利運作，每個僅供讀取複本具備的運算和儲存資源數量應與來源資料庫執行個體相同。若您擴展來源資料庫執行個體，也要擴展僅供讀取複本。

RDS for MySQL 支援階層式僅供讀取複本。若要了解如何設定階層式僅供讀取複本，請參閱 [搭配 RDS for MySQL 使用階層式僅供讀取複本](#)。

您可以同時執行多個僅供讀取複本建立，並刪除參考相同來源資料庫執行個體的動作。執行這些動作時，請將每個來源執行個體的僅供讀取複本數目限制在 15 個之內。

MySQL 資料庫執行個體的僅供讀取複本不能使用低於其來源資料庫執行個體的資料庫引擎版本。

準備使用 MyISAM 的 MySQL 資料庫執行個體

如果 MySQL 資料庫執行個體採用 MyISAM 等非交易儲存引擎，便需要執行下列步驟，藉以成功設定僅供讀取複本。請務必遵循以下步驟，才能確保僅供讀取複本的資料複本一致。然而，如果所有資料表均採用 InnoDB 等交易儲存引擎，就不需要進行這些步驟。

1. 在來源資料庫執行個體的非交易資料表上，停止所有資料處理語言 (DML) 和資料定義語言 (DDL) 操作，接著等候停止作業完成。SELECT 陳述式可以繼續執行。
2. 清空並鎖定來源資料庫執行個體中的資料表。
3. 使用以下各節的其中一種方法來建立僅供讀取複本。
4. 透過 DescribeDBInstances API 操作等方式，檢查僅供讀取複本的建立進度。僅供讀取複本可供使用時，請解除鎖定來源資料庫執行個體的資料表，並恢復資料庫正常操作。

使用 MySQL 設定複寫篩選條件

您可以使用複寫篩選條件來指定要與僅供讀取複本一起複寫的資料庫和資料表。複寫篩選條件可以包含複寫中的資料庫和資料表，或將其排除在複寫之外。

下列是複寫篩選條件的一些應用案例：

- 要縮小僅供讀取複本的大小。使用複寫篩選，您可以排除僅供讀取複本不需要的資料庫和資料表。
- 基於安全考量，要將資料庫和資料表從僅供讀取複本中排除。
- 為不同僅供讀取複本的特定應用案例複寫不同的資料庫和資料表。例如，您可以使用特定僅供讀取複本進行分析或分區。
- 對於具有不同僅供讀取複本的資料庫執行個體 AWS 區域，以複寫不同的資料庫或資料表。AWS 區域

Note

您也可以使用複寫篩選條件來指定要與主要 MySQL 資料庫執行個體 (設定為輸入複寫拓撲中的複寫) 一起複寫的資料庫和資料表。如需此組態的詳細資訊，請參閱「[使用外部來源執行個體設定二進位日誌檔案位置複寫](#)」。

主題

- [設定適用於 RDS for MySQL 的複寫篩選參數](#)
- [RDS for MySQL 的複寫篩選限制](#)
- [RDS for MySQL 的複寫篩選範例](#)
- [檢視僅供讀取複本的複寫篩選條件](#)

設定適用於 RDS for MySQL 的複寫篩選參數

要設定複寫篩選條件，請在僅供讀取複本上設定下列複寫篩選參數：

- `replicate-do-db` – 將變更複寫至指定的資料庫。當您為僅供讀取複本設定此參數時，只會複寫參數中指定的資料庫。
- `replicate-ignore-db` – 請勿將變更複寫至指定的資料庫。為僅供讀取複本設定 `replicate-do-db` 參數時，不會評估此參數。
- `replicate-do-table` – 將變更複製到指定的資料表。當您為僅供讀取複本設定此參數時，只會複寫參數中指定的資料表。此外，設定 `replicate-do-db` 或 `replicate-ignore-db` 參數時，請務必使用僅供讀取複本將包含指定資料表的資料庫納入複寫中。
- `replicate-ignore-table` – 請勿將變更複寫至指定的資料表。為僅供讀取複本設定 `replicate-do-table` 參數時，不會評估此參數。
- `replicate-wild-do-table` – 根據指定的資料庫和資料表名稱模式複寫資料表。支援 % 和 _ 萬用字元。設定 `replicate-do-db` 或 `replicate-ignore-db` 參數時，請務必使用僅供讀取複本將包含指定資料表的資料庫納入複寫中。
- `replicate-wild-ignore-table` – 請勿根據指定的資料庫和資料表名稱模式複寫資料表。支援 % 和 _ 萬用字元。為僅供讀取複本設定 `replicate-do-table` 或 `replicate-wild-do-table` 參數時，不會評估此參數。

系統會按照列出的順序對參數進行評估。如需有關這些參數如何運作的詳細資訊，請參閱 MySQL 文件：

- 如需一般資訊，請參閱[複本伺服器選項和變數](#)。
- 如需有關如何評估資料庫複寫篩選參數的資訊，請參閱[評估資料庫層級複寫和二進位日誌記錄選項](#)。
- 如需如何評估資料表複寫篩選參數的詳細資訊，請參閱[評估資料表層級複寫選項](#)。

根據預設，這些參數中的每個參數都有一個空值。在每個僅供讀取複本上，您可以使用這些參數來設定、變更和刪除複寫篩選條件。當您設定其中一個參數時，請使用逗號將每個篩選條件與其他篩選條件分隔。

您可以在 % 和 _ 參數中使用 `replicate-wild-do-table` 和 `replicate-wild-ignore-table` 萬用字元。% 萬用字元等同於任意數目的字元，而 _ 萬用字元只會等同於一個字元。

來源資料庫執行個體的二進位記錄格式對複寫非常重要，因為它會決定資料變更的記錄。binlog_format 參數的設定會決定複寫是以資料列為基礎還是以陳述式為基礎。如需詳細資訊，請參閱 [設定適用於 MySQL RDS](#)。

Note

無論來源資料庫執行個體上的 binlog_format 設定為何，所有資料定義語言 (DDL) 陳述式都會複寫為陳述式。

RDS for MySQL 的複寫篩選限制

下列限制適用於 RDS for MySQL 的複寫篩選：

- 每個複寫篩選參數都有 2,000 個字元的限制。
- 參數值的複寫篩選器不支援逗號。在參數清單中，逗號只能用作值分隔符號。例如，ParameterValue='`a,b`' 不支援，但 ParameterValue='a,b' 是。
- 不支援用於二進位日誌篩選的 MySQL --binlog-do-db 和 --binlog-ignore-db 選項。
- 複寫篩選不支援 XA 交易。

如需詳細資訊，請參閱 MySQL 文件中的 [XA 交易的限制](#)。

RDS for MySQL 的複寫篩選範例

若要設定僅供讀取複本的複寫篩選，請修改與僅供讀取複本關聯之參數群組中的複寫篩選參數。

Note

您無法修改預設參數群組。如果僅供讀取複本使用預設參數群組，請建立新的參數群組，並將它與僅供讀取複本建立關聯。如需資料庫參數群組的詳細資訊，請參閱 [使用參數群組](#)。

您可以使用 AWS Management Console、或 RDS API 在參數群組 AWS CLI 中設定參數。如需有關設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。當您在參數群組中設定參數時，與參數群組關聯的所有資料庫執行個體都會使用參數設定。如果您在參數群組中設定複寫篩選參數，請確定參數群組僅與僅供讀取複本相關聯。將來源資料庫執行個體的複寫篩選參數保留空白。

下列範例會使用 AWS CLI 設定參數。這些範例將 ApplyMethod 設定為 immediate，以便在 CLI 命令完成後立即發生參數變更。如果您想要在僅供讀取複本重新啟動後套用擱置變更，請將設定 ApplyMethod 為 pending-reboot。

下列範例會設定複寫篩選條件：

- [Including databases in replication](#)
- [Including tables in replication](#)
- [Including tables in replication with wildcard characters](#)
- [Excluding databases from replication](#)
- [Excluding tables from replication](#)
- [Excluding tables from replication using wildcard characters](#)

Example 在複寫中包含資料庫

下列範例包含複寫中的 mydb1 和 mydb2 資料庫。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-do-  
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myparametergroup ^  
  --parameters "ParameterName=replicate-do-  
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Example 在複寫中包含資料表

下列範例包含複寫資料庫 table1 中的 table2 和 mydb1 資料表。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-do-  
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myparametergroup ^  
  --parameters "ParameterName=replicate-do-  
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Example 使用萬用字元在複寫中包含資料表

下列範例包含複寫時在資料庫 `order` 中名稱開頭為 `return` 和 `mydb` 的資料表。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order  
%,mydb.return%',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myparametergroup ^  
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order  
%,mydb.return%',ApplyMethod=immediate"
```

Example 從複寫中排除資料庫

下列範例會從複寫中排除 `mydb5` 和 `mydb6` 資料庫。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```



```
--parameters "ParameterName=replicate-ignore-  
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myparametergroup ^  
  --parameters "ParameterName=replicate-ignore-  
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Example 從複寫中排除資料表

下列範例會從複寫中排除資料庫 mydb5 中的資料表 table1 和資料庫 mydb6 中的 table2。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-ignore-  
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myparametergroup ^  
  --parameters "ParameterName=replicate-ignore-  
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Example 使用萬用字元從複寫中排除資料表

下列範例會從複寫中排除資料庫 order 中名稱開頭為 return 和 mydb7 的資料表。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myparametergroup \  
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order  
%,mydb7.return%',ApplyMethod=immediate"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

檢視僅供讀取複本的複寫篩選條件

您可以使用下列方式檢視僅供讀取複本的複寫篩選條件：

- 檢查與僅供讀取複本關聯之參數群組中複寫篩選參數的設定。

如需說明，請參閱「[檢視資料庫參數群組的參數值](#)」。

- 在 MySQL 用戶端中，連線至僅供讀取複本並執行 SHOW REPLICA STATUS 陳述式。

在輸出中，下列欄位會顯示僅供讀取複本的複寫篩選條件：

- Replicate_Do_DB
- Replicate_Ignore_DB
- Replicate_Do_Table
- Replicate_Ignore_Table
- Replicate_Wild_Do_Table
- Replicate_Wild_Ignore_Table

如需有關這些欄位的詳細資訊，請參閱 MySQL 文件中的[檢查複寫狀態](#)。

Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

透過 MySQL 設定延遲複寫

您可以將延遲複寫做為災難復原的策略。利用延遲複寫功能來指定最短時間 (以秒為單位)，即可延遲來源到僅供讀取複本的複寫作業。在發生損毀之時 (例如不小心刪除資料表)，若要快速完成事後復原作業，則請執行下列步驟：

- 停止僅供讀取複本的複寫作業，以免系統將造成損毀的變更內容傳送到該複本。

使用 [mysql.rds_stop_replication](#) 預存程序來停止複寫作業。

- 啟動複寫作業並指定日誌檔案位置，則當系統複寫到該位置時，即會自動停止作業。

透過 [mysql.rds_start_replication_until](#) 預存程式，藉此在發生損毀前指定日誌檔案位置。

- 若要將僅供讀取複本提升為新的來源資料庫執行個體，請參照[提升僅供讀取複本為獨立的資料庫執行個體](#)中的指示。

Note

- 在 RDS for MySQL 8.0 版上，MySQL 8.0.28 版和更新版本可支援延遲複寫。在適用於 MySQL 5.7 的 RDS 上，MySQL 5.7.44 及更新版本支援延遲複寫。
- 使用預存程序來設定延遲複寫。您無法使用 AWS Management Console、或 Amazon RDS API 設定延遲複寫。AWS CLI
- 在適用於 MySQL 5.7.44 及更高版本的 MySQL 5.7 版和 MySQL 8.0.28 及更新版本 8.0 版本上，您可以在延遲複寫組態中使用以全域交易識別碼 (GTID) 為基礎的複寫。如果您使用 GTID 式複寫，請使用 [mysql.rds_start_replication_until_gtid](#) 預存程序，而非 [mysql.rds_start_replication_until](#) 預存程序。如需 GTID 式複寫的詳細資訊，請參閱[使用 GTID 式複寫](#)。

主題

- [在僅供讀取複本建立期間設定延遲複寫](#)
- [修改現有僅供讀取複本的延遲複寫](#)
- [設定位置以停止僅供讀取複本的複寫作業](#)
- [提升僅供讀取複本](#)

在僅供讀取複本建立期間設定延遲複寫

若要為任何未來從資料庫執行個體建立的僅供讀取複本設定延遲複寫，請利用 [mysql.rds_set_configuration](#) 參數來執行 target delay 預存程序。

在僅供讀取複本建立期間設定延遲複寫

1. 透過 MySQL 用戶端，以主要使用者的身分連接至 MySQL 資料庫執行個體，該執行個體將成為僅供讀取複本的來源。

2. 利用 [mysql.rds_set_configuration](#) 參數來執行 target delay 預存程序。

例如，您可以執行下列預存程序，藉此設定複寫至少會延遲一小時 (3600 秒)，並將該設定套用至目前資料庫執行個體所建立的任何僅供讀取複本。

```
call mysql.rds_set_configuration('target delay', 3600);
```

Note

執行此預存程序後，您使用 AWS CLI 或 Amazon RDS API 建立的任何僅供讀取複本，都會設定複寫延遲指定的秒數。

修改現有僅供讀取複本的延遲複寫

若要修改現有僅供讀取複本的延遲複寫，請執行 [mysql.rds_set_source_delay](#) 預存程序。

修改現有僅供讀取複本的延遲複寫

1. 透過 MySQL 用戶端，以主要使用者的身分連接至僅供讀取複本。
2. 使用 [mysql.rds_stop_replication](#) 預存程序來停止複寫作業。
3. 執行 [mysql.rds_set_source_delay](#) 預存程序。

例如，您可以執行下列預存程序，藉此設定複寫至少會延遲一小時 (3600 秒)，並將該設定套用至僅供讀取複本。

```
call mysql.rds_set_source_delay(3600);
```

4. 使用 [mysql.rds_start_replication](#) 預存程序來啟動複寫作業。

設定位置以停止僅供讀取複本的複寫作業

停止僅供讀取複本的複寫作業後，您可以使用 [mysql.rds_start_replication_until](#) 預存程序來啟動複寫，然後在特定的二進位日誌檔案位置停止複寫。

啟動僅供讀取複本的複寫作業，並在特定位置停止複寫

1. 透過 MySQL 用戶端，以主要使用者的身分連接至來源 MySQL 資料庫執行個體。
2. 執行 [mysql.rds_start_replication_until](#) 預存程序。

以下範例會啟動複寫並複寫變更，直到達到 120 二進位日誌檔案中的位置 `mysql-bin-changelog.000777` 為止。若要使用災難復原功能，請在發生損毀前將位置預設為 120。

```
call mysql.rds_start_replication_until(  
  'mysql-bin-changelog.000777',  
  120);
```

達到停止點時，複寫作業即會自動停止。而且，系統還會產生以下 RDS 事件：Replication has been stopped since the replica reached the stop point specified by the `rds_start_replication_until` stored procedure.

提升僅供讀取複本

使用災難復原功能時，您可以參照中的指示，在複寫作業停止後將僅供讀取複本提升為新的來源資料庫執行個體。如需提升僅供讀取複本的相關資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

使用 MySQL 更新僅供讀取複本

僅供讀取複本專門用來支援讀取查詢，但建議您不定期進行更新。例如，您可能需要新增索引，將特定查詢類型存取複本的速度最佳化。

雖然您可以透過在資料庫參數群組中，針對讀取複本將 `read_only` 參數設為 0 來啟用更新，但我們建議您不要這樣做，因為這樣可能會在讀取複本與來源資料庫執行個體不相容的情況下造成問題。對於維護操作，建議您使用藍/綠部署。如需詳細資訊，請參閱 [使用藍/綠部署進行資料庫更新](#)。

如果您停用僅供讀取複本的唯一功能，請盡快將 `read_only` 參數的值恢復為 1。

使用 MySQL 處理多可用區域僅供讀取複本的部署

您可以從單一可用區域或多可用區域的資料庫執行個體部署建立僅供讀取複本。您可使用異地同步備份部署來改善重要資料的耐用性和可用性，但無法使用異地同步備份部署來處理僅供讀取的查詢。反之，您可從高流量的多可用區域資料庫執行個體建立僅供讀取複本，藉此卸載僅供讀取的查詢。若異地同步備份部署的來源執行個體容錯移轉為次要，任何相關聯的僅供讀取複本會自動將複寫作業來源切換成次要（現為主要）。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。

您可以建立僅供讀取複本並用作多可用區域資料庫執行個體。Amazon RDS 會在另一個可用區域中建立您的複本的待命複本，以獲得該複本的容錯移轉支援。建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。

搭配 RDS for MySQL 使用階層式僅供讀取複本

RDS for MySQL 支援階層式僅供讀取複本。利用階層式僅供讀取複本，您可以擴展讀取，無須增加來源 RDS for MySQL 資料庫執行個體的額外負荷。

利用階層式僅供讀取複本，您的 RDS for MySQL 資料庫執行個體會將資料傳送至鏈結中的第一個僅供讀取複本。之後，該僅供讀取複本會將資料傳送至鏈結中的第二個複本，依此類推。最終結果是鏈結中的所有僅供讀取複本皆具有來自 RDS for MySQL 資料庫執行個體的變更，但並無僅在來源資料庫執行個體上的額外負荷。

您可從來源 RDS for MySQL 資料庫執行個體的鏈結中建立一系列最多三個僅供讀取複本。例如，假設您有一個 RDS for MySQL 資料庫執行個體 `mysql-main`。您可以執行下列作業：

- 從 `mysql-main` 開始，建立鏈結中的第一個僅供讀取複本 `read-replica-1`。
- 接下來，從 `read-replica-1`，建立鏈結中的下一個僅供讀取複本 `read-replica-2`。
- 最後，從 `read-replica-2`，建立鏈結中的第三個僅供讀取複本 `read-replica-3`。

除了 `mysql-main` 系列中的第三個階層式僅供讀取複本之外，您無法建立另一個僅供讀取複本。從 RDS for MySQL 來源資料庫執行個體至一系列階層式僅供讀取複本尾端的完整執行個體系列最多可包含四個資料庫執行個體。

若要使階層式僅供讀取複本可以運作，每個來源 RDS for MySQL 資料庫執行個體都必須開啟自動備份。若要在僅供讀取複本上開啟自動備份，首先建立該僅供讀取複本，然後修改該複本來開啟自動備份。如需詳細資訊，請參閱 [建立僅供讀取複本](#)。

與任何僅供讀取複本一樣，您可提升作為階層式一部分的僅供讀取複本。從僅供讀取複本鏈結中提升僅供讀取複本將會從鏈結中移除該僅供讀取複本。例如，假設您想要將部分工作負載從 `mysql-main` 資料庫執行個體移至新的執行個體，僅供會計部門使用。假設範例中的三個僅供讀取複本鏈結，您決定提升 `read-replica-2`。該鏈或受到下列影響：

- 提升 `read-replica-2` 會將其從複寫鏈結中移除。
 - 其現在是一個完整的讀取/寫入資料庫執行個體。
 - 這會持續複寫至 `read-replica-3`，就像在提升之前所做的一樣。
- 您的 `mysql-main` 會持續複寫至 `read-replica-1`。

如需提升僅供讀取複本的相關詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

監控 MySQL 僅供讀取複本

對於 MySQL 僅供讀取複本，您可以 CloudWatch 透過檢視 Amazon RDS ReplicaLag 指標來監控 Amazon 中的複寫延遲。ReplicaLag 指標會回報 Seconds_Behind_Master 命令的 SHOW REPLICA STATUS 欄位值。

Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

MySQL 複寫延遲的常見原因如下：

- 網路停機。
- 在僅供讀取複本上寫入的資料表有不同索引。僅供讀取複本上的 read_only 參數設為 0 時，若僅供讀取複本與來源資料庫執行個體變得不相容，則複寫可能會中斷。在僅供讀取複本上執行維護作業後，建議您將 read_only 參數設回 1。
- 使用非交易儲存引擎 (例如 MyISAM)。複寫功能僅支援 MySQL 上的 InnoDB 儲存引擎。

當 ReplicaLag 指標到達 0，複本即已跟上來源資料庫執行個體。如果 ReplicaLag 指標傳回 -1，表示複寫目前為非作用中。ReplicaLag = -1 相當於 Seconds_Behind_Master = NULL。

透過 MySQL 僅供讀取複本啟動及停用複寫作業

您可呼叫系統預存程序 [mysql.rds_stop_replication](#) 和 [mysql.rds_start_replication](#)，藉此停止並重新啟動 Amazon RDS 資料庫執行個體的複寫程序。針對長時間操作 (如建立大型索引) 在兩個 Amazon RDS 執行個體間進行複寫時，即可採取此操作。匯入或匯出資料庫時，也必須停止並啟動複寫作業。如需詳細資訊，請參閱 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 及 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

若複寫作業停止連續超過 30 天 (不論手動停止或由於複寫錯誤)，Amazon RDS 會終止來源資料庫執行個體和所有僅供讀取複本間的複寫作業。這樣做的原因在於，避免來源資料庫執行個體的儲存需求增加以及長期容錯移轉。僅供讀取複本資料庫執行個體仍可供使用。然而，無法恢復複寫作業，因為在複寫作業終止後，僅供讀取複本所需的二進位日誌已從來源資料庫執行個體刪除。您可建立新的僅供讀取複本供來源資料庫執行個體使用，以重新建立複寫作業。

對 MySQL 僅供讀取複本問題進行故障診斷

使用 MySQL DB 時，僅供讀取複本偶爾會出現複寫錯誤，或僅供讀取複本和來源資料庫執行個體間出現資料不一致的情況 (或兩者同時發生)。若您在僅供讀取複本或來源資料庫執行個體失敗的期間，沒有清空部分二進位日誌 (binlog) 事件或 InnoDB 重做日誌，就會出現上述問題。在這種情況下，手動刪除並重新建立僅供讀取複本。如果要降低此情況發生的機率，則可設定以下參數值：`sync_binlog=1` 和 `innodb_flush_log_at_trx_commit=1`。但這些設定可能會降低效能，所以請先測試這些設定所帶來的影響，再於生產環境中實作變更內容。

Warning

在與來源資料庫執行個體相關聯的參數群組中，建議保留以下參數值：`sync_binlog=1` 和 `innodb_flush_log_at_trx_commit=1`。這些參數是動態參數。如果您不想使用這些設定，建議在對來源資料庫執行個體執行任何可能導致重新啟動的操作之前，暫時設定這些值。這些操作包括但不限於重新啟動、透過容錯移轉重新啟動、升級資料庫版本，以及變更資料庫執行個體類或其儲存體。這些建議同樣適用於為來源資料庫執行個體建立新的僅供讀取複本時。

若未遵循此指南的說明操作，會增加僅供讀取複本出現複寫錯誤，或僅供讀取複本和來源資料庫執行個體間出現資料不一致的情況 (或兩者同時發生) 的風險。

MySQL 採用非同步複寫技術，因此，來源資料庫執行個體的 `BinLogDiskUsage` 和僅供讀取複本上的 `ReplicaLag` 預料會偶爾增加。例如，來源資料庫執行個體可同時出現大量寫入操作。相對而言，僅供讀取複本的寫入操作則使用單一輸入/輸出執行緒序列化，這可能導致來源執行個體和僅供讀取複本之間的延遲。如需唯讀複本的詳細資訊，請參閱 MySQL 文件中的 [複寫實作詳細資訊](#)。

您可執行下列動作，減少來源資料庫執行個體的更新與僅供讀取複本的后續更新間的延遲：

- 將僅供讀取複本的儲存體大小和資料庫執行個體類別大小，設定為等同於來源資料庫執行個體。
- 確保來源資料庫執行個體和僅供讀取複本所使用之資料庫參數群組中的參數設定相容。如需詳細資訊和範例，請參閱本節稍後討論的 `max_allowed_packet` 參數。

Amazon RDS 會監控複寫您的僅供讀取複本的狀態，並在複寫因任何原因停止時，將僅供讀取複本執行個體上的 `Replication State` (複寫狀態) 欄位更新為 `Error` (錯誤)。一個可能的範例為，僅供讀取複本上執行的 DML 查詢，與來源資料庫執行個體的更新相衝突。

您可以檢視 `Replication Error` 欄位，藉此檢閱 MySQL 引擎擲出的相關聯錯誤詳細資訊。也會產生指出僅供讀取複本之狀態的事件，包括 [RDS-EVENT-0045](#)、[RDS-EVENT-0046](#) 和 [RDS-](#)

[EVENT-0047](#)。如需事件和訂閱事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。如果傳回 MySQL 錯誤訊息，請在 [MySQL 錯誤訊息文件](#) 中檢閱錯誤號碼。

其中一個常見問題就是僅供讀取複本的 `max_allowed_packet` 參數值小於來源資料庫執行個體的 `max_allowed_packet` 參數值，因而導致複寫錯誤。`max_allowed_packet` 參數是您可以在資料庫參數群組中設定的自訂參數。您可以使用 `max_allowed_packet` 以指定可以在資料庫上執行的 DML 程式碼的大小上限。在部分案例中，與僅供讀取複本建立關聯之資料庫參數群組中的 `max_allowed_packet` 值，會小於與來源資料庫執行個體建立關聯之資料庫參數群組中的 `max_allowed_packet` 值。在這些情況下，複寫程序可能會擲回錯誤 `Packet bigger than 'max_allowed_packet' bytes` 並停止複寫。若要修正錯誤，可讓來源資料庫執行個體和僅供讀取複本使用具有相同 `max_allowed_packet` 參數值的資料庫參數群組。

可能造成複寫錯誤的其他常見情況包括下列：

- 寫入僅供讀取複本上的資料表。在某些情況下，您可能會在僅供讀取複本上建立與來源資料庫執行個體索引不同的索引。如果您這樣做，請將 `read_only` 參數設定為 0 以建立索引。當寫入僅供讀取複本上的資料表時，若僅供讀取複本與來源資料庫執行個體變得不相容，則複寫可能會中斷。在僅供讀取複本上執行維護作業後，建議您將 `read_only` 參數設回 1。
- 使用非交易儲存引擎 (例如 MyISAM)。僅供讀取複本需要交易儲存引擎。複寫功能僅支援 MySQL 上的 InnoDB 儲存引擎。
- 使用不安全的非確定性查詢 (例如 `SYSDATE()`)。如需詳細資訊，請參閱[二進位日誌中安全和不安全陳述式的判定](#)。

若您認為能夠安全略過錯誤，請遵循[略過目前複寫錯誤](#)中所述的步驟。否則，您可以先刪除僅供讀取複本。您可以使用的相同資料庫執行個體識別符建立執行個體，使得端點可與您的舊僅供讀取複本端點保持相同。如果複寫錯誤已修復，Replication State (複寫狀態) 會變更為 `replicating` (複寫中)。

使用 GTID 式複寫

以下內容說明如何在 Amazon RDS for MySQL 資料庫執行個體之間使用全域交易識別碼 (GTID) 搭配二進位日誌 (binlog) 複寫。

如果您使用 binlog 複寫，而且不熟悉 MySQL 以 GTID 為基礎的複寫，請參閱 MySQL 文件中的[使用全域交易識別碼進行複寫](#)。

所有 RDS for MySQL 5.7 版，以及 RDS for MySQL 8.0.26 版和更新的 8.0 版本都支援 GTID 式複寫。複寫組態中的所有 MySQL 資料庫執行個體都必須符合此要求。

主題

- [全域交易識別符 \(GTID\) 的概觀](#)
- [GTID 式複寫的參數](#)
- [為新的僅供讀取複本設定 GTID 式複寫](#)
- [為現有的僅供讀取複本設定 GTID 式複寫](#)
- [為具備僅供讀取複本的 MySQL 資料庫執行個體停用 GTID 式複寫](#)

全域交易識別符 (GTID) 的概觀

「全域交易識別符 (GTID)」是系統為遞交的 MySQL 交易所產生的唯一識別符。GTID 能讓 binlog 複寫的操作更簡單，也更容易進行故障診斷。

進行 binlog 複寫作業時，MySQL 會使用兩種不同類型的交易：

- GTID 交易 – 透過 GTID 識別的交易所。
- 匿名交易 – 未指派 GTID 的交易所。

在複寫組態中，全部的資料庫執行個體都有各自不同的 GTID。GTID 可簡化複寫組態，因為使用時不需要參照日誌檔案位置。GTID 也使得追蹤複寫的交易更容易，而且可決定來源執行個體和複本是否一致。

您可使用 GTID 式複寫來複寫 RDS for MySQL 僅供讀取複本的資料。您可在建立新的僅供讀取複本時設定 GTID 式複寫，也可轉換現有的僅供讀取複本以使用 GTID 式複寫。


如果是 RDS for MySQL，您也可以延遲的複寫組態中使用 GTID 式複寫。如需詳細資訊，請參閱 [透過 MySQL 設定延遲複寫](#)。

GTID 式複寫的參數

使用以下參數來設定 GTID 式複寫。

參數	有效值	描述
gtid_mode	OFF, OFF_PERMISSIVE , ON_PERMISSIVE , ON	OFF 指定新交易是匿名交易 (也就是沒有 GTID)，而且交易必須是匿名交易才能複寫。 OFF_PERMISSIVE 指定新交易是匿名交易，但全部交易都可以複寫。

參數	有效值	描述
		<p>ON_PERMISSIVE 指定新交易是 GTID 交易，而且全部交易都可以複寫。</p> <p>ON 指定新交易是 GTID 交易，而且交易必須是 GTID 交易才能複寫。</p>
enforce_gtid_consistency	OFF, ON, WARN	<p>OFF 允許交易違反 GTID 一致性。</p> <p>ON 可避免交易違反 GTID 一致性。</p> <p>WARN 允許交易違反 GTID 一致性，但會在出現違反行為時產生警告。</p>

 Note

在中 AWS Management Console，gtid_mode 參數會顯示為 gtid-mode。

進行 GTID 式複寫時，您可以使用下列設定來配置資料庫執行個體或僅供讀取複本的參數群組：

- ON 和 ON_PERMISSIVE 僅適用於從 RDS 資料庫執行個體的傳出複寫作業。這兩個值都會導致 RDS 資料庫執行個體使用 GTID 以供複寫的交易使用。ON 要求目標資料庫也使用 GTID 式複寫。ON_PERMISSIVE 使 GTID 式複寫可在目標資料庫上選用。
- 若設定 OFF_PERMISSIVE，表示 RDS 資料庫執行個體可接受來源資料庫的傳入複寫作業。無論來源資料庫是否使用 GTID 式複寫，皆可進行。
- 若設定 OFF，表示唯有來源資料庫未使用 GTID 式複寫的情況下，RDS 資料庫執行個體才能接受來自該資料庫的傳入複寫作業。

如需參數群組的詳細資訊，請參閱[使用參數群組](#)。

為新的僅供讀取複本設定 GTID 式複寫

當 RDS for MySQL 資料庫執行個體的 GTID 式複寫處於啟用狀態時，系統會自動為資料庫執行個體的僅供讀取複本設定 GTID 式複寫。

為新的僅供讀取複本啟用 GTID 式複寫

1. 確定與資料庫執行個體有關的參數群組有下列參數設定：

- `gtid_mode` – ON 或 ON_PERMISSIVE
- `enforce_gtid_consistency` – ON

如需使用參數群組設定組態參數的詳細資訊，請參閱[使用參數群組](#)。

2. 如果已變更資料庫執行個體的參數群組，請重新啟動資料庫執行個體。如需如何進行的詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。
3. 建立資料庫執行個體的一個或多個僅供讀取複本。如需如何進行的詳細資訊，請參閱 [建立僅供讀取複本](#)。

Amazon RDS 嘗試使用 `MASTER_AUTO_POSITION`，在 MySQL 資料庫執行個體與僅供讀取複本之間建立 GTID 式複寫。如果嘗試失敗，Amazon RDS 會使用日誌檔案位置進行僅供讀取複本的複寫。如需 `MASTER_AUTO_POSITION` 的詳細資訊，請參閱 [GTID 自動定位功能](#)。

為現有的僅供讀取複本設定 GTID 式複寫

如果現有的 MySQL 資料庫執行個體具備僅供讀取複本，且並未使用 GTID 式複寫，則您可設定資料庫執行個體和僅供讀取複本間的 GTID 式複寫。

為現有的僅供讀取複本啟用 GTID 式複寫

1. 如果資料庫執行個體或任何僅供讀取複本正在使用低於 8.0.26 版的 RDS for MySQL 8.0 版，請將資料庫執行個體或僅供讀取複本升級為 8.0.26 版或更新的 MySQL 8.0 版本。所有 RDS for MySQL 5.7 版都支援 GTID 式複寫。

如需詳細資訊，請參閱 [升級 MySQL 資料庫引擎](#)。

2. (選用) 重設 GTID 參數，並測試資料庫執行個體和僅供讀取複本的行為：

- a. 確定與資料庫執行個體和每個僅供讀取複本相關聯的參數群組有設定為 `enforce_gtid_consistency` 的 `WARN` 參數。

如需使用參數群組設定組態參數的詳細資訊，請參閱[使用參數群組](#)。

- b. 如果已變更資料庫執行個體的參數群組，請重新啟動資料庫執行個體。如果已變更僅供讀取複本的參數群組，請重新啟動僅供讀取複本。

如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

- c. 對於一般的工作負載執行資料庫執行個體和僅供讀取複本，並監視日誌檔案。

如果您看見 GTID 不相容交易的警告，請調整應用程式，以便僅使用 GTID 相容功能。確定資料庫執行個體未產生與 GTID 不相容交易的任何警告，再進行下一個步驟。

3. 對於允許匿名交易的 GTID 式複寫重設 GTID 參數，直到僅供讀取複本處理全部的參數為止。
 - a. 確定與資料庫執行個體有關的參數群組和每個僅供讀取複本有下列參數設定：
 - `gtid_mode – ON_PERMISSIVE`
 - `enforce_gtid_consistency – ON`
 - b. 如果已變更資料庫執行個體的參數群組，請重新啟動資料庫執行個體。如果已變更僅供讀取複本的參數群組，請重新啟動僅供讀取複本。
4. 等候全部的匿名交易完成複寫。若要檢查這些是否已複寫，請執行下列動作：
 - a. 對於來源資料庫執行個體執行下列陳述式。

```
SHOW MASTER STATUS;
```

請注意 File 和 Position 欄中的值。

- b. 對於每個僅供讀取複本，在上一個步驟中使用來源執行個體的檔案和位置資訊來執行下列查詢。

```
SELECT MASTER_POS_WAIT('file', position);
```

例如，如果檔案名稱是 `mysql-bin-changelog.000031`，而且位置是 107，則執行下列陳述式。

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

如果僅供讀取複本已超過指定的位置，會立即傳回查詢。否則，該函式會等候一段時間。所有僅供讀取複本的查詢傳回時，繼續進行下一個步驟。

5. 僅重設 GTID 式複寫的 GTID 參數。
 - a. 確定與資料庫執行個體有關的參數群組和每個僅供讀取複本有下列參數設定：

- `gtid_mode` – ON
 - `enforce_gtid_consistency` – ON
- b. 重新啟動資料庫執行個體和每個僅供讀取複本。
6. 在每個僅供讀取複本上，執行下列程序。

```
CALL mysql.rds_set_master_auto_position(1);
```

為具備僅供讀取複本的 MySQL 資料庫執行個體停用 GTID 式複寫

您可以為 具備僅供讀取複本的 MySQL 資料庫執行個體停用 GTID 式複寫。

為 具備僅供讀取複本的 MySQL 資料庫執行個體停用 GTID 式複寫

1. 在每個僅供讀取複本上，執行下列程序：

```
CALL mysql.rds_set_master_auto_position(0);
```

2. 將 `gtid_mode` 重設為 `ON_PERMISSIVE`。
 - a. 確定與 MySQL 資料庫執行個體和每個僅供讀取複本關聯的參數群組將 `gtid_mode` 設定為 `ON_PERMISSIVE`。

如需使用參數群組設定組態參數的詳細資訊，請參閱[使用參數群組](#)。
 - b. 重新啟動 MySQL 資料庫執行個體和每個僅供讀取複本。如需重新啟動的詳細資訊，請參閱[重新啟動中的資料庫執行個體](#)。
3. 將 `gtid_mode` 重設為 `OFF_PERMISSIVE`。
 - a. 確定與 MySQL 資料庫執行個體和每個僅供讀取複本關聯的參數群組將 `gtid_mode` 設定為 `OFF_PERMISSIVE`。
 - b. 重新啟動 MySQL 資料庫執行個體和每個僅供讀取複本。
4. 等待系統將全部的 GTID 交易套用至所有僅供讀取複本。要檢查這些是否應用，請執行以下步驟：
 - a. 在 MySQL 資料庫執行個體上，執行 `SHOW MASTER STATUS` 命令。

您的輸出應類似於以下輸出。

```
File                               Position
-----
mysql-bin-changelog.000031        107
-----
```

請注意輸出中的檔案和位置。

- b. 在每個僅供讀取複本上，使用其來源執行個體的檔案和位置資訊，在上一個步驟中執行下列查詢：

適用於 MySQL 8.0.26 及更高版本

```
SELECT SOURCE_POS_WAIT('file', position);
```

對於 MySQL 5.7 版本

```
SELECT MASTER_POS_WAIT('file', position);
```

例如，如果檔案名稱為mysql-bin-changelog.000031且位置為107，請執行下列陳述式：

適用於 MySQL 8.0.26 及更高版本

```
SELECT SOURCE_POS_WAIT('mysql-bin-changelog.000031', 107);
```

對於 MySQL 5.7 版本

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

5. 重設 GTID 參數以停用以 GTID 為基礎的複寫。
 - a. 確定與 MySQL 資料庫執行個體有關的參數群組和每個僅供讀取複本有下列參數設定：
 - `gtid_mode` – OFF
 - `enforce_gtid_consistency` – OFF
 - b. 重新啟動 MySQL 資料庫執行個體和每個僅供讀取複本。

使用外部來源執行個體設定二進位日誌檔案位置複寫

您可使用二進位日誌檔案複寫，在 RDS for MySQL 或 MariaDB 資料庫執行個體，以及位於 Amazon RDS 外部的 MySQL 或 MariaDB 執行個體之間，設定複寫。

主題

- [開始之前](#)
- [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)

開始之前

您可以使用複寫交易的二進制日誌檔案位置設定複寫。

在 Amazon RDS 資料庫執行個體上啟動複寫功能所需的許可有其限制，不供 Amazon RDS 主要使用者使用。因此，請確定使用 Amazon RDS [mysql.rds_set_external_master](#) 和 [mysql.rds_start_replication](#) 命令，設定即時資料庫與 Amazon RDS 資料庫之間的複寫作業。

若要為 MySQL 或 MariaDB 資料庫設定二進位日誌記錄格式，請更新 `binlog_format` 參數。如果資料庫執行個體使用預設的資料庫執行個體參數群組，請建立新的資料庫參數群組來修改 `binlog_format` 設定。建議您在 `binlog_format` 中使用預設設定，亦即 MIXED。不過，如果需要特定的二進位日誌 (binlog) 格式，也可以將 `binlog_format` 設定為 ROW 或 STATEMENT。重新啟動資料庫執行個體，讓變更生效。

如需有關設定 `binlog_format` 參數的資訊，請參閱[設定適用於 MySQL RDS](#)。如需各種 MySQL 複寫類型隱含意義的相關資訊，請參閱 MySQL 文件中的[基於陳述式和基於列的複寫的優缺點](#)。

Note

從適用於 MySQL 的 RDS 版本 8.0.36 開始，Amazon RDS 不會複寫資料庫。mysql 因此，如果 Amazon RDS 複本上需要外部資料庫上有使用者，請務必手動建立這些使用者。

使用外部來源執行個體設定二進位日誌檔案位置複寫

在 Amazon RDS 上設定外部來源執行個體和複本時，按照這些準則操作：

- 監控 Amazon RDS 資料庫執行個體 (您的複本) 的容錯移轉事件。若發生容錯移轉，資料庫執行個體 (您的複本) 可能會以不同的網路地址，在新主機上重新建立。如需如何監控容錯移轉事件的資訊，請參閱[使用 Amazon RDS 事件通知](#)。

- 直到您已驗證二進位日誌已套用至複本前，都要將二進位日誌保存在來源執行個體上。如此一來，發生故障時，這種維護可確保您能夠還原來源執行個體。
- 開啟 Amazon RDS 資料庫執行個體的自動備份功能。如果您需要重新同步來源執行個體與複本，開啟自動備份功能可確保您能夠將複本還原至特定時間點。如需備份與 point-in-time 還原的資訊，請參閱[備份、還原和匯出資料](#)。

使用外部來源執行個體設定二進位日誌檔案複寫

1. 將來源 MySQL 或 MariaDB 執行個體設成唯讀狀態。

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. 在來源 MySQL 或 MariaDB 執行個體上執行 SHOW MASTER STATUS 命令，以確定二進位日誌的位置。

您會獲得類似下列範例的輸出。

```
File                                Position
-----
mysql-bin-changelog.000031         107
-----
```


3. 使用 mysqldump，從外部執行個體將資料庫複製到 Amazon RDS 資料庫執行個體。如果是非常大型的資料庫，您可能需要使用 [將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#) 中的程序。

對於LinuxmacOS、或Unix：

```
mysqldump --databases database_name \  
  --single-transaction \  
  --compress \  
  --order-by-primary \  
  -u local_user \  
  -plocal_password | mysql \  
  --host=hostname \  
  --port=3306 \  
  -u RDS_user_name \  
  -pRDS_password
```

在 Windows 中：

```
mysqldump --databases database_name ^
--single-transaction ^
--compress ^
--order-by-primary ^
-u local_user ^
-plocal_password | mysql ^
--host=hostname ^
--port=3306 ^
-u RDS_user_name ^
-pRDS_password
```

 Note

請注意 -p 選項與輸入的密碼之間不能有空格。

若要指定連線到 Amazon RDS 資料庫執行個體的主機名稱、使用者名稱、連接埠和密碼，請使用 --host 命令中的 --user (-u)、--port、-p 和 mysql 選項。主機名稱是 Amazon RDS 資料庫執行個體端點的網域名稱服務 (DNS) 名稱，例如 myinstance.123456789012.us-east-1.rds.amazonaws.com。您可在 AWS Management Console 的執行個體詳細資訊中找到端點值。

4. 將來源 MySQL 或 MariaDB 執行個體重新設為可寫入狀態。

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

如需如何製作備份以搭配複寫作業使用的詳細資訊，請參閱 [MySQL 文件](#)。

5. 在中 AWS Management Console，將託管外部資料庫的伺服器 IP 位址新增到 Amazon RDS 資料庫執行個體的虛擬私有雲端 (VPC) 安全群組。如需有關修改 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 安全群組](#)。

IP 地址會在滿足下列條件時變更：

- 您使用公有 IP 地址供外部來源執行個體和資料庫執行個體通訊。
- 外部來源執行個體停止後又重新啟動。

如果符合這些條件，請先驗證 IP 地址再新增它。

您可能還需要設定本機網路，以允許從 Amazon RDS 資料庫執行個體的 IP 地址建立連線。如此，您的本機網路就能與外部 MySQL 或 MariaDB 執行個體進行通訊。若要尋找 Amazon RDS 資料庫執行個體的 IP 地址，請使用 `host` 命令。

```
host db_instance_endpoint
```

主機名稱是 Amazon RDS 資料庫執行個體端點的 DNS 名稱。

6. 使用您選擇的用戶端連線至外部執行個體，接著建立執行複寫作業所需的使用者。只對於複寫作業使用此帳戶，並限制其存取您的網域，以提升安全性。以下是範例。

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

7. 若為外部執行個體，請將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。舉例來說，若要將所有資料庫的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的「`repl_user`」使用者，請發出下列命令。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. 將 Amazon RDS 資料庫執行個體設為複本。若要執行此操作，請先以主要使用者身分連線至 Amazon RDS 資料庫執行個體。使用 [mysql.rds_set_external_master](#) 命令找到外部 MySQL 或 MariaDB 資料庫，將其視為來源執行個體。使用您在步驟 2 中所確定的主控端日誌檔案名稱與主控端日誌位置。以下是範例。

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,  
'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

Note

在 RDS for MySQL 上，您可以改為執行 [mysql.rds_set_external_master_with_delay](#) 預存程序，進而選擇使用延遲複寫。在 RDS for MySQL 上，使用延遲複寫的原因之一，是為

了透過 [mysql.rds_start_replication_until](#) 預存程序來開啟災難復原功能。目前，RDS for MariaDB 支援延遲複寫，但不支援 `mysql.rds_start_replication_until` 程序。

9. 在 Amazon RDS 資料庫執行個體上發出 [mysql.rds_start_replication](#) 命令，啟動複寫作業。

```
CALL mysql.rds_start_replication;
```

multi-source-replication 為適用於 MySQL 的 RDS 進行設定

使用多來源複寫，您可以將 Amazon RDS for MySQL 資料庫執行個體設定為複本，以接收來自多個 RDS for MySQL 來源資料庫執行個體的二進位記錄事件。執行下列引擎版本的 RDS for MySQL 資料庫執行個體支援多來源複寫：

- 8.0.35 及更高版本的次要版本
- 5.7.44 及更高版本的次要版本

如需 MySQL 多重來源複寫的相關資訊，請參閱 [MySQL 文件中的 MySQL 多重來源複寫](#)。MySQL 文件包含有關此功能的詳細資訊，而本主題說明如何在 RDS for MySQL 資料庫執行個體上設定和管理多來源複寫通道。

主題

- [多重來源複製的使用案例](#)
- [多重來源複寫的考量與最佳作法](#)
- [多重來源複製的先決條件](#)
- [在適用於 MySQL 資料庫執行個體的 RDS 上設定多重來源複寫](#)
- [透過多重來源複製使用篩選器](#)
- [監視多重來源複製通道](#)
- [適用於 MySQL 的 RDS 上的多重來源複寫的限制](#)

多重來源複製的使用案例

下列情況適用於在適用於 MySQL 的 RDS 上使用多來源複寫：

- 需要將不同資料庫執行個體上的多個碎片合併或合併為單一碎片的應用程式。

- 需要從多個來源合併的資料產生報告的應用程式。
- 針對在多個 RDS for MySQL 資料庫執行個體之間分散的資料建立合併長期備份的需求。

多重來源複寫的考量與最佳作法

在 RDS for MySQL 上使用多來源複寫之前，請先檢閱下列考量事項和最佳做法：

- 確定設定為多來源複本的資料庫執行個體具有足夠的資源，例如輸送量、記憶體、CPU 和 IOPS，以處理來自多個來源執行個體的工作負載。
- 定期監控多重來源複本的資源使用率，並調整儲存體或執行個體組態以處理工作負載，而不會耗費資源。
- 您可以 `replica_parallel_workers` 將系統變數設定為大於的值，在多來源複本上設定多執行緒複製。0 在這種情況下，分配給每個通道的線程數是此變量的值，加上一個協調器線程來管理應用程式線程。
- 適當地設定複寫篩選器以避免衝突。若要將整個資料庫複製到複本上的另一個資料庫，您可以使用 `--replicate-rewrite-db` 此選項。例如，您可以將資料庫 A 中的所有表格複製到複本執行個體上的資料庫 B。當所有來源執行個體都使用相同的結構描述命名慣例時，這個方法會很有用。如需選 `--replicate-rewrite-db` 項的相關資訊，請參閱 MySQL 文件中的 [複本伺服器選項和變數](#)。
- 若要避免複寫錯誤，請避免寫入複本。我們建議您在多重來源複本上啟用 `read_only` 參數，以封鎖寫入作業。這樣做有助於消除寫入作業衝突所造成的複寫問題。
- 若要提高在多來源複本上執行的讀取作業 (例如排序和高負載聯結) 的效能，請考慮使用 RDS 最佳化讀取。此功能可協助處理依賴大型暫存資料表或排序檔案的查詢。如需詳細資訊，請參閱 [the section called “使用 RDS Optimized Reads 改善查詢效能”](#)。
- 若要將複寫延遲降到最低並改善多重來源複本的效能，請考慮啟用最佳化寫入。如需詳細資訊，請參閱 [the section called “使用 RDS Optimized Writes 改善 MySQL 的寫入效能”](#)。
- 一次在一個通道上執行管理作業 (例如變更組態)，並避免從多個連線對多個通道執行變更。這些作法可能會導致複寫作業發生衝突。例如，從多個連接同時執行 `rds_skip_repl_error_for_channel` 和 `rds_start_replication_for_channel` 程序可能會導致跳過與預期不同的通道上的事件。
- 您可以在多來源複寫執行個體上啟用備份，並將資料從該執行個體匯出到 Amazon S3 儲存貯體以進行長期存放。不過，也請務必在個別來源執行個體上設定適當保留的備份。如需將快照資料匯出到 Amazon S3 的相關資訊，請參閱 [the section called “將資料庫快照資料匯出至 Amazon S3”](#)。
- 若要在多來源複本上分配僅供讀取工作負載，您可以從多來源複本建立僅供讀取複本。您可以根據應用程式的需求，找出不同 AWS 區域的僅供讀取複本。如需僅供讀取複本的詳細資訊，請參閱 [the section called “使用 MySQL 僅供讀取複本”](#)。

多重來源複製的先決條件

設定多重來源複製之前，請先完成下列先決條件。

- 確定每個來源 RDS for MySQL 資料庫執行個體都已啟用自動備份。啟用自動備份可啟用二進位記錄。若要瞭解如何啟用自動備份，請參閱[the section called “啟用自動備份”](#)。
- 為避免複寫錯誤，建議您封鎖來源資料庫執行個體的寫入作業。您可以在連接至 RDS for MySQL 來源資料庫執行個體的自訂參數群組ON中，將參數設定為。read-only您可以使用 AWS Management Console 或建立新的 AWS CLI 自訂參數群組，或修改既有參數群組。如需詳細資訊，請參閱 [the section called “建立資料庫參數群組”](#) 及 [the section called “修改資料庫參數群組中的參數”](#)。
- 針對每個來源資料庫執行個體，將執行個體的 IP 位址新增至多來源資料庫執行個體的 Amazon 虛擬私有雲 (VPC) 安全群組。若要識別來源資料庫執行個體的 IP 位址，您可以執行命令dig **RDS Endpoint**。在與目標多來源資料庫執行個體相同的 VPC 中，從 Amazon EC2 執行個體執行命令。
- 針對每個來源資料庫執行個體，使用用戶端連線至資料庫執行個體，並建立具有複寫所需權限的資料庫使用者，如下列範例所示。

```
CREATE USER 'repl_user' IDENTIFIED BY 'password';  
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user';
```

在適用於 MySQL 資料庫執行個體的 RDS 上設定多重來源複寫

設定多來源複製通道與設定單一來源複製類似。對於多重來源複製，您必須先在來源執行個體上開啟二進位記錄。然後，您將資料從來源匯入到多來源複本。然後，您可以使用二進位記錄座標或使用 GTID 自動定位，從每個來源開始複寫。

若要將 RDS for MySQL 資料庫執行個體設定為兩個或多個適用於 MySQL 資料庫執行個體的多來源複本，請執行下列步驟。

主題

- [步驟 1：將資料從來源資料庫執行個體匯入多來源複本](#)
- [步驟 2：開始從來源資料庫執行個體複寫到多來源複本](#)

步驟 1：將資料從來源資料庫執行個體匯入多來源複本

在每個來源資料庫執行個體上執行下列步驟。

在將資料從來源匯入多來源複本之前，請執行SHOW MASTER STATUS指令來判斷目前的二進位記錄檔和位置。請記下這些詳細資料，以便在下一個步驟中使用。在此範例輸出中，檔案為mysql-bin-changelog.000031，位置為107。

```
File                Position
-----
mysql-bin-changelog.000031    107
-----
```

現在，使用，將資料庫從來源資料庫執行個體複製到多來源複本mysqldump，如下列範例所示。

```
mysqldump --databases database_name \
  --single-transaction \
  --compress \
  --order-by-primary \
  -u RDS_user_name \
  -p RDS_password \
  --host=RDS Endpoint | mysql \
  --host=RDS Endpoint \
  --port=3306 \
  -u RDS_user_name \
  -p RDS_password
```

複製資料庫之後，您可以在來源資料庫執行個體OFF上將唯讀參數設定為。

步驟 2：開始從來源資料庫執行個體複寫到多來源複本

針對每個來源資料庫執行個體，使用主要使用者身分證明連線至執行個體，然後執行下列兩個預存程序。這些預存程序會在通道上設定複寫並開始複寫。此範例使用上一個步驟中範例輸出的 binlog 檔案名稱和位置。

```
CALL mysql.rds_set_external_source_for_channel('mysourcehost.example.com', 3306,
  'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0, 'channel_1');
CALL mysql.rds_start_replication_for_channel('channel_1');
```

如需有關使用這些預存程序及其他預存程序來設定和管理複寫通道的詳細資訊，請參閱[the section called “管理多重來源複製”](#)。

透過多重來源複製使用篩選器

您可以使用複寫篩選來指定要在多來源複本中複製哪些資料庫和表格。複寫篩選條件可以包含複寫中的資料庫和資料表，或將其排除在複寫之外。如需複製篩選器的詳細資訊，請參閱[the section called “使用 MySQL 設定複寫篩選條件”](#)。

透過多重來源複製，您可以全域或在通道層級設定複寫篩選器。通道層級篩選僅適用於執行 8.0 版的受支援資料庫執行個體。下列範例說明如何在全域或頻道層級設定篩選器。

請注意下列多重來源複寫中篩選的需求和行為：

- 需要通道名稱周圍的反引號 (`)。
- 如果您變更參數群組中的複寫篩選器，則會重新啟動所有具有更新之通道的多來源複本，以動態套用變更。sql_thread如果更新涉及全域篩選器，則會重新啟動處於執行中狀態的所有複寫通道。
- 所有全域篩選器都會在任何通道特定篩選器之前套用。
- 如果在通道層級全域套用篩選器，則只會套用通道層級的篩選器。例如，如果濾鏡是`replicate_ignore_db="db1, `channel_22`:db2"`，則`replicate_ignore_db`設定為會套用至db1除了 for 以外的所有色版channel_22，並且只channel_22會忽略來自的變更db2。

範例 1：設定全域篩選

在下列範例中，temp_data資料庫會從每個通道的複寫中排除。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
--db-parameter-group-name myparametergroup \  
--parameters "ParameterName=replicate-ignore-  
db,ParameterValue='temp_data',ApplyMethod=immediate"
```

範例 2：設定通道層級篩選

在下列範例中，來自sample22資料庫的變更只包含在通道中channel_22。同樣地，來自sample99資料庫的變更只包含在通道中channel_99。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
--db-parameter-group-name myparametergroup \  
--channel-filtering channel_22=sample22,channel_99=sample99
```



```
--parameters "ParameterName=replicate-do-db,ParameterValue='\`channel_22\`:sample22,\`channel_99\`:sample99',ApplyMethod=immediate"
```

監視多重來源複製通道

您可以使用下列方法監視多重來源複本中的個別通道：

- 若要監視所有通道或特定通道的狀態，請連線到多來源複本並執行 `SHOW REPLICA STATUS` 或 `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'` 命令。如需詳細資訊，請參閱 MySQL 文件中的 [檢查複寫狀態](#)。
- 若要在複寫通道啟動、停止或移除時接收通知，請使用 RDS 事件通知。如需詳細資訊，請參閱 [the section called “使用 Amazon RDS 事件通知”](#)。
- 若要監視特定通道的延遲，請檢查該通道的 `ReplicationChannelLag` 指標。此量度的資料點期間為 60 秒 (1 分鐘)，可使用 15 天。若要找出通道的複製通道延遲，請使用執行個體識別碼和複製通道名稱。若要在此延遲超過特定臨界值時接收通知，您可以設定 CloudWatch 警示。如需詳細資訊，請參閱 [the section called “用 CloudWatch 監控 RDS”](#)。

適用於 MySQL 的 RDS 上的多重來源複寫的限制

下列限制適用於適用於 MySQL 的 RDS 上的多來源複寫：

- 目前，適用於 MySQL 的 RDS 支援為多來源複本設定最多 15 個通道。
- 僅供讀取複本執行個體無法設定為多來源複本。
- 若要在執行引擎 5.7 版的 RDS for MySQL 上設定多來源複寫，必須在複本執行個體上啟用效能結構描述。在執行 8.0 版引擎的 RDS 上，啟用效能結構描述是選擇性的。
- 對於執行 5.7 版引擎的 RDS for MySQL，複寫篩選器會套用至所有複寫通道。對於執行 8.0 版引擎的 RDS for MySQL，您可以設定套用至所有複寫通道或個別通道的篩選器。
- 還原 RDS 快照或執行 Point-in-time-還原 (PITR) 不會還原多來源複本通道組態。
- 當您建立多來源複本的僅供讀取複本時，它只會複寫來自多來源執行個體的資料。它不會恢復任何通道配置。
- MySQL 不支持為每個通道設置不同數量的 `parallel` 工作程序。根據該 `replica_parallel_workers` 值，每個通道都會獲得相同數量的 `parallel` 工作程序。

如果您的多重來源複寫目標是異地同步備份資料庫叢集，則適用下列額外限制：

- 在對該執行個體進行任何寫入之前，必須先為來源 RDS for MySQL 執行個體設定通道。

- 每個適用於 MySQL 的來源 RDS 執行個體都必須啟用以 GTID 為基礎的複寫。
- 資料庫叢集上的容錯移轉事件會移除多來源複寫組態。還原該組態需要重複設定步驟。

設定適用於 MySQL 的 RDS 的主動-主動式叢集

您可以使用 MySQL 群組複寫外掛程式，為 RDS for MySQL 設定主動-主動式叢集。執行 8.0.35 版及更高版本次要版本的 MySQL 資料庫執行個體適用的 RDS 版本支援群組複寫外掛程式。

如需 MySQL 群組複寫的相關資訊，請參閱 MySQL 說明文件中的[群組複寫](#)。MySQL 文件包含有關此功能的詳細資訊，而本主題說明如何在 RDS for MySQL 資料庫執行個體上設定和管理外掛程式。

Note

為了簡潔起見，本主題中所有提及的「主動-主動」叢集都是指使用 MySQL 群組複寫外掛程式的主動-主動式叢集。

主題

- [主動-主動叢集的使用案例](#)
- [主動-主動式叢集的考量事項和最佳做法](#)
- [跨 VPC 主動-主動式叢集的先決條件](#)
- [主動-主動叢集的必要參數設定](#)
- [將現有的資料庫執行個體轉換為主動-主動叢集](#)
- [使用新的資料庫執行個體設定主動-主動叢集](#)
- [將資料庫執行個體新增至主動-主動叢集](#)
- [監視主動-主動叢集](#)
- [停止主動-主動叢集中資料庫執行個體上的群組複寫](#)
- [重新命名主動-主動叢集中的資料庫執行個體](#)
- [從主動-主動叢集移除資料庫執行個體](#)
- [適用於 MySQL 的 RDS 主動-主動式叢集的限制](#)

主動-主動叢集的使用案例

下列案例是使用主動-主動式叢集的最佳選擇：

- 需要叢集中所有資料庫執行個體以支援寫入作業的應用程式。群組複製外掛程式可讓主動-主動叢集中每個資料庫執行個體上的資料保持一致。如需其運作方式的詳細資訊，請參閱 MySQL 文件中的[群組複寫](#)。

- 需要資料庫持續可用性的應用程式。使用主動-主動叢集時，資料會保留在叢集中的所有資料庫執行個體上。如果一個資料庫執行個體失敗，應用程式可以將流量重新路由到叢集中的另一個資料庫執行個體。
- 可能需要在叢集中的不同資料庫執行個體之間分割讀取和寫入作業以進行負載平衡的應用程式。使用主動-主動叢集，您的應用程式可以傳送讀取流量至特定資料庫執行個體，並將流量寫入其他執行個體。您也可以隨時切換要傳送讀取或寫入的資料庫執行個體。

主動-主動式叢集的考量事項和最佳做法

在您使用 RDS for MySQL 主動-主動式叢集之前，請先檢閱下列考量事項和最佳做法：

- 主動-主動叢集不能有九個以上的資料庫執行個體。
- 使用群組複寫外掛程式，您可以控制主動-主動叢集的交易一致性保證。如需詳細資訊，請參閱 MySQL 文件中的[交易一致性保證](#)。
- 當不同的資料庫執行個體更新主動-主動叢集中的相同資料列時，可能會發生衝突。如需有關衝突和解決衝突的資訊，請參閱 MySQL 文件中的[群組複寫](#)。
- 對於容錯，請在主動-主動叢集中包含至少三個資料庫執行個體。您可以僅使用一或兩個資料庫執行個體來設定主動-主動叢集，但該叢集不具備容錯能力。如需容錯的相關資訊，請參閱 MySQL 文件中的[容錯](#)。
- 當資料庫執行個體加入現有的主動-主動叢集並執行與叢集中最低引擎版本相同的引擎版本時，資料庫執行個體會以讀寫模式加入。
- 當資料庫執行個體加入現有的主動-主動叢集並執行的引擎版本高於叢集中的最低引擎版本時，資料庫執行個體必須保持為唯讀模式。
- 如果您在資料庫參數群組中將資料庫執行個體的 `rds.group_replication_enabled` 參數設定為來啟用資料庫執行個體的群組複製，但複寫尚未啟動或無法啟動，則資料庫執行個體會進入 `super-read-only` 模式以避免資料不一致。如需有關 `super-read-only` 模式的資訊，請參閱 [MySQL 文件](#)。
- 您可以升級主動-主動叢集中的資料庫執行個體，但資料庫執行個體是唯讀的，直到主動-主動叢集中的所有其他資料庫執行個體升級至相同的引擎版本或更高版本的引擎。升級資料庫執行個體時，資料庫執行個體會自動加入相同的主動-主動叢集。若要避免資料庫執行個體意外切換為唯讀模式，請停用該執行個體的自動次要版本升級。如需升級 MySQL 資料庫執行個體的相關資訊，請參閱[升級 MySQL 資料庫引擎](#)。
- 您可以將異地同步備份資料庫執行個體部署中的資料庫執行個體新增至現有的主動-主動叢集。您也可以將主動-主動叢集中的單一可用區資料庫執行個體轉換為異地同步備份資料庫執行個體部署。如果異地同步備份部署中的主要資料庫執行個體失敗，則該主要執行個體容錯移轉至待命執行個體。新

的主資料庫執行個體會在容錯移轉完成後自動加入相同的叢集。如需異地同步備份資料庫執行個體部署的詳情，請參閱[多可用區域資料庫執行個體部署](#)。

- 建議主動-主動叢集中的資料庫執行個體對其維護時段有不同的時間範圍。此做法可避免叢集中的多個資料庫執行個體同時離線進行維護。如需詳細資訊，請參閱[Amazon RDS 維護時段](#)。
- 主動-主動叢集可以在資料庫執行個體之間的連線使用 SSL。[若要設定 SSL 連線，請設定群組複製恢復使用 SSL 和群組複製模式參數](#)。這些參數的值必須與主動-主動叢集中的所有資料庫執行個體相符。

目前，主動-主動叢集不支援之間的連線的憑證授權單位 (CA) 驗證。AWS 區域因此，必須將[群組複製 _ssl_mode](#) 參數設定為 DISABLED (預設值) 或針對跨區域叢集。REQUIRED

- RDS for MySQL 主動-主動式叢集會以多主要模式執行。[群組複製 _ 強制 _ 更新 _ 無處不在檢查](#)的預設值和參數是靜態的。ON將此參數設定為時ON，應用程式無法插入具有階層式外部索引鍵條件約束的資料表中。
- 適用於 MySQL 的 RDS 主動-主動式叢集會使用 MySQL 通訊堆疊來確保連線安全性，而不是 XCOM。如需詳細資訊，請參閱 MySQL 文件中[的連線安全性管理的通訊堆疊](#)。
- 當資料庫參數群組與主動-主動叢集中的資料庫執行個體相關聯時，我們建議您只將此資料庫參數群組與叢集中的其他資料庫執行個體建立關聯。
- 主動-主動式叢集僅支援適用於 MySQL 資料庫執行個體的 RDS。這些資料庫執行個體必須執行受支援的資料庫引擎版本。
- 當主動-主動叢集中的資料庫執行個體發生意外故障時，RDS 會自動啟動資料庫執行個體的復原。如果資料庫執行個體無法復原，建議您使用叢集中運作良好的資料庫執行個體進行 point-in-time 復原，以新的資料庫執行個體取代它。如需說明，請參閱[使用復原將資料庫執行個體新增至主動-主動叢集 point-in-time](#)。
- 您可以刪除主動-主動叢集中的資料庫執行個體，而不會影響叢集中的其他資料庫執行個體。如需刪除資料庫執行個體的相關資訊，請參閱[刪除資料庫執行個體](#)。

跨 VPC 主動-主動式叢集的先決條件

您可以在多個 VPC 中使用資料庫執行個體來設定主動-主動叢集。VPC 可以位於相同AWS 區域或不 AWS 區域同。

Note

在多個之間傳送流量AWS 區域可能會產生額外費用。如需詳細資訊，請參閱[通用架構的資料傳輸成本概觀](#)。

如果您要在單一 VPC 中設定主動-主動叢集，則可以略過這些步驟並繼續到。[使用新的資料庫執行個體設定主動-主動叢集](#)

為資料庫執行個體位於多個 VPC 中的主動-主動叢集做好準備

1. 請確定 CIDR 區塊中的 IPv4 位址範圍符合下列需求：

- 虛擬私人雲端的 CIDR 區塊中的 IPv4 位址範圍無法重疊。
- *CIDR ##### IPv4 #####128.0.0.0/subnet_mask### 128.0.0.0/ #####*

下列範圍說明了這些需求：

- 10.1.0.0/16在一個 VPC 和另一個 VPC 10.2.0.0/16 中受到支援。
- 172.1.0.0/16在一個 VPC 和另一個 VPC 172.2.0.0/16 中受到支援。
- 10.1.0.0/16在一個 VPC 和另一個 VPC 10.1.0.0/16 中不受支援，因為範圍重疊。
- 10.1.0.0/16在一個 VPC 和另一個 VPC 172.1.0.0/16 中不受支持，因為一個在下面128.0.0.0/*subnet_mask*，另一個在上面。128.0.0.0/*subnet_mask*

如需 CIDR 區塊的相關資訊，請參閱 Amazon VPC 使用者指南中的 [V PC CIDR 區塊](#)。

2. 在每個 VPC 中，請確定 DNS 解析和 DNS 主機名稱均已啟用。

如需指示，請參閱 Amazon VPC 使用者指南中的檢視和更新虛擬私人雲端的 [DNS 屬性](#)。

3. 設定 VPC，以便您可以使用下列其中一種方式來路由 VPC 之間的流量：

- 在 VPC 之間建立 VPC 對等連線。

如需指示，請參閱 Amazon [VPC 對等互連指南中的建立](#) VPC 對等連線。在每個 VPC 中，請確定安全群組有輸入規則，這些規則會參考對等 VPC 中的安全群組。執行此作業，可允許流量傳入和傳出與互連 VPC 中參考之安全群組相關聯的執行個體。如需指示，請參閱 Amazon VPC [對等互連指南中的更新安全群組以參考對等安全群組](#)。

- 在 VPC 之間建立傳輸閘道。

如需指示，請參閱 Amazon [VPC 傳輸閘道中的傳輸閘道入門](#)。在每個 VPC 中，請確定安全群組有允許來自其他 VPC 的流量的輸入規則，例如指定其他 VPC CIDR 的輸入規則。這樣做可讓流量往返於與主動-主動叢集中參照安全性群組相關聯的執行個體。如需詳細資訊，請參閱 Amazon VPC 使用者指南 [中的使用安全群組控制AWS資源流量](#)。

主動-主動叢集的必要參數設定

當您設定 RDS for MySQL 主動-主動式叢集時，需要下列參數設定。

參數	描述	必要的設定
<code>binlog_format</code>	設定二進位記錄格式。適用於 MySQL 的 RDS 的預設值為 MIXED。如需詳細資訊，請參閱 MySQL 文件 。	ROW
<code>enforce_gtid_consistency</code>	強制執行陳述式的 GTID 一致性。適用於 MySQL 的 RDS 的預設值為 OFF。如需詳細資訊，請參閱 MySQL 文件 。	ON
<code>group_replication_group_name</code>	將群組複寫名稱設定為 UUID。UUID 格式為。11111111-2222-3333-4444-555555555555 您可以透過連線至 MySQL 資料庫執行個體並執行來產生 MySQL UUID。SELECT UUID() 主動-主動叢集中所有資料庫執行個體的值必須相同。如需詳細資訊，請參閱 MySQL 文件 。	一個 MySQL 的 UUID
<code>gtid-mode</code>	控制以 GTID 為基礎的記錄。適用於 MySQL 的 RDS 的預設值為 OFF_PERMISSIVE。如需詳細資訊，請參閱 MySQL 文件 。	ON
<code>rds.custom_dns_resolution</code>	指定是否允許從 VPC 擬私人雲端中的 Amazon DNS 伺服器進行 DNS 解析。使用 <code>rds.group_replicat</code>	1

參數	描述	必要的設定
	<p><code>ion_enabled</code> 參數啟用群組複寫時，必須啟用 DNS 解析。使用 <code>rds.group_replication_enabled</code> 參數停用群組複寫時，無法啟用 DNS 解析。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 Amazon DNS 伺服器。</p>	
<code>rds.group_replication_enabled</code>	指定是否啟用資料庫執行個體的群組複寫。必須在主動-主動叢集中的資料庫執行個體上啟用群組複寫。	1
<code>slave_preserve_commit_order</code>	控制交易在複本上確認的順序。適用於 MySQL 的 RDS 的預設值為 ON。如需詳細資訊，請參閱 MySQL 文件 。	ON

將現有的資料庫執行個體轉換為主動-主動叢集

您要移轉至主動-主動叢集的資料庫執行個體的資料庫引擎版本必須是 MySQL 8.0.35 或更新版本。如果您需要升級引擎版本，請參閱 [升級 MySQL 資料庫引擎](#)。

如果要在多個 VPC 中設定具有資料庫執行個體的主動-主動叢集，請確定已完成中的先決條件。 [跨 VPC 主動-主動式叢集的先決條件](#)

完成下列步驟，將現有的資料庫執行個體移轉到適用於 RDS for MySQL 的主動-主動叢集。

主題

- [步驟 1：在一或多個自訂參數群組中設定主動-主動叢集參數](#)
- [步驟 2：將資料庫執行個體與已設定所需群組複寫參數的資料庫參數群組產生關聯](#)
- [步驟 3：建立主動-主動叢集](#)
- [步驟 4：為主動-主動式叢集建立其他 RDS for MySQL 資料庫執行個體](#)


```

--parameters
"ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

"ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

"ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
    "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

"ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

"ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
\

"ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"

```

在Windows中：

```

aws rds modify-db-parameter-group ^
--db-parameter-group-name myactivepg ^
--parameters
"ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

"ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

"ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
    "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

"ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

"ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
^

"ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"

```

步驟 2：將資料庫執行個體與已設定所需群組複寫參數的資料庫參數群組產生關聯

將資料庫執行個體與您在上一個步驟中建立或修改的參數群組產生關聯。如需說明，請參閱[將資料庫參數群組與資料庫執行個體建立關聯](#)。

重新啟動資料庫執行個體以使新參數設定生效。如需說明，請參閱[重新啟動中的資料庫執行個體](#)。

步驟 3：建立主動-主動叢集

在與資料庫執行個體關聯的資料庫參數群組中，將`group_replication_group_seeds`參數設定為要轉換之資料庫執行個體的端點。

您可以使用AWS Management Console或AWS CLI來設定參數。設定此參數後，您不需要重新啟動資料庫執行個體。如需設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

下列範例會執行[modify-db-parameter-group](#) AWS CLI命令來設定參數：

對於Linux macOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name myactivepg \  
  --parameters  
  "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

在Windows中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myactivepg ^  
  --parameters  
  "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

步驟 4：為主動-主動式叢集建立其他 RDS for MySQL 資料庫執行個體

若要為主動-主動叢集建立其他資料庫執行個體，請在要轉換的資料庫執行個體上執行 point-in-time 復原。如需說明，請參閱[使用復原將資料庫執行個體新增至主動-主動叢集 point-in-time](#)。

主動-主動叢集最多可以有九個資料庫執行個體。在資料庫執行個體上執行 point-in-time 復原，直到您擁有要用於叢集的資料庫執行個體數量為止。執行時 point-in-recovery，請務必將要新增的資料庫執行

個體與 `rds.group_replication_enabled` 設定為的資料庫參數群組建立關聯¹。否則，群組複寫將不會在新增的資料庫執行個體上啟動。

步驟 5：在要轉換的資料庫執行個體上初始化群組

初始化群組並開始複寫：

1. Connect 到要在 SQL 用戶端中轉換的資料庫執行個體。如需連線至適用於 MySQL 的 RDS 資料庫執行個體的詳細資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。
2. 在 SQL 用戶端中，執行下列預存程序，並以使用者的密碼取代 `Group_replication_user_password`。 `rdsgrepladmin` 使用者會保留給主動-主動叢集中的群組複寫連線使用。在主動-主動叢集中的所有資料庫執行個體上，此使用者的密碼必須相同。

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(1);
```

此範例將 `binlog retention hours` 值設定為 168，表示二進位記錄檔會在資料庫執行個體上保留七天。您可以調整此值以符合您的需求。

此範例 1 在 `mysql.rds_group_replication_start` 預存程序中指定使用目前資料庫執行個體初始化新群組。

如需範例中呼叫之預存程序的詳細資訊，請參閱[管理主動-主動叢集](#)。

步驟 6：在主動-主動叢集中的其他資料庫執行個體上開始複寫

對於主動-主動叢集中的每個資料庫執行個體，請使用 SQL 用戶端連線至執行個體，然後執行下列預存程序。以使用者 `#####`。 `rdsgrepladmin`

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

此範例將binlog retention hours值設定為168，表示每個資料庫執行個體會保留二進位記錄檔七天。您可以調整此值以符合您的需求。

此範例在mysql.rds_group_replication_start預存程序中指定將目前資料庫執行個體加入現有群組。

Tip

請務必在主動-主動叢集中的所有其他資料庫執行個體上執行這些預存程序。

步驟 7：(建議) 檢查作用中-主動叢集的狀態

若要確定叢集的每個成員都已正確設定，請連線至主動-主動叢集中的資料庫執行個體，然後執行下列 SQL 命令，以檢查叢集的狀態：

```
SELECT * FROM performance_schema.replication_group_members;
```

您MEMBER_STATE的輸出應ONLINE為每個資料庫執行個體顯示，如下列範例輸出所示：

```
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST      |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 | | |
| 3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
| 3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
| 3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
3 rows in set (0.00 sec)
```

如需可能MEMBER_STATE值的相關資訊，請參閱 MySQL 文件中的[群組複寫伺服器狀態](#)。

使用新的資料庫執行個體設定主動-主動叢集

完成下列步驟，以使用新的 RDS for MySQL 資料庫執行個體設定主動-主動叢集。

如果要在多個 VPC 中設定具有資料庫執行個體的主動-主動叢集，請確定已完成中的先決條件。[跨 VPC 主動-主動式叢集的先決條件](#)

主題

- [步驟 1：在一或多個自訂參數群組中設定主動-主動叢集參數](#)
- [步驟 2：為主動-主動式叢集建立新的 RDS for MySQL 資料庫執行個體](#)
- [步驟 4：指定主動-主動叢集中的資料庫執行個體](#)
- [步驟 5：初始化資料庫執行個體上的群組並開始複寫](#)
- [步驟 6：在主動-主動叢集中的其他資料庫執行個體上開始複寫](#)
- [步驟 7：\(建議\) 檢查作用中-主動叢集的狀態](#)
- [步驟 8：\(選擇性\) 將資料匯入主動-主動叢集中的資料庫執行個體](#)

步驟 1：在一或多個自訂參數群組中設定主動-主動叢集參數

主動-主動叢集中的 RDS for MySQL 資料庫執行個體必須與具有所需參數設定正確的自訂參數群組相關聯。如需有關參數和每個參數所需設定的資訊，請參閱[主動-主動叢集的必要參數設定](#)。

您可以在新參數群組或現有參數群組中設定這些參數。但是，為了避免意外影響不屬於主動-主動叢集的資料庫執行個體，我們強烈建議您建立新的自訂參數群組。主動-主動叢集中的資料庫執行個體可以與相同的資料庫參數群組或不同的資料庫參數群組產生關聯。

您可以使用AWS Management Console或建AWS CLI立新的自訂參數群組。如需詳細資訊，請參閱[建立資料庫參數群組](#)。下列範例會執行命[create-db-parameter-group](#)AWS CLI令，以建立名為的自訂資料庫參數群組`myactivepg`：

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name myactivepg \  
  --db-parameter-group-family mysql8.0 \  
  --description "Parameter group for active-active clusters"
```

在Windows中：

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

您也可以使用AWS Management Console或AWS CLI來設定自訂參數群組中的參數。如需詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

下列範例會執行[modify-db-parameter-group](#) AWS CLI命令來設定參數：

對於Linux macOS、或Unix：

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
  "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-reboot" \

  "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-reboot" \

  "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-reboot" \
  "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-reboot" \

  "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

  "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate" \

  "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555'
  reboot"
```

在Windows中：

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
  "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-reboot" ^
```

```

"ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-reboot" ^

"ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-reboot" ^
    "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-reboot" ^

"ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

"ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate" ^

"ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555-reboot"

```

步驟 2：為主動-主動式叢集建立新的 RDS for MySQL 資料庫執行個體

適用於 MySQL 資料庫執行個體的 RDS 版本 8.0.35 及更高版本支援主動-主動式叢集。您最多可以為叢集建立九個新的資料庫執行個體。

您可以使用 AWS Management Console 或來建立新 AWS CLI 的資料庫執行個體。如需建立資料庫執行個體的詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。建立資料庫執行個體時，請將其與您在上一個步驟中建立或修改的資料庫參數群組產生關聯。

步驟 4：指定主動-主動叢集中的資料庫執行個體

在與每個資料庫執行個體關聯的資料庫參數群組中，將 `group_replication_group_seeds` 參數設定為要包含在叢集中的資料庫執行個體的端點。

您可以使用 AWS Management Console 或 AWS CLI 來設定參數。設定此參數後，您不需要重新啟動資料庫執行個體。如需設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

下列範例會執行 [modify-db-parameter-group](#) AWS CLI 命令來設定參數：

對於 Linux macOS、或 Unix：

```

aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
  "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-

```



```
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-  
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

在Windows中：

```
aws rds modify-db-parameter-group ^  
  --db-parameter-group-name myactivepg ^  
  --parameters  
  "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-  
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-  
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-  
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Tip

請務必在與主動-主動叢集中的資料庫執行個體相關聯的每個資料庫參數群組中設定參數。group_replication_group_seeds

步驟 5：初始化資料庫執行個體上的群組並開始複寫

您可以選擇任何新的資料庫來初始化群組並開始複寫。若要這樣做，請完成下列步驟：

1. 在主動-主動叢集中選擇資料庫執行個體，然後在 SQL 用戶端中連線到該資料庫執行個體。如需連線至適用於 MySQL 的 RDS 資料庫執行個體的詳細資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。
2. 在 SQL 用戶端中，執行下列預存程序，並以使用者的密碼取代 *Group_replication_user_password*。rdsgreprepladminrdsgreprepladmin 使用者會保留給主動-主動叢集中的群組複寫連線使用。在主動-主動叢集中的所有資料庫執行個體上，此使用者的密碼必須相同。

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog  
call mysql.rds_group_replication_create_user('group_replication_user_password');  
call  
  mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');  
call mysql.rds_group_replication_start(1);
```

此範例將binlog retention hours值設定為168，表示二進位記錄檔會在資料庫執行個體上保留七天。您可以調整此值以符合您的需求。

此範例1在mysql.rds_group_replication_start預存程序中指定使用目前資料庫執行個體初始化新群組。

如需範例中呼叫之預存程序的詳細資訊，請參閱[管理主動-主動叢集](#)。

步驟 6：在主動-主動叢集中的其他資料庫執行個體上開始複寫

對於主動-主動叢集中的每個資料庫執行個體，請使用 SQL 用戶端連線至執行個體，然後執行下列預存程序。以使用者#####。rdsgrprepladmin

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

此範例將binlog retention hours值設定為168，表示每個資料庫執行個體會保留二進位記錄檔七天。您可以調整此值以符合您的需求。

此範例0在mysql.rds_group_replication_start預存程序中指定將目前資料庫執行個體加入現有群組。

Tip

請務必在主動-主動叢集中的所有其他資料庫執行個體上執行這些預存程序。

步驟 7：(建議) 檢查作用中-主動叢集的狀態

若要確定叢集的每個成員都已正確設定，請連線至主動-主動叢集中的資料庫執行個體，然後執行下列 SQL 命令，以檢查叢集的狀態：

```
SELECT * FROM performance_schema.replication_group_members;
```

您MEMBER_STATE的輸出應ONLINE為每個資料庫執行個體顯示，如下列範例輸出所示：

```
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
```

```

| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
3306 | ONLINE      | PRIMARY    | 8.0.35        | MySQL          |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
3 rows in set (0.00 sec)

```

如需可能MEMBER_STATE值的相關資訊，請參閱 MySQL 文件中的[群組複寫伺服器狀態](#)。

步驟 8：(選擇性) 將資料匯入主動-主動叢集中的資料庫執行個體

您可以將資料從 MySQL 資料庫匯入主動-主動叢集中的資料庫執行個體。匯入資料後，群組複寫會將其複寫到叢集中的其他資料庫執行個體。

如需匯入資料的資訊，請參閱[將資料匯入至 Amazon RDS MariaDB 或 MySQL 資料庫，並減少停機時間](#)。

將資料庫執行個體新增至主動-主動叢集

您可以透過還原資料庫快照或將資料庫執行個體還原到某個時間點，將資料庫執行個體新增至主動-主動叢集。主動-主動叢集最多可包含九個資料庫執行個體。

當您將資料庫執行個體復原到某個時間點時，它通常會包含比從資料庫快照還原的資料庫執行個體更新的交易。當資料庫執行個體有較近的交易時，開始複寫時需要套用的交易較少。因此，使用 point-in-time 復原將資料庫執行個體新增至叢集通常比從資料庫快照還原更快。

主題

- [使用復原將資料庫執行個體新增至主動-主動叢集 point-in-time](#)
- [使用資料庫快照將資料庫執行個體新增至主動-主動叢集](#)

使用復原將資料庫執行個體新增至主動-主動叢集 point-in-time

您可以透過對叢集中的資料庫執行個體執行 point-in-time 復原，將資料庫執行個體新增至主動-主動叢集。

如需將資料庫執行個體復原至其他時間點的相關資訊AWS 區域，請參閱[將自動備份複製到另一個 AWS 區域](#)。

使用復原將資料庫執行個體新增至主動-主動叢集 point-in-time

1. 在主動-主動叢集中的資料庫執行個體上執行 point-in-time 復原，以建立新的資料庫執行個體。

您可以在叢集中的任何資料庫執行個體執行 point-in-time 復原，以建立新的資料庫執行個體。如需說明，請參閱[將資料庫執行個體還原至指定的時間](#)。

Important

在期間 point-in-time-recovery，請將新資料庫執行個體與已設定主動-主動叢集參數的資料庫參數群組產生關聯。否則，群組複寫將不會在新的資料庫執行個體上啟動。如需有關參數和每個參數所需設定的資訊，請參閱[主動-主動叢集的必要參數設定](#)。

Tip

如果您在開始 point-in-time 復原之前拍攝資料庫執行個體的快照，您可能可以減少在新資料庫執行個體上套用交易所需的時間。

2. 將資料庫執行個體新增至與主動-主動叢集中資料庫執行個體相關聯的每個資料庫參數群組中的參數，包括您與新資料庫執行個體相關聯的資料庫參數群組。group_replication_group_seeds

如需設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

3. 在 SQL 用戶端中，連接到新的資料庫執行個體，然後呼叫[神秘的 rds_ 群組複製_ 設定恢復通道預存程序](#)。以使用者#####。rdsgprepladmin

```
call
mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
```

4. 使用 SQL 用戶端呼叫[我們的群組複製啟動](#)預存程序以開始複寫：

```
call mysql.rds_group_replication_start(0);
```

使用資料庫快照將資料庫執行個體新增至主動-主動叢集

您可以在叢集中建立資料庫執行個體的資料庫快照，然後還原資料庫快照，將資料庫執行個體新增至主動-主動叢集。

如需將快照複製到其他快照的詳細資訊AWS 區域，請參閱[the section called “跨區域複製”](#)。

使用資料庫快照將資料庫執行個體新增至主動-主動叢集

1. 在主動-主動叢集中建立資料庫執行個體的資料庫快照。

您可以建立叢集中任何資料庫執行個體的資料庫快照。如需說明，請參閱[為單一可用區資料庫執行個體建立資料庫快照](#)。

2. 從資料庫快照還原資料庫執行個體。

在快照還原作業期間，請將新資料庫執行個體與已設定主動-主動叢集參數的資料庫參數群組建立關聯。如需有關參數和每個參數所需設定的資訊，請參閱[主動-主動叢集的必要參數設定](#)。

如需從資料庫快照還原資料庫執行個體的詳細資訊，請參閱[從資料庫快照還原](#)。

3. 將資料庫執行個體新增至與主動-主動叢集中資料庫執行個體相關聯的每個資料庫參數群組中的參數，包括您與新資料庫執行個體相關聯的資料庫參數群組。group_replication_group_seeds

如需設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

4. 在 SQL 用戶端中，連接到新的資料庫執行個體，然後呼叫[神秘的 rds_ 群組複製 _ 設定恢復通道預存程序](#)。以使用者#####。rdsgrepladmin

```
call
mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
```

5. 使用 SQL 用戶端呼叫[我們的群組複製啟動](#)預存程序以開始複製：

```
call mysql.rds_group_replication_start(0);
```

監視主動-主動叢集

您可以透過連線至叢集中的資料庫執行個體，並執行下列 SQL 命令來監視主動-主動叢集：

```
SELECT * FROM performance_schema.replication_group_members;
```

您MEMBER_STATE的輸出應ONLINE為每個資料庫執行個體顯示，如下列範例輸出所示：

```
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST      |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 | | |
| 3306 | ONLINE      | PRIMARY    | 8.0.35         | MySQL          |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
| 3306 | ONLINE      | PRIMARY    | 8.0.35         | MySQL          |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
| 3306 | ONLINE      | PRIMARY    | 8.0.35         | MySQL          |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+
3 rows in set (0.00 sec)
```

如需可能MEMBER_STATE值的相關資訊，請參閱 MySQL 文件中的[群組複寫伺服器狀態](#)。

停止主動-主動叢集中資料庫執行個體上的群組複寫

您可以在主動-主動叢集中的資料庫執行個體上停止群組複寫。停止群組複寫時，資料庫執行個體會進入 super-read-only 模式，直到重新啟動複製或從主動-主動叢集移除該資料庫執行個體為止。如需有關 super-read-only 模式的資訊，請參閱 [MySQL 文件](#)。

暫時停止主動-主動叢集的群組複製

1. 使用 SQL 用戶端 Connect 至主動-主動叢集中的資料庫執行個體。

如需連線至適用於 MySQL 的 RDS 資料庫執行個體的詳細資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。

2. 在 SQL 用戶端中，呼叫預[神秘的 rds_ 群組複製 _ 停止](#)存程序：

```
call mysql.rds_group_replication_stop();
```

重新命名主動-主動叢集中的資料庫執行個體

您可以變更主動-主動叢集中資料庫執行個體的名稱。若要重新命名主動-主動叢集中的多個資料庫執行個體，請一次執行一個資料庫執行個體。因此，在重新命名下一個資料庫執行個體之前，請重新命名一個資料庫執行個體，並將其重新連接

重新命名主動-主動叢集中的資料庫執行個體

1. Connect 至 SQL 用戶端中的資料庫執行個體，然後呼叫[神秘的 rds_ 群組複製 _ 停止](#)預存程序：

```
call mysql.rds_group_replication_stop();
```

2. 依照中的指示重新命名資料庫執行個體[重新命名資料庫執行個體](#)。
3. 修改與主動-主動叢集中資料庫執行個體相關聯的每個資料庫參數群組中的參數。group_replication_group_seeds

在參數設定中，將舊的資料庫執行個體端點取代為新的資料庫執行個體端點。如需設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

4. Connect 至 SQL 用戶端中的資料庫執行個體，然後呼叫[我們的群組複製啟動](#)預存程序：

```
call mysql.rds_group_replication_start(0);
```

從主動-主動叢集移除資料庫執行個體

從主動-主動叢集中移除資料庫執行個體時，該執行個體會還原為獨立資料庫執行個體。

從主動-主動叢集移除資料庫執行個體

1. Connect 至 SQL 用戶端中的資料庫執行個體，然後呼叫[神秘的 rds_ 群組複製 _ 停止](#)預存程序：

```
call mysql.rds_group_replication_stop();
```

2. 修改將保留在主動-主動叢集中的資料庫執行個體的group_replication_group_seeds參數。

在 `group_replication_group_seeds` 參數中，刪除要從主動-主動叢集移除的資料庫執行個體。如需設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

3. 修改要從主動-主動叢集移除的資料庫執行個體參數，使其不再是叢集的一部分。

您可以將資料庫執行個體與不同的參數群組產生關聯，也可以修改與資料庫執行個體關聯的資料庫參數群組中的參數。要修改的參數包括 `group_replication_group_namerds.group_replication_enabled`、`group_replication_group_seeds`。如需有關主動-主動叢集參數的詳細資訊，請參閱 [主動-主動叢集的必要參數設定](#)

如果您修改資料庫參數群組中的參數，請確定資料庫參數群組未與主動-主動叢集中的其他資料庫執行個體相關聯。

4. 重新啟動您從主動-主動叢集移除的資料庫執行個體，以使新參數設定生效。

如需說明，請參閱 [重新啟動中的資料庫執行個體](#)。

適用於 MySQL 的 RDS 主動-主動式叢集的限制

下列限制適用於適用於 MySQL 版 RDS 的主動-主動式叢集：

- 主要使用者名稱不能用 `rdsgrepladmin` 於主動-主動叢集中的資料庫執行個體。此使用者名稱會保留給群組複寫連線使用。
- 對於主動-主動叢集中具有僅供讀取複本的資料庫執行個體，長時間複寫狀態 `Replicating` 可能會導致記錄檔超出儲存限制。如需僅供讀取複本狀態的相關資訊，請參閱 [監控僅供讀取複本](#)。
- 主動-主動叢集中的資料庫執行個體不支援藍/綠部署。如需詳細資訊，請參閱 [使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。
- 主動-主動叢集中的資料庫執行個體不支援 Kerberos 驗證。如需詳細資訊，請參閱 [針對 MySQL 使用 Kerberos 身分驗證](#)。
- 異地同步備份資料庫叢集中的資料庫執行個體無法新增至主動-主動叢集。

不過，異地同步備份資料庫執行個體部署中的資料庫執行個體可新增至主動-主動叢集。

如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。

- 沒有主索引鍵的資料表不會在主動-主動叢集中複寫，因為群組複寫外掛程式拒絕寫入。
- 非 InnoDB 表不會在主動-主動叢集中複寫。
- 主動-主動叢集不支援叢集中不同資料庫執行個體上的並行 DML 和 DDL 陳述式。

- 您無法將主動-主動叢集設定為針對群組的複寫模式使用單一主要模式。對於此配置，我們建議改用異地同步備份資料庫叢集。如需詳細資訊，請參閱[多可用區域資料庫叢集部署](#)。
- 主動-主動叢集中的資料庫執行個體不支援多來源複寫。
- 跨區域主動-主動叢集無法針對群組複寫連線強制執行憑證授權單位 (CA) 驗證。

使用複寫從 MySQL 資料庫執行個體匯出資料

如要從 RDS for MySQL 資料庫執行個體，將資料匯出至在 Amazon RDS 外部執行的 MySQL 執行個體，您可使用複寫。在此案例中，MySQL 資料庫執行個體是來源 MySQL 資料庫執行個體，而在 Amazon RDS 外部執行的 MySQL 執行個體則是外部 MySQL 資料庫。

外部 MySQL 資料庫可以在資料中心以內部部署方式執行，或在 Amazon EC2 執行個體上執行。外部 MySQL 資料庫必須執行與來源 MySQL 資料庫執行個體相同的版本，或更新版本。

只有在從來源 MySQL 資料庫執行個體匯出資料庫期間，才支援複寫到外部 MySQL 資料庫。在匯出資料後且應用程式可以開始存取外部 MySQL 執行個體時，就應該終止複寫。

下列清單顯示要採取的步驟。後續幾節會詳細討論各步驟。

1. 準備外部 MySQL 資料庫執行個體。
2. 準備用於複寫的來源 MySQL 資料庫執行個體。
3. 使用 `mysqldump` 公用程式將資料庫從來源 MySQL 資料庫執行個體轉移到外部 MySQL 資料庫。
4. 開始複寫到外部 MySQL 資料庫。
5. 匯出完成之後，請停止複寫。

準備外部 MySQL 資料庫

請執行下列步驟來準備外部 MySQL 資料庫。

準備外部 MySQL 資料庫

1. 安裝外部 MySQL 資料庫。
2. 以主要使用者身分連線至外部 MySQL 資料庫。然後建立支援存取資料庫的管理員、應用程式和服務所需的使用者。
3. 遵循 MySQL 文件中的指示，準備外部 MySQL 資料庫作為複本。如需詳細資訊，請參閱 [MySQL 文件](#)。
4. 設定輸出規則，讓外部 MySQL 資料庫在匯出期間充當僅供讀取複本。輸出規則可讓外部 MySQL 資料庫在複寫期間連線到來源 MySQL 資料庫執行個體。指定輸出規則，以允許經由傳輸控制通訊協定 (TCP) 連接至來源 MySQL 資料庫執行個體的連接埠和 IP 位址。

為您的環境指定適當的輸出規則：

- 如果外部 MySQL 資料庫正在以 Amazon VPC 服務為基礎之 Virtual Private Cloud (VPC) 的 Amazon EC2 執行個體中執行，請在 VPC 安全群組中指定輸出規則。如需詳細資訊，請參閱[使用安全群組控制存取](#)。
 - 如果外部 MySQL 資料庫以內部部署方式安裝，請在防火牆中指定輸出規則。
5. 如果外部 MySQL 資料庫在 VPC 中執行，除了安全群組輸出規則之外，另請為 VPC 存取控制清單 (ACL) 規則設定規則：
- 設定 ACL 輸入規則，以允許 TCP 流量從來源 MySQL 資料庫執行個體的 IP 位址流向連接埠 1024–65535。
 - 設定 ACL 輸出規則，以允許傳出 TCP 流量流向來源 MySQL 資料庫執行個體的連接埠和 IP 位址。
- 如需 Amazon VPC 網路 ACL 的詳細資訊，請參閱《Amazon VPC 使用者指南》中的[網路 ACL](#)。
6. (選用) 將 `max_allowed_packet` 參數設定為最大值，以避免複寫錯誤。我們建議您使用此設定。

準備來源 MySQL 資料庫執行個體

執行下列步驟，準備來源 MySQL 資料庫執行個體做為複寫來源。

準備來源 MySQL 資料庫執行個體

1. 確定用戶端電腦有足夠磁碟空間在設定複寫時儲存二進位日誌。
2. 連線至來源 MySQL 資料庫執行個體，並遵循 MySQL 文件中[建立複寫的使用者](#)的指示來建立複寫帳戶。
3. 在執行來源 MySQL 資料庫執行個體的系統上設定輸入規則，以允許外部 MySQL 資料庫在複寫期間連線。指定輸入規則，以允許從外部 MySQL 資料庫的 IP 地址，經由 TCP 連接至來源 MySQL 資料庫執行個體所使用的連接埠。
4. 指定輸出規則：
 - 如果來源 MySQL 資料庫執行個體在 VPC 中執行，請在 VPC 安全群組中指定輸入規則。如需詳細資訊，請參閱[使用安全群組控制存取](#)。
5. 如果來源 MySQL 資料庫執行個體在 VPC 中執行，則除了安全群組輸入規則，還需要設定 VPC ACL 規則。

- 設定 ACL 輸入規則，以允許從外部 MySQL 資料庫的 IP 位址，經由 TCP 連接至 Amazon RDS 執行個體所使用的連接埠。
- 設定 ACL 輸出規則，以允許從連接埠 1024–65535 經由 TCP 連接至外部 MySQL 資料庫的 IP 位址。

如需 Amazon VPC 網路 ACL 的詳細資訊，請參閱《Amazon VPC 使用者指南》中的[網路 ACL](#)。

6. 確定備份保留期的設定夠長，以避免在匯出期間清除二進位日誌。如果在匯出完成之前清除任何日誌，則您必須重新開始複寫。如需設定備份保留期的詳細資訊，請參閱[備份簡介](#)。
7. 使用 `mysql.rds_set_configuration` 預存程序來設定夠長的二進位日誌保留期，以避免在匯出期間清除二進位日誌。如需詳細資訊，請參閱[存取 MySQL 二進位日誌](#)。
8. 請從來源 MySQL 資料庫執行個體建立 Amazon RDS 僅供讀取複本，以進一步確保不會清除來源 MySQL 資料庫執行個體的二進位日誌。如需詳細資訊，請參閱[建立僅供讀取複本](#)。
9. 建立 Amazon RDS 僅供讀取複本之後，請呼叫 `mysql.rds_stop_replication` 預存程序來停止複寫程序。來源 MySQL 資料庫執行個體不再清除其二進位日誌檔案，因此可用於複寫程序。
10. (選用) 將 `max_allowed_packet` 參數和 `slave_max_allowed_packet` 參數同時設定為最大值，以避免複寫錯誤。這兩個參數的最大值為 1 GB。我們建議兩個參數都使用此設定。如需有關設定參數的詳細資訊，請參閱[修改資料庫參數群組中的參數](#)。

複製資料庫

執行下列步驟來複製資料庫。

複製資料庫

1. 連線到來源 MySQL 資料庫執行個體的 RDS 僅供讀取複本，然後執行 MySQL `SHOW REPLICATION STATUS\G` 陳述式。請注意下列項目的值：
 - `Master_Host`
 - `Master_Port`
 - `Master_Log_File`
 - `Exec_Master_Log_Pos`

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

2. 使用 `mysqldump` 公用程式建立快照，以將資料從 Amazon RDS 複製到本機用戶端電腦。確定用戶端電腦有足夠空間，可保留待複寫資料庫的 `mysqldump` 檔案。如果資料庫非常龐大，此程序可能需要數小時的時間。按照 MySQL 文件中[使用 `mysqldump` 建立資料快照](#)的指示進行。

下列範例會在用戶端上執行 `mysqldump`，並將傾印寫入檔案。

對於LinuxmacOS、或Unix：

```
mysqldump -h source_MySQL_DB_instance_endpoint \  
  -u user \  
  -ppassword \  
  --port=3306 \  
  --single-transaction \  
  --routines \  
  --triggers \  
  --databases database database2 > path/rds-dump.sql
```

在Windows中：

```
mysqldump -h source_MySQL_DB_instance_endpoint ^  
  -u user ^  
  -ppassword ^  
  --port=3306 ^  
  --single-transaction ^  
  --routines ^  
  --triggers ^  
  --databases database database2 > path\rds-dump.sql
```

您可以將備份檔案載入至外部 MySQL 資料庫中。如需詳細資訊，請參閱 MySQL 文件中的[重新載入 SQL 格式備份](#)。您可以執行另一個公用程式，將資料載入至外部 MySQL 資料庫中。

完成匯出

執行下列步驟以完成匯出。

完成匯出

1. 使用 MySQL `CHANGE MASTER` 陳述式來設定外部 MySQL 資料庫。指定獲授予 `REPLICATION SLAVE` 許可之使用者的 ID 和密碼。指定您從 RDS 僅供讀取複本上執行的 `MySQL SHOW REPLICA STATUS\G` 陳述式所取得的 `Master_Host`、`Master_Port`、`Relay_Master_Log_File` 和 `Exec_Master_Log_Pos` 值。如需詳細資訊，請參閱 [MySQL 文件](#)。

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

2. 使用 MySQL `START REPLICA` 命令啟動從來源 MySQL 資料庫執行個體複寫到外部 MySQL 資料庫。

這樣做會從來源 MySQL 資料庫執行個體開始複寫，並匯出自停止從僅供 Amazon RDS 讀取複本複寫之後發生的所有來源變更。

Note

MySQL 以前的版本使用 `START SLAVE` 而不是 `START REPLICA`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `START SLAVE`。

3. 在外部 MySQL 資料庫上執行 `MySQL SHOW REPLICA STATUS\G` 命令，以確認它是以僅供讀取複本的方式運作。如需解譯結果的詳細資訊，請參閱 [MySQL 文件](#)。
4. 當外部 MySQL 資料庫上的複寫已與來源 MySQL 資料庫執行個體同步之後，請使用 `MySQL STOP REPLICA` 命令停止來源 MySQL 資料庫執行個體複寫。

Note

MySQL 以前的版本使用 `STOP SLAVE` 而不是 `STOP REPLICA`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `STOP SLAVE`。

5. 在 Amazon RDS 僅供讀取複本上，呼叫 `mysql.rds_start_replication` 預存程序。這麼做可讓 Amazon RDS 開始從來源 MySQL 資料庫執行個體中清除二進位日誌檔案。

MySQL 資料庫執行個體的選項

以下說明執行 MySQL 資料庫引擎的 Amazon RDS 執行個體適用的選項或其他功能。若要啟用這些選項，您可以將它們新增至自訂選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需使用選項群組的相關詳細資訊，請參閱[使用選項群組](#)。

Amazon RDS 支援 MySQL 的下列選項：

選項	選項 ID	引擎版本
MariaDB 稽核外掛程式支援 MySQL	MARIADB_AUDIT_PLUGIN	MySQL 8.0.28 和更新的 8.0 版 所有 MySQL 5.7 版本
MySQL memcached 支援	MEMCACHED	所有 MySQL 5.7 和 8.0 版本

MariaDB 稽核外掛程式支援 MySQL

Amazon RDS 針對 MySQL 資料庫執行個體提供以開放原始碼 MariaDB 稽核外掛程式為基礎的稽核外掛程式。如需詳細資訊，請參閱 [MySQL Server GitHub 儲存庫的稽核外掛程式](#)。

Note

MySQL 的稽核外掛程式以 MariaDB 稽核外掛程式為基礎。在本文中，我們將其稱為 MariaDB 稽核外掛程式。

MariaDB 稽核外掛程式會記錄資料庫活動，包含登入資料庫的使用者，以及針對資料庫執行的查詢。資料庫活動的記錄會儲存在日誌檔中。

Note

目前，MariaDB 稽核外掛程式僅支援下列 RDS for MySQL 版本：


- MySQL 8.0.28 和更新的 8.0 版
- 所有 MySQL 5.7 版本

稽核外掛程式選項設定

Amazon RDS 支援 MariaDB 稽核外掛程式的下列設定。

選項設定	有效值	預設值	描述
SERVER_AUDIT_FILE_PATH	/rdsdbdata/log/audit/	/rdsdbdata/log/audit/	日誌檔的位置。日誌檔包含 SERVER_AUDIT_EVENTS 中指定之活動的記錄。如需更多詳細資訊，請參閱 檢視並列出資料庫日誌檔案 及 MySQL 資料庫日誌檔案 。
SERVER_AUDIT_ROTATE_SIZE	1–100000000	1000000	達到時導致檔案輪換的大小 (以位元組表示)。如需更多詳細資訊，請參閱 RDS for MySQL 資料庫日誌概觀 。

選項設定	有效值	預設值	描述
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	要在 <code>server_audit_output_type=file</code> 時儲存的日誌輪換次數。如果設定為 0，則日誌檔永遠不會輪換。如需詳細資訊，請參閱 RDS for MySQL 資料庫日誌概觀 及 下載資料庫日誌檔案 。
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL, , QUERY_DML, , QUERY_DML_NO_SELECT, , QUERY_DCL	CONNECT, QUERY	<p>要記錄在日誌中之活動的類型。安裝 MariaDB 稽核外掛程式是自行記錄的。</p> <ul style="list-style-type: none"> CONNECT：記錄與資料庫的成功和不成功連線，以及與資料庫的中斷連線。 QUERY：記錄針對資料庫執行之所有查詢的文字。 QUERY_DDL：類似於 QUERY 事件，但只傳回資料定義語言 (DDL) 查詢 (CREATE、ALTER 等)。 QUERY_DML：類似於 QUERY 事件，但只傳回資料處理語言 (DML) 查詢 (INSERT、UPDATE 和 SELECT 等)。 QUERY_DML_NO_SELECT：類似於 QUERY_DML 事件，但不會記錄 SELECT 查詢。 <p>此 QUERY_DML_NO_SELECT 設定僅支援 RDS for MySQL 5.7.34 和更新的 5.7 版本以及 8.0.25 版和更新的 8.0 版本。</p> <ul style="list-style-type: none"> QUERY_DCL：類似於 QUERY 事件，但只傳回資料控制語言 (DCL) 查詢 (GRANT、REVOKE 等)。 <p>對於 MySQL，不支援 TABLE。</p>

選項設定	有效值	預設值	描述
SERVER_AUDIT_INCL_USERS	多個以逗號分隔的值	無	只包括來自所指定使用者的活動。預設情況下，會記錄所有使用者的活動。SERVER_AUDIT_INCL_USERS 和 SERVER_AUDIT_EXCL_USERS 是相互排斥的。如果您將值新增至 SERVER_AUDIT_INCL_USERS ，請確保未將任何值新增至 SERVER_AUDIT_EXCL_USERS 。
SERVER_AUDIT_EXCL_USERS	多個以逗號分隔的值	無	<p>排除來自所指定使用者的活動。預設情況下，會記錄所有使用者的活動。SERVER_AUDIT_INCL_USERS 和 SERVER_AUDIT_EXCL_USERS 是相互排斥的。如果您將值新增至 SERVER_AUDIT_EXCL_USERS ，請確保未將任何值新增至 SERVER_AUDIT_INCL_USERS 。</p> <p>rdsadmin 使用者會每次查詢資料庫，以檢查資料庫的運作狀態。根據您的其他設定，此活動可能會導致您的日誌檔大小非常快速地變得非常大。如果您不需要記錄此活動，請將 rdsadmin 使用者新增至 SERVER_AUDIT_EXCL_USERS 清單。</p> <div data-bbox="829 1283 1507 1549" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>即使針對此選項設定指定了使用者，也會一律記錄所有使用者的 CONNECT 活動。</p> </div>
SERVER_AUDIT_LOGGING	ON	ON	記錄為作用中狀態。唯一有效的值為 ON。Amazon RDS 不支援停用記錄。如果想要停用記錄，請移除 MariaDB 稽核外掛程式。如需更多詳細資訊，請參閱 移除 MariaDB 稽核外掛程式 。

選項設定	有效值	預設值	描述
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	記錄中查詢字串長度的限制。

新增 MariaDB 稽核外掛程式

將 MariaDB 稽核外掛程式新增至資料庫執行個體的一般程序如下：

- 建立新的選項群組，或是複製或修改現有選項群組
- 將選項新增至選項群組
- 將選項群組與資料庫執行個體關聯

在您新增 MariaDB 稽核外掛程式之後，不需要重新啟動資料庫執行個體。只要選項群組作用中，稽核就會立即開始。

Important

新增 MariaDB 稽核外掛程式至資料庫執行個體可能會導致停機。我們建議在維護視窗下或資料庫低負載的時候新增 MariaDB 稽核外掛程式。

新增 MariaDB 稽核外掛程式

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則，建立自訂的資料庫選項群組。為 Engine (引擎) 選擇 mysql，並為 Major engine version (主要引擎版本) 選擇 5.7 或 8.0。如需更多詳細資訊，請參閱 [建立選項群組](#)。
2. 將 MARIADB_AUDIT_PLUGIN 選項新增至選項群組，並設定選項設定。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。如需每項設定的詳細資訊，請參閱 [稽核外掛程式選項設定](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體。
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

- 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

稽核日誌格式

日誌檔案以 UTF-8 格式的逗號分隔變數 (CSV) 檔案表示。

Tip

日誌檔案項目不會循序排列。若要排序項目，請使用時間戳記值。若要查看最新事件，您可能必須檢閱所有日誌檔案。為了更靈活排序和搜尋日誌資料，請開啟設定，將稽核日誌上傳到 CloudWatch，然後使用 CloudWatch 界面檢視這些日誌。

若要查看包含更多欄位類型和以 JSON 格式輸出的稽核資料，您也可以使用資料庫活動串流功能。如需更多詳細資訊，請參閱 [使用資料庫活動串流來監控 Amazon RDS](#)。

稽核日誌檔案的每一列依指定順序包含以下逗號分隔的資訊：

欄位	描述
timestamp	所記錄事件的 YYYYMMDD 後有 HH:MI:SS (24 小時制)。
serverhost	為其記錄事件之執行個體的名稱。
username	使用者的連線使用者名稱。
host	使用者連線來源的主機。
connectionid	所記錄操作的連線 ID 號碼。
queryid	查詢 ID 號碼，可用來尋找關聯式資料表事件和相關的查詢。若為 TABLE 事件，則會新增多行。
operation	記錄的動作類型。可能值為：CONNECT、QUERY、READ、WRITE、CREATE、ALTER、RENAME 及 DROP。
database	由 USE 命令設定的作用中資料庫。

欄位	描述
物件	若為 QUERY 事件，此值指出資料庫所執行的查詢。若為 TABLE 事件，則指出資料表名稱。
retcode	所記錄操作的傳回碼。
connection_type	與伺服器之連線的安全狀態。可能值為： <ul style="list-style-type: none"> • 0 – 未定義 • 1 – TCP/IP • 2 – 槽 • 3 – 具名管道 • 4 – SSL/TLS • 5 – 共用記憶體 <p>只有 RDS for MySQL 5.7.34 版和更新的 5.7 版及所有 8.0 版中包含此欄位。</p>

檢視和下載 MariaDB 稽核外掛程式日誌

在啟用 MariaDB 稽核外掛程式之後，您可以採取與存取任何其他文字型日誌檔相同的方式來存取日誌檔中的結果。稽核日誌檔位於 `/rdsdbdata/log/audit/`。如需在主控台中檢視日誌檔的相關資訊，請參閱[檢視並列出資料庫日誌檔案](#)。如需下載日誌檔的相關資訊，請參閱[下載資料庫日誌檔案](#)。

修改 MariaDB 稽核外掛程式設定

在啟用 MariaDB 稽核外掛程式之後，您可以修改設定。如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。如需每項設定的詳細資訊，請參閱[稽核外掛程式選項設定](#)。

移除 MariaDB 稽核外掛程式

Amazon RDS 不支援在 MariaDB 稽核外掛程式中關閉記錄。不過，您可以從資料庫執行個體中移除外掛程式。當您移除 MariaDB 稽核外掛程式時，資料庫執行個體會自動重新啟動來停止稽核。

若要從資料庫執行個體中移除 MariaDB 稽核外掛程式，請執行下列其中一個動作：

- 從其所屬的選項群組中移除 MariaDB 稽核外掛程式。此變更會影響使用選項群組的所有資料庫執行個體。如需詳細資訊，請參閱[從選項群組移除選項](#)

- 修改資料庫執行個體，並指定不包括外掛程式的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

MySQL memcached 支援

Amazon RDS 支援透過 memcached 界面，藉此使用 MySQL 5.6 所推出的 InnoDB 資料表。藉由 memcached API，應用程式即可以類似 NoSQL 索引鍵/值資料儲存的方式，使用 InnoDB 資料表。

memcached 界面是簡單的索引鍵型快取。應用程式使用 memcached 來插入、操控及擷取快取的索引鍵-值資料對。MySQL 5.6 引進一個實作協助程式服務的外掛程式，該服務透過 memcached 通訊協定公開 InnoDB 資料表中的資料。如需有關 MySQL memcached 外掛程式的詳細資訊，請參閱 [InnoDB 與 Memcached 的整合](#)。

如要啟用 RDS for MySQL 資料庫執行個體的 memcached 支援

1. 決定要用於控制 memcached 界面存取的安全群組。如果已經在使用此 SQL 界面的應用程式和將會存取 memcached 界面的應用程式是同一批，您可以使用此 SQL 界面使用的現有 VPC 安全群組。如果是不同的應用程式將存取 memcached 界面，請定義新的 VPC 或資料庫安全群組。如需管理安全群組的詳細資訊，請參閱 [使用安全群組控制存取](#)
2. 請建立自訂資料庫選項群組，選取 MySQL 做為引擎類型和版本。如需建立選項群組的相關資訊，請參閱 [建立選項群組](#)。
3. 將 MEMCACHED 選項新增至選項群組。指定 memcached 界面將使用的連接埠，以及指定安全群組來控制對界面的存取。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
4. 如有需要，修改選項設定以便設定 memcached 參數。如需如何修改選項設定的詳細資訊，請參閱 [修改選項設定](#)。
5. 將選項群組套用至執行個體。一旦套用選項群組，Amazon RDS 即會啟用該執行個體的 memcached 支援。
 - 當您在啟動新執行個體時指定自訂選項群組，即啟用了新執行個體的 memcached 支援。如需有關啟動 MySQL 執行個體的詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 當您在修改現有執行個體時指定自訂選項群組，即啟用了現有執行個體的 memcached 支援。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
6. 指定可以透過 memcached 界面存取 MySQL 資料表中的哪幾欄。memcached 外掛程式會在名為 containers 的專用資料庫中建立名為 innodb_memcache 的目錄資料表。您在 containers 資料表中插入一個資料列，以映射透過 memcached 存取的 InnoDB 資料表。指定 InnoDB 資料表中的一欄用於存放 memcached 索引鍵值，並指定一或多欄用於存放與此索引鍵相關聯的資料值。也要指定名稱，讓 memcached 應用程式可用來指向這幾欄。如需在 containers 資料表中插入資料列的詳細資訊，請參閱 [InnoDB Memcached 外掛程式內部](#)。如需映射 InnoDB 資料表以及透過 memcached 存取它的範例，請參閱 [撰寫 InnoDB Memcached 外掛程式的應用程式](#)。

7. 如果存取 memcached 界面的應用程式所在的電腦或 EC2 執行個體，和使用 SQL 界面的應用程式所在的不同，請為這些電腦新增連線資訊，以連線至與 MySQL 執行個體關聯的 VPC 安全群組。如需管理安全群組的詳細資訊，請參閱 [使用安全群組控制存取](#)。

修改執行個體並指定您的 MySQL 版本的預設選項群組，即可關閉執行個體的 memcached 支援。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

MySQL memcached 安全考量

memcached 通訊協定不支援使用者驗證。如需 MySQL memcached 安全性考量的詳細資訊，請參閱 MySQL 文件中的 [InnoDB Memcached 外掛程式的安全性考量](#)。

您可採取以下動作幫助增加 memcached 界面的安全：

- 在將 MEMCACHED 選項新增至選項群組時，指定非預設 (11211) 的連接埠。
- 確保您將 memcached 界面與 VPC 安全群組 (會限制對已知信任用戶端地址或 EC2 執行個體的存取) 相關聯。如需管理安全群組的詳細資訊，請參閱 [使用安全群組控制存取](#)。

MySQL memcached 連線資訊

應用程式必須同時指定 Amazon RDS 執行個體的 DNS 名稱與 memcached 連接埠號碼，才能存取 memcached 界面。例如，如果執行個體的 DNS 名稱是 my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com，且 memcached 界面使用 11212 連接埠，則在 PHP 中指定的連線資訊就是：

```
<?php
$cache = new Memcache;
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com',11212);
?>
```

尋找 MySQL 資料庫執行個體的 DNS 名稱及 Memcached 連接埠

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在的右上角 AWS Management Console，選取包含資料庫執行個體的區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。

4. 選擇 MySQL 資料庫執行個體名稱以顯示其詳細資訊。
5. 請注意 Connect (連線) 區段中 Endpoint (端點) 欄位的值。DNS 名稱與端點名稱相同。此外，請注意 Connect (連線) 區段中的連接埠不是用於存取 memcached 界面。
6. 在 Details (詳細資訊) 區段中，注意 Option Group (選項群組) 欄位中列出的名稱。
7. 在導覽窗格中，選擇 Option groups (選項群組)。
8. 選擇 MySQL 資料庫執行個體使用的選項群組名稱，以顯示選項群組的詳細資訊。在 Options (選項) 區段中，請注意 MEMCACHED 選項的 Port (連接埠) 設定值。

MySQL memcached 選項設定

Amazon RDS 會公開 MySQL memcached 參數，並將其做為 Amazon RDS MEMCACHED 選項中的選項設定。

MySQL memcached 參數

- DAEMON_MEMCACHED_R_BATCH_SIZE – 整數，指定要執行多少個 memcached 讀取操作 (get) 之後才 COMMIT (遞交) 開始新交易。允許的值為 1 到 4294967295；預設值是 1。此選項要在執行個體重新啟動後才會生效。
- DAEMON_MEMCACHED_W_BATCH_SIZE – 整數，指定要執行多少個 memcached 寫入操作 (例如 add、set 或 incr) 之後才 COMMIT (遞交) 開始新交易。允許的值為 1 到 4294967295；預設值是 1。此選項要在執行個體重新啟動後才會生效。
- INNODB_API_BK_COMMIT_INTERVAL – 整數，指定自動遞交使用 InnoDB memcached 界面的閒置連線的遞交頻率。允許的值為 1 到 1073741824；預設值是 5。此選項會立即生效，不需要重新啟動執行個體。
- INNODB_API_DISABLE_ROWLOCK – 布林值，用於停用 (1 (true)) 或啟用 (0 (false)) 使用 InnoDB memcached 界面時的資料列鎖定。預設值為 0 (false)。此選項要在執行個體重新啟動後才會生效。
- INNODB_API_ENABLE_MDL – 布林值，設為 0 (false) 可鎖定 InnoDB memcached 外掛程式所使用的資料表，使您無法透過 SQL 界面刪除資料表或以 DDL 更改資料表。預設值為 0 (false)。此選項要在執行個體重新啟動後才會生效。
- INNODB_API_TRX_LEVEL – 整數，指定 memcached 界面處理的查詢的交易隔離層級。允許的值為 0 到 3。預設為 0。此選項要在執行個體重新啟動後才會生效。

Amazon RDS 會設定這些 MySQL memcached 參數，並且無法修改以下項目：DAEMON_MEMCACHED_LIB_NAME、DAEMON_MEMCACHED_LIB_PATH 與

INNODB_API_ENABLE_BINLOG。MySQL 管理員使用 `daemon_memcached_options` 所設定的參數，在 Amazon RDS 中皆為個別的 MEMCACHED 選項設定。

MySQL `daemon_memcached_options` 參數

- `BINDING_PROTOCOL` – 字串，指定要使用的繫結通訊協定。允許的值為 `auto`、`ascii` 或 `binary`。預設值是 `auto`，表示伺服器會自動與用戶端交涉通訊協定。此選項要在執行個體重新啟動後才會生效。
- `BACKLOG_QUEUE_LIMIT` – 整數，指定可以有幾個等待處理的網路連線。memcached 增加此限制可能會減少無法連線到 memcached 執行個體的用戶端收到的錯誤，但不會提高伺服器的效能。允許的值為 1 到 2048；預設值是 1024。此選項要在執行個體重新啟動後才會生效。
- `CAS_DISABLED` – 布林值，可啟用 (1 (true)) 或停用 (0 (false)) 比較與交換 (CAS)，CAS 可將每個項目的大小減少 8 個位元組。預設值為 0 (false)。此選項要在執行個體重新啟動後才會生效。
- `CHUNK_SIZE` – 整數，指定可為最小項目的金鑰、值、標記配置的區塊大小下限 (位元組)。允許的值為 1 到 48。預設值為 48，使用較小的值可大幅增進記憶體效能。此選項要在執行個體重新啟動後才會生效。
- `CHUNK_SIZE_GROWTH_FACTOR` – 浮動數值，控制新區塊的大小。新區塊的大小是先前區塊大小乘以 `CHUNK_SIZE_GROWTH_FACTOR`。允許的值為 1 到 2；預設值是 1.25。此選項要在執行個體重新啟動後才會生效。
- `ERROR_ON_MEMORY_EXHAUSTED` – 布林值，設為 1 (true) 即指定在已經沒有記憶體可儲存項目時，memcached 要傳回錯誤而不是移出項目。如果設為 0 (false)，memcached 會在沒有記憶體時移出項目。預設值為 0 (false)。此選項要在執行個體重新啟動後才會生效。
- `MAX_SIMULTANEOUS_CONNECTIONS` – 整數，指定並行連線的數量上限。設為小於 10 的值可阻止 MySQL 啟動。允許的值為 10 到 1024；預設值是 1024。此選項要在執行個體重新啟動後才會生效。
- `VERBOSITY` – 字串，指定 memcached 服務在 MySQL 錯誤日誌中記錄的資訊層級。預設值為 `v`。此選項要在執行個體重新啟動後才會生效。允許的值為：
 - `v` – 在執行主要事件迴圈時，記錄錯誤並發出警告。
 - `vv` – 除了 `v` 記錄的資訊之外，也記錄每個用戶端命令和回應。
 - `vvv` – 除了 `w` 記錄的資訊之外，也記錄內部狀態轉換。

Amazon RDS 會自行設定以下不可修改的 MySQL `DAEMON_MEMCACHED_OPTIONS` 參

數：`DAEMON_PROCESS`、`LARGE_MEMORY_PAGES`、`MAXIMUM_CORE_FILE_LIMIT`、`MAX_ITEM_SIZE`、`LOG` 與 `USER`。

MySQL 的參數

依預設，MySQL 資料庫執行個體會使用 MySQL 資料庫專用的資料庫參數群組。此參數群組包含 MySQL 資料庫引擎的參數。如需使用參數群組和設定參數的相關資訊，請參閱[使用參數群組](#)。

RDS for MySQL 參數會設為您已選取之儲存引擎的預設值。如需 MySQL 參數的詳細資訊，請參閱[MySQL 文件](#)。如需 MySQL 儲存引擎的詳細資訊，請參閱[支援的 RDS for MySQL 儲存引擎](#)。

您可以使用 RDS 主控台或 AWS CLI 查看特定 RDS for MySQL 版本的可用參數。如需有關在 RDS 主控台中查看 MySQL 參數群組參數的資訊，請參閱[檢視資料庫參數群組的參數值](#)。

藉由 AWS CLI，您可以執行 `describe-engine-default-parameters` 命令來查看 RDS for MySQL 版本的參數。為 `--db-parameter-group-family` 選項指定下列其中一個值：

- `mysql8.0`
- `mysql5.7`

例如，若要檢視 RDS for MySQL 8.0 版的參數，請執行下列命令。

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0
```

您的輸出結果類似以下內容。

```
{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "activate_all_roles_on_login",
        "ParameterValue": "0",
        "Description": "Automatically set all granted roles as active after the user has authenticated successfully.",
        "Source": "engine-default",
        "ApplyType": "dynamic",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": true
      },
      {
        "ParameterName": "allow-suspicious-udfs",
        "Description": "Controls whether user-defined functions that have only an xxx symbol for the main function can be loaded",
```

```
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": false
    },
    {
        "ParameterName": "auto_generate_certs",
        "Description": "Controls whether the server autogenerates SSL key and
certificate files in the data directory, if they do not already exist.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": false
    },
    ...
```

若要檢視 RDS for MySQL 8.0 版的可修改參數，請執行下列命令。

對於LinuxmacOS、或Unix：

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 \
--query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

在Windows中：

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 ^
--query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

MySQL 資料庫執行個體的常用 DBA 任務

在下列內容中，您可以找到針對執行 MySQL 資料庫引擎的資料庫執行個體的一些常見 DBA 任務之 Amazon RDS 特定實作的說明。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。此外，其也會將存取權限制在某些需要進階權限的系統程序和資料表。

如需在 Amazon RDS 上使用 MySQL 日誌檔案的詳細資訊，請參閱 [MySQL 資料庫日誌檔案](#)

主題

- [了解預先定義的](#)
- [角色型權限模型](#)
- [結束工作階段或查詢](#)
- [略過目前複寫錯誤](#)
- [使用 InnoDB 資料表空間來縮短損毀復原時間](#)
- [管理全域狀態歷史記錄](#)

了解預先定義的

Amazon RDS 使用新的適用於 MySQL 資料庫執行個體的 RDS 自動建立多個預先定義 無法變更預先定義的使用者及其權限。您無法刪除、重新命名或修改這些預先定義使用者的權限。嘗試這麼做會造成錯誤。

- `rdsadmin` — 建立來處理許多具有 `superuser` 權限的管理員會在獨立 MySQL 資料庫上執行的管理工作的使用者。此使用者會由 RDS for MySQL 用於內部使用，以執行許多管理工作。
- `rdsrepladmin` — Amazon RDS 在內部使用的使用者，用來支援適用於 MySQL 資料庫執行個體和叢集的 RDS 複寫活動。

角色型權限模型

從適用於 MySQL 的 RDS 版本 8.0.36 開始，您無法直接修改 `mysql` 資料庫中的資料表。特別是，您無法透過對資料 `grant` 表執行資料操作語言 (DML) 作業來建立資料庫使用者。而是使用 MySQL 帳戶管理陳述式 (例如、和) `CREATE USERGRANT`，`REVOKE` 將以角色為基礎的權限授與使用者。您也無法建立其他類型的物件，例如 `mysql` 資料庫中已存放的程序。您仍然可以查詢 `mysql` 資料表。如果您使用二進位記錄複寫，直接對來源資料庫執行個體上的 `mysql` 表格所做的變更不會複寫到目標叢集。

在某些情況下，您的應用程式可能會使用捷徑來建立使用者或其他物件，方法是插入至 `mysql` 資料表。若是這樣，請變更您應用程式的程式碼來使用對應的陳述式，例如 `CREATE USER`。

若要在從外部 MySQL 資料庫移轉期間匯出資料庫使用者的詮釋資料，請使用下列其中一種方法：

- 使用 MySQL Shell 的執行個體傾印公用程式搭配篩選器來排除使用者、角色和授權。下列範例顯示要使用的命令語法。確保它 `outputUrl` 是空的。

```
mysqlsh user@host -- util.dumpInstance(outputUrl,{excludeSchemas:['mysql'],users:
true})
```

如需詳細資訊，請參閱 MySQL 參考手冊中的 [執行個體傾印公用程式](#)、[架構傾印公用程式](#) 和 [資料表傾印公用程式](#)。

- 使用用戶 `mysqlpump` 端公用程式。此範例包括除了 `mysql` 系統資料庫中的資料表以外的所有資料表。它也包括 `CREATE USER` 和 `GRANT` 陳述式，來重新產生所遷移資料庫中的所有 MySQL 使用者。

```
mysqlpump --exclude-databases=mysql --users
```

若要簡化許多使用者或應用程式的權限管理，您可以使用 `CREATE ROLE` 陳述式來建立具有一組權限的角色。然後，您可以使用 `GRANT` 和 `SET ROLE` 陳述式，以及 `current_role` 函數，將角色指派給使用者或應用程式、切換目前角色，並檢查哪些角色有效。如需 MySQL 8.0 中角色型權限系統的詳細資訊，請參閱《MySQL 參考手冊》中的 [使用角色](#)。

Important

我們強烈建議您不要直接在您的應用程式中使用主要使用者。而是遵循最佳實務，使用以應用程式所需的最低權限建立的資料庫使用者。

從 8.0.36 版開始，適用於 MySQL 的 RDS 包含具有下列所有權限的特殊角色。此角色已命名為 `rds_superuser_role`。每個資料庫執行個體的主要管理使用者都已授與此角色。`rds_superuser_role` 角色包含所有資料庫物件的下列權限：

- ALTER
- APPLICATION_PASSWORD_ADMIN
- ALTER ROUTINE

- CREATE
- CREATE ROLE
- CREATE ROUTINE
- CREATE TEMPORARY TABLES
- CREATE USER
- CREATE VIEW
- DELETE
- DROP
- DROP ROLE
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES
- PROCESS
- REFERENCES
- RELOAD
- REPLICATION CLIENT
- REPLICATION SLAVE
- ROLE_ADMIN
- SET_USER_ID
- SELECT
- SHOW DATABASES
- SHOW VIEW
- TRIGGER
- UPDATE
- XA_RECOVER_ADMIN

角色定義還包括 WITH GRANT OPTION，以便管理使用者可以將該角色授與其他使用者。特別是，系統管理員必須授與以 MySQL 叢集做為目標執行二進位記錄複寫所需的任何權限。

i Tip

若要查看權限的完整詳細資料，請使用下列陳述式。

```
SHOW GRANTS FOR rds_superuser_role@'%';
```

當您使用適用於 MySQL 版本 8.0.36 及更新版本的 RDS 中的角色來授與存取權時，也可以使用 SET ROLE *role_name* 或 SET ROLE ALL 陳述式來啟動角色。下列範例會顯示作法。以適當的角色名稱替代 CUSTOM_ROLE。

```
# Grant role to user
mysql> GRANT CUSTOM_ROLE TO 'user'@'domain-or-ip-address'

# Check the current roles for your user. In this case, the CUSTOM_ROLE role has not
  been activated.
# Only the rds_superuser_role is currently in effect.
mysql> SELECT CURRENT_ROLE();
+-----+
| CURRENT_ROLE()          |
+-----+
| `rds_superuser_role`@`%` |
+-----+
1 row in set (0.00 sec)

# Activate all roles associated with this user using SET ROLE.
# You can activate specific roles or all roles.
# In this case, the user only has 2 roles, so we specify ALL.
mysql> SET ROLE ALL;
Query OK, 0 rows affected (0.00 sec)

# Verify role is now active
mysql> SELECT CURRENT_ROLE();
+-----+
| CURRENT_ROLE()          |
+-----+
| `CUSTOM_ROLE`@`%`,`rds_superuser_role`@`%` |
+-----+
```

結束工作階段或查詢

您可以使用 `rds_kill` 和 `rds_kill_query` 命令，結束資料庫執行個體上的使用者工作階段或查詢。首先連接到您的 MySQL 資料庫執行個體，然後發出適當的命令，如下所示。如需更多詳細資訊，請參閱 [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

例如，若要結束正在執行緒 99 上執行的工作階段，您將輸入下列命令：

```
CALL mysql.rds_kill(99);
```

若要結束正在執行緒 99 上執行的查詢，您將輸入下列命令：

```
CALL mysql.rds_kill_query(99);
```

略過目前複寫錯誤

您可以略過僅供讀取複本上的錯誤，前提是此錯誤導致僅供讀取複本停止回應，而且此錯誤不會影響資料的完整性。

Note

首先驗證是否可以安全地略過有問題的錯誤。在 MySQL 公用程式中，連線至僅供讀取複本並執行下列 MySQL 命令：

```
SHOW REPLICA STATUS\G
```

如需傳回值的相關資訊，請參閱 [MySQL 文件](#)。

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

您可採用以下方式，略過僅供讀取複本上的錯誤。

主題

- [呼叫 mysql.rds_skip_repl_error 程序](#)
- [設定 slave_skip_errors 參數](#)

呼叫 mysql.rds_skip_repl_error 程序

Amazon RDS 提供一種預存程序，您可以呼叫此程序，以略過僅供讀取複本上的錯誤。首先連接到您的僅供讀取複本，然後發出適當的命令，如下所示。如需詳細資訊，請參閱 [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。

若要略過錯誤，請發出下列命令：

```
CALL mysql.rds_skip_repl_error;
```

如果您在來源資料庫執行個體上，或在未發現複寫錯誤的僅供讀取複本上執行此命令，則此命令沒有效用。

如需詳細資訊，例如支援 mysql.rds_skip_repl_error 的 MySQL 版本，請參閱 [mysql.rds_skip_repl_error](#)。

Important

如果您嘗試呼叫 mysql.rds_skip_repl_error 並遇到下列錯誤：ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist，請將 MySQL 資料庫執行個體升級到最新次要版本，或 [mysql.rds_skip_repl_error](#) 列出的其中一個最低的次要版本。

設定 slave_skip_errors 參數

若要略過一或多個錯誤，您可以在僅供讀取複本上設定 slave_skip_errors 靜態參數。您可以設定此參數，以略過一或多個特定的複寫錯誤碼。目前，您只能針對 RDS for MySQL 5.7 資料庫執行個體設定此參數。在變更此參數的設定之後，請務必重新啟動您的資料庫執行個體，新設定才會生效。如需設定此參數的相關資訊，請參閱 [MySQL 文件](#)：

建議您在個別的資料庫參數群組中設定此參數。您只能將此資料庫參數群組與需要略過錯誤的僅供讀取複本建立關聯。遵循此最佳實務可減少對其他資料庫執行個體和僅供讀取複本的潛在影響。

⚠ Important

針對此參數設定非預設值可能會導致複寫不一致。只有在您已用盡其他選項來解決問題，並確定對僅供讀取複本資料產生潛在影響時，才會將此參數設定為非預設值。

使用 InnoDB 資料表空間來縮短損毀復原時間

MySQL 中的每個資料表都包含資料表定義、資料和索引。MySQL 儲存引擎 InnoDB 會將資料表資料和索引儲存在資料表空間中。InnoDB 會建立全域共用的資料表空間，其中包含資料字典和其他相關中繼資料，而且它可以包含資料表資料和索引。InnoDB 也可以針對每個資料表和分割區建立個別的資料表空間。這些個別的資料表空間會儲存在副檔名為 .ibd 的檔案中，而且每個資料表空間的標題都包含一個專門識別它的數字。

Amazon RDS 會在 MySQL 參數群組中提供名為 `innodb_file_per_table` 的參數。此參數會控制 InnoDB 是否將新的資料表資料和索引加入至共用的資料表空間 (方法為將參數值設為 0) 或加入至個別的資料表空間 (方法為將參數值設為 1)。Amazon RDS 會將 `innodb_file_per_table` 參數的預設值設為 1，這允許您捨棄個別的 InnoDB 資料表，並回收那些資料表針對資料庫執行個體使用的儲存體。在大部分使用案例中，將 `innodb_file_per_table` 參數設為 1 是建議的設定。

當您有大量的資料表 (例如，當您使用標準 (磁性) 或一般用途 SSD 儲存體時超過 1000 個資料表，或當您使用佈建 IOPS 儲存體時超過 10,000 個資料表) 時，您應該將 `innodb_file_per_table` 參數設為 0。將此參數設為 0 時，不會建立個別資料表空間，而且這樣做可縮短資料庫損毀復原所需的時間。

MySQL 會在損毀復原週期處理每個中繼檔案。相較於有多個資料表時，處理數千個資料表檔案所需的時間，可以忽略 MySQL 在共用的資料表空間中處理中繼資訊所需的時間。因為資料表空間號碼會儲存在每個檔案的標題內，所以讀取所有資料表空間檔案的合計時間可能高達七小時。例如，標準儲存體上的一百萬個 InnoDB 資料表空間在損毀復原週期可能需要五到八小時的處理時間。在某些情況下，InnoDB 可以判斷在損毀復原週期之後是否需要額外清除，以便它將開始另一個損毀復原週期，這將延長復原時間。請記住，除了處理資料表空間資訊外，損毀復原週期也會包含轉返交易、修正中斷的頁面，以及其他操作。

因為 `innodb_file_per_table` 參數位於常數群組中，所以您可以變更參數值，方法為編輯資料庫執行個體所使用的參數群組，而不必重新啟動資料庫執行個體。例如，在將設定從 1 (建立個別資料表) 變更為 0 (使用共用的資料表空間) 之後，當現有的資料表繼續具有個別資料表空間時，新的 InnoDB 資料表將新增至共用的資料表空間。若要將 InnoDB 資料表移至共用的資料表空間，您必須使用 `ALTER TABLE` 命令。

將多個資料表空間遷移至共用的資料表空間

您可以將 InnoDB 資料表的中繼資料移至它自己的資料表空間，這將根據 `innodb_file_per_table` 參數設定，重建資料表中繼資料。首先連接到您的 MySQL 資料庫執行個體，然後發出適當的命令，如下所示。如需更多詳細資訊，請參閱 [連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

例如，以下查詢會針對不在共用的資料表空間中的每一個 InnoDB 資料表傳回 ALTER TABLE 陳述式。

若為 MySQL 5.7 資料庫執行個體：

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), `.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), ` ENGINE=InnoDB,
ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

適用於 MySQL 8.0 資料庫執行個體：

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), `.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), ` ENGINE=InnoDB,
ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

重建 MySQL 資料表將資料表的中繼資料移至共用的資料表空間，暫時需要額外的儲存空間來重建資料表，所以資料庫執行個體必須具有可用的儲存空間。重建期間，會鎖定資料表，而且查詢無法存取它。對於不經常存取的小型表格或表格，可能不需要擔心此問題。對於在大量並行環境中經常存取的大型表格或表格，您可以在僅供讀取複本上重建資料表。

您可以建立僅供讀取複本，並將資料表中繼資料遷移至僅供讀取複本上共用的資料表空間。當 ALTER TABLE 陳述式封鎖僅供讀取複本上的存取權時，來源資料庫執行個體不會受到影響。在資料表重建過程中，當僅供讀取複本延遲時，來源資料庫執行個體將繼續產生其二進位記錄。由於重建需要額外的儲存空間，重播日誌檔案可能因此變大，所以您應該建立其配置的儲存體大於來源資料庫執行個體的僅供讀取複本。

若要建立僅供讀取複本並重建 InnoDB 資料表，以使用共用的資料表空間，請採取以下步驟：

1. 確保已在來源資料庫執行個體上啟用備份保留，以便啟用二進位記錄。
2. 使用 AWS Management Console 或建立 AWS CLI 來源資料庫執行個體的僅供讀取複本。因為建立僅供讀取複本涉及許多與損毀復原相同的程序，所以如果有大量 InnoDB 資料表空間，則建立程序可能需要一些時間。在僅供讀取複本上配置更多的儲存空間，超過目前在來源資料庫執行個體上使用的儲存空間。
3. 建立僅供讀取複本後，請使用參數設定 `read_only = 0` 和 `innodb_file_per_table = 0` 建立參數群組。接著請將參數群組與僅供讀取複本產生關聯。
4. 針對您要在複本上遷移的所有資料表，發出下列 SQL 陳述式：

```
ALTER TABLE name ENGINE = InnoDB
```

5. 在僅供讀取複本上完成了所有 ALTER TABLE 陳述式時，請驗證僅供讀取複本是否連線至來源資料庫執行個體，以及這兩個執行個體是否同步。
6. 使用主控台或 CLI 將僅供讀取複本升級為執行個體。確定用於新獨立資料庫執行個體的參數群組已將 `innodb_file_per_table` 參數設為 0。變更新獨立資料庫執行個體的名稱，並將任何應用程式指向新的獨立資料庫執行個體。

管理全域狀態歷史記錄

Tip

若要分析資料庫效能，您也可以使用 Amazon RDS 上的 Performance Insights。如需詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。

MySQL 會維護許多提供其操作相關資訊的狀態變數。此值可協助您偵測資料庫執行個體上的鎖定或記憶體問題。這些狀態變數的值是從上次啟動資料庫執行個體以來累積的。您可以使用 FLUSH STATUS 命令，將大部分狀態變數重設為 0。

為了允許在一段時間內監控這些值，Amazon RDS 提供一組程序，將在一段時間內快照這些狀態變數的值，並將它們以及自從上次快照後的任何變更寫入至資料表。此基礎設施 (稱為全域狀態歷史記錄 (GoSH)) 安裝在所有從版本 5.5.23 開始的 MySQL 資料庫執行個體。GoSH 預設為停用。

若要啟用 GoSH，首先從資料庫參數群組啟用事件排程器，方法是將參數 `event_scheduler` 設為 ON。對於執行 MySQL 5.7 的 MySQL 資料庫執行個體，同樣將參數 `show_compatibility_56` 設為

1. 如需建立和修改資料庫參數群組的詳細資訊，請參閱[使用參數群組](#)。如需啟用此參數的副作用相關資訊，請參閱《MySQL 5.7 參考手冊》中的 [show_compatibility_56](#)。

然後，您可以使用下表中的程序，來啟用和設定 GoSH。首先連接到您的 MySQL 資料庫執行個體，然後發出適當的命令，如下所示。如需更多詳細資訊，請參閱[連接至執行 MySQL 資料庫引擎的資料庫執行個體](#)。對於每個程序，輸入下列命令：

```
CALL procedure-name;
```

其中 `procedure-name` 是資料表中的其中一個程序。

程序	描述
<code>mysql.rds_enable_gsh_collector</code>	啟用 GoSH 來依據 <code>rds_set_gsh_collector</code> 指定的間隔建立預設快照。
<code>mysql.rds_set_gsh_collector</code>	指定快照之間的時間隔 (以分鐘為單位)。預設值為 5。
<code>mysql.rds_disable_gsh_collector</code>	停用快照。
<code>mysql.rds_collect_global_status_history</code>	隨需建立快照。
<code>mysql.rds_enable_gsh_rotation</code>	啟用依 <code>mysql.rds_global_status_history</code> 指定的間隔將 <code>mysql.rds_global_status_history_old</code> 資料表的內容輪換至 <code>rds_set_gsh_rotation</code> 。
<code>mysql.rds_set_gsh_rotation</code>	指定資料表輪換之間的時間隔 (以天為單位)。預設值為 7。
<code>mysql.rds_disable_gsh_rotation</code>	停用資料表輪換。
<code>mysql.rds_rotate_global_status_history</code>	隨需將 <code>mysql.rds_global_status_history</code> 資料表的內容輪換至 <code>mysql.rds_global_status_history_old</code> 。

當 GoSH 執行中時，您可以查詢寫入它的資料表。例如，若要查詢 Innodb 緩衝集區的命中率，您將發出下列查詢：

```
select a.collection_end, a.collection_start, (( a.variable_Delta-b.variable_delta)/
a.variable_delta)*100 as "HitRatio"
  from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b
 on a.collection_end = b.collection_end
  where a.variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
 'Innodb_buffer_pool_reads'
```


MySQL 資料庫執行個體的本機時區

MySQL 資料庫執行個體的時區預設為國際標準時間 (UTC)。您也可針對應用程式，將資料庫執行個體的時區設為本機時區。

若要設定資料庫執行個體的本機時區，請將資料庫執行個體參數群組中的 `time_zone` 參數設為本節稍後所列任一支援的值。當您設定參數群組的 `time_zone` 參數後，原本使用該參數群組的所有資料庫執行個體和僅供讀取複本會改為使用新的本機時區。如需設定參數群組參數的相關資訊，請參閱[使用參數群組](#)。

設定本機時區後，所有資料庫的新連線都會反映此變更。如果在您變更本機時區時有任何已開啟的連線，在您關閉連線然後開啟新連線後，才會看見本機時區更新。

您可為資料庫執行個體以及其中一或多個僅供讀取複本，設定不同的本機時區。若要執行這項操作，請為該資料庫執行個體和僅供讀取複本使用不同的參數群組，然後將各個參數群組中的 `time_zone` 參數設定為不同的本機時區。

如果您正在執行跨 AWS 區域 複寫作業，來源資料庫執行個體和僅供讀取複本會使用不同的參數群組 (參數群組為 AWS 區域 專屬)。若要在每個執行個體使用相同的本機時區，您必須設定執行個體和僅供讀取複本之參數群組中的 `time_zone` 參數。

從資料庫快照還原資料庫執行個體時，本機時區會設為 UTC。您也可在還原作業完成後，將時區更新為本機時區。如果您將資料庫執行個體還原至某個時間點，該還原資料庫執行個體的本機時區會使用還原資料庫執行個體參數群組中的時區設定。

網際網路號碼分配局 (IANA) 每年會在 <https://www.iana.org/time-zones> 發布數次新時區。每次 RDS 發布 MySQL 的新次要維護版本時，它都會在發布時隨附最新的時區資料。當您使用最新的 RDS for MySQL 版本時，您會有來自 RDS 的最新時區資料。為確保您的資料庫執行個體擁有最新的時區資料，建議您升級至更高的資料庫引擎版本。或者，您也可以手動修改 MariaDB 資料庫執行個體中的時區表。若要這麼做，您可以使用 SQL 命令或在 SQL 用戶端執行 [mysql_tzinfo_to_sql tool](#)。手動更新時區資料後，請重新啟動資料庫執行個體以讓變更生效。RDS 不會修改或重設執行中資料庫執行個體的時區資料。只有在您執行資料庫引擎版本升級時，才會安裝新的時區資料。

您可以將本機時區設為下列任一值。

Africa/Cairo	Asia/Riyadh
Africa/Casablanca	Asia/Seoul
Africa/Harare	Asia/Shanghai

Africa/Monrovia	Asia/Singapore
Africa/Nairobi	Asia/Taipei
Africa/Tripoli	Asia/Tehran
Africa/Windhoek	Asia/Tokyo
America/Araguaina	Asia/Ulaanbaatar
America/Asuncion	Asia/Vladivostok
America/Bogota	Asia/Yakutsk
America/Buenos_Aires	Asia/Yerevan
America/Caracas	Atlantic/Azores
America/Chihuahua	Australia/Adelaide
America/Cuiaba	Australia/Brisbane
America/Denver	Australia/Darwin
America/Fortaleza	Australia/Hobart
America/Guatemala	Australia/Perth
America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens

America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa
Asia/Karachi	US/Alaska
Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

Amazon RDS for MySQL 的已知問題和限制

使用 Amazon RDS for MySQL 的已知問題和限制如下所示。

主題

- [InnoDB 保留字詞](#)
- [Amazon RDS for MySQL 的儲存已滿行為](#)
- [InnoDB 緩衝集區大小不一致](#)
- [索引合併最佳化傳回不正確的結果](#)
- [Amazon RDS 資料庫執行個體的 MySQL 參數例外狀況](#)
- [Amazon RDS 中的 MySQL 檔案大小限制](#)
- [不支援 MySQL Keyring 外掛程式](#)
- [自訂連接埠](#)
- [MySQL 預存程序限制](#)
- [使用外部來源執行個體進行 GTID 式複寫](#)
- [MySQL 默認身份驗證插件](#)
- [覆蓋緩衝池大小](#)

InnoDB 保留字詞

InnoDB 是用於 RDS for MySQL 的保留字詞。此名稱無法用於 MySQL 資料庫。

Amazon RDS for MySQL 的儲存已滿行為

當 MySQL 資料庫執行個體的儲存空間已滿時，可能會出現中繼資料不一致、字典不符和孤立表格。若要避免這些問題，Amazon RDS 應自動停止達到 storage-full 狀態的資料庫執行個體。

MySQL 資料庫執行個體在下列情況下達到 storage-full 狀態：

- 資料庫執行個體的儲存空間少於 20,000 MiB，可用儲存空間達到 200 MiB 或更少。
- 資料庫執行個體擁有超過 102,400 MiB 的儲存空間，可用儲存空間達到 1024 MiB 或更少。
- 資料庫執行個體的儲存空間介於 20,000 MiB 和 102,400 MiB 之間，且可用的儲存空間少於 1%。

由於資料庫執行個體達到 storage-full 狀態，因此 Amazon RDS 自動停止它之後，您仍然可以進行修改。若要重新啟動資料庫執行個體，請至少完成下列其中一項：

- 修改資料庫執行個體以啟用儲存體自動擴展。

如需儲存體自動擴展的詳細資訊，請參閱[使用 Amazon RDS 儲存體自動擴展自動管理容量](#)。

- 修改資料庫執行個體以增加其儲存容量。

如需增加儲存容量的詳細資訊，請參閱[增加資料庫執行個體的儲存容量](#)。

進行其中一項變更後，資料庫執行個體會自動重新啟動。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

InnoDB 緩衝集區大小不一致

若為 MySQL 5.7，目前在管理 InnoDB 緩衝集區大小的方式中有一個錯誤。MySQL 5.7 可能將 `innodb_buffer_pool_size` 參數的值調整為非常大的值，因而導致 InnoDB 緩衝集區過大並使用過多的記憶體。此效果可能導致 MySQL 資料庫引擎停止執行，或可能阻止其啟動。對於可用記憶體更少的資料庫執行個體類別，此問題更常發生。

若要解決此問題，請將 `innodb_buffer_pool_size` 參數的值設為 `innodb_buffer_pool_instances` 參數值與 `innodb_buffer_pool_chunk_size` 參數值之乘積的倍數。例如，您可能將 `innodb_buffer_pool_size` 參數值設為 `innodb_buffer_pool_instances` 與 `innodb_buffer_pool_chunk_size` 參數值之乘積的八倍，如下列範例所示。

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

如需此 MySQL 5.7 錯誤的詳細資訊，請參閱 MySQL 文件中的 <https://bugs.mysql.com/bug.php?id=79379>。

索引合併最佳化傳回不正確的結果

由於 MySQL 5.5.37 推出的 MySQL 查詢最佳化工具有錯誤，使用索引合併最佳化的查詢可能會傳回不正確的結果。當您對具有多個索引的資料表發出查詢請求，最佳化工具可以根據多個索引來掃描資料列範圍，卻無法正確地合併結果。如需查詢最佳化器錯誤的詳細資訊，請參閱 MySQL 錯誤資料庫中的 <http://bugs.mysql.com/bug.php?id=72745> 和 <http://bugs.mysql.com/bug.php?id=68194>。

例如，考慮對具有兩個索引的資料表進行查詢，其中搜尋引數會參考具索引的資料欄。

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

在此情況下，搜尋引擎將搜尋這兩個索引。不過，由於錯誤，合併的結果會不正確。

若要解決此問題，您可以執行下列項目之一：

- 對於您的 MySQL 資料庫執行個體，將 `optimizer_switch` 參數設為資料庫參數群組中的 `index_merge=off`。如需設定資料庫參數群組參數的相關資訊，請參閱[使用參數群組](#)。
- 將 MySQL 資料庫執行個體升級為 MySQL 5.7 或 8.0 版。如需詳細資訊，請參閱[升級 MySQL 資料庫引擎](#)。
- 如果您無法升級執行個體或變更 `optimizer_switch` 參數，則可以藉由明確地識別查詢的索引來解決錯誤，例如：

```
SELECT * FROM table1
USE INDEX covering_index
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

如需詳細資訊，請參閱 MySQL 文件中的[索引合併最佳化](#)。

Amazon RDS 資料庫執行個體的 MySQL 參數例外狀況

與 Amazon RDS 資料庫執行個體搭配使用時，部分 MySQL 參數需要特殊考量。

`lower_case_table_names`

因為 Amazon RDS 會使用區分大小寫檔案系統，所以不支援將 `lower_case_table_names` 伺服器參數的值設為 2 (名稱按原狀儲存，但以小寫形式進行比較)。以下是 Amazon RDS for MySQL 資料庫執行個體支援的值：

- 所有 RDS for MySQL 版本都支援 0 (名稱按原狀儲存，且比較會區分大小寫)。
- RDS for MySQL 5.7 版和 8.0.28 版和更新的 8.0 版支援 1。(名稱儲存為小寫，且比較不區分大小寫)。

在建立資料庫執行個體之前，請在自訂資料庫參數群組中設定 `lower_case_table_names` 參數。然後，在建立資料庫執行個體時，指定自訂資料庫參數群組。

當參數群組與版本低於 8.0 的 MySQL 資料庫執行個體相關聯時，建議您避免變更參數群組中的 `lower_case_table_names` 參數。變更它可能會導致 point-in-time 復原備份和僅供讀取複本資料庫執行個體不一致。

當參數群組與 8.0 版的 MySQL 資料庫執行個體相關聯時，您無法修改參數群組中的 `lower_case_table_names` 參數。

僅供讀取複本應該一律使用與來源資料庫執行個體相同的 `lower_case_table_names` 參數值。

長查詢時間

您可以將 `long_query_time` 參數設為浮點值，以便您可以利用微秒解析，將緩慢查詢記錄至 MySQL 緩慢查詢日誌。您可以設定一值，例如 0.1 秒 (即 100 微秒)，以在偵錯所花時間不到一秒的緩慢交易時提供協助。

Amazon RDS 中的 MySQL 檔案大小限制

對於 MySQL 資料庫執行個體，在使用 InnoDB file-per-table 表空間時，佈建的最大儲存限制會將資料表的大小限制為 16 TB 的大小上限。此限制也會將系統資料表空間限制為最大 16 TB。InnoDB 表 file-per-table 空間 (每個表都在自己的表格空間中) 默認情況下為 MySQL 數據庫實例設置。

Note

部分現有的資料庫執行個體具有較低的限制。例如，在 2014 年 4 月之前建立的 MySQL 資料庫執行個體具有 2 TB 的檔案和資料表大小限制。無論何時建立資料庫執行個體，此 2-TB 檔案大小限制也適用於從 2014 年 4 月之前取得之快照建立的資料庫執行個體或僅供讀取複本。

使用 InnoDB file-per-table 表空間有優點和缺點，具體取決於您的應用程序。若要判斷應用程式的最佳方法，請參閱 MySQL 文件中的 [File-per-table 表格空間](#)。

不建議允許資料表成長至檔案大小上限。一般來說，更理想的做法是將資料表分割成更小的資料表，如此可以改善效能並縮短復原時間。

您可以用於將大型資料表分成更小資料表的選項為分割。「分割」會根據您指定的規則，將大型資料表的部分分散成個別檔案。例如，如果您依日期儲存交易，則可以建立分割規則，使用分割將較舊的交易分配成個別檔案。然後，您可以定期封存不需要立即可供應用程式使用的歷史交易資料。如需詳細資訊，請參閱 MySQL 文件中的 [分割](#)。

因為沒有可提供所有資料表和 InnoDB 系統資料表空間大小的單一系統資料表或檢視，因此您必須查詢多個資料表，以判斷資料表空間的大小。

決定 InnoDB 系統資料表空間和資料字典資料表空間的大小

- 使用下列 SQL 命令來判斷您的任何資料表空間是否太大，且為進行分割的候選者。

Note

資料字典資料表空間是 MySQL 8.0 特有的。

```
select FILE_NAME, TABLESPACE_NAME, ROUND((((TOTAL_EXTENTS*EXTENT_SIZE)
/1024/1024/1024), 2) as "File Size (GB)" from information_schema.FILES
where tablespace_name in ('mysql','innodb_system');
```

決定 InnoDB 系統資料表空間之外的 InnoDB 使用者資料表大小 (適用於 MySQL 5.7 版)

- 使用下列 SQL 命令來判斷您是否有太大的資料表，且為進行分割的候選者。

```
SELECT SPACE, NAME, ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

決定 InnoDB 系統資料表空間之外的 InnoDB 使用者資料表大小 (適用於 MySQL 8.0 版)

- 使用下列 SQL 命令來判斷您是否有太大的資料表，且為進行分割的候選者。

```
SELECT SPACE, NAME, ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_TABLESPACES ORDER BY 3 DESC;
```

決定非 InnoDB 使用者資料表的大小

- 使用下列 SQL 命令，判斷您的任何非 InnoDB 使用者資料表是否太大。

```
SELECT TABLE_SCHEMA, TABLE_NAME, round((((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```


若要啟用 InnoDB file-per-table 表格空間

- 在資料庫執行個體的參數群組中，將 `innodb_file_per_table` 參數設為 1。

若要停用 InnoDB file-per-table 表格空間

- 在資料庫執行個體的參數群組中，將 `innodb_file_per_table` 參數設為 0。

如需更新參數群組的相關資訊，請參閱[使用參數群組](#)。

當您啟用或停用 InnoDB 表 file-per-table 格空間時，您可以發出 ALTER TABLE 指令，將表格從全域表格空間移至自己的表格空間，或從其本身的表格空間移至全域表格空間，如下列範例所示：

```
ALTER TABLE table_name ENGINE=InnoDB;
```

不支援 MySQL Keyring 外掛程式

Amazon RDS for MySQL 目前不支援 MySQL `keyring_aws` Amazon Web Services Keyring 外掛程式。

自訂連接埠

Amazon RDS 會封鎖 MySQL 引擎的自訂連接埠 33060 的連線。為您的 MySQL 引擎選擇不同的連接埠。

MySQL 預存程序限制

如果 MySQL 使用者名稱長度超過 16 個字元，則在以下版本的 RDS for MySQL 上，[mysql.rds_kill](#) 和 [mysql.rds_kill_query](#) 預存程序無法終止該使用者擁有的工作階段或查詢：

- 8.0.32 和較低的 8 版本
- 5.7.41 和較低的 5.7 版本

使用外部來源執行個體進行 GTID 式複寫

根據全域交易識別符 (GTID)，Amazon RDS 不支援從外部 MySQL 執行個體複寫至需要在設定期設定 GTID_PURGED 的 Amazon RDS for MySQL 資料庫執行個體。

MySQL 默認身份驗證插件

適用於 MySQL 的 RDS 版本 8.0.34 及更高版本使用該插件 `mysql_native_password`。您無法變更 `default_authentication_plugin` 設定。

覆蓋緩衝池大小

對於微型或小型資料庫執行個體類別，`innodb_buffer_pool_size` 參數的預設值可能與執行下列命令所傳回的值不同：

```
mysql> SELECT @@innodb_buffer_pool_size;
```

當 Amazon RDS 需要在管理資料庫執行個體類別時覆寫預設值時，可能會發生這種差異。如有必要，您可以覆寫預設值，並將其設定為資料庫執行個體類別支援的值。若要判斷有效值，請新增資料庫執行個體上可用的記憶體使用量和總記憶體。如需詳細資訊，請參閱 [Amazon RDS 執行個體類型](#)。

如果您的資料庫執行個體只有 4 GB 的記憶體，則無法 `innodb_buffer_pool_size` 將其設定為 8 GB，但是可以將其設定為 3 GB，這取決於您為其他參數配置了多少記憶體。

如果您輸入的值太大，Amazon RDS 會將該值降低到以下限制：

- 微型資料庫執行個體類別：256 MB
- 微型資料庫執行個體類別：128 MB

RDS for MySQL 預存程序參考

這些主題描述執行 MySQL 資料庫引擎之 Amazon RDS 執行個體可用的系統預存程序。主要使用者必須執行這些程序。

主題

- [設定](#)
- [結束工作階段或查詢](#)
- [日誌](#)
- [管理主動-主動叢集](#)
- [管理多重來源複製](#)
- [管理全域狀態歷史記錄](#)
- [複寫](#)
- [預備 InnoDB 快取](#)

設定

下列預存程序會設定並顯示組態參數，例如二進位日誌檔案保留。

主題

- [mysql.rds_set_configuration](#)
- [mysql.rds_show_configuration](#)

mysql.rds_set_configuration

指定保留二進位日誌的小時數，或延遲複寫的秒數。

語法

```
CALL mysql.rds_set_configuration(name, value);
```

參數

name

要設定之組態參數的名稱。

#

組態參數的值。

使用須知

mysql.rds_set_configuration 程序支援下列組態參數：

- [binlog 保留 \(小時\)](#)
- [來源延遲](#)
- [目標延遲](#)

組態參數會永久存放，且在任何資料庫執行個體重新啟動或容錯移轉後依然存在。

binlog 保留 (小時)

`binlog retention hours` 參數用於指定保留二進位日誌檔的小時數。Amazon RDS 通常會儘快清除二進位日誌，但複寫 RDS 外部的 MySQL 資料庫時可能仍需要二進位日誌。

`binlog retention hours` 的預設值為 NULL。對於 RDS for MySQL，NULL 意味著不會保留二進位日誌 (0 小時)。

若要指定在資料庫執行個體上保留二進位日誌的時數，請使用 `mysql.rds_set_configuration` 預存程序，並指定讓複寫有足夠時間進行的期間，如下列範例所示。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Note

不可針對 `binlog retention hours` 使用值 0。

若為 MySQL 資料庫執行個體，`binlog retention hours` 最大值為 168 (7 天)。

設定保留期間之後，請監控資料庫執行個體的儲存體用量，確定保留的二進位日誌沒有佔用太多儲存體。

來源延遲

在僅供讀取複本中使用 `source delay` 參數，來指定延遲幾秒後僅供讀取複本才會複寫至其來源資料庫執行個體。Amazon RDS 通常會儘快複寫變更，但您可能想要某些環境延遲複寫。例如，當延遲複寫時，您可以將延遲的僅供讀取複本向前捲動到災難發生的時間點之前。若資料表被意外的捨棄，您可以使用延遲的複寫快速將其復原。`target delay` 的預設值為 0 (不延遲複寫)。

當您使用此參數時，它會執行 [mysql.rds_set_source_delay](#) 並套用 `CHANGE primary TO MASTER_DELAY = input value`。如果成功，程序會將 `source delay` 參數儲存到 `mysql.rds_configuration` 資料表。

若要指定 Amazon RDS 延遲複寫至來源資料庫執行個體的秒數，請使用 `mysql.rds_set_configuration` 預存程序，並指定延遲複寫的秒數。在下列範例中，複寫會延遲至少一小時 (3,600 秒)。

```
call mysql.rds_set_configuration('source delay', 3600);
```

此程序接著執行 `mysql.rds_set_source_delay(3600)`。

`source delay` 參數的限制為一日 (86400 秒)。

Note

RDS for MySQL 8.0 版或低於 10.2 的 MariaDB 版本不支援 `source delay` 參數。

目標延遲

使用 `target delay` 參數，指定要延遲幾秒後才會在資料庫執行個體與從這個執行個體建立的任何未來 RDS 受管僅供讀取複本之間進行複寫。非 RSS 受管僅供讀取複本會忽略此參數。Amazon RDS 通常會盡快複寫變更，但您可能想要某些環境延遲複寫。例如，當延遲複寫時，您可以將延遲的僅供讀取複本向前捲動到災難發生的時間點之前。若資料表被意外的捨棄，您可以使用延遲的複寫快速將其復原。`target delay` 的預設值為 0 (不延遲複寫)。

進行災難復原時，您可以使用此組態參數搭配 [mysql.rds_start_replication_until](#) 預存程序或 [mysql.rds_start_replication_until_gtid](#) 預存程序。若要將延遲的僅供讀取複本向前捲動到災難發生前的時間點，您可執行 `mysql.rds_set_configuration` 程序搭配此參數集。在 `mysql.rds_start_replication_until` 或 `mysql.rds_start_replication_until_gtid` 程序停止複寫後，您可使用 [提升僅供讀取複本為獨立的資料庫執行個體](#) 中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

若要使用 `mysql.rds_rds_start_replication_until_gtid` 程序，需啟用依據 GTID 的複寫。若要略過已知會導致災難的特定 GTID 型交易，可以使用 [mysql.rds_skip_transaction_with_gtid](#) 預存程序。如需有關依據 GTID 複寫的詳細資訊，請參閱 [使用 GTID 式複寫](#)。

若要指定 Amazon RDS 秒數以延遲至僅供讀取複本的複寫，請使用 `mysql.rds_set_configuration` 預存程序，並指定延遲複寫秒數。下列範例指定了已延遲至少一小時 (3600 秒) 的複寫。

```
call mysql.rds_set_configuration('target delay', 3600);
```

`target delay` 參數的限制為一日 (86400 秒)。

Note

RDS for MySQL 8.0 版或低於 10.2 的 MariaDB 版本不支援 `target delay` 參數。

mysql.rds_show_configuration

保留二進位日誌的時數。

語法

```
CALL mysql.rds_show_configuration;
```

使用須知

若要驗證 Amazon RDS 保留二進位日誌的時數，請使用 `mysql.rds_show_configuration` 預存程序。

範例

下列範例顯示保留期間：

```
call mysql.rds_show_configuration;
      name                               value  description
      binlog retention hours             24     binlog retention hours specifies
the duration in hours before binary logs are automatically deleted.
```

結束工作階段或查詢

下列預存程序會結束工作階段或查詢。

主題

- [mysql.rds_kill](#)
- [mysql.rds_kill_query](#)

mysql.rds_kill

結束 MySQL 伺服器的連線。

語法

```
CALL mysql.rds_kill(processID);
```

參數

processID

要結束之連線執行緒的身分。

使用須知

MySQL 伺服器的每個連線都在個別執行緒中執行。若要結束連線，請使用 `mysql.rds_kill` 程序並傳入該連線的執行緒 ID。若要取得執行緒 ID，請使用 MySQL [SHOW PROCESSLIST](#) 命令。

如需限制的詳細資訊，請參閱[MySQL 預存程序限制](#)。

範例

下列範例結束執行緒 ID 為 4243 的連線：

```
CALL mysql.rds_kill(4243);
```

mysql.rds_kill_query

結束對 MySQL 伺服器執行的查詢。

語法

```
CALL mysql.rds_kill_query(processID);
```

參數

processID

正在執行要結束之查詢的處理序或執行緒的身分。

使用須知

若要停止對 MySQL 伺服器執行的查詢，請使用 `mysql_rds_kill_query` 程序並傳入執行查詢之執行緒的連線 ID。然後程序就會終止連線。

若要取得 ID，請查詢 MySQL [INFORMATION_SCHEMA.PROCESSLIST](#) 資料表或使用 MySQL [SHOW PROCESSLIST](#) 命令。SHOW PROCESSLIST 或 SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST 中 ID 欄的值为 *processID*。

如需限制的詳細資訊，請參閱[MySQL 預存程序限制](#)。

範例

下列範例會停止查詢執行緒 ID 為 230040 的查詢：

```
CALL mysql.rds_kill_query(230040);
```

日誌

下列預存程序會將 MySQL 日誌輪換為備份資料表。如需更多詳細資訊，請參閱 [MySQL 資料庫日誌檔案](#)。

主題

- [mysql.rds_rotate_general_log](#)
- [mysql.rds_rotate_slow_log](#)

mysql.rds_rotate_general_log

將 `mysql.general_log` 資料表輪換至備份資料表。

語法

```
CALL mysql.rds_rotate_general_log;
```

使用須知

您可以呼叫 `mysql.general_log` 程序，將 `mysql.rds_rotate_general_log` 資料表輪換至備份資料表。輪換日誌資料表時，目前日誌資料表會複製到備份日誌資料表，並移除目前日誌資料表中的項目。如果備份日誌資料表已存在，則其會在目前日誌資料表複製到備份之前遭到刪除。如有需要，您可以查詢備份日誌資料表。`mysql.general_log` 資料表的備份日誌資料表名為 `mysql.general_log_backup`。

當 `log_output` 參數設定為 `TABLE` 時，您僅可執行此程序。

mysql.rds_rotate_slow_log

將 `mysql.slow_log` 資料表輪換至備份資料表。

語法

```
CALL mysql.rds_rotate_slow_log;
```

使用須知

您可以呼叫 `mysql.slow_log` 程序，將 `mysql.rds_rotate_slow_log` 資料表輪換至備份資料表。輪換日誌資料表時，目前日誌資料表會複製到備份日誌資料表，並移除目前日誌資料表中的項目。如果備份日誌資料表已存在，則其會在目前日誌資料表複製到備份之前遭到刪除。

如有需要，您可以查詢備份日誌資料表。`mysql.slow_log` 資料表的備份日誌資料表名為 `mysql.slow_log_backup`。

管理主動-主動叢集

下列預存程序會設定和管理適用於 MySQL 的主動-主動式叢集的 RDS。如需詳細資訊，請參閱 [the section called “設定主動-主動叢集”](#)。

這些預存程序僅適用於執行 8.0.35 版及更高版本次要版本的 RDS for MySQL 資料庫執行個體。

主題

- [我們的域名群組複製 _ 進階](#)
- [mysql.rds_ 群組複製 _ 建立使用者](#)
- [神秘的 rds_ 群組複製 _ 設定恢復通道](#)
- [我們的群組複製啟動](#)
- [神秘的 rds_ 群組複製 _ 停止](#)

我們的域名群組複製 _ 進階

在目前的資料庫執行個體上建立預留位置 GTID。

語法

```
CALL mysql.rds_group_replication_advance_gtid(  
  begin_id  
  , end_id  
  , server_uuid  
);
```

參數

##

要建立的開始交易 ID。

#####

要建立的最終交易 ID。

##

要group_replication_group_name建立之交易的。

在group_replication_group_name與資料庫執行個體相關聯的資料庫參數群組中，指定為 UUID。

使用須知

在主動-主動叢集中，若要讓資料庫執行個體加入群組，在新資料庫執行個體上執行的所有 GTID 交易都必須存在於叢集中的其他成員上。在異常情況下，在將執行個體加入到群組之前執行交易時，新的資料庫執行個體可能會有更多的交易。在此情況下，您無法移除任何現有的交易，但可以使用此程序在群組中的資料庫執行個體上建立對應的預留位置 GTID。在執行此操作之前，請確認交易不會影響複寫的資料。

當您呼叫此程序時，的 GTID 交易 `server_uuid:begin_id-end_id` 會以空白內容建立。若要避免複寫問題，請勿在任何其他情況下使用此程序。

Important

請避免在主動-主動叢集正常運作時呼叫此程序。除非您了解正在創建的交易可能產生的後果，否則不要調用此過程。呼叫此程序可能會導致資料不一致。

範例

下列範例會在目前的資料庫執行個體上建立預留位置 GTID。：

```
CALL mysql.rds_group_replication_advance_gtid(5, 6,  
'11111111-2222-3333-4444-555555555555');
```

mysql.rds_ 群組複製 _ 建立使用者

為資料庫執行個體上 `rdsgrepladmin` 的群組複寫建立複寫使用者。

語法

```
CALL mysql.rds_group_replication_create_user(  
replication_user_password  
);
```

參數

replication_user_password

複製使用者的密碼 `rdsgrepladmin`。

使用須知

- 主動-主動叢集中所有資料庫執行個體的複寫使用者密碼`rdsgrepladmin`必須相同。
- `rdsgrepladmin`使用者名稱會保留給群組複寫連線使用。其他使用者 (包括主要使用者) 都無法擁有此使用者名稱。

範例

下列範例會為資料庫執行個體上`rdsgrepladmin`的群組複寫建立複寫使用者：

```
CALL mysql.rds_group_replication_create_user('password');
```

神秘的 `rds_ 群組複製 _ 設定恢復通道`

設定主動-主動叢集的`group_replication_recovery`通道。此程序會使用保留的使用`rdsgrepladmin`者來設定頻道。

語法

```
CALL mysql.rds_group_replication_set_recovery_channel(  
replication_user_password);
```

參數

replication_user_password

複製使用者的密碼`rdsgrepladmin`。

使用須知

主動-主動叢集中所有資料庫執行個體的複寫使用者密碼`rdsgrepladmin`必須相同。呼叫`mysql.rds_group_replication_create_user`指定密碼。

範例

下列範例會設定主動-主動叢集的`group_replication_recovery`通道：

```
CALL mysql.rds_group_replication_set_recovery_channel('password');
```

我們的群組複製啟動

在目前的資料庫執行個體上啟動群組複寫。

語法

```
CALL mysql.rds_group_replication_start(  
bootstrap  
);
```

參數

##

指定要初始化新群組還是加入現有群組的值。 1使用目前的資料庫執行個體初始化新群組。 0透過連接至與資料庫執行個體相關聯之資料庫參數群組中group_replication_group_seeds參數中定義的端點，將目前的資料庫執行個體加入至現有群組。

範例

下列範例會使用目前的資料庫執行個體初始化新群組：

```
CALL mysql.rds_group_replication_start(1);
```

神秘的 rds_ 群組複製 _ 停止

停止目前資料庫執行個體上的群組複寫。

語法

```
CALL mysql.rds_group_replication_stop();
```

使用須知

當您停止資料庫執行個體上的複寫時，不會影響主動-主動叢集中的任何其他資料庫執行個體。

管理多重來源複製

下列預存程序會設定和管理 RDS for MySQL 多來源複本上的複寫通道。如需詳細資訊，請參閱 [the section called “設定多重來源複製”](#)。

這些預存程序僅適用於執行下列引擎版本的 RDS for MySQL 資料庫執行個體：

- 8.0.35 及更高版本的次要版本
- 5.7.44 及更高版本的次要版本

Note

雖然本文件將來源資料庫執行個體稱為適用於 MySQL 資料庫執行個體的 RDS，但這些程序也適用於 Amazon RDS 外部執行的 MySQL 執行個體。

主題

- [下一個源代碼日誌通道](#)
- [神秘的 rds_外部來源_通道](#)
- [神秘的 rds_set_外部源代碼通道](#)
- [使用自動位置_通道的 mysql.rds_set_外部來源_來源](#)
- [神秘的 rds_set_外部_源_使用_延遲層_for_通道](#)
- [神秘的 rds_集合來源_自動位置_通道](#)
- [神秘的 rds_集合來源_延遲_通道](#)
- [為通道提供錯誤的解決方案](#)
- [我們的域名開始複製通道](#)
- [mysql.rds_開始複製_取消通道](#)
- [我們的域名開始複製_取消通道](#)
- [我們的域名_停止複製_通道](#)

下一個源代碼日誌通道

將來源資料庫執行個體記錄位置變更為通道之來源資料庫執行個體上下一個二進位記錄檔的開頭。只有在多來源複本上收到複寫 I/O 錯誤 1236 時，才使用此程序。

語法

```
CALL mysql.rds_next_source_log_for_channel(  
curr_master_log,  
channel_name  
);
```

參數

curr_master_log

目前來源日誌檔案的索引。例如，若目前檔案的名稱是 `mysql-bin-changelog.012345`，則索引為 12345。若要查明目前來源日誌檔案名稱，請執行 `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'` 命令並檢視 `Source_Log_File` 欄位。

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_next_source_log_for_channel` 程序。例如，如果發生 `IO_Thread` 錯誤，您可以使用此程序略過目前二進位記錄檔中的所有事件，並從中指定的通道的下一個二進位記錄檔繼續複寫。`channel_name`

範例

假設多來源複本上的通道上的複寫失敗。在多重來源複本 `SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G` 上執行會傳回下列結果：

```
mysql> SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G  
***** 1. row *****
```

```

Replica_IO_State: Waiting for source to send event
  Source_Host: myhost.XXXXXXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
  Source_User: ReplicationUser
  Source_Port: 3306
  Connect_Retry: 60
  Source_Log_File: mysql-bin-changelog.012345
Read_Source_Log_Pos: 1219393
  Relay_Log_File: replica-relay-bin.000003
  Relay_Log_Pos: 30223388
Relay_Source_Log_File: mysql-bin-changelog.012345
  Replica_IO_Running: No
  Replica_SQL_Running: Yes
  Replicate_Do_DB:.
.
.
  Last_IO_Errno: 1236
  Last_IO_Error: Got fatal error 1236 from master when reading data from
binary log: 'Client requested master to start replication from impossible position;
the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
'/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
  Last_SQL_Errno: 0
  Last_SQL_Error:
.
.
  Channel_name: channel_1
.
.
-- Some fields are omitted in this example output

```

Last_IO_Errno 欄位顯示執行個體收到輸入/輸出錯誤 1236。Source_Log_File 欄位顯示檔案名稱是 mysql-bin-changelog.012345，這表示日誌檔案索引為 12345。要解決錯誤，您可以使用 `mysql.rds_next_source_log_for_channel` 使用以下參數調用：

```
CALL mysql.rds_next_source_log_for_channel(12345, 'channel_1');
```

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

神秘的 rds_ 外部來源 _ 通道

停止指定通道上的複製程序，並從多來源複本中移除通道和相關組態。

Important

若要執行此程序，必須啟用 `autocommit`。若要啟用它，請將 `autocommit` 參數設定為 1。
如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_reset_external_source_for_channel (channel_name);
```

參數

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_reset_external_source_for_channel` 程序。此程序會刪除屬於要移除之通道的所有轉送記錄。

神秘的 rds_set_ 外部源代碼通道

在適用於 MySQL 的 RDS 資料庫執行個體上設定複寫通道，以便從另一個適用於 MySQL 的 RDS 資料庫執行個體複製資料。

Important

若要執行此程序，必須啟用 `autocommit`。若要啟用它，請將 `autocommit` 參數設定為 1。
如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

Note

您可以改用 [the section called “神秘的 rds_set_外部_源_使用_延遲層_for_通道”](#) 預存程序來設定延遲複寫的這個通道。

語法

```
CALL mysql.rds_set_external_source_for_channel (  
  host_name  
  , host_port  
  , replication_user_name  
  , replication_user_password  
  , mysql_binary_log_file_name  
  , mysql_binary_log_file_location  
  , ssl_encryption  
  , channel_name  
);
```

參數*host_name*

適用於 MySQL 的 RDS 來源資料庫執行個體的主機名稱或 IP 位址。

host_port

適用於 MySQL 的來源資料庫執行個體的 RDS 所使用的連接埠。如果網路組態包含會轉換連線埠號碼的安全殼層 (SSH) 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在適用於 MySQL 的 RDS 來源資料庫執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 權限的使用者識別碼。我們建議您提供僅用於複寫來源資料庫執行個體的帳戶。

replication_user_password

replication_user_name 中指定之使用者 ID 的密碼。

mysql_binary_log_file_name

來源資料庫執行個體上包含複寫資訊的二進位記錄名稱。

mysql_binary_log_file_location

複寫在 `mysql_binary_log_file_name` 二進位日誌中開始讀取複寫資訊的位置。

您可以透過在來源資料庫執行個體 `SHOW MASTER STATUS` 上執行來判斷 binlog 檔案名稱和位置。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 `MASTER_SSL_VERIFY_SERVER_CERT` 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

####

複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_set_external_source_for_channel` 程序。此程序必須在您要建立複寫通道的目標 RDS for MySQL 資料庫執行個體上執行。

執行之前 `mysql.rds_set_external_source_for_channel`，請在來源資料庫執行個體上設定具有多來源複本所需權限的複寫使用者。若要將多來源複本連線到來源資料庫執行個體，您必須指定 `replication_user_name` 具有來源資料庫執行個體 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限的複寫使用者 `replication_user_password` 值。

在來源資料庫執行個體上設定複寫使用者

1. 使用您選擇的 MySQL 用戶端連線到來源資料庫執行個體，並建立用於複寫的使用者帳戶。以下是範例。

Important

為了安全性最佳作法，請指定下列範例中顯示的預留位置值以外的密碼。

MySQL 8.0

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY  
'password';
```

MySQL 5.7

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. 在來源資料庫執行個體上，將複寫使用者授REPLICATION CLIENT與REPLICATION SLAVE權限。下列範例將所有資料庫上的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的 'repl_user' 使用者。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

若要使用加密複寫，請將來源資料庫執行個體設定為使用 SSL 連線。

[我們的域名開始複製通道](#)在呼叫設mysql.rds_set_external_source_for_channel定此複寫通道之後，您可以呼叫複本以在通道上啟動複寫程序。您可[the section called “神秘的 rds_外部來源_通道”](#)以呼叫停止通道上的複寫，並從複本中移除通道組態。

撥打電話時mysql.rds_set_external_source_for_channel，Amazon RDS 會set channel source在mysql.rds_history表格中記錄的時間、使用者和動作，而不含通道特定的詳細資訊，並在mysql.rds_replication_status表格中記錄頻道名稱。此資訊僅供內部使用和監控用途進行記錄。若要記錄完整的程序呼叫以進行稽核，請考慮根據應用程式的特定需求啟用稽核記錄或一般記錄。

範例

在 RDS for MySQL 資料庫執行個體上執行時，下列範例會設定channel_1在此資料庫執行個體上命名的複寫通道，從主機sourcedb.example.com和連接埠指定的來源複寫資料。3306

```
call mysql.rds_set_external_source_for_channel(  
  'sourcedb.example.com',  
  3306,  
  'repl_user',  
  'password',  
  'mysql-bin-changelog.0777',  
  120,
```

```
0,  
'channel_1');
```

使用自動位置 _ 通道的 mysql.rds_set_外部來源_來源

在 RDS for MySQL 資料庫執行個體上設定複寫通道，並具有選擇性的複寫延遲。複寫是以全域交易識別碼 (GTID) 為基礎。

Important

若要執行此程序，必須啟用 `autocommit`。若要啟用它，請將 `autocommit` 參數設定為 1。如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_set_external_source_with_auto_position_for_channel (  
  host_name  
  , host_port  
  , replication_user_name  
  , replication_user_password  
  , ssl_encryption  
  , delay  
  , channel_name  
);
```

參數

host_name

適用於 MySQL 的 RDS 來源資料庫執行個體的主機名稱或 IP 位址。

host_port

適用於 MySQL 的來源資料庫執行個體的 RDS 所使用的連接埠。如果網路組態包含會轉換連線埠號碼的安全殼層 (SSH) 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在適用於 MySQL 的 RDS 來源資料庫執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 權限的使用者識別碼。我們建議您提供僅用於複寫來源資料庫執行個體的帳戶。

replication_user_password

`replication_user_name` 中指定之使用者 ID 的密碼。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 `MASTER_SSL_VERIFY_SERVER_CERT` 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

##

延遲來源資料庫執行個體複寫的最小秒數。

此參數的限制為一日 (86400 秒)。

####

複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行

`mysql.rds_set_external_source_with_auto_position_for_channel` 程序。此程序必須在您要建立複寫通道的目標 RDS for MySQL 資料庫執行個體上執行。

執行之前 `rds_set_external_source_with_auto_position_for_channel`，請在來源資料庫執行個體上設定具有多來源複本所需權限的複寫使用者。若要將多來源複本連線到來源資料庫執行個體，您必須指定 `replication_user_name` 具有來源資料庫執行個體 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限的複寫使用者 `replication_user_password` 值。

在來源資料庫執行個體上設定複寫使用者

1. 使用您選擇的 MySQL 用戶端連線到來源資料庫執行個體，並建立用於複寫的使用者帳戶。以下是範例。

⚠ Important

為了安全性最佳作法，請指定下列範例中顯示的預留位置值以外的密碼。

MySQL 8.0

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY  
'password';
```

MySQL 5.7

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. 在來源資料庫執行個體上，將複寫使用者授 REPLICATION CLIENT 與 REPLICATION SLAVE 權限。下列範例將所有資料庫上的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的 'repl_user' 使用者。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

若要使用加密複寫，請將來源資料庫執行個體設定為使用 SSL 連線。

在呼叫 `mysql.rds_set_external_source_with_auto_position_for_channel` 將 Amazon RDS 資料庫執行個體設定為特定通道上的僅供讀取複本後，您可以呼叫 [the section called “我們的域名開始複製通道”](#) 僅供讀取複本以在該通道上啟動複寫程序。

[我們的域名開始複製通道](#) 在呼叫

設 `mysql.rds_set_external_source_with_auto_position_for_channel` 定此複寫通道之後，您可以呼叫複本以在通道上啟動複寫程序。您可 [the section called “神秘的 rds_外部來源_通道”](#) 以呼叫停止通道上的複寫，並從複本中移除通道組態。

範例

在 RDS for MySQL 資料庫執行個體上執行時，下列範例會將此資料庫執行個體 `channel_1` 上命名的複寫通道設定為從主機 `sourcedb.example.com` 和連接埠指定的來源複寫資料。3306 它會將最短複寫延遲設定為一小時 (3,600 秒)。這表示來源 RDS for MySQL 資料庫執行個體的變更至少不會套用至少一個小時至少在多來源複本上。

```
call mysql.rds_set_external_source_with_auto_position_for_channel(  
  'sourcedb.example.com',  
  3306,  
  'repl_user',  
  'password',  
  0,  
  3600,  
  'channel_1');
```

神秘的 rds_set_外部_源_使用_延遲層_for_通道

在 RDS for MySQL 資料庫執行個體上設定具有指定複製延遲的複製通道。

Important

若要執行此程序，必須啟用 autocommit。若要啟用它，請將 autocommit 參數設定為 1。如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_set_external_source_with_delay_for_channel (  
  host_name  
  , host_port  
  , replication_user_name  
  , replication_user_password  
  , mysql_binary_log_file_name  
  , mysql_binary_log_file_location  
  , ssl_encryption  
  , delay  
  , channel_name  
);
```

參數

host_name

適用於 MySQL 的 RDS 來源資料庫執行個體的主機名稱或 IP 位址。

host_port

適用於 MySQL 的來源資料庫執行個體的 RDS 所使用的連接埠。如果網路組態包含會轉換連線埠號碼的安全殼層 (SSH) 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在適用於 MySQL 的 RDS 來源資料庫執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 權限的使用者識別碼。我們建議您提供僅用於複寫來源資料庫執行個體的帳戶。

replication_user_password

replication_user_name 中指定之使用者 ID 的密碼。

mysql_binary_log_file_name

來源資料庫執行個體上的二進位記錄名稱包含複寫資訊。

mysql_binary_log_file_location

在 mysql_binary_log_file_name 二進位日誌中的複寫將開始讀取複寫資訊的位置。

您可以藉由在來源資料庫執行個體上執行 SHOW MASTER STATUS 來判斷 binlog 檔案名稱和位置。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 MASTER_SSL_VERIFY_SERVER_CERT 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

##

延遲來源資料庫執行個體複寫的最小秒數。

此參數的限制為一日 (86400 秒)。

####

複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_set_external_source_with_delay_for_channel` 程序。此程序必須在您要建立複寫通道的目標 RDS for MySQL 資料庫執行個體上執行。

執行之前 `mysql.rds_set_external_source_with_delay_for_channel`，請在來源資料庫執行個體上設定具有多來源複本所需權限的複寫使用者。若要將多來源複本連線到來源資料庫執行個體，您必須指定 `replication_user_name` 具有來源資料庫執行個體 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限的複寫使用者 `replication_user_password` 值。

在來源資料庫執行個體上設定複寫使用者

1. 使用您選擇的 MySQL 用戶端連線到來源資料庫執行個體，並建立用於複寫的使用者帳戶。以下是範例。

Important

為了安全性最佳作法，請指定下列範例中顯示的預留位置值以外的密碼。

MySQL 8.0

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY  
'password';
```

MySQL 5.7

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. 在來源資料庫執行個體上，將複寫使用者授 `REPLICATION CLIENT` 與 `REPLICATION SLAVE` 權限。下列範例將所有資料庫上的 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限授予您網域中的 'repl_user' 使用者。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

若要使用加密複寫，請將來源資料庫執行個體設定為使用 SSL 連線。

[我們的域名開始複製通道在呼叫](#)

設 `mysql.rds_set_external_source_with_delay_for_channel` 定此複寫通道之後，您可以

呼叫複本以在通道上啟動複寫程序。您可[the section called “神秘的 rds_外部來源_通道”](#)以呼叫停止通道上的複寫，並從複本中移除通道組態。

撥打電話時mysql.rds_set_external_source_with_delay_for_channel，Amazon RDS 會set channel source在mysql.rds_history表格中記錄的時間、使用者和動作，而不含通道特定的詳細資訊，並在mysql.rds_replication_status表格中記錄頻道名稱。此資訊僅供內部使用和監控用途進行記錄。若要記錄完整的程序呼叫以進行稽核，請考慮根據應用程式的特定需求啟用稽核記錄或一般記錄。

範例

在 RDS for MySQL 資料庫執行個體上執行時，下列範例會將此資料庫執行個體channel_1上命名的複寫通道設定為從主機sourcedb.example.com和連接埠指定的來源複寫資料。3306它會將最短複寫延遲設定為一小時 (3,600 秒)。這表示來源 RDS for MySQL 資料庫執行個體的變更至少不會套用至少一個小時至少在多來源複本上。

```
call mysql.rds_set_external_source_with_delay_for_channel(  
  'sourcedb.example.com',  
  3306,  
  'repl_user',  
  'password',  
  'mysql-bin-changelog.000777',  
  120,  
  0,  
  3600,  
  'channel_1');
```

神秘的 rds_集合來源_自動位置_通道

根據二進位記錄檔位置或全域交易識別碼 (GTID)，將指定通道的複寫模式設定為基礎。

語法

```
CALL mysql.rds_set_source_auto_position_for_channel (  
  auto_position_mode  
  , channel_name  
);
```

參數

auto_position_mode

此值指示要使用日誌檔案位置複寫或是 GTID 複寫：

- 0 – 使用依據二進制日誌檔案位置的複寫模式。預設值為 0。
- 1 – 使用依據 GTID 的複寫方法。

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_set_source_auto_position_for_channel` 程序。此程序會重新啟動指定通道上的複寫，以套用指定的 `auto` 定位模式。

範例

下列範例會將 `channel_1` 的 `auto` 定位模式設定為使用以 GTID 為基礎的複製方法。

```
call mysql.rds_set_source_auto_position_for_channel(1,'channel_1');
```

神秘的 `rds_集合來源_延遲_通道`

設定延遲從來源資料庫執行處理複製到指定通道的多來源複本的最小秒數。

語法

```
CALL mysql.rds_set_source_delay_for_channel(delay, channel_name);
```

參數

##

延遲來源資料庫執行個體複寫的最小秒數。

此參數的限制為一日 (86400 秒)。

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_set_source_delay_for_channel` 程序。若要使用此程序，請先呼叫 `mysql.rds_stop_replication_for_channel` 停止複寫。然後，呼叫此程序來設定複寫延遲值。設定延遲時，`mysql.rds_start_replication_for_channel` 請呼叫以重新啟動複寫。

範例

下列範例會將多來源複本上 `channel_1` 的來源資料庫執行處理複寫延遲設定至少一小時 (3,600 秒)。

```
CALL mysql.rds_set_source_delay_for_channel(3600, 'channel_1');
```

為通道提供錯誤的解決方案

略過二進位記錄事件，並刪除指定通道的 MySQL DB 多來源複本上的複寫錯誤。

語法

```
CALL mysql.rds_skip_repl_error_for_channel(channel_name);
```

參數

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須在僅供讀取複本上執行 `mysql.rds_skip_repl_error_for_channel` 程序。您可以用類似的方式來使用 `mysql.rds_skip_repl_error` 此程序來略過僅供讀取複本上的錯誤。如需詳細資訊，請參閱 [呼叫 `mysql.rds_skip_repl_error` 程序](#)。

Note

若要略過 GTID 型複寫中的錯誤，建議您改用此程序。[the section called “mysql.rds_skip_transaction_with_gtid”](#)

若要判斷是否有錯誤，執行 MySQL SHOW REPLICA STATUS FOR CHANNEL '*channel_name*'\G 命令。如果複寫錯誤不嚴重，您可以執行 `mysql.rds_skip_repl_error_for_channel` 來略過錯誤。如果有多個錯誤，請 `mysql.rds_skip_repl_error_for_channel` 刪除指定複製通道上的第一個錯誤，然後警告其他錯誤存在。然後，您可以使用 SHOW REPLICA STATUS FOR CHANNEL '*channel_name*'\G，以針對下一個錯誤判斷正確的行動步驟。如需傳回值的相關資訊，請參閱 MySQL 文件中的 [SHOW REPLICA STATUS 陳述式](#)。

我們的域名開始複製通道

從 RDS for MySQL 資料庫執行個體起始複寫到指定通道上的多來源複本。

Note

您可使用 [mysql.rds_開始複製_取消通道](#) 或 [我們的域名開始複製_取消通道](#) 預存程序從 RDS for MySQL 資料庫執行個體來啟動複寫，並從特定的二進位日誌檔案位置停止複寫。

語法

```
CALL mysql.rds_start_replication_for_channel(channel_name);
```

參數

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 `mysql.rds_start_replication_for_channel` 程序。從來源 RDS for MySQL 資料庫執行個體匯入資料後，請在多來源複本上執行此命令，以在指定的通道上開始複寫。

範例

下列範例會在多來源複channel_1本上啟動複寫。

```
CALL mysql.rds_start_replication_for_channel('channel_1');
```

mysql.rds_ 開始複製 _ 取消通道

從指定通道上的 RDS for MySQL 資料庫執行個體起始複寫，並在指定的二進位記錄檔位置停止複寫。

語法

```
CALL mysql.rds_start_replication_until_for_channel (  
  replication_log_file  
  , replication_stop_point  
  , channel_name  
);
```

參數

replication_log_file

來源資料庫執行個體上的二進位記錄名稱包含複寫資訊。

replication_stop_point

在 replication_log_file 二進位日誌中的複寫將停止的位置。

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 mysql.rds_start_replication_until_for_channel 程序。透過此程序，複寫會在到達指定的 binlog 檔案位置時啟動，然後停止。對於 8.0 版，程序只會停止 SQL_Thread。對於版本 5.7，程序會同時停止 SQL_Thread 和 IO_Thread。

為 replication_log_file 參數指定的檔案名稱必須與來源資料庫執行個體 binlog 檔案名稱相符。

當 replication_stop_point 參數指定過去的停止位置時，會立即停止複寫。

範例

下列範例會在上啟動複寫channel_1，並複寫變更，直到到達mysql-bin-changelog.000777二進位記錄檔120中的位置為止。

```
call mysql.rds_start_replication_until_for_channel(  
    'mysql-bin-changelog.000777',  
    120,  
    'channel_1'  
);
```

我們的域名開始複製 _ 取消通道

從 RDS for MySQL 資料庫執行個體在指定的通道上啟動複寫，並在指定的全域交易識別碼 (GTID) 停止複寫。

語法

```
CALL mysql.rds_start_replication_until_gtid_for_channel(gtid, channel_name);
```

參數

gtid

之後要停止複寫的 GTID。

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 mysql.rds_start_replication_until_gtid_for_channel 程序。程序會在指定的通道上開始複寫，並將所有變更套用到指定的 GTID 值。然後，它會停止通道上的複製。

當 gtid 參數指定了複本已經執行的交易時，複寫會立即停止。

在執行此程序之前，您必須slave_parallel_workers將replica_parallel_workers0或的值設定為來停用多執行緒複製。

範例

下列範例會在上啟動複寫channel_1，並複寫變更，直到到達 GTID 為止。3E11FA47-71CA-11E1-9E33-C80AA9429562:23

```
call mysql.rds_start_replication_until_gtid_for_channel('3E11FA47-71CA-11E1-9E33-C80AA9429562:23','channel_1');
```

我們的域名 _ 停止複製 _ 通道

停止從指定通道上的 MySQL 資料庫執行個體複寫。

語法

```
CALL mysql.rds_stop_replication_for_channel(channel_name);
```

參數

####

多來源複本上複製通道的名稱。每個複寫通道都會從特定主機和連接埠上執行的單一來源 RDS for MySQL 資料庫執行個體接收二進位記錄事件。

使用須知

主要使用者必須執行 mysql.rds_stop_replication_for_channel 程序。

範例

下列範例會停止多來源複本channel_1的複寫。

```
CALL mysql.rds_stop_replication_for_channel('channel_1');
```

管理全域狀態歷史記錄

Amazon RDS 會提供一組程序，將在一段時間內快照狀態變數的值，並將它們以及自從上次快照後的任何變更寫入至資料表。此基礎設施稱為「全域狀態歷史記錄」。如需更多詳細資訊，請參閱[管理全域狀態歷史記錄](#)。

下列預存程序會管理收集及維護「全域狀態歷史記錄」的方式。

主題

- [mysql.rds_collect_global_status_history](#)
- [mysql.rds_disable_gsh_collector](#)
- [mysql.rds_disable_gsh_rotation](#)
- [mysql.rds_enable_gsh_collector](#)
- [mysql.rds_enable_gsh_rotation](#)
- [mysql.rds_rotate_global_status_history](#)
- [mysql.rds_set_gsh_collector](#)
- [mysql.rds_set_gsh_rotation](#)

mysql.rds_collect_global_status_history

隨需建立全域狀態歷史記錄的快照。

語法

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_disable_gsh_collector

停用全域狀態歷史記錄建立的快照。

語法

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_disable_gsh_rotation

關閉 `mysql.global_status_history` 表格的輪換。

語法

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_enable_gsh_collector

啟用全域狀態歷史記錄來依據 `rds_set_gsh_collector` 指定的間隔建立預設快照。

語法

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_enable_gsh_rotation

啟用依 `mysql.global_status_history` 指定的間隔將 `mysql.global_status_history_old` 資料表的內容輪換至 `rds_set_gsh_rotation`。

語法

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

隨需將 `mysql.global_status_history` 資料表的內容輪換至 `mysql.global_status_history_old`。

語法

```
CALL mysql.rds_rotate_global_status_history;
```

mysql.rds_set_gsh_collector

指定全域狀態歷史記錄建立快照之間的時間隔 (以分鐘為單位)。

語法

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

參數

intervalPeriod

快照之間的時間隔 (以分鐘為單位)。預設值為 5。

mysql.rds_set_gsh_rotation

指定 `mysql.global_status_history` 資料表輪換之間的時間隔 (以天為單位)。

語法

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

參數

intervalPeriod

資料表輪換之間的時間隔 (以天為單位)。預設值為 7。

複寫

下列預存程序控制了外部資料庫的交易複製到 RDS for MySQL 中，或從 RDS for MySQL 複製到外部資料庫的方式。如要瞭解透過 RDS for MySQL 使用以全域交易識別碼 (GTID) 為基礎的複本使用方式，請參閱 [使用 GTID 式複寫](#)。

主題

- [mysql.rds_next_master_log](#)
- [mysql.rds_reset_external_master](#)
- [mysql.rds_set_external_master](#)
- [mysql.rds_set_external_master_with_auto_position](#)
- [mysql.rds_set_external_master_with_delay](#)
- [mysql.rds_set_master_auto_position](#)
- [mysql.rds_set_source_delay](#)
- [mysql.rds_skip_transaction_with_gtid](#)
- [mysql.rds_skip_repl_error](#)
- [mysql.rds_start_replication](#)
- [mysql.rds_start_replication_until](#)
- [mysql.rds_start_replication_until_gtid](#)
- [mysql.rds_stop_replication](#)

mysql.rds_next_master_log

將來源資料庫執行個體日誌位置變更為來源資料庫執行個體上下一個二進位日誌的開頭。只有當您在僅供讀取複本上收到複寫輸入/輸出錯誤 1236 時，才使用此程序。

語法

```
CALL mysql.rds_next_master_log(  
curr_master_log  
);
```

參數

curr_master_log

目前主控端日誌檔案的索引。例如，若目前檔案的名稱是 `mysql-bin-changelog.012345`，則索引為 12345。若要查明目前主控端日誌檔案名稱，請執行 `SHOW REPLICA STATUS` 命令並檢視 `Master_Log_File` 欄位。

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

使用須知

主要使用者必須執行 `mysql.rds_next_master_log` 程序。

Warning

只有當複寫來源的異地同步備份資料庫執行個體在容錯移轉之後複寫失敗時，且 `mysql.rds_next_master_log` 的 `Last_IO_Errno` 欄位報告輸入/輸出錯誤 1236，才呼叫 `SHOW REPLICA STATUS`。

在容錯移轉事件發生之前，如果來源執行個體中的交易未寫入磁碟上的二進位日誌，則呼叫 `mysql.rds_next_master_log` 會導致僅供讀取複本遺失資料。

您可以將來源執行個體參數 `sync_binlog` 和 `innodb_support_xa` 設為 1，以降低此情況發生的機率，雖然這麼做會降低效能。如需詳細資訊，請參閱 [對 MySQL 僅供讀取複本問題進行故障診斷](#)。

範例

假設 RDS for MySQL 僅供讀取複本上的複寫失敗。在僅供讀取複本上執行 `SHOW REPLICA STATUS \G` 將傳回下列結果：

```
***** 1. row *****
      Replica_IO_State:
        Source_Host: myhost.XXXXXXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
        Source_User: MasterUser
        Source_Port: 3306
```



```
    Connect_Retry: 10
    Source_Log_File: mysql-bin-changelog.012345
  Read_Source_Log_Pos: 1219393
    Relay_Log_File: relaylog.012340
    Relay_Log_Pos: 30223388
  Relay_Source_Log_File: mysql-bin-changelog.012345
    Replica_IO_Running: No
    Replica_SQL_Running: Yes
    Replicate_Do_DB:
    Replicate_Ignore_DB:
    Replicate_Do_Table:
    Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
    Exec_Source_Log_Pos: 30223232
    Relay_Log_Space: 5248928866
    Until_Condition: None
    Until_Log_File:
    Until_Log_Pos: 0
    Source_SSL_Allowed: No
    Source_SSL_CA_File:
    Source_SSL_CA_Path:
    Source_SSL_Cert:
    Source_SSL_Cipher:
    Source_SSL_Key:
  Seconds_Behind_Master: NULL
Source_SSL_Verify_Server_Cert: No
    Last_IO_Errno: 1236
    Last_IO_Error: Got fatal error 1236 from master when reading data from
binary log: 'Client requested master to start replication from impossible position;
the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
'/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
    Last_SQL_Errno: 0
    Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
    Source_Server_Id: 67285976
```

Last_IO_Errno 欄位顯示執行個體收到輸入/輸出錯誤 1236。Master_Log_File 欄位顯示檔案名稱是 mysql-bin-changelog.012345，這表示日誌檔案索引為 12345。若要解決錯誤，您可以呼叫 mysql.rds_next_master_log 並指定下列參數：

```
CALL mysql.rds_next_master_log(12345);
```

Note

MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

mysql.rds_reset_external_master

將 RDS for MySQL 資料庫執行個體重新設定為不再是 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。

Important

若要執行此程序，必須啟用 autocommit。若要啟用它，請將 autocommit 參數設定為 1。如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_reset_external_master;
```

使用須知

主要使用者必須執行 mysql.rds_reset_external_master 程序。此程序必須在要做為 MySQL 執行個體 (在 Amazon RDS 外部執行) 之僅供讀取複本的將被移除的 MySQL 資料庫執行個體上執行。

Note

如果可能，建議您使用僅供讀取複本來管理兩個 Amazon RDS 資料庫執行個體間的複寫。當您這麼做時，建議您僅使用此程序和其他與複寫相關的預存程序。這些做法使得 Amazon RDS 資料庫執行個體之間能夠實行更複雜的複寫拓撲。我們提供這些預存程序主要是為了對

Amazon RDS 外部執行的 MySQL 執行個體啟用複寫。如需 Amazon RDS 資料庫執行個體之間管理複寫的相關資訊，請參閱[使用資料庫執行個體僅供讀取複本](#)。

如需使用複寫從 Amazon RDS 外部執行的 MySQL 執行個體匯入資料的詳細資訊，請參閱[使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。

mysql.rds_set_external_master

將 RDS for MySQL 資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。

Important

若要執行此程序，必須啟用 autocommit。若要啟用它，請將 autocommit 參數設定為 1。如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

Note

您可以使用 [mysql.rds_set_external_master_with_delay](#) 預存程序來設定外部來源資料庫執行個體與延遲的複寫。

語法

```
CALL mysql.rds_set_external_master (  
    host_name  
    , host_port  
    , replication_user_name  
    , replication_user_password  
    , mysql_binary_log_file_name  
    , mysql_binary_log_file_location  
    , ssl_encryption  
);
```

參數

host_name

要成為來源資料庫執行個體之 MySQL 執行個體 (在 Amazon RDS 外部執行) 的主機名稱或 IP 地址。

host_port

要設定為來源資料庫執行個體之 MySQL 執行個體 (在 Amazon RDS 外部執行) 所使用的連線埠。如果網路組態包含會轉換連線埠號碼的安全殼層 (SSH) 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在 Amazon RDS 外部執行的 MySQL 執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 許可的使用者 ID。我們建議您提供單獨用於外部執行個體複寫的帳戶。

replication_user_password

`replication_user_name` 中指定之使用者 ID 的密碼。

mysql_binary_log_file_name

來源資料庫執行個體上包含複寫資訊之二進位日誌的名稱。

mysql_binary_log_file_location

複寫在 `mysql_binary_log_file_name` 二進位日誌中開始讀取複寫資訊的位置。

您可以藉由在來源資料庫執行個體上執行 `SHOW MASTER STATUS` 來判斷 binlog 檔案名稱和位置。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 `MASTER_SSL_VERIFY_SERVER_CERT` 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

使用須知

主要使用者必須執行 `mysql.rds_set_external_master` 程序。此程序必須在要設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 之僅供讀取複本的 MySQL 資料庫執行個體上執行。

在執行 `mysql.rds_set_external_master` 之前，您必須將 Amazon RDS 外部執行的 MySQL 執行個體設定為來源資料庫執行個體。若要連線至 Amazon RDS 外部執行的 MySQL 執行個體，您必須指定 `replication_user_name` 和 `replication_user_password` 值，以指出在外部 MySQL 執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 許可的複寫使用者。

將外部 MySQL 執行個體設定為來源資料庫執行個體

1. 使用您選擇的 MySQL 用戶端，連線至外部 MySQL 執行個體，並建立用於複寫的使用者帳戶。以下是範例。

MySQL 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY 'password';
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

2. 在外部 MySQL 執行個體上，將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。下列範例將所有資料庫上的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的 'repl_user' 使用者。

MySQL 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

若要使用加密複寫，請將來源資料庫執行個體設定為使用 SSL 連線。

Note

如果可能，建議您使用僅供讀取複本來管理兩個 Amazon RDS 資料庫執行個體間的複寫。當您這麼做時，建議您僅使用此程序和其他與複寫相關的預存程序。這些做法使得 Amazon RDS 資料庫執行個體之間能夠實行更複雜的複寫拓撲。我們提供這些預存程序主要是為了對 Amazon RDS 外部執行的 MySQL 執行個體啟用複寫。如需 Amazon RDS 資料庫執行個體之間管理複寫的相關資訊，請參閱[使用資料庫執行個體僅供讀取複本](#)。

呼叫 `mysql.rds_set_external_master` 將 Amazon RDS 資料庫執行個體設定為僅供讀取複本之後，您可以在僅供讀取複本上呼叫 [mysql.rds_start_replication](#) 來啟動複寫程序。您可以呼叫 [mysql.rds_reset_external_master](#) 來移除僅供讀取複本組態。

呼叫 `mysql.rds_set_external_master` 時，Amazon RDS 將時間、使用者和 `set master` 的動作記錄在 `mysql.rds_history` 和 `mysql.rds_replication_status` 資料表中。

範例

在 MySQL 資料庫執行個體上執行時，下列範例會將資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。

```
call mysql.rds_set_external_master(  
  'Externaldb.some.com',  
  3306,  
  'repl_user',  
  'password',  
  'mysql-bin-changelog.0777',  
  120,  
  0);
```

`mysql.rds_set_external_master_with_auto_position`

將 RDS for MySQL 資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。此程序也會依據全域交易識別符 (GTID) 設定延遲複寫和複寫。

⚠ Important

若要執行此程序，必須啟用 `autocommit`。若要啟用它，請將 `autocommit` 參數設定為 1。如需修改參數的相關資訊，請參閱 [修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_set_external_master_with_auto_position (  
    host_name  
    , host_port  
    , replication_user_name  
    , replication_user_password  
    , ssl_encryption  
    , delay  
);
```

參數***host_name***

要成為來源資料庫執行個體之 MySQL 執行個體 (在 Amazon RDS 外部執行) 的主機名稱或 IP 地址。

host_port

要設定為來源資料庫執行個體之 MySQL 執行個體 (在 Amazon RDS 外部執行) 所使用的連線埠。如果網路組態包含會轉換連線埠號碼的安全殼層 (SSH) 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在 Amazon RDS 外部執行的 MySQL 執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 許可的使用者 ID。我們建議您提供單獨用於外部執行個體複寫的帳戶。

replication_user_password

`replication_user_name` 中指定之使用者 ID 的密碼。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 MASTER_SSL_VERIFY_SERVER_CERT 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

##

從來源資料庫執行個體延遲複寫的最低秒數。

此參數的限制為一日 (86400 秒)。

使用須知

主要使用者必須執行 `mysql.rds_set_external_master_with_auto_position` 程序。此程序必須在要設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 之僅供讀取複本的 MySQL 資料庫執行個體上執行。

RDS for MySQL 5.7 版，以及 RDS for MySQL 8.0.26 版和更新的 8.0 版都支援此程序。

在執行 `mysql.rds_set_external_master_with_auto_position` 之前，您必須將 Amazon RDS 外部執行的 MySQL 執行個體設定為來源資料庫執行個體。若要連線到 Amazon RDS 外部執行的 MySQL 執行個體，您必須指定 `replication_user_name` 和 `replication_user_password` 的值。這些值必須指明具有 MySQL 外部執行個體 REPLICATION CLIENT 和 REPLICATION SLAVE 許可的複寫使用者。

將外部 MySQL 執行個體設定為來源資料庫執行個體

1. 使用您選擇的 MySQL 用戶端，連線至外部 MySQL 執行個體，並建立用於複寫的使用者帳戶。以下是範例。

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

2. 在外部 MySQL 執行個體上，將 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予複寫使用者。下列範例將所有資料庫上的 REPLICATION CLIENT 和 REPLICATION SLAVE 權限授予您網域中的 'repl_user' 使用者。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```


如需更多詳細資訊，請參閱 [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。

Note

如果可能，建議您使用僅供讀取複本來管理兩個 Amazon RDS 資料庫執行個體間的複寫。當您這麼做時，建議您僅使用此程序和其他與複寫相關的預存程序。這些做法使得 Amazon RDS 資料庫執行個體之間能夠實行更複雜的複寫拓撲。我們提供這些預存程序主要是為了對 Amazon RDS 外部執行的 MySQL 執行個體啟用複寫。如需 Amazon RDS 資料庫執行個體之間管理複寫的相關資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

呼叫 `mysql.rds_set_external_master_with_auto_position` 將 Amazon RDS 資料庫執行個體設定為僅供讀取複本之後，您可以在僅供讀取複本上呼叫 [mysql.rds_start_replication](#) 來啟動複寫程序。您可以呼叫 [mysql.rds_reset_external_master](#) 來移除僅供讀取複本組態。

當您呼叫 `mysql.rds_set_external_master_with_auto_position`，Amazon RDS 會將時間、使用者、set master 的動作記錄在 `mysql.rds_history` 和 `mysql.rds_replication_status` 資料表中。

進行災難復原時，您可以透過 [mysql.rds_start_replication_until](#) 或 [mysql.rds_start_replication_until_gtid](#) 預存程序來使用此程序。若要變更延遲的僅供讀取複本，並向前捲動到災難發生前的時間點，可以執行 `mysql.rds_set_external_master_with_auto_position` 程序。在 `mysql.rds_start_replication_until_gtid` 程序停止複寫後，您可使用 [提升僅供讀取複本為獨立的資料庫執行個體](#) 中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

若要使用 `mysql.rds_start_replication_until_gtid` 程序，需啟用依據 GTID 的複寫。若要略過已知會導致災難的特定 GTID 型交易，可以使用 [mysql.rds_skip_transaction_with_gtid](#) 預存程序。如需有關依據 GTID 複寫的詳細資訊，請參閱 [使用 GTID 式複寫](#)。

範例

在 MySQL 資料庫執行個體上執行時，下列範例會將資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。它會在 MySQL 資料庫執行個體上設最小複寫延遲為一小時 (3600 秒)。在 Amazon RDS 外部執行的 MySQL 來源資料庫執行個體，其中變更至少一小時內不會套用在 MySQL 資料庫執行個體僅供讀取複本上。

```
call mysql.rds_set_external_master_with_auto_position(  
    'Externaldb.some.com',
```

```
3306,  
'repl_user',  
'SomePassW0rd',  
0,  
3600);
```

mysql.rds_set_external_master_with_delay

將 RDS for MySQL 資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本，並設定延遲複寫。

Important

若要執行此程序，必須啟用 autocommit。若要啟用它，請將 autocommit 參數設定為 1。如需修改參數的相關資訊，請參閱[修改資料庫參數群組中的參數](#)。

語法

```
CALL mysql.rds_set_external_master_with_delay (  
  host_name  
  , host_port  
  , replication_user_name  
  , replication_user_password  
  , mysql_binary_log_file_name  
  , mysql_binary_log_file_location  
  , ssl_encryption  
  , delay  
);
```

參數

host_name

要成為來源資料庫執行個體、在 Amazon RDS 外部執行的 MySQL 執行個體的主機名稱或 IP 地址。

host_port

要設定為來源資料庫執行個體之 MySQL 執行個體 (在 Amazon RDS 外部執行) 所使用的連線埠。如果網路組態包含會轉換連線埠號碼的 SSH 連線埠複寫，請指定 SSH 所公開的連線埠號碼。

replication_user_name

在 Amazon RDS 外部執行的 MySQL 執行個體上具有 REPLICATION CLIENT 和 REPLICATION SLAVE 許可的使用者 ID。我們建議您提供單獨用於外部執行個體複寫的帳戶。

replication_user_password

`replication_user_name` 中指定之使用者 ID 的密碼。

mysql_binary_log_file_name

來源資料庫執行個體上包含複寫資訊之二進位日誌的名稱。

mysql_binary_log_file_location

在 `mysql_binary_log_file_name` 二進位日誌中的複寫將開始讀取複寫資訊的位置。

您可以藉由在來源資料庫執行個體上執行 `SHOW MASTER STATUS` 來判斷 binlog 檔案名稱和位置。

ssl_encryption

此值指定在複寫連線上是否使用 Secure Socket Layer (SSL) 加密。1 指定使用 SSL 加密，0 指定不使用加密。預設為 0。

Note

不支援 `MASTER_SSL_VERIFY_SERVER_CERT` 選項。此選項設定為 0，表示連線已加密，但憑證未經過驗證。

##

從來源資料庫執行個體延遲複寫的最低秒數。

此參數的限制為一日 (86400 秒)。

使用須知

主要使用者必須執行 `mysql.rds_set_external_master_with_delay` 程序。此程序必須在要設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 之僅供讀取複本的 MySQL 資料庫執行個體上執行。

在執行 `mysql.rds_set_external_master_with_delay` 之前，您必須將 Amazon RDS 外部執行的 MySQL 執行個體設定為來源資料庫執行個體。若要連線到 Amazon RDS 外部執行的 MySQL 執行個體，您必須指定 `replication_user_name` 和 `replication_user_password` 的值。這些值必須指明具有 MySQL 外部執行個體 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 許可的複寫使用者。

將外部 MySQL 執行個體設定為來源資料庫執行個體

1. 使用您選擇的 MySQL 用戶端，連線至外部 MySQL 執行個體，並建立用於複寫的使用者帳戶。以下是範例。

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

2. 在外部 MySQL 執行個體上，將 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限授予複寫使用者。下列範例將所有資料庫上的 `REPLICATION CLIENT` 和 `REPLICATION SLAVE` 權限授予您網域中的 'repl_user' 使用者。

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

如需更多詳細資訊，請參閱 [使用外部來源執行個體設定二進位日誌檔案位置複寫](#)。

Note

如果可能，建議您使用僅供讀取複本來管理兩個 Amazon RDS 資料庫執行個體間的複寫。當您這麼做時，建議您僅使用此程序和其他與複寫相關的預存程序。這些做法使得 Amazon RDS 資料庫執行個體之間能夠實行更複雜的複寫拓撲。我們提供這些預存程序主要是為了對 Amazon RDS 外部執行的 MySQL 執行個體啟用複寫。如需 Amazon RDS 資料庫執行個體之間管理複寫的相關資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

呼叫 `mysql.rds_set_external_master_with_delay` 將 Amazon RDS 資料庫執行個體設定為僅供讀取複本之後，您可以在僅供讀取複本上呼叫 [mysql.rds_start_replication](#) 來啟動複寫程序。您可以呼叫 [mysql.rds_reset_external_master](#) 來移除僅供讀取複本組態。

當您呼叫 `mysql.rds_set_external_master_with_delay`，Amazon RDS 會將時間、使用者、set master 的動作記錄在 `mysql.rds_history` 和 `mysql.rds_replication_status` 資料表中。

進行災難復原時，您可以透過 [mysql.rds_start_replication_until](#) 或 [mysql.rds_start_replication_until_gtid](#) 預存程序來使用此程序。若要變更延遲的僅供讀取複本，並向前捲動到災難發生前的時間點，可以執行 `mysql.rds_set_external_master_with_delay` 程序。在 `mysql.rds_start_replication_until` 程序停止複寫後，您可使用 [提升僅供讀取複本為獨立的資料庫執行個體](#) 中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

若要使用 `mysql.rds_rds_start_replication_until_gtid` 程序，需啟用依據 GTID 的複寫。若要略過已知會導致災難的特定 GTID 型交易，可以使用 [mysql.rds_skip_transaction_with_gtid](#) 預存程序。如需有關依據 GTID 複寫的詳細資訊，請參閱[使用 GTID 式複寫](#)。

以下的 RDS for MySQL 版本中可使用 `mysql.rds_set_external_master_with_delay` 程序：

- MySQL 8.0.26 和更新的 8.0 版
- 所有 5.7 版

範例

在 MySQL 資料庫執行個體上執行時，下列範例會將資料庫執行個體設定為 MySQL 執行個體 (在 Amazon RDS 外部執行) 的僅供讀取複本。它會在 MySQL 資料庫執行個體上設最小複寫延遲為一小時 (3600 秒)。在 Amazon RDS 外部執行的 MySQL 來源資料庫執行個體，其中變更至少一小時內不會套用在 MySQL 資料庫執行個體僅供讀取複本上。

```
call mysql.rds_set_external_master_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  0,
  3600);
```

mysql.rds_set_master_auto_position

將複寫模式設為依據二進制日誌檔案位置或全域交易識別符 (GTID)。

語法

```
CALL mysql.rds_set_master_auto_position (
  auto_position_mode
```

```
);
```

參數

auto_position_mode

此值指示要使用日誌檔案位置複寫或是 GTID 複寫：

- 0 – 使用依據二進制日誌檔案位置的複寫模式。預設值為 0。
- 1 – 使用依據 GTID 的複寫方法。

使用須知

主要使用者必須執行 `mysql.rds_set_master_auto_position` 程序。

RDS for MySQL 5.7 版，以及 RDS for MySQL 8.0.26 版和更新的 8.0 版都支援此程序。

`mysql.rds_set_source_delay`

設定最低秒數，以延遲來源資料庫執行個體到目前僅供讀取複本的複寫。當您連線到僅供讀取複本時，請使用此程序來延遲從來源資料庫執行個體的複寫。

語法

```
CALL mysql.rds_set_source_delay(  
delay  
);
```

參數

##

從來源資料庫執行個體延遲複寫的最低秒數。

此參數的限制為一日 (86400 秒)。

使用須知

主要使用者必須執行 `mysql.rds_set_source_delay` 程序。

進行災難復原時，您可以透過 [mysql.rds_start_replication_until](#) 預存程序或 [mysql.rds_start_replication_until_gtid](#) 預存程序來使用此程序。若要變更延遲的僅供讀取複本，

並向前捲動到災難發生前的時間點，可以執行 `mysql.rds_set_source_delay` 程序。在 `mysql.rds_start_replication_until` 或 `mysql.rds_start_replication_until_gtid` 程序停止複製後，您可使用[提升僅供讀取複本為獨立的資料庫執行個體](#)中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

若要使用 `mysql.rds_rds_start_replication_until_gtid` 程序，需啟用依據 GTID 的複製。若要略過已知會導致災難的特定 GTID 型交易，可以使用 [mysql.rds_skip_transaction_with_gtid](#) 預存程序。如需依據 GTID 複製的詳細資訊，請參閱[使用 GTID 式複製](#)。

以下的 RDS for MySQL 版本中可使用 `mysql.rds_set_source_delay` 程序：

- MySQL 8.0.26 和更新的 8.0 版
- 所有 5.7 版

範例

若要延遲來源資料庫執行個體到目前僅供讀取複本的複製至少一小時 (3,600 秒)，您可透過下列參數呼叫 `mysql.rds_set_source_delay`：

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_skip_transaction_with_gtid

略過 MySQL 資料庫執行個體上具有指定全域交易識別符 (GTID) 之交易的複製。

若已知特定 GTID 交易導致錯誤，可以使用此程序進行災難復原。使用此預存程序來略過有問題的交易。有問題的交易範例包括停用複製、刪除重要資料或導致資料庫執行個體無法使用的交易。

語法

```
CALL mysql.rds_skip_transaction_with_gtid (  
gtid_to_skip  
);
```

參數

gtid_to_skip

要略過的複製交易的 GTID。

使用須知

主要使用者必須執行 `mysql.rds_skip_transaction_with_gtid` 程序。

RDS for MySQL 5.7 版，以及 RDS for MySQL 8.0.26 版和更新的 8.0 版都支援此程序。

範例

下列範例會略過使用 GTID 3E11FA47-71CA-11E1-9E33-C80AA9429562:23 進行交易的複寫。

```
CALL mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_skip_repl_error

略過和刪除 MySQL 資料庫僅供讀取複本上的複寫錯誤。

語法

```
CALL mysql.rds_skip_repl_error;
```

使用須知

主要使用者必須在僅供讀取複本上執行 `mysql.rds_skip_repl_error` 程序。如需有關此程序的詳細資訊，請參閱[呼叫 `mysql.rds_skip_repl_error` 程序](#)。

若要判斷是否有錯誤，執行 MySQL `SHOW REPLICA STATUS\G` 命令。如果複寫錯誤不嚴重，您可以執行 `mysql.rds_skip_repl_error` 來略過錯誤。如果有多個錯誤，`mysql.rds_skip_repl_error` 會刪除第一個錯誤，然後警告還有其他錯誤。然後，您可以使用 `SHOW REPLICA STATUS\G`，以針對下一個錯誤判斷正確的行動步驟。如需傳回值的相關資訊，請參閱 MySQL 文件中的 [SHOW REPLICA STATUS 陳述式](#)。

Note

MySQL 以前的版本使用 `SHOW SLAVE STATUS` 而不是 `SHOW REPLICA STATUS`。如果您使用的 MySQL 是 8.0.23 之前的版本，請使用 `SHOW SLAVE STATUS`。

如需有關解決 Amazon RDS 複寫錯誤的詳細資訊，請參閱 [對 MySQL 僅供讀取複本問題進行故障診斷](#)。

複寫已停止錯誤

當您呼叫 `mysql.rds_skip_repl_error` 程序時，可能會收到錯誤訊息，指出複本已關閉或停用。

如果您在主要執行個體而非僅供讀取複本上執行程序，此錯誤訊息就會出現。您必須在僅供讀取複本上執行此程序，程序才能運作。

如果您在僅供讀取複本上執行程序，但複寫無法成功重新啟動，此錯誤訊息也可能出現。

如果您需要略過大量錯誤，複寫延遲可能增加至超出二進位日誌(binlog) 檔案的預設保留期間。在此情況下，由於在清除 binlog 檔案之前已在僅供讀取複本上重播該檔案，您可能會遇到嚴重錯誤。此清除動作會導致複寫停止，而您將無法再呼叫 `mysql.rds_skip_repl_error` 命令來略過複寫錯誤。

透過增加 binlog 檔案在來源資料庫執行個體上保留的小時數，即可以減輕此問題。在延長二進位記錄檔保留時間之後，您可以重新啟動複寫，並視需要呼叫 `mysql.rds_skip_repl_error` 命令。

若要設定 binlog 保留時間，請使用 [mysql.rds_set_configuration](#) 程序，並指定 'binlog retention hours' 組態參數加上資料庫叢集上保留 binlog 檔案的時數。下列範例會將 binlog 檔案的保留期間設定為 48 小時。

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_start_replication

從 RDS for MySQL 資料庫執行個體起始複寫。

Note

您可使用 [mysql.rds_start_replication_until](#) 或 [mysql.rds_start_replication_until_gtid](#) 預存程序從 RDS for MySQL 資料庫執行個體來啟動複寫，並從特定的二進位日誌檔案位置停止複寫。

語法

```
CALL mysql.rds_start_replication;
```

使用須知

主要使用者必須執行 `mysql.rds_start_replication` 程序。

如果要從 Amazon RDS 外部的 MySQL 執行個體匯入資料，請呼叫 `mysql.rds_set_external_master` 來建置複寫組態，再呼叫僅供讀取複本上的 `mysql.rds_start_replication` 來啟動複寫程序。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

若要將資料匯出至 Amazon RDS 外部的 MySQL 執行個體，請在僅供讀取複本上呼叫 `mysql.rds_start_replication` 和 `mysql.rds_stop_replication` 來控制某些複寫動作，例如清除二進位日誌。如需詳細資訊，請參閱 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

您也可以對僅供讀取複本呼叫 `mysql.rds_start_replication`，以重新啟動您先前呼叫 `mysql.rds_stop_replication` 所停止的任何複寫程序。如需詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

`mysql.rds_start_replication_until`

從 RDS for MySQL 資料庫執行個體啟動複寫，並從特定的二進位日誌檔案位置停止複寫。

語法

```
CALL mysql.rds_start_replication_until (  
  replication_log_file  
  , replication_stop_point  
);
```

參數

replication_log_file

來源資料庫執行個體上包含複寫資訊之二進位日誌的名稱。

replication_stop_point

在 `replication_log_file` 二進位日誌中的複寫將停止的位置。

使用須知

主要使用者必須執行 `mysql.rds_start_replication_until` 程序。

以下的 RDS for MySQL 版本中可使用 `mysql.rds_start_replication_until` 程序：

- MySQL 8.0.26 和更新的 8.0 版

- 所有 5.7 版

您可以透過延遲複寫來使用此程序進行災難復原。若您已延遲複寫設定，您可使用此程序變更延遲的僅供讀取複本，並向前捲動到災難發生前的時間點。在此程序停止複寫後，您可使用 [提升僅供讀取複本為獨立的資料庫執行個體](#) 中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

您可使用以下預存程序來設定延遲複寫：

- [mysql.rds_set_configuration](#)
- [mysql.rds_set_external_master_with_delay](#)
- [mysql.rds_set_source_delay](#)

`replication_log_file` 參數的指定檔名必須與來源資料庫執行個體 binlog 檔案的名稱相同。

當 `replication_stop_point` 參數指定了過去的一個停止位置，複寫即會立即停止。

範例

以下範例會啟動複寫並複寫變更，直到達到 120 二進位日誌檔案中的位置 `mysql-bin-changelog.000777` 為止。

```
call mysql.rds_start_replication_until(  
    'mysql-bin-changelog.000777',  
    120);
```

mysql.rds_start_replication_until_gtid

從 RDS for MySQL 資料庫執行個體 啟動複寫，並在指定的全域交易識別碼 (GTID) 之後立即停止複寫。

語法

```
CALL mysql.rds_start_replication_until_gtid(gtid);
```

參數

gtid

在此 GTID 後停止複寫。

使用須知

主要使用者必須執行 `mysql.rds_start_replication_until_gtid` 程序。

RDS for MySQL 5.7 版，以及 RDS for MySQL 8.0.26 版和更新的 8.0 版都支援此程序。

您可以透過延遲複寫來使用此程序進行災難復原。若您已延遲複寫設定，您可使用此程序變更延遲的僅供讀取複本，並向前捲動到災難發生前的時間點。在此程序停止複寫後，您可使用 [提升僅供讀取複本為獨立的資料庫執行個體](#) 中的說明來提升該僅供讀取複本成為新的主要資料庫執行個體。

您可使用以下預存程序來設定延遲複寫：

- [mysql.rds_set_configuration](#)
- [mysql.rds_set_external_master_with_delay](#)
- [mysql.rds_set_source_delay](#)

當 `gtid` 參數指定了複本已經執行的交易時，複寫會立即停止。

範例

以下範例會啟動複寫並複寫變更，直到達到 GTID 3E11FA47-71CA-11E1-9E33-C80AA9429562:23 為止。

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_stop_replication

從 MySQL 資料庫執行個體停止複寫。

語法

```
CALL mysql.rds_stop_replication;
```

使用須知

主要使用者必須執行 `mysql.rds_stop_replication` 程序。

如果您要將複寫設定為從 Amazon RDS 外部執行的 MySQL 執行個體匯入資料，在匯入完成之後，請在僅供讀取複本上呼叫 `mysql.rds_stop_replication` 來停止複寫程序。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

如果您要設定複寫將資料匯出至 Amazon RDS 外部的 MySQL 執行個體，請在僅供讀取複本上呼叫 `mysql.rds_start_replication` 和 `mysql.rds_stop_replication` 來控制某些複寫動作，例如清除二進位日誌。如需詳細資訊，請參閱 [使用複寫從 MySQL 資料庫執行個體匯出資料](#)。

您也可以使用 `mysql.rds_stop_replication`，以停止兩個 Amazon RDS 資料庫執行個體之間的複寫。您停止複寫通常是為了在僅供讀取複本上執行長時間的操作，例如在僅供讀取複本上建立大型索引。您可以在僅供讀取複本上呼叫 [mysql.rds_start_replication](#)，以重新啟動您已停止的任何複寫程序。如需詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

預備 InnoDB 快取

下列預存程序會在 RDS for MySQL 資料庫執行個體上儲存、載入或取消載入 InnoDB 緩衝集區。如需更多詳細資訊，請參閱 [Amazon RDS 上 MySQL 的 InnoDB 快取預備](#)。

主題

- [mysql.rds_innodb_buffer_pool_dump_now](#)
- [mysql.rds_innodb_buffer_pool_load_abort](#)
- [mysql.rds_innodb_buffer_pool_load_now](#)

mysql.rds_innodb_buffer_pool_dump_now

將目前的緩衝集區狀態傾出至磁碟。

語法

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

使用須知

主要使用者必須執行 `mysql.rds_innodb_buffer_pool_dump_now` 程序。

mysql.rds_innodb_buffer_pool_load_abort

正在載入已儲存的緩衝集區狀態時取消載入。

語法

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

使用須知

主要使用者必須執行 `mysql.rds_innodb_buffer_pool_load_abort` 程序。

mysql.rds_innodb_buffer_pool_load_now

從磁碟載入已儲存的緩衝集區狀態。

語法

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

使用須知

主要使用者必須執行 `mysql.rds_innodb_buffer_pool_load_now` 程序。

Amazon RDS for Oracle

Amazon RDS 會支援執行下列版本和 Oracle 資料庫版本的資料庫執行個體：

- Oracle Database 21c (21.0.0.0)
- Oracle 資料庫 19c (19.0.0.0)

Note

Oracle Database 11g、Oracle Database 12c 和 Oracle Database 18c 是 Amazon RDS 中已不再支援的舊版。

建立資料庫執行個體之前，請先完成本指南的[設定 Amazon RDS](#)一節中的步驟。當您使用主帳戶建立資料庫執行個體時，該帳戶會獲得 DBA 權限，但有一些限制。將此帳戶用於管理任務，例如建立其他資料庫帳戶。您無法使用 SYS、SYSTEM 或其他 Oracle 提供的管理帳戶。

您可以建立下列作業：

- 資料庫執行個體
- 資料庫快照
- 時間點還原
- 自動備份
- 手動備份

您可在 VPC 內使用執行 Oracle 的資料庫執行個體。您也可以透過啟用各種選項，新增功能至 Oracle 資料庫執行個體。Amazon RDS 支援 Oracle 的多個可用區部署，做為高可用性、容錯移轉解決方案。

Important

為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。而且會將存取權限制在某些需要進階權限的系統程序和資料表。您可以使用 Oracle SQL *Plus 等標準 SQL 用戶端存取資料庫。不過，您無法使用 Telnet 或 Secure Shell (SSH) 直接存取主機。

主題

- [Oracle on Amazon RDS 概觀](#)
- [連線至 RDS for Oracle 資料庫執行個體](#)
- [保護 Oracle 資料庫執行個體連線的安全](#)
- [在 RDS for Oracle 中使用 CDB](#)
- [管理您的 RDS for Oracle 資料庫執行個體](#)
- [設定進階 RDS for Oracle 功能](#)
- [將資料匯入 Amazon RDS 上的 Oracle](#)
- [使用適用於 Amazon RDS for Oracle 的僅供讀取複本](#)
- [將選項新增至 Oracle 資料庫執行個體](#)
- [升級 RDS for Oracle 資料庫引擎](#)
- [將 RDS for Oracle 資料庫執行個體與第三方軟體搭配使用](#)
- [Oracle Database 引擎版本備註](#)

Oracle on Amazon RDS 概觀

您可以閱讀以下各節以取得 RDS for Oracle 的概觀。

主題

- [RDS for Oracle 功能](#)
- [RDS for Oracle 版本](#)
- [RDS for Oracle 授權選項](#)
- [RDS for Oracle 的使用者和權限](#)
- [RDS for Oracle 執行個體類別](#)
- [RDS for Oracle 資料庫架構](#)
- [RDS for Oracle 參數](#)
- [RDS for Oracle 字元集](#)
- [RDS for Oracle 限制](#)

RDS for Oracle 功能

Amazon RDS for Oracle 支援 Oracle 資料庫的大多數特性和功能。部分功能的支援或權限可能有所限制。某些功能僅在 Enterprise Edition 中提供，有些功能則需要其他授權。如需特定 Oracle 資料庫版本的 Oracle 資料庫功能詳細資訊，請參閱您所使用版本的 Oracle 資料庫授權資訊使用者手冊。

您可以在[資料庫的最新資訊](#)頁面上篩選新的 Amazon RDS 功能。對於 Products (產品)，請選擇 Amazon RDS。然後使用關鍵字搜尋，例如 **Oracle 2022**。

Note

以下清單並非詳盡清單。

主題

- [RDS for Oracle 中的新功能](#)
- [RDS for Oracle 中支援的功能](#)
- [RDS for Oracle 中不支援的功能](#)

RDS for Oracle 中的新功能

若要查看適用於 Oracle 的 RDS 中的新功能，請使用下列技巧：

- 搜尋關鍵字 **Oracle** 的 [文件歷史記錄](#)。
- 在[資料庫的新增功能中篩選 Amazon RDS 新功能？](#) 頁面。對於 Products (產品)，請選擇 Amazon RDS。然後搜尋 **Oracle YYYY**，其中 **YYYY** 代表年份，如 **2024**。

RDS for Oracle 中支援的功能

Amazon RDS for Oracle 支援以下 Oracle 資料庫功能：

- 進階壓縮
- Application Express (APEX)

如需更多詳細資訊，請參閱 [Oracle Application Express \(APEX\)](#)。

- 自動記憶體管理

- 自動復原管理
- 自動工作負載儲存庫 (AWR)

如需詳細資訊，請參閱 [使用自動工作負載儲存庫 \(AWR\) 產生效能報告。](#)

- 在相同 AWS 區域或跨區 AWS 域提供最高效能的作用中資料保全

如需詳細資訊，請參閱 [使用適用於 Amazon RDS for Oracle 的僅供讀取複本。](#)

- Blockchain 資料表 (Oracle Database 21c 及更新版本)

如需詳細資訊，請參閱 Oracle Database 文件中的 [管理 Blockchain 資料表。](#)

- 連續查詢通知

如需詳細資訊，請參閱 Oracle 文件中的 [使用持續查詢通知 \(CQN\)。](#)

- Data Redaction

- 連續查詢通知

如需詳細資訊，請參閱 Oracle 文件中的 [資料庫變更通知。](#)

- 記憶體內資料庫
- 分散式查詢和交易
- 版本式重新定義

如需詳細資訊，請參閱 [設定資料庫執行個體的預設版本。](#)

- EM Express (12c 及更新版本)

如需詳細資訊，請參閱 [Oracle Enterprise Manager。](#)

- 精細稽核
- Flashback 資料表、Flashback 查詢、Flashback 交易查詢
- 應用程式的逐步密碼輪替 (Oracle Database 21c 及更新版本)

如需詳細資訊，請參閱 Oracle Database 文件中的 [管理應用程式的逐步資料庫密碼輪替。](#)

- HugePages

如需詳細資訊，請參閱 [對 RDS for Oracle 執行個體開啟巨型分頁。](#)

- 匯入/匯出 (舊版和 Data Pump) 和 SQL*Loader

如需更多詳細資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle。](#)

- Java 虛擬機器 (JVM)

如需詳細資訊，請參閱 [Oracle Java 虛擬機器](#)。

- JavaScript (甲骨文資料庫 21c 及更高版本)

如需詳細資訊，請參閱 Oracle Database 文件中的 [DBMS_MLE](#)。

- Label Security

如需詳細資訊，請參閱 [Oracle Label Security](#)。

- Locator

如需更多詳細資訊，請參閱 [Oracle Locator](#)。

- 具體化檢視

- 多租用戶

所有 Oracle Database 19c 及更新版本都支援 Oracle 多租用戶架構。如需詳細資訊，請參閱 [在 RDS for Oracle 中使用 CDB](#)。

- 網路加密

如需更多詳細資訊，請參閱 [Oracle 原生網路加密](#) 和 [Oracle Secure Sockets Layer](#)。

- 資料分割

- 實際應用程式測試

若要使用完整擷取和重新顯示功能，您必須使用 Amazon Elastic File System (Amazon EFS) 存取 Oracle Real 應用程式測試所產生的檔案。如需詳細資訊，請參閱 [Amazon EFS 整合](#) 和部落格文章 [與 Amazon RDS 版 Oracle 搭配使用 Oracle 實際應用程式測試功能](#)。

- 在應用模組層次分割 (但不是「Oracle 分片」功能)

- 空間和圖表

如需更多詳細資訊，請參閱 [Oracle Spatial](#)。

- Star 查詢最佳化

- 串流與進階佇列

- 摘要管理 – 具體化的檢視查詢改寫

- 文字 (不支援檔案與 URL 資料存放區類型)

- 全面調用

- 透明資料加密 (TDE)

如需詳細資訊，請參閱 [Oracle 透明資料加密](#)。

- Unified Auditing，混合模式

如需詳細資訊，請參閱 Oracle 文件中的 [混合模式稽核](#)。

- XML 資料庫 (無 XML 資料庫通訊協定伺服器)

如需更多詳細資訊，請參閱 [Oracle XML 資料庫](#)。

- 虛擬私有資料庫

RDS for Oracle 中不支援的功能

Amazon RDS for Oracle 不支援以下 Oracle 資料庫功能：

- 自動儲存體管理 (ASM)
- 資料庫保存庫
- Flashback 資料庫

Note

如需替代解決方案，請參閱適用於 Oracle 的 [Amazon RDS 中的資料 AWS 庫部落格項目 Oracle 倒溯資料庫功能的替代方案](#)。

- FTP 和 SFTP
- 混合式分割資料表
- 簡訊閘道
- Oracle Enterprise Manager Cloud Control 管理儲存庫
- Real Application Clusters (Oracle RAC)
- 實際應用程式安全性 (RAS)
- Unified Auditing，純模式
- Workspace Manager (WMSYS) 結構描述

Note

上述清單並不詳盡。

Warning

一般而言，Amazon RDS 不會阻止您為不支援的功能建立資料架構。不過，如果您為需要 SYSDBA 權限的 Oracle 功能和元件建立結構描述，則可能會損壞資料說明並影響資料庫執行個體的使用狀態。僅使用在 [將選項新增至 Oracle 資料庫執行個體](#) 中可用的支援功能和結構描述。

RDS for Oracle 版本

適用於甲骨文的 RDS 支持多個 Oracle 資料庫版本。

Note

如需升級版本的相關資訊，請參閱 [升級 RDS for Oracle 資料庫引擎](#)。

主題

- [Amazon RDS Oracle Database 21c](#)
- [Amazon RDS Oracle 資料庫 19c](#)

Amazon RDS Oracle Database 21c

Amazon RDS 支持甲骨文資料庫 21c，其中包括甲骨文企業版和甲骨文標準版 2。相較於舊版，Oracle Database 21c (21.0.0.0) 包含許多新功能和更新。一個關鍵的變更是 Oracle Database 21c 僅支援多重租用架構：您不能再將資料庫建立為傳統的非 CDB。若要進一步瞭解 CDB 和非 CDB 之間的區別，請參閱 [RDS for Oracle CDB 的限制](#)。

在此小節中，您可以找到在 Amazon RDS 上使用 Oracle Database 21c (21.0.0.0) 的重要功能和變更。如需變更的完整清單，請參閱 [Oracle Database 21c](#) 文件。如需每個 Oracle Database 21c 版本支援的完整功能清單，請參閱 Oracle 文件中 [Oracle 資料庫產品允許的功能、選項和管理套件](#)。

Oracle Database 21c (21.0.0.0) 的 Amazon RDS 參數變更

Oracle Database 21c (21.0.0.0) 包括數個新參數以及具有新範圍和新預設值的參數。

主題

- [新參數](#)

- [相容參數的變更](#)
- [已移除的參數](#)

新參數

下表顯示 Oracle Database 21c (21.0.0.0) 的新 Amazon RDS 參數。

名稱	值範圍	預設值	可修改	描述
blockchain_table_max_no_drop	NONE 0	NONE	Y	可讓您控制在建立區塊鏈表時可指定的最大閒置時間長度。
dbnest_enable	NONE CDB_RESOURCE_PDB_ALL	NONE	N	可讓您啟用或停用資料庫 Nest。DbNest 為 PDB 提供作業系統資源隔離與管理、檔案系統隔離，以及安全運算。
dbnest_pdb_fs_conf	NONE <i>pathname</i>	NONE	N	為 PDB 指定 dbNest 檔案系統組態檔案。
diagnostics_control	ERROR WARNING IGNORE	IGNORE	Y	允許您控制和監視執行在不安全的資料庫診斷操作的使用者。
drpc_dedicated_opt	YES NO	YES	Y	啟用或停用搭配資料庫駐留連線集區 (DRCP) 使用專屬最佳化。
enable_per_pdb_drpc	true false	true	N	控制資料庫駐留連線集區 (DRCP) 是為整個 CDB 設定一個連線集區，或是為各個 PDB 設定一個隔離的連線集區。
inmemory_deep_vectorization	true false	true	Y	啟用或停用深度向量化框架。

名稱	值範圍	預設值	可修改	描述
mandatory_user_profile	<i>profile_name</i>	N/A	N	為 CDB 或 PDB 指定必要的使用者描述檔。
optimizer_capture_sql_quarantine	true false	false	Y	啟用或停用深度向量化框架。
optimizer_use_sql_quarantine	true false	false	Y	啟用或停用 SQL 隔離組態的自動建立。
result_cache_execution_threshold	0 設定為 68719476736	2	Y	指定 PL/SQL 函數在其結果存放於結果快取之前可執行的最大次數。
result_cache_max_temp_result	0 設定為 100	5	Y	指定任何單一快取的查詢結果都可耗用的 RESULT_CACHE_MAX_TEMP_SIZE 百分比。
result_cache_max_temp_size	0 設定為 219902325552	RESULT_CACHE_SIZE * 10	Y	指定結果快取可耗用的最大臨時資料表空間數量 (以位元組為單位)。
sga_min_size	0 至 219902325552 (最大值為 sga_target 的 50%)	0	Y	指示可插入式資料庫 (PDB) 的 SGA 使用量的可能最小值。

名稱	值範圍	預設值	可修改	描述
tablespace_encryption_default_algorithm	GOST256 SEED128 ARIA256 ARIA192 ARIA128 3DES168 AES256 AES192 AES128	AES128	Y	指定資料庫在加密資料表空間時使用的預設演算法。

相容參數的變更

對於 Amazon RDS 上的 Oracle Database 21c (21.0.0.0) , `compatible` 參數有新的最大值。下表顯示新的預設值。

參數名稱	Oracle Database 21c (21.0.0.0) 最大值
compatible	21.0.0

已移除的參數

Oracle Database 21c (21.0.0.0) 中已移除下列參數：

- `remote_os_authent`
- `sec_case_sensitive_logon`
- `unified_audit_sga_queue_size`

Amazon RDS Oracle 資料庫 19c

Amazon RDS 支援 Oracle 資料庫 19c , 其中包括 Oracle Enterprise Edition 和 Oracle Standard Edition Two。

相較於舊版，Oracle 資料庫 19c (19.0.0.0) 包含許多新功能和更新。在此小節中，您可以找到在 Amazon RDS 上使用 Oracle 資料庫 19c (19.0.0.0) 的重要功能和變更。如需變更的完整清單，請參閱 [Oracle Database 19c](#) 文件。如需每個 Oracle 資料庫 19c 版本支援的完整功能清單，請參閱 Oracle 文件中 [Oracle 資料庫產品允許的功能、選項和管理套件](#)。

Oracle 資料庫 19c (19.0.0.0) 的 Amazon RDS 參數變更

Oracle 資料庫 19c (19.0.0.0) 包括數個新參數以及具有新範圍和新預設值的參數。

主題

- [新參數](#)
- [相容參數的變更](#)
- [已移除的參數](#)

新參數

下表顯示 Oracle 資料庫 19c (19.0.0.0) 的新 Amazon RDS 參數。

名稱	值	可修改	描述
lob_signature_enable	TRUE、FALSE (預設值)	Y	啟用或停用 LOB 定位器簽章功能。
max_datapump_parallel_per_job	1 至 1024，或 AUTO	Y	指定每個 Oracle Data Pump 工作允許的平行程序數量上限。

相容參數的變更

對於 Amazon RDS 上的 Oracle 資料庫 19c (19.0.0.0)，compatible 參數有新的最大值。下表顯示新的預設值。

參數名稱	Oracle 資料庫 19c (19.0.0.0) 最大值
compatible	19.0.0

已移除的參數

Oracle 資料庫 19c (19.0.0.0) 中已移除下列參數：

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

RDS for Oracle 授權選項

Amazon RDS for Oracle 有兩個授權選項：「已包含授權 (LI)」和「使用自有授權 (BYOL)」。在 Amazon RDS 上建立 Oracle 資料庫執行個體之後，您可以修改資料庫執行個體來變更授權模式。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Important

請確定您擁有適用於資料庫執行個體類別和 Oracle 資料庫版本的「軟體更新授權與 Support」的「Oracle 資料庫」授權。此外，請確定您擁有任何個別授權的「Oracle 資料庫」功能的授權。

主題

- [SE2 的已包含授權模型](#)
- [為 EE 和 SE2 攜帶您自己的授權 \(BYOL\)](#)
- [授權 Oracle 異地同步備份部署](#)

SE2 的已包含授權模型

在「已包含授權」模式中，您不需要另外購買 Oracle 資料庫授權。AWS 持有 Oracle 資料庫軟體的授權。僅適用於 Oracle 資料庫標準版 2 (SE2) 的 Amazon RDS 上支援「已包含授權」模式。

在此模型中，如果您的 AWS Support 帳戶具有案例支援，請聯絡 AWS Support Amazon RDS 和 Oracle 資料庫服務請求。您對 Oracle 版 RDS 的 LI 選項的使用受 [AWS 服務條款](#) 第 10.3.1 節的約束。

為 EE 和 SE2 攜帶您自己的授權 (BYOL)

在 BYOL 模型中，您可以使用現有 Oracle 資料庫授權在 Amazon RDS 上執行資料庫部署。Amazon RDS 僅支援 Oracle 資料庫企業版 (EE) 和甲骨文資料庫標準版 2 (SE2) 的 BYOL 模型。

對要執行的資料庫執行個體類別和 Oracle 資料庫版本，請確保具有適當的 Oracle 資料庫授權 (含軟體更新授權和支援)。您也必須遵循 Oracle 對雲端運算環境中授權 Oracle 資料庫軟體的政策。如需 Amazon EC2 的 Oracle 授權原則詳細資訊，請參閱[在雲端運算環境中授權 Oracle 軟體](#)。

在此模型中，您會繼續使用作用中 Oracle 支援帳戶，並且針對 Oracle 資料庫的服務請求，請直接聯絡 Oracle。如果您擁有提供案例支援的 AWS Support 帳戶，您可以聯絡以 AWS Support 取得 Amazon RDS 問題。Amazon Web Services 和 Oracle 皆提供多廠商支援程序，而您可視需要向這兩個組織尋求協助。

與整合 AWS License Manager

為了讓您更輕鬆監控 BYOL 模型內的 Oracle 授權用量，[AWS License Manager](#) 可與 Amazon RDS for Oracle 整合。License Manager 會支援追蹤 RDS for Oracle 引擎版本和以虛擬核心 (vCPU) 為基礎的授權套件。您也可以搭配使用 License Manager AWS Organizations 來集中管理所有組織帳戶。

下表顯示適用於 RDS for Oracle 的產品資訊篩選條件。

篩選條件	名稱	描述
引擎版本	oracle-ee	Oracle Database Enterprise Edition (EE)
	oracle-se2	甲骨文資料庫標準版 2 (第 2 季)
授權套件	data guard	請參閱 使用適用於 Amazon RDS for Oracle 的僅供讀取複本 (Oracle Active Data Guard)
	olap	請參閱 Oracle OLAP
	ols	請參閱 Oracle Label Security
	diagnostic pack sqlt	請參閱 Oracle SQLT
	tuning pack sqlt	請參閱 Oracle SQLT

若要追蹤 Oracle 資料庫執行個體的授權使用情況，您可以建立自我管理的授權。在此情況下，符合產品資訊篩選器的 RDS for Oracle 資源會自動與自我管理授權相關聯。探索 Oracle 資料庫執行個體最多可能需要 24 小時。

主控台

建立自我管理授權以追蹤 Oracle 資料庫執行個體的授權使用情況

1. 移至 <https://console.aws.amazon.com/license-manager/>。
2. 建立自我管理授權。

如需指示，請參閱AWS License Manager 使用指南中的[建立自我管理授權](#)。

在 Product Information (產品資訊) 面板中，新增 RDS Product Information Filter (RDS 產品資訊篩選條件) 的規則。

如需詳細資訊，請參閱 AWS License Manager API 參考[ProductInformation](#)中的。

AWS CLI

若要使用建立自我管理的授權 AWS CLI，請呼叫指[create-license-configuration](#)令。使用 `--cli-input-json` 或 `--cli-input-yaml` 參數，將參數傳遞至命令。

Example

下列範例會建立 Oracle 企業版的自我管理授權。

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-ee.json
```

下列是範例中使用的範例 `rds-oracle-ee.json` 檔案。

```
{
  "Name": "rds-oracle-ee",
  "Description": "RDS Oracle Enterprise Edition",
  "LicenseCountingType": "vCPU",
  "LicenseCountHardLimit": false,
  "ProductInformationList": [
    {
      "ResourceType": "RDS",
      "ProductInformationFilterList": [
        {
          "ProductInformationFilterName": "Engine Edition",
          "ProductInformationFilterValue": ["oracle-ee"],
          "ProductInformationFilterComparator": "EQUALS"
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

如需關於產品資訊的詳細資訊，請參閱《AWS License Manager 使用者指南》中的[自動化探索資源庫存](#)。

如需有關 `--cli-input` 參數的詳細資訊，請參閱[AWS CLI 使用指南中的從 JSON 或 YAML 輸入檔案產生 AWS CLI 架構和輸入參數](#)。

在 Oracle 版本之間遷移

如果擁有適合該版本的未使用 BYOL Oracle 授權和計劃執行的資料庫執行個體類別，您可從 Standard Edition 2 (SE2) 遷移至 Enterprise Edition (EE)。您無法從 Enterprise Edition 遷移到其他版本。

若要變更版本並保留資料

1. 建立資料庫執行個體的快照。

如需更多詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

2. 將快照還原到新的資料庫執行個體，並選取要使用的 Oracle 資料庫版本。

如需更多詳細資訊，請參閱 [從資料庫快照還原](#)。

3. (選用) 刪除舊資料庫執行個體，除非您希望其保持執行並擁有適當的 Oracle 資料庫授權。

如需更多詳細資訊，請參閱 [刪除資料庫執行個體](#)。

授權 Oracle 異地同步備份部署

Amazon RDS 支援 Oracle 的多個可用區部署，做為高可用性、容錯移轉解決方案。對於生產工作負載，建議使用多個可用區。如需更多詳細資訊，請參閱 [設定及管理多可用區部署](#)。

如果您使用「使用自有授權」模型，則必須具有多個可用區部署中主要資料庫執行個體和待命資料庫執行個體的授權。

RDS for Oracle 的使用者和權限

當您建立 Amazon RDS for Oracle 資料庫執行個體時，預設主要使用者在資料庫執行個體上擁有大部分的使用者許可上限。請將此主要使用者帳戶用於任何管理任務，例如在您的資料庫中建立額外的使用者帳戶。由於 RDS 是受管服務，因此不允許您以 SYS 和 SYSTEM 登入，因此沒有 SYSDBA 權限。

主題

- [Oracle DBA 權限的限制](#)
- [如何管理 SYS 物件的權限](#)

Oracle DBA 權限的限制

在資料庫中，角色是一組權限，您可以用來授予給使用者或撤銷。Oracle 資料庫使用角色來提供安全性。如需詳細資訊，請參閱 Oracle 資料庫說明文件中的[設定權限和角色授權](#)。

預先定義角色 DBA 一般會在 Oracle 資料庫上允許所有管理權限。建立資料庫執行個體時，您的主要使用者帳戶會取得 DBA 權限 (有一些限制)。為了提供受管體驗，RDS for Oracle 資料庫不會針對 DBA 角色提供下列權限：

- ALTER DATABASE
- ALTER SYSTEM
- CREATE ANY DIRECTORY
- DROP ANY DIRECTORY
- GRANT ANY PRIVILEGE
- GRANT ANY ROLE

如需 RDS for Oracle 權限和角色的詳細資訊，請參閱 [主要使用者帳戶權限](#)。

如何管理 SYS 物件的權限

您可以透過使用 `rdsadmin.rdsadmin_util` 套件管理 SYS 物件上的權限。例如，如果您建立資料庫使用者 `myuser`，您可以使用 `rdsadmin.rdsadmin_util.grant_sys_object` 程序將 SELECT 權限授予 `V_$SQLAREA` 至 `myuser`。如需詳細資訊，請參閱下列主題：

- [將 SELECT 或 EXECUTE 權限授予 SYS 物件](#)
- [撤銷 SYS 物件上的 SELECT 或 EXECUTE 權限](#)
- [將權限授予非主要使用者](#)

RDS for Oracle 執行個體類別

RDS for Oracle 資料庫執行個體的運算和記憶體容量取決於其執行個體類別。您需要的資料庫執行個體類別取決於您的處理能力和記憶體需求。

支援的 RDS for Oracle 執行個體類別

支援的 RDS for Oracle 資料庫執行個體類別是 RDS 資料庫執行個體類別的子集。如需 RDS 執行個體類別的完整清單，請參閱 [資料庫執行個體類別](#)。

適用 Oracle 記憶體最佳化執行個體類別的 RDS

RDS for Oracle 也提供針對需要額外記憶體、儲存空間和每個 vCPU 輸入/輸出的工作負載進行最佳化的執行個體類別。這些執行個體類別使用下列命名慣例：

```
db.r5b.instance_size.tpcthreads_per_core.memratio  
db.r5.instance_size.tpcthreads_per_core.memratio
```

下列是受支援的執行個體類別範例：

```
db.r5b.4xlarge.tpc2.mem2x
```

上述的執行個體類別的元件如下所示：

- db.r5b.4xlarge – 執行個體類別的名稱。
- tpc2 – 每個核心的執行緒。值為 2 表示多執行緒已開啟。如果值為 1，則表示多執行緒已關閉。
- mem2x – 執行個體類別的額外記憶體與標準記憶體的比率。在此範例中，最佳化提供的記憶體數量是標準 db.r5.4xlarge 執行個體的兩倍。

適用於 Oracle 的 RDS 中支援的版本、執行個體類別和授權組合

如果您使用 RDS 主控台，可以選擇 [建立資料庫] 並指定其他選項，以瞭解是否支援特定版本、執行個體類別和授權組合。在中 AWS CLI，您可以執行下列命令：

```
aws rds describe-orderable-db-instance-options --engine engine-type --license-model license-type
```

下表列出適用於 Oracle 的 RDS 支援的所有版本、執行個體類別和授權類型。如需各類型之記憶體屬性的詳細資訊，請參閱 [RDS for Oracle 執行個體類型](#)。如需定價的相關資訊，請參閱 [適用於甲骨文定價模式的 Amazon RDS](#)。

Oracle 版本	甲骨文資料庫 19c 及更高版本
Enterprise Edition (EE)	標準執行個體類別
使用自有授權 (BYOL)	数据库 .m6. 大 — 数据库 m6i.32 X 大 db.m5d.large–db.m5d.24xlarge db.m5.large–db.m5.24xlarge
	記憶體最佳化資料庫執行個體類別 db.r6i.large–db.r6i.32xlarge db.r5d.large–db.r5d.24xlarge db.r5b.8xlarge.tpc2.mem3x db.r5b.6xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem3x db.r5b.4xlarge.tpc2.mem2x db.r5b.2xlarge.tpc2.mem8x db.r5b.2xlarge.tpc2.mem4x db.r5b.2xlarge.tpc1.mem2x db.r5b.xlarge.tpc2.mem4x db.r5b.xlarge.tpc2.mem2x db.r5b.large.tpc1.mem2x db.r5b.large–db.r5b.24xlarge db.r5.12xlarge.tpc2.mem2x db.r5.8xlarge.tpc2.mem3x

Oracle 版本	甲骨文資料庫 19c 及更高版本
	db.r5.6xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.24xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge db.x2iezn.2xlarge–db.x2iezn.12xlarge db.x2idn.16xlarge–db.x2idn.32xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.z1d.large–db.z1d.12xlarge
	爆量效能執行個體類別 db.t3.small–db.t3.2xlarge
Standard Edition 2 (SE2) 使用自有授權 (BYOL)	標準執行個體類別

Oracle 版本	甲骨文資料庫 19c 及更高版本
	DB.m6. 大 — 數據庫. m6 l.4X 大 db.m5d.large–db.m5d.4xlarge db.m5.large–db.m5.4xlarge
	記憶體最佳化資料庫執行個體類別 db.r6i.large–db.r6i.4xlarge db.r5d.large–db.r5d.4xlarge db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.4xlarge db.r5b.large–db.r5b.4xlarge db.x2iedn.xlarge–db.x2iedn.4xlarge db.x2iezn.2xlarge–db.x2iezn.4xlarge db.z1d.large–db.z1d.3xlarge
	爆量效能執行個體類別

Oracle 版本	甲骨文資料庫 19c 及更高版本
	db.t3.small–db.t3.2xlarge
Standard Edition 2 (SE2)	標準執行個體類別
已包含授權	db.m5.large–db.m5.4xlarge
	記憶體最佳化資料庫執行個體類別
	db.r6i.large–db.r6i.4xlarge
	db.r5.large–db.r5.4xlarge
	爆量效能執行個體類別
	db.t3.small–db.t3.2xlarge

Note

我們鼓勵所有 BYOL 客戶參閱其授權合約，來評估 Amazon RDS for Oracle 取代的影響。如需 RDS for Oracle 支援之資料庫執行個體類別的運算容量詳細資訊，請參閱 [資料庫執行個體類別](#) 和 [針對 RDS for Oracle 中的資料庫執行個體類別設定處理器](#)。

Note

如果您有資料庫執行個體的資料庫快照而該資料庫執行個體使用已取代的資料庫執行個體類別，則您可在還原資料庫快照時，選擇使用尚未取代的資料庫執行個體類別。如需詳細資訊，請參閱 [從資料庫快照還原](#)。

Oracle 資料庫執行個體類別已淘汰的 RDS

下面是 RDS for Oracle 取代的資料庫執行個體類別：

- db.m1、db.m2、db.m3、db.m4
- db.t1、db.t2

- db.r1、db.r2、db.r3、db.r4

上述資料庫執行個體類別已由全面供應、價格更低且效能更好的資料庫執行個體類別取代。如果您有資料庫執行個體使用已棄用的資料庫執行個體類別，則具有下列選項：

- 允許 Amazon RDS 自動修改每個資料庫執行個體，以使用類似的非棄用資料庫執行個體類別。如需棄用時間軸，請參閱 [資料庫執行個體類別的類型](#)。
- 修改資料庫執行個體來自行變更資料庫執行個體類別。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

如果您有資料庫執行個體的資料庫快照而該資料庫執行個體使用已取代的資料庫執行個體類別，則您可在還原資料庫快照時，選擇使用尚未取代的資料庫執行個體類別。如需更多詳細資訊，請參閱 [從資料庫快照還原](#)。

RDS for Oracle 資料庫架構

Oracle 多租戶架構 (也稱為 CDB 架構) 可讓 Oracle 資料庫作為多租戶容器資料庫 (CDB) 運作。CDB 可以包含顧客建立的插入式資料庫 (PDB)。非 CDB 是使用傳統架構的 Oracle 資料庫，無法包含 PDB。如需多租用戶架構的詳細資訊，請參閱 [Oracle 多租用戶管理員指南](#)。

對於 Oracle Database 19c 和更新版本，您可以建立一個使用 CDB 架構的 RDS for Oracle 資料庫執行個體。您的用戶端應用程式會在 PDB 層級而非 CDB 層級連線。RDS for Oracle 支援 CDB 架構的下列組態：

多租戶組態

此 RDS 平台功能允許 RDS for Oracle CDB 執行個體包含 1—30 個租用戶資料庫之間，具體取決於資料庫版本和任何必要的選項授權租用戶資料庫 (PDB)。多租戶組態不支援應用程式 PDB 或 Proxy PDB。您可以使用 RDS API 來新增、修改和移除租戶資料庫。

Note

Amazon RDS 功能稱為「multi-tenant」而不是「multitenant」，因為它是 RDS 平台的一項功能，而不只是 Oracle 資料庫引擎而已。「Oracle 多租戶」一詞僅指與內部部署和 RDS 部署相容的 Oracle 資料庫架構。

單一租戶組態

此 RDS 平台功能將適用於 Oracle CDB 執行個體的 RDS 限制為 1 個租用戶資料庫 (PDB)。您無法使用 RDS API 新增更多的 PDB。單一租戶組態會使用與非 CDB 架構相同的 RDS API。因此，在單一租戶組態中使用 CDB 的體驗大多與使用非 CDB 相同。

您可以將使用單一租用戶組態的 CDB 轉換為多租用戶組態，以便將 PDB 新增至 CDB。此架構變更是永久且不可復原。如需詳細資訊，請參閱[將單一租戶組態轉換為多租戶](#)。

Note

您無法存取 CDB。

在 Oracle 資料庫 21c 以上版本中，所有資料庫皆為 CDB。反之，您可以將 Oracle 資料庫 19c 資料庫執行個體建立為 CDB 或非 CDB。您無法將非 CDB 升級為 CDB，但可將 Oracle 資料庫 19c 非 CDB 轉換為 CDB，然後再加以升級。您無法將 CDB 轉換為非 CDB。

如需詳細資訊，請參閱下列資源：

- [在 RDS for Oracle 中使用 CDB](#)
- [RDS for Oracle CDB 的限制](#)
- [建立 Amazon RDS 資料庫執行個體](#)

RDS for Oracle 參數

資料庫參數群組

在 Amazon RDS 中，您可以使用資料庫參數群組來管理參數。如需詳細資訊，請參閱[使用參數群組](#)。若要檢視特定 Oracle 資料庫版本和版本支援的初始化參數，請執行 AWS CLI 命令[describe-engine-default-parameters](#)。

例如，若要檢視 Oracle 資料庫 19c 企業版支援的初始化參數，請執行下列命令。

```
aws rds describe-engine-default-parameters \  
  --db-parameter-group-family oracle-ee-19
```

Oracle 資料庫初始化參數

若要尋找初始化參數的說明文件，請參閱 Oracle 資料庫說明文件中的[初始化參數](#)。下列初始化參數有特殊考量：

- ARCHIVE_LAG_TARGET

此參數會在指定的時間過後強制執行重做日誌切換。在適用於甲骨文的 RDS 中，設定 ARCHIVE_LAG_TARGET 300 為，因為復原點目標 (RPO) 為 5 分鐘。為了實現此目標，Oracle 版 RDS 每 5 分鐘切換一次線上重做日誌，並將其存放在 Amazon S3 儲存貯體中。如果記錄切換器的頻率造成 RDS for Oracle 資料庫的效能問題，您可以將資料庫執行個體和儲存擴展到 IOPS 和輸送量較高的執行個體和儲存體。或者，如果您使用適用於 Oracle 的 RDS 自訂或在 Amazon EC2 上部署 Oracle 資料庫，則可以調整 ARCHIVE_LAG_TARGET 初始化參數的設定。

RDS for Oracle 字元集

RDS for Oracle 支援兩種類型的字元集：資料庫字元集和國家字元集。

資料庫字元集

Oracle 資料庫字元集用於 CHAR、VARCHAR2 和 CLOB 資料類型。資料庫也會將此字元集用於中繼資料，例如資料表名稱、資料欄名稱，以及 SQL 陳述式。Oracle 資料庫字元集通常稱為資料庫字元集。

當您建立資料庫執行個體時，可設定字元集。建立資料庫之後，即無法變更資料庫字元集。

支援的資料庫字元集

下表列出 Amazon RDS 中支援的 Oracle 資料庫字元集。對於 AWS CLI [create-db-instance](#) 命令的 `--character-set-name` 參數，或 Amazon RDS API [CreateDBInstance](#) 操作的 `CharacterSetName` 參數，您可以使用此資料表中的值。

Note

CDB 的字元集始終是 AL32UTF8。您只能為 PDB 設定不同的字元集。

Value	描述
AL32UTF8	Unicode 5.0 UTF-8 通用字元集 (預設值)

Value	描述
AR8ISO8859P6	ISO 8859-6 拉丁/阿拉伯文
AR8MSWIN1256	Microsoft Windows 字碼頁 1256 8 位元拉丁/阿拉伯文
BLT8ISO8859P13	ISO 8859-13 波羅的海文
BLT8MSWIN1257	Microsoft Windows 字碼頁 1257 8 位元波羅的海文
CL8ISO8859P5	ISO 8859-5 拉丁/斯拉夫文
CL8MSWIN1251	Microsoft Windows 字碼頁 1251 8 位元拉丁/斯拉夫文
EE8ISO8859P2	ISO 8859-2 東歐
EL8ISO8859P7	ISO 8859-7 拉丁/希臘文
EE8MSWIN1250	Microsoft Windows 字碼頁 1250 8 位元東歐
EL8MSWIN1253	Microsoft Windows 字碼頁 1253 8 位元拉丁/希臘文
IW8ISO8859P8	ISO 8859-8 拉丁/希伯來文
IW8MSWIN1255	Microsoft Windows 字碼頁 1255 8 位元拉丁/希伯來文
JA16EUC	EUC 24 位元日文
JA16EUCTILDE	同於 JA16EUC，差別在於往返 Unicode 對應波浪線和波狀符號
JA16SJIS	Shift-JIS 16 位元日文
JA16SJISTILDE	同於 JA16SJIS，差別在於往返 Unicode 對應波浪線和波狀符號

Value	描述
KO16MSWIN949	Microsoft Windows 字碼頁 949 韓文
NE8ISO8859P10	ISO 8859-10 北歐
NEE8ISO8859P4	ISO 8859-4 北歐和東北歐
TH8TISASCII	泰文工業標準 620-2533-ASCII 8 位元
TR8MSWIN1254	Microsoft Windows 字碼頁 1254 8 位元土耳其文
US7ASCII	ASCII 7 位元美語
UTF8	Unicode 3.0 UTF-8 通用字元集，CESU-8 相容
VN8MSWIN1258	Microsoft Windows 字碼頁 1258 8 位元越南文
WE8ISO8859P1	西歐 8 位元 ISO 8859 第 1 部
WE8ISO8859P15	ISO 8859-15 西歐
WE8ISO8859P9	ISO 8859-9 西歐和土耳其文
WE8MSWIN1252	Microsoft Windows 字碼頁 1252 8 位元西歐
ZHS16GBK	GBK 16 位元簡體中文
ZHT16HKSCS	Microsoft Windows 字碼頁 950，含香港增補字元集 HKSCS-2001。字元集轉換根據 Unicode 3.0。
ZHT16MSWIN950	Microsoft Windows 字碼頁 950 繁體中文
ZHT32EUC	EUC 32 位元繁體中文

NLS_LANG 環境變數

地區設定是處理語言和文化需求的一組資訊，這些資訊對應至特定語言和國家。需要指定 Oracle 的地區設定行為時，最簡單的方法是在用戶端的環境中設定 NLS_LANG 環境變數。此變數設定用戶端應用

程式和資料庫伺服器所使用的語言和地區。此參數也指出用戶端的字元集，對應於用戶端應用程式所輸入或顯示的字元集。如需 NLS_LANG 和字元集的詳細資訊，請參閱 Oracle 文件中的[什麼是字元集或字碼頁？](#)。

NLS 初始化參數

對於 Amazon RDS 中的 Oracle 資料庫執行個體，您也可以執行個體層級上設定下列國家語言支援 (NLS) 初始化參數：

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

如需修改執行個體參數的相關資訊，請參閱[使用參數群組](#)。

您可以在 SQL 用戶端設定其他 NLS 初始化參數。例如，下列陳述式在連接 Oracle 資料庫執行個體的 SQL 用戶端，將 NLS_LANGUAGE 初始化參數設為 GERMAN：

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

如需從 SQL 用戶端連接至 Oracle 資料庫執行個體的相關資訊，請參閱[連線至 RDS for Oracle 資料庫執行個體](#)。

國家字元集

國家字元集用於 NCHAR、NVARCHAR2 和 NLOB 資料類型。國家字元集通常稱為 NCHAR 字元集。與資料庫字元集不同，NCHAR 字元集不會影響資料庫中繼資料。

NCHAR 字元集支援下列字元集：

- AL16UTF16 (預設值)
- UTF8

您可以指定 [create-db-instance](#) 命令 (僅限 `--nchar-character-set-name` 版本 2) 的任一值與 AWS CLI 參數。如果您使用 Amazon RDS API，請指定 [CreateDBInstance](#) 操作的 `NcharCharacterSetName` 參數。建立資料庫之後，您無法變更國家字元集。

如需有關 Unicode in Oracle databases 的詳細資訊，請參閱 Oracle 文件中的 [使用 unicode 支援多語言資料庫](#)。

RDS for Oracle 限制

在下列章節中，你可以找到使用 RDS for Oracle 的重要限制。如需 CDB 的特定限制，請參閱 [RDS for Oracle CDB 的限制](#)。

Note

此清單並不詳盡。

主題

- [Amazon RDS 中的 Oracle 檔案大小限制](#)
- [Oracle 提供之結構描述的公有同義詞](#)
- [不支援功能的結構描述](#)
- [Oracle DBA 權限的限制](#)
- [取代 TLS 1.0 和 1.1 Transport Layer Security](#)

Amazon RDS 中的 Oracle 檔案大小限制

RDS Oracle 資料庫執行個體上的單一檔案大小上限為 16 TiB。此限制是由執行個體使用的 ext4 檔案系統所施加的。因此，Oracle 大型檔案資料檔案限制為 16 TiB。如果您嘗試將大型檔案資料表空間中的資料檔案大小調整為超過限制的值，您會收到如下的錯誤。

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Oracle 提供之結構描述的公有同義詞

請不要為 Oracle 提供的結構描述 (包括 SYS、SYSTEM 和 RDSADMIN) 建立或修改公有同義詞。此類動作可能導致核心資料庫元件失效，並影響您資料庫執行個體的可用性。

您可以在自己的結構描述中建立參考物件的公有同義詞。

不支援功能的結構描述

一般而言，Amazon RDS 不會阻止您為不支援的功能建立資料架構。不過，如果您為需要 SYS 權限的 Oracle 功能和元件建立結構描述，則可能會損壞資料說明並影響執行個體可用性。僅使用在 [將選項新增至 Oracle 資料庫執行個體](#) 中可用的支援功能和結構描述。

Oracle DBA 權限的限制

在資料庫中，角色是一組權限，您可以用來授予給使用者或撤銷。Oracle 資料庫使用角色來提供安全性。

預先定義角色 DBA 一般會在 Oracle 資料庫上允許所有管理權限。建立資料庫執行個體時，您的主要使用者帳戶會取得 DBA 權限 (有一些限制)。為了提供受管體驗，RDS for Oracle 資料庫不會針對 DBA 角色提供下列權限：

- ALTER DATABASE
- ALTER SYSTEM
- CREATE ANY DIRECTORY
- DROP ANY DIRECTORY
- GRANT ANY PRIVILEGE
- GRANT ANY ROLE

請將此主要使用者帳戶用於管理任務，例如在資料庫中建立額外的使用者帳戶。您無法使用 SYS、SYSTEM 和其他 Oracle 提供的管理帳戶。

取代 TLS 1.0 和 1.1 Transport Layer Security

Transport Layer Security 通訊協定 1.0 版和 1.1 版 (TLS 1.0 和 TLS 1.1) 已被取代。根據安全最佳實務，Oracle 已停用 TLS 1.0 和 TLS 1.1。為了符合您的安全性需求，RDS for Oracle 強烈建議您改用 TLS 1.2。

連線至 RDS for Oracle 資料庫執行個體

在 Amazon RDS 佈建您的 Oracle 資料庫執行個體之後，您就可以使用任何標準 SQL 用戶端應用程式來登入您的資料庫執行個體。由於 RDS 是受管理的服務，因此您無法以 SYS 或 SYSTEM 的身分登入。如需更多詳細資訊，請參閱 [RDS for Oracle 的使用者和權限](#)。

在本主題中，您將學習如何使用 Oracle SQL Developer 或 SQL*Plus 連線至 RDS for Oracle 資料庫執行個體。如需引導您進行建立和連接至範例資料庫執行個體的程序範例，請參閱 [建立並連線至 Oracle 資料庫執行個體](#)。

主題

- [尋找您的 RDS for Oracle 資料庫執行個體的端點](#)
- [使用 Oracle SQL Developer 連線至您的資料庫執行個體](#)
- [使用 SQL*Plus 連線至您的資料庫執行個體](#)
- [安全群組的考量](#)
- [程序架構的考量](#)
- [對您的 Oracle 資料庫執行個體的連線進行故障診斷](#)
- [使用 sqlnet.ora 參數修改連線屬性](#)

尋找您的 RDS for Oracle 資料庫執行個體的端點

每個 Amazon RDS 資料庫執行個體會有一個端點，並且每個端點有資料庫執行個體的 DNS 名稱和連接埠號碼。若要使用 SQL 用戶端應用程式連接至您的資料庫執行個體，您需要資料庫執行個體的 DNS 名稱和連接埠號碼。

您可以使用 Amazon RDS 主控台或 AWS CLI，藉此尋找資料庫執行個體的端點。

Note

如果您是使用 Kerberos 身分驗證，請參閱 [使用 Kerberos 身分驗證連接至 Oracle](#)。

主控台

使用主控台尋找端點

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在主控台的右上角，選擇資料庫執行個體的 AWS 區域。
3. 尋找您的資料庫執行個體的 DNS 名稱和連接埠號碼。
 - a. 選擇 Databases (資料庫) 來顯示資料庫執行個體的清單。
 - b. 選擇 Oracle 資料庫執行個體名稱以顯示執行個體詳細資訊。
 - c. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連接至資料庫執行個體。

The screenshot displays the AWS Management Console interface for an Oracle RDS instance. The instance name is 'database-test1'. The 'Connectivity & security' tab is active, showing the following details:

Endpoint & port	Networking	Security
Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups rds-ec2-1 (sg-0a1234567b8cd9e01)
Port 1521	VPC vpc-1a2c3c4d	Active default (sg-0a1bcd2e)

AWS CLI

若要使用 AWS CLI 來尋找 Oracle 資料庫執行個體的端點，請呼叫 [describe-db-instances](#) 命令。

Example 使用 AWS CLI 尋找端點

```
aws rds describe-db-instances
```

在輸出中搜尋 Endpoint 以尋找資料庫執行個體的 DNS 名稱和連接埠號碼。輸出中的 Address 行包含 DNS 名稱。下列是 JSON 端點輸出的範例。

```
"Endpoint": {
  "HostedZoneId": "Z1PVIF0B656C1W",
  "Port": 3306,
  "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"
},
```

Note

輸出可能包含多個資料庫執行個體的資訊。

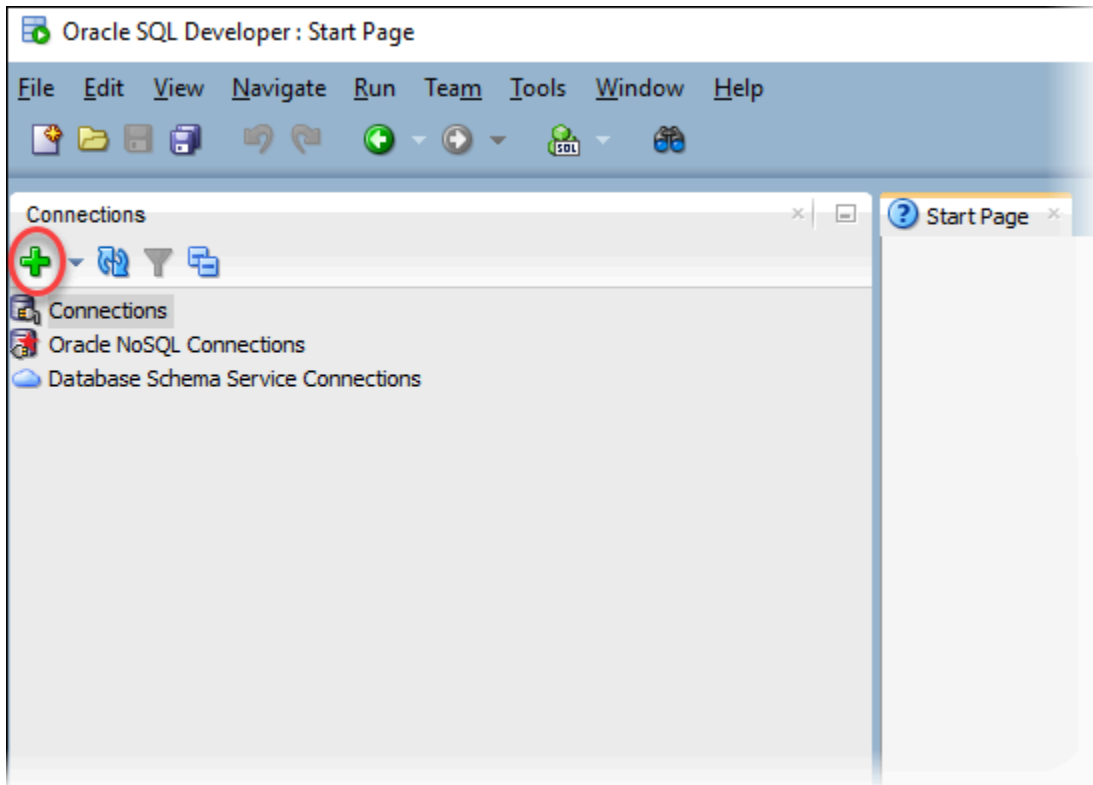
使用 Oracle SQL Developer 連線至您的資料庫執行個體

在此程序中，您會使用 Oracle SQL Developer 連接至資料庫執行個體。若要下載此公用程式的獨立版本，請參閱 [Oracle SQL 開發人員下載頁面](#)。

若要連接至資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找資料庫執行個體的 DNS 名稱和連接埠號碼的詳細資訊，請參閱 [尋找您的 RDS for Oracle 資料庫執行個體的端點](#)。

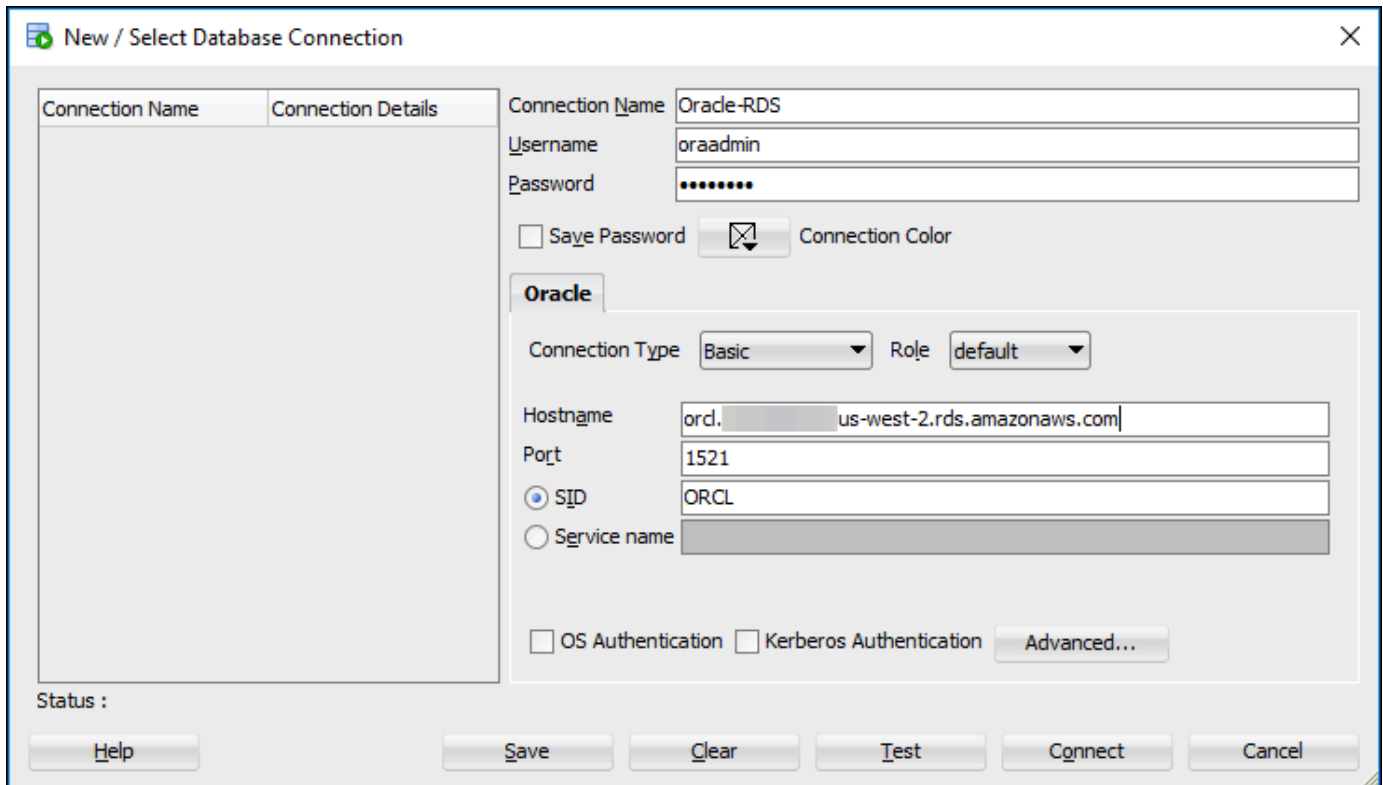
使用 SQL Developer 連線至資料庫執行個體

1. 啟動 Oracle SQL Developer。
2. 在 Connections (連接) 標籤上，選擇 add (+) (新增 (+)) 圖示。



3. 在 New/Select Database Connection (新增/選取資料庫連接) 對話方塊中，提供資料庫執行個體的資訊：
 - 針對 Connection Name (連線名稱)，輸入可描述連線的名稱，例如 Oracle-RDS。
 - 針對 Username (使用者名稱)，輸入資料庫執行個體之資料庫管理員的名稱。
 - 針對 Password (密碼)，輸入資料庫管理員的密碼。
 - 針對 Hostname (主機名稱)，輸入資料庫執行個體的 DNS 名稱。
 - 針對 Port (連接埠)，輸入連接埠號碼。
 - 對於 SID，輸入資料庫名稱。您可以在資料庫詳細資訊頁面的 Configuration (組態) 索引標籤上尋找資料庫名稱。

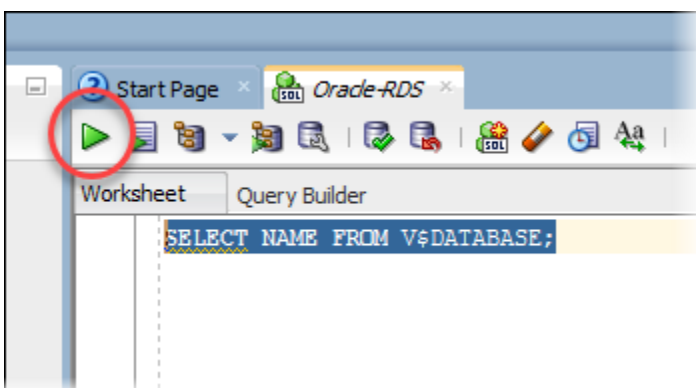
完成的對話方塊看起來應與下列類似。



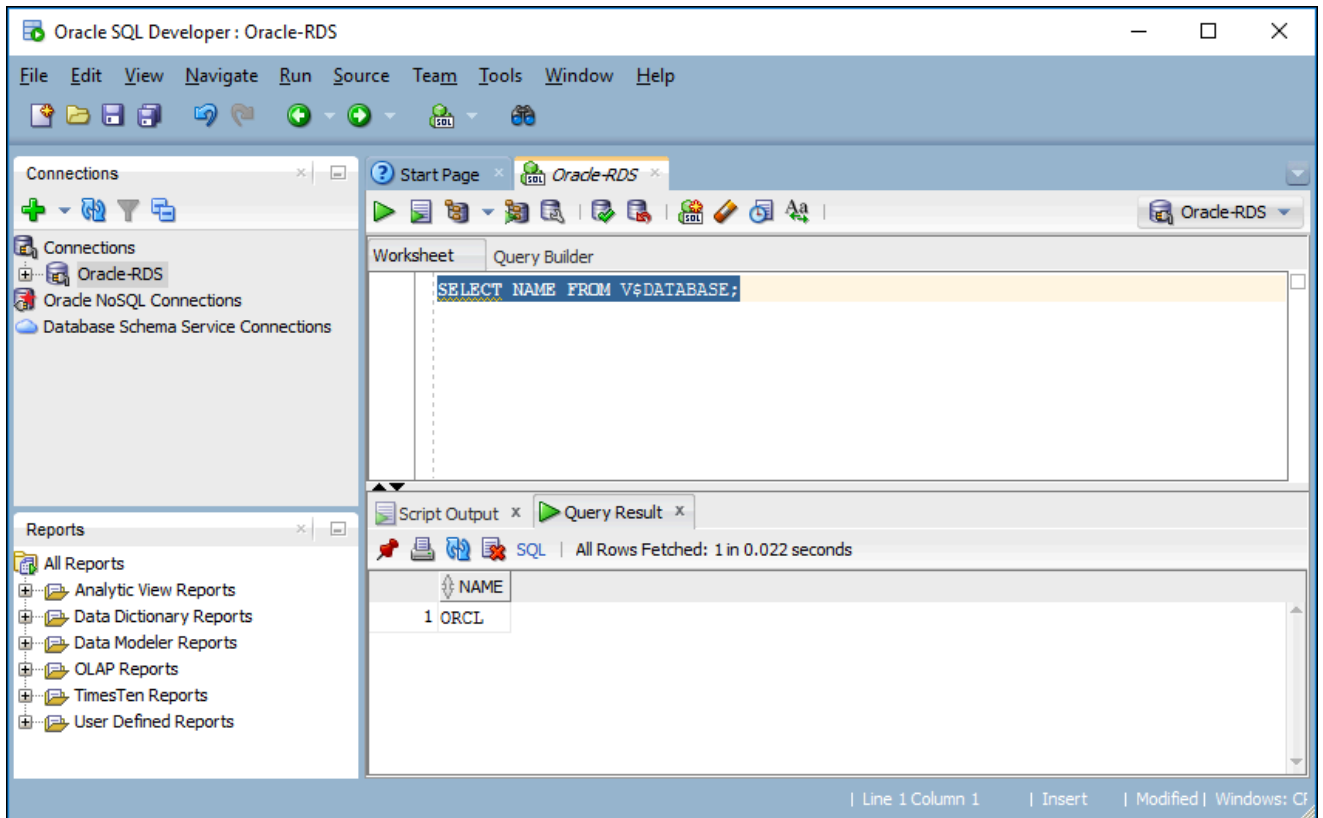
4. 選擇 Connect (連線)。
5. 您現在可以開始建立您自己的資料庫，並照常對您的資料庫執行個體和資料庫執行查詢。若要對您的資料庫執行個體執行測試查詢，請執行下列動作：
 - a. 在連線的 Worksheet (工作表) 標籤中，輸入下列 SQL 查詢。

```
SELECT NAME FROM V$DATABASE;
```

- b. 選擇 execute (執行) 圖示來執行查詢。



SQL Developer 會傳回資料庫名稱。



使用 SQL*Plus 連線至您的資料庫執行個體

您可以使用 SQL*Plus 之類的公用程式來連接至執行 Oracle 的 Amazon RDS 資料庫執行個體。若要下載包含獨立版 SQL*Plus 的 Oracle 即時用戶端，請參閱 [Oracle 即時用戶端下載](#)。

若要連接至資料庫執行個體，您需要其 DNS 名稱和連接埠號碼。如需尋找資料庫執行個體的 DNS 名稱和連接埠號碼的詳細資訊，請參閱 [尋找您的 RDS for Oracle 資料庫執行個體的端點](#)。

Example 使用 SQL*Plus 連接至 Oracle 資料庫執行個體

在下列範例中，替代資料庫執行個體管理員的使用者名稱。同時，替代資料庫執行個體的 DNS 名稱，然後包含連接埠號碼和 Oracle SID。SID 值為建立資料庫執行個體時所指定之資料庫執行個體的資料庫名稱，而不是資料庫執行個體的名稱。

對於LinuxmacOS、或Unix：

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'
```

在Windows中：

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))  
(CONNECT_DATA=(SID=database_name)))
```

您應該會看到類似下列的輸出。

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

輸入使用者的密碼之後，隨即會出現 SQL 提示。

```
SQL>
```

Note

較短格式連線字串 (EZ connect) (例如 `sqlplus USER/PASSWORD@longer-than-63-chars-rds-endpoint-here:1521/database-identifier`) 可能遇到字元數上限，因此建議您不要將其用來連線。

安全群組的考量

為了讓您連接至資料庫執行個體，它必須與安全群組關聯，該群組包含必要的 IP 地址和網路組態。您的資料庫執行個體可能會使用預設安全群組。如果在建立資料庫執行個體時指派預設、非設定的安全群組，防火牆會阻止連線。如需建立新安全群組的詳細資訊，請參閱[使用安全群組控制存取](#)。

建立新安全群組之後，您可以修改資料庫執行個體，讓它與該安全群組建立關聯。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

您可以使用 SSL 來加密對資料庫執行個體的連線，藉此增強安全性。如需詳細資訊，請參閱[Oracle Secure Sockets Layer](#)。

程序架構的考量

伺服器程序會處理對 Oracle 資料庫執行個體的使用者連線。依預設，Oracle 資料庫執行個體會使用專用伺服器程序。利用專用伺服器程序，每個伺服器程序只會服務一個使用者程序。您可以選擇性地設定共用伺服器程序。利用共用伺服器程序，每個伺服器程序可以服務多個使用者程序。

當伺服器上大量的使用者工作階段使用太多記憶體時，您可能會考慮使用共用伺服器程序。當工作階段經常連接和中斷連接，而造成效能問題時，您可能也會考慮共用伺服器程序。使用共用伺服器程序也有缺點。例如，它們可能拖累 CPU 資源，且它們的設定和管理較為複雜。

如需專用和共用伺服器程序的詳細資訊，請參閱 Oracle 文件中的[關於專用和共用伺服器程序](#)。如需在 Oracle 資料庫執行個體上設定共用伺服器程序的詳細資訊，請參閱知識中心的[如何設定 Amazon RDS for Oracle Database 以搭配使用共用伺服器？](#)一文。

對您的 Oracle 資料庫執行個體的連線進行故障診斷

下列是在您嘗試連接至 Oracle 資料庫執行個體時可能遇到的問題。

問題	故障診斷建議
無法連接至資料庫執行個體。	對於新建立的資料庫執行個體，資料庫執行個體的狀態為 creating (建立中)，直到它可供使用為止。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新的資料庫執行個體才會可用。
無法連接至資料庫執行個體。	如果您無法透過建立資料庫執行個體時指定的連接埠傳送或接收通訊，則無法連接至資料庫執行個體。請洽詢您的網路管理員，以驗證您為資料庫執行個體指定的連接埠可允許傳入和傳出通訊。
無法連接至資料庫執行個體。	<p>您的本機防火牆強制執行的存取規則，與您獲授權可存取資料庫執行個體安全群組中資料庫執行個體的 IP 地址可能不符。問題很可能是您防火牆上的傳入或傳出規則引起的。</p> <p>您可以新增或編輯安全群組中的傳入規則。針對 Source (來源)，選擇 My IP (我的 IP)。此舉允許透過您的瀏覽器中偵測到的 IP 地址存取資料庫執行個體。如需更多詳細資訊，請參閱 Amazon VPC 和 Amazon RDS。</p> <p>如需安全群組的詳細資訊，請參閱使用安全群組控制存取。</p> <p>若要逐步進行設定安全群組規則的程序，請參閱 教學課程：建立要與資料庫執行個體搭配使用的 VPC (僅限 IPv4)。</p>
Connect failed because target host or object does not exist – Oracle, Error: ORA-12545 (連接失敗，因為目標主機或物件不存在)	<p>確定您正確指定伺服器名稱和連接埠號碼。針對 Server name (伺服器名稱)，輸入來自主控台的 DNS 名稱。</p> <p>如需尋找資料庫執行個體的 DNS 名稱和連接埠號碼的詳細資訊，請參閱 尋找您的 RDS for Oracle 資料庫執行個體的端點。</p>

問題	故障診斷建議
- Oracle 發生錯誤：ORA-12545)	
Invalid username/ password; logon denied – Oracle, Error: ORA-01017 (無效的使用者名稱/密碼； 登入遭拒 - Oracle 發生錯 誤：ORA-01017)	您可以連接資料庫執行個體，但連線被拒。這通常是因為提供了錯誤的使用者名稱或密碼所造成。驗證使用者名稱和密碼，然後重試。
TNS:listener 目前不知道在 連線描述項中指定的 SID - Oracle，錯誤：ORA-1250 5	請確定輸入的 SID 是正確的。SID 與您的資料庫名稱相同。在執行個體之 Databases (資料庫) 頁面的 Configuration (組態) 索引標籤中尋找資料庫名稱。您也可以使用 AWS CLI 尋找資料庫名稱： <pre>aws rds describe-db-instances --query 'DBInstances[*]. [DBInstanceIdentifier,DBName]' --output text</pre>

如需連線問題的詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)。

使用 sqlnet.ora 參數修改連線屬性

sqlnet.ora 檔案包括在 Oracle 資料庫伺服器 and 用戶端上設定 Oracle Net 功能的參數。使用 sqlnet.ora 檔案中的參數，您可以修改進出資料庫的連線屬性。

如需為何您可能設定 sqlnet.ora 參數的詳細資訊，請參閱 Oracle 文件中的 [設定描述檔參數](#)。

設定 sqlnet.ora 參數

Amazon RDS for Oracle 參數群組包括 sqlnet.ora 參數的子集。您可以使用與您設定其他 Oracle 參數相同的方式來設定它們。sqlnetora. 字首識別哪些參數是 sqlnet.ora 參數。例如，在 Amazon RDS 的 Oracle 參數群組中，default_sdu_size sqlnet.ora 參數是 sqlnetora.default_sdu_size。

如需管理參數群組和設定參數值的相關資訊，請參閱 [使用參數群組](#)。

支援的 sqlnet.ora 參數

Amazon RDS 支援下列 sqlnet.ora 參數。動態 sqlnet.ora 參數的變更會立即生效。

參數	有效值	靜態/ 動態	描述
<code>sqlnetora.default_sdu_size</code>	512 至 209715	動態	工作階段資料單位 (SDU) 大小，以位元組為單位。 SDU 是一次放入緩衝區並跨網路傳送的資料數量。
<code>sqlnetora.diag_adr_enabled</code>	ON, OFF	動態	啟用或停用自動診斷儲存庫 (ADR) 追蹤。 ON 指定使用 ADR 檔案追蹤。 OFF 指定使用非 ADR 檔案追蹤。
<code>sqlnetora.recv_buf_size</code>	8192 至 268435	動態	工作階段之接收操作的緩衝區空間限制，受 TCP/IP、TCP/IP (含 SSL) 和 SDP 通訊協定支援。
<code>sqlnetora.send_buf_size</code>	8192 至 268435	動態	工作階段之傳送操作的緩衝區空間限制，受 TCP/IP、TCP/IP (含 SSL) 和 SDP 通訊協定支援。
<code>sqlnetora.sqlnet.allowed_login_version_client</code>	8, 10, 11, 12	動態	為了讓用戶端及扮演用戶端的伺服器對 Oracle 資料庫執行個體建立連線，所允許的最低驗證通訊協定版本。
<code>sqlnetora.sqlnet.allowed_login_version_server</code>	8, 9, 10, 11, 12, 12a	動態	允許對 Oracle 資料庫執行個體建立連線的最低驗證通訊協定版本。

參數	有效值	靜態/ 動態	描述
<code>sqlnetora.sqlnet.expire_time</code>	0 至 1440	動態	傳送檢查以驗證用戶端-伺服器連線是否作用中的時間間隔 (以分鐘為單位)。
<code>sqlnetora.sqlnet.inbound_connect_timeout</code>	0 或 10 到 7200	動態	用戶端與資料庫伺服器連線，並提供必要身分驗證資訊的時間 (以秒為單位)。
<code>sqlnetora.sqlnet.outbound_connect_timeout</code>	0 或 10 到 7200	動態	用戶端建立 Oracle Net 與資料庫執行個體連線的時間 (以秒為單位)。
<code>sqlnetora.sqlnet.recv_timeout</code>	0 或 10 到 7200	動態	在建立連線之後，資料庫伺服器等待用戶端資料的時間 (以秒為單位)。
<code>sqlnetora.sqlnet.send_timeout</code>	0 或 10 到 7200	動態	在建立連線之後，資料庫伺服器對用戶端完成傳送操作的時間 (以秒為單位)。
<code>sqlnetora.tcp.connect_timeout</code>	0 或 10 到 7200	動態	用戶端建立資料庫伺服器之 TCP 連線的時間 (以秒為單位)。
<code>sqlnetora.trace_level_server</code>	0, 4, 10, 16, OFF, USER, ADMIN, SUPPOF	動態	對於非 ADR 追蹤，在指定的間隔開啟伺服器追蹤或關閉它。

每個支援 `sqlnet.ora` 參數的預設值，都是該版本的「Oracle 資料庫」預設值。

檢視 sqlnet.ora 參數

您可以使用 AWS Management Console、或 SQL 從屬端來檢視 sqlnet.ora 參數及其設定值。AWS CLI

使用主控台檢視 sqlnet.ora 參數

如需檢視參數群組中參數的相關資訊，請參閱[使用參數群組](#)。

在 Oracle 參數群組中，sqlnetora. 字首識別哪些參數是 sqlnet.ora 參數。

使用 AWS CLI 檢視 sqlnet.ora 參數

[若要檢視 Oracle 參數群組中設定的 sqlnet.ora 參數，請使用描述-db-參數命令。AWS CLI](#)

[若要檢視 Oracle 資料庫執行個體的所有 sqlnet.ora 參數，請呼叫下載 db-log 檔案部分命令。AWS CLI](#)指定資料庫執行個體識別符、日誌檔案名稱和輸出類型。

Example

下列代碼列出 mydbinstance 的所有 sqlnet.ora 參數。

對於LinuxmacOS、或Unix：

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier mydbinstance \  
  --log-file-name trace/sqlnet-parameters \  
  --output text
```

在 Windows 中：

```
aws rds download-db-log-file-portion ^  
  --db-instance-identifier mydbinstance ^  
  --log-file-name trace/sqlnet-parameters ^  
  --output text
```

使用 SQL 用戶端檢視 sqlnet.ora 參數

在您連接至 SQL 用戶端的 Oracle 資料庫執行個體之後，下列查詢會列出 sqlnet.ora 參數。

```
SELECT * FROM TABLE  
  (rdsadmin.rds_file_util.read_text_file(  
    'rdsadmin:/rdsadmin/sqlnet.ora')
```



```
p_directory => 'BDUMP',  
p_filename  => 'sqlnet-parameters')));
```

如需連接至 SQL 用戶端中 Oracle 資料庫執行個體的相關資訊，請參閱[連線至 RDS for Oracle 資料庫執行個體](#)。

保護 Oracle 資料庫執行個體連線的安全

Amazon RDS for Oracle 支援 SSL/TLS 加密的連線，以及 Oracle 原生網路加密 (NNE) 選項，以加密應用程式與 Oracle 資料庫執行個體之間的連線。如需 Oracle 原生網路加密選項的詳細資訊，請參閱 [Oracle 原生網路加密](#)。

主題

- [使用 SSL 搭配 RDS for Oracle 資料庫執行個體](#)
- [更新應用程式使用新的 SSL/TLS 憑證來連線至 Oracle 資料庫執行個體](#)
- [使用原生網路加密搭配 RDS for Oracle 資料庫執行個體](#)
- [為 Amazon RDS for Oracle 設定 Kerberos 身分驗證](#)
- [使用憑證和 Oracle 錢包設定 UTL_HTTP 存取](#)

使用 SSL 搭配 RDS for Oracle 資料庫執行個體

Secure Sockets Layer (SSL) 是一種產業標準通訊協定，可保護用戶端與伺服器之間的網路連線。SSL 3.0 版之後的名稱已變更為 Transport Layer Security (TLS)，但我們通常仍會將通訊協定稱為 SSL。Amazon RDS 可支援在 Oracle 資料庫執行個體上啟用 SSL 加密。使用 SSL，您可以加密應用程式用戶端與 Oracle 資料庫執行個體之間的連接。所有 AWS 區域均支援 Oracle 使用 SSL。

在與資料庫執行個體相關聯的選項群組中新增 Oracle SSL 選項，就能對 Oracle 資料庫執行個體啟用 SSL 加密。Amazon RDS 會使用第二個連接埠 (依 Oracle 要求) 作為 SSL 連線用途。如此可讓資料庫執行個體與 Oracle 用戶端之間同時進行純文字和 SSL 加密的通訊。例如，您可以使用連接埠搭配純文字通訊與 VPC 內的其他資源通訊，同時使用連接埠搭配 SSL 加密的通訊與 VPC 外部的資源通訊。

如需更多詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。

Note

您不能在相同的資料庫執行個體上同時使用 SSL 和 Oracle 原生網路加密 (NNE)。在您可以使用 SSL 加密之前，必須停用其他任何連線加密。

更新應用程式使用新的 SSL/TLS 憑證來連線至 Oracle 資料庫執行個體

自 2023 年 1 月 13 日起，Amazon RDS 已發佈新的憑證認證機構 (CA) 憑證，使用 Secure Socket Layer 或 Transport Layer Security (SSL/TLS) 來連線至 RDS 資料庫執行個體。接下來，您可以找到更新應用程式使用新憑證的相關資訊。

本主題可協助您判斷任何用戶端應用程式是否使用 SSL/TLS 連線至您的資料庫執行個體。

Important

當您變更 Amazon RDS for Oracle 資料庫執行個體的憑證時，只會重新啟動資料庫接聽程式。資料庫執行個體不會重新啟動。現有的資料庫連線不會受到影響，但是新的連線將會在重新啟動接聽程式時於短時間內出現錯誤。

Note

對於使用 SSL/TLS 來連線至資料庫執行個體的用戶端應用程式，您必須更新用戶端應用程式信任存放區來包含新的 CA 憑證。

更新用戶端應用程式信任存放區中的 CA 憑證之後，您就可以在資料庫執行個體輪換憑證。強烈建議先在開發或預備環境中測試這些步驟，再於生產環境中實作。

如需憑證輪換的詳細資訊，請參閱[輪換您的 SSL/TLS 憑證](#)。如需下載憑證的詳細資訊，請參閱。如需對 Oracle 資料庫執行個體使用 SSL/TLS 的資訊，請參閱[Oracle Secure Sockets Layer](#)。

主題

- [找出應用程式是否使用 SSL 進行連線](#)
- [更新應用程式信任存放區](#)
- [建立 SSL 連線的 Java 程式碼範例](#)

找出應用程式是否使用 SSL 進行連線

如果 Oracle 資料庫執行個體使用的選項群組已加上 SSL 選項，表示您可能在使用 SSL。遵循[列出選項群組的選項和選項設定](#)中的指示，檢查是否如此。如需 SSL 選項的資訊，請參閱 [Oracle Secure Sockets Layer](#)。

檢查接聽程式日誌，判斷是否有 SSL 連線。下列為接聽程式日誌的輸出範例。

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcps)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

當某個項目的 PROTOCOL 是 *tcps* 值時，即表示 SSL 連線。不過，當 HOST 為 127.0.0.1 時，您可以忽略該項目。來自 127.0.0.1 的連線是資料庫執行個體上的本機管理代理程式。這些連線不是外部 SSL 連線。因此，如果您看到接聽程式日誌項目的 PROTOCOL 是 *tcps*，而 HOST 不是 127.0.0.1，表示有應用程式使用 SSL 來連線。

若要檢查接聽程式日誌，您可以將日誌發佈到 Amazon CloudWatch Logs。如需更多詳細資訊，請參閱 [將 Oracle 日誌發佈到 Amazon CloudWatch 日誌](#)。

更新應用程式信任存放區

您可以為使用 SQL*Plus 或 JDBC 建立 SSL/TLS 連線的應用程式更新信任存放區。

為 SQL*Plus 更新應用程式信任存放區

您可以為使用 SQL*Plus 建立 SSL/TLS 連線的應用程式更新信任存放區。

Note

更新信任存放區時，除了新增憑證，您還可以保留舊憑證。

為 SQL*Plus 應用程式更新信任存放區

1. 下載適用於所有 AWS 區域的新根憑證，並將檔案放入 `ssl_wallet` 目錄。

如需下載根憑證的資訊，請參閱。

2. 執行下列命令來更新 Oracle 錢包。

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
$ORACLE_HOME/ssl_wallet/ssl-cert.pem -auto_login_only
```

將檔案名稱換成您下載的檔案名稱。

3. 執行下列命令，以確認已成功更新錢包。

```
prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
```

輸出應包含以下內容。

```
Trusted Certificates:
Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,
Inc.,L=Seattle,ST=Washington,C=US
```

為 JDBC 更新應用程式信任存放區

您可以為使用 JDBC 建立 SSL/TLS 連線的應用程式更新信任存放區。

如需下載根憑證的資訊，請參閱。

如需匯入憑證的範例指令碼，請參閱 [將憑證匯入信任存放區的範例指令碼](#)。

建立 SSL 連線的 Java 程式碼範例

下列程式碼範例示範如何使用 JDBC 來設定 SSL 連線。

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-
group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
            "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
```

```
        DB_SERVER_NAME, SSL_PORT, DB_SID);
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
properties.put("oracle.jdbc.J2EE13Compliant", "true");
properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
properties.put("javax.net.ssl.trustStoreType", "JKS");
properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
final Connection connection = DriverManager.getConnection(connectionString,
properties);
    // If no exception, that means handshake has passed, and an SSL connection can
    be opened
    }
}
```

Important

在確定了資料庫連線使用 SSL/TLS 並已更新應用程式信任存放區之後，您可以將資料庫更新為使用 rds-ca-rsa2048-g1 憑證。如需說明，請參閱[透過修改資料庫執行個體或叢集來更新 CA 憑證](#)中的步驟 3。

使用原生網路加密搭配 RDS for Oracle 資料庫執行個體

Oracle Database 提供兩種透過網路加密資料的方式：原生網路加密 (NNE) 和 Transport Layer Security (TLS)。NNE 是專屬 Oracle 安全性能，而 TLS 是業界標準。RDS for Oracle 支援 Oracle Database 所有版本的 NNE。

NNE 具有超過 TLS 的下列優點：

- 您可以使用 NNE 選項中的設定來控制用戶端和伺服器上的 NNE：
 - SQLNET.ALLOW_WEAK_CRYPTOClients 和 SQLNET.ALLOW_WEAK_CRYPTO
 - SQLNET.CRYPTO_CHECKSUM_CLIENT 和 SQLNET.CRYPTO_CHECKSUM_SERVER
 - SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT 和 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
 - SQLNET.ENCRYPTION_CLIENT 和 SQLNET.ENCRYPTION_SERVER
 - SQLNET.ENCRYPTION_TYPES_CLIENT 和 SQLNET.ENCRYPTION_TYPES_SERVER
- 在大多數情況下，您不需要設定用戶端或伺服器。相反地，TLS 要求您同時設定用戶端和伺服器。

- 不需要憑證。在 TLS 中，伺服器需要憑證 (其最終會過期)，而用戶端需要發出伺服器憑證之憑證授權單位提供的受信任根憑證。

若要針對 Oracle 資料庫執行個體啟用 NNE 加密，請在與資料庫執行個體相關聯的選項群組中新增 Oracle NNE 選項，。如需詳細資訊，請參閱[Oracle 原生網路加密](#)。

Note

您無法在相同的資料庫執行個體上同時使用 NNE 和 TLS。

為 Amazon RDS for Oracle 設定 Kerberos 身分驗證

您可以使用 Kerberos 身分驗證，在使用者連線至 Amazon RDS for Oracle 資料庫執行個體時驗證使用者身分。在此組態中，您的資料庫執行個體可搭配 AWS Directory Service for Microsoft Active Directory 使用，也稱為 AWS Managed Microsoft AD。當使用者向加入信任網域的 RDS for Oracle 資料庫執行個體進行驗證時，身分驗證請求會轉送到您使用 AWS Directory Service 建立的目錄。

將您的所有登入資料保留在相同目錄可以節省您的時間和精力。您有一個集中位置，用於存放及管理多個資料庫執行個體的登入資料。目錄也可以改善您的整體安全性描述檔。

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域 而有所不同。如需有關 RDS for Oracle 搭配 Kerberos 驗證的版本和區域可用性的詳細資訊，請參閱 [Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎](#)。

Note

RDS for Oracle 資料庫執行個體棄用的資料庫執行個體類別不支援 Kerberos 身分驗證。如需詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。

主題

- [針對 Oracle 資料庫執行個體設定 Kerberos 身分驗證](#)
- [管理網域中的資料庫執行個體](#)
- [使用 Kerberos 身份驗證連接至 Oracle](#)

針對 Oracle 資料庫執行個體設定 Kerberos 身分驗證

使用 AWS Directory Service for Microsoft Active Directory(也稱為 AWS Managed Microsoft AD) 為 Oracle 資料庫執行個體設定 Kerberos 驗證。若要設定 Kerberos 身分驗證，請完成下列步驟：

- [步驟 1：使用 AWS Managed Microsoft AD](#)
- [步驟 2：建立信任](#)
- [步驟 3：設定 Amazon RDS 的 IAM 許可](#)
- [步驟 4：建立和設定使用者](#)
- [步驟 5：啟用目錄和資料庫執行個體之間的跨 VPC 流量](#)
- [步驟 6：建立或修改 Oracle 資料庫執行個體](#)
- [步驟 7：建立 Kerberos 身分驗證 Oracle 登入](#)
- [步驟 8：設定 Oracle 用戶端](#)

Note

在安裝期間，RDS 會建立名為 *managed_service_user@example.com* 且擁有 CREATE SESSION 權限的 Oracle 資料庫使用者，其中 *example.com* 是您的網域名稱。此使用者對應 Directory Service 在受管 Active Directory 內建立的使用者。RDS 會定期使用 Directory Service 提供的登入資料登入 Oracle 資料庫。之後，RDS 會立即銷毀票證快取。

步驟 1：使用 AWS Managed Microsoft AD

AWS Directory Service 在 AWS 雲端中建立完全受管理的作用中目錄。當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會代表您建立兩個網域控制站和網域名稱系統 (DNS) 伺服器。目錄伺服器是在 VPC 的不同子網路中建立。此備援有助於確保即使發生故障，您仍然可以存取目錄。

建立目 AWS Managed Microsoft AD 錄時，請代表您 AWS Directory Service 執行下列工作：

- 在 VPC 內設定 Active Directory。
- 建立含有使用者名稱 Admin 與指定密碼的目錄管理員帳戶。您可以使用此帳戶來管理目錄。

Note

請務必儲存此密碼。AWS Directory Service 不儲存它。您可以重設此密碼，但是無法擷取此密碼。

- 建立目錄控制器的安全群組。

啟動時 AWS Managed Microsoft AD，AWS 會建立包含所有目錄物件的組織單位 (OU)。此 OU 有您在建立目錄時所輸入的 NetBIOS 名稱，且位於根網域中。網域根目錄擁有及管理 AWS。

使用您 AWS Managed Microsoft AD 目錄建立的 Admin 帳戶具有 OU 最常見系統管理活動的權限：

- 建立、更新或刪除使用者
- 將資源 (例如檔案或列印伺服器) 新增至您的網域，然後對您 OU 中的使用者指派這些資源的許可
- 建立額外的 OU 和容器
- 委派授權
- 從 Active Directory 資源回收筒還原已刪除的物件
- 在活動目錄 Web 服務上運行 AD 和 DNS 視窗 PowerShell 模塊

管理員帳戶也有權執行下列全網域活動：

- 管理 DNS 組態 (新增、移除或更新記錄、區域和轉寄站)
- 檢視 DNS 事件日誌
- 檢視安全事件日誌

若要建立目錄 AWS Management Console，請使用 AWS CLI、或 AWS Directory Service API。請務必開啟目錄安全性群組上的相關輸出連接埠，以便目錄可以與 Oracle 資料庫執行個體通訊。

若要建立目錄 AWS Managed Microsoft AD

1. 請登入 AWS Management Console 並開啟 AWS Directory Service 主控台，網址為 <https://console.aws.amazon.com/directoryservicev2/>。
2. 在導覽窗格中，選擇 Directories (目錄)，然後選擇 Set up directory (設定目錄)。
3. 選擇 AWS Managed Microsoft AD。AWS Managed Microsoft AD 是您目前可以與 Amazon RDS 一起使用的唯一選項。

4. 輸入下列資訊：

目錄 DNS 名稱

目錄的完全合格名稱，例如 **corp.example.com**。

目錄 NetBIOS 名稱

目錄的簡短名稱，例如：**CORP**。

目錄描述

(選用) 目錄的描述。

管理員密碼

目錄管理員的密碼。目錄建立程序會建立含有使用者名稱 Admin 與這組密碼的管理員帳戶。

目錄管理員密碼不得包含 "admin" 字組。密碼區分大小寫，長度須為 8 至 64 個字元。至少須有一位字元屬於以下四種類型中的三類：

- 小寫字母 (a-z)
- 大寫字母 (A-Z)
- 數字 (0-9)
- 非英數字元 (~!@#%&* _-+=`|\(){}[];'"<>.,?/)

Confirm password (確認密碼)

重新輸入的管理員密碼。

5. 選擇下一步。

6. 在 Networking (聯網) 區段輸入以下資訊，然後選擇 Next (下一步)。

VPC

目錄的 VPC。在相同 VPC 中建立 Oracle 資料庫執行個體。

子網路

目錄伺服器的子網路。這兩個子網路必須位於不同的可用區域。

7. 檢閱目錄資訊，並進行必要的變更。若資訊無誤，請選擇 Create directory (建立目錄)。

Review & create

Review

Directory type Microsoft AD	VPC vpc-8b6b78e9 ([REDACTED])
Directory DNS name corp.example.com	Subnets subnet-75128d10 ([REDACTED] , us-east-1a) subnet-f51665dd ([REDACTED] , us-east-1b)
Directory NetBIOS name CORP	
Directory description My directory	

Pricing

Edition Standard	Free trial eligible Learn more 30-day limited trial
~USD [REDACTED] *	
* Includes two domain controllers, USD [REDACTED] /mo for each additional domain controller.	

Cancel Previous **Create directory**

建立目錄需要幾分鐘的時間。成功建立時，Status (狀態) 值會變更為 Active (作用中)。

如要查看您目錄的資訊，請在目錄清單中選擇目錄名稱。請記下 Directory ID (目錄 ID) 值，因為您在建立或修改 Oracle 資料庫執行個體時將需要這個值。

The screenshot shows the AWS Directory Service console for a directory with ID d-90670a8d36. The 'Directory details' section is displayed, showing various configuration parameters. The 'Directory ID' is highlighted with a red circle.

Directory details		Reset user password	Refresh
Directory type	VPC	Status	
Microsoft AD	vpc-6594f31c	Active	
Edition	Subnets	Last updated	
Standard	subnet-7d36a227 subnet-a2ab49c6	Tuesday, January 7, 2020	
Directory ID	Availability zones	Launch time	
d-90670a8d36	us-east-1c, us-east-1d	Tuesday, January 7, 2020	
Directory DNS name	DNS address		
corp.example.com			
Directory NetBIOS name			
CORP			
Description - Edit			
My directory			

Application management | Scale & share | Networking & security | Maintenance

步驟 2：建立信任

如果您 AWS Managed Microsoft AD 只打算使用，請移至[步驟 3：設定 Amazon RDS 的 IAM 許可](#)。

若要使用現場部署或自行託管的 Microsoft Active Directory 獲得 Kerberos 身分驗證，請建立樹系信任或外部信任。信任可以是單向或雙向。如需使用設定樹系信任的詳細資訊 AWS Directory Service，請參閱《AWS Directory Service 管理指南》中的[建立信任關係的時機](#)。

步驟 3：設定 Amazon RDS 的 IAM 許可

若要撥打電話 AWS Directory Service 給您，Amazon RDS 需要使用受管身分與存取權管理政策的 IAM 角色 AmazonRDSDirectoryServiceAccess。此角色允許 Amazon RDS 呼叫 AWS Directory Service。

Note

若要讓角色允許存取，AWS Security Token Service (AWS STS) 端點必須以正確 AWS 區域的方式啟動 AWS 帳戶。AWS STS 端點預設在所有端點都處於作用中狀態 AWS 區域，您可以使用它們，而無需任何進一步的動作。如需詳細資訊，請參閱《IAM 使用者指南》[AWS STS](#) AWS 區域中的「啟用和停用」。

建立 IAM 角色

當您使用建立資料庫執行個體 AWS Management Console，且主控台使用者具有 iam:CreateRole 權限時，主控台 rds-directoryservice-kerberos-access-role 會自動建立。否則，您必須手動建立 IAM 角色。手動建立 IAM 角色時 Directory Service，請選擇 AWS 受管政策並將其附加 AmazonRDSDirectoryServiceAccess 到該角色。

如需為服務建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以將許可委派給 AWS 服務](#)。

Note

用於 Windows Authentication for RDS for Microsoft SQL Server 的 IAM 角色不可用於 RDS for Oracle。

手動建立 IAM 信任政策

您可以選擇性地建立具有必要許可的資源政策，而不是使用受管 IAM 政策 AmazonRDSDirectoryServiceAccess。同時指定 directoryservice.rds.amazonaws.com 和 rds.amazonaws.com 作為主體。

若要限制 Amazon RDS 為特定資源提供另一項服務的許可，建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵。防範混淆代理人問題的最有效方法是使用 [aws:SourceArn](#) 全域條件內容索引鍵，以及 Amazon RDS 資源的完整 ARN。如需詳細資訊，請參閱 [防止跨服務混淆代理人問題](#)。

下列範例示範如何使用 Amazon RDS 中的 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵，來預防混淆代理人問題。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

對於選擇加入區域，您還必須以的 `directoryservice.rds.region_name.amazonaws.com` 形式包含該區域的服務主體。例如，在非洲 (開普敦) 區域，請使用下列信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "directoryservice.rds.af-south-1.amazonaws.com",
          "rds.amazonaws.com"
        ]
      }
    }
  ]
}
```

```

    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:rds:af-south-1:123456789012:db:mydbinstance"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
]
}

```

此角色也須具有下列 IAM 政策：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}

```

步驟 4：建立和設定使用者

您可以使用 Active Directory 使用者和電腦工具 (其為 Active Directory 網域服務和 Active Directory 輕量型目錄服務工具之一) 來建立使用者。在此情況下，使用者代表具有目錄存取權的個人或實體。

若要在 AWS Directory Service 目錄中建立使用者，您必須連線至屬於 AWS Directory Service 目錄成員的 Windows Amazon EC2 執行個體。同時，您必須以具有建立使用者之許可的使用者身分來登入。如需在 Microsoft Active Directory 建立使用者的詳細資訊，請參閱《AWS Managed Microsoft AD 管理指南》中的 [管理 AWS Directory Service 中的使用者和群組](#)。

步驟 5：啟用目錄和資料庫執行個體之間的跨 VPC 流量

如果您打算在相同 VPC 中尋找目錄和資料庫執行個體，請略過本步驟，並移至[步驟 6：建立或修改 Oracle 資料庫執行個體](#)。

[如果您計劃在不同的 AWS 帳戶或 VPC 中尋找目錄和資料庫執行個體，請使用 VPC 對等或 Transit Gateway 來設定跨 VPC 流量。AWS 下列程序會使用 VPC 互連來啟用 VPC 之間的流量。請遵循《Amazon Virtual Private Cloud 互連指南》中\[什麼是 VPC 互連？\]\(#\)的指示。](#)

使用 VPC 互連以啟用跨 VPC 流量

1. 設定適當的 VPC 路由規則，以確保網路流量可以雙向對流。
2. 確保資料庫執行個體安全群組可以從目錄的安全群組接收傳入流量。若要取得更多資訊，請參閱《AWS Managed Microsoft AD 管理指南》中的[AWS Directory Service 最佳實務](#)。
3. 確保沒有網路存取控制清單 (ACL) 規則來封鎖流量。

如果目錄擁有不同的 AWS 帳戶，您必須共用該目錄。

在 AWS 帳戶之間共用目錄

1. 按照 AWS Directory Service 管理指南中的[教學課程：共用目錄以進行無縫 EC2 網域加入中的說明，開始與將在其中建立資料庫執行個體的 AWS 帳戶共用 AWS Managed Microsoft AD 目錄](#)。
2. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台，並確保網域具有 SHARED 狀態，然後再繼續操作。
3. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台時，請記下目錄 ID 值。您可以使用此目錄 ID，將資料庫執行個體加入網域。

步驟 6：建立或修改 Oracle 資料庫執行個體

建立或修改要搭配您目錄使用的 Oracle 資料庫執行個體。您可以使用主控台、CLI 或 RDS API，將資料庫執行個體與目錄建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) CLI 命令或建立資料庫執行個體 RDS API [作業，建立新的 Oracle 資料庫執行個體](#)。

如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[modify-db-instance](#) CLI 命令或修改資料庫執行個體 RDS API 作業來[修改現有的 Oracle 資料庫執行個體](#)。

如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

- 使用主控台、[restore-db-instance-from-db 快照 CLI 命令](#)或[還原資料庫快照 RDS API 作業](#)，從[資料庫快照](#)還原 Oracle 資料庫執行個體。[InstanceFrom](#)

如需說明，請參閱[從資料庫 快照還原](#)。

- point-in-time 使用主控台、[restore-db-instance-to-point-in-time](#) CLI 命令或恢復 InstanceToPointInTime RDS API 作業，將 Oracle 資料庫執行個體還原至。

如需說明，請參閱「[將資料庫執行個體還原至指定的時間](#)」。

僅 VPC 中的 Oracle 資料庫執行個體支援 Kerberos 身分驗證。資料庫執行個體可在與目錄相同的 VPC 中，或在不同 VPC 中。當您建立或修改資料庫執行個體時，請執行以下動作：

- 請提供您建立目錄時產生的網域識別符 (d- * 識別符)。
- 請提供所建立的 IAM 角色名稱。
- 確保資料庫執行個體安全群組可以接收來自目錄安全群組的傳入流量，並將傳出流量傳送至目錄。

使用主控台建立資料庫執行個體時，請在 Database authentication (資料庫身分驗證) 區段中選擇 Password and Kerberos authentication (密碼和 Kerberos 身分驗證)。選擇 Browse Directory (瀏覽目錄) 並選取目錄，或是選擇 Create a new directory (建立新目錄)。

Database authentication

Database authentication options [Info](#)

- Password authentication
Authenticates using database passwords.
- Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Directory

當您使用主控台修改或還原資料庫執行個體時，請選擇 Kerberos authentication (Kerberos 身分驗證) 區段中的目錄，或選擇 Create a new directory (建立新目錄)。

Kerberos authentication

Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos authentication.

Refresh

Directory

None

Create a new directory

By choosing a directory and continuing with database instance creation you authorize Amazon RDS to create the IAM role necessary for using Kerberos authentication

使用時 AWS CLI，資料庫執行個體需要下列參數才能使用您建立的目錄：

- 針對 `--domain` 參數，使用您建立目錄時產生的網域識別符 ("d-*" 識別符)。
- 針對 `--domain-iam-role-name` 參數，使用您建立的規則，其會使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess`。

例如，下列 CLI 命令會修改資料庫執行個體來使用目錄。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --domain d-ID \  
  --domain-iam-role-name role-name
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --domain d-ID ^  
  --domain-iam-role-name role-name
```

Important

如果您修改資料庫執行個體，以啟用 Kerberos 身分驗證，請在進行變更後重新啟動資料庫執行個體。

Note

MANAGED_SERVICE_USER 是一個服務帳戶，其名稱是由 RDS 的 Directory Service 隨機產生。在 Kerberos 身分驗證設定期間，RDS for Oracle 會建立具有相同名稱的使用者，並為其指派 CREATE SESSION 權限。Oracle 資料庫使用者在外部識別為 *MANAGED_SERVICE_USER@EXAMPLE.COM*，其中 *EXAMPLE.COM* 是您網域的名稱。RDS 會定期使用 Directory Service 提供的登入資料登入 Oracle 資料庫。之後，RDS 會立即銷毀票證快取。

步驟 7：建立 Kerberos 身分驗證 Oracle 登入

如同對任何其他資料庫執行個體一般，使用 Amazon RDS 主要使用者登入資料來連接至 Oracle 資料庫執行個體。資料庫執行個體已加入 AWS Managed Microsoft AD 網域。因此，您可以從網域中的 Microsoft Active Directory 使用者和群組佈建 Oracle 登入和使用者。若要管理資料庫許可，請將標準 Oracle 許可授與這些登入，以及撤銷其許可。

允許 Microsoft Active Directory 使用者與 Oracle 進行身分驗證

1. 使用 Amazon RDS 主要使用者登入資料來連線至 Oracle 資料庫執行個體。
2. 在 Oracle 資料庫中建立外部身分驗證的使用者。

在下列範例中，以使用者名稱和網域名稱取代 *KRBUSER@CORP.EXAMPLE.COM*。

```
CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;  
GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
```

來自您網域的使用者 (人員和應用程式兩者) 現在可以使用 Kerberos 身分驗證從加入網域的用戶端機器連線至 Oracle 資料庫執行個體。

步驟 8：設定 Oracle 用戶端

若要設定 Oracle 用戶端，必須符合下列需求：

- 請建立一個名為 krb5.conf (Linux) 或 krb5.ini (Windows) 的組態檔案，以指向該網域。設定 Oracle 用戶端以使用此組態檔案。

- 確認流量可以在用戶端主機之間流動，以及 AWS Directory Service 透過 TCP/UDP、Kerberos 連接埠 (受管理的是 88 和 464 AWS Directory Service)，以及透過 TCP 透過 LDAP 連接埠 389 在 DNS 連接埠 53 之間流動。
- 確定流量可透過資料庫連接埠在用戶端主機和資料庫執行個體之間往來。

以下是的示例內容 AWS Managed Microsoft AD。

```
[libdefaults]
  default_realm = EXAMPLE.COM
[realms]
  EXAMPLE.COM = {
    kdc = example.com
    admin_server = example.com
  }
[domain_realm]
  .example.com = CORP.EXAMPLE.COM
  example.com = CORP.EXAMPLE.COM
```

以下是適用於內部部署 Microsoft AD 的範例內容。在您的 krb5.conf 或 krb5.ini 檔案中，將名稱取代為內部部署 *on-prem-ad-serverAD #####*稱。

```
[libdefaults]
  default_realm = ONPREM.COM
[realms]
  AWSAD.COM = {
    kdc = awsad.com
    admin_server = awsad.com
  }
  ONPREM.COM = {
    kdc = on-prem-ad-server-name
    admin_server = on-prem-ad-server-name
  }
[domain_realm]
  .awsad.com = AWSAD.COM
  awsad.com= AWSAD.COM
  .onprem.com = ONPREM.COM
  onprem.com= ONPREM.COM
```

Note

在您設定 `krb5.ini` 或 `krb5.conf` 檔案後，我們建議您重新啟動伺服器。

以下是適用於 SQL*Plus 組態的範例 `sqlnet.ora` 內容：

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

如需 SQL Developer 組態的範例，請參閱 Oracle Support 的[文件 1609359.1](#)。

管理網域中的資料庫執行個體

您可以使用主控台、CLI 或 RDS API，來管理資料庫執行個體，以及其與 Microsoft Active Directory 的關係。例如，您可以使 Microsoft Active Directory 產生關聯，以啟用 Kerberos 身份驗證。您也可以將 Microsoft Active Directory 解除關聯，以停用 Kerberos 身份驗證。您可以將要由某個 Microsoft Active Directory 外部識別的資料庫執行個體移至另一個 Microsoft Active Directory。

例如，使用 CLI，您可以執行下列動作：

- 若要對失敗的成員資格重新嘗試啟用 Kerberos 身份驗證，請使用 [modify-db-instance](#) CLI 命令，並為 `--domain` 選項指定目前成員資格的目錄 ID。
- 若要在資料庫執行個體上停用 Kerberos 身份驗證，請使用 [modify-db-instance](#) CLI 命令，並為 `none` 選項指定 `--domain`。
- 若要將資料庫執行個體從某個網域移至另一個網域，請使用 [modify-db-instance](#) CLI 命令，並為 `--domain` 選項指定新網域的網域識別符。

檢視網域成員資格的狀態

在您建立或修改資料庫執行個體之後，此資料庫執行個體會成為網域的成員。您可以在主控台中或執行 [describe-db-instances](#) CLI 命令，來檢視資料庫執行個體的網域成員資格狀態。資料庫執行個體的狀態可以是下列其中一個：

- `kerberos-enabled` – 資料庫執行個體已啟用 Kerberos 身分驗證。
- `enabling-kerberos` – AWS 正在此資料庫執行個體上啟用 Kerberos 身分驗證。
- `pending-enable-kerberos` – 在此資料庫執行個體上擱置 Kerberos 身分驗證的啟用。

- `pending-maintenance-enable-kerberos` – AWS 將在下次排定的維護時段嘗試在資料庫執行個體上啟用 Kerberos 身分驗證。
- `pending-disable-kerberos` – 在此資料庫執行個體上擱置 Kerberos 身分驗證的停用。
- `pending-maintenance-disable-kerberos` – AWS 將在下次排定的維護時段嘗試在資料庫執行個體上停用 Kerberos 身分驗證。
- `enable-kerberos-failed` – 有一個組態問題已禁止 AWS 在資料庫執行個體上啟用 Kerberos 身分驗證。請更正問題，然後重新發出命令來修改資料庫執行個體。
- `disabling-kerberos` – AWS 正在此資料庫執行個體上停用 Kerberos 身分驗證。

由於網路連線問題或 IAM 角色不正確，請求啟用 Kerberos 身分驗證可能失敗。當您建立或修改資料庫執行個體時，如果嘗試啟用 Kerberos 身分驗證失敗，請確定您使用的是正確的 IAM 角色。然後，修改資料庫執行個體來加入網域。

Note

僅在 Kerberos 身分驗證搭配 Amazon RDS for Oracle 時，才會將流量傳送至網域的 DNS 伺服器。所有其他 DNS 都被視為執行 Oracle 之資料庫執行個體上的傳出網路存取。如需傳出網路存取搭配 Amazon RDS for Oracle 的詳細資訊，請參閱[設定自訂 DNS 伺服器](#)。

強制輪換 Kerberos 金鑰

私密金鑰在 AWS Managed Microsoft AD 和 Amazon RDS for Oracle 資料庫執行個體之間共享。此金鑰每 45 天自動輪換。您可以使用以下 Amazon RDS 程序以強制輪換此金鑰：

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM DUAL;
```

Note

在僅供讀取複本組態中，此程序只能在來源資料庫執行個體上使用，無法在僅供讀取複本上使用。

SELECT 陳述式以 VARCHAR2 資料類型傳回任務 ID。您可在 `bdump` 檔案之中，檢視進行中作業的狀態。`bdump` 檔案位在 `/rdsdbdata/log/trace` 目錄：每個 `bdump` 檔案的名稱格式如下。

```
dbtask-task-id.log
```

您可以透過顯示任務的輸出檔案來檢視結果。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

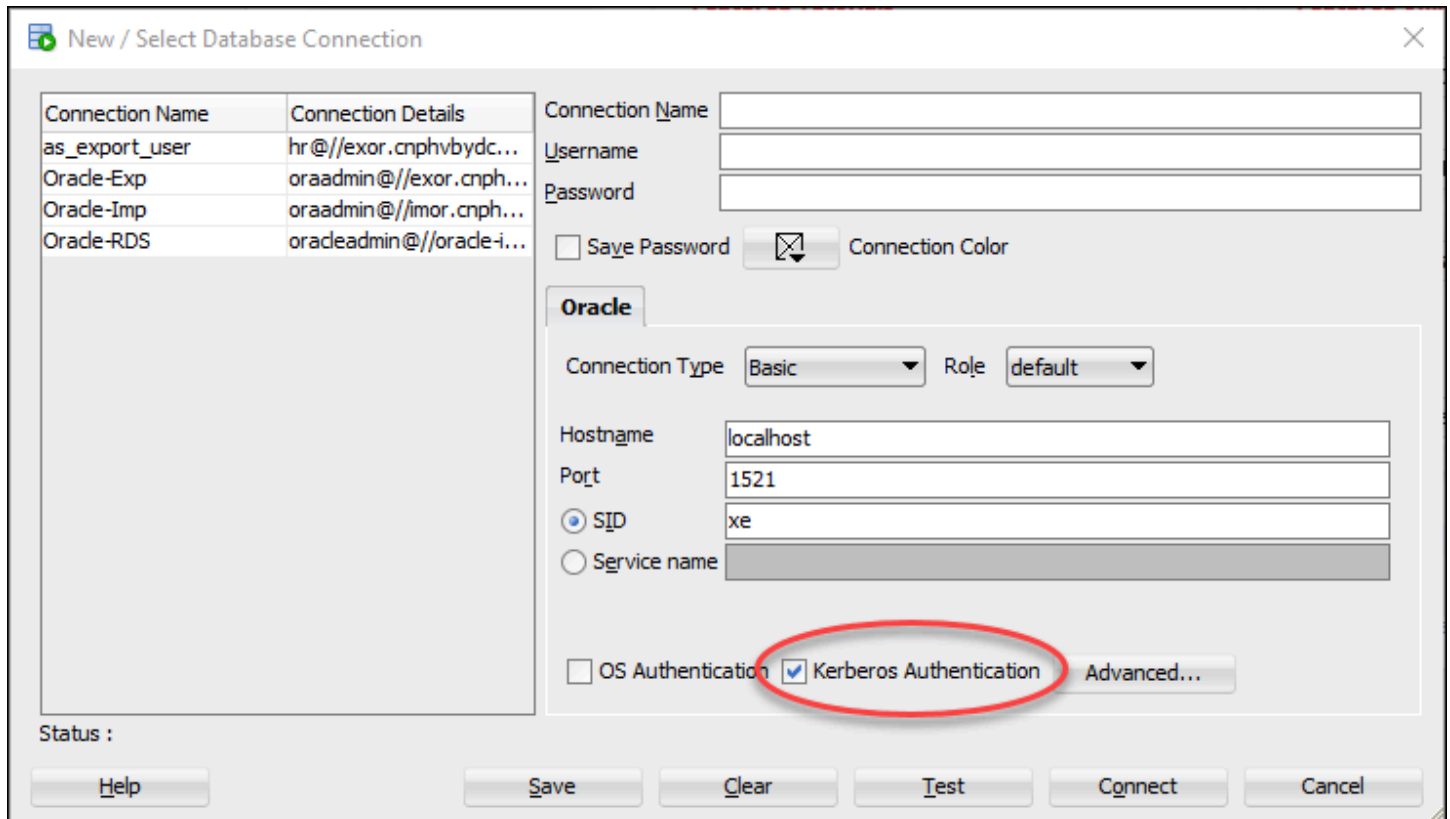
以程序傳回的任務 ID 取代 *task-id*。

Note

任務是以非同步方式執行。

使用 Kerberos 身份驗證連接至 Oracle

本節假設您已設定如[步驟 8：設定 Oracle 用戶端](#)所述 Oracle 用戶端。若要使用 Kerberos 身份驗證連接至 Oracle 資料庫，請使用 Kerberos 身份驗證類型登入。例如，在啟動 Oracle SQL Developer 之後，請選擇 Kerberos Authentication (Kerberos 身份驗證) 做為身份驗證類型，如下所示。



若要透過 SQL*Plus 使用 Kerberos 身份驗證連接至 Oracle：

1. 在命令提示中，執行下列命令：

```
kinit username
```

將 *username* 取代為使用者名稱，並在提示中，輸入 Microsoft Active Directory 中針對使用者存放的密碼。

2. 開啟 SQL*Plus，然後使用 Oracle 資料庫執行個體的 DNS 名稱和連接埠號碼進行連線。

如需在 SQL*Plus 連接至資料庫執行個體的詳細資訊，請參閱[使用 SQL*Plus 連線至您的資料庫執行個體](#)。

使用憑證和 Oracle 錢包設定 UTL_HTTP 存取

Amazon RDS 支援在適用於 Oracle 資料庫執行個體的 RDS 上進行輸出網路存取。若要將資料庫執行個體連線至網路，您可以使用下列 PL/SQL 套件：

UTL_HTTP

此套件會從 SQL 和 PL/SQL 進行 HTTP 呼叫。您可以用其透過 HTTP 存取網際網路上的資料。如需詳細資訊，請參閱 Oracle 文件中的 [UTL_HTTP](#)。

UTL_TCP

此套件會在 PL/SQL 中提供 TCP/IP 用戶端存取功能。此套件對於使用網際網路通訊協定和電子郵件的 PL/SQL 應用程式非常有用。如需詳細資訊，請參閱 Oracle 文件中的 [UTL_TCP](#)。

UTL_SMTP

此套件會提供 SMTP 命令的介面，讓用戶端能將電子郵件傳送至 SMTP 伺服器。如需詳細資訊，請參閱 Oracle 文件中的 [UTL_SMTP](#)。

完成下列任務，就可以設定 UTL_HTTP.REQUEST，搭配執行 SSL 交握期間需要用戶端身分驗證憑證的網站使用。您也可以修改 Oracle 錢包產生命令和 DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE 程序，為網站 UTL_HTTP 存取設定密碼身分驗證。如需詳細資訊，請參閱 Oracle 資料庫說明文件中的 [DBMS_NETWORK_ACL_ADMIN](#)。

Note

您可以針對 UTL_SMTP 調整下列任務，允許透過 SSL/TLS (包括 [Amazon Simple Email Service](#)) 傳送電子郵件。

主題

- [設定 UTL_HTTP 存取時的考量事項](#)
- [步驟 1：取得網站的根憑證](#)
- [步驟 2：建立 Oracle 錢包](#)
- [步驟 3：將 Oracle 錢包下載到 RDS for Oracle 執行個體](#)
- [步驟 4：授予使用者使用 Oracle 錢包的許可](#)
- [步驟 5：設定從資料庫執行個體存取網站的權限](#)
- [步驟 6：測試從資料庫執行個體連至網站的連線](#)

設定 UTL_HTTP 存取時的考量事項

在設定存取權之前，請考慮下列事項：

- 您可以將 SMTP 與郵件選項搭配使用。如需詳細資訊，請參閱 [Oracle UTL_MAIL](#)。
- 遠端主機的網域名稱伺服器 (DNS) 名稱可以是以下任何項目：
 - 可公開解析。
 - Amazon RDS 資料庫執行個體的端點。
 - 可透過自訂 DNS 伺服器解析。如需更多詳細資訊，請參閱 [設定自訂 DNS 伺服器](#)。
 - 相同 VPC 或對等 VPC 中 Amazon EC2 執行個體的私有 DNS 的名稱。在此情況下，請確定名稱可透過自訂 DNS 伺服器解析。或者，若要使用 Amazon 提供的 DNS，您可以在 VPC 設定中啟用 `enableDnsSupport` 屬性，並為 VPC 對等連接啟用 DNS 解析支援。如需詳細資訊，請參閱 [VPC 中的 DNS 支援](#) 和 [修改 VPC 對等連接](#)。
 - 如要安全連線到遠端 SSL/TLS 資來源，建議您建立並上傳自訂的 Oracle 錢包。通過使用 Amazon S3 與 Amazon RDS for Oracle 的功能整合，您可以將 Amazon S3 中的錢包下載到 Oracle 資料庫執行個體中。有關 Oracle Amazon S3 整合的詳細資訊，請參閱 [Amazon S3 整合](#)。
- 若 Oracle SSL 選項已為個別執行個體進行設定，則您可以透過 SSL/TLS 端點來建立 Oracle 資料庫執行個體之間的資料庫連結。無需進一步設定。如需詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。

步驟 1：取得網站的根憑證

若要讓 RDS for Oracle 資料庫執行個體與網站建立安全連線，請新增根 CA 憑證。Amazon RDS 使用根憑證將網站憑證簽入 Oracle 錢包。

您可以透過多種方式取得根憑證。例如，您可以執行下列動作：

1. 使用 Web 伺服器造訪受憑證保護的網站。
2. 下載用於簽署的根憑證。

若使用 AWS 服務，通常可以在 [Amazon Trust Services 儲存庫](#) 中找到這些憑證。

步驟 2：建立 Oracle 錢包

建立同時包含 Web 伺服器憑證和用戶端身分驗證憑證的 Oracle 錢包。RDS Oracle 執行個體使用 Web 伺服器憑證與網站建立安全連線。網站需要用戶端憑證用於驗證 Oracle 資料庫使用者。

您可能希望設定安全連線，而不透過用戶端憑證進行身分驗證。在此情況下，可以略過以下程序中的 Java 金鑰存放區步驟。

建立 Oracle 錢包

1. 將根憑證和用戶端憑證放在單一目錄中，然後變更為此目錄中。
2. 將 .p12 用戶端憑證轉換為 Java 金鑰存放區。

Note

如果您不透過用戶端憑證進行身分驗證，可略過此步驟。

下列範例將名為 *client_certificate.p12* 的用戶端憑證轉換為 *client_keystore.jks* Java 金鑰存放區。這樣金鑰存放區就包含在 Oracle 錢包中。金鑰存放區密碼為 *P12PASSWORD*。

```
orapki wallet pkcs12_to_jks -wallet ./client_certificate.p12 -  
jksKeyStoreLoc ./client_keystore.jks -jksKeyStorepwd P12PASSWORD
```

3. 為 Oracle 錢包建立與憑證目錄不同的目錄。

下列範例會建立 /tmp/wallet 目錄。

```
mkdir -p /tmp/wallet
```

4. 在錢包目錄中建立 Oracle 錢包。

下列範例會將 Oracle 錢包密碼設定為 *P12PASSWORD*，此密碼與上一步中 Java 金鑰存放區使用的密碼相同。使用相同的密碼很方便，但沒有必要。`-auto_login` 參數會開啟自動登入功能，因此不需要每次存取時都指定一次密碼。

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

```
orapki wallet create -wallet /tmp/wallet -pwd P12PASSWORD -auto_login
```

5. 將 Java 金鑰存放區新增到您的 Oracle 錢包。

Note

如果您不透過用戶端憑證進行身分驗證，可略過此步驟。

下列範例會將金鑰存放區 *client_keystore.jks* 新增到名為 */tmp/wallet* 的 Oracle 錢包。在此範例中，為 Java 金鑰存放區和 Oracle 錢包指定了相同的密碼。

```
orapki wallet jks_to_pkcs12 -wallet /tmp/wallet -pwd P12PASSWORD -  
keystore ./client_keystore.jks -jkspwd P12PASSWORD
```

6. 將目標網站的根憑證新增至 Oracle 錢包。

下列範例會新增名為 *Root_CA.cer* 的憑證。

```
orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Root_CA.cer -  
pwd P12PASSWORD
```

7. 新增任何中繼憑證。

下列範例會新增名為 *Intermediate.cer* 的憑證。若要載入所有中繼憑證，請視需要重複此步驟。

```
orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Intermediate.cer -  
pwd P12PASSWORD
```

8. 確認新建立的 Oracle 錢包包含有所需的憑證。

```
orapki wallet display -wallet /tmp/wallet -pwd P12PASSWORD
```

步驟 3：將 Oracle 錢包下載到 RDS for Oracle 執行個體

在此步驟中，將 Oracle 錢包上傳到 Amazon S3，然後將錢包從 Amazon S3 下載到 RDS for Oracle 執行個體。

將 Oracle 錢包下載到 RDS for Oracle 資料庫執行個體

1. 完成 Amazon S3 與 Oracle 整合的必要條件，然後新增 S3_INTEGRATION 選項至您的 Oracle 資料庫執行個體。確保選項的 IAM 角色可存取您使用的 Amazon S3 儲存貯體。

如需詳細資訊，請參閱[Amazon S3 整合](#)。

2. 以主要使用者身分登入資料庫執行個體，然後建立一個 Oracle 目錄來保存 Oracle 錢包。

下列範例會建立名為 *WALLET_DIR* 的 Oracle 目錄。

```
EXEC rdsadmin.rdsadmin_util.create_directory('WALLET_DIR');
```

如需詳細資訊，請參閱[在主要資料儲存空間中建立和捨棄目錄](#)。

3. 將 Oracle 錢包上傳至您的 Amazon S3 儲存貯體。

您可以使用任何支援的上傳技術。

4. 如要重新上傳 Oracle 錢包，請刪除現有錢包。否則，跳至下一步。

下列範例會移除名為 *cwallet.sso* 的現有錢包。

```
EXEC UTL_FILE.FREMOVE ('WALLET_DIR','cwallet.sso');
```

5. 將 Oracle 錢包從您的 Amazon S3 儲存貯體下載至 Oracle 資料庫執行個體。

下列範例會將名為 *cwallet.sso* 的錢包，從 Amazon S3 儲存貯體 *my_s3_bucket* 下載到資料庫執行個體目錄 *WALLET_DIR*。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'my_s3_bucket',
    p_s3_prefix   => 'cwallet.sso',
    p_directory_name => 'WALLET_DIR')
AS TASK_ID FROM DUAL;
```

6. (選用) 下載受密碼保護的 Oracle 錢包。

只有在您想要求每次使用錢包都輸入密碼時，才需下載此錢包。下列範例會下載受密碼保護的錢包 *ewallet.p12*。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'my_s3_bucket',
    p_s3_prefix   => 'ewallet.p12',
    p_directory_name => 'WALLET_DIR')
AS TASK_ID FROM DUAL;
```

7. 檢查您的資料庫任務。

下列範例中，將上述步驟傳回的任務 ID 替換為 *dbtask-1234567890123-4567.log*。

```
SELECT TEXT FROM
TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-1234567890123-4567.log'));
```

8. 檢查用於存放 Oracle 錢包的目錄內容。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

如需詳細資訊，請參閱[列出資料庫執行個體目錄中的檔案](#)。

步驟 4：授予使用者使用 Oracle 錢包的許可

您可以建立新的資料庫使用者或設定現有使用者。無論哪種情況，都必須為使用者設定 Oracle 錢包的存取權，以建立安全連線和使用憑證進行用戶端身分驗證。

授予使用者使用 Oracle 錢包的許可

1. 以主要使用者身分登入 RDS for Oracle 資料庫執行個體。
2. 如果不想設定現有的資料庫使用者，請建立一個新使用者。否則，跳至下一步。

下列範例會建立名為 *my-user* 的資料庫使用者。

```
CREATE USER my-user IDENTIFIED BY my-user-pwd;  
GRANT CONNECT TO my-user;
```

3. 向您的資料庫使用者授予許可，允許存取包含 Oracle 錢包的目錄。

下列範例會為 *my-user* 使用者授予目錄 *WALLET_DIR* 的讀取存取權。

```
GRANT READ ON DIRECTORY WALLET_DIR TO my-user;
```

4. 使用 UTL_HTTP 套件授予資料庫使用者許可。

下列 PL/SQL 程式授予 UTL_HTTP 存取權給使用者 *my-user*。

```
BEGIN  
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));  
END;  
/
```

5. 使用 UTL_FILE 套件授予資料庫使用者許可。

下列 PL/SQL 程式授予 UTL_FILE 存取權給使用者 *my-user*。

```
BEGIN  
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_FILE', UPPER('my-user'));  
END;  
/
```

步驟 5：設定從資料庫執行個體存取網站的權限

在此步驟中，會對 Oracle 資料庫使用者進行設定，允許該使用者使用 UTL_HTTP、您上傳的 Oracle 錢包和用戶端憑證連線至您的目標網站。如需詳細資訊，請參閱 Oracle 資料庫說明文件中的 [設定 Oracle 錢包的存取控制](#)。

設定從 RDS for Oracle 資料庫執行個體存取網站的權限

1. 以主要使用者身分登入 RDS for Oracle 資料庫執行個體。
2. 在安全連接埠上為使用者和目標網站建立主機存取控制項目 (ACE)。

下列範例會為 *my-user* 設定安全連接埠 443 上 *secret.encrypted-website.com* 的存取權。

```
BEGIN
DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host      => 'secret.encrypted-website.com',
  lower_port => 443,
  upper_port => 443,
  ace       => xs$ace_type(privilege_list => xs$name_list('http'),
                          principal_name => 'my-user',
                          principal_type => xs_acl.ptype_db));
-- If the program unit results in PLS-00201, set
-- the principal_type parameter to 2 as follows:
-- principal_type => 2));
END;
/
```

Important

上述程式單元可能會導致以下錯誤：PLS-00201: identifier 'XS_ACL' must be declared。如果傳回此錯誤，請以下列行取代指定值 `principal_type` 的行，然後重新執行程式單元：

```
principal_type => 2));
```

如需 PL/SQL 套裝程式中常數的詳細資訊 `XS_ACL`，請參閱 Oracle 資料庫說明文件中的 [「實際應用程式安全管理員和開發人員指南」](#)。

如需詳細資訊，請參閱 Oracle 資料庫說明文件中的 [設定外部網路服務的存取控制](#)。

3. (選用) 為標準連接埠上的使用者和目標網站建立 ACE。

如果某些網頁是從標準 Web 伺服器連接埠 (80) 而不是安全連接埠 (443) 提供的，可能需要使用標準連接埠。

```
BEGIN
DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host      => 'secret.encrypted-website.com',
  lower_port => 80,
```

```

upper_port => 80,
ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                        principal_name => 'my-user',
                        principal_type => xs_acl.ptype_db));
-- If the program unit results in PLS-00201, set
-- the principal_type parameter to 2 as follows:
-- principal_type => 2));

END;
/

```

4. 確認存取控制項目存在。

```

SET LINESIZE 150
COLUMN HOST FORMAT A40
COLUMN ACL FORMAT A50

SELECT HOST, LOWER_PORT, UPPER_PORT, ACL
FROM DBA_NETWORK_ACLS
ORDER BY HOST;

```

5. 使用 UTL_HTTP 套件授予資料庫使用者許可。

下列 PL/SQL 程式授予 UTL_HTTP 存取權給使用者 *my-user*。

```

BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
END;
/

```

6. 確認相關的存取控制清單存在。

```

SET LINESIZE 150
COLUMN ACL FORMAT A50
COLUMN PRINCIPAL FORMAT A20
COLUMN PRIVILEGE FORMAT A10

SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT,
       TO_CHAR(START_DATE, 'DD-MON-YYYY') AS START_DATE,
       TO_CHAR(END_DATE, 'DD-MON-YYYY') AS END_DATE
FROM DBA_NETWORK_ACL_PRIVILEGES
ORDER BY ACL, PRINCIPAL, PRIVILEGE;

```


7. 授予資料庫使用者許可，允許使用憑證進行用戶端身分驗證，並授予 Oracle 錢包進行連線的許可。

Note

如果您不透過用戶端憑證進行身分驗證，可略過此步驟。

```

DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
         INTO l_wallet_path
         FROM ALL_DIRECTORIES
         WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE(
    wallet_path => 'file:/' || l_wallet_path,
    ace         => xs$ace_type(privilege_list => xs
$name_list('use_client_certificates'),
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
END;
/

```

步驟 6：測試從資料庫執行個體連至網站的連線

在此步驟中，會對資料庫使用者進行設定，允許該使用者使用 UTL_HTTP、您上傳的 Oracle 錢包和用戶端憑證連線至您的網站。

設定從 RDS for Oracle 資料庫執行個體存取網站的權限

1. 以具有 UTL_HTTP 許可的資料庫使用者身分登入 RDS on Oracle 資料庫執行個體。
2. 確認連至目標網站的連線可以解析主機地址。

下列範例會從 *secret.encrypted-website.com* 取得主機地址。

```

SELECT UTL_INADDR.GET_HOST_ADDRESS(host => 'secret.encrypted-website.com')
FROM DUAL;

```

3. 測試失敗的連線。

以下查詢失敗，因為 UTL_HTTP 需要含有憑證的 Oracle 錢包的所在位置。

```
SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

4. 使用 UTL_HTTP.SET_WALLET 並從 DUAL 中選取來測試網站存取情況。

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
         INTO l_wallet_path
         FROM ALL_DIRECTORIES
         WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  UTL_HTTP.SET_WALLET('file:/' || l_wallet_path);
END;
/

SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

5. (選用) 將查詢儲存在變數中並使用 EXECUTE IMMEDIATE 來測試網站存取情況。

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
  v_webpage_sql VARCHAR2(1000);
  v_results      VARCHAR2(32767);
BEGIN
  SELECT DIRECTORY_PATH
         INTO l_wallet_path
         FROM ALL_DIRECTORIES
         WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  v_webpage_sql := 'SELECT UTL_HTTP.REQUEST(''secret.encrypted-website.com'', ''',
                  ''file:/' || l_wallet_path || ''') FROM DUAL';
  DBMS_OUTPUT.PUT_LINE(v_webpage_sql);
  EXECUTE IMMEDIATE v_webpage_sql INTO v_results;
  DBMS_OUTPUT.PUT_LINE(v_results);
END;
/
```

6. (選用) 尋找 Oracle 錢包目錄的檔案系統位置。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

使用上一個命令的輸出執行 HTTP 請求。例如，如果目錄為 *rdsdbdata/userdirs/01*，請執行下列查詢。

```
SELECT UTL_HTTP.REQUEST('https://secret.encrypted-website.com/', '',  
  'file://rdsdbdata/userdirs/01')  
FROM DUAL;
```

在 RDS for Oracle 中使用 CDB

在 Oracle 多租戶架構中，容器資料庫 (CDB) 可包含客戶建立的插入式資料庫 (PDB)。如需 CDB 的詳細資訊，請參閱 Oracle 資料庫文件中的[多租戶架構簡介](#)。

主題

- [RDS for Oracle CDB 概觀](#)
- [設定 RDS for Oracle CDB](#)
- [備份與還原 CDB](#)
- [將 RDS for Oracle 非 CDB 轉換為 CDB](#)
- [將單一租戶組態轉換為多租戶](#)
- [將 RDS for Oracle 租戶資料庫新增至 CDB 執行個體](#)
- [修改 RDS for Oracle 租戶資料庫](#)
- [從 CDB 刪除 RDS for Oracle 租戶資料庫](#)
- [檢視租戶資料庫詳細資訊](#)
- [升級 CDB](#)

RDS for Oracle CDB 概觀

在執行 Oracle 資料庫 19c 或更新版本時，您可以建立 RDS for Oracle 資料庫執行個體作為容器資料庫 (CDB)。從 Oracle 資料庫 21c 開始，所有資料庫皆為 CDB。一個 CDB 從非 CDB 不同，因為它可以包含可插拔的數據庫 (PDB)，這是所謂的 RDS 為甲骨文租戶數據庫。PDB 是結構描述和物件的可攜性集合，會對應用程式顯示為單獨的資料庫。

您可以在建立 CDB 執行個體時建立初始承租人資料庫 (PDB)。在 RDS 為甲骨文，您的客戶端應用程式與 PDB，而不是 CDB 交互。使用 PDB 的經驗與使用非 CDB 的經驗大致相同。

主題

- [CDB 架構的多租戶組態](#)
- [CDB 架構的單一租戶組態](#)
- [CDB 的建立和轉換選項](#)
- [CDB 中的使用者帳戶和權限](#)
- [CDB 中的參數群組系列](#)
- [RDS for Oracle CDB 的限制](#)

CDB 架構的多租戶組態

適用於 Oracle 的 RDS 支援 Oracle 多租戶架構的多租戶組態，也稱為 CDB 架構。在此組態中，您的 RDS for Oracle CDB 執行個體可以包含 1—30 個租用戶資料庫，視資料庫版本和任何必要的選項授權而定。在 Oracle 資料庫中，租用戶資料庫是 PDB。您的資料庫執行個體必須使用 Oracle 資料庫版本 19.0.0.0.ru-2022-01.rur-2022.r1 或更新版本。

Note

Amazon RDS 功能稱為「multi-tenant」而不是「multitenant」，因為它是 RDS 平台的一項功能，而不只是 Oracle 資料庫引擎而已。「Oracle 多租戶」一詞僅指與內部部署和 RDS 部署相容的 Oracle 資料庫架構。

您可以設定下列設定：

- 租戶資料庫名稱
- 租戶資料庫主要使用者名稱
- 租戶資料庫主要密碼
- 租戶資料庫字元集
- 租戶資料庫國家字元集

租戶資料庫字元集可以不同於 CDB 字元集。同樣適用於國家字元集。在建立初始租戶資料庫之後，您可以使用 RDS API 建立、修改或刪除租戶資料庫。CDB 名稱預設為 RDSCDB 且無法變更。如需詳細資訊，請參閱 [資料庫執行個體的設定](#) 及 [修改 RDS for Oracle 租戶資料庫](#)。

CDB 架構的單一租戶組態

RDS for Oracle 支援 Oracle 多租戶架構的舊版組態，稱為單一租戶組態。在此組態中，RDS for CDB 執行個體只能包含一個租戶 (PDB)。您稍後無法建立更多的 PDB。

CDB 的建立和轉換選項

Oracle Database 21c 只支援 CDB，但 Oracle Database 19c 同時支援 CDB 和非 CDB。所有 RDS for Oracle CDB 執行個體都支援多租戶和單一租戶組態。

Oracle 資料庫架構的建立、轉換和升級選項

下表顯示用於建立和升級 RDS for Oracle 資料庫的不同架構選項。

發行版本	資料庫建立選項	架構轉換選項	主要版本升級目標
Oracle Database 21c	僅限 CDB 架構	N/A	N/A
Oracle 資料庫 19c	CDB 或非 CDB 架構	非 CDB 至 CDB 架構 (2021 年 4 月 RU 或更新版本)	甲骨文数据库

如前表所示，您不能在新的主要資料庫版本中直接將非 CDB 升級為 CDB。但您可將 Oracle 資料庫 19c 非 CDB 轉換為 Oracle 資料庫 19c CDB，然後將 Oracle 資料庫 19c CDB 升級為 Oracle 資料庫 21c CDB。如需詳細資訊，請參閱 [將 RDS for Oracle 非 CDB 轉換為 CDB](#)。

CDB 架構組態的轉換選項

下表顯示用於轉換 RDS for Oracle 資料庫執行個體之架構組態的不同選項。

目前架構和組態	轉換為 CDB 架構的單一租戶組態	轉換為 CDB 架構的多租戶組態	轉換為非 CDB 架構
非 CDB	支援	支援*	N/A
使用單一租戶組態的 CDB	N/A	支援	不支援
使用多租戶組態的 CDB	不支援	N/A	不支援

您無法在單一操作中將非 CDB 轉換為多租戶組態。當您將非 CDB 轉換為 CDB 時，您的 CDB 會處於單一租戶組態中。然後，您可以在個別的操作中將單一租戶轉換為多租戶組態。

CDB 中的使用者帳戶和權限

在 Oracle 多租用戶架構中，所有使用者帳戶都是一般使用者或本機使用者。CDB 一般使用者為資料庫使用者，其單一身分和密碼在根 CDB 及所有現有和未來的 PDB 中皆為已知狀態。相反地，本機使用者只存在於單一 PDB。

RDS 主要使用者是 PDB 中的本機使用者帳戶，您可在建立資料庫執行個體時命名該帳戶。若您建立新的使用者帳戶，這些使用者也會是位於 PDB 中的本機使用者。您無法使用任何使用者帳戶來建立新的 PDB 或修改現有 PDB 的狀態。

rdsadmin 使用者是一般使用者帳戶。您可以執行此帳戶中既有的 RDS for Oracle 套件，但無法以 rdsadmin 身分登入。如需詳細資訊，請參閱 Oracle 文件中的[關於一般使用者和本機使用者](#)。

CDB 中的參數群組系列

CDB 有自己的參數群組系列和預設參數值。CDB 參數群組系列如下所示：

- oracle-ee-cdb-21
- oracle-se2-cdb-21
- oracle-ee-cdb-19
- oracle-se2-cdb-19

RDS for Oracle CDB 的限制

RDS for Oracle 支援可在內部部署 CDB 中使用的功能子集。

CDB 限制

下列限制適用於 RDS for Oracle CDB：

- 您無法連線到 CDB。您一律會連線到租戶資料庫 (PDB)，而不是 CDB。指定 PDB 的端點，和操作非 CDB 一樣。唯一的差異在於您將資料庫名稱指定為 pdb_name，pdb_name 即是您為 PDB 選擇的名稱。
- 您無法將多租戶組態中的 CDB 轉換為單一租戶轉換中的 CDB。轉換為多租戶組態是單向且是不可復原的。
- 如果您的資料庫執行個體使用低於 19.0.0.0.ru-2022-01.rur-2022.r1 的 Oracle 資料庫版本，您無法啟用或轉換為多租戶組態。
- 您不能使用 RDS for Oracle CDB 搭配 ORDS v22 及更高版本。作為因應措施，您可以使用較低版本的 ORDS，或使用 Oracle 資料庫 19c 非 CDB。
- 您不能使用 RDS 為甲骨文 CDB 與 ORDS 22 及更高版本。作為因應措施，您可以使用較低版本的 ORDS，或使用 Oracle 資料庫 19c 非 CDB。

是否支援下列功能取決於架構組態。


功能	在單一租戶中受到支援	在多租戶中受到支援
Oracle Data Guard	是	否
Oracle Label Security	否	否
Oracle Enterprise Manager (OEM)	否	否
OEM 代理程式	否	否
資料庫活動串流	是	否

租戶資料庫 (PDB) 限制

下列限制適用於 RDS for Oracle 多租戶組態中的租戶資料庫：

- 您無法將租戶資料庫操作延遲至維護時段。所有變更都會立即發生。
- 您無法將租戶資料庫新增至使用單一租戶組態的 CDB。
- 您無法在單一操作中新增或修改多個租戶資料庫。您一次只能新增或修改一個租戶資料庫。
- 您無法修改要命名為 CDB\$ROOT 或 PDB\$SEED 的租戶資料庫。
- 如果租戶資料庫是 CDB 中的唯一租戶，則您無法刪除該資料庫。
- 並非所有資料庫執行個體類別類型都有足夠的資源，來支援 RDS for Oracle CDB 執行個體中的多個 PDB。增加的 PDB 計數會影響小型執行個體類別的效能和穩定性，並會增加大部分執行個體層級操作 (例如資料庫升級) 的時間。
- 您不能使用多個 AWS 帳戶 在同一 CDB 中創建 PDB。PDB 必須由與 PDB 託管所在資料庫執行個體相同的帳戶擁有。
- CDB 中的所有 PDB 都會使用相同的端點和資料庫接聽程式。
- PDB 層級不支援下列操作，但在 CDB 層級支援：
 - 備份與復原
 - 資料庫升級
 - 維護動作
- PDB 層級不支援下列功能，但在 CDB 層級支援：
 - 選項群組 (選項安裝在 CDB 執行個體的所有 PDB 上)

- 參數群組 (所有參數都衍生自與 CDB 執行個體相關聯的參數群組)
- 在內部部署 CDB 架構中支援，但在 RDS for Oracle CDB 中不支援的 PDB 層級操作包括下列項目：

 Note

以下清單並非詳盡清單。

- 應用程式 PDB
- Proxy PDB
- 啟動和停止 PDB
- 拔除和插入 PDB

要將資料移入或移出 CDB，請使用與非 CDB 相同的技術。如需遷移資料的詳細資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle](#)。

- 在 PDB 層級設定選項

PDB 從 CDB 選項群組繼承選項設定。如需關於設定選項的詳細資訊，請參閱 [使用參數群組](#)。如需最佳實務做法，請參閱「[使用資料庫參數群組](#)」。

- 在 PDB 中設定參數

PDB 繼承了 CDB 的參數設定。如需關於設定選項的詳細資訊，請參閱 [將選項新增至 Oracle 資料庫執行個體](#)。

- 在同一 CDB 中為 PDB 設定不同的接聽程式
- Oracle Flashback 功能
- 從 PDB 內稽核資訊

設定 RDS for Oracle CDB

設定 CDB 的方法類似於設定非 CDB。

主題

- [建立 RDS for Oracle 資料庫執行個體](#)
- [在 RDS for Oracle CDB 中連線至 PDB](#)

建立 RDS for Oracle 資料庫執行個體

在 RDS for Oracle 中，建立 CDB 的方式幾乎與建立非 CDB 相同。不同之處在於，建立資料庫執行個體時您選擇 Oracle 多租戶架構，而且也選擇架構組態：多租戶或單一租戶。如果您在多租戶組態中建立 CDB 時建立標籤，RDS 會將標籤傳播到初始租戶資料庫。若要建立 CDB，請使用 AWS Management Console、AWS CLI、或 RDS API。

主控台

建立 CDB 執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要建立 CDB 執行個體的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。
5. 在 Choose a database creation method (選擇資料庫建立方法) 中，選取 Standard Create (標準建立)。
6. 在 Engine options (引擎選項) 中，選擇 Oracle (Oracle)。
7. 請為資料庫管理類型選擇 Amazon RDS。
8. 對於架構設定，選擇 Oracle 多租戶架構。
9. 對於架構組態，請執行下列任一動作：
 - 選擇多租戶組態，然後繼續進行下一個步驟。
 - 選擇單一租戶組態，然後跳至步驟 11。
10. (多租戶組態) 對於租戶資料庫設定，請進行下列變更：
 - 對於租戶資料庫名稱，輸入初始 PDB 的名稱。PDB 名稱必須與 CDB 名稱 (預設為 RDSCDB) 不同。
 - 對於租戶資料庫主要使用者名稱，輸入 PDB 的主要使用者名稱。您不能使用租戶資料庫主要使用者名稱登入 CDB 本身。
 - 在租戶資料庫主要密碼中輸入密碼，或選擇自動產生密碼。
 - 對於租戶資料庫字元集，選擇 PDB 的字元集。您可以選擇不同於 CDB 字元集的租戶資料庫字元集。

預設的 PDB 字元集為 AL32UTF8。如果您選擇非預設的 PDB 字元集，CDB 的建立速度可能會變慢。

Note

您無法建立多個租戶資料庫做為 CDB 建立程序的一部分。您只能將 PDB 新增至已存在的 CDB。

11. (單一租戶組態) 根據 [資料庫執行個體的設定](#) 中列出的選項選擇所需的設定。注意下列事項：
 - 在主要使用者名稱中，輸入 PDB 中本機使用者的名稱。您不能使用主要使用者名稱登入 CDB 根目錄。
 - 請為初始資料庫名稱輸入 PDB 的名稱。您無法命名具有預設名稱 RDS CDB 的 CDB。
12. 選擇建立資料庫。

AWS CLI

若要在多租用戶組態中建立 CDB，請搭配下列參數使用 [create-db-instance](#) 命令：

- `--db-instance-identifier`
- `--db-instance-class`
- `--engine { oracle-ee-cdb | oracle-se2-cdb }`
- `--master-username`
- `--master-user-password`
- `--multi-tenant` (針對單一租戶組態，請不要指定 `multi-tenant` 或指定 `--no-multi-tenant`)
- `--allocated-storage`
- `--backup-retention-period`

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

下列範例會建立一個以多租用戶組態命名 *my-cdb-inst* 的 RDS for Oracle 資料庫執行個體。如果您指定 `--no-multi-tenant` 或未指定 `--multi-tenant`，預設 CDB 組態為單一租戶。引擎是 `oracle-ee-cdb`：指定 `oracle-ee` 和 `--multi-tenant` 的命令失敗，因為發生錯誤。初始租戶資料庫命名為 *mypdb*。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance \  
  --engine oracle-ee-cdb \  
  --db-instance-identifier my-cdb-inst \  
  --multi-tenant \  
  --db-name mypdb \  
  --allocated-storage 250 \  
  --db-instance-class db.t3.large \  
  --master-username pdb_admin \  
  --master-user-password pdb_admin_password \  
  --backup-retention-period 3
```

在Windows中：

```
aws rds create-db-instance ^  
  --engine oracle-ee-cdb ^  
  --db-instance-identifier my-cdb-inst ^  
  --multi-tenant ^  
  --db-name mypdb ^  
  --allocated-storage 250 ^  
  --db-instance-class db.t3.large ^  
  --master-username pdb_admin ^  
  --master-user-password pdb_admin_password ^  
  --backup-retention-period 3
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

此命令會產生類似下列的輸出。資料庫名稱、字元集、國家字元集和主要使用者不會包含在輸出中。您可以使用 CLI 命令 `describe-tenant-databases` 來檢視此資訊。

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "my-cdb-inst",  
    "DBInstanceClass": "db.t3.large",  
    "MultiTenant": true,  
    "Engine": "oracle-ee-cdb",
```

```
"DBResourceId": "db-ABCDEFGHJKLMNOPQRSTUVWXYZ",
"DBInstanceStatus": "creating",
"AllocatedStorage": 250,
"PreferredBackupWindow": "04:59-05:29",
"BackupRetentionPeriod": 3,
"DBSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-0a1bcd2e",
    "Status": "active"
  }
],
"DBParameterGroups": [
  {
    "DBParameterGroupName": "default.oracle-ee-cdb-19",
    "ParameterApplyStatus": "in-sync"
  }
],
"DBSubnetGroup": {
  "DBSubnetGroupName": "default",
  "DBSubnetGroupDescription": "default",
  "VpcId": "vpc-1234567a",
  "SubnetGroupStatus": "Complete",
  ...
}
```

RDS API

若要使用 Amazon RDS API 來建立資料庫執行個體，請呼叫 [CreateDBInstance](#) 操作。

如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

在 RDS for Oracle CDB 中連線至 PDB

您可以使用像 SQL*Plus 之類的公用程式連線至 PDB。若要下載包含獨立版 SQL*Plus 的 Oracle 即時用戶端，請參閱 [Oracle 即時用戶端下載](#)。

您需要下列資訊才能將 SQL*Plus 連線至 PDB：

- PDB 名稱
- 資料庫使用者名稱和密碼
- 資料庫執行個體的端點
- 連接埠號碼

如需了解如何尋找上述資訊，請參閱 [尋找您的 RDS for Oracle 資料庫執行個體的端點](#)。

Example 使用 SQL*Plus 連線至 PDB 的方法如下

在下列範例中，將 *master_user_name* 替換為您的主要使用者。同時，將 endpoint (端點) 替換為您的資料庫執行個體，然後包含連接埠號碼和 Oracle SID。SID 值是您在建立資料庫執行個體時指定的 PDB 名稱，而非資料庫執行個體識別符。

對於LinuxmacOS、或Unix：

```
sqlplus 'master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port)))(CONNECT_DATA=(SID=pdb_name)))'
```

在Windows中：

```
sqlplus master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port)))(CONNECT_DATA=(SID=pdb_name)))
```

您應該會看到類似下列的輸出。

```
SQL*Plus: Release 19.0.0.0.0 Production on Mon Aug 21 09:42:20 2021
```

輸入使用者的密碼之後，隨即會出現 SQL 提示。

```
SQL>
```

Note

簡短的格式連接字串 (Easy connect 或 EZCONNECT)，例如 `sqlplus username/password@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/database-identifier`，可能遇到最大字元限制，因此不應用來連接。

備份與還原 CDB

您可以使用 RDS 資料庫快照或 Recovery Manager (RMAN) 來備份和還原 CDB。

使用 CDB 快照備份和還原 CDB

資料庫快照在 CDB 和非 CDB 架構中的運作方式類似。主要差異如下所示：

- 還原 CDB 的資料庫快照時，您無法重新命名 CDB。CDB 命名為 RDSCDB 且無法變更。
- 還原 CDB 的資料庫快照時，您無法重新命名 PDB。您可以使用 [modify-tenant-database](#) 命令來修改 PDB 名稱。
- 若要在快照中尋找租戶資料庫，請使用 CLI 命令 [describe-db-snapshot-tenant-databases](#)。
- 在使用多租戶架構組態的 CDB 快照中，您無法直接與其中的租戶資料庫互動。如果還原資料庫快照，則會還原其所有租戶資料庫。
- RDS for Oracle 會隱含地將租戶資料庫上的標籤複製到資料庫快照中的租戶資料庫。當您還原租戶資料庫時，該標籤會出現在還原的資料庫中。
- 如果您還原資料庫快照並使用 `--tags` 參數指定新標籤，則新標籤會覆寫所有現有的標籤。
- 如果您對具有標籤的 CDB 執行個體建立資料庫快照，並指定 `--copy-tags-to-snapshot`，則 RDS for Oracle 會將這些標籤從租戶資料庫複製到快照中的租戶資料庫。

如需更多詳細資訊，請參閱 [Oracle 資料庫考量事項](#)。

使用 RMAN 備份和還原 CDB

若要了解如何使用 RMAN 備份和還原 CDB 或個別租戶資料庫，請參閱 [執行 Oracle 資料庫執行個體的一般 RMAN 任務](#)。

將 RDS for Oracle 非 CDB 轉換為 CDB

您可以使用命令將 Oracle 資料庫的架構從非 CDB 架構變更為 Oracle 多租戶架構 (也稱為 CDB 架構)。modify-db-instance 在大多數情況下，這種技術比創建新 CDB 和導入數據更好。轉換操作會導致停機時間。

升級資料庫引擎版本時，您無法在相同的操作中變更資料庫架構。因此，若要將 Oracle 資料庫 19c 非 CDB 升級為 Oracle 資料庫 21c CDB，您首先需要以一個步驟將非 CDB 轉換為 CDB，然後以個別步驟將 19c CDB 升級至 21c CDB。

非 CDB 轉換操作需求如下：

- 您必須為資料庫引擎類型指定 `oracle-ee-cdb` 或 `oracle-se2-cdb`。只有這些值才受支援。
- 您的資料庫引擎必須使用具有 2021 年 4 月或更晚版本更新 (RU) 的 Oracle Database 19c。

此操作具有下列限制：

- 您無法將 CDB 轉換為非 CDB。您無法將非 CDB 轉換為 CDB。

- 您無法在單一 `modify-db-instance` 呼叫中將非 CDB 轉換為多租戶組態。在您將非 CDB 轉換為 CDB 之後，您的 CDB 會處於單一租戶組態中。若要將單一租戶組態轉換為多租戶組態，請再次執行 `modify-db-instance`。如需詳細資訊，請參閱[將單一租戶組態轉換為多租戶](#)。
- 您無法轉換已啟用 Oracle Data Guard 的主要或複本資料庫。若要轉換具有僅供讀取複本的非 CDB，請先刪除所有僅供讀取複本。
- 您不能升級資料庫引擎版本，並在相同操作中將非 CDB 轉換為 CDB。
- 對於選項和參數群組的考量與升級資料庫引擎的考量相同。如需詳細資訊，請參閱[Oracle 資料庫升級的考量](#)。

主控台

將非 CDB 轉換為 CDB 的方法如下

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體所在的 AWS 區域。
3. 在導覽窗格中選擇資料庫，然後選擇要將其轉換為 CDB 執行個體的非 CDB 執行個體。
4. 選擇 Modify (修改)。
5. 對於架構設定，選取 Oracle 多租戶架構。轉換後，您的 CDB 將處於單一租戶組態中。
6. (選用)對於資料庫參數群組，請為 CDB 執行個體選擇新的參數群組。在轉換資料庫執行個體時，也需要考慮與升級資料庫執行個體時相同的參數群組。如需詳細資訊，請參閱[參數群組考量](#)。
7. (選用)在目標選項群組中，為 CDB 執行個體選擇新的選項群組。在轉換資料庫執行個體時，也需要考慮與升級資料庫執行個體時相同的選項群組。如需詳細資訊，請參閱[選項群組考量](#)。
8. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看修改摘要。
9. (選用)選擇 Apply immediately (立即套用) 以立即套用變更。在某些情況下，選擇此選項會導致停機。如需詳細資訊，請參閱[排程修改設定](#)。
10. 在確認頁面上，檢閱您的變更。如果都正確，請選擇修改資料庫執行個體。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要將資料庫執行個體上的非 CDB 轉換為單租用戶組態中的 CDB，請在命令中設定 `--engine` 為 `oracle-ee-cdb` 或 `oracle-se2-cdb`。AWS CLI [modify-db-instance](#) 如需詳細資訊，請參閱[資料庫執行個體的設定](#)。

下列範例會轉換名為的資料庫執行個體，*my-non-cdb*並指定自訂選項群組和參數群組。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier my-non-cdb \  
  --engine oracle-ee-cdb \  
  --option-group-name custom-option-group \  
  --db-parameter-group-name custom-parameter-group
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier my-non-cdb ^  
  --engine oracle-ee-cdb ^  
  --option-group-name custom-option-group ^  
  --db-parameter-group-name custom-parameter-group
```

RDS API

若要將非 CDB 轉換為 CDB，請在 RDS API 操作 [ModifyDBInstance](#) 中指定 Engine。

將單一租戶組態轉換為多租戶

您可以將 RDS for Oracle CDB 的架構從單一租戶組態修改為多租戶組態。轉換前後，您的 CDB 會包含單一租戶資料庫 (PDB)。

在轉換期間，RDS for Oracle 會將下列中繼資料遷移至新的租戶資料庫：

- 主要使用者名稱
- 資料庫名稱
- 字元集
- 國家字元集

轉換前，您可以使用 `describe-db-instances` 命令來檢視上述資訊。轉換後，您可以使用 `describe-tenant-database` 命令來檢視資訊。

轉換具有下列需求和限制：

- 在將單一租戶架構組態轉換為多租戶組態之後，您以後無法將架構轉換回單一租戶組態。此操作是不可復原的。
- 資料庫執行個體的標籤會傳播到轉換期間建立的初始租戶資料庫。
- 您無法轉換已啟用 Oracle Data Guard 的主要或複本資料庫。
- 您無法升級資料庫引擎版本，也無法在相同操作中轉換為多租戶組態。
- 您的 IAM 政策必須具有建立租戶資料庫的許可。

主控台

將使用單一租戶組態的 CDB 轉換為多租戶組態

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體所在的 AWS 區域。
3. 在導覽窗格中選擇資料庫，然後選擇要將其轉換為 CDB 執行個體的非 CDB 執行個體。
4. 選擇 Modify (修改)。
5. 對於架構設定，選取 Oracle 多租戶架構。
6. 對於架構組態，選取多租戶組態。
7. (選用)對於資料庫參數群組，請為 CDB 執行個體選擇新的參數群組。在轉換資料庫執行個體時，也需要考慮與升級資料庫執行個體時相同的參數群組。
8. (選用) 在目標選項群組中，為 CDB 執行個體選擇新的選項群組。在轉換資料庫執行個體時，也需要考慮與升級資料庫執行個體時相同的選項群組。
9. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看修改摘要。
10. 選擇 Apply immediately (立即套用)。當您切換到多租戶組態時，需要此選項。請注意，此選項會在某些情況下導致停機。
11. 在確認頁面上，檢閱您的變更。如果都正確，請選擇修改資料庫執行個體。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要使用單租用戶組態將 CDB 轉換為多承租人組態，請 `--multi-tenant` 在命令中指定。AWS CLI [modify-db-instance](#)

下列範例會將名為 `my-st-cdb` 的資料庫執行個體從單一租戶組態轉換為多租戶組態。`--apply-immediately` 選項為必要項目。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance --region us-east-1 \  
  --db-instance-identifier my-st-cdb \  
  --multi-tenant \  
  --apply-immediately
```

在Windows中：

```
aws rds modify-db-instance --region us-east-1 ^ \  
  --db-instance-identifier my-st-cdb ^ \  
  --multi-tenant ^ \  
  --apply-immediately
```

輸出類似如下所示。

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "my-st-cdb",  
    "DBInstanceClass": "db.r5.large",  
    "MultiTenant": false,  
    "Engine": "oracle-ee-cdb",  
    "DBResourceId": "db-AB1CDE2FGHIJK34LMNOPRLXTXU",  
    "DBInstanceStatus": "modifying",  
    "MasterUsername": "admin",  
    "DBName": "ORCL",  
    ...  
    "EngineVersion": "19.0.0.0.ru-2022-01.rur-2022-01.r1",  
    "AutoMinorVersionUpgrade": true,  
    "ReadReplicaDBInstanceIdentifiers": [],  
    "LicenseModel": "bring-your-own-license",  
    "OptionGroupMemberships": [  
      {  
        "OptionGroupName": "default:oracle-ee-cdb-19",  
        "Status": "in-sync"  
      }  
    ],  
    ...  
  }  
}
```

```
    "PendingModifiedValues": {  
      "MultiTenant": "true"  
    }  
  }  
}
```

將 RDS for Oracle 租戶資料庫新增至 CDB 執行個體

在 RDS for Oracle 多租戶組態中，租戶資料庫是 PDB。若要新增租戶資料庫，請確定符合下列先決條件：

- 您的 CDB 已啟用多租戶組態。如需詳細資訊，請參閱 [CDB 架構的多租戶組態](#)。
- 您擁有建立租戶資料庫的必要 IAM 許可。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 新增租戶資料庫。您無法在單一操作中新增多個租戶資料庫：您必須一次新增一個租戶資料庫。如果 CDB 已啟用備份保留，則 Amazon RDS 會在新增租戶資料庫之前和之後備份資料庫執行個體。

主控台

將租戶資料庫新增至您的資料庫執行個體

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要在其中建立租戶資料庫的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇您要將租戶資料庫新增至其中的 CDB 執行個體。您的資料庫執行個體必須使用 CDB 架構的多租戶組態。
5. 選擇動作，然後選擇新增租戶資料庫。
6. 對於租戶資料庫設定，請執行下列動作：
 - 對於租戶資料庫名稱，輸入新 PDB 的名稱。
 - 對於租戶資料庫主要使用者名稱，為 PDB 輸入主要使用者的名稱。此主要使用者與 CDB 的主要使用者不同。
 - 在租戶資料庫主要密碼中輸入密碼，或選取自動產生密碼。
 - 對於租戶資料庫字元集，選擇 PDB 的字元集。預設值為 AL32UTF8。您可以選擇不同於 CDB 字元集的 PDB 字元集。

- 對於租戶資料庫國家字元集，選擇 PDB 的國家字元集。預設值為 AL32UTF8。國家字元集只會針對使用 NCHAR 資料類型 (NCHAR、NVARCHAR2 和 NCLOB) 的資料行指定編碼，而且不會影響資料庫中繼資料。

如需上述設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

7. 選擇新增租戶。

AWS CLI

若要使用將承租人資料庫新增至 CDBAWS CLI，請使用 [create-tenant-database](#) 具有下列必要參數的命令：

- `--db-instance-identifier`
- `--tenant-db-name`
- `--master-username`
- `--master-user-password`

下列範例會在名為的 RDS 適用於 Oracle CDB 執行個體中建立名為 *mydb2* 的租用戶資料庫。*my-cdb-inst* PDB 字元集為 UTF-16。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-tenant-database --region us-east-1 \  
  --db-instance-identifier my-cdb-inst \  
  --tenant-db-name mypdb2 \  
  --master-username mypdb2-admin \  
  --master-user-password mypdb2-pwd \  
  --character-set-name UTF-16
```

在Windows中：

```
aws rds create-tenant-database --region us-east-1 \  
  --db-instance-identifier my-cdb-inst ^ \  
  --tenant-db-name mypdb2 ^ \  
  --master-username mypdb2-admin ^ \  
  --master-user-password mypdb2-pwd ^
```

```
--character-set-name UTF-16
```

輸出結果類似如下。

```
...}
  "TenantDatabase" :
    {
      "DbiResourceId" : "db-abc123",
      "TenantDatabaseResourceId" : "tdb-bac567",
      "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:mypdb2",
      "DBInstanceIdentifier" : "my-cdb-inst",
      "TenantDBName" : "mypdb2",
      "Status" : "creating",
      "MasterUsername" : "mypdb2",
      "CharacterSetName" : "UTF-16",
      ...
    }
}...
```

修改 RDS for Oracle 租戶資料庫

您只能修改 CDB 中租戶資料庫的 PDB 名稱和主要使用者密碼。請注意下列需求和限制：

- 若要修改資料庫執行個體中租戶資料庫的設定，租戶資料庫必須存在。
- 您無法在單一操作中修改多個租戶資料庫。您一次只能修改一個租戶資料庫。
- 您無法將租戶資料庫的名稱變更為 CDB\$ROOT 或 PDB\$SEED。

您可以使用 AWS Management Console、AWS CLI 或 RDS API 修改 PDB。

主控台

修改租戶資料庫的 PDB 名稱或主要密碼

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇要在其中建立租戶資料庫的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇您要修改其資料庫名稱或主要使用者密碼的租戶資料庫。

5. 選擇 Modify (修改)。
6. 對於租戶資料庫設定，請執行下列任一動作：
 - 對於租戶資料庫名稱，輸入新 PDB 的新名稱。
 - 對於租戶資料庫主要的密碼，輸入新的密碼。
7. 選擇修改租戶。

AWS CLI

若要使用修改承租人資料庫AWS CLI，請使用下列參數呼叫[modify-tenant-database](#)命令：

- `--db-instance-identifier` *##*
- `--tenant-db-name` *value*
- `[--new-tenant-db-name` *value*]
- `[--master-user-password` *value*]

下列範例會將租戶資料庫 `pdb1` 重新命名為資料庫執行個體 `my-cdb-inst` 上的 `pdb-hr`。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-tenant-database --region us-east-1 \  
  --db-instance-identifier my-cdb-inst \  
  --tenant-db-name pdb1 \  
  --new-tenant-db-name pdb-hr
```

在Windows中：

```
aws rds modify-tenant-database --region us-east-1 ^  
  --db-instance-identifier my-cdb-inst ^  
  --tenant-db-name pdb1 ^  
  --new-tenant-db-name pdb-hr
```

此命令會產生類似下列的輸出。

```
{  
  "TenantDatabase" : {
```

```
"DbiResourceId" : "db-abc123",
"TenantDatabaseResourceId" : "tdb-bac567",
"TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb1",
"DBInstanceIdentifier" : "my-cdb-inst",
"TenantDBName" : "pdb1",
"Status" : "modifying",
"MasterUsername" : "tenant-admin-user"
"Port" : "6555",
"CharacterSetName" : "UTF-16",
"MaxAllocatedStorage" : "1000",
"ParameterGroups": [
  {
    "ParameterGroupName": "pdb1-params",
    "ParameterApplyStatus": "in-sync"
  }
],
"OptionGroupMemberships": [
  {
    "OptionGroupName": "pdb1-options",
    "Status": "in-sync"
  }
],
"PendingModifiedValues": {
  "TenantDBName": "pdb-hr"
}
}
```

從 CDB 刪除 RDS for Oracle 租戶資料庫

您可以使用 AWS Management Console、AWS CLI 或 RDS API 刪除租戶資料庫 (PDB)。請考慮下列先決條件和限制：

- 租戶資料庫和資料庫執行個體必須存在。
- 若要成功刪除，必須存在下列其中一種情況：
 - 租戶資料庫和資料庫執行個體可用。

Note

您可以建立最終快照，但只有在您發出了 `delete-tenant-database` 命令之前，租戶資料庫和資料庫執行個體處於可用狀態時才能這樣做。

- 正在建立租戶資料庫。
- 資料庫執行個體正在修改租戶資料庫。
- 您無法在單一操作中刪除多個租戶資料庫。
- 如果租戶資料庫是 CDB 中的唯一租戶，則您無法刪除該資料庫。

主控台

刪除租戶資料庫

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇資料庫，然後選擇您要刪除的租戶資料庫。
3. 對於 Actions (動作)，請選擇 Delete (刪除)。
4. 若要為資料庫執行個體建立最終的資料庫快照，請選擇 Create final snapshot? (是否建立最終快照?)。
5. 如果您選擇建立最終快照，請輸入最終快照名稱。
6. 在方塊中輸入 **delete me**。
7. 選擇 Delete (刪除)。

AWS CLI

若要使用刪除承租人資料庫 AWS CLI，請使用下列參數呼叫 [delete-tenant-database](#) 命令：

- `--db-instance-identifier value`
- `--tenant-db-name value`
- `[--skip-final-snapshot | --no-skip-final-snapshot]`
- `[--final-snapshot-identifier value]`

下列範例會從名為的 CDB 刪除名為 *pdb-test* 的租用戶資料庫。*my-cdb-inst* 依預設，此操作會建立最終快照。

Example

對於 Linux macOS、或 Unix：

```
aws rds delete-tenant-database --region us-east-1 \  
  --db-instance-identifier my-cdb-inst \  
  --tenant-db-name pdb-test \  
  --final-snapshot-identifier final-snap-pdb-test
```

在Windows中：

```
aws rds delete-tenant-database --region us-east-1 ^  
  --db-instance-identifier my-cdb-inst ^  
  --tenant-db-name pdb-test ^  
  --final-snapshot-identifier final-snap-pdb-test
```

此命令會產生類似下列的輸出。

```
{  
  "TenantDatabase" : {  
    "DbiResourceId" : "db-abc123",  
    "TenantDatabaseResourceId" : "tdb-bac456",  
    "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb-  
test",  
    "DBInstanceIdentifier" : "my-cdb-inst",  
    "TenantDBName" : "pdb-test",  
    "Status" : "deleting",  
    "MasterUsername" : "pdb-test-admin"  
    "Port" : "6555",  
    "CharacterSetName" : "UTF-16",  
    "MaxAllocatedStorage" : "1000",  
    "ParameterGroups": [  
      {  
        "ParameterGroupName": "tenant-1-params",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "OptionGroupMemberships": [  
      {  
        "OptionGroupName": "tenant-1-options",  
        "Status": "in-sync"  
      }  
    ]  
  }  
}
```

檢視租戶資料庫詳細資訊

您可以採用與非 CDB 或 CDB 相同的方式，檢視租戶資料庫的詳細資料。

主控台

檢視租戶資料庫的詳細資訊

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台的右上角，選擇資料庫執行個體所在的 AWS 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。

Databases (2)		<input checked="" type="checkbox"/> Group resources	Refresh	Modify	Actions	Restore from S3	Create database
Filter by databases							
DB identifier	Status	Role	Engine	Region & AZ	Size	CPU	
<input type="radio"/> cdb-multi-config	Available	Instance	Oracle Enterprise Edition (CDB)		db.t3.small		
<input type="radio"/> PDB1	Available	Tenant DB	-	-	-	-	

在上圖中，唯一租戶資料庫 (PDB) 會顯示為資料庫執行個體的子項。

4. 選擇租戶資料庫的名稱。

PDB1

Tenant DBs (1) Refresh Modify Delete

Find resources

Tenant DB name	Status	Deletion protection
PDB1	Available	No

Configuration | Tags

Configuration : PDB1

Instance database cdb-multi-config	Tenant database resource ID tdb- [REDACTED]
Tenant database name PDB1	Deletion protection No
Tenant database (ARN) arn:aws:rds:us-west-2:[REDACTED]:tenant-database:tdb- [REDACTED]	Character Set AL32UTF8
Tenant database username admin	National Character Set AL16UTF16

AWS CLI

若要查看 PDB 的詳細資訊，請使用指AWS CLI令[describe-tenant-databases](#)。

下列範例描述指定區域中的所有租戶資料庫。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-tenant-databases --region us-east-1
```

在Windows中：

```
aws rds describe-tenant-databases --region us-east-1
```

此命令會產生類似下列的輸出。

```
"TenantDatabases" : [
  {
    "DBInstanceIdentifier" : "my-cdb-inst",
```

```

    "TenantDBName" : "pdb-test",
    "Status" : "available",
    "MasterUsername" : "pdb-test-admin",
    "DbiResourceId" : "db-abc123",
    "TenantDatabaseResourceId" : "tdb-bac456",
    "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb-test",
    "CharacterSetName": "AL32UTF8",
    "NcharCharacterSetName": "AL16UTF16",
    "DeletionProtection": false,
    "PendingModifiedValues": {
        "MasterUserPassword": "*****"
    },
    "TagList": []
},
{
    "DBInstanceIdentifier" : "my-cdb-inst2",
    "TenantDBName" : "pdb-dev",
    "Status" : "modifying",
    "MasterUsername" : "masterrdsuser"
    "DbiResourceId" : "db-xyz789",
    "TenantDatabaseResourceId" : "tdb-ghp890",
    "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst2:pdb-dev",
    "CharacterSetName": "AL32UTF8",
    "NcharCharacterSetName": "AL16UTF16",
    "DeletionProtection": false,
    "PendingModifiedValues": {
        "MasterUserPassword": "*****"
    },
    "TagList": []
},
... other truncated data

```

下列範例描述指定區域中資料庫執行個體 `my-cdb-inst` 上的租戶資料庫。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-tenant-databases --region us-east-1 \
  --db-instance-identifier my-cdb-inst
```

在Windows中：

```
aws rds describe-tenant-databases --region us-east-1 ^
  --db-instance-identifier my-cdb-inst
```

此命令會產生類似下列的輸出。

```
{
  "TenantDatabase": {
    "TenantDatabaseCreateTime": "2023-10-19T23:55:30.046Z",
    "DBInstanceIdentifier": "my-cdb-inst",
    "TenantDBName": "pdb-hr",
    "Status": "creating",
    "MasterUsername": "tenant-admin-user",
    "DbiResourceId": "db-abc123",
    "TenantDatabaseResourceId": "tdb-bac567",
    "TenantDatabaseARN": "arn:aws:rds:us-west-2:579508833180:pdb-hr:tdb-
    abcdefghijklmno2p3qrst4uvw5xy6zabc7defghi8jklmn90op",
    "CharacterSetName": "AL32UTF8",
    "NcharCharacterSetName": "AL16UTF16",
    "DeletionProtection": false,
    "PendingModifiedValues": {
      "MasterUserPassword": "*****"
    },
    "TagList": [
      {
        "Key": "TEST",
        "Value": "testValue"
      }
    ]
  }
}
```

下列範例描述美國東部 (維吉尼亞北部) 區域中資料庫執行個體 `my-cdb-inst` 上的租戶資料庫 `pdb1`。

Example

對於LinuxmacOS、或Unix：

```
aws rds describe-tenant-databases --region us-east-1 \
  --db-instance-identifier my-cdb-inst \
```

```
--tenant-db-name pdb1
```

在Windows中：

```
aws rds describe-tenant-databases --region us-east-1 ^  
--db-instance-identifier my-cdb-inst ^  
--tenant-db-name pdb1
```

此命令會產生類似下列的輸出。

```
{  
  "TenantDatabases" : [  
    {  
      "DbiResourceId" : "db-abc123",  
      "TenantDatabaseResourceId" : "tdb-bac567",  
      "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-  
inst:pdb1"  
      "DBInstanceIdentifier" : "my-cdb-inst",  
      "TenantDBName" : "pdb1",  
      "Status" : "ACTIVE",  
      "MasterUsername" : "masterawsuser"  
      "Port" : "1234",  
      "CharacterSetName": "UTF-8",  
      "ParameterGroups": [  
        {  
          "ParameterGroupName": "tenant-custom-pg",  
          "ParameterApplyStatus": "in-sync"  
        }  
      ],  
      {  
        "OptionGroupMemberships": [  
          {  
            "OptionGroupName": "tenant-custom-og",  
            "Status": "in-sync"  
          }  
        ]  
      }  
    }  
  ]  
}
```

升級 CDB

您可以將 CDB 升級到其他 Oracle Database 版本。例如，您可以將資料庫執行個體從 Oracle Database 19c 升級到 Oracle Database 21c。在升級期間無法變更資料庫架構。因此無法將非 CDB 升級到 CDB 或將 CDB 升級為非 CDB。

將 CDB 升級到 CDB 的程序與非 CDB 升級為非 CDB 的程序相同。如需更多詳細資訊，請參閱 [升級 RDS for Oracle 資料庫引擎](#)。

管理您的 RDS for Oracle 資料庫執行個體

以下是您可以使用 RDS for Oracle 資料庫執行個體來執行的一般管理任務。對於所有 RDS 資料庫執行個體，某些工作是相同的。其他任務是 RDS for Oracle 特定的。

以下任務適用於所有 RDS 資料庫，但 Oracle Database 具有特殊考量。例如，您可以使用 Oracle 從屬端 SQL*Plus 和 SQL 開發人員來連線到 Oracle 資料庫。

任務區域	相關文件
<p>執行個體類別、儲存體和 PIOPS</p> <p>如果您要建立生產執行個體，請了解執行個體類別、儲存體類型和佈建 IOPS 在 Amazon RDS 中的運作方式。</p>	<p>RDS for Oracle 執行個體類別</p> <p>Amazon RDS 儲存類型</p>
<p>異地同步備份部署</p> <p>生產資料庫執行個體應該使用多個可用區部署。「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。</p>	<p>設定及管理多可用區部署</p>
<p>Amazon VPC</p> <p>如果您的 AWS 帳戶有預設的虛擬私有雲端 (VPC)，那麼會在預設 VPC 內自動建立您的資料庫執行個體。如果您的帳戶沒有預設的 VPC，而您想要 VPC 中有該資料庫執行個體，則必須在建立執行個體之前，先建立 VPC 和子網路群組。</p>	<p>在 VPC 中使用資料庫執行個體</p>
<p>安全群組</p> <p>根據預設，資料庫執行個體會使用防火牆來防止存取。確定您以正確的 IP 地址和網路組態建立安全群組，來存取資料庫執行個體。</p>	<p>使用安全群組控制存取</p>
<p>參數群組</p> <p>如果您的資料庫執行個體將要求特定資料庫參數，則建立資料庫執行個體之前，須先建立參數群組。</p>	<p>使用參數群組</p>

任務區域	相關文件
<p>選項群組</p> <p>如果您的資料庫執行個體要求特定資料庫選項，則在建立資料庫執行個體之前，須先建立選項群組。</p> <p>連線到您的資料庫執行個體</p> <p>建立安全群組並與資料庫執行個體建立關聯後，您可以使用任何標準 SQL 用戶端應用程式 (例如 Oracle SQL *Plus) 來連接至資料庫執行個體。</p>	<p>將選項新增至 Oracle 資料庫執行個體</p> <p>連線至 RDS for Oracle 資料庫執行個體</p>
<p>備份和還原</p> <p>您可以設定資料庫執行個體以進行自動備份，或取得手動快照，然後從備份或快照還原執行個體。</p>	<p>備份、還原和匯出資料</p>
<p>監控</p> <p>您可以使用 CloudWatch Amazon RDS 指標、事件和增強型監控來監控 Oracle 資料庫執行個體。</p>	<p>在 Amazon RDS 主控台中檢視指標</p> <p>檢視 Amazon RDS 事件</p>
<p>日誌檔</p> <p>您可以存取 Oracle 資料庫執行個體の日誌檔案。</p>	<p>監控 Amazon RDS 日誌檔案</p>

接下來，您可以尋找 RDS Oracle 一般資料庫管理員任務的 Amazon RDS 特定實作。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。此外，RDS 也會將存取權限制在某些需要進階權限的系統程序和資料表。在許多任務中，您可以執行 rdsadmin 套件，這是一個可讓您管理資料庫的 Amazon RDS 特定工具。

使用執行 Oracle 的資料庫執行個體時，可能會處理到下列一般 DBA 任務：

- [系統任務](#)

[中斷工作階段的連線](#)

Amazon RDS 方法：rdsadmin.rdsadmin_util.disconnect

Oracle 方法：alter system disconnect session

終止工作階段	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.kill</code></p> <p>Oracle 方法 : <code>alter system kill session</code></p>
取消工作階段中的 SQL 陳述式	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.cancel</code></p> <p>Oracle 方法 : <code>alter system cancel sql</code></p>
啟用和停用受限制的工作階段	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.restricted_session</code></p> <p>Oracle 方法 : <code>alter system enable restricted session</code></p>
清空共用集區	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.flush_shared_pool</code></p> <p>Oracle 方法 : <code>alter system flush shared_pool</code></p>
清空緩衝區快取	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.flush_buffer_cache</code></p> <p>Oracle 方法 : <code>alter system flush buffer_cache</code></p>
將 SELECT 或 EXECUTE 權限授予 SYS 物件	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.grant_sys_object</code></p> <p>Oracle 方法 : <code>grant</code></p>
撤銷 SYS 物件上的 SELECT 或 EXECUTE 權限	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.revoke_sys_object</code></p> <p>Oracle 方法 : <code>revoke</code></p>
管理 Oracle 資料庫執行個體的 RDS_X 視觀表	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.create_sys_x\$view</code></p> <p>Oracle 方法 : <code>CREATE VIEW</code></p>
將權限授予非主要使用者	<p>Amazon RDS 方法 : <code>grant</code></p>

建立自訂函數以驗證密碼	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_password_verify.create_verify_function</code></p> <p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn</code></p>
設定自訂 DNS 伺服器	—
列出允許的系統診斷事件	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.list_allowed_system_events</code></p> <p>Oracle 方法 : —</p>
設定系統診斷事件	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.set_allowed_system_events</code></p> <p>Oracle 方法 : <code>ALTER SYSTEM SET EVENTS 'set_event_clause'</code></p>
列出已設定的系統診斷事件	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.list_set_system_events</code></p> <p>Oracle 方法 : <code>ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'</code></p>
取消設定系統診斷事件	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.unset_system_event</code></p> <p>Oracle 方法 : <code>ALTER SYSTEM SET EVENTS 'unset_event_clause'</code></p>

- [資料庫任務](#)

變更資料庫的全域名稱	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.rename_global_name</code></p> <p>Oracle 方法 : <code>alter database rename</code></p>
----------------------------	---

建立和調整資料表空間的大小	<p>Amazon RDS 方法 : <code>create tablespace</code></p> <p>Oracle 方法 : <code>alter database</code></p>
設定預設的資料表空間	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.alter_default_tablespace</code></p> <p>Oracle 方法 : <code>alter database default tablespace</code></p>
設定預設的暫存資料表空間	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.alter_default_temp_tablespace</code></p> <p>Oracle 方法 : <code>alter database default temporary tablespace</code></p>
在執行個體儲存體上建立暫存資料表空間	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.create_inst_store_tmp_tablespace</code></p> <p>Oracle 方法 : <code>create temporary tablespace</code></p>
設定資料庫的檢查點	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.checkpoint</code></p> <p>Oracle 方法 : <code>alter system checkpoint</code></p>
設定分散式復原	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.enable_distr_recovery</code></p> <p>Oracle 方法 : <code>alter system enable distributed recovery</code></p>
設定資料庫時區	<p>Amazon RDS 方法 : <code>rdsadmin.rdsadmin_util.alter_db_time_zone</code></p> <p>Oracle 方法 : <code>alter database set time_zone</code></p>
使用 Oracle 外部資料表	—

使用自動工作負載儲存庫 (AWR) 產生效能報告。	<p>Amazon RDS 方法：rdsadmin.rdsadmin_diagnostic_util 程序</p> <p>Oracle 方法：dbms_workload_repository 套件</p>
調整資料庫連結以搭配使用 VPC 中的資料庫執行個體	<p>—</p>
設定資料庫執行個體的預設版本	<p>Amazon RDS 方法：rdsadmin.rdsadmin_util.alter_default_edition</p> <p>Oracle 方法：alter database default edition</p>
為 SYS.AUD\$ 資料表啟用稽核	<p>Amazon RDS 方法：rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table</p> <p>Oracle 方法：audit</p>
為 SYS.AUD\$ 資料表停用稽核	<p>Amazon RDS 方法：rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table</p> <p>Oracle 方法：noaudit</p>
清除中斷的線上索引建置	<p>Amazon RDS 方法：rdsadmin.rdsadmin_dbms_repair.online_index_clean</p> <p>Oracle 方法：dbms_repair.online_index_clean</p>
略過損毀的區塊	<p>Amazon RDS 方法：數個 rdsadmin.rdsadmin_dbms_repair 程序</p> <p>Oracle 方法：dbms_repair 套件</p>

調整資料表空間、資料檔案和暫存檔的大小

Amazon RDS 方法：`rdsadmin.rdsadmin_util.resize_temp_tablespace`、`rdsadmin.rdsadmin_util.resize_tempfile` 或 `rdsadmin.rdsadmin_util.autoextend_tempfile` 程序

`rdsadmin.rdsadmin_util.resize_datafile` 或 `rdsadmin.rdsadmin_util.autoextend_datafile` 程序

Oracle 方法：—

清除資源回收筒

Amazon RDS 方法：`EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin`

Oracle 方法：`purge dba_recyclebin`

設定完全修訂的預設顯示值

Amazon RDS 方法：`EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val`

Oracle 方法：`exec dbms_redact.UPDATE_FULL_RED ACTION_VALUES`

• 日誌任務

設定強制記錄

Amazon RDS 方法：`rdsadmin.rdsadmin_util.force_logging`

Oracle 方法：`alter database force logging`

設定補充記錄	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.alter_supplemental_logging</code></p> <p>Oracle 方法：<code>alter database add supplemental log</code></p>
切換線上日誌檔案	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.switch_logfile</code></p> <p>Oracle 方法：<code>alter system switch logfile</code></p>
新增線上重做日誌	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.add_logfile</code></p>
捨棄線上重做日誌	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.drop_logfile</code></p>
調整線上重做日誌的大小	<p>—</p>
保留封存的重做日誌	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.set_configuration</code></p>

[從 Amazon S3 下載封存的 redo 日誌](#)

Amazon RDS 方法：`rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`

Amazon RDS 方法：`rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range`

[存取線上和封存的 redo 日誌](#)

Amazon RDS 方法：`rdsadmin.rdsadmin_master_util.create_archive_log_dir`

Amazon RDS 方法：`rdsadmin.rdsadmin_master_util.create_online_log_dir`

- [RMAN 任務](#)

[在 RDS 中為甲骨文驗證數據庫文件](#)

Amazon RDS 方法：`rdsadmin.rman_util.procedure`

Oracle 方法：`RMAN VALIDATE`

啟用和停用區塊變更追蹤功能	<p>Amazon RDS 方法：<code>rdsadmin_rman_util.procedure</code></p> <p>Oracle 方法：ALTER DATABASE</p>
交叉檢查封存的 redo 日誌	<p>Amazon RDS 方法：<code>rdsadmin_rman_util.crosscheck_archive_log</code></p> <p>Oracle 方法：RMAN BACKUP</p>
備份存檔 redo 日誌檔	<p>Amazon RDS 方法：<code>rdsadmin_rman_util.procedure</code></p> <p>Oracle 方法：RMAN BACKUP</p>
執行完整資料庫備份	<p>Amazon RDS 方法：<code>rdsadmin_rman_util.backup_database_full</code></p> <p>Oracle 方法：RMAN BACKUP</p>
執行增量資料庫備份	<p>Amazon RDS 方法：<code>rdsadmin_rman_util.backup_database_incremental</code></p> <p>Oracle 方法：RMAN BACKUP</p>

[備份資料表空間](#)

Amazon RDS 方法：
rdsadmin_rman_util.backup_database_tablespace

Oracle 方法：RMAN BACKUP

- [Oracle Scheduler 任務](#)

[修改 DBMS_SCHEDULER 任務](#)

Amazon RDS 方法：
dbms_scheduler.set_attribute

Oracle 方法：dbms_scheduler.set_attribute

[修改 AutoTask 維護時段](#)

Amazon RDS 方法：
dbms_scheduler.set_attribute

Oracle 方法：dbms_scheduler.set_attribute

[設定 Oracle Scheduler 任務的時區](#)

Amazon RDS 方法：
dbms_scheduler.set_scheduler_attribute

Oracle 方法：dbms_scheduler.set_scheduler_attribute

<p>關閉 SYS 擁有的 Oracle Scheduler 任務</p>	<p>Amazon RDS 方法： rdsadmin. rdsadmin_dbms_scheduler.disable</p> <p>Oracle 方法： dbms_scheduler.disable</p>
<p>開啟 SYS 擁有的 Oracle Scheduler 任務</p>	<p>Amazon RDS 方法： rdsadmin. rdsadmin_dbms_scheduler.enable</p> <p>Oracle 方法： dbms_scheduler.enable</p>
<p>針對 CALENDAR 類型的任務，修改 Oracle Scheduler 重複間隔</p>	<p>Amazon RDS 方法： rdsadmin. rdsadmin_dbms_scheduler.set_attribute</p> <p>Oracle 方法： dbms_scheduler.set_attribute</p>
<p>針對 NAMED 類型的任務，修改 Oracle Scheduler 重複間隔</p>	<p>Amazon RDS 方法： rdsadmin. rdsadmin_dbms_scheduler.set_attribute</p> <p>Oracle 方法： dbms_scheduler.set_attribute</p>

[建立 Oracle Scheduler 任務時關閉自動認可](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_dbms_scheduler.set_no_commit_flag

Oracle 方法：
dbms_isched.set_no_commit_flag

- [診斷工作](#)

[列出事件](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_adrci_util.list_adrci_incidents

Oracle 方法：
ADRCI 命令
show incident

[列出問題](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_adrci_util.list_adrci_problem

Oracle 方法：
ADRCI 命令
show problem

[建立事件套件](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_adrci_util.
create_adrci_package

Oracle 方法：ADRCI 命令
ips create package

[顯示追蹤檔案](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_adrci_util.
show_adrci_tracefile

Oracle 方法：ADRCI 命令
show tracefile

- [其他任務](#)

[在主要資料儲存空間中建立和捨棄目錄](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_util.
create_directory

Oracle 方法：CREATE
DIRECTORY

Amazon RDS 方法：
rdsadmin.
rdsadmin_util.
drop_directory

Oracle 方法：DROP
DIRECTORY

列出資料庫執行個體目錄中的檔案	<p>Amazon RDS 方法：<code>rdsadmin.rds_file_util.list_dir</code></p> <p>Oracle 方法：—</p>
讀取資料庫執行個體目錄中的檔案	<p>Amazon RDS 方法：<code>rdsadmin.rds_file_util.read_text_file</code></p> <p>Oracle 方法：—</p>
存取 Opatch 檔案	<p>Amazon RDS 方法：<code>rdsadmin.rds_file_util.read_text_file</code> 或 <code>rdsadmin.tracefile_listing</code></p> <p>Oracle 方法：<code>opatch</code></p>
設定建議程式任務的參數	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.advisor_task_set_parameter</code></p> <p>Oracle 方法：各種存放的套件程序</p>
停用 AUTO_STATS_ADVISOR_TASK	<p>Amazon RDS 方法：<code>rdsadmin.rdsadmin_util.advisor_task_drop</code></p> <p>Oracle 方法：—</p>

[重新啟用 AUTO_STATS_ADVISOR_TASK](#)

Amazon RDS 方法：
rdsadmin.
rdsadmin_util.dbms
_stats_init

Oracle 方法：—

您也可以使用 Amazon RDS 程序讓 Amazon S3 整合 Oracle，以及執行 OEM Management Agent 資料庫任務。如需更多詳細資訊，請參閱 [Amazon S3 整合](#) 及 [使用 Management Agent 執行資料庫任務](#)。

執行 Oracle 資料庫執行個體的一般系統任務

您可以在下文中找到如何在執行 Oracle 的 Amazon RDS 資料庫執行個體上執行與系統相關的一般 DBA 任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的 Shell 存取權，而且會將存取權限制在某些需要進階權限的系統程序和資料表。

主題

- [中斷工作階段的連線](#)
- [終止工作階段](#)
- [取消工作階段中的 SQL 陳述式](#)
- [啟用和停用受限制的工作階段](#)
- [清空共用集區](#)
- [清空緩衝區快取](#)
- [排清資料庫智慧型快閃記憶體快取](#)
- [將 SELECT 或 EXECUTE 權限授予 SYS 物件](#)
- [撤銷 SYS 物件上的 SELECT 或 EXECUTE 權限](#)
- [管理 Oracle 資料庫執行個體的 RDS_X 視觀表](#)
- [將權限授予非主要使用者](#)
- [建立自訂函數以驗證密碼](#)
- [設定自訂 DNS 伺服器](#)
- [設定和取消設定系統診斷事件](#)

中斷工作階段的連線

若要透過結束專屬的伺服器程序來中斷與目前工作階段的連線，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.disconnect`。`disconnect` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>sid</code>	數字	—	是	工作階段識別符。
<code>serial</code>	數字	—	是	工作階段的序號。
<code>method</code>	<code>varchar</code>	'IMMEDIATE'	否	有效值為 'IMMEDIATE' 或 'POST_TRANSACTION'。

下列範例會中斷工作階段的連線。

```
begin
  rdsadmin.rdsadmin_util.disconnect(
    sid    => sid,
    serial => serial_number);
end;
/
```

若要取得工作階段識別符和工作階段序號，請查詢 `V$SESSION` 檢視。下列範例會取得使用者 `AWSUSER` 的所有工作階段。

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

請務必開啟資料庫，才能使用此方法。如需中斷工作階段連線的詳細資訊，請參閱 Oracle 文件中的 [ALTER SYSTEM](#)。

終止工作階段

若要終止工作階段，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.kill`。`kill` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
sid	數字	—	是	工作階段識別符。
serial	數字	—	是	工作階段的序號。
method	varchar	null	否	<p>有效值為 'IMMEDIATE' 或 'PROCESS'。若您指定 IMMEDIATE，它與執行下列陳述式具有相同的效果：</p> <pre>ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE</pre> <p>若您指定 PROCESS，則會終止與工作階段相關聯的程序。若使用 IMMEDIATE 終止工作階段未成功，則僅指定 PROCESS。</p>

若要取得工作階段識別符和工作階段序號，請查詢 V\$SESSION 檢視。下列範例會取得使用者的所有工作階段 *AWSUSER*。

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

下列範例會終止工作階段。

```
BEGIN
  rdsadmin.rdsadmin_util.kill(
    sid    => sid,
    serial => serial_number,
    method => 'IMMEDIATE');
END;
/
```

下列範例會終止與工作階段相關聯的程序。

```
BEGIN
  rdsadmin.rdsadmin_util.kill(
    sid    => sid,
    serial => serial_number,
    method => 'PROCESS');
END;
/
```

取消工作階段中的 SQL 陳述式

若要取消工作階段中的 SQL 陳述式，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.cancel`。

Note

Oracle Database 19c (19.0.0) 和 Oracle RDS 的所有較新的主要版本和次要版本支援此程序。

`cancel` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>sid</code>	數字	—	是	工作階段識別符。
<code>serial</code>	數字	—	是	工作階段的序號。
<code>sql_id</code>	<code>varchar2</code>	<code>null</code>	否	SQL 陳述式的 SQL 識別符。

下列範例會取消工作階段中的 SQL 陳述式。

```
begin
  rdsadmin.rdsadmin_util.cancel(
    sid    => sid,
    serial => serial_number,
    sql_id => sql_id);
end;
```

/

若要取得工作階段識別符、工作階段序號和 SQL 陳述式的 SQL 識別符，請查詢 V\$SESSION 檢視。下列範例會取得使用者 AWSUSER 的所有工作階段和 SQL 識別碼。

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

啟用和停用受限制的工作階段

若要啟用和停用受限制的工作階段，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.restricted_session`。`restricted_session` 程序具有下列參數。

參數名稱	資料類型	預設	是	描述
<code>p_enable</code>	布林值	<code>true</code>	否	設定為 <code>true</code> 可啟用受限制的工作階段，設定為 <code>false</code> 則可停用受限制的工作階段。

下列範例顯示如何啟用和停用受限制的工作階段。

```
/* Verify that the database is currently unrestricted. */
SELECT LOGINS FROM V$INSTANCE;

LOGINS
-----
ALLOWED

/* Enable restricted sessions */
EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => true);

/* Verify that the database is now restricted. */
SELECT LOGINS FROM V$INSTANCE;
```

```
LOGINS
-----
RESTRICTED

/* Disable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => false);

/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-----
ALLOWED
```

清空共用集區

若要清空共用集區，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.flush_shared_pool`。 `flush_shared_pool` 程序沒有參數。

下列範例會清空共用集區。

```
EXEC rdsadmin.rdsadmin_util.flush_shared_pool;
```

清空緩衝區快取

若要清空緩衝區快取，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.flush_buffer_cache`。 `flush_buffer_cache` 程序沒有參數。

下列範例會清空緩衝區快取。

```
EXEC rdsadmin.rdsadmin_util.flush_buffer_cache;
```

排清資料庫智慧型快閃記憶體快取

若要排清資料庫智慧型快閃記憶體快取，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.flush_flash_cache`。 `flush_flash_cache` 程序沒有參數。下列範例會排清資料庫智慧型快閃記憶體快取。

```
EXEC rdsadmin.rdsadmin_util.flush_flash_cache;
```

如需搭配 RDS for Oracle 使用資料庫智慧型快閃記憶體快取的詳細資訊，請參閱 [將暫存資料存放在 RDS for Oracle 執行個體儲存體中](#)。

將 SELECT 或 EXECUTE 權限授予 SYS 物件

您通常可以使用角色來傳輸權限，角色中可能包含許多物件。若要授予權限給單一物件，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.grant_sys_object`。該程序只會授予主要使用者透過角色或直接授予所取得的權限。

`grant_sys_object` 程序具有下列參數。

Important

針對所有參數值，除非您已使用區分大小寫的識別符來建立使用者，否則請使用大寫字母。例如，如果您執行 `CREATE USER myuser` 或 `CREATE USER MYUSER`，資料字典會存放 `MYUSER`。不過，如果您在 `CREATE USER "MyUser"` 中使用雙引號，資料字典則會存放 `MyUser`。

參數名稱	資料類型	預設	必要	描述
<code>p_obj_name</code>	<code>varchar2</code>	—	是	要從其授予權限的來源物件名稱。物件可以是目錄、函數、套件、程序、順序、資料表或檢視。物件名稱的拼寫必須完全符合 <code>DBA_OBJECTS</code> 中的顯示內容。多數系統物件是以大寫定義，因此建議您先嘗試採用大寫。
<code>p_grantee</code>	<code>varchar2</code>	—	是	要對其授予權限的目標物件名稱。物件可以是結構描述或角色。
<code>p_privilege</code>	<code>varchar2</code>	<code>null</code>	是	—

參數名稱	資料類型	預設	必要	描述
p_grant_option	布林值	false	否	設定為 true 以使用 WITH GRANT OPTION。

下列範例會將名為 V_\$SESSION 物件上的 SELECT 權限授予名為 USER1 的使用者。

```
begin
  rdsadmin.rdsadmin_util.grant_sys_object(
    p_obj_name => 'V_$SESSION',
    p_grantee  => 'USER1',
    p_privilege => 'SELECT');
end;
/
```

下列範例會將名為 V_\$SESSION 物件上的 SELECT 權限授予名為 USER1 且擁有 GRANT OPTION 的使用者。

```
begin
  rdsadmin.rdsadmin_util.grant_sys_object(
    p_obj_name      => 'V_$SESSION',
    p_grantee       => 'USER1',
    p_privilege     => 'SELECT',
    p_grant_option  => true);
end;
/
```

您必須使用 GRANT OPTION 或透過使用 with admin option 授予的角色直接將這些權限授予帳戶，才能夠在物件上授予權限。在最常見的情況下，您可能需要在獲授予 SELECT 角色的 DBA 檢視上授予 SELECT_CATALOG_ROLE。如果您尚未使用 with admin option 將該角色直接授予使用者，則無法轉移權限。如果您擁有 DBA 權限，就能夠將該角色直接授予另一位使用者。

下列範例會授予 SELECT_CATALOG_ROLE 和 EXECUTE_CATALOG_ROLE 給 USER1。由於使用了 with admin option，USER1 現在可以將存取權授予給已獲授予 SELECT_CATALOG_ROLE 的 SYS 物件。

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

已獲授予 PUBLIC 的物件不需要重新授予。如果使用 `grant_sys_object` 程序來重新授予存取，則程序呼叫會成功。

撤銷 SYS 物件上的 SELECT 或 EXECUTE 權限

若要撤銷單一物件上的權限，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.revoke_sys_object`。該程序只會撤銷已透過角色或直接授與主帳戶的權限。

`revoke_sys_object` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_obj_name</code>	<code>varchar2</code>	—	是	要撤銷其權限的物件名稱。物件可以是目錄、函數、套件、程序、順序、資料表或檢視。物件名稱的拼寫必須完全符合 <code>DBA_OBJECTS</code> 中的顯示內容。多數系統物件是以大寫定義，因此建議您先嘗試採用大寫。
<code>p_revokee</code>	<code>varchar2</code>	—	是	要撤銷其權限的物件名稱。物件可以是結構描述或角色。
<code>p_privilege</code>	<code>varchar2</code>	<code>null</code>	是	—

下列範例會從名為 `V_$SESSION` 的使用者撤銷名為 `USER1` 物件上的 `SELECT` 權限。

```
begin
  rdsadmin.rdsadmin_util.revoke_sys_object(
    p_obj_name => 'V_$SESSION',
    p_revokee  => 'USER1',
    p_privilege => 'SELECT');
end;
/
```


管理 Oracle 資料庫執行個體的 RDS_X 視觀表

您可能需要存取SYS.X\$固定資料表，這些資料表只能由存取SYS。若要在合格的資X\$料表上建立SYS.RDS_X\$檢視，請使用rdsadmin.rdsadmin_util封裝中的程序。系統會自動授予您的主要用戶SELECT ... WITH GRANT OPTION對RDS_X\$視圖的權限。

這些程rdsadmin.rdsadmin_util序適用於下列資料庫引擎版本：

- 21.0.0.0.ru-2023-10.rur-2023-10.r1和更高的甲骨文數據庫 21c 版本
- 19.0.0.0.ru-2023-10.rur-2023-10.r1和更高的甲骨文數據庫 19c 版本

Important

在內部，rdsadmin.rdsadmin_util封裝會在X\$資料表上建立檢視。這些X\$表格是「Oracle 資料庫」說明文件中未說明的內部系統物件。我們建議您在非生產資料庫中測試特定檢視，並且只在 Oracle 客戶 Support 務部的指導下，在生產資料庫中建立檢視。

列出可在 RDS_X \$ 檢視中使用的 X\$ 固定資料表

若要列出有資格在RDS_X\$檢視中使用的 X\$ 資料表，請使用 RDS 程序rdsadmin.rdsadmin_util.list_allowed_sys_x\$_views。此程序不接受任何參數。下列陳述式會列出所有符合資格的資X\$料表 (包括範例輸出)。

```
SQL> SET SERVEROUTPUT ON
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_allowed_sys_x$_views);

'X$BH'
'X$K2GTE'
'X$KCBWBDP'
'X$KCBWDS'
'X$KGLLK'
'X$KGLOBAL'
'X$KGLPN'
'X$KSLHOT'
'X$KSMSPP'
'X$KSPPCV'
'X$KSPPPI'
'X$KSPPSV'
'X$KSQEQ'
```

```
'X$KSQRS'
'X$KTUXE'
'X$KQRF'P'
```

符合條件的X\$表格清單可能會隨時間變更。若要確定符合資格的X\$固定表格清單是最新的，請執行 `list_allowed_sys_x$_views` 定期重新執行。

建立系統視圖

若要在合格的資料X\$表上建立RDS_X\$檢視，請使用 RDS 程序 `rdsadmin.rdsadmin_util.create_sys_x$_view`。您只能為的輸出中列出的表格建立檢視 `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`。`create_sys_x$_view` 程序接受下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_x\$_tbl</code>	<code>varchar2</code>	Null	是	有效的X\$資料表名稱。該值必須是報告的其中一個X\$表格 <code>list_allowed_sys_x\$_views</code> 。
<code>p_force_creation</code>	Boolean	FALSE	否	指出是否強制建立資料表已存在的RDS_X\$檢視X\$表的值。根據預設，如果RDS已存在，則不會建立檢視。若要強制建立，請將此參數設定為TRUE。

下列範例會在資料表上建立 `SYS.RDS_X$KGLOBAL` 檢視 `X$KGLOBAL`。視圖名稱的格式為 `RDS_X$tablename`。

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.create_sys_x$_view('X$KGLOBAL');

PL/SQL procedure successfully completed.
```

下列資料字典查詢會列出檢視SYS.RDS_X\$KGLOBAL並顯示其狀態。您的主要使用者會自動授與此檢視SELECT ... WITH GRANT OPTION的權限。

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOBAL';
```

OWNER	OBJECT_NAME	STATUS
SYS	RDS_X\$KGLOBAL	VALID

Important

X\$不保證表在升級之前和之後保持不變。RDS for Oracle 會在引擎升級期間放棄並重新建立X\$資料表上的RDS_X\$檢視。然後，它將SELECT ... WITH GRANT OPTION權限授予主用戶。升級之後，視需要授與對應RDS_X\$檢視的權限給資料庫使用者。

列出 X 視圖

若要列出現有的RDS_X\$檢視，請使用 RDS 程

序rdsadmin.rdsadmin_util.list_created_sys_x\$_views。此程序只會列出程序所建立的視圖create_sys_x\$_view。下列範例會列出具有對應RDS_X\$檢視X\$表 (包含範例輸出) 的資料表。

```
SQL> SET SERVEROUTPUT ON
SQL> COL XD_TBL_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_created_sys_x$_views);
```

XD_TBL_NAME	STATUS
X\$BH	VALID
X\$K2GTE	VALID
X\$KCBWBD	VALID

```
3 rows selected.
```

刪除 RDS_X\$ 視圖

若要刪除SYS.RDS_X\$檢視，請使用 RDS 程序rdsadmin.rdsadmin_util.drop_sys_x\$_view。您只能刪除的輸出中列示的視圖rdsadmin.rdsadmin_util.list_allowed_sys_x\$_views。drop_sys_x\$_view 程序接受下列參數。

參數名稱	資料類型	預設	必要	描述
p_x\$_tbl	varchar2	Null	是	有效的X\$固定資料表名稱。該值必須是其中一個由報告的X\$固定表格list_created_sys_x\$_views。

下面的例子刪除，這是在表上創建的RDS_X\$KGLOBAL視圖X\$KGLOBAL。

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.drop_sys_x$_view('X$KGLOBAL');

PL/SQL procedure successfully completed.
```

下列範例顯示SYS.RDS_X\$KGLOBAL已卸除檢視 (包含範例輸出)。

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOBAL';

no rows selected
```

將權限授予非主要使用者

您可以使用 SYS 角色，將選取權限授予 SELECT_CATALOG_ROLE 結構描述中的許多物件。SELECT_CATALOG_ROLE 角色可提供使用者資料字典檢視上的 SELECT 權限。下列範例會將角色 SELECT_CATALOG_ROLE 授予名為 `user1` 的使用者。

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

您可以使用 EXECUTE 角色，將 SYS 權限授予 EXECUTE_CATALOG_ROLE 結構描述中的許多物件。EXECUTE_CATALOG_ROLE 角色可為使用者提供資料字典中套件和程序的 EXECUTE 權限。下列範例會將 EXECUTE_CATALOG_ROLE 角色授予名為「`user1`」的使用者。

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

下列範例會取得 SELECT_CATALOG_ROLE 角色和 EXECUTE_CATALOG_ROLE 角色允許的許可。

```
SELECT *
  FROM ROLE_TAB_PRIVS
 WHERE ROLE IN ('SELECT_CATALOG_ROLE', 'EXECUTE_CATALOG_ROLE')
 ORDER BY ROLE, TABLE_NAME ASC;
```

下列範例會建立名為 `user1` 的非主要使用者，並授予其 CREATE SESSION 權限，以及名為 `sh.sales` 資料庫上的 SELECT 權限。

```
CREATE USER user1 IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO user1;
GRANT SELECT ON sh.sales TO user1;
```

建立自訂函數以驗證密碼

您可以透過以下方式建立自訂密碼驗證函數。

- 若要使用標準驗證邏輯，以及在 SYS 結構描述中存放您的函數，請使用 `create_verify_function` 程序。
- 若要使用自訂驗證邏輯，或避免在 SYS 結構描述中存放您的函數，請使用 `create_passthrough_verify_fcn` 程序。

create_verify_function 程序

您可以建立自訂函數以使用 Amazon RDS 程序

`rdsadmin.rdsadmin_password_verify.create_verify_function` 來驗證密碼。

該 `create_verify_function` 程序適用於甲骨文的所有版本的 RDS 都支持。

`create_verify_function` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_verify_function_name</code>	varchar2	—	是	自訂函數的名稱。系統會在 SYS 結構描述中為您建立此函數。您可以將此函數指派給使用者描述檔。
<code>p_min_length</code>	數字	8	否	需要的字元數下限。
<code>p_max_length</code>	數字	256	否	允許的字元數上限。
<code>p_min_letters</code>	數字	1	否	需要的字母數下限。
<code>p_min_uppercase</code>	數字	0	否	需要的大寫字母數下限。
<code>p_min_lowercase</code>	數字	0	否	需要的小寫字母數下限。
<code>p_min_digits</code>	數字	1	否	需要的數字下限。
<code>p_min_special</code>	數字	0	否	需要的特殊字元數下限。
<code>p_min_different_chars</code>	數字	3	否	新舊密碼間需要的不同字元數下限。
<code>p_disallow_username</code>	布林值	true	否	設定為 true 可禁止在密碼中使用使用者名稱。
<code>p_disallow_reverse</code>	布林值	true	否	設定為 true 可禁止在密碼中使用倒寫的使用者名稱。

參數名稱	資料類型	預設	必要	描述
p_disallow_db_name	布林值	true	否	設定為 true 可禁止在密碼中使用資料庫或伺服器名稱。
p_disallow_simple_strings	布林值	true	否	設定為 true 可禁止使用簡易字串做為密碼。
p_disallow_whitespace	布林值	false	否	設定為 true 可禁止在密碼中使用空白字元。
p_disallow_at_sign	布林值	false	否	設定為 true 可禁止在密碼中使用 @ 字元。

您可以建立多個密碼驗證函數。

自訂函數的名稱有一些限制。您的自訂函數名稱不能與現有系統物件相同，且該名稱長度不得超過 30 個字元。此外，名稱需包括下列其中一個字串：PASSWORD、VERIFY、COMPLEXITY、ENFORCE 或 STRENGTH。

下列範例會建立名為 CUSTOM_PASSWORD_FUNCTION 的函數。該函數要求密碼至少有 12 個字元、2 個大寫字元、1 個數字和 1 個特殊字元，並且密碼禁止使用 @ 字元。

```
begin
  rdsadmin.rdsadmin_password_verify.create_verify_function(
    p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
    p_min_length           => 12,
    p_min_uppercase       => 2,
    p_min_digits          => 1,
    p_min_special         => 1,
    p_disallow_at_sign    => true);
end;
/
```

若要查看您驗證函數的文字，請查詢 DBA_SOURCE。下列範例會取得名為 CUSTOM_PASSWORD_FUNCTION 的自訂密碼函數的文字。

```
COL TEXT FORMAT a150
```

```
SELECT TEXT
FROM DBA_SOURCE
WHERE OWNER = 'SYS'
AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
ORDER BY LINE;
```

若要將您的驗證函數與使用者描述檔建立關聯，請使用 `alter profile`。下列範例會將驗證函數與 `DEFAULT` 使用者描述檔建立關聯。

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

若要查看什麼使用者描述檔已與什麼驗證函數建立關聯，請查詢 `DBA_PROFILES`。下列範例會取得已與名為 `CUSTOM_PASSWORD_FUNCTION` 的自訂驗證函數關聯的設定檔。

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
'CUSTOM_PASSWORD_FUNCTION';
```

PROFILE	RESOURCE_NAME	RESOURCE	LIMIT
-----	-----	-----	

DEFAULT	PASSWORD_VERIFY_FUNCTION	PASSWORD	
CUSTOM_PASSWORD_FUNCTION			

下列範例會取得所有設定檔和其關聯的密碼驗證函數。

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';
```

PROFILE	RESOURCE_NAME	RESOURCE	LIMIT
-----	-----	-----	

DEFAULT	PASSWORD_VERIFY_FUNCTION	PASSWORD	
CUSTOM_PASSWORD_FUNCTION			
RDSADMIN	PASSWORD_VERIFY_FUNCTION	PASSWORD	NULL

create_passthrough_verify_fcn 程序

該 `create_passthrough_verify_fcn` 程序適用於甲骨文的所有版本的 RDS 都支持。

您可以建立自訂函數以使用 Amazon RDS 程序

`rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn` 來驗證密碼。 `create_passthrough_verify_fcn` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_verify_function_name	varchar2	—	是	自訂驗證函數的名稱。這是系統為您在 SYS 結構描述中所建立的包裝函數，並且不包含任何驗證邏輯。您可以將此函數指派給使用者描述檔。
p_target_owner	varchar2	—	是	自訂驗證函數的結構描述擁有者。
p_target_function_name	varchar2	—	是	現有的自訂函數名稱，其中包含驗證邏輯。您的自訂函數必須傳回布林值。如果密碼有效，您的函數應該傳回 true，以及如果密碼無效，則傳回 false。

下列範例會使用來自名為 PASSWORD_LOGIC_EXTRA_STRONG 函數的邏輯，來建立密碼驗證函數。

```
begin
  rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
    p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
    p_target_owner         => 'TEST_USER',
    p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

若要將驗證函數與使用者描述檔建立關聯，請使用 alter profile。下列範例會將驗證函數與 DEFAULT 使用者描述檔建立關聯。

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

設定自訂 DNS 伺服器

Amazon RDS 在您執行 Oracle 的資料庫執行個體上支援傳出網路存取。如需傳出網路存取的詳細資訊 (包括先決條件)，請參閱 [使用憑證和 Oracle 錢包設定 UTL_HTTP 存取](#)。

Amazon RDS Oracle 允許透過客戶擁有的自訂 DNS 伺服器進行網域名稱服務 (DNS) 解析。您僅可以透過您的自訂 DNS 伺服器，解析來自您 Amazon RDS 資料庫執行個體的完整網域名稱。

設定您的自訂 DNS 名稱伺服器之後，需要最多 30 分鐘的時間，才能將變更傳播到您的資料庫執行個體。變更傳播到您的資料庫執行個體之後，所有傳出網路流量都需要透過連接埠 53 對您的 DNS 伺服器進行 DNS 查閱查詢。

若要為您的 Amazon RDS for Oracle 資料庫執行個體設定自訂的 DNS 伺服器，請執行下列動作：

- 從連接至您 Virtual Private Cloud (VPC) 的 DHCP 選項集，將 `domain-name-servers` 選項設定為您 DNS 名稱伺服器的 IP 地址。如需詳細資訊，請參閱 [DHCP 選項集](#)。

Note

`domain-name-servers` 選項會接受最多四個值，但您的 Amazon RDS 資料庫執行個體只會使用第一個值。

- 請確保您的 DNS 伺服器可以解析所有查閱查詢，包括公有 DNS 名稱、Amazon EC2 私有 DNS 名稱，以及客戶特定的 DNS 名稱。如果傳出網路流量包含您的 DNS 伺服器無法處理的任何 DNS 查閱，您的 DNS 伺服器就必須設定適當的上游 DNS 提供者。
- 設定您的 DNS 伺服器以產生 512 個位元組或更少的使用者資料包通訊協定 (UDP) 回應。
- 設定您的 DNS 伺服器以產生 1024 個位元組或更少的傳輸控制通訊協定 (TCP) 回應。
- 設定您的 DNS 伺服器以允許來自您的 Amazon RDS 資料庫執行個體透過連接埠 53 的傳入流量。如果您的 DNS 伺服器位於 Amazon VPC 中，VPC 必須具有一個安全群組，其中包含允許連接埠 53 上的 UDP 和 TCP 流量的傳入規則。如果您的 DNS 伺服器不在 Amazon VPC 中，則必須具有適當的防火牆允許清單，才能允許連接埠 53 上的 UDP 和 TCP 傳入流量。

如需更多詳細資訊，請參閱 [VPC 安全群組](#) 和 [新增與移除規則](#)。

- 設定您 Amazon RDS 資料庫執行個體的 VPC，以允許透過連接埠 53 的傳出流量。您的 VPC 必須具有一個安全群組，其中包含允許連接埠 53 上的 UDP 和 TCP 流量的傳出規則。

如需更多詳細資訊，請參閱 [VPC 安全群組](#) 和 [新增與移除規則](#)。

- 您必須正確設定 Amazon RDS 資料庫執行個體和 DNS 伺服器之間的路由路徑，才能允許 DNS 流量。

- 如果 Amazon RDS 資料庫執行個體和 DNS 伺服器不在相同的 VPC 中，則必須在兩者之間設定對等連接。如需詳細資訊，請參閱[什麼是 VPC 互連？](#)

設定和取消設定系統診斷事件

若要在工作階段層級設定和取消設定診斷事件，您可以使用 Oracle SQL 陳述式 ALTER SESSION SET EVENTS。但是，要在系統層級設定事件，您不能使用 Oracle SQL。而是使用 rdsadmin.rdsadmin_util 套件中的系統事件程序。下列引擎版本提供系統事件程序：

- 所有 Oracle Database 21c 版本
- 19.0.0.0.ru-2020-10.rur-2020-10.r1 及更新的 Oracle Database 19c 版本

如需詳細資訊，請參閱 Amazon RDS for Oracle 發行說明中的[版本 19.0.0.0.0.ru-2020-10.r1](#)

Important

在內部，rdsadmin.rdsadmin_util 套件會使用 ALTER SYSTEM SET EVENTS 陳述式來設定事件。此 ALTER SYSTEM 陳述式並未記錄在 Oracle 資料庫文件中。某些系統診斷事件可能會產生大量的追蹤資訊、造成爭用或影響資料庫可用性。建議您在非生產資料庫中測試特定的診斷事件，並且只在 Oracle Support 的指導下，在生產資料庫中設定事件。

列出允許的系統診斷事件

若要列出您可以設定的系統事件，請使用 Amazon RDS Procedure (程序)rdsadmin.rdsadmin_util.list_allowed_system_events。此程序不接受任何參數。

下列範例會列出您可以設定的所有系統事件。

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

下列範例輸出列出事件編號及其描述。使用 Amazon RDS 程序 set_system_event 來設定這些事件並 unset_system_event 以取消這些設定。

```
604 - error occurred at recursive SQL level
942 - table or view does not exist
```

```

1401 - inserted value too large for column
1403 - no data found
1410 - invalid ROWID
1422 - exact fetch returns more than requested number of rows
1426 - numeric overflow
1427 - single-row subquery returns more than one row
1476 - divisor is equal to zero
1483 - invalid length for DATE or NUMBER bind variable
1489 - result of string concatenation is too long
1652 - unable to extend temp segment by in tablespace
1858 - a non-numeric character was found where a numeric was expected
4031 - unable to allocate bytes of shared memory ("","","","")
6502 - PL/SQL: numeric or value error
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object failed to load/unload and is being skipped due to error:

```

Note

允許的系統事件清單可能會隨著時間而變更。若要確定您擁有最新的符合資格事件的清單，請使用 `rdsadmin.rdsadmin_util.list_allowed_system_events`。

設定系統診斷事件

若要設定系統事件，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.set_system_event`。您只能設定

`rdsadmin.rdsadmin_util.list_allowed_system_events` 的輸出中列出的事

件。`set_system_event` 程序接受下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_event</code>	數字	—	是	系統事件編號。該值必須是由 <code>list_allo</code>

參數名稱	資料類型	預設	必要	描述
				wed_system_events 報告的其中一個事件編號。
p_level	數字	—	是	事件層級。如需不同層級值的說明，請參閱 Oracle 資料庫文件或 Oracle 支援。

程序 `set_system_event` 會根據下列原則建構並執行所需的 ALTER SYSTEM SET EVENTS 陳述式：

- 自動決定事件類型 (context 或 errorstack)。
- 表單 ALTER SYSTEM SET EVENTS '*event* LEVEL *event_level*' 中的陳述式會設定內容事件。此標記法等同於 ALTER SYSTEM SET EVENTS '*event* TRACE NAME CONTEXT FOREVER, LEVEL *event_level*'。
- 表單中 ALTER SYSTEM SET EVENTS '*event* ERRORSTACK (*event_level*)' 的陳述式會設定錯誤堆疊事件。此標記法等同於 ALTER SYSTEM SET EVENTS '*event* TRACE NAME ERRORSTACK LEVEL *event_level*'。

下列範例會在層級 3 設定事件 942，並在層級 10 設定事件 10442。包括範例輸出。

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.
```

列出已設定的系統診斷事件

若要列出目前設定的系統事件，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.list_set_system_events`。此程序只會報告在系統層級由 `set_system_event` 設定的事件。

下列範例會列出作用中的系統事件。

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;
```

下列範例輸出顯示事件清單、事件類型、目前設定事件的層級，以及設定事件的時間。

```
942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.
```

取消設定系統診斷事件

若要取消設定系統事件，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.unset_system_event`。您只能取消設定 `rdsadmin.rdsadmin_util.list_allowed_system_events` 的輸出中列出的事件。`unset_system_event` 程序接受下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_event</code>	數字	—	是	系統事件編號。該值必須是由 <code>list_allowed_system_events</code> 報告的其中一個事件編號。

下列範例會取消設定事件 942 和 10442。包括範例輸出。

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'
```

```
PL/SQL procedure successfully completed.
```

```
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);  
Unsetting system event 10442 with: alter system set events '10442 off'
```

```
PL/SQL procedure successfully completed.
```

執行 Oracle 資料庫執行個體的一般資料庫任務

您可以在下文中找到如何在執行 Oracle 的 Amazon RDS 資料庫執行個體上執行與資料庫相關的一般 DBA 任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。Amazon RDS 也會將存取權限制在某些需要進階權限的系統程序和資料表。

主題

- [變更資料庫的全域名稱](#)
- [建立和調整資料表空間的大小](#)
- [設定預設的資料表空間](#)
- [設定預設的暫存資料表空間](#)
- [在執行個體儲存體上建立暫存資料表空間](#)
- [將暫存檔新增至僅供讀取複本上的執行個體儲存體](#)
- [捨棄僅供讀取複本上的暫存檔](#)
- [設定資料庫的檢查點](#)
- [設定分散式復原](#)
- [設定資料庫時區](#)
- [使用 Oracle 外部資料表](#)
- [使用自動工作負載儲存庫 \(AWR\) 產生效能報告。](#)
- [調整資料庫連結以搭配使用 VPC 中的資料庫執行個體](#)
- [設定資料庫執行個體的預設版本](#)
- [為 SYS.AUD\\$ 資料表啟用稽核](#)
- [為 SYS.AUD\\$ 資料表停用稽核](#)
- [清除中斷的線上索引建置](#)
- [略過損毀的區塊](#)
- [調整資料表空間、資料檔案和暫存檔的大小](#)

- [清除資源回收筒](#)
- [設定完全修訂的預設顯示值](#)

變更資料庫的全域名稱

若要變更資料庫的全域名稱，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.rename_global_name`。`rename_global_name` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_new_global_name</code>	<code>varchar2</code>	—	是	資料庫的新全域名稱。

必須開啟資料庫，才會發生名稱變更。如需變更資料庫全域名稱的詳細資訊，請參閱 Oracle 文件中的 [ALTER DATABASE](#)。

下列範例會將資料庫的全域名稱變更為 `new_global_name`。

```
EXEC rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

建立和調整資料表空間的大小

Amazon RDS 僅支援將 Oracle Managed Files (OMF) 用於資料檔案、日誌檔案和控制檔案。建立資料檔案和日誌檔案時，您不可以指定實體檔案名稱。

依預設，如果未指定資料檔案大小，則資料表空間建立時會使用預設值 `AUTOEXTEND ON`，並且沒有大小上限。在下列範例中，資料表空間 `users1` 是可自動擴展的。

```
CREATE TABLESPACE users1;
```

由於這些預設設定，資料表空間可能成長以致於耗用所有配置的儲存體。建議您對永久和暫存資料表空間指定適當的大小，並且謹慎監控空間使用情況。

下列範例會建立名為 `users2` 的資料表空間，其中的起始大小為 1 GB。由於已指定資料檔案大小，但未指定 `AUTOEXTEND ON`，因此資料表空間無法自動擴展。

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G;
```


下列範例會建立名為 *users3* 的資料表空間，其中的起始大小為 1 GB、開啟自動擴展，且大小上限為 10 GB：

```
CREATE TABLESPACE users3 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

下列範例會建立名為 *temp01* 的暫存資料表空間：

```
CREATE TEMPORARY TABLESPACE temp01;
```

ALTER TABLESPACE 可幫助您調整大型檔案資料表空間的大小，您可以用 KB (K)、MB (M)、GB (G) 或 TB (T) 來指定大小。下列範例會將名為 *users_bf* 的大型檔案資料表空間大小調整為 200 MB。

```
ALTER TABLESPACE users_bf RESIZE 200M;
```

下列範例會將額外的資料檔案新增至名為 *users_sf* 的小型檔案資料表空間。

```
ALTER TABLESPACE users_sf ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m  
MAXSIZE UNLIMITED;
```

設定預設的資料表空間

若要設定預設的資料表空間，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.alter_default_tablespace`。alter_default_tablespace 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
tablespace_name	varchar	—	是	預設資料表空間的名稱。

下列範例會將預設資料表空間設定為 *users2*：

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

設定預設的暫存資料表空間

若要設定預設的暫存資料表空間，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.alter_default_temp_tablespace`。alter_default_temp_tablespace 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
tablespace_name	varchar	—	是	預設暫存資料表空間的名稱。

下列範例會將預設的暫存資料表空間設定為 *temp01*。

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

在執行個體儲存體上建立暫存資料表空間

若要在執行個體儲存體上建立暫存資料表空間，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace`。 `create_inst_store_tmp_tblspace` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_tablespace_name	varchar	—	是	暫存資料表空間的名稱。

下列範例會在執行個體儲存體中建立暫存資料表空間 *temp01*。

```
EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace(p_tablespace_name =>
'temp01');
```

Important

執行 `rdsadmin_util.create_inst_store_tmp_tblspace` 時，新建立的暫存資料表空間不會自動設為預設暫存資料表空間。若要將其設為預設值，請參閱 [設定預設的暫存資料表空間](#)。

如需詳細資訊，請參閱 [將暫存資料存放在 RDS for Oracle 執行個體儲存體中](#)。

將暫存檔新增至僅供讀取複本上的執行個體儲存體

當您在主要資料庫執行個體上建立暫存資料表空間時，僅供讀取複本不會建立暫存檔。假設出於以下原因之一，僅供讀取複本上存在空的暫存資料表空間：

- 您已從僅供讀取複本的資料表空間捨棄暫存檔。如需詳細資訊，請參閱 [捨棄僅供讀取複本上的暫存檔](#)。
- 您已在主要資料庫執行個體上建立新的暫存資料表空間。在此情況下，RDS for Oracle 會將中繼資料同步至僅供讀取複本。

您可以將暫存檔新增至空的暫存資料表空間，並將暫存檔存放在執行個體儲存體中。若要在執行個體儲存體上建立暫存檔，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.add_inst_store_tempfile`。您只能在僅供讀取複本上使用此程序。程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_tablespace_name</code>	<code>varchar</code>	—	是	僅供讀取複本上暫存資料表空間的名稱。

在下列範例中，僅供讀取複本上存在空的暫存資料表空間 `temp01`。執行下列命令，為此資料表空間建立暫存檔，並將它存放在執行個體儲存體中。

```
EXEC rdsadmin.rdsadmin_util.add_inst_store_tempfile(p_tablespace_name => 'temp01');
```

如需詳細資訊，請參閱 [將暫存資料存放在 RDS for Oracle 執行個體儲存體中](#)。

捨棄僅供讀取複本上的暫存檔

您無法捨棄僅供讀取複本上的現有暫存資料表空間。您可以將僅供讀取複本上的暫存檔儲存體從 Amazon EBS 變更為執行個體儲存體，或從執行個體儲存體變更為 Amazon EBS。若要實現這些目標，請執行下列操作：

1. 捨棄僅供讀取複本上暫存資料表空間中的目前暫存檔。
2. 在不同的儲存體上建立新的暫存檔。

若要捨棄暫存檔，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.drop_replica_tempfiles`。您只能在僅供讀取複本上使用此程序。`drop_replica_tempfiles` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_tablespace_name	varchar	—	是	僅供讀取複本上暫存資料表空間的名稱。

假設名為 *temp01* 的暫存資料表空間位於僅供讀取複本的執行個體儲存體中。執行下列命令，捨棄此資料表空間中的所有暫存檔。

```
EXEC rdsadmin.rdsadmin_util.drop_replica_tempfiles(p_tablespace_name => 'temp01');
```

如需詳細資訊，請參閱 [將暫存資料存放在 RDS for Oracle 執行個體儲存體中](#)。

設定資料庫的檢查點

若要設定資料庫的檢查點，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.checkpoint`。 `checkpoint` 程序沒有參數。

下列範例會設定資料庫的檢查點。

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

設定分散式復原

若要設定分散式復原，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.enable_distr_recovery` 和 `disable_distr_recovery`。這些程序沒有參數。

下列範例會啟用分散式復原。

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

下列範例會停用分散式復原。

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

設定資料庫時區

您可以使用下列方式設定 Amazon RDS Oracle 資料庫的時區：

- Timezone 選項

Timezone 選項會在主機層級變更時區並且會影響所有日期欄和值，例如 SYSDATE。如需更多詳細資訊，請參閱 [Oracle 時區](#)。

- Amazon RDS 程序 `rdsadmin.rdsadmin_util.alter_db_time_zone`

`alter_db_time_zone` 程序只會變更某些資料類型的時區，並且不會變更 SYSDATE。[Oracle 文件](#)中列出了設定時區的額外限制。

Note

您也可以設定「Oracle 排程器」的預設時區。如需更多詳細資訊，請參閱 [設定 Oracle Scheduler 任務的時區](#)。

`alter_db_time_zone` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_new_tz</code>	<code>varchar2</code>	—	是	此參數會以具名區域或國際標準時間 (UTC) 的絕對偏移量來表示新時區。有效的偏移範圍從 -12:00 至 +14:00。

下列範例會將時區變更為 UTC 加 3 小時。

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

下列範例會將時區變更為非洲/阿爾及爾時區。

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

使用 `alter_db_time_zone` 程序更改時區後，您必須重新啟動資料庫執行個體，變更才會生效。如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。如需升級時區的相關資訊，請參閱 [時區考量](#)。

使用 Oracle 外部資料表

Oracle 外部資料表，指的是該資料表所具有的資料不在資料庫中。反而資料是位在資料庫可以存取的外部檔案中。您可以使用外部資料表來存取資料，而不需將資料載入資料庫。如需外部資料表的詳細資訊，請參閱 Oracle 文件中的[管理外部資料表](#)。

您可以使用 Amazon RDS 在目錄物件中儲存外部資料表檔案。您可以建立目錄物件，也可以使用 Oracle 資料庫中預先定義的目錄物件，例如 DATA_PUMP_DIR 目錄。如需建立目錄物件的詳細資訊，請參閱 [在主要資料儲存空間中建立和捨棄目錄](#)。您可以查詢 ALL_DIRECTORIES 檢視，來列出您 Amazon RDS Oracle 資料庫執行個體的目錄物件。

Note

目錄物件會指向您執行個體使用的主要資料儲存空間 (Amazon EBS 磁碟區)。根據配置的儲存體來計算的已使用空間 (以及資料檔案、重做日誌、稽核、追蹤和其他檔案)。

您可以使用 [DBMS_FILE_TRANSFER](#) 套件或 [UTL_FILE](#) 套件，在不同 Oracle 資料庫之間移動外部資料檔案。外部資料檔案會從來源資料庫上的目錄，移動至目的地資料庫上指定的目錄。如需使用 DBMS_FILE_TRANSFER 的詳細資訊，請參閱 [使用 Oracle Data Pump 匯入](#)。

移動外部資料檔案後，該檔案即可用來建立外部資料表。下列範例會建立採用 USER_DIR1 目錄中 emp_xt_file1.txt 檔案的外部資料表。

```
CREATE TABLE emp_xt (
  emp_id      NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY USER_DIR1
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
    (emp_id,first_name,last_name,user_name)
  )
  LOCATION ('emp_xt_file1.txt')
)
```

```
PARALLEL
REJECT LIMIT UNLIMITED;
```

假設您要將 Amazon RDS Oracle 資料庫執行個體中的資料移動至外部資料表檔案。在此情況下，您可以透過建立外部資料表並從資料庫中的資料表選取資料，來填入外部資料檔案。例如，下列 SQL 陳述式會透過查詢資料庫中的 `orders_xt` 資料表來建立 `orders` 外部資料表。

```
CREATE TABLE orders_xt
  ORGANIZATION EXTERNAL
  (
    TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY DATA_PUMP_DIR
    LOCATION ('orders_xt.dmp')
  )
AS SELECT * FROM orders;
```

在此範例中，資料會填入 `DATA_PUMP_DIR` 目錄中的 `orders_xt.dmp` 檔案。

使用自動工作負載儲存庫 (AWR) 產生效能報告。

若要收集效能資料並產生報告，Oracle 建議您使用自動工作負載儲存庫 (AWR)。AWR 需要 Oracle Database Enterprise Edition 以及診斷和調整套件的授權。若要啟用 AWR，請將 `CONTROL_MANAGEMENT_PACK_ACCESS` 初始化參數設定為 `DIAGNOSTIC` 或 `DIAGNOSTIC +TUNING`。

在 RDS 中使用 AWR 報告

若要產生 AWR 報告，您可以執行指令碼，例如 `awrrpt.sql`。這些指令碼會安裝在資料庫主機伺服器上。在 Amazon RDS 中，您不能直接存取主機。不過，您可以從另一個 Oracle 資料庫安裝版本取得 SQL 指令碼的複本。

您也可以執行 `SYS.DBMS_WORKLOAD_REPOSITORY` PL/SQL 套件中的程序來使用 AWR。您可以使用此套件來管理基準和快照，也可以用來顯示 ASH 和 AWR 報告。例如，若要以文字格式產生 AWR 報告，請執行 `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT` 程序。不過，您無法從 AWS Management Console 連上這些 AWR 報告。

使用 AWR 時，我們建議您使用 `rdsadmin.rdsadmin_diagnostic_util` 程序。您可以使用這些程序來產生下列項目：

- AWR 報告

- 作用中工作階段歷史記錄 (ASH) 報告
- 自動資料庫診斷監控 (ADDM) 報告
- AWR 資料的 Oracle 資料匯出傾印檔案

`rdsadmin_diagnostic_util` 程序會將報告儲存至資料庫執行個體檔案系統。您可以從主控台存取這些報告。您也可以使用 `rdsadmin.rds_file_util` 程序來存取報告，而且可以使用 S3 整合選項來存取複製到 Amazon S3 的報告。如需更多詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#) 及 [Amazon S3 整合](#)。

您可以在下列 Amazon RDS for Oracle 資料庫引擎版本中使用 `rdsadmin_diagnostic_util` 程序：

- 所有 Oracle Database 21c 版本
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 及更新的 Oracle Database 19c 版本

如需說明如何在複寫案例中使用診斷報告的部落格，請參閱 [為 Amazon RDS for Oracle 僅供讀取複本產生 AWR 報告](#)。

診斷公用程式套件的常見參數

透過 `rdsadmin_diagnostic_util` 套件管理 AWR 和 ADDM 時，通常會使用下列參數。

參數	資料類型	預設	必要	描述
<code>begin_snap_id</code>	NUMBER	—	是	開始快照的 ID。
<code>end_snap_id</code>	NUMBER	—	是	結束快照的 ID。
<code>dump_directory</code>	VARCHAR	BDUMP	否	要寫入報告或匯出檔案的目錄。如果您指定非預設目錄，則執行 <code>rdsadmin_diagnostic_util</code> 程序的使用者必須具有目錄的寫入許可。
<code>p_tag</code>	VARCHAR	—	否	可用於區分備份的字串，以指明備份的目的或用法，例如 <code>incremental</code> 或 <code>daily</code> 。

參數	資料類型	預設	必要	描述
				<p>最多可以指定 30 個字元。有效字元為 a-z、A-Z、0-9、底線 (_)、破折號 (-) 和句點 (.)。標籤不區分大小寫。無論輸入標籤時使用的大小寫為何，RMAN 一律以大寫形式儲存標籤。</p> <p>標籤不需是唯一的，因此多個備份可以有相同的標籤。如果沒有指定標籤，RMAN 會使用 TAGYYYYMMDDT HHMMSS 格式自動指派預設標籤，其中 YYYY 是年、MM 是月、DD 是日、HH 是時 (24 小時制)、MM 是分、SS 是秒。日期和時間指出 RMAN 開始備份的時間。例如，具有預設標籤 TAG20190927T214517 的備份，表示備份是在 2019-09-27 的 21:45:17 開始。</p> <p>下列 RDS for Oracle 資料庫引擎版本支援 p_tag 參數：</p> <ul style="list-style-type: none"> • Oracle Database 21c (21.0.0) • 使用 19.0.0.0.ru-2021-10.rur-2021-10.r1 及更新版本的 Oracle Database 19c (19.0.0)
report_type	VARCHAR	HTML	否	報告的格式。有效值為 TEXT 和 HTML。
dbid	NUMBER	—	否	顯示在 Oracle DBA_HIST_DATABASE_INSTANCE 檢視中的有效資料庫識別碼 (DBID)。如果未指定此參數，RDS 會使用 V\$DATABASE.DBID 檢視中顯示的目前 DBID。

透過 **rdsadmin_diagnostic_util** 套件管理 ASH 時，您通常使用下列參數。

參數	資料類型	預設	必要	描述
begin_time	DATE	—	是	ASH 分析的開始時間。

參數	資料類型	預設	必要	描述
end_time	DATE	—	是	ASH 分析的結束時間。
slot_width	NUMBER	0	否	ASH 報告的「最常發生的活動」區段中使用的時段持續時間 (秒)。如果未指定此參數，則 begin_time 與 end_time 之間的時間間隔不得超過 10 個時段。
sid	NUMBER	Null	否	工作階段 ID。
sql_id	VARCHAR2	Null	否	SQL ID。
wait_classes	VARCHAR2	Null	否	等待類別名稱。
service_hash	NUMBER	Null	否	服務名稱雜湊。
module_name	VARCHAR2	Null	否	模組名稱。
action_name	VARCHAR2	Null	否	動作名稱。
client_id	VARCHAR2	Null	否	資料庫工作階段的應用程式特定 ID。
plsql_entry	VARCHAR2	Null	否	PL/SQL 進入點。

產生 AWR 報告

若要產生 AWR 報告，請使用 `rdsadmin.rdsadmin_diagnostic_util.awr_report` 程序。

下列範例會產生快照範圍 101–106 的 AWR 報告。輸出文字檔案命名為 `awrrpt_101_106.txt`。您可以從 AWS Management Console 存取此報告。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

下列範例會產生快照範圍 63–65 的 HTML 報告。輸出的 HTML 檔案命名為 `awrrpt_63_65.html`。此程序會將報告寫入名為 `AWR_RPT_DUMP` 的非預設資料庫目錄。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

將 AWR 資料擷取至傾印檔案

若要將 AWR 資料擷取至傾印檔案，請使用 `rdsadmin.rdsadmin_diagnostic_util.awr_extract` 程序。

下列範例會擷取快照集範圍 101–106。輸出傾印檔案命名為 `awrextract_101_106.dmp`。您可以透過主控台存取此檔案。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

下列範例會擷取快照範圍 63–65。輸出傾印檔案命名為 `awrextract_63_65.dmp`。檔案存放在名為 `AWR_RPT_DUMP` 的非預設資料庫目錄中。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

產生 ADDM 報告

若要產生 ADDM 報告，請使用 `rdsadmin.rdsadmin_diagnostic_util.addm_report` 程序。

下列範例會產生快照範圍 101–106 的 ADDM 報告。輸出文字檔案命名為 `addmrpt_101_106.txt`。您可以透過主控台存取報告。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

下列範例會產生快照範圍 63–65 的 ADDM 報告。輸出文字檔案命名為 `addmrpt_63_65.txt`。檔案存放在名為 `ADDM_RPT_DUMP` 的非預設資料庫目錄中。

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

產生 ASH 報告

若要產生 ASH 報告，請使用 `rdsadmin.rdsadmin_diagnostic_util.ash_report` 程序。

下列範例會產生 ASH 報告，其中包含 14 分鐘前到目前時間的資料。輸出檔案的名稱使用格式 `ashrptbegin_timeend_time.txt`，其中 `begin_time` 和 `end_time` 使用格式 `YYYYMMDDHH24MISS`。您可以透過主控台存取該檔案。

```
BEGIN
  rdsadmin.rdsadmin_diagnostic_util.ash_report(
    begin_time    =>    SYSDATE-14/1440,
    end_time      =>    SYSDATE,
    report_type   =>    'TEXT');
END;
/
```

下列範例會產生 ASH 報告，其中包含 2019 年 11 月 18 日下午 6 時 07 分至 2019 年 11 月 18 日下午 6 時 15 分的資料。輸出 HTML 報告的名稱為 `ashrpt_20190918180700_20190918181500.html`。報告存放在名為 `AWR_RPT_DUMP` 的非預設資料庫目錄中。

```
BEGIN
  rdsadmin.rdsadmin_diagnostic_util.ash_report(
    begin_time    =>    TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
    end_time      =>    TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
    report_type   =>    'html',
    dump_directory =>    'AWR_RPT_DUMP');
END;
/
```

從主控台或 CLI 存取 AWR 報告

若要存取 AWR 報告或匯出傾印檔案，您可以使用 AWS Management Console 或 AWS CLI。如需詳細資訊，請參閱 [下載資料庫日誌檔案](#)。

調整資料庫連結以搭配使用 VPC 中的資料庫執行個體

若要使用 Oracle 資料庫連結搭配相同 Virtual Private Cloud (VPC) 或對等 VPC 內的 Amazon RDS 資料庫執行個體，兩個資料庫執行個體之間應具備有效路由。請使用您的 VPC 路由資料表和網路存取控制清單 (ACL)，來驗證資料庫執行個體之間的有效路由。

這兩個資料庫執行個體的安全群組，皆必須允許輸入和輸出到其他資料庫執行個體。傳入和傳出規則可以參考相同 VPC 或對等連接 VPC 的安全群組。如需詳細資訊，請參閱 [更新您的安全群組以參考對等連接的 VPC 安全群組](#)。

如果已使用您 VPC 中的 DHCP 選項集設定了自訂 DNS 伺服器，您的自訂 DNS 伺服器就必須可解析資料庫連結目標的名稱。如需更多詳細資訊，請參閱 [設定自訂 DNS 伺服器](#)。

如需使用資料庫連結搭配 Oracle Data Pump 的詳細資訊，請參閱 [使用 Oracle Data Pump 匯入](#)。

設定資料庫執行個體的預設版本

您可以重新定義稱為版本的私有環境中的資料庫物件。您可以使用版本式重新定義來升級應用程式的資料庫物件，只需要最短的停機時間。

您可以使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.alter_default_edition` 來設定 Amazon RDS Oracle 資料庫執行個體的預設版本。

下列範例會將 Amazon RDS Oracle 資料庫執行個體的預設版本設定為 `RELEASE_V1`。

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

下列範例會將 Amazon RDS Oracle 資料庫執行個體的預設版本設定回 Oracle 預設值。

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

如需 Oracle 版本式重新定義的詳細資訊，請參閱 Oracle 文件中的 [關於版本和版本式重新定義](#)。

為 SYS.AUD\$ 資料表啟用稽核

若要在 `SYS.AUD$` 資料庫稽核線索資料表上啟用稽核功能，請使用 `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` 程序 Amazon RDS。ALL 是唯一支援的稽核屬性，您無法選擇是否稽核個別陳述式或操作。

執行下列版本的 Oracle 資料庫執行個體皆支援啟用稽核功能：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

`audit_all_sys_aud_table` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_by_access</code>	布林值	true	否	設定 true 可稽核 BY ACCESS。設定 false 可稽核 BY SESSION。

Note

在單一租用戶 CDB 中，以下操作雖然有效，但沒有客戶可見機制能夠偵測操作的當前狀態。稽核資訊無法從 PDB 中取得。如需更多詳細資訊，請參閱 [RDS for Oracle CDB 的限制](#)。

下列查詢會傳回資料庫中 SYS.AUD\$ 目前的稽核組態。

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

下列命令會在 ALL SYS.AUD\$ 上啟用 BY ACCESS 的稽核功能。

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;  
  
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

下列命令會在 ALL SYS.AUD\$ 上啟用 BY SESSION 的稽核功能。

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

如需詳細資訊，請參閱 Oracle 文件中的 [AUDIT \(傳統稽核功能\)](#)。

為 SYS.AUD\$ 資料表停用稽核

若要在 SYS.AUD\$ 資料庫稽核線索資料表上停用稽核功能，請使用 `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` 程序 Amazon RDS。這個程序不需採用任何參數。

下列查詢會傳回資料庫中 SYS.AUD\$ 目前的稽核組態：

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

下列命令會在 ALL 上停用 SYS.AUD\$ 的稽核功能。

```
EXEC rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

如需詳細資訊，請參閱 Oracle 文件中的 [NOAUDIT \(傳統稽核功能\)](#)。

清除中斷的線上索引建置

若要清除失敗的線上索引建置，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_dbms_repair.online_index_clean`。

`online_index_clean` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>object_id</code>	binary_integer	ALL_INDEX_ID	否	索引的物件 ID。通常，您可以使用來自 ORA-08104 錯誤文字的物件 ID。
<code>wait_for_lock</code>	binary_integer	<code>rdsadmin.rdsadmin_dbms_repair.lock_wait</code>	否	指定 <code>rdsadmin.rdsadmin_dbms_repair.lock_wait</code> (預設值)，來嘗試取得對基礎物件的鎖定，並在鎖定失敗時，重試直到達到內部限制為止。 指定 <code>rdsadmin.rdsadmin_dbms_repair.lock_nowait</code> ，來嘗試取得對基礎物件的鎖定，但若鎖定失敗，不會重試。

下列範例清除失敗的線上索引建置：

```
declare
  is_clean boolean;
begin
  is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
    object_id      => 1234567890,
    wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
```

```
);  
end;  
/
```

如需詳細資訊，請參閱 Oracle 文件中的 [ONLINE_INDEX_CLEAN 函式](#)。

略過損毀的區塊

若要在索引和資料表掃描期間略過損毀的區塊，請使用 `rdsadmin.rdsadmin_dbms_repair` 套件。

下列程序包裝 `sys.dbms_repair.admin_table` 程序的功能，而且不會採用任何參數：

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

下列程序會採用與其在 Oracle 資料庫 `DBMS_REPAIR` 套件中的對等項目相同的參數：

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`
- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

如需處理資料庫損毀的詳細資訊，請參閱 Oracle 文件中的 [DBMS_REPAIR](#)。

Example 回應損毀的區塊

此範例顯示回應損毀區塊的基本工作流程。您的步驟將取決於區塊損毀的位置和性質。

Important

嘗試修復損毀的區塊前，請仔細檢閱 [DBMS_REPAIR](#) 文件。

如何在索引和資料表掃描期間略過損毀的區塊

1. 執行下列程序來建立修復資料表 (如果尚未存在的話)。

```
EXEC rdsadmin.rdsadmin_dbms_repair.create_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. 執行下列程序來檢查是否存在現有記錄，並清除它們 (若適用)。

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

EXEC rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. 執行下列程序來檢查是否存在損毀的區塊。

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
    v_num_corrupt := 0;
    rdsadmin.rdsadmin_dbms_repair.check_object (
        schema_name => '&corruptionOwner',
        object_name => '&corruptionTable',
        corrupt_count => v_num_corrupt
    );
    dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
FROM   DBA_TABLES
WHERE  OWNER = '&corruptionOwner'
```

```
AND TABLE_NAME = '&corruptionTable';
```

4. 使用 `skip_corrupt_blocks` 程序，針對受影響的資料表啟用或停用損毀略過。您可能也需要根據情況，將資料擷取到新的資料表，然後捨棄包含損毀區塊的資料表。

執行下列程序來對受影響的資料表啟用損毀略過。

```
begin
  rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
    flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
end;
/
select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
= '&corruptionTable';
```

執行下列程序來啟用損毀略過。

```
begin
  rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
    flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
end;
/

select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
= '&corruptionTable';
```

5. 當您完成所有修復作業時，請執行下列程序來捨棄修復資料表。

```
EXEC rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

調整資料表空間、資料檔案和暫存檔的大小

依預設，Oracle 資料表空間建立時會開啟自動擴展，並且沒有大小上限。由於這些預設設定，資料表空間有時可能會成長得太大。建議您對永久和暫存資料表空間指定適當的大小，並且謹慎監控空間使用情況。

調整永久資料表空間的大小

若要調整 RDS for Oracle 資料庫執行個體中永久資料表空間的大小，請使用下列任一 Amazon RDS 程序：

- `rdsadmin.rdsadmin_util.resize_datafile`
- `rdsadmin.rdsadmin_util.autoextend_datafile`

`resize_datafile` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_data_file_id</code>	數字	—	是	要調整大小的資料檔案識別符。
<code>p_size</code>	<code>varchar2</code>	—	是	資料檔案的大小。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。

`autoextend_datafile` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_data_file_id</code>	數字	—	是	要調整大小的資料檔案識別符。
<code>p_autoextend_state</code>	<code>varchar2</code>	—	是	自動擴展功能的狀態。指定 ON 以自動擴展資料檔案，以及指定 OFF 以關閉自動擴展。

參數名稱	資料類型	預設	必要	描述
p_next	varchar2	—	否	下一個資料檔案增量的大小。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。
p_maxsize	varchar2	—	否	允許自動擴展的磁碟空間上限。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。您可以指定 UNLIMITED 來移除檔案大小限制。

下列範例會將資料檔案 4 的大小調整為 500 MB。

```
EXEC rdsadmin.rdsadmin_util.resize_datafile(4,'500M');
```

下列範例會關閉資料檔案 4 的自動擴展。它也會開啟資料檔案 5 的自動擴展，增量為 128 MB 且沒有大小上限。

```
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(4,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(5,'ON','128M','UNLIMITED');
```

調整暫存資料表空間的大小

若要調整 RDS for Oracle 資料庫執行個體中者存資料表空間的大小，包括僅供讀取複本，請使用下列任一 Amazon RDS 程序：

- rdsadmin.rdsadmin_util.resize_temp_tablespace
- rdsadmin.rdsadmin_util.resize_tempfile
- rdsadmin.rdsadmin_util.autoextend_tempfile

resize_temp_tablespace 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_temp_tablespace_name	varchar2	—	是	要調整大小的暫時資料表空間的名稱。
p_size	varchar2	—	是	資料表空間的大小。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。

resize_tempfile 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_temp_file_id	數字	—	是	要調整大小的暫存檔識別符。
p_size	varchar2	—	是	暫存檔的大小。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。

autoextend_tempfile 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_temp_file_id	數字	—	是	要調整大小的暫存檔識別符。
p_autoextend_state	varchar2	—	是	自動擴展功能的狀態。指定 ON 以自動擴展暫存檔，以及指定 OFF 以關閉自動擴展。
p_next	varchar2	—	否	下一個暫存檔增量的大小。以位元組 (預設)、KB

參數名稱	資料類型	預設	必要	描述
				(K)、MB (M) 或 GB (G) 為單位指定大小。
p_maxsize	varchar2	—	否	允許自動擴展的磁碟空間上限。以位元組 (預設)、KB (K)、MB (M) 或 GB (G) 為單位指定大小。您可以指定 UNLIMITED 來移除檔案大小限制。

下列範例會將名為 TEMP 的暫存資料表空間大小調整為 4 GB。

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

下列範例會根據檔案識別符為 1 的暫存檔，將暫時資料表空間大小調整為 2 MB。

```
EXEC rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

下列範例會關閉暫存檔 1 的自動擴展。它還將暫存檔 2 的自動擴展大小上限設定為 10 GB，增量為 100 MB。

```
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(1,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(2,'ON','100M','10G');
```

如需有關 Oracle 資料庫執行個體的僅供讀取複本的詳細資訊，請參閱 [使用適用於 Amazon RDS for Oracle 的僅供讀取複本](#)。

清除資源回收筒

當您丟棄資料表時，Oracle 資料庫不會立即移除其儲存空間。資料庫會重新命名該資料表，並將其和任何相關聯的物件放在資源回收筒中。清除資源回收筒會移除這些項目，並釋放其儲存空間。

若要清除整個資源回收筒，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.purge_dba_recyclebin`。不過，此程序無法清除 SYS 和 RDSADMIN 物件的資源回收筒。如果您需要清除這些物件，請聯絡 AWS Support。

下列範例會清除整個資源回收筒。

```
EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

設定完全修訂的預設顯示值

若要更改 Amazon RDS Oracle 執行個體上完全修訂的預設顯示值，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val`。請注意，您可以使用 DBMS_REDACTPL/SQL 套裝建立修訂政策，如 Oracle 資料庫說明文件所述。`dbms_redact_upd_full_rdct_val` 程序會針對受現有政策影響的不同資料類型，指定其顯示的字元。

`dbms_redact_upd_full_rdct_val` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_number_val</code>	數字	Null	否	修改 NUMBER 資料類型資料欄的預設值。
<code>p_binfloat_val</code>	binary_float	Null	否	修改 BINARY_FLOAT 資料類型資料欄的預設值。
<code>p_bindouble_val</code>	binary_double	Null	否	修改 BINARY_DOUBLE 資料類型資料欄的預設值。
<code>p_char_val</code>	char	Null	否	修改 CHAR 資料類型資料欄的預設值。
<code>p_varchar_val</code>	varchar2	Null	否	修改 VARCHAR2 資料類型資料欄的預設值。
<code>p_nchar_val</code>	nchar	Null	否	修改 NCHAR 資料類型資料欄的預設值。

參數名稱	資料類型	預設	必要	描述
p_nvarchar_val	nvarchar2	Null	否	修改 NVARCHAR2 資料類型資料欄的預設值。
p_date_val	date	Null	否	修改 DATE 資料類型資料欄的預設值。
p_ts_val	timestamp	Null	否	修改 TIMESTAMP 資料類型資料欄的預設值。
p_tswtz_val	timestamp with time zone	Null	否	修改 TIMESTAMP WITH TIME ZONE 資料類型資料欄的預設值。
p_blob_val	blob	Null	否	修改 BLOB 資料類型資料欄的預設值。
p_clob_val	clob	Null	否	修改 CLOB 資料類型資料欄的預設值。
p_nclob_val	nclob	Null	否	修改 NCLOB 資料類型資料欄的預設值。

以下範例將 CHAR 資料類型的預設修訂值更改為 *：

```
EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(p_char_val => '*');
```

以下範例更改了 NUMBER、DATE 和 CHAR 資料類型的預設修訂值：

```
BEGIN
rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(
  p_number_val=>1,
  p_date_val=>to_date('1900-01-01','YYYY-MM-DD'),
  p_varchar_val=>'X');
END;
/
```


使用 `dbms_redact_upd_full_rdct_val` 程序變更完整修訂的預設值後，必須重新啟動資料庫執行個體，變更才會生效。如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

執行 Oracle 資料庫執行個體的一般日誌相關任務

您可以在下文中找到如何在執行 Oracle 的 Amazon RDS 資料庫執行個體上執行與記錄日誌相關的一般 DBA 任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的 Shell 存取權，而且會將存取權限制在某些需要進階權限的系統程序和資料表。

如需更多詳細資訊，請參閱 [Oracle 資料庫日誌檔案](#)。

主題

- [設定強制記錄](#)
- [設定補充記錄](#)
- [切換線上日誌檔案](#)
- [新增線上重做日誌](#)
- [捨棄線上重做日誌](#)
- [調整線上重做日誌的大小](#)
- [保留封存的重做日誌](#)
- [存取線上和封存的重做日誌](#)
- [從 Amazon S3 下載封存的重做日誌](#)

設定強制記錄

在強制記錄模式中，Oracle 會將所有變更記錄至資料庫，暫存資料表空間和暫存區段中的變更除外 (會忽略 `NOLOGGING` 子句)。如需詳細資訊，請參閱 Oracle 文件中的 [指定 FORCE LOGGING 模式](#)。

若要設定強制記錄，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.force_logging`。 `force_logging` 程序具有下列參數。

參數名稱	資料類型	預設	是	描述
<code>p_enable</code>	布林值	<code>true</code>	否	設定為 <code>true</code> 可將資料庫設定為強制記錄模式，設定為 <code>false</code> 則可為資料庫解除強制記錄模式。

下列範例會將資料庫設為強制記錄模式。

```
EXEC rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

設定補充記錄

如果您啟用補充記錄日誌，就會 LogMiner 提供支援鏈結資料列和叢集資料表的必要資訊。如需詳細資訊，請參閱 Oracle 文件中的[補充記錄](#)。

Oracle 資料庫預設不會啟用補充記錄。若要啟用和停用補充記錄，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.alter_supplemental_logging`。如需 Amazon RDS 如何為 Oracle 資料庫執行個體管理已封存重做日誌保留的詳細資訊，請參閱 [保留封存的 redo 日誌](#)。

`alter_supplemental_logging` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_action</code>	<code>varchar2</code>	—	是	'ADD' 可新增補充記錄，'DROP' 可捨棄補充記錄。
<code>p_type</code>	<code>varchar2</code>	<code>null</code>	否	補充記錄的類型。有效值為 'ALL'、'FOREIGN KEY'、'PRIMARY KEY'、'UNIQUE' 或 'PROCEDURAL'。

下列範例會啟用補充記錄。

```
begin
  rdsadmin.rdsadmin_util.alter_supplemental_logging(
    p_action => 'ADD');
end;
/
```

下列範例會為所有長度固定的最大資料欄啟用補充記錄。

```
begin
  rdsadmin.rdsadmin_util.alter_supplemental_logging(
```

```

        p_action => 'ADD',
        p_type   => 'ALL');
end;
/

```

下列範例會為主索引鍵資料欄啟用補充記錄。

```

begin
  rdsadmin.rdsadmin_util.alter_supplemental_logging(
    p_action => 'ADD',
    p_type   => 'PRIMARY KEY');
end;
/

```

切換線上日誌檔案

若要切換日誌檔案，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.switch_logfile`。`switch_logfile` 程序沒有參數。

下列範例會切換日誌檔案。

```
EXEC rdsadmin.rdsadmin_util.switch_logfile;
```

新增線上重做日誌

執行 Oracle 的 Amazon RDS 資料庫執行個體會從四個線上重做日誌開始，每個為 128 MB。若要新增額外的重做日誌，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.add_logfile`。

`add_logfile` 程序具有下列參數。

Note

參數是互斥的。

參數名稱	資料類型	預設	必要	描述
bytes	正數	null	否	日誌檔案的大小 (以位元組為單位)。

參數名稱	資料類型	預設	必要	描述
p_size	varchar2	—	是	日誌檔案的大小。您可以使用 KB (K)、MB (M) 或 GB (G) 來指定大小。

下列命令會新增 100 MB 的日誌檔案。

```
EXEC rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

捨棄線上重做日誌

若要捨棄重做日誌，請使用 Amazon RDS 程序

rdsadmin.rdsadmin_util.drop_logfile。drop_logfile 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
grp	正數	—	是	日誌的群組編號。

下列範例會捨棄群組編號 3 的日誌。

```
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

您只可以捨棄狀態為「未使用」或「非作用中」的日誌。下列範例會取得日誌的狀態。

```
SELECT GROUP#, STATUS FROM V$LOG;
```

```
GROUP#      STATUS
-----  -----
1           CURRENT
2           INACTIVE
3           INACTIVE
4           UNUSED
```

調整線上重做日誌的大小

執行 Oracle 的 Amazon RDS 資料庫執行個體會從四個線上重做日誌開始，每個為 128 MB。下列範例顯示如何使用 Amazon RDS 程序來調整您的日誌的大小，從每個 128 MB 到每個 512 MB。

```
/* Query V$LOG to see the logs. */
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
-----
1           134217728  INACTIVE
2           134217728  CURRENT
3           134217728  INACTIVE
4           134217728  INACTIVE

/* Add four new logs that are each 512 MB */

EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);

/* Query V$LOG to see the logs. */
/* Now there are 8 logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
-----
1           134217728  INACTIVE
2           134217728  CURRENT
3           134217728  INACTIVE
4           134217728  INACTIVE
5           536870912  UNUSED
6           536870912  UNUSED
7           536870912  UNUSED
8           536870912  UNUSED

/* Drop each inactive log using the group number. */

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 4);
```

```
/* Query V$LOG to see the logs. */
/* Now there are 5 logs.          */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#      BYTES      STATUS
-----  -
2           134217728  CURRENT
5           536870912  UNUSED
6           536870912  UNUSED
7           536870912  UNUSED
8           536870912  UNUSED

/* Switch logs so that group 2 is no longer current. */

EXEC rdsadmin.rdsadmin_util.switch_logfile;

/* Query V$LOG to see the logs.          */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
-----  -
2           134217728  ACTIVE
5           536870912  CURRENT
6           536870912  UNUSED
7           536870912  UNUSED
8           536870912  UNUSED

/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to
"INACTIVE". */

EXEC rdsadmin.rdsadmin_util.checkpoint;

/* Query V$LOG to see the logs.          */
/* Now the final original log is inactive. */
```

```

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#      BYTES      STATUS
-----
2           134217728  INACTIVE
5           536870912  CURRENT
6           536870912  UNUSED
7           536870912  UNUSED
8           536870912  UNUSED

# Drop the final inactive log.

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 2);

/* Query V$LOG to see the logs.    */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
-----
5           536870912  CURRENT
6           536870912  UNUSED
7           536870912  UNUSED
8           536870912  UNUSED

```

保留封存的重做日誌

您可以在資料庫執行個體本機上保留已存檔的重做日誌，以便與 Oracle LogMiner (DBMS_LOGMNR) 等產品搭配使用。保留重做日誌之後，就可以用 LogMiner 來分析日誌。如需詳細資訊，請參閱 Oracle 說明文件中的 [使用 LogMiner 來分析重做日誌檔](#)。


若要保留封存的重做日誌，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.set_configuration`。 `set_configuration` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>name</code>	<code>varchar</code>	—	是	待更新組態的名稱。
<code>value</code>	<code>varchar</code>	—	是	組態值。

下列範例會保留 24 小時內的重做日誌。

```
begin
  rdsadmin.rdsadmin_util.set_configuration(
    name => 'archivelog retention hours',
    value => '24');
end;
/
commit;
```

 Note

需要遞交命令，變更才會生效。

若要檢視系統能為資料庫執行個體保留多久的封存重做日誌，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.show_configuration`。

下列範例會顯示日誌的保留時間。

```
set serveroutput on
EXEC rdsadmin.rdsadmin_util.show_configuration;
```

輸出會顯示 `archivelog retention hours` 目前的設定，下列輸出顯示封存的重做日誌會保留 48 小時。

```
NAME:archivelog retention hours
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo
log files are automatically deleted.
```

因為封存的重做日誌會保留在資料庫執行個體上，請確保資料庫執行個體有足夠的配置儲存空間能存放這些保留的日誌。若要判斷資料庫執行個體在過去 X 小時中使用了多少空間，則可執行下列查詢，並以小時數取代 X。

```
SELECT SUM(BLOCKS * BLOCK_SIZE) bytes
  FROM V$ARCHIVED_LOG
 WHERE FIRST_TIME >= SYSDATE-(X/24) AND DEST_ID=1;
```


RDS for Oracle 只有在資料庫執行個體的備份保留期間大於零時，才會產生封存的 redo 日誌。依預設，備份保留期間會大於零。

當封存日誌保留期間到期時，RDS for Oracle 會從資料庫執行個體移除封存的 redo 日誌。為了支援將資料庫執行個體還原到某個時間點，Amazon RDS 會根據備份保留期間在資料庫執行個體之外保留封存的 redo 日誌。若要修改備份保留期間，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Note

在某些情況下，您可能會在 Linux 上使用 JDBC 來下載封存的 redo 日誌，而遇到長時間的延遲和連線重設等問題。在這種情況下，上述問題可能是因 Java 用戶端上的預設隨機編號產生器設定所造成，所以建議您使用非封鎖的隨機編號產生器來設定 JDBC 驅動程式。

存取線上和封存的 redo 日誌

您可能想要存取線上和存檔的 redo 日誌檔，以便使用外部工具，例如 Attunity GoldenGate、資訊和其他工具進行採礦。若要存取這些檔案，請執行下列動作：

1. 建立目錄物件，以便提供實體檔案路徑的唯讀存取權。

使用 `rdsadmin.rdsadmin_master_util.create_archivelog_dir` 和 `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`

2. 讀取使用 PL/SQL 的檔案。

您可以透過使用 PL/SQL 讀取檔案。如需從目錄物件讀取檔案的詳細資訊，請參閱 [列出資料庫執行個體目錄中的檔案](#) 和 [讀取資料庫執行個體目錄中的檔案](#)。

下列版本支援存取交易日誌：

- Oracle Database 21c
- Oracle 資料庫 19c

下列程式碼所建立的目錄，可為您提供線上和已封存 redo 日誌檔案的唯讀存取：

Important

此程式碼也會撤銷 `DROP ANY DIRECTORY` 權限。

```
EXEC rdsadmin.rdsadmin_master_util.create_archivelog_dir;  
EXEC rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

下列程式碼會捨棄線上和封存重做日誌檔案的目錄。

```
EXEC rdsadmin.rdsadmin_master_util.drop_archivelog_dir;  
EXEC rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

下列程式碼會授予和撤銷 DROP ANY DIRECTORY 權限。

```
EXEC rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;  
EXEC rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

從 Amazon S3 下載封存的重新日誌

您可以使用 `rdsadmin.rdsadmin_archive_log_download` 套件下載資料庫執行個體上的封存重做日誌。如果封存的重新日誌不在資料庫執行個體上，您可能想要從 Amazon S3 再次下載這些日誌。然後，您可以挖掘日誌或使用它們來恢復或複寫資料庫。

Note

您無法在僅供讀取複本執行個體上下載封存重做日誌。

下載封存的重新日誌：基本步驟

封存重做日誌的可用性取決於下列保留政策：

- 備份保留政策 — Amazon S3 提供此政策之內的日誌。移除此政策之外的日誌。
- 封存日誌保留政策 — 您的資料庫執行個體提供此政策之內的日誌。移除此政策之外的日誌。

如果日誌不在您的執行個體上，但受到備份保留期間的保護，請使用 `rdsadmin.rdsadmin_archive_log_download` 再次下載。RDS for Oracle 會將日誌儲存至資料庫執行個體上的 `/rdsdbdata/log/arch` 目錄。

從 Amazon S3 下載封存的重新日誌

1. 設定您的保留期間，以確保您下載的封存重做日誌在您需要保留期間之內。務必 COMMIT 您的變更。

RDS 會根據封存的日誌保留政策保留您下載的日誌檔，從日誌下載的時間開始。若要了解如何設定保留政策，請參閱 [保留封存的重新日誌](#)。

2. 最多等待 5 分鐘，讓封存日誌保留政策變更生效。
3. 使用 `rdsadmin.rdsadmin_archive_log_download` 從 Amazon S3 下載封存的重新日誌。

如需更多詳細資訊，請參閱 [下載單一封存重新日誌](#) 及 [下載一系列的封存重新日誌](#)。

Note

RDS 會在下載前自動檢查可用的儲存空間。如果要求的日誌會耗用高百分比的空間，您會收到提醒。

4. 確認日誌已成功從 Amazon S3 下載。

您可在 `bdump` 檔案之中檢視下載任務的狀態。`bdump` 檔案的路徑名稱是 `/rdsdbdata/log/trace/dbtask-task-id.log`。在上述的下載步驟中，您執行的 `SELECT` 陳述式會傳回 `VARCHAR2` 資料類型的任務 ID。如需更多詳細資訊，請參閱 [監控檔案傳輸狀態](#) 中的類似範例。

下載單一封存重新日誌

若要將單一封存重新日誌下載至 `/rdsdbdata/log/arch` 目錄，請使用 `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`。此程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>seqnum</code>	數字	—	是	封存重新日誌的序號。

下列範例會下載序號為 20 的日誌。

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum(seqnum => 20)
       AS TASK_ID
FROM   DUAL;
```

下載一系列的封存重做日誌

若要將一系列的封存重做日誌下載至 `/rdsdbdata/log/arch` 目錄，請使用 `download_logs_in_seqnum_range`。您的下載限制為每個請求 300 個日誌。`download_logs_in_seqnum_range` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>start_seq</code>	數字	—	是	系列的起始序號。
<code>end_seq</code>	數字	—	是	系列的結束序號。

下列範例會下載序列 50 到 100 的日誌。

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range(start_seq
=> 50, end_seq => 100)
      AS TASK_ID
FROM   DUAL;
```

執行 Oracle 資料庫執行個體的一般 RMAN 任務

下節會說明在執行 Oracle 的 Amazon RDS 資料庫執行個體上進行 Oracle Recovery Manager (RMAN) DBA 任務的方法。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的殼層存取權。而且會將存取權限制在某些需要進階權限的系統程序和資料表。

使用 Amazon RDS 套件 `rdsadmin.rdsadmin_rman_util`，將 Amazon RDS for Oracle 資料庫執行 RMAN 備份至磁碟。`rdsadmin.rdsadmin_rman_util` 套件可支援完整和增量資料庫檔案備份、資料表空間備份，以及封存的重做日誌檔備份。

RMAN 備份作業完成後，您即可從 Amazon RDS for Oracle 資料庫執行個體主機複製備份檔案。如此一來，您就能透過該檔案還原至非 RDS 主機，或用來長期儲存備份資料。例如，您可以將備份檔案複製到 Amazon S3 儲存貯體。如需詳細資訊，請參閱 [Amazon S3 整合](#) 的使用方式。

RMAN 備份作業產生的備份檔案會保留在 Amazon RDS 資料庫執行個體主機上，直到您手動移除這些檔案為止。您可使用 `UTL_FILE.FREMOVE` Oracle 程序由目錄移除檔案。如需詳細資訊，請參閱 Oracle 資料庫文件中的 [FREMOVE 程序](#)。

您無法使用 RMAN 來還原 RDS for Oracle 資料庫執行個體。不過，您可以使用 RMAN 將備份還原到內部部署或 Amazon EC2 執行個體。如需詳細資訊，請參閱部落格文章 [將 Amazon RDS for Oracle 執行個體還原到自我管理的執行個體](#)。

Note

若要備份或還原至另一個 Amazon RDS for Oracle 資料庫執行個體，則可繼續使用 Amazon RDS 備份和還原功能。如需詳細資訊，請參閱 [備份、還原和匯出資料](#)。

主題

- [RMAN 備份的先決條件](#)
- [RMAN 程序的常見參數](#)
- [在 RDS 中為甲骨文驗證數據庫文件](#)
- [啟用和停用區塊變更追蹤功能](#)
- [交叉檢查封存的 redo 日誌](#)
- [備份存檔 redo 日誌檔](#)
- [執行完整資料庫備份](#)
- [執行租戶資料庫的完整備份](#)
- [執行增量資料庫備份](#)
- [執行租戶資料庫的增量備份](#)
- [備份資料表空間](#)
- [備份控制檔案](#)
- [執行區塊媒體復原](#)

RMAN 備份的先決條件

使用 `rdsadmin.rdsadmin_rman_util` 套件備份資料庫之前，請確定您符合下列先決條件：

- 確保你的 RDS for Oracle 資料庫處於 ARCHIVELOG 模式。若要啟用此模式，請將備份保留期間設定為非零值。
- 在備份封存的 redo 日誌，或是執行包含封存 redo 日誌的完整或增量備份，以及備份資料庫時，請確保將 redo 日誌保留期間設定為非零值。需要存檔的 redo 日誌，才能在復原期間使資料庫檔案保持一致。如需詳細資訊，請參閱 [保留封存的 redo 日誌](#)。

- 請確保您的資料庫執行個體有足夠的可用空間來存放備份。當您備份資料庫時，可以在程序呼叫中將 Oracle 目錄物件指定為參數。RMAN 會將檔案放在指定目錄中。您能夠使用預設目錄 (如 DATA_PUMP_DIR)，也能夠建立新目錄。如需詳細資訊，請參閱 [在主要資料儲存空間中建立和捨棄目錄](#)。

您可以使用 CloudWatch 測量結果來監督 RDS for Oracle 執行處理中目前的可用空間 FreeStorageSpace。我們建議您的可用空間超過資料庫目前的大小，儘管 RMAN 僅備份格式化的區塊並支援壓縮。

RMAN 程序的常見參數

您可以使用 Amazon RDS 套件 `rdsadmin.rdsadmin_rman_util` 中的程序來執行 RMAN 任務。有多個參數對該套件中的程序而言都是共通的，以下即為該套件的常見參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>p_directory_name</code>	varchar	有效的資料庫目錄名稱。	—	是	要包含備份檔案的目錄名稱。
<code>p_label</code>	varchar	a-z, A-Z, 0-9, '_', '-', '.'	—	否	包含在備份檔案名稱中的唯一字串。 Note 長度限制為 30 個字元。
<code>p_owner</code>	varchar	<code>p_directory_name</code> 中指定的有效目錄擁有者。	—	是	要包含備份檔案的目錄擁有者。
<code>p_tag</code>	varchar	a-z, A-Z, 0-9, '_', '-', '.'	NULL	否	可用於區分備份的字串，以指明備份的目的或用法，例如每日、每週或增量層級備份。

參數名稱	資料類型	有效值	預設	必要	描述
					<p>長度限制為 30 個字元。標籤不區分大小寫。無論輸入標籤時使用的大小寫為何，一律以大寫形式儲存標籤。</p> <p>標籤不需是唯一的，因此多個備份可以有相同的標籤。</p> <p>如果沒有指定標籤，則 RMAN 會使用 <code>TAGYYYYMMDDTHHMMSS</code> 格式自動指派預設標籤，其中 <code>YYYY</code> 是年、<code>MM</code> 是月、<code>DD</code> 是日、<code>HH</code> 是時 (24 小時制)、<code>MM</code> 是分、<code>SS</code> 是秒。日期和時間指的是 RMAN 開始備份的時間。</p> <p>例如，如果備份是在 2019-09-27 的 21:45:17 開始的，接收到的標籤可能是 <code>TAG20190927T214517</code>。</p> <p>下列 Amazon RDS for Oracle 資料庫引擎版本支援 <code>p_tag</code> 參數：</p> <ul style="list-style-type: none"> • Oracle Database 21c (21.0.0) • 使用 19.0.0.0.ru-2021-10.rur-2021-10.r1 或更新版本的 Oracle Database 19c (19.0.0)
<code>p_compress</code>	boolean	TRUE, FALSE	FALSE	否	<p>指定 TRUE 可啟用 BASIC 備份壓縮。</p> <p>指定 FALSE 可停用 BASIC 備份壓縮。</p>

參數名稱	資料類型	有效值	預設	必要	描述
p_include_archive_logs	布林值	TRUE, FALSE	FALSE	否	<p>指定 TRUE 可在備份中包含封存的重做日誌。</p> <p>指定 FALSE 可排除備份中的封存重做日誌。</p> <p>如果您選擇在備份中包含封存的重做日誌，請使用 <code>rdsadmin.rdsadmin_util.set_configuration</code> 程序將保留期間設定為一小時以上。同時，您還需立即呼叫 <code>rdsadmin.rdsadmin_rman_util.crosscheck_archive_log</code> 程序，再開始執行備份。否則，備份作業可能會因缺少遭 Amazon RDS 管理程序刪除的封存重做日誌檔而失敗。</p>
p_include_controlfile	布林值	TRUE, FALSE	FALSE	否	<p>指定 TRUE 可在備份中包含控制檔案。</p> <p>指定 FALSE 可排除備份中的控制檔案。</p>
p_optimize	布林值	TRUE, FALSE	TRUE	否	<p>指定 TRUE 可啟用備份最佳化。在包含封存重做日誌的情況下，此參數可縮減備份大小。</p> <p>指定 FALSE 可停用備份最佳化。</p>

參數名稱	資料類型	有效值	預設	必要	描述
p_parallel	數字	Oracle Database Enterprise Edition (EE)，介於 1 與 254 之間的有效整數 1其他 Oracle 資料庫版本為	1	否	通道的數目。
p_rman_to_dbms_output	布林值	TRUE, FALSE	FALSE	否	<p>為 TRUE 時，RMAN 輸出除了傳送到 DBMS_OUTPUT 目錄中的檔案以外，還會傳送到 BDUMP 套件。在 SQL*Plus 中，使用 SET SERVEROUTPUT ON 來查看輸出。</p> <p>設定為 FALSE 時，系統只會將 RMAN 輸出傳送至 BDUMP 目錄中的檔案。</p>
p_section_size_mb	數字	有效的整數	NULL	否	<p>區段大小，以 MB 為單位。</p> <p>將每個檔案分割為指定的區段大小來平行驗證。</p> <p>為 NULL 時，會忽略參數。</p>

參數名稱	資料類型	有效值	預設	必要	描述
p_validation_type	varchar	'PHYSICAL', 'PHYSICAL+LOGICAL'	'PHYS'	否	<p>損毀偵測的層級。</p> <p>指定 'PHYSICAL' 來檢查實體損毀。實體損毀的其中一例是標頭和頁尾中不相符的區塊。</p> <p>指定 'PHYSICAL+LOGICAL' 來檢查邏輯不一致以及實體損毀。邏輯損毀的其中一例，就是損毀的區塊。</p>

在 RDS 中為甲骨文驗證數據庫文件

您可以使用 Amazon RDS 套件 `rdsadmin.rdsadmin_rman_util` 來驗證適用於 Oracle 資料庫檔案的 Amazon RDS，例如資料檔案、表格空間、控制檔和伺服器參數檔案 (SPFiles)。

如需 RMAN 驗證的詳細資訊，請參閱 Oracle 文件中的 [驗證資料庫檔案和備份](#)，以及 [VALIDATE](#)。

主題

- [驗證資料庫](#)
- [驗證租戶資料庫](#)
- [驗證資料表空間](#)
- [驗證控制檔案](#)
- [驗證 SPFILE](#)
- [驗證甲骨文數據文件](#)

驗證資料庫

若要驗證 Oracle 適用於 Oracle 的 RDS 資料庫所使用的所有相關檔案，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_database`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_validation_type
- p_parallel

- `p_section_size_mb`
- `p_rman_to_dbms_output`

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列範例會使用參數的預設值來驗證資料庫。

```
EXEC rdsadmin.rdsadmin_rman_util.validate_database;
```

下列範例會使用指定的參數值來驗證資料庫。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.validate_database(
    p_validation_type      => 'PHYSICAL+LOGICAL',
    p_parallel             => 4,
    p_section_size_mb     => 10,
    p_rman_to_dbms_output => FALSE);
END;
/
```

`p_rman_to_dbms_output` 參數設定為 `FALSE` 時，系統會將 RMAN 輸出寫入 BDUMP 目錄中的檔案。

若要檢視 BDUMP 目錄中的檔案，請執行下列 SELECT 陳述式。

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

若要檢視 BDUMP 目錄中的檔案內容，請執行下列 SELECT 陳述式。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

請使用要檢視的檔案名稱來取代原先的檔案名稱。

驗證租戶資料庫

若要驗證容器資料庫 (CDB) 中租戶資料庫的資料檔案，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_tenant`。

這個程序僅適用於目前的租戶資料庫，並會將下列常見參數用於 RMAN 任務：

- p_validation_type
- p_parallel
- p_section_size_mb
- p_rman_to_dbms_output

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。下列資料庫引擎版本支援這個程序：

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

下列範例會使用參數的預設值來驗證目前的資料庫執行個體。

```
EXEC rdsadmin.rdsadmin_rman_util.validate_tenant;
```

下列範例會使用參數的指定值來驗證目前的資料庫執行個體。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.validate_tenant(
    p_validation_type    => 'PHYSICAL+LOGICAL',
    p_parallel           => 4,
    p_section_size_mb   => 10,
    p_rman_to_dbms_output => FALSE);
END;
/
```

p_rman_to_dbms_output 參數設定為 FALSE 時，系統會將 RMAN 輸出寫入 BDUMP 目錄中的檔案。

若要檢視 BDUMP 目錄中的檔案，請執行下列 SELECT 陳述式。

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

若要檢視 BDUMP 目錄中的檔案內容，請執行下列 SELECT 陳述式。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

請使用要檢視的檔案名稱來取代原先的檔案名稱。

驗證資料表空間

若要驗證與資料表空間相關聯的檔案，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_tablespace`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>p_tablespace_name</code>	<code>varchar2</code>	有效的資料表空間名稱	—	是	資料表空間的名稱。

驗證控制檔案

若要只驗證 Amazon RDS Oracle 資料庫執行個體使用的控制檔案，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_current_controlfile`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_validation_type`
- `p_rman_to_dbms_output`

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

驗證 SPFILE

若要只驗證 Amazon RDS Oracle 資料庫執行個體使用的伺服器參數檔案 (SPFILE)，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_spfile`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_validation_type
- p_rman_to_dbms_output

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

驗證甲骨文數據文件

若要驗證資料檔案，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.validate_datafile`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_validation_type
- p_parallel
- p_section_size_mb
- p_rman_to_dbms_output

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_datafile	varchar2	有效資料檔案 ID 編號或包含完整路徑的有效資料檔案名稱	—	是	資料檔案 ID 編號 (來自 v\$datafile.file#) 或包含路徑的完整資料檔案名稱 (來自 v\$datafile.name)。
p_from_block	數字	有效的整數	NULL	否	資料檔案內開始驗證的區塊數量。當此參數為 NULL 時，表示使用的數量為 1。

參數名稱	資料類型	有效值	預設	必要	描述
p_to_block	數字	有效的整數	NULL	否	資料檔案內結束驗證的區塊數量。當此參數為 NULL 時，表示資料檔案內使用的區塊數量達到上限。

啟用和停用區塊變更追蹤功能

區塊變更追蹤會紀錄追蹤檔案中變更的區塊。這項技術可以改善 RMAN 增量備份的效能。如需詳細資訊，請參閱 Oracle 資料庫文件中的 [使用區塊變更追蹤來改善增量備份效能](#)。

讀取複本不支援 RMAN 功能。不過，在您的高可用性策略中，您可能會選擇使用程序 `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking` 在唯讀複本中啟用區塊追蹤。如果您將此唯讀複本提升至來源資料庫執行個體，則會為新來源執行個體啟用區塊變更追蹤。因此，您的執行個體可能因快速增量備份而受益。

下列資料庫引擎版本支援企業版限定的區塊變更追蹤程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

Note

在單一租用戶 CDB 中，以下操作雖然有效，但沒有客戶可見機制能夠偵測操作的當前狀態。另請參閱 [RDS for Oracle CDB 的限制](#)。

您可以使用 Amazon RDS 程序

`rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`，為資料庫執行個體啟用區塊變更追蹤功能。若要停用區塊變更追蹤，請使用 `disable_block_change_tracking`。這些程序不需採用任何參數。

若要判斷資料庫執行個體是否啟用區塊變更追蹤功能，則可執行下列查詢。

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

下方範例會為資料庫執行個體啟用區塊變更追蹤功能。

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

下方範例會為資料庫執行個體停用區塊變更追蹤功能。

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

交叉檢查封存的重做日誌

您可以使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog` 來交叉檢查封存的重做日誌。

您能夠利用這個程序來交叉檢查控制檔案中註冊的封存重做日誌，並選擇刪除過期的記錄。當 RMAN 製作備份時，它會在控制檔中建立記錄。經過一段時間後，這些記錄會增加控制檔案的大小，建議您定期移除過期的記錄。

Note

標準的 Amazon RDS 備份作業不會使用 RMAN，所以系統不會在控制檔案中建立記錄。

這個程序會採用常見參數 `p_rman_to_dbms_output` 來處理 RMAN 任務。

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>p_delete_expired</code>	布林值	TRUE, FALSE	TRUE	否	當此參數為 TRUE 時，系統會從控制檔案中刪除過期的封存重做日誌。 當此參數為 FALSE 時，系統會保留控制檔案中過期的封存重做日誌。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

下列範例會將控制檔中的封存重做日誌記錄標示為過期，但不會刪除記錄。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
    p_delete_expired      => FALSE,
    p_rman_to_dbms_output => FALSE);
END;
/
```

下方範例會從控制檔案中刪除過期的封存重做日誌。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
    p_delete_expired      => TRUE,
    p_rman_to_dbms_output => FALSE);
END;
/
```

備份存檔重做日誌檔

您可以使用 Amazon RDS 套件 `rdsadmin.rdsadmin_rman_util` 來備份 Amazon RDS Oracle 資料庫執行個體的封存重做日誌。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援備份封存重做日誌的程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

主題

- [備份所有封存的重新日誌](#)
- [備份特定日期範圍內的封存重做日誌](#)
- [備份特定 SCN 範圍內的封存重做日誌](#)
- [備份特定序號範圍內的封存重做日誌](#)

備份所有封存的 redo 日誌

若要備份 Amazon RDS Oracle 資料庫執行個體的所有封存 redo 日誌，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_archivelog_all`。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下方範例會備份資料庫執行個體的所有封存 redo 日誌。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_parallel       => 4,
    p_tag            => 'MY_LOG_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

備份特定日期範圍內的封存 redo 日誌

若要透過指定某個日期範圍來備份 Amazon RDS Oracle 資料庫執行個體的特定封存 redo 日誌，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_archivelog_date`。該日期範圍可用來指定要進行備份的封存 redo 日誌。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_owner`

- p_directory_name
- p_label
- p_parallel
- p_compress
- p_rman_to_dbms_output
- p_tag

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_from_date	日期	介於磁碟上封存重做日誌 start_date 和 next_date 間的日期。該值必須小於或等於您為 p_to_date 指定的值。	—	是	封存日誌備份的開始日期。
p_to_date	日期	介於磁碟上封存重做日誌 start_date 和 next_date 間的日期。該	—	是	封存日誌備份的結束日期。

參數名稱	資料類型	有效值	預設	必要	描述
		值必須大於或等於您為 p_from_date 指定的值。			

下方範例會備份日期範圍內的資料庫執行個體封存重做日誌。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_from_date      => '03/01/2019 00:00:00',
    p_to_date        => '03/02/2019 00:00:00',
    p_parallel       => 4,
    p_tag            => 'MY_LOG_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

備份特定 SCN 範圍內的封存重做日誌

若要透過指定某個系統變更編號 (SCN) 來備份 Amazon RDS Oracle 資料庫執行個體的特定封存重做日誌，請使用 Amazon RDS 程序

rdsadmin.rdsadmin_rman_util.backup_archivelog_scn。該 SCN 範圍可用來指定要進行備份的封存重做日誌。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_compress

- p_rman_to_dbms_output
- p_tag

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_from_scn	數字	磁碟上封存重做日誌的 SCN。該值必須小於或等於您為 p_to_scn 指定的值。	—	是	封存日誌備份的起始 SCN。
p_to_scn	數字	磁碟上封存重做日誌的 SCN。該值必須大於或等於您為 p_from_scn 指定的值。	—	是	封存日誌備份的結束 SCN。

下方範例會備份 SCN 範圍內的資料庫執行個體封存重做日誌。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
    p_owner => 'SYS',
```

```

    p_directory_name      => 'MYDIRECTORY',
    p_from_scn            => 1533835,
    p_to_scn             => 1892447,
    p_parallel           => 4,
    p_tag                => 'MY_LOG_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/

```

備份特定序號範圍內的封存重做日誌

若要透過指定某個序號範圍來備份 Amazon RDS Oracle 資料庫執行個體的特定封存重做日誌，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_archive_log_sequence`。該序號範圍可用來指定要進行備份的封存重做日誌。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>p_from_sequence</code>	數字	磁碟上封存重做日誌的序號。該值必須小於或等	—	是	封存日誌備份的起始序號。

參數名稱	資料類型	有效值	預設	必要	描述
		於您為 p_to_sequence 指定的值。			
p_to_sequence	數字	磁碟上封存重做日誌的序號。該值必須大於或等於您為 p_from_sequence 指定的值。	—	是	封存日誌備份的結束序號。

下方範例會備份序號範圍內的資料庫執行個體封存重做日誌。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_from_sequence  => 11160,
    p_to_sequence    => 11160,
    p_parallel       => 4,
    p_tag            => 'MY_LOG_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

執行完整資料庫備份

您可以使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_database_full`，對備份中包含的所有資料檔案區塊執行備份作業。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

下列範例會使用參數的指定值來執行資料庫執行個體的完整備份。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_database_full(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_parallel       => 4,
    p_section_size_mb => 10,
    p_tag            => 'FULL_DB_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

執行租戶資料庫的完整備份

您可以執行所有資料區塊的備份，包括容器資料庫 (CDB) 中的租戶資料庫。使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_tenant_full` 這個程序僅適用於目前的資料庫備份，並會將下列常見參數用於 RMAN 任務：

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列 RDS for Oracle 資料庫引擎版本支援 rdsadmin_rman_util.backup_tenant_full 程序：

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

下列範例會使用參數的指定值來執行目前租戶資料庫的完整備份。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_tenant_full(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_parallel       => 4,
    p_section_size_mb => 10,
    p_tag            => 'FULL_TENANT_DB_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

執行增量資料庫備份

您可以使用 Amazon RDS 程序

rdsadmin.rdsadmin_rman_util.backup_database_incremental 來執行資料庫執行個體的增量備份作業。

如需增量備份的詳細資訊，請參閱 Oracle 文件中的[增量備份](#)。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_level	數字	0, 1	0	否	指定 0 可啟用完整增量備份。 指定 1 可啟用非累積增量備份。

下列範例會使用參數的指定值來執行資料庫執行個體的增量備份。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_database_incremental(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_level          => 1,
    p_parallel       => 4,
    p_section_size_mb => 10,
    p_tag            => 'MY_INCREMENTAL_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

執行租戶資料庫的增量備份

您可以在 CDB 中執行目前租戶資料庫的增量備份。使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_tenant_incremental`

如需增量備份的詳細資訊，請參閱 Oracle Database 文件中的 [增量備份](#)。

這個程序僅適用於目前的租戶資料庫，並會將下列常見參數用於 RMAN 任務：

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_level	數字	0, 1	0	否	指定 0 可啟用完整增量備份。 指定 1 可啟用非累積增量備份。

下列範例會使用參數的指定值來執行目前租戶資料庫的增量備份。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_tenant_incremental(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_level          => 1,
    p_parallel       => 4,
    p_section_size_mb => 10,
    p_tag            => 'MY_INCREMENTAL_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

備份資料表空間

您可以使用 Amazon RDS 程序 `rdsadmin.rdsadmin_rman_util.backup_tablespace` 來備份資料表空間。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_owner
- p_directory_name
- p_label
- p_parallel

- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此外，這個程序還會額外使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_tablespace_name	varchar2	有效的資料表空間名稱。	—	是	要備份的資料表空間名稱。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

下列範例會使用參數的指定值來執行資料表空間備份。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_tablespace(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_tablespace_name => 'MYTABLESPACE',
    p_parallel       => 4,
    p_section_size_mb => 10,
    p_tag            => 'MYTABLESPACE_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

備份控制檔案

您可以使用 Amazon RDS 程序

`rdsadmin.rdsadmin_rman_util.backup_current_controlfile` 來備份控制檔案。

這個程序會採用下列常見參數來處理 RMAN 任務：

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

如需更多詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

下列範例使用參數的指定值來備份控制檔案。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.backup_current_controlfile(
    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_tag            => 'CONTROL_FILE_BACKUP',
    p_rman_to_dbms_output => FALSE);
END;
/
```

執行區塊媒體復原

您可以使用 Amazon RDS 程序復原個別資料區塊 (稱為區塊媒體復原)

`rdsadmin.rdsadmin_rman_util.recover_datafile_block`。您可以使用此過載程序來復原個別資料區塊或一系列資料區塊。

這個程序會採用下列常見參數來處理 RMAN 任務：

- p_rman_to_dbms_output

如需詳細資訊，請參閱 [RMAN 程序的常見參數](#)。

此程序會使用下列其他參數。

參數名稱	資料類型	有效值	預設	必要	描述
p_datafile	NUMBER	有效的資料檔 ID 號碼。	—	是	<p>包含損毀區塊的資料檔案。使用下列其中一種方式指定資料檔案：</p> <ul style="list-style-type: none"> • 資料檔 ID 號碼，位於 V\$DATAFILE.FILE# • 完整的資料檔案名稱，包括路徑，位於 V\$DATAFILE.NAME
p_block	NUMBER	有效的整數。	—	是	<p>要恢復的單個塊的數量。</p> <p>下列參數是互斥的：</p> <ul style="list-style-type: none"> • p_block • p_from_block 和 p_to_block
p_from_block	NUMBER	有效的整數。	—	是	<p>要恢復的區塊範圍內的第一個區塊編號。</p> <p>下列參數是互斥的：</p> <ul style="list-style-type: none"> • p_block • p_from_block 和 p_to_block
p_to_block	NUMBER	有效的整數。	—	是	<p>要恢復的區塊範圍內的最後一個區塊編號。</p> <p>下列參數是互斥的：</p>

參數名稱	資料類型	有效值	預設	必要	描述
					<ul style="list-style-type: none"> p_block p_from_block 和 p_to_block

下列 Amazon RDS for Oracle 資料庫引擎版本皆支援這個程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

下列範例會復原資料檔 5 中的區塊 100。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.recover_datafile_block(
    p_datafile      => 5,
    p_block         => 100,
    p_rman_to_dbms_output => TRUE);
END;
/
```

下列範例會復原資料檔 5 中的區塊 100 到 150。

```
BEGIN
  rdsadmin.rdsadmin_rman_util.recover_datafile_block(
    p_datafile      => 5,
    p_from_block    => 100,
    p_to_block      => 150,
    p_rman_to_dbms_output => TRUE);
END;
/
```

執行 Oracle 資料庫執行個體的一般排程任務

SYS 擁有的部分排程器任務可能會干擾正常的資料庫作業。Oracle Support 建議您停用這些任務或修改排程。如要執行 SYS 所擁有之 Oracle Scheduler 工作的任務，請使用 Amazon RDS 套件 `rdsadmin.rdsadmin_dbms_scheduler`。

下列 Amazon RDS for Oracle 資料庫引擎版本支援 `rdsadmin.rdsadmin_dbms_scheduler` 程序：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c

Oracle Scheduler 程序的常見參數

若要使用 Oracle Scheduler 執行任務，請使用 Amazon RDS 套件 `rdsadmin.rdsadmin_dbms_scheduler` 中的程序。有多個參數對該套件中的程序而言都是共通的，以下即為該套件的常見參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>name</code>	<code>varchar2</code>	'SYS.BSLN_MAINTAIN_STATS_JOB', 'SYS.NUP_ONLINE_IND_BUILD'	—	是	要修改的工作名稱。 <div data-bbox="1182 898 1510 1507" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px;"> <p>Note</p> <p>目前您只能修改 <code>SYS.CLEANUP_ONLINE_IND_BUILD</code> 和 <code>SYS.BSLN_MAINTAIN_STATS_JOB</code> 任務。</p> </div>
<code>attribute</code>	<code>varchar2</code>	'REPEAT_INTERVAL_NAME'	—	是	要修改的屬性。 若要修改任務的重複間隔，請指定 'REPEAT_INTERVAL'。

參數名稱	資料類型	有效值	預設	必要	描述
					若要修改任務的排程名稱，請指定 'SCHEDULE_NAME' 。
value	varchar2	根據所使用的屬性，有效的排程間隔或排程名稱。	–	是	屬性的新值。

修改 DBMS_SCHEDULER 任務

如要修改 Oracle Scheduler 的某些元件，請使用 Oracle 程序 `dbms_scheduler.set_attribute`。如需詳細資訊，請參閱 Oracle 文件中的 [DBMS_SCHEDULER](#) 和 [SET_ATTRIBUTE 程序](#)。

使用 Amazon RDS 資料庫執行個體時，請將結構描述名稱 SYS 加至物件名稱的前端。下列範例會設定 Monday 視窗物件的資源計劃屬性。

```
BEGIN
  DBMS_SCHEDULER.SET_ATTRIBUTE(
    name      => 'SYS.MONDAY_WINDOW',
    attribute => 'RESOURCE_PLAN',
    value     => 'resource_plan_1');
END;
/
```

修改 AutoTask 維護時段

Amazon RDS for Oracle 執行個體是以維護視窗的預設設定所建立的。在這些視窗期間執行自動維護任務，例如最佳化處理程式統計數字收集。依預設，維護視窗會開啟 Oracle 資料庫資源管理員。

如要修改視窗，請使用 DBMS_SCHEDULER 套件。您可能需要修改維護視窗設定，原因如下：

- 您希望在不同時間執行的維護任務，使用不同的設定，或完全不執行。例如，可能想要修改視窗持續時間，或變更重複時間和間隔。

- 您想要避免在維護期間啟用資源管理員的效能影響。例如，若已指定預設維護計劃，且在資料庫負載下時開啟維護視窗，您可能會看到等待事件，例如 `resmgr:cpu quantum`。此等待事件與資料庫資源管理員有關。您有下列選項：
 - 確保資料庫執行個體的離峰時間期間維護視窗處於作用中狀態。
 - 藉由將 `resource_plan` 屬性設定為空字串，停用預設維護計劃。
 - 將您參數群組中的 `resource_manager_plan` 參數設為 `FORCE:`。若您的執行個體使用企業版，此設定會防止資料庫資源管理員計劃的啟動。

如要修改您的維護視窗設定

1. 使用 Oracle SQL 用戶端連線至您的資料庫。
2. 查詢排程器視窗的目前組態。

下列範例會查詢 `MONDAY_WINDOW` 的組態。

```
SELECT ENABLED, RESOURCE_PLAN, DURATION, REPEAT_INTERVAL
FROM   DBA_SCHEDULER_WINDOWS
WHERE  WINDOW_NAME='MONDAY_WINDOW';
```

下列輸出顯示該窗口正使用預設值。

```
ENABLED          RESOURCE_PLAN          DURATION          REPEAT_INTERVAL
-----
TRUE             DEFAULT_MAINTENANCE_PLAN  +000 04:00:00
freq=daily;byday=MON;byhour=22
;byminute=0;
bysecond=0
```

3. 使用 `DBMS_SCHEDULER` 套件修改視窗。

下列範例會將資源計劃設定為 `null`，則資源管理員不會在維護視窗期間執行。

```
BEGIN
  -- disable the window to make changes
  DBMS_SCHEDULER.DISABLE(name=>'SYS"."MONDAY_WINDOW"', force=>TRUE);

  -- specify the empty string to use no plan
```

```
DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'SYS"."MONDAY_WINDOW"',
attribute=>'RESOURCE_PLAN', value=>');

-- re-enable the window
DBMS_SCHEDULER.ENABLE(name=>'SYS"."MONDAY_WINDOW"');
END;
/
```

下列範例會將視窗的最長持續時間設定為 2 小時。

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'SYS"."MONDAY_WINDOW"', force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'SYS"."MONDAY_WINDOW"',
attribute=>'DURATION', value=>'0 2:00:00');
  DBMS_SCHEDULER.ENABLE(name=>'SYS"."MONDAY_WINDOW"');
END;
/
```

下列範例會將重複間隔設定為每個星期一上午 10 點。

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'SYS"."MONDAY_WINDOW"', force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'SYS"."MONDAY_WINDOW"',
attribute=>'REPEAT_INTERVAL',
value=>'freq=daily;byday=MON;byhour=10;byminute=0;bysecond=0');
  DBMS_SCHEDULER.ENABLE(name=>'SYS"."MONDAY_WINDOW"');
END;
/
```

設定 Oracle Scheduler 任務的時區

若要修改 Oracle Scheduler 的時區，您可以使用 Oracle 程序 `dbms_scheduler.set_scheduler_attribute`。如需 `dbms_scheduler` 套件的詳細資訊，請參閱 Oracle 文件中的 [DBMS_SCHEDULER](#) 和 [SET_SCHEDULER_ATTRIBUTE](#)。

修改目前時區設定

1. 使用如 SQL Developer 的用戶端連線至資料庫。如需更多詳細資訊，請參閱 [使用 Oracle SQL Developer 連線至您的資料庫執行個體](#)。
2. 設置預設時區，如下所示，將您的時區替換為 `time_zone_name`。

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'time_zone_name'
  );
END;
/
```

在下列範例中，您將時區變更為亞洲/上海。

從查詢目前的時區開始，如下所示。

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE
  ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

輸出顯示目前的時區是 ETC/UTC。

```
VALUE
-----
Etc/UTC
```

然後您將時區設定為亞洲/上海。

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'Asia/Shanghai'
  );
END;
/
```

如需變更系統時區的詳細資訊，請參閱[Oracle 時區](#)。

關閉 SYS 擁有的 Oracle Scheduler 任務

若要停用 SYS 使用者擁有的 Oracle Scheduler 任務，請使用 `rdsadmin.rdsadmin_dbms_scheduler.disable` 程序。

這個程序會對 Oracle Scheduler 任務使用 `name` 一般參數。如需更多詳細資訊，請參閱 [Oracle Scheduler 程序的常見參數](#)。

下列範例會停用 SYS.CLEANUP_ONLINE_IND_BUILD Oracle Scheduler 任務。

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

開啟 SYS 擁有的 Oracle Scheduler 任務

若要開啟 STS 擁有的 Oracle Scheduler 任務，請使用 rdsadmin.rdsadmin_dbms_scheduler.enable 程序。

這個程序會對 Oracle Scheduler 任務使用 name 一般參數。如需更多詳細資訊，請參閱 [Oracle Scheduler 程序的常見參數](#)。

下列範例會啟用 SYS.CLEANUP_ONLINE_IND_BUILD Oracle Scheduler 任務。

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

針對 CALENDAR 類型的任務，修改 Oracle Scheduler 重複間隔

若要修改重複間隔以修改 CALENDAR 類型、系統擁有的 Oracle Scheduler 任務，請使用 rdsadmin.rdsadmin_dbms_scheduler.disable 程序。

這個程序會對 Oracle Scheduler 任務使用下列一般參數：

- name
- attribute
- value

如需更多詳細資訊，請參閱 [Oracle Scheduler 程序的常見參數](#)。

下列範例會修改 SYS.CLEANUP_ONLINE_IND_BUILD Oracle Scheduler 任務的重複間隔。

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
        name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
```

```
        attribute => 'repeat_interval',
        value      => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

針對 NAMED 類型的任務，修改 Oracle Scheduler 重複間隔

有些 Oracle Scheduler 任務會使用排程名稱，而非間隔。針對此類型的任務，您必須在主要使用者結構描述中建立新的指定排程。請使用標準的 Oracle `sys.dbms_scheduler.create_schedule` 程序來執行此動作。同時，使用 `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` procedure 將新命名的排程指派給任務。

這個程序會對 Oracle Scheduler 任務使用下列一般參數：

- name
- attribute
- value

如需更多詳細資訊，請參閱 [Oracle Scheduler 程序的常見參數](#)。

下列範例會修改 `SYS.BSLN_MAINTAIN_STATS_JOB` Oracle Scheduler 任務的重複間隔。

```
BEGIN
  DBMS_SCHEDULER.CREATE_SCHEDULE (
    schedule_name => 'rds_master_user.new_schedule',
    start_date    => SYSTIMESTAMP,
    repeat_interval =>
'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
    end_date      => NULL,
    comments      => 'Repeats daily forever');
END;
/

BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
    name          => 'SYS.BSLN_MAINTAIN_STATS_JOB',
    attribute     => 'schedule_name',
    value         => 'rds_master_user.new_schedule');
END;
/
```

建立 Oracle Scheduler 任務時關閉自動認可

DBMS_SCHEDULER.CREATE_JOB 建立 Oracle Scheduler 任務時，其會立即建立任務並認可變更。您可能需要在使用者交易中納入 Oracle Scheduler 任務的建立，來執行下列動作：

- 復原使用者交易後，復原 Oracle Schedule 任務。
- 認可主要使用者交易後，建立 Oracle Scheduler 任務。

您可以使用程序 `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` 開啟此行為。這個程序不需採用任何參數。您可以在下列 RDS for Oracle 版本中使用此程序：

- 21.0.0.0.ru-2022-07.rur-2022-07.r1 和更新版本
- 19.0.0.0.ru-2022-07.rur-2022-07.r1 和更新版本

下列範例會關閉 Oracle Scheduler 的自動認可、建立 Oracle Scheduler 任務，然後復原交易。因為自動認可已關閉，所以此資料庫也會復原 Oracle Scheduler 任務的建立。

```
BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag;
  DBMS_SCHEDULER.CREATE_JOB(job_name => 'EMPTY_JOB',
                             job_type => 'PLSQL_BLOCK',
                             job_action => 'begin null; end;',
                             auto_drop => false);

  ROLLBACK;
END;
/

PL/SQL procedure successfully completed.

SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME='EMPTY_JOB';

no rows selected
```

執行 Oracle 資料庫執行個體的一般診斷任務

Oracle 資料庫包含可用來調查資料庫問題的故障性基礎架構。在 Oracle 術語中，problem (問題) 是嚴重錯誤，例如程式碼錯誤或資料損毀。Incident (事件) 則是發生問題。如果同樣的錯誤發生三次，則基礎結構會顯示這個問題的三個事件。如需詳細資訊，請參閱 Oracle 資料庫文件中的[診斷和解決問題](#)。

自動診斷儲存器命令解譯器 (ADRCI) 公用程式是 Oracle 命令列工具，您可以用來管理診斷資料。例如，您可以使用這個工具來調查問題，並封裝診斷資料。incident package (事件套件) 包含事件或參考特定問題之所有事件的診斷資料。您可以將事件套件 (實作為 .zip 檔案) 上傳至 Oracle 支援。

為了提供受管的服務體驗，Amazon RDS 並不會提供 ADRCI 的 Shell 存取權。若要針對 Oracle 執行處理執行診斷作業，請改用 Amazon RDS 封裝 rdsadmin.rdsadmin_adrci_util。

透過使用 rdsadmin_adrci_util 中的函數，您可以列出並封裝問題和事件，也可以顯示追蹤檔案。所有函數都會傳回工作 ID。此 ID 構成包含 ADRCI 輸出的記錄檔名稱的一部分，如 dbtask-*task_id*.log 中所示。記錄檔位於 BDUMP 目錄中。您可以依照中所述的程序下載記錄檔 [下載資料庫日誌檔案](#)。

診斷程序的常見參數

若要執行診斷工作，請使用 Amazon RDS 封裝 rdsadmin.rdsadmin_adrci_util 中的函數。以下即為該套件的常見參數。

參數名稱	資料類型	有效值	預設	必要	描述
incident_id	數字	有效的事件 ID 或 null	Null	否	如果值為 null，則函數會顯示所有事件。如果值不是 null 且代表有效的事件 ID，則函數會顯示指定的事件。
problem_id	數字	有效的問題 ID 或 null	Null	否	如果該值為 null，該函數會顯示所有問題。如果值不為 null 且代表有效的問題 ID，則函數會顯示指定的問題。
last	數字	大於 0 或 null 的有效整數	Null	否	如果該值為空，則該函數最多顯示 50 個項目。如果該值不為 null，函數會顯示指定的數字。

列出事件

若要列出 Oracle 的診斷事件，請使用此 Amazon RDS 函數 `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents`。您可以在基本或詳細模式中列出事件。依預設，此函數會列出最近 50 個事件。

此函數使用下列一般參數：

- `incident_id`
- `problem_id`
- `last`

如果您指定 `incident_id` 和 `problem_id`，則 `incident_id` 會覆寫 `problem_id`。如需詳細資訊，請參閱[診斷程序的常見參數](#)。

此函數使用以下附加參數。

參數名稱	資料類型	有效值	預設	必要	描述
<code>detail</code>	布林值	TRUE 或 FALSE *	FALSE	否	如果 TRUE，該函數會以詳細模式列出事件。如果 FALSE，此函數會以基本模式列出事件。

如要列出所有事件，請查詢不含任何引數的

`rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` 函數。此查詢會傳回任務 ID。

```
SQL> SELECT rdsadmin.rdsadmin_adrci_util.list_adrci_incidents AS task_id FROM DUAL;

TASK_ID
-----
1590786706158-3126
```

或者呼叫不含任何引數的 `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` 函數，並將輸出存放在 SQL 用戶端變數中。您可以在其他陳述式中使用變數。

```
SQL> VAR task_id VARCHAR2(80);
```

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;

PL/SQL procedure successfully completed.
```

若要讀取日誌檔案，請呼叫 Amazon RDS 程序 `rdsadmin.rds_file_util.read_text_file`。提供任務 ID 做為檔案名稱的一部分。下列輸出顯示三個事件：53523、53522 和 53521。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
INCIDENT_ID PROBLEM_KEY                                     CREATE_TIME
-----
53523      ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
20:15:20.928000 +00:00
53522      ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
20:15:15.247000 +00:00
53521      ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
20:15:06.047000 +00:00
3 rows fetched

2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.
```

若要列出特定事件，請使用 `incident_id` 參數指定其 ID。在下列範例中，您只查詢事件 53523 的日誌檔案。

```
SQL> EXEC :task_id :=
rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.
```

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
INCIDENT_ID          PROBLEM_KEY
  CREATE_TIME
-----
53523                ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003
2020-05-29 20:15:20.928000 +00:00
1 rows fetched

2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.
```

列出問題

若要列出 Oracle 的診斷問題，請使用 Amazon RDS 函數 `rdsadmin.rdsadmin_adrci_util.list_adrci_problems`。

依預設，該函數會列出 50 個最近的問題。

此函數使用一般參數 `problem_id` 和 `last`。如需詳細資訊，請參閱 [診斷程序的常見參數](#)。

若要取得所有問題的任務 ID，請呼叫不含任何引數的 `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` 函數，並將輸出存放在 SQL 用戶端變數中。

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;

PL/SQL procedure successfully completed.
```

若要讀取日誌檔案，請呼叫 `rdsadmin.rds_file_util.read_text_file` 函數，並提供任務 ID 做為檔案名稱的一部分。在下列輸出中，日誌檔案會顯示三個問題：1、2 和 3。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
PROBLEM_ID   PROBLEM_KEY                                     LAST_INCIDENT
          LASTINC_TIME
-----
2           ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 53523
2020-05-29 20:15:20.928000 +00:00
3           ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
2020-05-29 20:15:15.247000 +00:00
1           ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 53521
2020-05-29 20:15:06.047000 +00:00
3 rows fetched

2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.

14 rows selected.
```

在下列範例中，您只列出問題 3。

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

若要讀取問題 3 的日誌檔案，請呼叫 `rdsadmin.rds_file_util.read_text_file`。提供任務 ID 做為檔案名稱的一部分。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
```

```

2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
PROBLEM_ID PROBLEM_KEY                                LAST_INCIDENT
LASTINC_TIME
-----
-----
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
2020-05-29 20:15:15.247000 +00:00
1 rows fetched

2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.

12 rows selected.

```

建立事件套件

您可以使用該 Amazon RDS 函數 `rdsadmin.rdsadmin_adrci_util.create_adrci_package` 建立事件套件。該輸出是一個您可以提供給 Oracle 支援的 .zip 檔案。

此函數使用下列一般參數：

- `problem_id`
- `incident_id`

請務必指定上述其中一個參數。如果您同時指定兩個參數，則 `incident_id` 會覆寫 `problem_id`。如需更多詳細資訊，請參閱 [診斷程序的常見參數](#)。

若要為特定事件建立套件，請使用 `rdsadmin.rdsadmin_adrci_util.create_adrci_package` 參數呼叫 Amazon RDS 函數 `incident_id`。下列範例會為事件 53523 建立套件。

```

SQL> EXEC :task_id :=
rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.

```

若要讀取日誌檔案，請呼叫 `rdsadmin.rds_file_util.read_text_file`。您可以提供任務 ID 做為檔案名稱的一部分。該輸出會顯示您產生的事件套件 `ORA700EVE_20200529212043_COM_1.zip`。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));
```

TEXT

```
-----
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsbdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

若要封裝特定問題的診斷資料，請使用 `problem_id` 參數指定其 ID。在下列範例中，您只封裝問題 3 的資料。

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);
```

```
PL/SQL procedure successfully completed.
```

若要讀取任務輸出，請呼叫 `rdsadmin.rds_file_util.read_text_file`，提供任務 ID 做為檔案名稱的一部分。該輸出會顯示您產生的事件套件 `ORA700EVE_20200529212111_COM_1.zip`。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));
```

TEXT

```
-----
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsbdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

您也可以下載記錄檔。如需詳細資訊，請參閱 [下載資料庫日誌檔案](#)。

顯示追蹤檔案

您可以使用 Amazon RDS 函數 `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile`，列出追蹤目錄下的追蹤檔案，以及目前 ADR 主目錄下所有事件目錄下的追蹤檔案。您也可以顯示追蹤檔案和事件追蹤檔案的內容。

此函數使用下列參數。

參數名稱	資料類型	有效值	預設	必要	描述
filename	varchar2	有效的追蹤檔案名稱	Null	否	如果值為 null，函數會顯示所有追蹤檔案。如果它不是 null，該函數顯示指定的檔案。

若要顯示追蹤檔案，請呼叫 Amazon RDS 函數

`rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile`。

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;
```

```
PL/SQL procedure successfully completed.
```

若要列出追蹤檔案名稱，請呼叫 Amazon RDS 程序

`rdsadmin.rds_file_util.read_text_file`，提供任務 ID 做為檔案名稱的一部分。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';
```

```
TEXT
```

```
-----
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log
```

```
9 rows selected.
```

在下列範例中，您會產生的輸出 `alert_ORCL.log`。

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/
orcl_a/ORCL/trace/alert_ORCL.log');
```



```
PL/SQL procedure successfully completed.
```

若要讀取日誌檔案，請呼叫 `rdsadmin.rds_file_util.read_text_file`。提供任務 ID 做為檔案名稱的一部分。該輸出會顯示 `alert_ORCL.log` 的前 10 行。

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;

TEXT
-----
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
Fri May 29 00:04:10 2020

10 rows selected.
```

您也可以下載記錄檔。如需詳細資訊，請參閱 [下載資料庫日誌檔案](#)。

執行 Oracle 資料庫執行處理的雜項任務

您可以在下文中找到如何在執行 Oracle 的 Amazon RDS 資料庫執行個體上執行雜項 DBA 任務。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的 Shell 存取權，而且會將存取權限制在某些需要進階權限的系統程序和資料表。

主題

- [在主要資料儲存空間中建立和捨棄目錄](#)
- [列出資料庫執行個體目錄中的檔案](#)
- [讀取資料庫執行個體目錄中的檔案](#)
- [存取 Opatch 檔案](#)
- [管理建議程式任務](#)
- [可傳輸的資料表空間](#)

在主要資料儲存空間中建立和捨棄目錄

若要建立目錄，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.create_directory`。您可以建立最多 10,000 個目錄，全位於您的主要資料儲存空間。若要建立目錄，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.drop_directory`。

`create_directory` 和 `drop_directory` 程序具有下列必要參數。

參數名稱	資料類型	預設	必要	描述
<code>p_directory_name</code>	VARCHAR2	—	是	目錄名稱。

下列範例會建立名為 `PRODUCT_DESCRIPTIONS` 的新目錄。

```
EXEC rdsadmin.rdsadmin_util.create_directory(p_directory_name =>
'product_descriptions');
```

資料字典會以大寫存放目錄名稱。您可以查詢 `DBA_DIRECTORIES` 來列出目錄。系統會自動選擇實際的主機路徑名稱。下列範例會取得名為 `PRODUCT_DESCRIPTIONS` 目錄的目錄路徑：

```
SELECT DIRECTORY_PATH
FROM DBA_DIRECTORIES
WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';

DIRECTORY_PATH
-----
/rdsdbdata/userdirs/01
```

資料庫執行個體的主要使用者名稱會具有新目錄中的讀寫權限，並且可將存取權授予其他使用者。EXECUTE 權限不適用於資料庫執行個體上的目錄。目錄會在您的主要資料儲存空間中建立，並且將耗用空間和輸入/輸出頻寬。

下列範例會捨棄名為 `PRODUCT_DESCRIPTIONS` 的目錄。

```
EXEC rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

Note

您也可以使用 Oracle SQL 命令 DROP DIRECTORY 捨棄目錄。

捨棄目錄不會移除其內容。這是因為 rdsadmin.rdsadmin_util.create_directory 程序可能會重複使用路徑名稱，而已捨棄目錄中的檔案可能會出現在新建立的目錄中。捨棄目錄之前，我們建議您使用 UTL_FILE.FREMOVE 從目錄中移除檔案。如需更多詳細資訊，請參閱 Oracle 文件中的 [FREMOVE 程序](#)。

列出資料庫執行個體目錄中的檔案

若要列出目錄中的檔案，請使用 Amazon RDS 程序 rdsadmin.rds_file_util.listdir。Oracle 複本不支援此程序。listdir 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_directory	varchar2	—	是	要列出的目錄名稱。

下列範例授予目錄 PRODUCT_DESCRIPTIONS 的讀/寫權限給使用者 rdsadmin，然後列出此目錄中的檔案。

```
GRANT READ,WRITE ON DIRECTORY PRODUCT_DESCRIPTIONS TO rdsadmin;
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
'PRODUCT_DESCRIPTIONS'));
```

讀取資料庫執行個體目錄中的檔案

若要讀取文字檔案，請使用 Amazon RDS 程序 rdsadmin.rds_file_util.read_text_file。read_text_file 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
p_directory	varchar2	—	是	包含檔案的目錄名稱。
p_filename	varchar2	—	是	要讀取的檔案名稱。

下列範例會在目錄 `rice.txt` 中建立檔案 `PRODUCT_DESCRIPTIONS`。

```
declare
  fh sys.utl_file.file_type;
begin
  fh := utl_file.fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
    open_mode=>'w');
  utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
  utl_file.fclose(file=>fh);
end;
/
```

下列範例會從目錄 `rice.txt` 讀取 `PRODUCT_DESCRIPTIONS` 檔案。

```
SELECT * FROM TABLE
  (rdsadmin.rds_file_util.read_text_file(
    p_directory => 'PRODUCT_DESCRIPTIONS',
    p_filename => 'rice.txt'));
```

存取 Opatch 檔案

Opatch 是可將修補程式套用和復原到 Oracle 軟體的 Oracle 公用程式。針對確定已將哪些修補程式套用至資料庫，Oracle 採用機制是 `opatch lsinventory` 命令。為針對使用自有授權 (BYOL) 客戶開立服務請求，Oracle 支援會要求 `lsinventory` 檔案，有時也會要求 Opatch 產生的 `lsinventory_detail` 檔案。

為了提供受管的服務體驗，Amazon RDS 並不會提供 Opatch 的 Shell 存取權。相反地，BDUMP 目錄中的 `lsinventory-dbv.txt` 包含與您目前引擎版本相關的修補程式資訊。當您執行次要或主要升級時，Amazon RDS 會在套用修補程式的一小時內更新 `lsinventory-dbv.txt`。如要確認套用的修補程式，請參閱 `lsinventory-dbv.txt`。此動作與執行 `opatch lsinventory` 命令類似。

Note

本節的範例假設 BDUMP 目錄已命名為 BDUMP。在僅供讀取複本上，BDUMP 目錄名稱不同。若要了解如何透過在僅供讀取複本上查詢 `V$DATABASE.DB_UNIQUE_NAME` 以取得 BDUMP 名稱，請參閱 [列出檔案](#)。

庫存檔案會使用 Amazon RDS 命名慣例 `lsinventory-dbv.txt` 和 `lsinventory_detail-dbv.txt`，其中 `dbv` 是資料庫版本的完整名稱。`lsinventory-dbv.txt`

檔案可在所有資料庫版本上使用。相應產品可 `lsinventory_detail-dbv.txt` 於 2020 年 1 月 19 日或更新版本提供。

例如，若您的資料庫版本為 `19.0.0.0.ru-2021-07.rur-2021-07.r1`，則您的庫存檔案具有下列名稱。

```
lsinventory-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt  
lsinventory_detail-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
```

確保您下載的檔案符合您資料庫引擎的目前版本。

主控台

如何使用主控台下載庫存檔案

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您想檢視的日誌檔案所在的資料庫執行個體的名稱。
4. 選擇 Logs & events (日誌與事件) 標籤。
5. 向下捲動至 Logs (日誌) 區段。
6. 在日誌區段中，搜尋 `lsinventory`。
7. 選取您要存取的檔案，然後選擇下載。

SQL

若要在 SQL 用戶端中讀取 `lsinventory-dbv.txt`，您可以使用 SELECT 陳述式。針對此技術，請使用下列 `rdsadmin` 函數：`rdsadmin.rds_file_util.read_text_file` 或 `rdsadmin.tracefile_listing`。

在下列範例查詢中，將 `dbv` 取代為您的 Oracle 資料庫版本。例如，您的資料庫版本可能是 `19.0.0.0.ru-2020-04.rur-2020-04.r1`。

```
SELECT text  
FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

若要在 SQL 用戶端中讀取 `lsinventory-dbv.txt`，您可以寫入 PL/SQL 程式。此程式會使用 `utl_file` 來讀取檔案，並使用 `dbms_output` 來列印檔案。這些是 Oracle 提供的套件。

在下列範例程式中，將 *dbv* 取代為您的 Oracle 資料庫版本。例如，您的資料庫版本可能是 19.0.0.0.ru-2020-04.rur-2020-04.r1。

```
SET SERVEROUTPUT ON
DECLARE
  v_file          SYS.UTL_FILE.FILE_TYPE;
  v_line          VARCHAR2(1000);
  v_oracle_home_type VARCHAR2(1000);
  c_directory     VARCHAR2(30) := 'BDUMP';
  c_output_file   VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
    EXCEPTION
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

或者，查詢 `rdsadmin.tracefile_listing`，然後將輸出多工緩衝處理至檔案。下列範例會將輸出多工緩衝處理至 `/tmp/tracefile.txt`。

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

管理建議程式任務

Oracle 資料庫包含許多建議程式。每個建議程式均支援自動化和手動任務。您可以使用 `rdsadmin.rdsadmin_util` 套件來管理一些建議程式任務。

下列引擎版本提供建議程式任務程序：

- Oracle Database 21c (21.0.0)

- 19.0.0.0.ru-2021-01.rur-2021-01.r1 版及更新的 Oracle Database 19c 版本

如需詳細資訊，請參閱 Amazon RDS for Oracle 版本備註中的 [版本 19.0.0.0.ru-2021-01.rur-2021-01.r1](#)。

主題

- [設定建議程式任務的參數](#)
- [停用 AUTO_STATS_ADVISOR_TASK](#)
- [重新啟用 AUTO_STATS_ADVISOR_TASK](#)

設定建議程式任務的參數

若要設定某些建議程式任務的參數，請使用 Amazon RDS 程序

`rdsadmin.rdsadmin_util.advisor_task_set_parameter`。 `advisor_task_set_parameter` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_task_name</code>	<code>varchar2</code>	—	是	您要變更其參數的建議程式任務名稱。有效值如下： <ul style="list-style-type: none"> • <code>AUTO_STATS_ADVISOR_TASK</code> • <code>INDIVIDUAL_STATS_ADVISOR_TASK</code> • <code>SYS_AUTO_SPM_EVOLVE_TASK</code> • <code>SYS_AUTO_SQL_TUNING_TASK</code>
<code>p_parameter</code>	<code>varchar2</code>	—	是	任務參數的名稱。若要尋找建議程式任務的有效參數，請執行下列查詢。用 <code>p_task_name</code> 有效值替代 <i><code>p_task_name</code></i> ： <pre>COL PARAMETER_NAME FORMAT a30 COL PARAMETER_VALUE FORMAT a30 SELECT PARAMETER_NAME, PARAMETER_VALUE FROM DBA_ADVISOR_PARAMETERS WHERE TASK_NAME=' <i>p_task_name</i> '</pre>

參數名稱	資料類型	預設	必要	描述
				<pre>AND PARAMETER_VALUE != 'UNUSED' ORDER BY PARAMETER_NAME;</pre>
p_value	varchar2	—	是	<p>任務參數的值。若要尋找任務參數的有效值，請執行下列查詢。用 p_task_name 有效值替代 <i>p_task_name</i>：</p> <pre>COL PARAMETER_NAME FORMAT a30 COL PARAMETER_VALUE FORMAT a30 SELECT PARAMETER_NAME, PARAMETER _VALUE FROM DBA_ADVISOR_PARAMETERS WHERE TASK_NAME=' <i>p_task_name</i> ' AND PARAMETER_VALUE != 'UNUSED' ORDER BY PARAMETER_NAME;</pre>

對於 ACCEPT_PLANS，下列 PL/SQL 程式將 FALSE 設定為 SYS_AUTO_SPM_EVOLVE_TASK。SQL 計劃管理自動化任務會驗證計劃，並產生其問題清單的報告，但不會自動演進計劃。您可以使用報告來識別新的 SQL 計劃基準並手動接受它們。

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value      => 'FALSE');
END;
```

對於 EXECUTION_DAYS_TO_EXPIRE，下列 PL/SQL 程式將 10 設定為 AUTO_STATS_ADVISOR_TASK。預先定義的任務 AUTO_STATS_ADVISOR_TASK 每天會在維護視窗中自動執行一次。此範例會將任務執行的保留期設定為 10 天。

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value      => '10');
```



```
END;
```

停用 AUTO_STATS_ADVISOR_TASK

若要停用 AUTO_STATS_ADVISOR_TASK，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.advisor_task_drop`。`advisor_task_drop` 程序接受下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_task_name</code>	<code>varchar2</code>	—	是	待停用建議程式任務的名稱。唯一有效的值為 <code>AUTO_STATS_ADVISOR_TASK</code> 。

下列命令會捨棄 AUTO_STATS_ADVISOR_TASK。

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK');
```

您可以使用 `rdsadmin.rdsadmin_util.dbms_stats_init` 重新啟用 AUTO_STATS_ADVISOR_TASK。

重新啟用 AUTO_STATS_ADVISOR_TASK

若要重新啟用 AUTO_STATS_ADVISOR_TASK，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.dbms_stats_init`。`dbms_stats_init` 程序不需採用任何參數。

下列命令會重新啟用 AUTO_STATS_ADVISOR_TASK。

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init();
```

可傳輸的資料表空間

使用 Amazon RDS 套件 `rdsadmin.rdsadmin_transport_util`，將一組資料表空間從內部部署 Oracle 資料庫複製到 RDS for Oracle 資料庫執行個體。在實體層級，可傳輸資料表空間功能可將來源資料檔案與中繼資料檔案逐步複製到您的目標執行個體。您可以使用 Amazon EFS 或 Amazon S3 傳輸檔案。如需詳細資訊，請參閱 [使用 Oracle 可傳輸資料表空間進行移轉](#)。

主題

- [將傳輸資料表空間匯入資料庫執行個體](#)
- [將可傳輸資料表空間中繼資料匯入您的資料庫執行個體](#)

- [在資料表空間匯入後列出遺棄的檔案](#)
- [在資料表空間匯入後刪除遺棄的資料檔案](#)

將傳輸資料表空間匯入資料庫執行個體

使用程序 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`，從來源資料庫執行個體還原先前匯出的資料表空間。在傳輸階段中，您會備份唯讀資料表空間並匯出 Data Pump 中繼資料、將這些檔案傳輸到目標資料庫執行個體，然後匯入資料表空間。如需詳細資訊，請參閱 [階段 4：傳輸資料表空間](#)。

語法

```
FUNCTION import_xtts_tablespaces(
  p_tablespace_list IN CLOB,
  p_directory_name  IN VARCHAR2,
  p_platform_id     IN NUMBER DEFAULT 13,
  p_parallel        IN INTEGER DEFAULT 0) RETURN VARCHAR2;
```

參數

參數名稱	資料類型	預設	必要	描述
<code>p_tablespace_list</code>	CLOB	—	是	要匯入的資料表空間清單。
<code>p_directory_name</code>	VARCHAR2	—	是	包含資料表空間備份的目錄。
<code>p_platform_id</code>	NUMBER	13	否	提供與備份階段中指定平台 ID 相符的 ID。若要尋找平台列表，請查詢 <code>V\$TRANSPORTABLE_PLATFORM</code> 。預設平台是 Linux x86 64 位元，這是小端。
<code>p_parallel</code>	INTEGER	0	否	平行處理的程度。根據預設，會停用平行處理。

範例

以下範例從目錄 `DATA_PUMP_DIR` 匯入資料表 `TBS1`、`TBS2` 和 `TBS3`。來源平台是以 AIX 為基礎的系統 (64 位元)，其平台識別碼為 6。您可以通過查詢找到平台 ID `V$TRANSPORTABLE_PLATFORM`。

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
    'TBS1,TBS2,TBS3',
    'DATA_PUMP_DIR',
    p_platform_id => 6);
END;
/

PRINT task_id
```

將可傳輸資料表空間中繼資料匯入您的資料庫執行個體

使用 `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` 程序將可傳輸的資料表空間中繼資料匯入 RDS for Oracle 資料庫執行個體。在操作期間，資料表 `rdsadmin.rds_xtts_operation_info` 會顯示中繼資料匯入的狀態。如需詳細資訊，請參閱 [步驟 5：匯入目標資料庫執行個體上的資料表空間中繼資料](#)。

語法

```
PROCEDURE import_xtts_metadata(
  p_datapump_metadata_file IN SYS.DBA_DATA_FILES.FILE_NAME%TYPE,
  p_directory_name         IN VARCHAR2,
  p_exclude_stats          IN BOOLEAN DEFAULT FALSE,
  p_remap_tablespace_list  IN CLOB DEFAULT NULL,
  p_remap_user_list        IN CLOB DEFAULT NULL);
```

參數

參數名稱	資料類型	預設	必要	描述
<code>p_datapump_metadata_file</code>	<code>SYS.DBA_DATA_FILES.FILE_NAME%TYPE</code>	—	是	包含可傳輸資料表空間描述資料的 Oracle 資料汲取檔案名稱。

參數名稱	資料類型	預設	必要	描述
p_directory_name	VARCHAR2	—	是	包含資料汲取檔案的目錄名稱。
p_exclude_stats	BOOLEAN	FALSE	否	指示是否排除統計資料的標誌。
p_remap_tablespace_list	CLOB	NULL	否	在中繼資料匯入期間要重新對應的資料表空間清單。使用 <i>from_tbs:to_tbs</i> 格式。例如，指定 <code>users:user_data</code> 。
p_remap_user_list	CLOB	NULL	否	在中繼資料匯入期間要重新對應的資料表空間清單。使用 <i>from_schema_name:to_schema_name</i> 格式。例如，指定 <code>hr:human_resources</code> 。

範例

此範例會從檔案匯入資料表空間中繼資料 *xtdump.dmp*，它位於目錄中 *DATA_PUMP_DIR*。

```
BEGIN
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xtdump.dmp','DATA_PUMP_DIR');
END;
/
```

在資料表空間匯入後列出遺棄的檔案

使用 `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` 列出在資料表空間匯入之後遺棄的資料檔的程序。識別出資料文件後，您可以通過呼叫 `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` 將其刪除。

語法

```
FUNCTION list_xtts_orphan_files RETURN xtts_orphan_files_list_t PIPELINED;
```

範例

下列範例執行 `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`。輸出會顯示兩個遺棄的資料檔案。

```
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
```

FILENAME	FILESIZE
-----	-----
datafile_7.dbf	104865792
datafile_8.dbf	104865792

在資料表空間匯入後刪除遺棄的資料檔案

使用 `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` 程序刪除在資料表空間匯入之後遺棄的資料檔案。執行此命令會產生使用 BDUMP 目錄中名稱格式的日誌檔 `rds-xtts-delete_xtts_orphaned_files-YYYY-MM-DD.HH24-MI-SS.FF.log`。使用 `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` 程序找到遺棄的檔案。您可以透過呼叫 `rdsadmin.rds_file_util.read_text_file` 程序來讀取日誌檔。如需詳細資訊，請參閱 [階段 6：清理剩餘檔案](#)。

語法

```
PROCEDURE cleanup_incomplete_xtts_import(  
    p_directory_name IN VARCHAR2);
```

參數

參數名稱	資料類型	預設	必要	描述
p_directory_name	VARCHAR2	—	是	包含遺棄的資料檔案的目錄。

範例

下列範例會刪除 *DATA_PUMP_DIR* 中遺棄的檔案。

```
BEGIN
  rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

下列範例會讀取上一個命令所產生的日誌檔。

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
  p_directory => 'BDUMP',
  p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
-----
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

設定進階 RDS for Oracle 功能

RDS for Oracle 支援各種進階功能，包括 HugePages、執行個體儲存體和延伸資料類型。

主題

- [將暫存資料存放在 RDS for Oracle 執行個體儲存體中](#)
- [對 RDS for Oracle 執行個體開啟巨型分頁](#)
- [在 RDS for Oracle 中開啟擴充資料類型](#)

將暫存資料存放在 RDS for Oracle 執行個體儲存體中

在支援的 RDS for Oracle DB 資料庫執行個體類別上，將執行個體儲存體用於暫存資料表空間和資料庫智慧型快閃記憶體快取 (快閃記憶體快取)。

主題

- [RDS for Oracle 執行個體儲存體的概觀](#)
- [開啟 RDS for Oracle 執行個體儲存體](#)
- [設定 RDS for Oracle 執行個體儲存體](#)
- [變更資料庫執行個體類型時的考量](#)
- [使用 Oracle 僅供讀取複本上的執行個體儲存體](#)
- [在執行個體儲存體和 Amazon EBS 上設定暫存資料表空間群組](#)
- [移除 RDS for Oracle 執行個體儲存體](#)

RDS for Oracle 執行個體儲存體的概觀

「執行個體儲存體」為 RDS for Oracle 資料庫執行個體提供暫時的區塊層級儲存體。您可以使用執行個體儲存體來暫時儲存經常變更的資訊。

執行個體儲存體是以實際連接至主機電腦的非揮發性記憶體儲存裝置 (NVMe) 為基礎。此儲存體已針對低延遲、隨機 I/O 效能，以及循序讀取輸送量進行最佳化。

執行個體儲存體的大小會因資料庫執行個體類型而有所不同。如需執行個體儲存體的詳細資訊，請參閱《適用於 Linux 執行個體的 Amazon Elastic Compute Cloud 使用者指南》中的 [Amazon EC2 執行個體儲存體](#)。

主題

- [RDS for Oracle 執行個體儲存體中的資料類型](#)
- [RDS for Oracle 執行個體儲存體的好處](#)
- [RDS for Oracle 執行個體儲存體支援的執行個體類別](#)
- [RDS for Oracle 執行個體儲存體支援的引擎版本](#)
- [RDS for Oracle 執行個體儲存體支援的 AWS 區域](#)
- [RDS for Oracle 執行個體儲存體的成本](#)

RDS for Oracle 執行個體儲存體中的資料類型

您可以將下列類型的 RDS for Oracle 暫存資料放置在執行個體儲存體中：

暫存資料表空間

Oracle 資料庫會使用暫存資料表空間，存放不適合記憶體的中繼查詢結果。較大的查詢可能會產生大量中繼資料，這些資料需要暫時快取，但不需要持續存在。尤其，暫存資料表空間對於排序、雜湊彙總和聯結很有用。如果 RDS for Oracle 資料庫執行個體使用企業版或標準版 2，您可以在執行個體儲存體中放置暫存資料表空間。

快閃記憶體快取

快閃記憶體快取可改善傳統路徑中單一區塊隨機讀取的效能。最佳實務是調整快取的大小，以容納大部分的作用中資料集。如果 RDS for Oracle 資料庫執行個體使用企業版，您可以在執行個體儲存體中放置快閃記憶體快取。

根據預設，執行個體儲存體會針對暫存資料表空間設定，但不會針對快閃記憶體快取設定。您無法在執行個體儲存體中放置 Oracle 資料檔和資料庫日誌檔。

RDS for Oracle 執行個體儲存體的好處

您可能會考慮使用執行個體儲存體，存放可能遺失的暫存檔案和快取。如果您想要改善資料庫效能，或是如果不斷增加的工作負載導致 Amazon EBS 儲存體發生效能問題，請考慮擴展到支援執行個體儲存體的執行個體類別。

將暫存資料表空間和快閃記憶體快取放置在執行個體儲存體上，您可以獲得下列好處：

- 較低的讀取延遲
- 較高的輸送量
- 減少 Amazon EBS 磁碟區的負載

- 降低 Amazon EBS 負載，因為減少了 Amazon EBS 負載
- 更不需要佈建高 IOPS，可能會降低整體成本

將暫存資料表空間放置在執行個體儲存體上，您可以立即提升使用暫存空間的查詢效能。將快閃記憶體快取放置在執行個體儲存體時，快取區塊讀取的延遲通常比 Amazon EBS 讀取的延遲低得多。快閃記憶體快取必須先「熱身」，然後才能提供效能優勢。快取會自行熱身，因為資料庫會在區塊老化而離開資料庫緩衝區快取時，將這些區塊寫入至快閃記憶體快取。

Note

在某些情況下，快閃記憶體快取會由於快取管理而造成效能負荷。在生產環境中開啟快閃記憶體快取之前，建議您先分析工作負載並在測試環境中測試快取。

RDS for Oracle 執行個體儲存體支援的執行個體類別

Amazon RDS 支援下列資料庫執行個體類別的執行個體儲存體：

- db.m5d
- db.r5d
- db.x2idn
- db.x2iedn

RDS for Oracle 僅支援 BYOL 授權模式的上述資料庫執行個體類別。如需詳細資訊，請參閱[支援的 RDS for Oracle 執行個體類別](#)及為 [EE](#) 和 [SE2](#) 攜帶您自己的授權 (BYOL)。

若要查看所支援資料庫執行個體類型的總執行個體儲存體，請在 AWS CLI 中執行下列命令。

Example

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=*5d.*large*" \
  --query "InstanceTypes[?contains(InstanceType, 'm5d')||contains(InstanceType, 'r5d')]" \
  [InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

上述命令會傳回執行個體儲存體的原始裝置大小。RDS for Oracle 會將這個空間的一小部分用於組態。執行個體儲存體中可供暫存資料表空間或快閃記憶體快取使用的空間略小。

RDS for Oracle 執行個體儲存體支援的引擎版本

下列 RDS for Oracle 引擎版本支援執行個體儲存體。

- 21.0.0.0.ru-2022-01.rur-2022-01.r1 或更高的 Oracle Database 21c 版本
- 19.0.0.0.ru-2021-10.rur-2021-10.r1 或更高的 Oracle Database 19c 版本

RDS for Oracle 執行個體儲存體支援的 AWS 區域

執行個體儲存體適用於支援一或多個這些執行個體類型的所有 AWS 區域。如需 db.m5d 和 db.r5d 執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。如需 Amazon RDS for Oracle 支援之執行個體類別的詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。

RDS for Oracle 執行個體儲存體的成本

執行個體儲存體的成本內建於執行個體儲存體開啟的執行個體成本中。在 RDS for Oracle 資料庫執行個體上啟用執行個體儲存體，不會產生額外成本。如需執行個體儲存體開啟的執行個體的詳細資訊，請參閱 [RDS for Oracle 執行個體儲存體支援的執行個體類別](#)。

開啟 RDS for Oracle 執行個體儲存體

若要開啟 RDS for Oracle 暫存資料的執行個體儲存體，請執行下列其中一個動作：

- 使用支援的執行個體類別建立 RDS for Oracle 資料庫執行個體。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改現有的 RDS for Oracle 資料庫執行個體來使用支援的執行個體類別。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

設定 RDS for Oracle 執行個體儲存體

根據預設，100% 的執行個體儲存體空間會配置給暫存資料表空間。若要設定執行個體儲存體，將空間配置給快閃記憶體快取和暫存資料表空間，請在執行個體的參數群組中設定下列參數：

```
db_flash_cache_size={DBInstanceStore*{0,2,4,6,8,10}/10}
```

此參數指定配置給快閃記憶體快取的儲存空間量。此參數僅適用於 Oracle 資料庫企業版。預設值為 {DBInstanceStore*0/10}。如果您為 db_flash_cache_size 設定非零值，則您的 RDS for Oracle 執行個體會在重新啟動執行個體之後啟用快閃記憶體快取。

`rds.instance_store_temp_size={DBInstanceStore*{0,2,4,6,8,10}/10}`

此參數指定配置給暫存資料表空間的儲存空間量。預設值為 $\{DBInstanceStore*10/10\}$ 。此參數對 Oracle 資料庫企業版而言是可修改的，對於標準版 2 而言則是唯讀的。如果您為 `rds.instance_store_temp_size` 設定非零值，Amazon RDS 會在執行個體儲存體中為暫存資料表空間配置空間。

您可以針對不使用執行個體儲存體的資料庫執行個體設定 `db_flash_cache_size` 和 `rds.instance_store_temp_size` 參數。在此情況下，這兩個設定都會評估為 0，這會關閉功能。在此情況下，您可以針對不同的執行個體大小和未使用執行個體儲存體的執行個體使用相同的參數群組。如果您修改這些參數，請務必重新啟動相關聯的執行個體，以便變更可以生效。

Important

如果您為暫存資料表空間配置空間，則 Amazon RDS 不會自動建立暫存資料表空間。若要了解如何在執行個體儲存體上建立暫存資料表空間，請參閱 [在執行個體儲存體上建立暫存資料表空間](#)。

上述參數的合併值不得超過 10/10 或 100%。下表說明有效和無效的參數設定。

<code>db_flash_cache_size</code> setting	<code>rds.instance_store_temp_size</code> setting	說明
<code>db_flash_cache_size={DBInstanceStore*0/10}</code>	<code>rds.instance_store_temp_size={DBInstanceStore*10/10}</code>	這是所有 Oracle 資料庫版本的有效組態。Amazon RDS 會將 100% 的執行個體儲存體空間配置給暫存資料表空間。此為預設值。
<code>db_flash_cache_size={DBInstanceStore*10/10}</code>	<code>rds.instance_store_temp_size={DBInstanceStore*0/10}</code>	這是僅適用於 Oracle 資料庫企業版的組態。Amazon RDS 會將 100% 的執

db_flash_cache_size setting	rds.instance_store_temp_size setting	說明
		行個體儲存體空間配置給快閃記憶體快取。
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	這是僅適用於 Oracle 資料庫企業版的組態。Amazon RDS 會將 20% 的執行個體儲存體空間配置給快閃記憶體快取，而將 80% 的執行個體儲存體空間配置給暫存資料表空間。
db_flash_cache_size={DBInstanceStore*6/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	這是僅適用於 Oracle 資料庫企業版的組態。Amazon RDS 會將 60% 的執行個體儲存體空間配置給快閃記憶體快取，而將 40% 的執行個體儲存體空間配置給暫存資料表空間。

db_flash_cache_size setting	rds.instance_store_temp_size setting	說明
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	這是僅適用於 Oracle 資料庫企業版的組態。Amazon RDS 會將 20% 的執行個體儲存體空間配置給快閃記憶體快取，而將 40% 的執行個體儲存體空間配置給暫存資料表空間。
db_flash_cache_size={DBInstanceStore*8/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	這是無效的組態，因為執行個體儲存體空間的合併百分比超過 100%。在此情況下，Amazon RDS 無法嘗試。

變更資料庫執行個體類型時的考量

如果您變更資料庫執行個體類型，其可能會影響快閃記憶體快取的組態或執行個體儲存體上的暫存資料表空間。請考慮下列修改及其影響：

您可以針對支援執行個體儲存體的資料庫執行個體進行縱向擴展或縮減規模。

下列值會隨著執行個體儲存體的新大小，按比例增加或減少：

- 快閃記憶體快取的新大小。
- 配置給位於執行個體儲存體之暫存資料表空間的空間。

例如，db.m5d.4xlarge 執行個體上的設定

db_flash_cache_size={DBInstanceStore*6/10} 會提供約 340 GB 的快閃記憶體快取

空間。如果您將執行個體類型縱向擴展至 db.m5d.8xlarge，快閃記憶體快取空間會增加到約 680 GB。

您可以將不使用執行個體儲存體的資料庫執行個體修改為確實使用執行個體儲存體的執行個體。

如果 `db_flash_cache_size` 設定為大於 0 的值，則會設定快閃記憶體快取。如果 `rds.instance_store_temp_size` 設定為大於 0 的值，則會配置執行個體儲存體空間，供暫存資料表空間使用。RDS for Oracle 不會自動將暫存檔移至執行個體儲存體。如需使用所配置空間的相關資訊，請參閱 [在執行個體儲存體上建立暫存資料表空間](#) 或 [將暫存檔新增至僅供讀取複本上的執行個體儲存體](#)。

您可以將使用執行個體儲存體的資料庫執行個體修改為不使用執行個體儲存體的執行個體。

在此情況下，RDS for Oracle 會移除快閃記憶體快取。RDS 會重新建立目前位於 Amazon EBS 磁碟區上執行個體儲存體的暫存檔。新暫存檔的大小上限是 `rds.instance_store_temp_size` 參數的先前大小。

使用 Oracle 僅供讀取複本上的執行個體儲存體

僅供讀取複本支援執行個體儲存體上的快閃記憶體快取和暫存資料表空間。雖然快閃記憶體快取的運作方式與主要資料庫執行個體相同，但請注意下列暫存資料表空間的差異：

- 您無法在僅供讀取複本上建立暫存資料表空間。如果您在主要執行個體上建立新的暫存資料表空間，RDS for Oracle 會複寫不含暫存檔的資料表空間資訊。若要新增暫存檔，請使用下列任一技巧：
 - 使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.add_inst_store_tempfile` RDS for Oracle 會在僅供讀取複本的執行個體儲存體中建立暫存檔，並將其新增至指定的暫存資料表空間。
 - 執行 `ALTER TABLESPACE ... ADD TEMPFILE` 命令。RDS for Oracle 將暫存檔放置在 Amazon EBS 儲存體上。

Note

主要資料庫執行個體和僅供讀取複本上的暫存檔大小和儲存類型可能不同。

- 您只能在主要資料庫執行個體上管理預設暫存資料表空間設定。RDS for Oracle 會將設定複寫到所有僅供讀取複本。
- 您只能在主要資料庫執行個體上設定暫存資料表空間群組。RDS for Oracle 會將設定複寫到所有僅供讀取複本。

在執行個體儲存體和 Amazon EBS 上設定暫存資料表空間群組

您可以設定暫存資料表空間群組，在執行個體儲存體和 Amazon EBS 上同時包含暫存資料表空間。當您想要的暫存儲存體超過 `rds.instance_store_temp_size` 最大設定允許的空間時，此技巧很有用。

當您同時在執行個體儲存體和 Amazon EBS 上設定暫存資料表空間群組時，這兩個表格空間的效能特性明顯不同。Oracle 資料庫會根據內部演算法選擇要為查詢提供服務的資料表空間。因此，類似的查詢可能會在效能方面有所不同。

一般而言，您可以在執行個體儲存體中建立暫存資料表空間，如下所示：

1. 在執行個體儲存體中建立暫存資料表空間。
2. 將新的表格空間設定為資料庫預設暫存資料表空間。

如果執行個體儲存體中的資料表空間大小不足，您可以建立額外的暫存儲存體，如下所示：

1. 將執行個體儲存體中的暫存表格空間指派給暫存資料表空間群組。
2. 請在 Amazon EBS 中建立新的暫存資料表空間 (如果不存在)。
3. 將 Amazon EBS 中的暫存資料表空間指派給包含執行個體儲存體資料表空間的相同資料表空間群組。
4. 將表格空間群組設定為預設暫存資料表空間。

下列範例假設執行個體儲存體中的暫存資料表空間大小不符合您的應用程式要求。此範例會在執行個體儲存體中建立暫存資料表空間 `temp_in_inst_store`、將其指派給資料表空間群組 `temp_group`、將名為 `temp_in_ebs` 的現有 Amazon EBS 資料表空間新增至這個群組，然後將此群組設定為預設暫存資料表空間。

```
SQL> EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace('temp_in_inst_store');  
  
PL/SQL procedure successfully completed.  
  
SQL> ALTER TABLESPACE temp_in_inst_store TABLESPACE GROUP temp_group;  
  
Tablespace altered.  
  
SQL> ALTER TABLESPACE temp_in_ebs TABLESPACE GROUP temp_group;  
  
Tablespace altered.
```

```
SQL> EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace('temp_group');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM DBA_TABLESPACE_GROUPS;

GROUP_NAME                                TABLESPACE_NAME
-----
TEMP_GROUP                                TEMP_IN_EBS
TEMP_GROUP                                TEMP_IN_INST_STORE

SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE
PROPERTY_NAME='DEFAULT_TEMP_TABLESPACE';

PROPERTY_VALUE
-----
TEMP_GROUP
```

移除 RDS for Oracle 執行個體儲存體

若要移除執行個體存放區，請修改您的 RDS for Oracle 資料庫執行個體，以使用不支援執行個體儲存體的執行個體類型，例如 db.m5 或 db.r5。

對 RDS for Oracle 執行個體開啟巨型分頁

Amazon RDS for Oracle 支援 Linux 核心巨型分頁，以提高資料庫可擴展性。巨型分頁會造成較小型的分頁資料表，以及耗費在記憶體管理方面較少的 CPU，增加大型資料庫執行個體的效能。如需詳細資訊，請參閱 Oracle 文件中的[巨型分頁的概觀](#)。

您可以使用 HugePages 搭配 RDS for Oracle 的所有支援版本：

`use_large_pages` 參數可控制是否已對資料庫執行個體開啟巨型分頁。此參數可能的設定為 `ONLY`、`FALSE` 和 `{DBInstanceClassHugePagesDefault}`。在 Oracle 的預設資料庫參數群組中，`use_large_pages` 參數是設為 `{DBInstanceClassHugePagesDefault}`。

若要控制是否對資料庫執行個體自動開啟巨型分頁，您可以在參數群組中使用 `DBInstanceClassHugePagesDefault` 公式變數。該值的判斷方式如下所示：

- 針對下表提及的資料庫執行個體類別，依預設 `DBInstanceClassHugePagesDefault` 一律會評估為 `FALSE`，而 `use_large_pages` 則評估為 `FALSE`。如果資料庫執行個體類別擁有至少 14 GiB 的記憶體，則您可對這些資料庫執行個體手動開啟巨型分頁。

- 針對下表未提及的資料庫執行個體類別，如果執行個體類別有少於 14 GiB 的記憶體，DBInstanceClassHugePagesDefault 一律會評估為 FALSE。此外，use_large_pages 評估為 FALSE。
- 針對下表未提及的資料庫執行個體類別，如果執行個體類別有至少 14 GiB，且少於 100 GiB 的記憶體，依預設 DBInstanceClassHugePagesDefault 會評估為 TRUE。此外，use_large_pages 評估為 ONLY。您可將 use_large_pages 設定為 FALSE，手動關閉巨型分頁。
- 針對下表未提及的資料庫執行個體類別，如果執行個體類別有至少 100 GiB 的記憶體，DBInstanceClassHugePagesDefault 一律會評估為 TRUE。此外，use_large_pages 評估為 ONLY 且無法停用巨型分頁。

針對下列資料庫執行個體類別，依預設不會開啟巨型分頁。

資料庫執行個體類別系列	依預設不會開啟具有巨型分頁的資料庫執行個體類別
db.m5	db.m5.large
db.m4	db.m4.large、db.m4.xlarge、db.m4.2xlarge、db.m4.4xlarge、db.m4.10xlarge
db.t3	db.t3.micro、db.t3.small、db.t3.medium、db.t3.large

如需資料庫執行個體類別的詳細資訊，請參閱 [的資料庫執行個體類別的硬體規格](#)。

若要對新的或現有的資料庫執行個體手動開啟巨型分頁，請將 use_large_pages 參數設定為 ONLY。您無法將巨型分頁用於 Oracle 自動記憶體管理 (AMM)。如果您將參數 use_large_pages 設定為 ONLY，則必須也將 memory_target 和 memory_max_target 設定為 0。如需設定資料庫執行個體之資料庫參數的詳細資訊，請參閱 [使用參數群組](#)。

您也可以設定 sga_target、sga_max_size 和 pga_aggregate_target 參數。設定系統全域區域 (SGA) 和程式全域區域 (PGA) 記憶體參數時，請將值一起新增。從您的可用執行個體記憶體 (DBInstanceClassMemory) 減去這個總計，以判斷巨型分頁配置以外的可用記憶體。您必須保留至少 2 GiB 的可用記憶體，或可用執行個體記憶體總數的 10%，以較小者為準。

設定參數之後，必須將您的資料庫執行個體重新開機，變更才能生效。如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

Note

Oracle 資料庫執行個體會延遲對 SGA 相關初始化參數的變更，直到您重新啟動執行個體而不進行容錯移轉為止。在 Amazon RDS 主控台中，選擇 Reboot (重新開機)，但不要選擇 Reboot with failover (使用容錯移轉重新開機)。在 AWS CLI 中，使用 `reboot-db-instance` 參數呼叫 `--no-force-failover` 命令。在容錯移轉期間或導致執行個體重新啟動的其他維護作業期間，資料庫執行個體不會處理 SGA 相關的參數。

下列是可手動啟用巨型分頁之巨型分頁的參數組態範例。您應該設定這些值以符合您的需求。

```
memory_target           = 0
memory_max_target      = 0
pga_aggregate_target   = {DBInstanceClassMemory*1/8}
sga_target              = {DBInstanceClassMemory*3/4}
sga_max_size           = {DBInstanceClassMemory*3/4}
use_large_pages        = ONLY
```

假設參數群組中設定了下列參數值。

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
  {DBInstanceClassMemory*3/4})
memory_max_target      = IF({DBInstanceClassHugePagesDefault}, 0,
  {DBInstanceClassMemory*3/4})
pga_aggregate_target   = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*1/8}, 0)
sga_target              = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*3/4}, 0)
sga_max_size           = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*3/4}, 0)
use_large_pages        = {DBInstanceClassHugePagesDefault}
```

參數群組由具有少於 100 GiB 記憶體之 `db.r4` 資料庫執行個體類別所使用。使用這些參數設定並將 `use_large_pages` 設定為 `{DBInstanceClassHugePagesDefault}`，巨型分頁會在 `db.r4` 執行個體上開啟。

考慮在參數群組中設定了下列參數值的另一個範例。

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
  {DBInstanceClassMemory*3/4})
```

```
memory_max_target      = IF({DBInstanceClassHugePagesDefault}, 0,
  {DBInstanceClassMemory*3/4})
pga_aggregate_target   = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*1/8}, 0)
sga_target             = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*3/4}, 0)
sga_max_size           = IF({DBInstanceClassHugePagesDefault},
  {DBInstanceClassMemory*3/4}, 0)
use_large_pages        = FALSE
```

參數群組是由 db.r4 DB 資料庫執行個體類別和 db.r5 資料庫執行個體類別所使用，兩個的記憶體都小於 100 GiB。使用這些參數設定，巨型分頁會在 db.r4 和 db.r5 執行個體上關閉。

Note

如果此參數群組是由具有至少 100 GiB 記憶體的 db.r4 資料庫執行個體類別或 db.r5 資料庫執行個體類別所使用，即會覆寫 FALSE 的 `use_large_pages` 設定，並將其設定為 ONLY。在此情況下，即會傳送關於覆寫的通知給客戶。

巨型分頁在您的資料庫執行個體上作用後，您可以透過啟用增強型監控來檢視巨型分頁資訊。如需更多詳細資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

在 RDS for Oracle 中開啟擴充資料類型

Amazon RDS for Oracle 支援擴充資料類型。使用擴充資料類型，VARCHAR2、NVARCHAR2 和 RAW 資料類型的大小上限是 32,767 個位元組。若要使用擴充資料類型，請將 MAX_STRING_SIZE 參數設為 EXTENDED。如需詳細資訊，請參閱 Oracle 文件中的 [擴充資料類型](#)。

如果不要使用擴充資料類型，請將 MAX_STRING_SIZE 參數保持設為 STANDARD (預設值)。在此情況下，VARCHAR2 和 NVARCHAR2 資料類型的大小限制為 4,000 個位元組，而 RAW 資料類型的大小限制為 2,000 個位元組。

您可以在新的或現有的資料庫執行個體上執行擴充資料類型。對於新的資料庫執行個體，開啟擴充資料類型時，資料庫執行個體建立時間一般較長。對於現有的資料庫執行個體，在轉換期間無法使用資料庫執行個體。

擴充資料類型的考量

針對資料庫執行個體啟用擴充資料類型時，請考慮下列事項：

- 開啟擴充資料類型時，您無法將資料庫執行個體變回使用資料類型的標準大小。將資料庫執行個體轉換為使用擴充資料類型後，如果重新將 MAX_STRING_SIZE 參數設定為 STANDARD，則會產生 incompatible-parameters 結果。
- 您還原使用擴充資料類型的資料庫執行個體時，必須將具有 MAX_STRING_SIZE 參數的參數群組設為 EXTENDED。在還原期間，如果您指定預設參數群組，或指定 MAX_STRING_SIZE 設為 STANDARD 的其他任何參數群組，則會產生 incompatible-parameters 狀態。
- 資料庫執行個體狀態由於 incompatible-parameters 設定而成為 MAX_STRING_SIZE 時，在您將 MAX_STRING_SIZE 參數設為 EXTENDED 並重新啟動資料庫執行個體之前，都無法使用資料庫執行個體。
- 對於 t2.micro 資料庫執行個體類別上執行的 Oracle 資料庫執行個體，建議不要開啟擴充資料類型。

針對新的資料庫執行個體開啟擴充資料類型

針對新的資料庫執行個體開啟擴充資料類型

1. 在參數群組中，將 MAX_STRING_SIZE 參數設為 EXTENDED。

若要設定參數，可以建立新的參數群組，也可以修改現有的參數群組。

如需更多詳細資訊，請參閱 [使用參數群組](#)。

2. 建立新的 RDS for Oracle 資料庫執行個體。

如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

3. 將 MAX_STRING_SIZE 設為 EXTENDED 的參數群組與資料庫執行個體建立關聯。

如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

針對現有的資料庫執行個體開啟擴充資料類型

修改資料庫執行個體以開啟擴充資料類型時，RDS 會將資料庫中的資料轉換為使用擴充大小。當您下次在參數變更之後重新啟動資料庫時，就會發生轉換和停機。在轉換期間無法使用資料庫執行個體。

轉換資料所需的時間量取決於資料庫執行個體類別、資料庫大小，以及最後一個資料庫快照的時間。若要減少停機時間，請考慮在重新開機之前立即建立快照。這樣可以縮短轉換工作流程期間備份的時間。

Note

開啟擴充資料類型後，無法在轉換期間執行時間的時間點還原。您可還原到緊接在轉換前或轉換後的時間。

針對現有的資料庫執行個體開啟擴充資料類型**1. 擷取資料庫的快照。**

如果資料庫有無效的物件，Amazon RDS 會嘗試重新編譯這些物件。如果 Amazon RDS 無法重新轉譯無效的物件，擴充資料類型的轉換可能會失敗。如果轉換發生問題，快照可讓您還原資料庫。務必在轉換前檢查無效的物件，並修正或捨棄無效的物件。對於生產資料庫，建議先對於其中一個資料庫執行個體複本測試轉換程序。

如需更多詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

2. 在參數群組中，將 MAX_STRING_SIZE 參數設為 EXTENDED。

若要設定參數，可以建立新的參數群組，也可以修改現有的參數群組。

如需更多詳細資訊，請參閱 [使用參數群組](#)。

3. 修改資料庫執行個體，以便與 MAX_STRING_SIZE 設為 EXTENDED 的參數群組產生關聯。

如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

4. 為了使參數變生效，請重新啟動資料庫執行個體。

如需更多詳細資訊，請參閱 [重新啟動中的資料庫執行個體](#)。

將資料匯入 Amazon RDS 上的 Oracle

將資料匯入 Amazon RDS for Oracle 資料庫執行個體的方式取決於下列項目：

- 您擁有的資料量
- 資料庫中資料庫物件的數量
- 資料庫中資料庫物件的多樣性

例如，您可以根據您的要求使用以下工具：

- Oracle SQL Developer — 匯入簡單的 20 MB 資料庫。
- Oracle Data Pump — 匯入複雜的資料庫，或是大小為數百 MB 或數 TB 的資料庫。例如，您可以將資料表空間從內部部署資料庫傳輸到 RDS for Oracle 資料庫執行個體。您可以使用 Amazon S3 或 Amazon EFS 來傳輸資料檔案和中繼資料。如需詳細資訊，請參閱 [使用 Oracle 可傳輸資料表空間進行移轉](#)、[Amazon EFS 整合](#) 及 [Amazon S3 整合](#)。
- AWS Database Migration Service (AWS DMS) — 移轉資料庫而不會停機。如需詳細資訊 AWS DMS，請參閱「[什麼是](#)」AWS Database Migration Service 和部落格文章 [使用 AWS DMS 移轉 Oracle 資料庫，停機時間幾乎為零](#)。

Important

在您使用上述的遷移技術之前，我們建議您備份您的資料庫。匯入資料後，您可用建立快照的方式備份您的 RDS for Oracle 資料庫執行個體。之後，您可以還原快照。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

對於許多資料庫引擎，持續複寫可以繼續到您準備好切換至目標資料庫為止。您可以使 AWS DMS 用從相同的資料庫引擎或不同的引擎移轉至適用於 Oracle 的 RDS。如果您從不同的資料庫引擎移轉，您可以使用移 AWS Schema Conversion Tool 轉 AWS DMS 未移轉的結構描述物件。

主題

- [使用 Oracle SQL Developer 匯入](#)
- [使用 Oracle 可傳輸資料表空間進行移轉](#)
- [使用 Oracle Data Pump 匯入](#)
- [使用 Oracle Export/Import 進行匯入](#)

- [使用 Oracle SQL*Loader 匯入](#)
- [使用 Oracle 具體化檢視進行遷移](#)

使用 Oracle SQL Developer 匯入

甲骨文 SQL 開發人員是甲骨文沒有成本分發的圖形 Java 工具。SQL Developer 提供選項讓您在兩個 Oracle 資料庫之間遷移資料，或從其他資料庫 (如 MySQL) 將資料遷移到 Oracle 資料庫。此工具最適合移轉小型資料庫。

您可以將此工具安裝在桌上型電腦 (Windows、Linux 或 Mac)，或安裝在您的其中一台伺服器上。安裝 SQL Developer 後，可用它連線至您的來源和目標資料庫。使用「工具」功能表上的「資料庫複製」命令，將資料複製到 RDS for Oracle 資料庫執行個體。

若要下載 SQL Developer，請前往 <http://www.oracle.com/technetwork/developer-tools/sql-developer>。

建議您先閱讀 Oracle SQL Developer 產品文件，再開始遷移資料。Oracle 也有文件說明如何從其他資料庫 (包括 MySQL 和 SQL Server) 進行遷移。如需詳細資訊，請參閱 Oracle 文件中的 <http://www.oracle.com/technetwork/database/migration>。

使用 Oracle 可傳輸資料表空間進行移轉

您可以使用 Oracle 可傳輸資料表空間功能將一組資料表空間從內部部署 Oracle 資料庫複製到 RDS for Oracle 資料庫執行個體。在實體層級，您可以使用 Amazon EFS 或 Amazon S3 將來源資料檔案和中繼資料檔案傳輸到目標資料庫執行個體。可傳輸表格空間功能會使用封裝。rdsadmin.rdsadmin_transport_util 如需此套件的語法和語意，請參閱 [可傳輸的資料表空間](#)。

如需說明如何傳輸表格空間的部落格文章，請參閱 [將 Oracle 資料庫遷移為 AWS 使用可傳輸表格空間](#) 和使用 RMAN 進行 [Oracle 可傳輸表格空間的 Amazon RDS](#)。

主題

- [Oracle 可傳輸資料表空間概觀](#)
- [階段 1：設定來源主機](#)
- [階段 2：準備完整的資料表空間備份](#)
- [階段 3：製作和傳輸增量備份](#)
- [階段 4：傳輸資料表空間](#)
- [階段 5：驗證所傳輸的資料表空間](#)

- [階段 6：清理剩餘檔案](#)

Oracle 可傳輸資料表空間概觀

可傳輸資料表空間集包含所要傳輸資料表空間集的资料檔案，以及含有資料表空間描述資料的匯出傾印檔案。在實體移轉解決方案 (例如可傳輸的資料表空間) 中，您可以傳輸實體檔案：資料檔案、組態檔案和 Data Pump 傾印檔案。

主題

- [可傳輸資料表空間的優缺點](#)
- [可傳輸資料表空間的限制](#)
- [可傳輸資料表空間的先決條件](#)

可傳輸資料表空間的優缺點

當您需要以最短停機時間將一或多個大型資料表空間移轉至 RDS 時，建議您使用可傳輸的資料表空間。相較於邏輯移轉，可傳輸資料表空間具有下列優點：

- 停機時間低於大多數其他 Oracle 移轉解決方案。
- 由於可傳輸資料表空間功能僅複製實體檔案，因此可避免邏輯移轉中可能發生的資料完整性錯誤和邏輯損毀。
- 不需要額外的授權。
- 您可以在不同平台和位元組類型之間移轉一組資料表空間，例如從 Oracle Solaris 平台移轉至 Linux。但不支援在 Windows Server 之間傳輸資料表空間。

Note

Linux 經過完整的測試和支援。並非所有的 UNIX 版本都經過測試。

如果您使用可傳輸的資料表空間，則可以使用 Amazon S3 或 Amazon EFS 來傳輸資料：

- 使用 EFS 時，您的備份會在匯入期間保留在 EFS 檔案系統中。您可以在之後刪除檔案。在此技術中，您無需為資料庫執行個體佈建 EBS 儲存體。因此，我們建議使用 Amazon EFS 而不是 S3。如需詳細資訊，請參閱 [Amazon EFS 整合](#)。
- 使用 S3 時，您可以將 RMAN 備份下載到資料庫執行個體所連接的 EBS 儲存體。在匯入期間，檔案會保留在 EBS 儲存體中。匯入之後，您可以釋放此空間，該空間仍會配置給您的資料庫執行個體。

可傳輸資料表空間的主要缺點，在於您需要具備關於 Oracle 資料庫相對進階的知識。如需詳細資訊，請參閱 Oracle 資料庫管理員指南中的 [在資料庫之間傳輸資料表空間](#)。

可傳輸資料表空間的限制

當您在 RDS for Oracle 中使用此功能時，可傳輸資料表空間的 Oracle 資料庫限制也同樣適用。如需詳細資訊，請參閱 Oracle 資料庫管理員指南中的 [可傳輸資料表空間的限制](#) 和 [傳輸資料的一般限制](#)。請注意，下列 RDS 適用於 Oracle 可傳輸資料表空間的額外限制：

- 來源或目標資料庫都不能使用標準版 2 (SE2)。僅支援企業版。
- 您不能使用 Oracle 數據庫 11g 數據庫作為源。RMAN 跨平台可傳輸表格空間功能依賴 Oracle 數據庫 11g 不支援的 RMAN 傳輸機制。
- 您無法使用可傳輸的資料表空間從 RDS for Oracle 數據庫執行個體移轉資料。可傳輸的資料表空間僅可用於將資料移轉至 RDS for Oracle DB 數據庫執行個體。
- 不支援 Windows 作業系統。
- 在較低版本層級，您無法將資料表空間傳輸到數據庫中。目標數據庫必須與來源數據庫位於相同或更新的版本層級。例如，您無法將資料表空間從 Oracle 數據庫 21c 傳輸到 Oracle 數據庫 19c。
- 您無法傳輸系統管理專用的資料表空間，例如 SYSTEM 和 SYSAUX。
- 您無法傳輸非資料物件，例如 PL/SQL 套件、Java 類別、檢視表、觸發程序、序列、使用者、角色和暫存資料表。若要傳輸非資料物件，請手動建立它們，或使用資料汲取中繼資料匯出和匯入。如需詳細資訊，請參閱 [我的甲骨文客戶 Support 注意事項 1454872.1](#)。
- 您無法傳輸已加密或使用加密資料欄的資料表空間。
- 如果您使用 Amazon S3 傳輸檔案，則支援的檔案大小上限為 5 TiB。
- 如果來源數據庫使用 Oracle 選項 (例如 Spatial)，則除非在目標數據庫上設定相同的選項，否則您無法傳輸資料表空間。
- 您無法將資料表空間傳輸到 Oracle 複本組態中的 RDS for Oracle 數據庫執行個體。解決方法是刪除所有複本、傳輸資料表空間，然後重新建立複本。

可傳輸資料表空間的先決條件

開始之前，請先完成以下任務：

- 請於 My Oracle Support 檢閱下列文件所述的可傳輸資料表空間需求：
 - [使用跨平台增量備份來減少可傳輸資料表空間停機時間 \(文件 ID 2471245.1\)](#)
 - [可傳輸資料表空間 \(TTS\) 的約束與限制：適用的詳細資訊、參考和版本 \(文件 ID 1454872.1\)](#)

- [可傳輸資料表空間 \(TTS\) 的主要備註 - 常見疑問與問題 \(文件 ID 1166564.1\)](#)
- 規劃端序轉換。如果您指定來源平台 ID，RDS for Oracle 會自動轉換端序。若要了解如何找到平台 ID，請參閱[相同 Data Guard 組態中異質主要和實體待命的 Data Guard 支援 \(文件 ID 413484.1\)](#)。
- 確定已在目標資料庫執行個體上啟用可傳輸資料表空間功能。僅在您執行下列查詢而未收到 ORA-20304 錯誤訊息時，才會啟用此功能：

```
SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
```

如果未啟用可傳輸資料表空間功能，請重新啟動資料庫執行個體。如需詳細資訊，請參閱[重新啟動中的資料庫執行個體](#)。

- 如果您規劃使用 Amazon S3 傳輸檔案，請執行以下操作：
 - 確保 Amazon S3 儲存貯體可用於檔案傳輸，並且 Amazon S3 儲存貯體與您的資料庫執行個體位於相同的 AWS 區域。如需指示說明，請參閱《Amazon Simple Storage Service 入門指南》中的[建立儲存貯體](#)。
 - 依照 [設定 IAM 許可權限以進行 RDS for Oracle 與 Amazon S3 的整合](#) 的指示來準備 Amazon S3 儲存貯體，進行 Amazon RDS 整合。
- 如果您規劃使用 Amazon EFS 傳輸檔案，請確定您已根據 [Amazon EFS 整合](#) 中的指示設定 EFS。
- 強烈建議您在目標資料庫執行個體中開啟自動備份。由於[中繼資料匯入步驟](#)可能失敗，因此請務必將資料庫執行個體還原至匯入前的狀態，以免需要重新備份、傳輸和匯入資料表空間。

階段 1：設定來源主機

在此步驟中，您會複製 My Oracle Support 提供的傳輸資料表空間指令碼，並設定必要的組態檔案。在下列步驟中，來源主機執行的資料庫含有預定傳輸至目標執行個體的資料表空間。

設定來源主機的方法如下

1. 以 Oracle 主目錄擁有者身分登入來源主機。
2. 請確定您的 ORACLE_HOME 和 ORACLE_SID 環境變數指向您的來源資料庫。
3. 以系統管理員身分登入資料庫，並確認時區版本、資料庫字元集和國家字元集與目標資料庫中的相同。

```
SELECT * FROM V$TIMEZONE_FILE;  
SELECT * FROM NLS_DATABASE_PARAMETERS  
WHERE PARAMETER IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');
```

4. 依照 [Oracle Support 備註 2471245.1](#) 的說明設定可傳輸資料表空間公用程式。

此設定包括編輯來源主機上的 `xtt.properties` 檔案。下列範例 `xtt.properties` 檔案會指定 `/dsk1/backups` 目錄中三個資料表空間的備份。這些是您規劃傳輸到目標資料庫執行個體的資料表空間。它還會指定來源平台 ID 以自動轉換端序。

Note

如需有效的平台 ID，請參閱[相同 Data Guard 組態中異質主要和實體待命的 Data Guard 支援 \(文件 ID 413484.1\)](#)。

```
#linux system
platformid=13
#list of tablespaces to transport
tablespaces=TBS1,TBS2,TBS3
#location where backup will be generated
src_scratch_location=/dsk1/backups
#RMAN command for performing backup
usermantransport=1
```

階段 2：準備完整的資料表空間備份

您會在此階段初次備份資料表空間、將備份傳輸到目標主機，然後使用 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` 程序還原這些備份。完成此階段時，初始資料表空間備份會存放在目標資料庫執行個體上，並且可以使用增量備份加以更新。

主題

- [步驟 1：備份來源主機上的資料表空間](#)
- [步驟 2：將備份檔案傳輸到目標資料庫執行個體](#)
- [步驟 3：在目標資料庫執行個體上匯入資料表空間](#)

步驟 1：備份來源主機上的資料表空間

在此步驟中，您會使用 `xttdriver.pl` 指令碼進行資料表空間的完整備份。`xttdriver.pl` 的輸出會儲存在 `TMPDIR` 環境變數中。

備份資料表空間的方法如下

- 如果您的資料表空間處於唯讀模式，請以具有 ALTER TABLESPACE 權限的使用者身分登入來源資料庫，然後將資料表空間設為讀取/寫入模式。否則，跳至下一步。

下列範例將 tbs1、tbs2 和 tbs3 設為讀取/寫入模式。

```
ALTER TABLESPACE tbs1 READ WRITE;
ALTER TABLESPACE tbs2 READ WRITE;
ALTER TABLESPACE tbs3 READ WRITE;
```

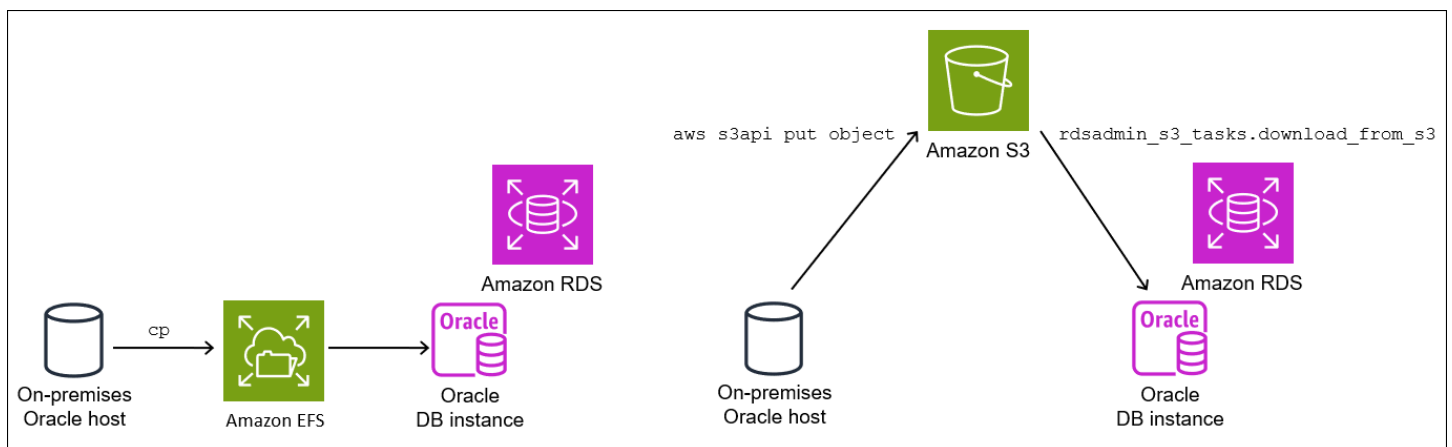
- 使用指令碼 xttdriver.pl 備份您的資料表空間。或者，您可以指定 --debug 在偵錯模式下執行指令碼。

```
export TMPDIR=location_of_log_files
cd location_of_xttdriver.pl
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
```

步驟 2：將備份檔案傳輸到目標資料庫執行個體

在此步驟中，請將備份和組態檔案從暫用位置複製到目標資料庫執行個體。請選擇下列其中一個選項：

- 如果來源主機和目標主機共用 Amazon EFS 檔案系統，請使用作業系統公用程式 (例如 cp) 將備份檔案和 res.txt 檔案從暫用位置複製到共用目錄。然後跳至 [步驟 3：在目標資料庫執行個體上匯入資料表空間](#)。
- 如需將備份暫存到 Amazon S3 儲存貯體，請完成以下步驟。



步驟 2.2：將傾印檔案上傳至 Amazon S3 儲存貯體

將備份和 `res.txt` 檔案從暫存目錄上傳到 Amazon S3 儲存貯體。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的 [上傳物件](#)。

步驟 2.3：將傾印檔案從 Amazon S3 儲存貯體下載到您的目標資料庫執行個體

在此步驟中，您可以使用 `rdsadmin.rdsadmin_s3_tasks.download_from_s3` 程序將備份下載到您的 RDS for Oracle 資料庫執行個體。

從 Amazon S3 儲存貯體下載備份的方法如下

1. 啟動 SQL*Plus 或 Oracle SQL Developer，並登入 RDS for Oracle 資料庫執行個體。
2. 使用 Amazon RDS 程序 `rdsadmin.rdsadmin_s3_tasks.download_from_s3` 將備份從 Amazon S3 儲存貯體下載到目標資料庫執行個體。以下範例會從名為 `DOC-EXAMPLE-BUCKET` 的 Amazon S3 儲存貯體下載所有檔案至 `DATA_PUMP_DIR` 目錄。

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'res.txt');
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name => 'DOC-EXAMPLE-BUCKET',
  p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

SELECT 陳述式以 VARCHAR2 資料類型傳回任務 ID。如需詳細資訊，請參閱 [將檔案從 Amazon S3 儲存貯體下載至 Oracle 資料庫執行個體](#)。

步驟 3：在目標資料庫執行個體上匯入資料表空間

若要將表格空間還原到目標資料庫執行個體，請使用程序 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`。此程序會自動將資料檔案轉換為正確的 endian 格式。

如果您從 Linux 以外的平台匯入，請在呼叫 `p_platform_id` 時使用參數指定來源平台 `import_xtts_tablespaces`。請確定您指定的平台 ID 與 `xtt.properties` 檔案中指定的平台 ID 相符 [步驟 2：匯出來源主機上的資料表空間描述資料](#)。

匯入目標資料庫執行個體上的資料表空間

1. 啟動 Oracle SQL 用戶端，並以主要使用者身分登入 RDS for Oracle 資料庫執行個體。

2. 執行程序 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`，指定要匯入的資料表空間及含有備份的目錄。

以下範例從目錄 `DATA_PUMP_DIR` 匯入資料表 `TBS1`、`TBS2` 和 `TBS3`。來源平台是以 AIX 為基礎的系統 (64 位元)，其平台識別碼為 6。您可以通過查詢找到平台 ID `V$TRANSPORTABLE_PLATFORM`。

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
    'TBS1,TBS2,TBS3',
    'DATA_PUMP_DIR',
    p_platform_id => 6);
END;
/

PRINT task_id
```

3. (選用) 透過查詢資料表 `rdsadmin.rds_xtts_operation_info` 來監控進度。 `xtts_operation_state` 資料欄會顯示值 `EXECUTING`、`COMPLETED` 或 `FAILED`。

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

Note

對於長時間執行的操作，您也可以查詢 `V$SESSION_LONGOPS`、`V$RMAN_STATUS`、和 `V$RMAN_OUTPUT`。

4. 使用上一個步驟的任務 ID 來檢視已完成匯入的記錄。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-||&task_id||'.log'));
```

請確定匯入已成功完成，再繼續下一個步驟。

階段 3：製作和傳輸增量備份

在這個階段中，您會在來源資料庫作用的同時定期建立和傳輸增量備份。此技術可減少最終資料表空間備份的大小。如果您進行多個增量備份，則必須在最後一次增量備份之後複製 `res.txt` 檔案，然後才能將其套用至目標執行個體。

這些步驟與 [階段 2：準備完整的資料表空間備份](#) 中的相同，不同之處在於匯入步驟具有選擇性。

階段 4：傳輸資料表空間

在此階段中，您會備份唯讀資料表空間並匯出 Data Pump 中繼資料、將這些檔案傳輸到目標主機，以及匯入資料表空間和中繼資料。

主題

- [步驟 1：備份唯讀資料表空間](#)
- [步驟 2：匯出來源主機上的資料表空間描述資料](#)
- [步驟 3：\(僅限 Amazon S3\) 將備份和匯出檔案傳輸到目標資料庫執行個體](#)
- [步驟 4：在目標資料庫執行個體上匯入資料表空間](#)
- [步驟 5：匯入目標資料庫執行個體上的資料表空間中繼資料](#)

步驟 1：備份唯讀資料表空間

此步驟與 [步驟 1：備份來源主機上的資料表空間](#) 相同，但有一項關鍵差異：在最後一次備份資料表空間之前，需將資料表空間設為唯讀模式。

下列範例將 `tbs1`、`tbs2` 和 `tbs3` 設為唯讀模式。

```
ALTER TABLESPACE tbs1 READ ONLY;  
ALTER TABLESPACE tbs2 READ ONLY;  
ALTER TABLESPACE tbs3 READ ONLY;
```

步驟 2：匯出來源主機上的資料表空間描述資料

在來源主機上執行 `expdp` 公用程式，以匯出資料表空間中繼資料。以下範例將資料表空間 `TBS1`、`TBS2` 及 `TBS3` 匯出至目錄 `DATA_PUMP_DIR` 中的傾印檔案 `xtdump.dmp`。

```
expdp username/pwd \  
dumpfile=xtdump.dmp \  

```



```
directory=DATA_PUMP_DIR \  
statistics=NONE \  
transport_tablespaces=TBS1,TBS2,TBS3 \  
transport_full_check=y \  
logfile=tts_export.log
```

如果 `DATA_PUMP_DIR` 是 Amazon EFS 中的共用目錄，請跳至。 [步驟 4：在目標資料庫執行個體上匯入資料表空間](#)

步驟 3：(僅限 Amazon S3) 將備份和匯出檔案傳輸到目標資料庫執行個體

如果您使用 Amazon S3 來暫存資料表空間備份和 Data Pump 匯出檔案，請完成以下步驟。

步驟 3.1：將備份和傾印檔案從來源主機上傳到 Amazon S3 儲存貯體

將備份和傾印檔案從來源主機上傳到 Amazon S3 儲存貯體 如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的 [上傳物件](#)。

步驟 3.2：將備份和傾印檔案從 Amazon S3 儲存貯體下載到您的目標資料庫執行個體

在此步驟中，您可以使用 `rdsadmin.rdsadmin_s3_tasks.download_from_s3` 程序將備份與傾印檔案下載到您的 RDS for Oracle 資料庫執行個體。請遵循 [步驟 2.3：將傾印檔案從 Amazon S3 儲存貯體下載到您的目標資料庫執行個體](#) 中的步驟。

步驟 4：在目標資料庫執行個體上匯入資料表空間

使用程序 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` 來還原資料表空間。如需此程序的語法和語意，請參閱 [將傳輸資料表空間匯入資料庫執行個體](#)

Important

完成最後的資料表空間匯入程序後，下一個步驟是 [匯入 Oracle Data Pump 中繼資料](#)。如果匯入失敗，請務必在資料庫執行個體回復失敗前的狀態。因此，我們建議您按照 [為單一可用區資料庫執行個體建立資料庫快照](#) 中的說明建立資料庫執行個體的資料庫快照。快照會包含所有匯入的資料表空間，因此如果匯入失敗，您不需要重複備份和匯入程序。

如果目標資料庫執行個體已開啟自動備份，且 Amazon RDS 在匯入中繼資料之前未偵測到任何有效快照已啟動，則 RDS 會嘗試建立快照。取決於您的執行個體活動，此快照可能但未必會成功。如果未偵測到有效的快照或無法啟動快照，則會結束中繼資料匯入程序，並顯示錯誤訊息。

匯入目標資料庫執行個體上的資料表空間

1. 啟動 Oracle SQL 用戶端，並以主要使用者身分登入 RDS for Oracle 資料庫執行個體。
2. 執行程序 `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`，指定要匯入的資料表空間及含有備份的目錄。

以下範例從目錄 `DATA_PUMP_DIR` 匯入資料表 `TBS1`、`TBS2` 和 `TBS3`。

```
BEGIN

  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces('TBS1,TBS2,TBS3','DATA_
END;
/
PRINT task_id
```

3. (選用) 透過查詢資料表 `rdsadmin.rds_xtts_operation_info` 來監控進度。 `xtts_operation_state` 資料欄會顯示值 `EXECUTING`、`COMPLETED` 或 `FAILED`。

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

Note

對於長時間執行的操作，您也可以查詢 `V$SESSION_LONGOPS`、`V$RMAN_STATUS`、和 `V$RMAN_OUTPUT`。

4. 使用上一個步驟的任務 ID 來檢視已完成匯入的記錄。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-||&task_id||.log'));
```

請確定匯入已成功完成，再繼續下一個步驟。

5. 請依照 [為單一可用區資料庫執行個體建立資料庫快照](#) 中的指示擷取手動資料庫快照。

步驟 5：匯入目標資料庫執行個體上的資料表空間中繼資料

在此步驟中，您可以使用 `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` 程序將可傳輸的資料表空間中繼資料匯入 RDS for Oracle 資料庫執行個體。如需此程序的語法

和語意，請參閱 [將可傳輸資料表空間中繼資料匯入您的資料庫執行個體](#)。在操作期間，資料表 `rdsadmin.rds_xtts_operation_info` 會顯示匯入狀態。

⚠ Important

強烈建議您在匯入資料表空間之後確認已成功建立資料庫快照，然後再匯入中繼資料。如果匯入步驟失敗，請還原資料庫執行個體、解決匯入錯誤，然後再次嘗試匯入。

將 Data Pump 中繼資料匯入 RDS for Oracle 資料庫執行個體。

1. 啟動 Oracle SQL 用戶端，並以主要使用者身分登入目標資料庫執行個體。
2. 如果這些使用者不存在，請在所傳輸資料表空間中建立擁有結構描述的使用者。

```
CREATE USER tbs_owner IDENTIFIED BY password;
```

3. 匯入中繼資料，指定傾印檔案的名稱及其目錄位置。

```
BEGIN  
  
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp', 'DATA_PUMP_DIR');  
END;  
/
```

4. (選用) 查詢可傳輸的資料表空間歷史記錄表，以查看中繼資料匯入的狀態。

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

操作完成時，您的資料表空間會處於唯讀模式。

5. (選用) 檢視日誌檔案。

下列範例列出 BDUMP 目錄的內容，然後查詢匯入記錄。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'BDUMP'));  
  
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file(  
  p_directory => 'BDUMP',  
  p_filename => 'rds-xtts-  
import_xtts_metadata-2023-05-22.01-52-35.560858000.log'));
```

階段 5：驗證所傳輸的資料表空間

在此選用步驟中，您可以使用程序 `rdsadmin.rdsadmin_rman_util.validate_tablespace` 來驗證所傳輸的資料表空間，然後將資料表空間設為讀取/寫入模式。

驗證所傳輸資料的方法如下

1. 動 SQL*Plus 或 SQL Developer，並以主要使用者身分登入目標資料庫執行個體。
2. 使用程序 `rdsadmin.rdsadmin_rman_util.validate_tablespace` 驗證資料表空間。

```
SET SERVEROUTPUT ON
BEGIN
  rdsadmin.rdsadmin_rman_util.validate_tablespace(
    p_tablespace_name      => 'TBS1',
    p_validation_type      => 'PHYSICAL+LOGICAL',
    p_rman_to_dbms_output => TRUE);
  rdsadmin.rdsadmin_rman_util.validate_tablespace(
    p_tablespace_name      => 'TBS2',
    p_validation_type      => 'PHYSICAL+LOGICAL',
    p_rman_to_dbms_output => TRUE);
  rdsadmin.rdsadmin_rman_util.validate_tablespace(
    p_tablespace_name      => 'TBS3',
    p_validation_type      => 'PHYSICAL+LOGICAL',
    p_rman_to_dbms_output => TRUE);
END;
/
```

3. 將資料表空間設唯讀取/寫入模式。

```
ALTER TABLESPACE TBS1 READ WRITE;
ALTER TABLESPACE TBS2 READ WRITE;
ALTER TABLESPACE TBS3 READ WRITE;
```

階段 6：清理剩餘檔案

在此可選步驟中，您會刪除所有不需要的檔案。使用

`rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` 程序列出在資料表空間匯入之後遺棄的資料檔，然後使用

`rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` 程序將其刪除。如需這些

程序的語法和語意，請參閱 [在資料表空間匯入後列出遺棄的檔案](#) 和 [在資料表空間匯入後刪除遺棄的資料檔案](#)。

清除剩餘檔案

1. 移除 `DATA_PUMP_DIR` 中的舊有備份，如下所示：

a. 執行 `rdsadmin.rdsadmin_file_util.listdir` 以列出備份檔案。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
'DATA_PUMP_DIR'));
```

b. 呼叫 `UTL_FILE.FREMOVE` 以逐一移除備份。

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'backup_filename');
```

2. 若已匯入資料表空間，但未匯入這些資料表空間的中繼資料，您可以依照下列方式刪除孤立的資料檔案：

a. 列出您需要刪除的孤立資料檔案。下列範例執行 `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`。

```
SQL> SELECT * FROM
TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME          FILESIZE
-----
datafile_7.dbf    104865792
datafile_8.dbf    104865792
```

b. 執行程序 `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` 以刪除孤立檔案。

```
BEGIN

rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

清除操作會產生日誌檔案，該檔案在 BDUMP 目錄中使用名稱格式 `rds-xtts-delete_xtts_orphaned_files-YYYY-MM-DD.HH24-MI-SS.FF.log`。

- c. 讀取上一個步驟中產生的日誌檔案。讀取日誌 `rds-xtts-delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log` 如下列範例所示：

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
    p_directory => 'BDUMP',
    p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
-----
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

3. 若已匯入這些資料表空間及其中繼資料，但發生相容性錯誤或其他 Oracle Data Pump 問題，請按以下方式清除部分傳輸的資料檔案：
 - a. 查詢 DBA_TABLESPACES 列出含有部分傳輸資料檔案的資料表空間。

```
SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE PLUGGED_IN='YES';

TABLESPACE_NAME
-----
TBS_3
```

- b. 刪除資料表空間和部分傳輸的資料檔案。

```
DROP TABLESPACE TBS_3 INCLUDING CONTENTS AND DATAFILES;
```

使用 Oracle Data Pump 匯入

Oracle Data Pump 是一個公用程式，可讓您將 Oracle 資料匯出至傾印檔案，並將其匯入至另一個 Oracle 資料庫。是 Oracle 匯出/匯入公用程式的長期替代方案。將大量資料從 Oracle 資料庫移動到 Amazon RDS 資料庫執行個體時，建議使用 Oracle Data Pump。

本節中的範例顯示將資料匯入 Oracle 資料庫的一種方法，但是 Oracle Data Pump 支援其他技術。如需詳細資訊，請參閱 [Oracle 資料庫文件](#)。

本節中的範例使用 DBMS_DATAPUMP 套件。您可以使用 Oracle Data Pump 命令列公用程式 impdp 和 expdp 來完成相同的工作。您可以在安裝 Oracle 用戶端 (包括 Oracle Instant Client) 期間，將這些公用程式安裝在遠端主機上。如需詳細資訊，請參閱《[如何使用 Oracle 用戶端為我的 Amazon RDS for Oracle 資料庫執行個體執行 Data Pump 來匯入或匯出？](#)》

主題

- [Oracle Data Pump 概觀](#)
- [以 Oracle Data Pump 及 Amazon S3 儲存貯體匯入資料](#)
- [以 Oracle Data Pump 及資料庫連結匯入資料](#)

Oracle Data Pump 概觀

Oracle Data Pump 由下列元件組成：

- 命令列用戶端 expdp 和 impdp
- DBMS_DATAPUMP PL/SQL 套件
- DBMS_METADATA PL/SQL 套件

Oracle Data Pump 可以用於以下案例：

- 將資料由 Oracle 資料庫 (內部部署或 Amazon EC2 執行個體) 匯入 RDS for Oracle 資料庫執行個體。
- 將資料從 RDS for Oracle 資料庫執行個體匯入 Oracle 資料庫 (內部部署或 Amazon EC2 執行個體)。
- 在 RDS for Oracle 資料庫執行個體之間匯入資料 (例如將資料從 EC2-Classical 遷移到 VPC)。

如需下載 Oracle Data Pump 公用程式，請參閱 Oracle Technology Network 網站的 [Oracle 資料庫軟體下載](#)。如需了解在 Oracle 資料庫各版本間遷移的相容性考量，請參閱 [Oracle 資料庫文件](#)。

Oracle Data Pump 工作流程

通常，您會在下列階段使用 Oracle Data Pump：

1. 將資料匯出至來源資料庫上的傾印檔案。

2. 將傾印檔案上傳至目的地 RDS for Oracle 資料庫執行個體。您可使用 Amazon S3 儲存貯體或兩個資料庫之間的資料庫連結進行傳輸。
3. 將資料從傾印檔案匯入 RDS for Oracle 資料庫執行個體。

Oracle Data Pump 最佳實務

當您使用 Oracle Data Pump 將資料匯入 RDS for Oracle 執行個體時，我們建議您採用下列最佳實務：

- schema 或 table 模式執行匯入，以匯入特定結構描述及物件。
- 匯入的結構描述僅限於應用程式需要的結構描述。
- 請勿匯入 full 模式或匯入系統維護元件的結構描述。

由於 RDS for Oracle 不允許存取 SYS 或 SYSDBA 管理使用者，這些動作可能會損害 Oracle 資料目錄，進而影響資料庫的穩定度。

- 當載入大量資料時，請執行下列作業：
 1. 將傾印檔案傳輸至目標 RDS for Oracle 資料庫執行個體。
 2. 拍攝您執行個體的資料庫快照。
 3. 測試匯入，以確認其是否成功。

如果資料庫元件失效，您可以刪除資料庫執行個體，並由資料庫快照重新建立。還原的資料庫執行個體，包括您在製作資料庫快照時，暫存於資料庫執行個體的任何傾印檔案。

- 請勿匯入使用 Oracle Data Pump 匯出參數 `TRANSPORT_TABLESPACES`、`TRANSPORTABLE` 或 `TRANSPORT_FULL_CHECK` 建立的傾印檔案。RDS for Oracle 資料庫執行個體不支援匯入這些傾印檔案。
- 請勿匯入包含 `SYS`、`SYSTEM`、`RDSADMIN`、`RDSSEC` 和 `RDS_DATAGUARD` 中之 Oracle Scheduler 物件的傾印檔案，且屬於下列類別：
 - 任務
 - 程式
 - Schedules
 - Chain
 - 規則
 - 評估內容
 - 規則集

RDS for Oracle 資料庫執行個體不支援匯入這些傾印檔案。

- 若要排除不支援的 Oracle Scheduler 物件，請在 Data Pump 匯出期間使用其他指令。如果您使用 DBMS_DATAPUMP，您可以在 DBMS_METADATA.START_JOB 之前新增一個額外的 METADATA_FILTER：

```
DBMS_DATAPUMP.METADATA_FILTER(
  v_hdn1,
  'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM SYS.OBJ$
        WHERE TYPE# IN (66,67,74,79,59,62,46)
        AND OWNER# IN
          (SELECT USER# FROM SYS.USER$
           WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC'))
        )
  ]',
  'PROCOBJ'
);
```

如果您使用 expdp，請建立包含 exclude 下列範例中所示之指令的參數檔案。然後透過 PARFILE=*parameter_file* 命令使用 expdp。

```
exclude=procoobj:"IN
(SELECT NAME FROM sys.OBJ$
 WHERE TYPE# IN (66,67,74,79,59,62,46)
 AND OWNER# IN
  (SELECT USER# FROM SYS.USER$
   WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC'))
 )"
)"
```

以 Oracle Data Pump 及 Amazon S3 儲存貯體匯入資料

以下匯入程序使用 Oracle Data Pump 及 Amazon S3 儲存貯體。步驟如下：

1. 使用 Oracle [DBMS_DATAPUMP](#) 套件匯出來源資料庫上的資料庫。
2. 將傾印檔案放入 Amazon S3 儲存貯體。

3. 從 Amazon S3 儲存貯體將傾印檔案下載至目標 Amazon RDS for Oracle 資料庫執行個體的 DATA_PUMP_DIR 目錄。
4. 使用套件 DBMS_DATAPUMP，將資料從複製的傾印檔案匯入 RDS for Oracle 資料庫執行個體。

主題

- [要求以 Oracle Data Pump 及 Amazon S3 儲存貯體匯入資料](#)
- [步驟 1：授予權限給 RDS for Oracle 目標資料庫執行個體上的資料庫使用者](#)
- [步驟 2：使用 DBMS_DAPUMP 將資料匯出倒傾印檔案](#)
- [步驟 3：將傾印檔案上傳至 Amazon S3 儲存貯體](#)
- [步驟 4：將傾印檔案從 Amazon S3 儲存貯體下載到您的目標資料庫執行個體](#)
- [步驟 5：使用 DBMS_DATAPUMP 將您的傾印檔案匯入目標資料庫執行個體](#)
- [步驟 6：清除](#)

要求以 Oracle Data Pump 及 Amazon S3 儲存貯體匯入資料

此程序有以下要求：

- 請確定 Amazon S3 儲存貯體可用於檔案傳輸，而且 Amazon S3 儲存貯體與資料庫執行個體位於 AWS 區域相同的位置。如需指示說明，請參閱《Amazon Simple Storage Service 入門指南》中的[建立儲存貯體](#)。
- 您上傳到 Amazon S3 儲存貯體的物件必須為 5 TB 或更少。如需在 Amazon S3 中使用物件的詳細資訊，請參閱《[Amazon Simple Storage Service 使用者指南](#)》。

Note

如果您傾印檔案超過 5 TB，您可以使用平行選項執行 Oracle Data Pump 匯出。這項作業會將資料分散到多個傾印檔案，讓您不會超過個別檔案的 5 TB 限制。

- 您必須遵循 [設定 IAM 許可權限以進行 RDS for Oracle 與 Amazon S3 的整合](#) 的指示內容，準備 Amazon S3 儲存貯體進行 Amazon RDS 整合。
- 您必須確定在來源執行個體和目前的資料庫執行個體上，有足夠的儲存空間可容納傾印檔案。

Note

此程序會將傾印檔案匯入 DATA_PUMP_DIR 目錄，這是在所有 Oracle 資料庫執行個體上預先設定好的目錄。此目錄和您的資料檔案位於相同的儲存磁碟區。您匯入傾印檔案時，現有 Oracle 資料檔案會使用更多空間。因此請確保資料庫執行個體能夠容納額外使用的空間。匯入的傾印檔案不會自動刪除或從 DATA_PUMP_DIR 目錄清除。如欲移除匯入的傾印檔案，請使用 Oracle 網站上的 [UTL_FILE.FREMOVE](#)。

步驟 1：授予權限給 RDS for Oracle 目標資料庫執行個體上的資料庫使用者

在此步驟中，您會建立要匯入資料的結構描述，並授與使用者必要的權限。

若要在 Oracle 目標執行個體的 RDS 上建立使用者並授與必要權限

1. 使用 SQL*Plus 或 Oracle SQL Developer 以主要使用者的身分登入，將 RDS for Oracle 資料庫執行個體連線至資料將匯入之處。如需連線至資料庫執行個體的詳細資訊，請參閱 [連線至 RDS for Oracle 資料庫執行個體](#)。
2. 匯入資料前，請先建立所需的資料表空間。如需詳細資訊，請參閱 [建立和調整資料表空間的大小](#)。
3. 如果接收匯入資料的使用者帳戶不存在，請建立使用者帳戶並授予所需的許可和角色。如果要將資料匯入多個使用者結構描述，請建立每個使用者帳戶並授予所需的權限和角色。

例如，以下 SQL 陳述式會建立新的使用者，並授予必需的許可和角色，以便將資料匯入該使用者擁有的結構描述。請將此步驟及後續步驟的 *schema_1* 替換為結構描述名稱。

```
CREATE USER schema_1 IDENTIFIED BY my_password;  
GRANT CREATE SESSION, RESOURCE TO schema_1;  
ALTER USER schema_1 QUOTA 100M ON users;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

上述的陳述式會授予新使用者 CREATE SESSION 權限和 RESOURCE 角色。您可能會需要其他權限和角色，視您匯入的資料庫物件而定。

步驟 2：使用 DBMS_DAPUMP 將資料匯出倒傾印檔案

若要建立傾印檔案，請使用 DBMS_DATAPUMP 套件。

將 Oracle 資料匯出到傾印檔案

1. 以管理使用者身分，使用 SQL Plus 或 Oracle SQL Developer 連線至來源 RDS for Oracle 資料庫執行個體。如果來源資料庫是 RDS for Oracle 資料庫執行個體，請以 Amazon RDS 主要使用者身分進行連線。
2. 透過呼叫 DBMS_DATAPUMP 程序匯出資料。

下列指令碼會匯出 *SCHEMA_1* 結構描述至 DATA_PUMP_DIR 目錄中名為 sample.dmp 的傾印檔案。以您要匯出之結構描述的名稱取代 *SCHEMA_1*。

```
DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle     => v_hdn1,
    filename   => 'sample.dmp',
    directory  => 'DATA_PUMP_DIR',
    filetype   => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle     => v_hdn1,
    filename   => 'sample_exp.log',
    directory  => 'DATA_PUMP_DIR',
    filetype   => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(v_hdn1, 'SCHEMA_EXPR', 'IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdn1,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM SYS.OBJ$
          WHERE TYPE# IN (66,67,74,79,59,62,46)
          AND OWNER# IN
            (SELECT USER# FROM SYS.USER$
             WHERE NAME IN ('RDSADMIN', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'RDSSEC'))]
```

```

        )
    )
]',
'PROCOBJ'
);
DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

Note

Data Pump 會以非同步方式開始工作。如需有關監控 Data Pump 任務的詳細資訊，請參閱 Oracle 文件中的[監控工作狀態](#)。

- (選用) 透過呼叫 `rdsadmin.rds_file_util.read_text_file` 程序檢視匯出日誌的內容。如需詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#)。

步驟 3：將傾印檔案上傳至 Amazon S3 儲存貯體

使用 Amazon RDS 程序 `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`，將傾印檔案複製到 Amazon S3 儲存貯體。以下範例將所有檔案從 `DATA_PUMP_DIR` 目錄上傳至名為 ***DOC-EXAMPLE-BUCKET*** 的 Amazon S3 儲存貯體。

```

SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name => 'DOC-EXAMPLE-BUCKET',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;

```

`SELECT` 陳述式以 `VARCHAR2` 資料類型傳回任務 ID。如需詳細資訊，請參閱 [將檔案從 RDS for Oracle 資料庫執行個體上傳至 Amazon S3 儲存貯體](#)。

步驟 4：將傾印檔案從 Amazon S3 儲存貯體下載到您的目標資料庫執行個體

使用 Amazon RDS 程序 `rdsadmin.rdsadmin_s3_tasks.download_from_s3` 執行此步驟。當您將文件下載到目錄時，如果目錄中已存在名稱相同的檔案，則程序 `download_from_s3` 會略過下載。如欲從下載目錄移除檔案，請使用 Oracle 網站上的 [UTL_FILE.FREMOVE](#)。

若要下載您的傾印檔案

1. 開始 SQL*Plus 或 Oracle SQL Developer，並以主要身分登入 Amazon RDS 目標 Oracle 資料庫執行個體
2. 使用 Amazon RDS 程序 `rdsadmin.rdsadmin_s3_tasks.download_from_s3` 下載傾印檔案。

以下範例會從名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體下載所有檔案至 `DATA_PUMP_DIR` 目錄。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    => 'DOC-EXAMPLE-BUCKET',
  p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

SELECT 陳述式以 VARCHAR2 資料類型傳回任務 ID。如需詳細資訊，請參閱 [將檔案從 Amazon S3 儲存貯體下載至 Oracle 資料庫執行個體](#)。

步驟 5：使用 DBMS_DATAPUMP 將您的傾印檔案匯入目標資料庫執行個體

使用 DBMS_DATAPUMP 將結構描述匯入 RDS for Oracle 資料庫執行個體。您可能需要額外選項，例如 METADATA_REMAP。

將資料匯入目標資料庫執行個體

1. 開始 SQL*Plus 或 SQL Developer，並以主要使用者身分登入 RDS for Oracle 資料庫執行個體。
2. 通過調用 DBMS_DATAPUMP 程序導入數據。

下列範例會從 `sample_copied.dmp` 將 *SCHEMA_1* 資料匯入目標資料庫執行個體。

```
DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdn1,
    filename  => 'sample_copied.dmp',
```

```
directory => 'DATA_PUMP_DIR',
filetype  => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.ADD_FILE(
  handle   => v_hdn1,
  filename => 'sample_imp.log',
  directory => 'DATA_PUMP_DIR',
  filetype => dbms_datapump.ku$_file_type_log_file);
DBMS_DATAPUMP.METADATA_FILTER(v_hdn1, 'SCHEMA_EXPR', 'IN (''SCHEMA_1'')');
DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/
```

Note

Data Pump 工作會以非同步的方式啟動。如需有關監控 Data Pump 任務的詳細資訊，請參閱 Oracle 文件中的[監控工作狀態](#)。您可以使用 `rdsadmin.rds_file_util.read_text_file` 程序來檢視匯入日誌的內容。如需詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#)。

3. 列出目標資料庫執行個體上的結構描述，以驗證資料匯入。

例如，以下查詢會傳回 *SCHEMA_1* 的資料表數目。

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

步驟 6：清除

匯入資料後，您可以刪除不繼續保留的檔案。

若要移除不需要的檔案

1. 開始 SQL*Plus 或 SQL Developer，並以主要使用者身分登入 RDS for Oracle 資料庫執行個體。
2. 使用以下命令列出 DATA_PUMP_DIR 中的檔案。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY
MTIME;
```

3. 使用以下命令刪除 DATA_PUMP_DIR 中不再需要的檔案。

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','filename');
```

例如以下命令會刪除名為 `sample_copied.dmp` 的檔案。

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

以 Oracle Data Pump 及資料庫連結匯入資料

以下匯入程序會使用 Oracle Data Pump 和 Oracle [DBMS_FILE_TRANSFER](#) 套件。步驟如下：

1. 連接至來源 Oracle 資料庫 (可以是內部部署資料庫、Amazon EC2 執行個體或 RDS for Oracle 資料庫執行個體)。
2. 使用 [DBMS_DATAPUMP](#) 套件匯出資料。
3. 使用 `DBMS_FILE_TRANSFER.PUT_FILE` 將傾印檔案從 Oracle 資料庫複製到目標 RDS for Oracle 資料庫執行個體的 `DATA_PUMP_DIR` 目錄 (資料庫執行個體使用資料庫連結與目標連線)。
4. 使用套件 `DBMS_DATAPUMP` 將資料從複製的傾印檔案匯入 RDS for Oracle 資料庫執行個體。

使用 Oracle Data Pump 及 `DBMS_FILE_TRANSFER` 套件的匯入程序具有下列步驟。

主題

- [匯入 Oracle Data Pump 及資料庫連結的要求](#)
- [步驟 1：授予權限給 RDS for Oracle 目標資料庫執行個體上的使用者](#)
- [步驟 2：向來源資料庫的使用者授予權限](#)
- [步驟 3：使用 DBMS_DATAPUMP 建立傾印檔案](#)
- [步驟 4：建立與目標資料庫執行個體的資料庫連結](#)
- [步驟 5：使用 DBMS_FILE_TRANSFER 將匯出的傾印檔案，複製到目標資料庫執行個體](#)
- [步驟 6：使用 DBMS_DATAPUMP 將資料檔案匯入至目標資料庫執行個體](#)
- [步驟 7：清理](#)

匯入 Oracle Data Pump 及資料庫連結的要求

此程序有以下要求：

- 您必須具有在 DBMS_FILE_TRANSFER 上和 DBMS_DATAPUMP 套件的執行權限。
- 您必須有寫入來源資料庫執行個體 DATA_PUMP_DIR 目錄的權限。
- 您必須確定在來源執行個體和目前的資料庫執行個體上，有足夠的儲存空間可容納傾印檔案。

Note

此程序會將傾印檔案匯入 DATA_PUMP_DIR 目錄，這是在所有 Oracle 資料庫執行個體上預先設定好的目錄。此目錄和您的資料檔案位於相同的儲存磁碟區。您匯入傾印檔案時，現有 Oracle 資料檔案會使用更多空間。因此請確保資料庫執行個體能夠容納額外使用的空間。匯入的傾印檔案不會自動刪除或從 DATA_PUMP_DIR 目錄清除。如欲移除匯入的傾印檔案，請使用 Oracle 網站上的 [UTL_FILE.FREMOVE](#)。

步驟 1：授予權限給 RDS for Oracle 目標資料庫執行個體上的使用者

如欲向 RDS for Oracle 目標資料庫執行個體的使用者授予權限，請進行下列步驟：

1. 使用 SQL Plus 或 Oracle SQL Developer，將 RDS for Oracle 資料庫執行個體連線至資料將匯入之處。以 Amazon RDS 主要使用者身分進行連線。如需連線至資料庫執行個體的詳細資訊，請參閱 [連線至 RDS for Oracle 資料庫執行個體](#)。
2. 匯入資料前，請先建立所需的資料表空間。如需更多詳細資訊，請參閱 [建立和調整資料表空間的大小](#)。
3. 如果接受匯入資料的使用者帳戶不存在，請建立使用者帳戶並授予所需的許可和角色。如果要將資料匯入多個使用者結構描述，請建立每個使用者帳戶並授予所需的權限和角色。


例如，以下命令會建立名為 *schema_1* 的新使用者，並授予必需的許可和角色，以便將資料匯入該使用者的結構描述。

```
CREATE USER schema_1 IDENTIFIED BY my-password;  
GRANT CREATE SESSION, RESOURCE TO schema_1;  
ALTER USER schema_1 QUOTA 100M ON users;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。


上述範例向新使用者授予 CREATE SESSION 權限及 RESOURCE 角色。您可能會需要其他權限和角色，視您匯入的資料庫物件而定。

 Note

請將此步驟及後續步驟的 *schema_1* 替換為結構描述名稱。

步驟 2：向來源資料庫的使用者授予權限


使用 SQL*Plus 或 Oracle SQL Developer，連線至包含要匯入之資料的 RDS for Oracle 資料庫執行個體。如有需要，建立使用者帳戶並授予必需的許可。

 Note

若來源資料庫是 Amazon RDS 執行個體，您可以略過此步驟。您會使用您的 Amazon RDS 主要使用者帳戶執行匯出。

以下命令會建立新的使用者並授予必需的許可。

```
CREATE USER export_user IDENTIFIED BY my-password;  
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;  
ALTER USER export_user QUOTA 100M ON users;  
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;  
GRANT SELECT_CATALOG_ROLE TO export_user;  
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;  
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

 Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

步驟 3：使用 DBMS_DATAPUMP 建立傾印檔案

若要建立傾印檔案，請執行下列步驟：

1. 以管理使用者或您在步驟 2 建立的使用者身分，使用 SQL*Plus 或 Oracle SQL Developer 連線至來源 Oracle 資料庫。如果來源資料庫是 Amazon RDS for Oracle 資料庫執行個體，請以 Amazon RDS 主要使用者身分進行連線。
2. 使用 Oracle Data Pump 公用程式建立傾印檔案。

以下指令碼會在 DATA_PUMP_DIR 目錄中建立名為 sample.dmp 的傾印檔案。

```

DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT' ,
    job_mode  => 'SCHEMA' ,
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle     => v_hdn1,
    filename  => 'sample.dmp' ,
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle     => v_hdn1 ,
    filename  => 'sample_exp.log' ,
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdn1 ,
    'SCHEMA_EXPR' ,
    'IN (''SCHEMA_1'')'
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdn1,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$
          WHERE TYPE# IN (66,67,74,79,59,62,46)
          AND OWNER# IN
            (SELECT USER# FROM SYS.USER$
             WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
          )
    ]',

```

```
'PROCOBJ'  
);  
DBMS_DATAPUMP.START_JOB(v_hdn1);  
END;  
/
```

Note

Data Pump 工作會以非同步的方式啟動。如需有關監控 Data Pump 任務的詳細資訊，請參閱 Oracle 文件中的[監控工作狀態](#)。您可以使用 `rdsadmin.rds_file_util.read_text_file` 程序檢視匯出日誌的內容。如需更多詳細資訊，請參閱[讀取資料庫執行個體目錄中的檔案](#)。

步驟 4：建立與目標資料庫執行個體的資料庫連結

在來源資料庫執行個體與目標資料庫執行個體之間建立資料庫連結。您的本機 Oracle 執行個體必須可以和資料庫執行個體網路連線，才能建立資料庫連結以及傳輸匯出傾印檔案。

請以前一步驟的相同使用者身分執行此步驟。

如果是在相同 VPC 或對等連接 VPC 內的兩個資料庫執行個體之間建立資料庫連結，兩個資料庫執行個體之間應具備有效路由。這兩個資料庫執行個體的安全群組，皆必須允許輸入和輸出到其他資料庫執行個體。安全群組的傳入和傳出規則可以參考相同 VPC 或對等連接 VPC 的安全群組。如需更多詳細資訊，請參閱[調整資料庫連結以搭配使用 VPC 中的資料庫執行個體](#)。

以下命令會建立名為 `to_rds` 的資料庫連結，可連接至目標資料庫執行個體的 Amazon RDS 主要使用者。

```
CREATE DATABASE LINK to_rds  
CONNECT TO <master_user_account> IDENTIFIED BY <password>  
USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)  
(PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

步驟 5：使用 DBMS_FILE_TRANSFER 將匯出的傾印檔案，複製到目標資料庫執行個體

使用 `DBMS_FILE_TRANSFER`，將傾印檔案從來源資料庫執行個體複製到目標資料庫執行個體。以下指令碼會將名為 `sample.dmp` 的傾印檔案，從來源執行個體複製到名為 `to_rds` 的目標資料庫連結 (在先前步驟中建立)。

```
BEGIN
```

```

DBMS_FILE_TRANSFER.PUT_FILE(
  source_directory_object      => 'DATA_PUMP_DIR',
  source_file_name             => 'sample.dmp',
  destination_directory_object => 'DATA_PUMP_DIR',
  destination_file_name        => 'sample_copied.dmp',
  destination_database         => 'to_rds' );
END;
/

```

步驟 6：使用 DBMS_DATAPUMP 將資料檔案匯入至目標資料庫執行個體

使用 Oracle Data Pump 匯入資料庫執行個體上的結構描述。您可能需要額外選項，例如 METADATA_REMAP。

使用 Amazon RDS 主要使用者帳戶連線到資料庫執行個體以進行匯入。

```

DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle      => v_hdn1,
    filename    => 'sample_copied.dmp',
    directory   => 'DATA_PUMP_DIR',
    filetype    => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle      => v_hdn1,
    filename    => 'sample_imp.log',
    directory   => 'DATA_PUMP_DIR',
    filetype    => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdn1, 'SCHEMA_EXPR', 'IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

Note

Data Pump 工作會以非同步的方式啟動。如需有關監控 Data Pump 任務的詳細資訊，請參閱 Oracle 文件中的 [監控工作狀態](#)。您可以使用

`rdsadmin.rds_file_util.read_text_file` 程序來檢視匯入日誌的內容。如需更多詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#)。

您可以檢視資料庫執行個體上的使用者資料表，以確認資料的匯入。例如，以下查詢會傳回 `schema_1` 的資料表數目。

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

步驟 7：清理

匯入資料後，您可以刪除不繼續保留的檔案。您可以使用以下命令列出 `DATA_PUMP_DIR` 中的檔案。

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

以下命令可用來刪除 `DATA_PUMP_DIR` 中不再需要的檔案。

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR', '<file name>');
```

例如以下命令會刪除名為 `"sample_copied.dmp"` 的檔案。

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR', 'sample_copied.dmp');
```

使用 Oracle Export/Import 進行匯入

在下列情況下，您可以考慮使用 Oracle Export/Import 公用程式來進行遷移：

- 您的資料很小。
- 不需要二進位浮點數和雙精度等資料類型。

匯入程序會建立必要的結構描述物件。因此不需要事先執行指令碼來建立物件。

安裝 Oracle 匯出和匯入公用程式的最簡單方法是安裝 Oracle Instant Client。若要下載該軟體，請前往 <https://www.oracle.com/database/technologies/instant-client.html>。如需說明文件，請參閱《Oracle 資料庫公用程式》手冊中的 [SQL*Loader 即時用戶端、匯出和匯入](#)。

匯出資料表然後匯入

1. 請使用 `exp` 命令從來源資料庫匯出資料表。

下列命令會匯出名為 tab1、tab2 及 tab3 的資料表。exp_file.dmp 是傾印檔案。

```
exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
```

匯出會建立一個二進位傾印檔案，其中包含指定資料表的結構描述和資料。

2. 使用 imp 命令將此結構描述和資料匯入目標資料庫。

下列命令會從傾印檔案 exp_file.dmp 匯入名為 tab1、tab2 及 tab3 的資料表。

```
imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \  
TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
```

匯出和匯入可能還有其版本更適合您的要求。請參閱 Oracle 資料庫文件以獲得完整的詳細資訊。

使用 Oracle SQL*Loader 匯入

您可以考慮針對有數量物件限制的大型資料庫使用 Oracle SQL*Loader。因為此程序會從來源資料庫匯出並載入到目標資料庫，所以與結構描述非常相關；以下範例會建立範例結構描述物件、從來源匯出、然後將資料載入到目標資料庫。

安裝 Oracle SQL*Loader 最簡單的方法是安裝 Oracle Instant Client。若要下載該軟體，請前往 <https://www.oracle.com/database/technologies/instant-client.html>。如需說明文件，請參閱《Oracle 資料庫公用程式》手冊中的 [SQL*Loader 即時用戶端、匯出和匯入](#)。

使用 Oracle SQL*Loader 匯入資料

1. 使用下列 SQL 陳述式建立樣本來源資料表。

```
CREATE TABLE customer_0 TABLESPACE users  
AS (SELECT ROWNUM id, o.*  
FROM ALL_OBJECTS o, ALL_OBJECTS x  
WHERE ROWNUM <= 1000000);
```

2. 在目標 RDS for Oracle 資料庫執行個體上建立目標資料表，用於載入資料。子句 WHERE 1=2 確保您複製 ALL_OBJECTS 的結構，但不複製任何資料列。

```
CREATE TABLE customer_1 TABLESPACE users  
AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
```

```
FROM ALL_OBJECTS
WHERE 1=2);
```

- 將資料從來源資料庫匯出成為文字檔。下列範例使用 SQL*Plus。針對您的資料，您可能需要建立指令碼來執行資料庫中所有物件的匯出。

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
FROM customer_0;

SPOOL OFF
```

- 建立控制檔案來描述資料。您可能需要撰寫指令碼來執行此步驟。

```
cat << EOF > sqlldr_1ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by '"'
(
  id          POSITION(01:10)    INTEGER EXTERNAL,
  owner       POSITION(12:41)    CHAR,
  object_name POSITION(43:72)    CHAR,
  created     POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

如有需要，請將先前程式碼產生的檔案複製到暫存區域，例如 Amazon EC2 執行個體中。

- 使用 SQL*Loader 和適當的目標資料庫使用者名稱與密碼來匯入資料。

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1ctl BINDSIZE=10485760 READSIZE=10485760
ROWS=1000
```

使用 Oracle 具體化檢視進行遷移

若要有效的遷移大型資料庫，您也可以使用 Oracle 的具體化檢視複寫功能。透過複寫，您可以使目標資料表與來源資料表維持同步。因此，您可以視需要之後切換為 Amazon RDS。

請確保您符合下列要求，才能使用具體化檢視進行遷移：

- 設定目標資料庫到來源資料庫之間的存取。在以下範例中，來源資料庫上所啟用的存取規則，會允許 RDS for Oracle 目標資料庫透過 SQL*Net 連線到來源。
- 建立從 RDS for Oracle 資料庫執行個體到來源資料庫之間的資料庫連結。

若要使用具體化檢視遷移資料

1. 在來源和 RDS for Oracle 目標執行個體上，建立可以使用相同密碼進行身分驗證的使用者帳戶。以下範例會建立名為 `dblink_user` 的使用者。

```
CREATE USER dblink_user IDENTIFIED BY my-password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

2. 使用您新建立的使用者，建立從 RDS for Oracle 目標執行個體到來源執行個體的資料庫連結。

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY my-password
  USING '(description=(address=(protocol=tcp) (host=my-host)
    (port=my-listener-port)) (connect_data=(sid=my-source-db-sid)))';
```


Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

3. 測試連結：

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. 在來源執行個體上，建立包含主要索引鍵和具體化檢視日誌的範例資料表。

```
CREATE TABLE customer_0 TABLESPACE users
AS (SELECT ROWNUM id, o.*
FROM ALL_OBJECTS o, ALL_OBJECTS x
WHERE ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;

CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. 在目標 RDS for Oracle 資料庫執行個體上建立具體化檢視。

```
CREATE MATERIALIZED VIEW customer_0
BUILD IMMEDIATE REFRESH FAST
AS (SELECT *
FROM cust_dba.customer_0@remote_site);
```

6. 在目標 RDS for Oracle 資料庫執行個體上重新整理具體化檢視。

```
EXEC DBMS_MV.REFRESH('CUSTOMER_0', 'f');
```

7. 捨棄具體化檢視，納入 PRESERVE TABLE 子句，以保留具體化檢視容器資料表及其內容。

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

前述保留資料表的名稱，與捨棄的具體化檢視相同。

使用適用於 Amazon RDS for Oracle 的僅供讀取複本

若要設定 Oracle 資料庫執行個體之間的複寫，您可以建立複本資料庫。如需 Amazon RDS 僅供讀取複本的概觀，請參閱 [Amazon RDS 僅供讀取複本的概觀](#)。如需 Oracle 複本和其他資料庫引擎之間差異的摘要，請參閱 [資料庫引擎的僅供讀取複本之間的差異](#)。

主題

- [RDS for Oracle 複本的概觀](#)
- [RDS for Oracle 複本的要求與考量](#)
- [準備建立 Oracle 複本](#)
- [在掛載模式中建立 RDS for Oracle 複本](#)
- [修改 RDS for Oracle 複本模式](#)
- [使用 RDS for Oracle 複本備份](#)
- [執行 Oracle Data Guard 容錯移轉](#)
- [對 RDS for Oracle 複本進行故障診斷](#)

RDS for Oracle 複本的概觀

Oracle 複本資料庫是主要資料庫的實體複本。唯讀模式中的 Oracle 複本稱為僅供讀取複本。掛載模式中的 Oracle 複本稱為掛載複本。Oracle 資料庫不允許在複本中寫入，但是您可以升級複本以使其可寫入。提升的僅供讀取複本會將複製的資料置於提升請求提出的時機。

以下影片提供 RDS for Oracle 災難復原的實用概觀。

如需詳細資訊，請參閱部落格文章[使用 Amazon RDS for Oracle 跨區域自動備份的受管災難復原第 1 部分](#)和[使用 Amazon RDS for Oracle 跨區域自動備份的受管災難復原第 2 部分](#)。

主題

- [唯讀和掛載複本](#)
- [CDB 的僅供讀取複本](#)
- [封存重做日誌保留](#)
- [Oracle 複製期間的中斷](#)

唯讀和掛載複本

建立或修改 Oracle 複本時，您可以將其置於下列任一模式中：

唯讀

此為預設值。Active Data Guard 會將來源資料庫的變更傳輸並套用至所有僅供讀取複本資料庫。

單一來源資料庫執行個體可建立至多 5 個僅供讀取複本。如需套用至所有資料庫引擎之僅供讀取複本的一般資訊，請參閱[使用資料庫執行個體僅供讀取複本](#)。如需 Oracle Data Guard 的詳細資訊，請參閱 Oracle 文件中的 [Oracle Data Guard concepts and administration](#)。

掛載

在此情況下，複寫會使用 Oracle Data Guard，但複本資料庫不接受使用者連線。掛載複本的主要用途是跨區域災難復原。

掛載複本無法提供唯讀工作負載。無論封存日誌保留政策為何，掛載複本會在套用封存重做日誌檔後刪除它們。

您可以為相同的來源資料庫執行個體建立掛載和唯讀資料庫複本的組合。您可以將唯讀複本變更為掛載模式，或將掛載複本變更為唯讀模式。無論是哪一種情況，Oracle 資料庫都會保留封存日誌的保留設定。

CDB 的僅供讀取複本

RDS for Oracle 僅在單一租戶組態中支援 Oracle Database 19c 和 21c CDB 的 Data Guard 僅供讀取複本。您可以在 CDB 中建立、管理和升級僅供讀取複本，就像在非 CDB 中一樣。也支援掛載的複本。您可以獲得以下好處：

- 受管的災難復原、高可用性，以及對您複本的僅供讀取存取
- 建立不同的僅供讀取複本的 AWS 區域能力。
- 與現有的 RDS 僅供讀取複本 API 整合：[建立資料庫 InstanceReadReplica](#)、[PromoteReadReplica](#) 和 [SwitchoverReadReplica](#)

若要使用此功能，您的複本和主要資料庫執行個體，都需要有效的 Data Guard 授權和 Oracle 資料庫企業版授權。不會產生使用 CDB 架構相關的額外費用。您只需為資料庫執行個體付費。

如需 CDB 架構之單一租戶和多租戶組態的詳細資訊，請參閱 [RDS for Oracle CDB 概觀](#)。

封存重做日誌保留

如果主要資料庫執行個體沒有跨區域僅供讀取複本，Amazon RDS for Oracle 在來源資料庫執行個體上會保留至少兩小時的封存重做日誌。無論 `rdsadmin.rdsadmin_util.set_configuration` 中 `archive_log retention hours` 的設定為何，都會成立。

兩小時之後，或過了封存日誌保留時數設定後 (視何者較長而定)，RDS 會從來源資料庫執行個體中清除日誌。只有在日誌已成功套用至資料庫後，RDS 才會在過了封存日誌保留時數設定之後從僅供讀取複本中清除日誌。

在某些情況下，一個主要資料庫執行個體可能有一或多個跨區域僅供讀取複本。如果是這樣，Amazon RDS for Oracle 會將交易日誌保留在來源資料庫執行個體上，直到傳輸並套用至所有跨區域僅供讀取複本為止。如需 `rdsadmin.rdsadmin_util.set_configuration` 的詳細資訊，請參閱[保留封存的日誌](#)。

Oracle 複製期間的中斷

建立僅供讀取複本時，Amazon RDS 會對來源資料庫執行個體建立資料庫快照，接著開始複寫。資料庫快照作業開始時，來源資料庫執行個體會遇到非常短暫的 I/O 暫停。I/O 懸吊通常持續約一秒鐘。假如來源資料庫執行個體為異地同步備份部署，則可避免 I/O 暫停，因為此時快照是從次要資料庫執行個體建立。

資料庫快照會變成 Oracle 複本。Amazon RDS 為來源資料庫和複本設定必要的參數和許可，而不會中斷服務。同樣地，如果您刪除複本，也不會發生中斷。

RDS for Oracle 複本的要求與考量

建立 Oracle 複本之前，請先深入了解下列要求與注意事項。

主題

- [RDS for Oracle 複本的版本和授權需求](#)
- [適用於 Oracle 複本之 RDS 的選項群組限制](#)
- [RDS for Oracle 複本的備份與還原考量](#)
- [RDS for Oracle 複本 Oracle 資料保護的需求和限制](#)
- [RDS for Oracle 複本的其他事項需求](#)

RDS for Oracle 複本的版本和授權需求

建立 RDS for Oracle 複本之前，請先考量下列項目：

- 如果複本處於唯讀模式，請確定您擁有 Active Data Guard 授權。如果您將複本置於掛載模式，則不需要 Active Data Guard 授權。只有 Oracle 資料庫引擎支援掛載的複本。
- Oracle 複本僅支援 Oracle 企業版 (EE)。
- 非 CDB 的 Oracle 複本僅支援使用執行 Oracle 資料庫 19c 的非 CDB 執行個體所建立的資料庫執行個體。
- Oracle 複本只適用於資料庫執行個體執行於資料庫執行個體類別具有兩個或以上的虛擬 vCPU。來源資料庫執行個體無法使用 db.t3 小型執行個體類別。
- 來源資料庫執行個體及其所有複本的 Oracle 資料庫引擎版本必須相同。Amazon RDS 升級來源資料庫執行個體後會立即升級複本，無論複本的維護時段為何。對於跨區域複本的主要版本升級，Amazon RDS 會自動執行下列動作：
 - 產生目標版本的選項群組。
 - 將原始選項群組中的所有選項和選項設定複製到新選項群組。
 - 將升級的跨區域複本與新選項群組建立關聯。

如需有關升級資料庫引擎版本的詳細資訊，請參閱[升級 RDS for Oracle 資料庫引擎](#)。

適用於 Oracle 複本之 RDS 的選項群組限制

建立 RDS for Oracle 複本之前，請先考量下列項目：

- 如果 Oracle 複本與其來源資料庫執行個體位於相同的 AWS 區域，則複本無法使用來源資料庫執行個體之外的不同選項群組。對來源選項群組或來源選項群組成員資格的修改會傳輸至 Oracle 複本。套用至來源資料庫執行個體後變更會立即套用至複本，不論複本是否在維護視窗。

如需選項群組的詳細資訊，請參閱[使用選項群組](#)。

- 您無法從為複本自動建立的專用選項群組中移除適用於 Oracle 跨區域複本的 RDS。
- 您無法將 RDS for Oracle 跨區域複本的專用選項群組新增至不同的資料庫執行個體。
- 您只能從 RDS for Oracle 跨區域複本的專用選項群組新增或移除下列非複製選項：
 - NATIVE_NETWORK_ENCRYPTION
 - OEM
 - OEM_AGENT
 - SSL

若要將其他選項新增至 RDS for Oracle 跨區域複本，請將選項新增至來源資料庫執行個體的選項群組。選項也安裝在所有來源資料庫執行個體的複本上。對於授權選項，請確定複本有足夠的授權。

當您提升 RDS for Oracle 跨區域複本時，提升的複本就像其他 Oracle 資料庫執行個體一樣運作，包括管理其選項。您可以刪除來源資料庫執行個體，以明確或隱含提升複本。

如需選項群組的詳細資訊，請參閱[使用選項群組](#)。

- 適用於 Oracle 跨區域複本的 RDS 不支援 EFS_INTEGRATION 此選項。

RDS for Oracle 複本的備份與還原考量

建立 RDS for Oracle 複本之前，請先考量下列項目：

- 若要建立 RDS for Oracle 複本的快照或開啟自動備份，請務必手動設定備份保留期間。根據預設，不會啟用自動備份。
- 當您還原複本備份時，會還原到資料庫時間，而不是開始備份的時間。資料庫時間指備份中資料最近套用的交易時間。其中的差異非常明顯，因為複本的時間可能會延後幾分鐘或幾小時。

若要了解差異，請使用 `describe-db-snapshots` 命令。比較 `snapshotDatabaseTime` (也就是複本備份的資料庫時間) 及 `OriginalSnapshotCreateTime` 欄位 (也就是主要資料庫上最新套用的交易時間)。

RDS for Oracle 複本 Oracle 資料保護的需求和限制

建立 RDS for Oracle 複本之前，請注意下列需求和限制：

- 如果您的主要資料庫執行個體使用多租戶架構的單一租戶組態，請考慮下列事項：
 - 您必須使用 Oracle 資料庫 19c 或更新版本與企業版搭配使用。
 - 您的主要 CDB 執行個體必須位於 ACTIVE 生命週期。
 - 您無法將非 CDB 主執行個體轉換為 CDB 執行個體，並在相同的作業中轉換其複本。相反，請刪除非 CDB 複本，將主要資料庫執行個體轉換為 CDB，然後建立新複本
- 確保主要執行個體上的登入觸發程序必須允許 RDS_DATAGUARD 使用者以及其 AUTHENTICATED_IDENTITY 值為 RDS_DATAGUARD 或 rdsdb 之任何使用者的存取權。此外，觸發程序不得設定 RDS_DATAGUARD 使用者的目前結構描述。
- 若要避免封鎖 Data Guard 中介處理作業的連線，請勿啟用受限制的階段作業。如需限制工作階段的詳細資訊，請參閱[啟用和停用受限制的工作階段](#)。

RDS for Oracle 複本的其他事項需求

建立 RDS for Oracle 複本之前，請先考量下列項目：

- 如果您的資料庫執行個體是一或多個跨區域複本的來源，則來源資料庫會保留其存檔重做日誌檔，直到將它們套用至所有跨區域複本為止。封存的重做日誌可能導致耗用更多儲存空間。
- 若要避免中斷 RDS 自動化，系統觸發程序必須允許特定使用者登入主要和複本資料庫。[系統觸發程序](#)包括 DDL、登入和資料庫角色觸發程序。我們建議您將程式碼新增至您的觸發程序，以排除下列範本程式碼中所列的使用者：

```
-- Determine who the user is
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or
Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- 唯讀複本支援區塊變更追蹤，但不支援掛載複本。您可以將掛載複本變更為唯讀複本，然後啟用區塊變更追蹤。如需更多詳細資訊，請參閱 [啟用和停用區塊變更追蹤功能](#)。

準備建立 Oracle 複本

在開始使用複本之前，請先執行下列工作。

主題

- [啟用自動備份](#)
- [啟用強制記錄模式](#)
- [變更您的記錄組態](#)
- [設定 MAX_STRING_SIZE 參數](#)
- [規劃運算和儲存資源](#)

啟用自動備份

當您將資料庫執行個體作為來源資料庫執行個體之前，請務必在來源資料庫執行個體上啟用自動備份。若要了解如何執行此程序，請參閱[啟用自動備份](#)。

啟用強制記錄模式

建議您啟用強制記錄模式。在強制記錄模式中，即使 NOLOGGING 與資料定義語言 (DDL) 陳述式搭配使用，Oracle 資料庫仍會寫入重做記錄。

啟用強制記錄模式

1. 使用用戶端工具 (例如 SQL Developer) 登入 Oracle 資料庫。
2. 執行下列程序，以啟用強制記錄模式。

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

如需此程序的詳細資訊，請參閱 [設定強制記錄](#)。

變更您的記錄組態

對於大小為 m 的 n 個線上重做日誌，RDS 會在主要資料庫執行個體和所有複本上自動建立大小為 m 的 $n+1$ 個待命日誌。每當您變更主要伺服器上的記錄組態時，變更都會自動傳播至複本。

如果您變更記錄組態，請考慮下列準則：

- 建議您先完成變更，然後再將資料庫執行個體設為複本的來源。RDS for Oracle 也支援在執行個體成為來源之後更新執行個體。
- 在您變更主要資料庫執行個體的記錄組態之前，請檢查每個複本是否有足夠的儲存空間來容納新的組態。

您可以使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.add_logfile` 和修改資料庫執行個體的記錄組態 `rdsadmin.rdsadmin_util.drop_logfile`。如需詳細資訊，請參閱 [新增線上重做日誌](#) 及 [捨棄線上重做日誌](#)。

設定 MAX_STRING_SIZE 參數

在您建立 Oracle 複本之前，請確保 `MAX_STRING_SIZE` 參數的設定與來源資料庫執行個體和複本上的設定相同。您可以使用相同的參數群組建立關聯以達到此目的。如果來源和複本有不同的參數群組，您可以將 `MAX_STRING_SIZE` 設定為相同的值。如需此參數的詳細資訊，請參閱 [針對新的資料庫執行個體開啟擴充資料類型](#)。

規劃運算和儲存資源

確保資料庫執行個體及複本都依照運算和儲存適當的調整大小，以搭配工作負載。如複本達到運算、網路或儲存資源上限時，複本會停止接受或套用來源的變更。Amazon RDS for Oracle 的來源資料庫執行個體與其複本之間不會相互干涉，以降低複寫時的高延遲。您可以從來源和其他複本獨立修改複本的儲存和 CPU 資源。

在掛載模式中建立 RDS for Oracle 複本

依預設，Oracle 複本是唯讀的。若要在掛載模式中建立複本，請使用主控台、AWS CLI 或 RDS API。

主控台

從來源 Oracle 資料庫執行個體建立掛載的複本

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇要做為掛載複本之來源的 Oracle 資料庫執行個體。
4. 在 Actions (動作) 中，選擇 Create replica (建立複本)。
5. 對於 Replica mode (複本模式)，選擇 Mounted (掛載)。
6. 選擇您想要使用的設定。在 DB instance identifier (資料庫執行個體識別符) 中，輸入僅供讀取複本的名稱。視需要調整其他設定。
7. 對於 Regions (區域)，選擇要啟動掛載複本的區域。
8. 選擇您的執行個體大小和儲存體類型。建議您使用與僅供讀取複本來源資料庫執行個體相同的資料庫執行個體類別和儲存類型。
9. 在 Multi-AZ deployment (異地同步備份部署) 中，選擇 Create a standby instance (建立待命執行個體)，以在另一個可用區域中，建立複本的待命複本，以便支援掛載的複本。建立您的掛載複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。
10. 選擇您想要使用的其他設定。
11. 選擇 Create replica (建立複本)。

在 Databases (資料庫) 頁面中，掛載的複本具有角色「複本」。

AWS CLI

若要在掛接模式下建立 Oracle 複本，請在AWS CLI指令 [create-db-instance-read-replica mounted](#) 中設定 `--replica-mode` 為。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier myreadreplica \  
  --source-db-instance-identifier mydbinstance \  
  --replica-mode mounted
```

在Windows中：

```
aws rds create-db-instance-read-replica ^  
  --db-instance-identifier myreadreplica ^  
  --source-db-instance-identifier mydbinstance ^  
  --replica-mode mounted
```

若要將唯讀複本變更為裝載狀態，請在AWS CLI指令 `mounted` 中設定 `--replica-mode` 為 [modify-db-instance](#)。若要將掛載複本置於唯讀模式，請將 `--replica-mode` 設定為 `open-read-only`。

RDS API

若要以掛接模式建立 Oracle 複本，請 `ReplicaMode=mounted` 在 RDS API 作業 [建立 InstanceReadReplica](#) 資料庫中指定。

修改 RDS for Oracle 複本模式

若要變更現有複本的複本模式，請使用主控台、AWS CLI 或 RDS API。當您變更為掛載模式時，複本會中斷所有作用中的連線。當您變更為唯讀模式時，Amazon RDS 會初始化 Active Data Guard。

變更作業可能需要幾分鐘的時間。在作業期間，資料庫執行個體狀態會變更為 `modifying` (正在修改)。如需狀態變更的詳細資訊，請參閱 [檢視 Aurora 叢集中 資料庫執行個體狀態](#)。

主控台

將 Oracle 複本的複本模式從掛載變更為唯讀

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。

2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇掛載的複本資料庫。
4. 選擇 Modify (修改)。
5. 對於 Replica mode (複本模式)，選擇 Read-only (唯讀)。
6. 選擇您想要變更的其他設定。
7. 選擇 Continue (繼續)。
8. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。
9. 選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要將僅供讀取複本變更為掛接模式，請在AWS CLI指令mounted中設定--replica-mode為[modify-db-instance](#)。若要將掛載複本變更為唯讀模式，請將--replica-mode 設定為 open-read-only。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier myreadreplica \  
  --replica-mode mode
```

在Windows中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier myreadreplica ^  
  --replica-mode mode
```

RDS API

若要將唯讀複本變更為掛載模式，請在 [ModifyDBInstance](#) 中設定 ReplicaMode=mounted。若要將掛載複本變更為唯讀模式，請設定 ReplicaMode=read-only。

使用 RDS for Oracle 複本備份

您可以建立和還原 RDS for Oracle 複本的備份。同時支援自動備份和手動快照。如需詳細資訊，請參閱[備份、還原和匯出資料](#)。下列部分說明管理主要複本備份和 RDS for Oracle 複本之間的主要差異。

開啟 RDS for Oracle 複本備份

預設情況下，Oracle 複本不會開啟自動備份功能。您可以將備份保留期間設定為正的非零值以開啟自動備份。

主控台

立即啟用自動備份

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中選擇資料庫，然後選擇您要修改的資料庫執行個體或多可用區域資料庫叢集。
3. 選擇 Modify (修改)。
4. 針對 Backup retention period (備份保留期)，選擇非零正值，例如 3 天。
5. 選擇 Continue (繼續)。
6. 選擇 Apply immediately (立即套用)。
7. 選擇修改資料庫執行個體或修改叢集，以儲存您的變更並啟用自動備份。

AWS CLI

若要啟用自動備份，請使用 AWS CLI [modify-db-instance](#) 或 [modify-db-cluster](#) 命令。

包含以下參數：

- `--db-instance-identifier` (或多可用區域資料庫叢集的 `--db-cluster-identifier`)
- `--backup-retention-period`
- `--apply-immediately` 或 `--no-apply-immediately *`

在以下範例中，我們將備份保留期間設定為三天來啟用自動備份。變更會立即套用。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --backup-retention-period 3 \  
  --apply-immediately
```

```
--backup-retention-period 3 \  
--apply-immediately
```

在Windows中：

```
aws rds modify-db-instance ^  
--db-instance-identifier mydbinstance ^  
--backup-retention-period 3 ^  
--apply-immediately
```

RDS API

若要啟用自動備份，請使用 RDS API [ModifyDBInstance](#) 或 [ModifyDBCluster](#) 操作，其中包含下列必要參數：

- DBInstanceIdentifier 或 DBClusterIdentifier
- BackupRetentionPeriod

還原 RDS for Oracle 複本備份

您可以還原 Oracle 複本備份，就像還原主要執行個體的備份一樣。如需詳細資訊，請參閱下列內容：

- [從資料庫快照還原](#)
- [將資料庫執行個體還原至指定的時間](#)

還原複本備份時的主要考量是決定要還原的時間點。資料庫時間指備份中資料最近套用的交易時間。還原複本備份時，您會還原到資料庫時間，而不是備份完成的時間。其中的差異很大，因為 RDS for Oracle 複本的時間可能會延後幾分鐘或幾小時。因此，複本備份的資料庫時間，以及您還原備份的時間點，可能會比備份建立時間早得多。

若要了解資料庫時間與建立時間之間的差異，請使用 `describe-db-snapshots` 命令。比較 `SnapshotDatabaseTime` (也就是複本備份的資料庫時間) 及 `OriginalSnapshotCreateTime` 欄位 (也就是主要資料庫上最新套用的交易時間)。下列範例會傳回兩個時間之間的差異：

```
aws rds describe-db-snapshots \  
--db-instance-identifier my-oracle-replica  
--db-snapshot-identifier my-replica-snapshot
```

```
{
  "DBSnapshots": [
    {
      "DBSnapshotIdentifier": "my-replica-snapshot",
      "DBInstanceIdentifier": "my-oracle-replica",
      "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
      ...
      "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
    }
  ]
}
```

執行 Oracle Data Guard 容錯移轉

容錯移轉是主要資料庫與待命資料庫之間的角色反轉。在容錯移轉期間，原始主要資料庫會轉換為待命角色，而原始待命資料庫則會轉換為主要角色。

在「Oracle Data Guard」環境中，主要資料庫支援一或多個待命資料庫。您可以從主要資料庫執行以容錯移轉為基礎的受管理角色轉換至待命資料庫。容錯移轉是主要資料庫與待命資料庫之間的角色反轉。在容錯移轉期間，原始主要資料庫會轉換為待命角色，而原始待命資料庫則會轉換為主要角色。

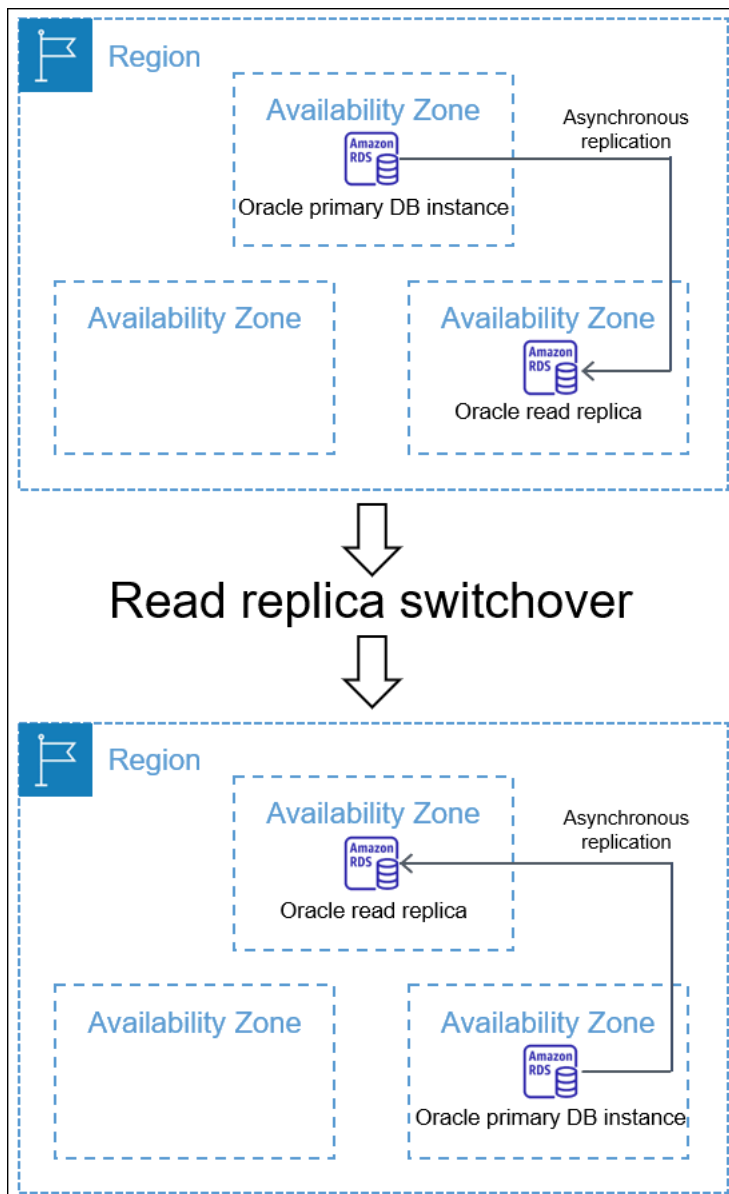
主題

- [Oracle Data Guard 容錯移轉概觀](#)
- [準備進行 Oracle Data Guard 容錯移轉](#)
- [啟動 Oracle Data Guard 容錯移轉](#)
- [監控 Oracle Data Guard 容錯移轉](#)

Oracle Data Guard 容錯移轉概觀

Amazon RDS 支援 Oracle 資料庫複本的全受管、以容錯移轉為基礎的角色轉換。您只能啟動容錯移轉為已裝載或以唯讀方式開啟的待命資料庫。

複本可以位於單一區域的個別 AWS 區域 或不同可用區域 (AZ) 中。所有 AWS 區域 的支持。



轉換不同於僅供讀取複本升級。在轉換過程中，來源和複本資料庫執行個體會變更角色。在促銷中，僅供讀取複本會變成來源資料庫執行個體，但來源資料庫執行個體不會成為複本。如需詳細資訊，請參閱[提升僅供讀取複本為獨立的資料庫執行個體](#)。

主題

- [Oracle Data Guard 容錯移轉的優點](#)
- [支援的 Oracle Database 版本](#)
- [Oracle Data Guard 容錯移轉的成本](#)
- [Oracle Data Guard 容錯移轉運作方式](#)

Oracle Data Guard 容錯移轉的優點

如同 RDS for Oracle 僅供讀取複本一樣，受管容錯移轉依賴 Oracle Data Guard。該操作被設計為零資料丟失。Amazon RDS 會將以下的容錯移轉方面自動化：

- 反轉主要資料庫和指定待命資料庫的角色，讓新待命資料庫處於與原始待命資料庫相同的狀態 (已裝載或唯讀)
- 確保資料一致性
- 轉換後維護您的複寫組態
- 支援重複回轉，允許新的待命資料庫恢復到原始的主要角色

支援的 Oracle Database 版本

Oracle 資料庫 19c 支援「Oracle 資料保全」轉換。

Oracle Data Guard 容錯移轉的成本

Oracle Data Guard 容錯移轉功能不會產生額外費用。Oracle Database Enterprise Edition 包含支援處於掛載模式的待命資料庫。若要以唯讀模式開啟待命資料庫，您需要 Oracle Active Data Guard 選項。

Oracle Data Guard 容錯移轉運作方式

Oracle Data Guard 容錯移轉是完全受管理的操作。您可以發出 CLI 命令 `switchover-read-replica` 來啟動待命資料庫的容錯移轉。然後，Amazon RDS 會修改複寫組態中的主要角色和待命角色。

所以此原始待機和原始主要是容錯移轉之前存在的角色。所以此新待機和新主要是容錯移轉之後存在的角色。待命者複本是一個複本資料庫，在 Oracle Data Guide 環境中做為待命資料庫，但不會切換角色。

主題

- [Oracle Data Guard 容錯移轉的階段](#)
- [Oracle Data Guard 容錯移轉後](#)

Oracle Data Guard 容錯移轉的階段

若要執行容錯移轉，Amazon RDS 必須採取下列步驟：

1. 封鎖原始主要資料庫上的新交易。在容錯移轉期間，Amazon RDS 會中斷 Oracle Data Guide 組態中所有資料庫的複寫。在容錯移轉期間，原始主要資料庫無法處理寫入請求。

2. 將未套用的交易傳輸至原始待命資料庫，然後套用這些交易。
3. 以唯讀或裝載模式重新啟動新的待命資料庫。此模式取決於容錯移轉前原始待命資料庫的開啟狀態。
4. 以讀寫模式開啟新的主要資料庫。

Oracle Data Guard 容錯移轉後

Amazon RDS 會切換主要和待命資料庫的角色。您有責任重新連接應用程式並執行任何其他所需的組態。

主題

- [成功條件](#)
- [連線到新的主要資料庫](#)
- [新主要資料庫的組態](#)

成功條件

當原始待命資料庫執行下列操作時，Oracle Data Guide 容錯移轉成功：

- 轉換為新主要資料庫的角色
- 完成其重新組態

為了限制停機時間，您的新主要資料庫會盡快變為作用中狀態。由於 Amazon RDS 會以非同步方式設定待命者複本，因此這些複本可能會在原始主要資料庫之後變為作用中狀態。

連線到新的主要資料庫

容錯移轉後，Amazon RDS 不會將您目前的資料庫連線傳播到新的主要資料庫。Oracle Data Guard 容錯移轉完成後，請將您的應用程式重新連線至新的主要資料庫。

新主要資料庫的組態

若要執行容錯移轉為新的主要資料庫，Amazon RDS 會將原始待命資料庫的模式變更為開啟。角色的變化是對資料庫的唯一更改。Amazon RDS 不會設定多可用區域複寫等功能。

如果您使用不同的選項執行跨區域複本的容錯移轉，新的主要資料庫會保留自己的選項。Amazon RDS 不會遷移原始主要資料庫上的選項。如果原始主要資料庫具有 SSL、NNE、OEM 和 OEM_AGENT 等選項，則 Amazon RDS 不會將它們傳播到新的主要資料庫。

準備進行 Oracle Data Guard 容錯移轉

啟動 Oracle Data Guard 容錯移轉前，請確定您的複本環境符合下列要求：

- 原始待命資料庫已裝載或以唯讀方式開啟。
- 原始待命資料庫上已啟用自動備份。
- 原始主要資料庫和原始待命資料庫處於可用狀態。
- 原始主要資料庫和原始待命資料庫沒有擱置的維護動作。
- 原始待命資料庫處於複寫狀態。
- 當主要資料庫或待命資料庫目前處於轉換生命週期時，您不會嘗試啟動容錯移轉。如果複本資料庫在容錯移轉後重新設定，Amazon RDS 會阻止您啟動另一次容錯移轉。

Note

待命者複本是 Oracle Data Guard 組態中不是容錯移轉目標的複本。待命者複本在容錯移轉期間可處於任何狀態。

- 原始待命資料庫的組態與原始主要資料庫盡可能接近。假設原始主要和原始待命資料庫有不同選項的案例。容錯移轉完成後，Amazon RDS 不會自動重新設定新的主要資料庫，使其具有與原始主要資料庫相同的選項。
- 您可以在起始切換之前，先設定所需的多可用區部署。Amazon RDS 不會在切換過程中管理多可用區域。多可用區部署會保持原狀。

假設 db_maz 是多可用區部署中的主要資料庫，而 db_saz 是單一可用區域複本。您起始從 db_maz 到 db_saz 的切換。之後，db_maz 是多可用區域複本資料庫，而 db_saz 是單一可用區域主要資料庫。新的主要資料庫現在未受多可用區部署支援。

- 為了準備跨區域切換，主要資料庫不會與複寫組態之外的資料庫執行個體使用相同的選項群組。若要讓跨區域切換成功，目前的主要資料庫及其僅供讀取複本，必須是唯一可使用目前主要資料庫之選項群組的資料庫執行個體。否則，Amazon RDS 會阻止切換。

啟動 Oracle Data Guard 容錯移轉

您可以將 RDS for Oracle 僅供讀取複本容錯移轉為主要角色，將原主要資料庫執行個體容錯移轉為複本角色。

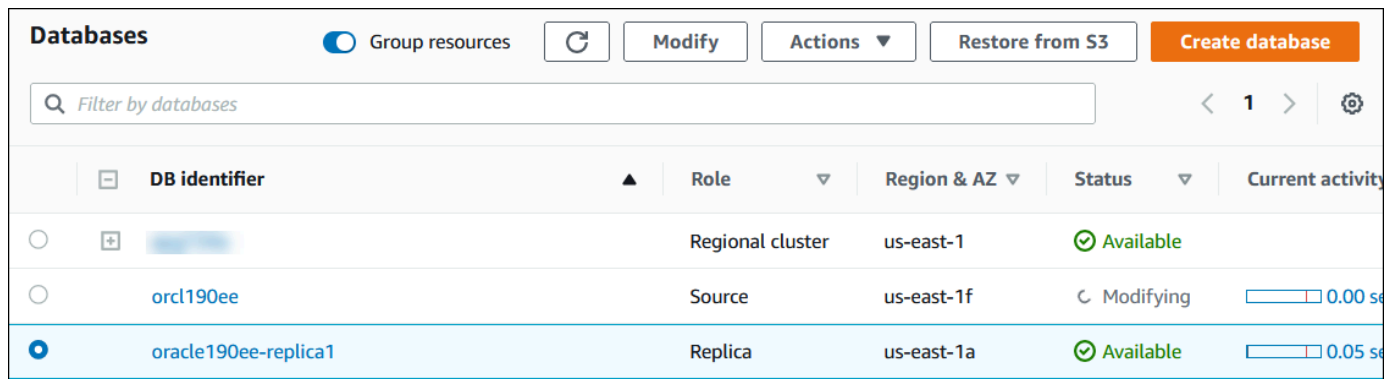
主控台

若要將 Oracle 僅供讀取複本容錯移轉為主要資料庫角色

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在 Amazon RDS 主控台，選擇 Databases (資料庫)。

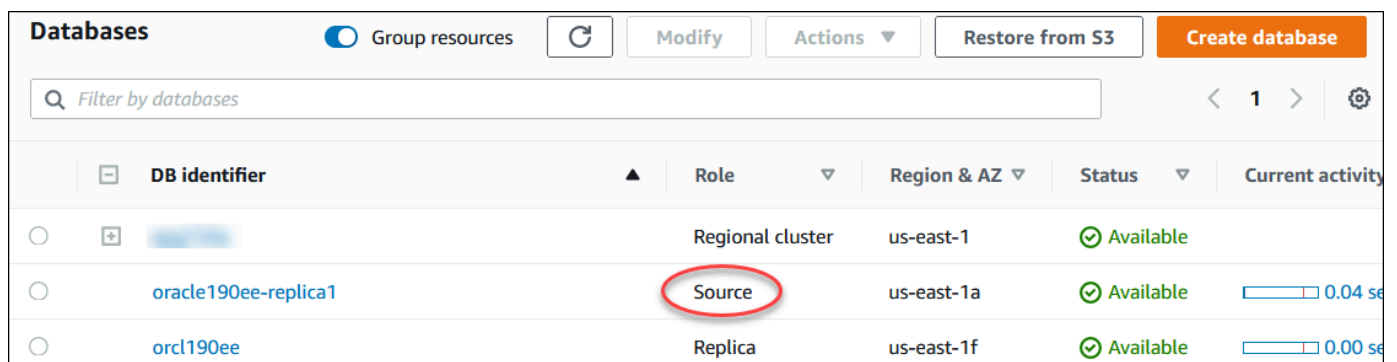
Databases (資料庫) 窗格隨即出現。每個僅供讀取複本會在 Role (角色) 欄中顯示 Replica (複本)。

3. 選擇您要容錯移轉為主要角色的僅供讀取複本。
4. 對於 Actions (動作)，選擇 Switch over replica (容錯移轉複本)。
5. 選擇 I acknowledge (我確認)。然後選擇 Switch over replica (容錯移轉複本)。
6. 在 Databases (資料庫) 頁面上監控容錯移轉進度。



DB identifier	Role	Region & AZ	Status	Current activity
[Redacted]	Regional cluster	us-east-1	Available	
orcl190ee	Source	us-east-1f	Modifying	0.00 s
oracle190ee-replica1	Replica	us-east-1a	Available	0.05 s

容錯移轉完成時，容錯移轉目標的角色會從 Replica (複本) 變成 Source (來源)。



DB identifier	Role	Region & AZ	Status	Current activity
[Redacted]	Regional cluster	us-east-1	Available	
oracle190ee-replica1	Source	us-east-1a	Available	0.04 s
orcl190ee	Replica	us-east-1f	Available	0.00 s

AWS CLI

若要將 Oracle 複本切換為主要資料庫角色，請使用 AWS CLI [switchover-read-replica](#) 命令。下列範例會讓名為 *replica-to-be-made-primary* 的 Oracle 複本成為新的主要資料庫。

Example

對於LinuxmacOS、或Unix：

```
aws rds switchover-read-replica \  
  --db-instance-identifier replica-to-be-made-primary
```

在 Windows 中：

```
aws rds switchover-read-replica ^  
  --db-instance-identifier replica-to-be-made-primary
```

RDS API

若要將 Oracle 複本容錯移轉為主要資料庫角色，請使用必要參數 `DBInstanceIdentifier` 呼叫 Amazon RDS API [SwitchoverReadReplica](#) 操作。此參數指定您要擔任主要資料庫角色之 Oracle 複本的名稱。

監控 Oracle Data Guard 容錯移轉

若要檢查執行個體的狀態，請使用 AWS CLI 指令 `describe-db-instances`。下列命令會檢查資料庫執行個體 *orcl2* 的狀態。此資料庫在容錯移轉前是待命資料庫，但在容錯移轉後是新的主要資料庫。

```
aws rds describe-db-instances \  
  --db-instance-identifier orcl2
```

若要確認容錯移轉成功完成，請查詢 `V$DATABASE.OPEN_MODE`。檢查新主要資料庫的值是否為 `READ WRITE`。

```
SELECT OPEN_MODE FROM V$DATABASE;
```

若要尋找切換相關事件，請使用 CLI 命令。AWS `describe-events` 下列範例會尋找 *orcl2* 執行個體。

```
aws rds describe-events \  
  --source-identifier orcl2 \  
  --source-type db-instance
```

對 RDS for Oracle 複本進行故障診斷

本節說明可能的複寫問題和解決方案。

主題

- [監控 Oracle 複寫延遲](#)
- [新增或修改觸發程序後，對 Oracle 複寫失敗進行疑難排解](#)

監控 Oracle 複寫延遲

若要在 Amazon CloudWatch 中監控複寫延遲，請檢視 Amazon RDS ReplicaLag 指標。如需複寫延遲時間的更多資訊，請參閱 [監控僅供讀取複本](#) 和 [Amazon RDS 的 Amazon CloudWatch 指標](#)。

對於僅供讀取複本，若複寫延遲太長，請查詢下列檢視：

- V\$ARCHIVED_LOG – 顯示已經套用於僅供讀取複本的認可。
- V\$DATAGUARD_STATS – 顯示元件的分解細節構成 ReplicaLag 指標。
- V\$DATAGUARD_STATUS – 顯示來自 Oracle 內部複寫過程的記錄輸出。

對於掛載複本，若延遲時間過長，則您無法查詢 V\$ 檢視。相反地，請執行下列動作：

- 檢查 CloudWatch 中的 ReplicaLag 指標。
- 在控制台中檢查複本的提醒日誌檔案。查找復原訊息中的錯誤。訊息包括日誌序列號，您可以將其與主序列號進行比較。如需詳細資訊，請參閱 [Oracle 資料庫日誌檔案](#)。

新增或修改觸發程序後，對 Oracle 複寫失敗進行疑難排解

如果您新增或修改任何觸發程序，而且如果在此之後複寫失敗，則可能會觸發問題。確定觸發程序排除了 RDS 複寫所需的下列使用者帳戶：

- 具有管理員權限的使用者帳戶
- SYS
- SYSTEM
- RDS_DATAGUARD
- rdsdb

如需詳細資訊，請參閱 [RDS for Oracle 複本的其他事項需求](#)。

將選項新增至 Oracle 資料庫執行個體

在 Amazon RDS 中，選項是額外的功能。以下說明了可新增至執行 Oracle 資料庫引擎之 Amazon RDS 執行個體的選項。

主題

- [Oracle 資料庫選項概觀](#)
- [Amazon S3 整合](#)
- [Oracle Application Express \(APEX\)](#)
- [Amazon EFS 整合](#)
- [Oracle Java 虛擬機器](#)
- [Oracle Enterprise Manager](#)
- [Oracle Label Security](#)
- [Oracle Locator](#)
- [Oracle 原生網路加密](#)
- [Oracle OLAP](#)
- [Oracle Secure Sockets Layer](#)
- [Oracle Spatial](#)
- [Oracle SQLT](#)
- [Oracle Statspack](#)
- [Oracle 時區](#)
- [Oracle 時區檔案自動升級](#)
- [Oracle 透明資料加密](#)
- [Oracle UTL_MAIL](#)
- [Oracle XML 資料庫](#)

Oracle 資料庫選項概觀

若要啟用您 Oracle 資料庫的選項，請將其新增至選項群組，然後使選項群組與您的資料庫執行個體產生關聯。如需詳細資訊，請參閱 [使用選項群組](#)。

主題

- [Oracle 資料庫選項的摘要](#)
- [支援不同版本的選項](#)
- [特定選項的記憶體需求](#)

Oracle 資料庫選項的摘要

對於 Oracle 資料庫執行個體，您可以新增下列選項。

選項	選項 ID
Amazon S3 整合	S3_INTEGRATION
Oracle Application Express (APEX)	APEX APEX-DEV
Oracle Enterprise Manager	OEM OEM_AGENT
Oracle Java 虛擬機器	JVM
Oracle Label Security	OLS
Oracle Locator	LOCATOR
Oracle 原生網路加密	NATIVE_NETWORK_ENCRYPTION
Oracle OLAP	OLAP
Oracle Secure Sockets Layer	SSL
Oracle Spatial	SPATIAL
Oracle SQLT	SQLT
Oracle Statspack	STATSPACK
Oracle 時區	Timezone

選項	選項 ID
Oracle 時區檔案自動升級	TIMEZONE_FILE_AUTO UPGRADE
Oracle 透明資料加密	TDE
Oracle UTL_MAIL	UTL_MAIL
Oracle XML 資料庫	XMLDB

支援不同版本的選項

RDS for Oracle 會阻止您將不受支援的選項新增至版本。若要找出不同版本的 Oracle 資料庫所支援的 RDS 選項，請使用命令 `aws rds describe-option-group-options`。下列範例列出 Oracle 資料庫 19c Enterprise Edition 的支援選項。

```
aws rds describe-option-group-options \  
  --engine-name oracle-ee \  
  --major-engine-version 19
```

如需詳細資訊，請參閱《AWS CLI 命令參考》中的 [describe-option-group-options](#)。

特定選項的記憶體需求

某些選項在資料庫執行個體上執行時需要額外記憶體。例如，「Oracle Enterprise Manager 資料庫控制」使用大約 300 MB 的 RAM。如果您對小型資料庫執行個體啟用此選項，則可能會因為記憶體受限而遭遇效能問題。您可以調整 Oracle 參數，使得資料庫需要較少的 RAM。或者，您也可以擴展為較大的資料庫執行個體。

Amazon S3 整合

您可在 RDS for Oracle 資料庫執行個體與 Amazon S3 儲存貯體之間傳輸檔案。您可以使用 Amazon S3 與 Oracle Data Pump 等 Oracle 資料庫功能整合。例如，您可從 Amazon S3 下載 Data Pump 檔案至 RDS for Oracle 資料庫執行個體。如需詳細資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle](#)。

Note

資料庫執行個體與 Amazon S3 儲存貯體必須在相同的 AWS 區域中。

主題

- [設定 IAM 許可權限以進行 RDS for Oracle 與 Amazon S3 的整合](#)
- [新增 Amazon S3 整合選項](#)
- [在 Amazon RDS for Oracle 和 Amazon S3 儲存貯體之間傳輸檔案](#)
- [針對 Amazon S3 整合進行疑難排解](#)
- [移除 Amazon S3 整合選項](#)

設定 IAM 許可權限以進行 RDS for Oracle 與 Amazon S3 的整合

若要讓 RDS for Oracle 與 Amazon S3 整合，則您的資料庫執行個體必須具有 Amazon S3 儲存貯體的存取權。您的資料庫執行個體使用的 Amazon VPC 不需要提供存取給 Amazon S3 端點。

RDS for Oracle 支援在一個帳戶中的資料庫執行個體和不同帳戶中的 Amazon S3 儲存貯體之間傳輸檔案。如需採取額外步驟，下列各節中會說明這些步驟。

主題

- [步驟 1：建立 Amazon RDS 角色的 IAM 政策](#)
- [步驟 2：\(選用\) 建立 Amazon S3 儲存貯體的 IAM 政策](#)
- [步驟 3：建立您的資料庫執行個體的 IAM 角色並附加政策](#)
- [步驟 4：建立 IAM 角色與 RDS for Oracle 資料庫執行個體的關聯](#)

步驟 1：建立 Amazon RDS 角色的 IAM 政策

在此步驟中，您會建立具有在 Amazon S3 儲存貯體和 RDS 資料庫執行個體之間傳輸檔案所需許可的 AWS Identity and Access Management (IAM) 政策。此步驟假設您已建立 S3 儲存貯體。

在您建立政策之前，請記住下列資訊：

- 儲存貯體的 Amazon Resource Name (ARN)
- 如果您的儲存貯體使用 SSE-KMS 或 SSE-S3 加 AWS KMS 密，則為您的金鑰提供 ARN

Note

RDS for Oracle 資料庫執行個體無法存取使用 SSE-C 加密的 Amazon S3 儲存貯體。

如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用伺服器端加密保護資料](#)。


主控台

建立 IAM 政策以允許 Amazon RDS 存取您的 Amazon S3 儲存貯體

1. 開啟 [IAM 管理主控台](#)。
2. 在 Access management (存取管理) 下，請選擇 Policies (政策)。
3. 選擇建立政策。
4. 在 Visual editor (視覺化編輯器) 標籤中，選擇 Choose a service (選擇服務)，然後選擇 S3。
5. 在 Actions (動作) 中，選擇 Expand all (全部展開)，然後選擇從 Amazon S3 儲存貯體傳輸檔案至 Amazon RDS 所需的儲存貯體許可和物件許可。例如，請執行以下操作：
 - 展開 [清單]，然後選取 ListBucket。
 - 展開 [讀取]，然後選取 GetObject。
 - 展開 [寫入]，然後選取 PutObject 和 DeleteObject。
 - 展開 [權限管理]，然後選取 [PutObjectAcl]。若您計劃將檔案上傳至另一個帳戶所擁有的儲存貯體，而此帳戶需要儲存貯體內容的完全控制，則需要此許可權限。

物件許可權限是 Amazon S3 中物件作業的許可權限。您必須授予其儲存貯體中的物件，而非儲存貯體本身。如需詳細資訊，請參閱[物件作業的許可權限](#)。

6. 選擇資源，然後執行下列動作：
 - a. 選擇特定。
 - b. 針對儲存貯體，請選擇新增 ARN。輸入您的儲存貯體 ARN。儲存貯體名稱會自動填入。接著選擇 Add (新增)。
 - c. 如果顯示物件資源，請選擇新增 ARN 以手動新增資源，或選擇任何。

 Note

您可以將 Amazon Resource Name (ARN) 設為更具體的 ARN 值，以允許 Amazon RDS 只能存取 Amazon S3 儲存貯體中的特定檔案或資料夾。如需如何為 Amazon S3 定義存取原則的詳細資訊，請參閱[管理 Amazon S3 資源的存取許可](#)。

7. (選用) 選擇 Add additional permissions (新增其他許可)，將資源新增至政策。例如，請執行以下操作：
 - a. 如果您的儲存貯體使用自訂 KMS 金鑰加密，請為服務選取 KMS。
 - b. 針對手動動作，請選取下列項目：
 - 加密
 - ReEncrypt 從和ReEncrypt 到
 - 解密
 - DescribeKey
 - GenerateData關鍵
 - c. 針對資源，請選擇特定。
 - d. 針對金鑰，請選擇新增 ARN。輸入自訂金鑰的 ARN 作為資源，然後選擇新增。

如需詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的使用存放於 AWS Key Management Service (SSE-KMS) 的 KMS 金鑰使用伺服器端加密來保護資料。
 - e. 如果您想要 Amazon RDS 存取其他儲存貯體，請新增這些儲存貯體的 ARN。(選擇性) 您也可以授權存取 Amazon S3 中的所有儲存貯體和物件。
8. 選擇 Next: Tags (下一步：標籤)，然後選擇 Next: Review (下一步：檢閱)。
9. 在 Name (名稱) 輸入您的 IAM 政策名稱，例如 rds-s3-integration-policy。您可使用此項名稱，建立與資料庫叢集相關的 IAM 角色。您也可以新增選用的 Description (描述) 值。
10. 選擇 Create policy (建立政策)。

AWS CLI

建立可授予 Amazon RDS 存取 Amazon S3 儲存貯體的 AWS Identity and Access Management (IAM) 政策。在您建立政策後，請記下政策的 ARN。您需要 ARN 才能進行後續步驟。

依據所需的存取類型，在政策中納入適當動作：

- `GetObject` – 必須從 Amazon S3 儲存貯體傳輸檔案至 Amazon RDS。
- `ListBucket` – 必須從 Amazon S3 儲存貯體傳輸檔案至 Amazon RDS。
- `PutObject` – 必須由 Amazon RDS 傳輸檔案至 Amazon S3 儲存貯體。

下列 AWS CLI 命令會建立以這些選項命名 *rds-s3-integration-policy* 的 IAM 政策。其中將授予存取至名為 *DOC-EXAMPLE-BUCKET* 的儲存貯體。

Example

對於 Linux/macOS、或 Unix：

```
aws iam create-policy \  
  --policy-name rds-s3-integration-policy \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "s3integration",  
        "Action": [  
          "s3:GetObject",  
          "s3:ListBucket",  
          "s3:PutObject"  
        ],  
        "Effect": "Allow",  
        "Resource": [  
          "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
          "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
        ]  
      }  
    ]  
  }'  
'
```

下列範例包含自訂 KMS 金鑰的許可。

```
aws iam create-policy \  

```

```
--policy-name rds-s3-integration-policy \  
--policy-document '{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "s3integration",  
      "Action": [  
        "s3:GetObject",  
        "s3:ListBucket",  
        "s3:PutObject",  
        "kms:Decrypt",  
        "kms:Encrypt",  
        "kms:ReEncrypt*",  
        "kms:GenerateDataKey",  
        "kms:DescribeKey",  
      ],  
      "Effect": "Allow",  
      "Resource": [  
        "arn:aws:s3::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3::DOC-EXAMPLE-BUCKET/*",  
        "arn:aws:kms::your-kms-arn"  
      ]  
    }  
  ]  
}'
```

在 Windows 中：

```
aws iam create-policy ^  
--policy-name rds-s3-integration-policy ^  
--policy-document '{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "s3integration",  
      "Action": [  
        "s3:GetObject",  
        "s3:ListBucket",  
        "s3:PutObject"  
      ],  
      "Effect": "Allow",  
      "Resource": [  
        "arn:aws:s3::DOC-EXAMPLE-BUCKET",
```

```

        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
}
]
}'

```

下列範例包含自訂 KMS 金鑰的許可。

```

aws iam create-policy ^
--policy-name rds-s3-integration-policy ^
--policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "s3integration",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey",
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
        "arn:aws:kms:::your-kms-arn"
      ]
    }
  ]
}'

```

步驟 2：(選用) 建立 Amazon S3 儲存貯體的 IAM 政策

只有在下列情況中才需要此步驟：

- 您計劃從一個帳戶 (帳戶 A) 將檔案上傳至 Amazon S3 儲存貯體，然後從另一個帳戶 (帳戶 B) 進行存取。
- 帳戶 B 擁有儲存貯體。

- 帳戶 B 需要載入儲存貯體中物件的完整控制。

若上述條件不適用於您，請跳到 [步驟 3：建立您的資料庫執行個體的 IAM 角色並附加政策](#)。

如要建立儲存貯體政策，請確定您具有下列項目：

- 帳戶 A 的帳戶 ID
- 帳戶 A 的使用者名稱
- 帳戶 B 中 Amazon S3 儲存貯體的 ARN 值

主控台

建立或編輯儲存貯體政策

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 <https://console.aws.amazon.com/s3/>。
2. 在 Buckets (儲存貯體) 清單中，選擇要建立儲存貯體政策的儲存貯體名稱，或您想編輯之儲存貯體政策的儲存貯體名稱。
3. 選擇 Permissions (許可)。
4. 在 Bucket policy (儲存貯體政策) 下方，選擇 Edit (編輯)。這會開啟「編輯儲存貯體政策」頁面。
5. 在 Edit bucket policy (編輯儲存貯體政策) 頁面，瀏覽《Amazon S3 使用者指南》中的 Policy examples (政策範例)、選擇 Policy generator (政策產生器) 以自動產生政策，或在 Policy (政策) 區段編輯 JSON。

如果您選擇「策略產生器」，則「AWS 策略產生器」會在新視窗中開啟：

- a. 在 AWS Policy Generator (政策產生器) 頁面上，在 Select Type of Policy (選取政策類型) 中選擇 S3 Bucket Policy (S3 儲存貯體政策)。
- b. 在提供的欄位中輸入資訊，以新增陳述式，然後選擇 Add Statement (新增陳述式)。針對您想要新增的任意數量陳述式重複此動作。如需這些欄位的詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。

Note

為方便起見，Edit bucket policy (編輯儲存貯體政策) 頁面會在 Policy (政策) 文字欄位上方顯示目前儲存貯體的 Bucket ARN (Amazon Resource Name) (儲存貯體 ARN)

(Amazon 資源名稱))。您可以複製此 ARN，以在 AWS Policy Generator (政策產生器) 頁面上的陳述式中使用。

- c. 完成新增陳述式後，選擇 Generate Policy (產生政策)。
 - d. 複製產生的政策文字，選擇 Close (關閉)，然後退回 Amazon S3 主控台中的 Edit bucket policy (編輯儲存貯體政策) 頁面。
6. 在 Policy (政策) 方塊中，編輯現有政策，或從政策產生器貼上儲存貯體政策。請務必先處理安全性警告、錯誤、一般警告，以及建議，然後再儲存政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-A-ID:account-A-user"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET",
        "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET/*"
      ]
    }
  ]
}
```

7. 選擇 Save changes (儲存變更)，此選項會跳至儲存貯體許可頁面。

步驟 3：建立您的資料庫執行個體的 IAM 角色並附加政策

此步驟會假設您已於 [步驟 1：建立 Amazon RDS 角色的 IAM 政策](#) 中建立了 IAM 政策。於此步驟中，您會建立 RDS for Oracle 資料庫執行個體的角色，然後將政策附加至角色。

主控台

建立 IAM 角色以允許 Amazon RDS 存取 Amazon S3 儲存貯體

1. 開啟 [IAM 管理主控台](#)。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 選擇 AWS 服務。
5. 對於其他 AWS 服務的使用案例：，選擇 RDS，然後選擇 RDS-將角色新增至資料庫。然後選擇下一步。
6. 請在許可政策之下的搜尋輸入您在 [步驟 1：建立 Amazon RDS 角色的 IAM 政策](#) 建立的 IAM 政策名稱，然後於政策出現在清單時加以選取。然後選擇下一步。
7. 在角色名稱中輸入 IAM 角色名稱，例如 `rds-s3-integration-role`。您也可以新增選用的 Description (描述) 值。
8. 選擇建立角色。

AWS CLI

如要建立角色並將政策附加至該角色

1. 建立 IAM 角色，讓 Amazon RDS 可代表您存取您的 Amazon S3 儲存貯體。

建議您在資源型信任關係中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，將服務的許可限定於特定資來源。這是防止[混淆代理人問題](#)最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在信任關係中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 Amazon 資源名稱 (ARN)。

下列 AWS CLI 命令會建立 `rds-s3-integration-role` 為此目的命名的角色。

Example

對於LinuxmacOS、或Unix：

```
aws iam create-role \  
  --role-name rds-s3-integration-role \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",  
        "Condition": {  
          "StringEquals": {  
            "aws:SourceAccount": my_account_ID,  
            "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"  
          }  
        }  
      }  
    ]  
  }'
```

在 Windows 中：

```
aws iam create-role ^  
  --role-name rds-s3-integration-role ^  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",  
        "Condition": {  
          "StringEquals": {  
            "aws:SourceAccount": my_account_ID,  
            "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"  
          }  
        }  
      }  
    ]  
  }'
```

```
}  
  }  
}'  
]
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

2. 角色建立後，請記下角色的 ARN。您需要 ARN 才能進行後續步驟。
3. 請將您建立的政策連接到您建立的角色。

下列 AWS CLI 命令會將原則附加至名為的角色 *rds-s3-integration-role*。

Example

對於LinuxmacOS、或Unix：

```
aws iam attach-role-policy \  
  --policy-arn your-policy-arn \  
  --role-name rds-s3-integration-role
```

在 Windows 中：

```
aws iam attach-role-policy ^  
  --policy-arn your-policy-arn ^  
  --role-name rds-s3-integration-role
```

將 *your-policy-arn* 替換為您前個步驟記下的政策 ARN。

步驟 4：建立 IAM 角色與 RDS for Oracle 資料庫執行個體的關聯

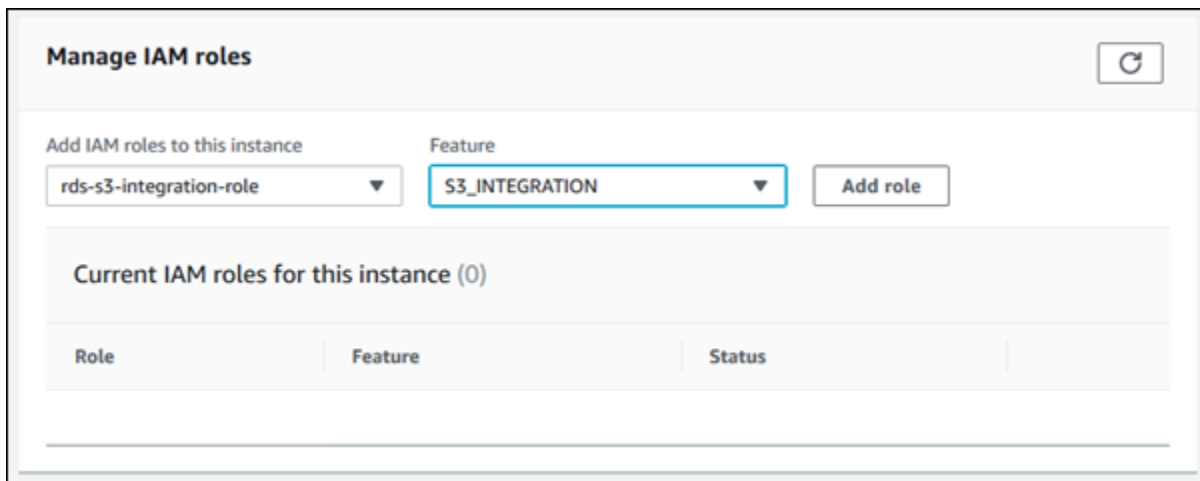
設定 Amazon S3 整合許可的最後一個步驟是將您的 IAM 角色與資料庫執行個體建立關聯。請注意以下要求：

- 您必須能存取連接了必要 Amazon S3 許可政策的角色。
- 您一次只能將一個 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯。
- 您的資料庫執行個體必須處於可用狀態。

主控台

如要建立 IAM 角色與 RDS for Oracle 資料庫執行個體的關聯

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇 RDS for Oracle 資料庫執行個體名稱以顯示其詳細資訊。
4. 在 Connectivity & security (連線和安全) 索引標籤上，向下捲動至頁面底部的 Manage IAM roles (管理 IAM 角色) 區段。
5. 對於新增 IAM 角色到此執行個體，請選擇您在 [步驟 3：建立您的資料庫執行個體的 IAM 角色並附加政策](#) 中建立的角色。
6. 針對 Feature (功能) 選擇 S3_INTEGRATION。



7. 選擇 Add role (新增角色)。

AWS CLI

下列 AWS CLI 命令會將角色新增至名為的 Oracle 資料庫執行個體 *mydbinstance*。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier mydbinstance \  
  --feature-name S3_INTEGRATION \  
  --role-arn your-role-arn
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --feature-name S3_INTEGRATION ^  
  --role-arn your-role-arn
```

將 *your-role-arn* 替換為您前個步驟記下的角色 ARN。S3_INTEGRATION 必須在 --feature-name 選項中指定。

新增 Amazon S3 整合選項

若要整合 Amazon RDS for Oracle 與 Amazon S3，您的資料庫執行個體必須與包含 S3_INTEGRATION 選項的選項群組建立關聯。

主控台

設定選項群組進行 Amazon S3 整合

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 S3_INTEGRATION 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 S3_INTEGRATION 選項新增至選項群組。

如需將選項新增至選項群組的相關資訊，請參閱[將選項新增至選項群組](#)。

3. 建立新的 RDS for Oracle 資料庫執行個體，將其與選項群組建立關聯，或修改 RDS for Oracle 資料庫執行個體，使其與選項群組建立關聯。

如需建立資料庫執行個體的相關資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。

如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

設定選項群組進行 Amazon S3 整合

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 S3_INTEGRATION 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 S3_INTEGRATION 選項新增至選項群組。

例如，下列 AWS CLI 指令會將 S3_INTEGRATION 選項新增至名為的選項群組 `myoptiongroup`。

Example

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name myoptiongroup \  
  --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name myoptiongroup ^  
  --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. 建立新的 RDS for Oracle 資料庫執行個體，將其與選項群組建立關聯，或修改 RDS for Oracle 資料庫執行個體，使其與選項群組建立關聯。

如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

如需修改 RDS for Oracle 資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

在 Amazon RDS for Oracle 和 Amazon S3 儲存貯體之間傳輸檔案

如要在 RDS for Oracle 資料庫執行個體和 Amazon S3 儲存貯體之間傳輸檔案，您可以使用 Amazon RDS 套件 `rdsadmin_s3_tasks`。您可在上傳檔案時裡用 GZIP 壓縮文件，並在下載時對其進行解壓縮。

主題

- [檔案傳輸的需求和限制](#)
- [將檔案從 RDS for Oracle 資料庫執行個體上傳至 Amazon S3 儲存貯體](#)
- [將檔案從 Amazon S3 儲存貯體下載至 Oracle 資料庫執行個體](#)
- [監控檔案傳輸狀態](#)

檔案傳輸的需求和限制

在資料庫執行個體和 Amazon S3 儲存貯體之間傳輸檔案之前，請注意以下事項：

- `rdsadmin_s3_tasks` 套件會傳輸位於單一目錄中的檔案。您無法在傳輸中包含子目錄。
- Amazon S3 儲存貯體中的物件大小上限為 5 TB。
- 以非同步方式 `rdsadmin_s3_tasks` 執行所建立的工作。
- 您可以從「資料汲取」目錄 (例如 `DATA_PUMP_DIR`，或任何使用者建立的目錄) 上傳檔案。您無法從 Oracle 背景處理作業所使用的目錄 (例如 `adumpbdump`、或 `trace` 目錄) 上傳檔案。
- 每個程序呼叫的下載限制為 2000 個檔案 `download_from_s3`。如果您需要從 Amazon S3 下載 2000 個以上的檔案，請將您的下載分割為不同的動作，每次程序呼叫不超過 2000 個檔案。
- 如果您的下載資料夾中存在檔案，且您嘗試下載具有相同名稱的檔案，則 `download_from_s3` 會跳過下載。[若要從下載目錄中移除檔案，請使用 PL/SQL 程序 UTL_F.F 移除。](#)

將檔案從 RDS for Oracle 資料庫執行個體上傳至 Amazon S3 儲存貯體

如要從資料庫執行個體上傳檔案至 Amazon S3 儲存貯體，請使用程序 `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`。例如，您可上傳 Oracle Recovery Manager (RMAN) 備份檔案或 Oracle Data Pump 檔案。如需使用物件的詳細資訊，請參閱《[Amazon Simple Storage Service 使用者指南](#)》。如需 RMAN 備份的詳細資訊，請參閱[執行 Oracle 資料庫執行個體的一般 RMAN 任務](#)。

`rdsadmin.rdsadmin_s3_tasks.upload_to_s3` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_bucket_name</code>	VARCHAR2	–	必要	上傳檔案目的地的 Amazon S3 儲存貯體名稱。
<code>p_directory_name</code>	VARCHAR2	–	必要	上傳檔案來源的 Oracle 目錄物件名稱。目錄可為 Data Pump 目錄任何由使用者定義的目錄物件，例如 <code>DATA_PUMP_DIR</code> 。您無法從背景處理程序所使用的目錄上傳檔

參數名稱	資料類型	預設	必要	描述
				<p>案adump，例如bdump、和trace。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>您只能由指定目錄上傳檔案。您不能由指定目錄之中的子目錄上傳檔案。</p> </div>
p_s3_prefix	VARCHAR2	–	必要	<p>上傳檔案的 Amazon S3 檔案名稱字首。空白字首會將所有檔案上傳至指定 Amazon S3 儲存貯體的頂層，不會在檔案名稱新增字首。</p> <p>例如若字首為 folder_1/oradb，檔案會上傳至 folder_1。在這種情況下，oradb 字首會新增至各個檔案。</p>
p_prefix	VARCHAR2	–	必要	<p>檔案名稱必須與其相符才能上傳的檔案名稱字首。空白字首會將上傳指定目錄之中的檔案。</p>
p_compression_level	NUMBER	0	選擇性	<p>GZIP 壓縮層級。有效值範圍從 0 到 9：</p> <ul style="list-style-type: none"> • 0 – 無壓縮 • 1 – 最快速的壓縮 • 9 – 最高壓縮

參數名稱	資料類型	預設	必要	描述
p_bucket_owner_full_control	VARCHAR2	–	選擇性	儲存貯體的存取控制設定。唯一有效值為 null 或 FULL_CONTROL 。僅當您將檔案從一個帳戶 (帳戶 A) 上傳至另一個帳戶 (帳戶 B) 所擁有的儲存貯體，且帳戶 B 需要完全控制這些檔案時，才需要此設定。

rdsadmin.rdsadmin_s3_tasks.upload_to_s3 程序的傳回值是一個任務 ID。

下列範例會將 *DATA_PUMP_DIR* 目錄中的所有檔案上傳到名為 *DOC/EXAMPLE* 儲存貯體的 Amazon S3 儲存貯體。檔案未壓縮。

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name => 'DOC-EXAMPLE-BUCKET',
    p_prefix      => '',
    p_s3_prefix   => '',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

以下範例將 *db* 目錄中所有字首為 *DATA_PUMP_DIR* 的檔案上傳至名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體。Amazon RDS 對檔案套用最高層級的 GZIP 壓縮。

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name      => 'DOC-EXAMPLE-BUCKET',
    p_prefix           => 'db',
    p_s3_prefix        => '',
    p_directory_name   => 'DATA_PUMP_DIR',
    p_compression_level => 9)
AS TASK_ID FROM DUAL;
```

下列範例將 *DATA_PUMP_DIR* 目錄的所有檔案，上傳至名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體。檔案上傳至 *dbfiles* 資料夾。於此範例中，GZIP 壓縮層級為 *1*，此為最快速的壓縮層級。

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name      => 'DOC-EXAMPLE-BUCKET',
  p_prefix           => '',
  p_s3_prefix        => 'dbfiles/',
  p_directory_name   => 'DATA_PUMP_DIR',
  p_compression_level => 1)
AS TASK_ID FROM DUAL;
```

以下範例將 *DATA_PUMP_DIR* 目錄的所有檔案，上傳至名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體。檔案上傳至 *dbfiles* 資料夾，而 *ora* 則新增至每個檔案名稱的開頭。未套用任何壓縮。

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name      => 'DOC-EXAMPLE-BUCKET',
  p_prefix           => '',
  p_s3_prefix        => 'dbfiles/ora',
  p_directory_name   => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

下列範例假定該命令執行於帳戶 A 中，但帳戶 B 需要儲存貯體內容的完全控制。命令 `rdsadmin_s3_tasks.upload_to_s3` 會將 *DATA_PUMP_DIR* 目錄中的所有檔案傳輸至名為 *s3bucketOwnedByAccountB* 的儲存貯體。存取控制設定為 `FULL_CONTROL`，則帳戶 B 可存取儲存貯體中的檔案。GZIP 壓縮層級為 *6*，此會平衡速度和檔案大小。

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name      => 's3bucketOwnedByAccountB',
  p_prefix           => '',
  p_s3_prefix        => '',
  p_directory_name   => 'DATA_PUMP_DIR',
  p_bucket_owner_full_control => 'FULL_CONTROL',
  p_compression_level => 6)
AS TASK_ID FROM DUAL;
```

在各項範例中，SELECT 陳述式以 VARCHAR2 資料類型傳回任務 ID。

您可以透過顯示任務的輸出檔案來檢視結果。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

以程序傳回的任務 ID 取代 *task-id*。

Note

任務是以非同步方式執行。

將檔案從 Amazon S3 儲存貯體下載至 Oracle 資料庫執行個體

若要從 Amazon S3 儲存貯體下載檔案至 Oracle 資料庫執行個體，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_s3_tasks.download_from_s3`。

`download_from_s3` 程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_bucket_name</code>	VARCHAR	–	必要	下載檔案來源的 Amazon S3 儲存貯體名稱。
<code>p_directory_name</code>	VARCHAR	–	必要	下載檔案目的地的 Oracle 目錄物件名稱。目錄可為 Data Pump 目錄任何由使用者定義的目錄物件，例如 <code>DATA_PUMP_DIR</code> 。
<code>p_error_on_zero_downloads</code>	VARCHAR	FALSE	選用	此旗標可決定當 Amazon S3 儲存貯體中沒有物件符合字首時，任務是否會引發錯誤。如果此參數未設定或設為 FALSE (預設值)，任務會列印訊息，指出找不到物件，但不會引發例外狀況或失敗。如果此參數為 TRUE，則任務會引發例外狀況並失敗。 可能無法進行比對測試的字首規格範例是字首中的空格 (如 <code>' import/test9.log'</code> 中所示) 和大小寫不符 (如 <code>test9.log</code> 和 <code>test9.LOG</code> 中所示)。
<code>p_s3_prefix</code>	VARCHAR	–	必要	檔案名稱必須與其相符才能下載的檔案名稱字首。空白字首會在指定的

參數名稱	資料類型	預設	必要	描述
				<p>Amazon S3 儲存貯體中下載所有第一層檔案，但不是在該儲存貯體資料夾中的檔案。</p> <p>程序僅會由符合字首的第一層資料夾下載 Amazon S3 物件。符合指定字串的巢狀目錄結構不會下載。</p> <p>例如假設 Amazon S3 儲存貯體具有資料夾結構 <code>folder_1/folder_2/folder_3</code>。您指定 <code>'folder_1/folder_2/'</code> 字首。在這種情況下，只會下載 <code>folder_2</code> 之中的檔案，不會下載 <code>folder_1</code> 或 <code>folder_3</code> 的檔案。</p> <p>如果您反而指定 <code>'folder_1/folder_2'</code> 字首，則會下載 <code>folder_1</code> 中符合 <code>'folder_2'</code> 字首的所有檔案，且不會下載 <code>folder_2</code> 中的任何檔案。</p>
<code>p_decompression_format</code>	VARCHAR	–	選用	解壓縮格式。有效值為 NONE 適用於未解壓縮，而 GZIP 適用於解壓縮。

`rdsadmin.rdsadmin_s3_tasks.download_from_s3` 程序的傳回值是一個任務 ID。

以下範例將名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體中所有的檔案下載至 *DATA_PUMP_DIR* 目錄。這些檔案未壓縮，因此無須套用解壓縮。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'DOC-EXAMPLE-BUCKET',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

以下範例將名為 *db* 的 Amazon S3 儲存貯體之中字首為 *DOC-EXAMPLE-BUCKET* 的所有檔案下載至 *DATA_PUMP_DIR* 目錄。這些檔案以 GZIP 進行壓縮，因此可套用解壓縮。參數 *p_error_on_zero_downloads* 會開啟字首錯誤檢查，因此，如果字首與儲存貯體中的任何檔案不符，任務就會引發例外狀況並失敗。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name          => 'DOC-EXAMPLE-BUCKET',
    p_s3_prefix            => 'db',
    p_directory_name       => 'DATA_PUMP_DIR',
    p_decompression_format => 'GZIP',
    p_error_on_zero_downloads => 'TRUE')
AS TASK_ID FROM DUAL;
```

以下範例將名為 *myfolder/* 的 Amazon S3 儲存貯體之中資料夾 *DOC-EXAMPLE-BUCKET* 的所有檔案下載至 *DATA_PUMP_DIR* 目錄。使用 *p_s3_prefix* 參數，指定 Amazon S3 資料夾。上傳的檔案使用 GZIP 進行壓縮，但於下載過程中不會解壓縮。

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name          => 'DOC-EXAMPLE-BUCKET',
    p_s3_prefix            => 'myfolder/',
    p_directory_name       => 'DATA_PUMP_DIR',
    p_decompression_format => 'NONE')
AS TASK_ID FROM DUAL;
```

以下範例將名為 *DOC-EXAMPLE-BUCKET* 的 Amazon S3 儲存貯體中的檔案 *mydumpfile.dmp* 下載至 *DATA_PUMP_DIR* 目錄。未套用任何解壓縮。


```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name          => 'DOC-EXAMPLE-BUCKET',
    p_s3_prefix            => 'mydumpfile.dmp',
    p_directory_name       => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

在各項範例中，SELECT 陳述式以 VARCHAR2 資料類型傳回任務 ID。

您可以透過顯示任務的輸出檔案來檢視結果。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

以程序傳回的任務 ID 取代 *task-id*。

 Note

任務是以非同步方式執行。

您可使用 `UTL_FILE.FREMOVE` Oracle 程序由目錄移除檔案。如需詳細資訊，請參閱 Oracle 文件中的 [FREMOVE 程序](#)。


監控檔案傳輸狀態

檔案傳輸作業會在開始及完成時發佈 Amazon RDS 事件。事件訊息包含檔案傳輸的任務 ID。如需檢視事件相關資訊，請參閱[檢視 Amazon RDS 事件](#)。

您可在 `bdump` 檔案之中，檢視進行中作業的狀態。`bdump` 檔案位在 `/rdsdbdata/log/trace` 目錄：每個 `bdump` 檔案的名稱格式如下。

```
dbtask-task-id.log
```

將 *task-id* 替換為您要監控作業的 ID。

 Note

任務是以非同步方式執行。

您可以使用 `rdsadmin.rds_file_util.read_text_file` 預存程序檢視 `bdump` 檔案內容。例如以下查詢會傳回 *dbtask-1234567890123-1234.log* `bdump` 檔案的內容。

```
SELECT text FROM  
table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-1234567890123-1234.log'));
```

下列範例顯示傳輸失敗的日誌。

```
TASK_ID
```

1234567890123-1234

TEXT

```
2023-04-17 18:21:33.993 UTC [INFO ] File #1: Uploading the file /rdsdbdata/datapump/
A123B4CDEF567890G1234567890H1234/sample.dmp to Amazon S3 with bucket name DOC-EXAMPLE-
BUCKET and key sample.dmp.
2023-04-17 18:21:34.188 UTC [ERROR] RDS doesn't have permission to write to Amazon S3
bucket name DOC-EXAMPLE-BUCKET and key sample.dmp.
2023-04-17 18:21:34.189 UTC [INFO ] The task failed.
```

針對 Amazon S3 整合進行疑難排解

如需疑難排解秘訣，請參閱 AWS Re: POST 文章 [當我將 Amazon RDS 版 Oracle 與 Amazon S3 整合時，如何解決問題？](#)。

移除 Amazon S3 整合選項

您可由資料庫執行個體移除 Amazon S3 整合選項。

若要從資料庫執行個體中移除 Amazon S3 整合選項，請執行下列其中一個動作：

- 若要從多個資料庫執行個體中移除 Amazon S3 整合選項，請從資料庫執行個體所屬的選項群組中移除 S3_INTEGRATION 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需詳細資訊，請參閱 [從選項群組移除選項](#)。

- 如要從單一資料庫執行個體移除 Amazon S3 整合，請修改執行個體，並指定不含 S3_INTEGRATION 選項的不同選項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Application Express (APEX)

Amazon RDS 會使用 APEX 和 APEX-DEV 選項，藉此支援 Oracle Application Express (APEX)。您可以將 Oracle APEX 部署成適用於 web 應用程式的執行時間環境或完整開發環境。Oracle APEX 可讓您完全在 Web 瀏覽器內建置應用程式。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Application Express](#)。

主題

- [APEX 元件](#)
- [APEX 版本需求](#)
- [Oracle APEX 和 ORDS 需求和限制](#)
- [新增 APEX 和 APEX-DEV 選項](#)
- [解除鎖定公有使用者帳戶](#)
- [設定適用於 Oracle APEX 的 RESTful 服務](#)
- [準備安裝 ORDS](#)
- [安裝和設定 ORDS 21 及更低版本](#)
- [安裝與設定 ORDS 22 及更新版本](#)
- [設定 Oracle APEX Listener](#)
- [升級 APEX 版本](#)
- [移除 APEX 選項](#)

APEX 元件

Oracle APEX 包含下列主要元件：

- 儲存庫，存放 APEX 應用程式和元件的中繼資料。儲存庫包含安裝於 Amazon RDS 資料庫執行個體中的資料表、索引及其他物件。
- 接聽程式，管理與 Oracle APEX 用戶端之間的 HTTP 通訊。接聽程式位於個別主機上，例如 Amazon EC2 執行個體、公司的內部部署伺服器或您的桌上型電腦。接聽程式接受 web 瀏覽器傳入的連線、將連線轉送至 Amazon RDS 資料庫執行個體來處理，然後將結果從儲存庫傳回給瀏覽器。RDS for Oracle 支援下列接聽程式類型：
 - 對於 APEX 5.0 及更高版本，請使用 Oracle REST 資料服務 (ORDS) 19.1 及更高版本。我們建議您使用最新支援的 Oracle APEX 和 ORDS 版本。本文件描述舊版僅供回溯相容性之用。
 - 若為 APEX 4.1.1 版，您可使用 Oracle APEX Listener 1.1.4 版。

- 您可以使用 Oracle HTTP Server 和 mod_plsql 接聽程式。

Note

Amazon RDS 不支援有內嵌 PL/SQL 閘道的 Oracle XML DB HTTP 伺服器；這不能做為 APEX 的接聽程式。一般而言，對於網際網路上執行的應用程式，Oracle 不建議使用內嵌 PL/SQL 閘道。

如需有關這些接聽程式類型的詳細資訊，請參閱 Oracle 文件中的[關於選擇 Web 接聽程式](#)。

當您將 Amazon RDS APEX 選項新增至 RDS for Oracle 資料庫執行個體時，Amazon RDS 僅會安裝 Oracle APEX 儲存庫。在個別主機上安裝您的接聽程式。

APEX 版本需求

APEX 選項將為您的資料庫執行個體使用資料庫執行個體類別上的儲存空間。下列為 Oracle APEX 支援的版本與約略儲存空間需求：

APEX 版本	儲存需求	支援的 Oracle Database 版本	備註
甲骨文顶点 23.2.V1 版	110 MiB	全部	此版本包含修補程式 35895964：APEX 23.2 的 PSE 套裝軟體 (位於 23.2.0 以上的 PSE)、修補程式版本 6。
Oracle APEX 23.1.v1 版	106 MiB	全部	此版本包含修補程式 35283657：PSE BUNDLE FOR APEX 23.1 (PSES ON TOP OF 23.1.0), PATCH_VERSION 2。
Oracle APEX 22.2.v1 版	106 MiB	全部	此版本包含修補程式 34628174：PSE BUNDLE FOR APEX 22.2 (PSES ON TOP OF 22.2.0), PATCH_VERSION 4。

APEX 版本	儲存需求	支援的 Oracle Database 版本	備註
Oracle APEX 22.1.v1 版	124 MiB	全部	此版本包含修補程式 34020981 : PSE BUNDLE FOR APEX 22.1 (PSES ON TOP OF 22.1.0), PATCH_VERSION 6。
Oracle APEX 21.2.v1 版	125 MiB	全部	此版本包含修補程式 33420059 : PSE BUNDLE FOR APEX 21.2 (PSES ON TOP OF 21.2.0), PATCH_VERSION 8。
Oracle APEX 21.1.v1 版	125 MiB	全部	此版本包含修補程式 32598392 : PSE BUNDLE FOR APEX 21.1 , PATCH_VERSION 3。
Oracle APEX 版本 20.2.v1	148 MiB	除甲骨文資料庫 21c 以外的全	<p>此版本包含修補程式 32006852 : PSE BUNDLE FOR APEX 20.2 , PATCH_VERSION 2020.11.12。您可以執行下列查詢來查看修補程編號和日期：</p> <pre>SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;</pre>
Oracle APEX 20.1.v1 版	173 MiB	除甲骨文資料庫 21c 以外的全	此版本包含修補程式 30990551 : PSE BUNDLE FOR APEX 20.1 , PATCH_VERSION 2020.07.15。
Oracle APEX 19.2.v1 版	149 MiB	除甲骨文資料庫 21c 以外的全	
Oracle APEX 19.1.v1 版	148 MiB	除甲骨文資料庫 21c 以外的全	

如需可下載的 APEX .zip 檔案，請參閱 Oracle 網站上的 [Oracle APEX 發行前封存](#)。

Oracle APEX 和 ORDS 需求和限制

請注意 APEX 和 ORDS 的以下需求：

- 您必須使用 Java 執行階段環境 (JRE)。
- 您的 Oracle 用戶端安裝必須包括下列項目：
 - 適用於系統管理任務的 SQL*Plus 或 SQL Developer
 - 用於設定 RDS for Oracle 資料庫執行個體連線的 Oracle Net Services

請注意 APEX 和 ORDS 的下列限制：

- 您不能使用 RDS 為甲骨文 CDB 與 ORDS 22 及更高版本。作為因應措施，您可以使用較低版本的 ORDS，或使用 Oracle 資料庫 19c 非 CDB。

新增 APEX 和 APEX-DEV 選項

若要將 APEX 和 APEX-DEV 選項新增至資料庫執行個體，請執行下列動作：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將 APEX 和 APEX-DEV 選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

當您新增 Amazon RDS APEX 選項時，在資料庫執行個體自動重新啟動時會發生短暫的中斷。

Note

APEX_MAIL 在已安裝 APEX 選項時可用。APEX_MAIL 套件的執行權限會授予 PUBLIC，因此您不需要有 APEX 管理帳戶就能使用它。

將 APEX 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇您想要使用的 Oracle 版本。所有版本都支援 APEX 選項。
 - b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

- 將選項新增至選項群組。如果您只想部署 Oracle APEX 執行時間環境，則只需要新增 APEX 選項。如果您想要部署完整開發環境，請新增 APEX 和 APEX-DEV 選項。

在 Version (版本) 中，選擇您要使用的 APEX 版本。

Important

如果您將 APEX 選項新增至已連接到一個或多個資料庫執行個體的現有選項群組，則會發生短暫的停機。在此停機期間，所有資料庫執行個體都會重新啟動。

如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。

- 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。當您將 APEX 選項新增至現有的資料庫執行個體時，在資料庫執行個體自動重新啟動時會發生短暫的中斷。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

解除鎖定公有使用者帳戶

安裝 Amazon RDS APEX 選項之後，請務必執行下列動作：

- 變更 APEX 公有使用者帳戶的密碼。
- 解除鎖定帳戶。

您可以使用 Oracle SQL*Plus 命令列公用程式來這麼做。以主要使用者身分連線至資料庫執行個體，然後發出以下命令。將 `new_password` 改為您選擇的密碼。

```
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;  
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

設定適用於 Oracle APEX 的 RESTful 服務

若要在 APEX 中設定 RESTful 服務 (APEX 4.1.1.V1 不需要)，請使用 SQL*Plus 以主要使用者身分連線至資料庫執行個體。完成此作業後，請執行 `rdsadmin.rdsadmin_run_apex_rest_config` 預存程序。執行預存程序時，您需要提供下列使用者的密碼：

- APEX_LISTENER
- APEX_REST_PUBLIC_USER

此預存程序會執行 `apex_rest_config.sql` 指令碼，而為這些使用者建立新的資料庫帳戶。

Note

若為 Oracle APEX 版本 4.1.1.v1，則不需要設定。僅針對此 Oracle APEX 版本，您不需要執行此預存程序。

下列命令會執行此預存程序。

```
EXEC rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',  
'apex_rest_public_user_password');
```

準備安裝 ORDS

在安裝 ORDS 之前，您必須先建立無特殊權限的作業系統使用者，然後下載並解壓縮 APEX 安裝檔案。

準備進行 ORDS 安裝

1. 以 `myapexhost.example.com` 身分登入 `root`。
2. 建立無特殊權限的作業系統使用者來負責接聽程式安裝。以下命令建立一個名為 `apexuser` 的新使用者。

```
useradd -d /home/apexuser apexuser
```

以下命令指派密碼給新使用者。

```
passwd apexuser;
```

3. 以 `myapexhost.example.com` 身分登入 `apexuser`，然後從 Oracle 下載 APEX 安裝檔案至您的 `/home/apexuser` 目錄：
 - <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
 - [Oracle Application Express 發行前封存](#)
4. 將檔案解壓縮在 `/home/apexuser` 目錄中。

```
unzip apex_version.zip
```

解壓縮檔案之後，`apex` 目錄中會有一個 `/home/apexuser` 目錄。

5. 在您仍以 `myapexhost.example.com` 身分登入 `apexuser` 時，從 Oracle 下載 Oracle REST Data Services 檔案至您的 `/home/apexuser` 目錄：<http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html>。

安裝和設定 ORDS 21 及更低版本

您現在可以安裝和設定 Oracle Rest Data Services (ORDS)，以便與 Oracle APEX 搭配使用。對於 APEX 5.0 及更新版本，請使用 ORDS 19.1 至 21 版本。若要瞭解如何安裝 ORDS 22 及更高版本，請參閱[安裝與設定 ORDS 22 及更新版本](#)。

將接聽程式安裝在另一部主機上，例如 Amazon EC2 執行個體、公司的現場部署伺服器或您的桌上型電腦。對於本節中的範例，我們假設您的主機名稱是 `myapexhost.example.com`，而且主機執行 Linux。

若要安裝和設定 ORDS 21 或更低版本，以便與甲骨文 APEX 搭配使用

1. 轉到[甲骨文 REST 數據服務](#)，並檢查自述文件。請確定您已安裝所需的 Java 版本。
2. 建立 ORDS 安裝的新目錄。

```
mkdir /home/apexuser/ORDS  
cd /home/apexuser/ORDS
```

3. 從 [Oracle REST 資料服務](#) 下載檔案 `ords.version.number.zip`。
4. 將檔案解壓縮在 `/home/apexuser/ORDS` 目錄中。
5. 如果要在多重租用資料庫中安裝 ORDS，請將下行新增至 `/home/apexuser/ORDS/params/ords_params.properties` 檔案：


```
pdb.disable.lockdown=false
```

6. 授與主要使用者安裝 ORDS 所需的權限。

安裝 Amazon RDS APEX 選項之後，請為主要使用者提供安裝 ORDS 結構描述所需的權限。您可以連線至資料庫並執行下列命令來完成此動作。以您的主要使用者的大寫名稱取代 *MASTER_USER*。

Important

輸入使用者名稱時，除非您使用區分大小寫的識別符來建立使用者，否則請使用大寫。例如，如果您執行 `CREATE USER myuser` 或 `CREATE USER MYUSER`，資料字典會存放 `MYUSER`。不過，如果您在 `CREATE USER "MyUser"` 中使用雙引號，資料字典則會存放 `MyUser`。如需更多詳細資訊，請參閱 [將 SELECT 或 EXECUTE 權限授予 SYS 物件](#)。

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER PROCEDURES', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
'EXECUTE', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',  
'EXECUTE', true);
```

Note

這些命令會套用至 ORDS 19.1 版和更新版本。

7. 使用下載的 ords.war 檔案來安裝 ORDS 結構描述。

```
java -jar ords.war install advanced
```

程式會提示您輸入下列資訊。方括號中是預設值。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle REST Data Services 簡介](#)。

- 輸入要存放組態資料的位置：

輸入 `/home/apexuser/ORDS`。這是 ORDS 組態檔案的位置。

- 指定要使用的資料庫連線類型。輸入 [1] 基本 [2] TNS [3] 自訂 URL [1] 的數字：

選擇所需的連線類型。

- 輸入資料庫伺服器的名稱 [localhost]：`DB_instance_endpoint`

選擇預設值或輸入正確值。

- 輸入資料庫接聽程式連接埠 [1521]：`DB_instance_port`

選擇預設值或輸入正確值。

- 輸入 1 以指定資料庫服務名稱，或輸入 2 以指定資料庫 SID [1]：

選擇 2 以指定資料庫 SID。

- 資料庫 SID [xe]

選擇預設值或輸入正確值。

- 若要驗證/安裝 Oracle REST Data Services 結構描述，請輸入 1，或輸入 2 以略過此步驟 [1]：

選擇 1。此步驟會建立名為 ORDS_PUBLIC_USER 的 Oracle REST Data Services 代理使用者。

- 輸入 ORDS_PUBLIC_USER 的資料庫密碼：


輸入密碼，然後確認。

- 需要使用管理員權限登入，以驗證 Oracle REST Data Services 結構描述。

輸入管理員使用者名稱：*master_user*

輸入 *master_user* 的資料庫密碼：*master_user_password*

確認密碼：*master_user_password*

 Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

- 輸入 ORDS_METADATA [SYSAUX] 的預設資料表空間。

輸入 ORDS_METADATA [TEMP] 的暫時資料表空間。

輸入 ORDS_PUBLIC_USER [USERS] 的預設資料表空間。

輸入 ORDS_PUBLIC_USER [TEMP] 的暫時資料表空間。

- 如果要使用 PL/SQL 閘道請輸入 1，或輸入 2 來略過此步驟。如果您使用 Oracle Application Express 或從 mod_plsql 遷移，則必須輸入 1 [1]。

選擇預設值。

- 輸入 PL/SQL Gateway 資料庫使用者名稱 [APEX_PUBLIC_USER]

選擇預設值。

- 輸入 APEX_PUBLIC_USER 的資料庫密碼：

輸入密碼，然後確認。

- 輸入 1 以指定 Application Express RESTful Services 資料庫使用者 (APEX_LISTENER, APEX_REST_PUBLIC_USER) 的密碼，或輸入 2 以略過此步驟 [1]：

若為 APEX 4.1.1.V1，請選擇 2；至於其他所有 APEX 版本，請選擇 1。

- [APEX 4.1.1.v1 不需要此步驟] APEX_LISTENER 的資料庫密碼

輸入密碼 (如需要)，然後確認。

- [APEX 4.1.1.v1 不需要此步驟] APEX_REST_PUBLIC_USER 的資料庫密碼

輸入密碼 (如需要)，然後確認。

- 輸入數字以選取要啟用的功能：

輸入 1 以啟用所有功能：SQL Developer Web、REST Enabled SQL 和 Database API。

- 若要以獨立模式啟動，請輸入 1，或輸入 2 以結束 [1]：

輸入 1。

- 輸入 APEX 靜態資源位置：

如果您將 APEX 安裝檔案解壓縮至 `/home/apexuser`，請輸入 `/home/apexuser/apex/images`。否則，請輸入 `unzip_path/apex/images`，其中 `unzip_path` 是您將檔案解壓縮的目錄所在位置。

- 若使用 HTTP，請輸入 1；或若使用 HTTPS [1]，則請輸入 2：

如果您輸入 1，請指定 HTTP 連接埠。如果您輸入 2，請指定 HTTPS 連接埠和 SSL 主機名稱。HTTPS 選項會提示您指定提供憑證的方式：

- 輸入 1 以使用自我簽署的憑證。
- 輸入 2 以提供您自己的憑證。如果您輸入 2，請指定 SSL 憑證的路徑，以及 SSL 憑證私密金鑰的路徑。

8. 設定 APEX admin 使用者的密碼。若要這麼做，請使用 SQL*Plus 以主要使用者身分連線至資料庫執行個體，然後執行以下命令。

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;  
grant APEX_ADMINISTRATOR_ROLE to master;  
@/home/apexuser/apex/apxchpwd.sql
```

將 `master` 改為您的主要使用者名稱。當 `apxchpwd.sql` 指令碼提示您時，請輸入新的 admin 密碼。

9. 啟動 ORDS 接聽程式。執行下列程式碼。

```
java -jar ords.war
```

第一次啟動 ORDS 時會提示您提供 APEX 靜態資源的位置。此影像資料夾位於 APEX 安裝目錄的 `/apex/images` 目錄中。

10. 在瀏覽器中返回 APEX 管理視窗，然後選擇 Administration (管理)。接著選擇 Application Express Internal Administration (Application Express 內部管理)。當提示您提供登入資料時，請輸入下列資訊：

- 使用者名稱 – admin
- 密碼 – 使用 `apxchpwd.sql` 指令碼所設定的密碼

選擇 Login (登入)，然後設定 admin 使用者的新密碼。

您的接聽程式現在可供使用。

安裝與設定 ORDS 22 及更新版本

您現在可以安裝和設定 Oracle Rest Data Services (ORDS)，以便與 Oracle APEX 搭配使用。ORDS 22 的說明與舊版的說明不同。

若要安裝和設定 ORDS 22 及更新版本，以便與甲骨文 APEX 搭配使用

1. 移至 [Oracle REST 資料服務](#)，並檢查您計劃下載之 ORDS 版本的讀我檔案。請確定您已安裝所需的 Java 版本。
2. 建立 ORDS 安裝的新目錄。

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

3. 下載檔案 `ords.version.number.zip` 或 `ords-latest.zip` 從 [Oracle REST 資料服務](#) 下載。
4. 將檔案解壓縮在 `/home/apexuser/ORDS` 目錄中。
5. 授與主要使用者安裝 ORDS 所需的權限。

安裝 Amazon RDS APEX 選項之後，請為主要使用者提供安裝 ORDS 結構描述所需的權限。您可以通過登錄到數據庫並運行以下命令來做到這一點。以您的主要使用者的大寫名稱取代 `MASTER_USER`。

Important

輸入使用者名稱時，除非您使用區分大小寫的識別符來建立使用者，否則請使用大寫。例如，如果您執行 `CREATE USER myuser` 或 `CREATE USER MYUSER`，資料字典會存放

MYUSER。不過，如果您在 CREATE USER "MyUser" 中使用雙引號，資料字典則會存放 MyUser。如需詳細資訊，請參閱 [將 SELECT 或 EXECUTE 權限授予 SYS 物件](#)。

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
'EXECUTE', true);

exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOB', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_ASSERT', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_OUTPUT', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SCHEDULER', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('HTP', 'MASTER_USER', 'EXECUTE',
true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('OWA', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPG_DOCLoad', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPT0', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_METADATA', 'MASTER_USER',
'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SQL', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('UTL SMTP', 'MASTER_USER', 'EXECUTE',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_NETWORK_ACL_ADMIN',
'MASTER_USER', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('SESSION_PRIVS', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_USERS', 'MASTER_USER', 'SELECT',
true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACL_PRIVILEGES',
'MASTER_USER', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACLs', 'MASTER_USER',
'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'MASTER_USER',
'SELECT', true);
```

Note

上述命令適用於 ORDS 22 及更高版本。

6. 使用下載的 ords 指令碼安裝 ORDS 結構描述。指定要包含組態檔和記錄檔的目錄。甲骨文建議不要將這些目錄放在包含 ORDS 產品軟體的目錄中。

```
mkdir -p /home/apexuser/ords_config /home/apexuser/ords_logs

/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs
```

對於執行容器資料庫 (CDB) 架構的資料庫執行個體，請使用 ORDS 23.2 及更高版本，並在安裝 ORDS 時傳遞 `--pdb-skip-disable-lockdown` 引數。

```
/home/apexuser/ORDS/bin/ords \  
  --config /home/apexuser/ords_config \  
  install --interactive --log-folder /home/apexuser/ords_logs --pdb-skip-disable-  
lockdown
```

程式會提示您輸入下列資訊。方括號中是預設值。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle REST Data Services 簡介](#)。

- Choose the type of installation:

選擇**2**在資料庫中安裝 ORDS 結構描述，並在本機 ORDS 組態檔中建立資料庫連線集區。

- Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL:

選擇所需的連線類型。此範例假設您選擇**1**。

- Enter the name of the database server [localhost]:

DB_instance_endpoint

選擇預設值或輸入正確值。

- Enter the database listener port [1521]: ***DB_instance_port***

選擇預設值**1521**或輸入正確的值。

- Enter the database service name [orcl]:

輸入適用於 Oracle 資料庫執行個體的 RDS 所使用的資料庫名稱。

- Provide database user name with administrator privileges

輸入適用於 Oracle 資料庫執行個體之 RDS 的主要使用者名稱。

- Enter the database password for [username]:

輸入適用於 Oracle 資料庫執行個體之 RDS 的主要使用者密碼。

- Enter the default tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [SYSAUX]:

- Enter the temporary tablespace for ORDS_METADATA [TEMP]. Enter the default tablespace for ORDS_PUBLIC_USER [USERS]. Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP].

- ~~Enter a number to select additional feature(s) to enable [1]:~~

- Enter a number to configure and start ORDS in standalone mode [1]:

選擇2以獨立模式立即略過啟動 ORDS。

- Enter a number to select the protocol [1] HTTP
- Enter the HTTP port [8080]:
- Enter the APEX static resources location:

輸入 APEX 安裝檔案的路徑 (/home/apexuser/apex/images)。

7. 設定 APEX admin 使用者的密碼。若要這麼做，請使用 SQL*Plus 以主要使用者身分連線至資料庫執行個體，然後執行以下命令。

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

將 *master* 改為您的主要使用者名稱。當 apxchpwd.sql 指令碼提示您時，請輸入新的 admin 密碼。

8. 使用指令ords碼搭配serve指令碼，以獨立模式執行 ORDS。對於生產部署，請考慮使用支持的 Java EE 應用程式服務器，如 Apache Tomcat 或甲骨文 WebLogic服務器。如需詳細資訊，請參閱 [Oracle 資料庫說明文件中的建置和監督 Oracle REST 資料服務](#)。

```
/home/apexuser/ORDS/bin/ords \  
  --config /home/apexuser/ords_config serve \  
  --port 8193 \  
  --apex-images /home/apexuser/apex/images
```

如果 ORDS 正在執行，但無法存取 APEX 安裝，您可能會看到下列錯誤，特別是在非 CDB 執行個體上。

```
The procedure named apex_admin could not be accessed, it may not be declared,  
or the user executing this request may not have been granted execute privilege  
on the procedure, or a function specified by security.requestValidationFunction  
configuration property has prevented access.
```

若要修正此錯誤，請使用命令執行config指令ords碼，變更 ORDS 使用的要求驗證函式。默認情況下，ORDS 使用該過程ords_util.authorize_plsql_gateway程，該過程僅在 CDB 實例上受支持。對於非 CDB 執行個體，您可以將此程序變更為套

件 `wwv_flow_epg_include_modules.authorize`。如需針對您的使用案例設定適當要求驗證功能的最佳作法，請參閱 Oracle 資料庫說明文件和 Oracle Support 部門。

9. 在瀏覽器中返回 APEX 管理視窗，然後選擇 Administration (管理)。接著選擇 Application Express Internal Administration (Application Express 內部管理)。當提示您提供登入資料時，請輸入下列資訊：

- 使用者名稱 – admin
- 密碼 – 使用 `apxchpwd.sql` 指令碼所設定的密碼

選擇 Login (登入)，然後設定 admin 使用者的新密碼。

您的接聽程式現在可供使用。

設定 Oracle APEX Listener

Note

Oracle APEX Listener 已淘汰。

Amazon RDS for Oracle 繼續支援 APEX 4.1.1 版和 Oracle APEX Listener 1.1.4 版。我們建議您使用最新支援的 Oracle APEX 和 ORDS 版本。

將 Oracle APEX Listener 安裝在另一部主機上，例如 Amazon EC2 執行個體、公司的現場部署伺服器或您的桌上型電腦。我們假設主機名稱為 `myapexhost.example.com`，且主機執行 Linux。

準備安裝 Oracle APEX Listener

在安裝 Oracle APEX Listener 之前，您必須先建立無特殊權限的作業系統使用者，然後下載並解壓縮 APEX 安裝檔案。

準備安裝 Oracle APEX Listener

1. 以 `myapexhost.example.com` 身分登入 `root`。
2. 建立無特殊權限的作業系統使用者來負責接聽程式安裝。以下命令建立一個名為 `apexuser` 的新使用者。

```
useradd -d /home/apexuser apexuser
```

以下命令指派密碼給新使用者。

```
passwd apexuser;
```

3. 以 `myapexhost.example.com` 身分登入 `apexuser`，然後從 Oracle 下載 APEX 安裝檔案至您的 `/home/apexuser` 目錄：

- <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
- [Oracle Application Express 發行前封存](#)

4. 將檔案解壓縮在 `/home/apexuser` 目錄中。

```
unzip apex_<version>.zip
```

解壓縮檔案之後，`apex` 目錄中會有一個 `/home/apexuser` 目錄。

5. 在您仍以 `myapexhost.example.com` 身分登入 `apexuser` 時，從 Oracle 下載 Oracle APEX Listener 檔案至您的 `/home/apexuser` 目錄。

安裝與設定 Oracle APEX Listener

在可以使用 APEX 之前，您需先下載 `apex.war` 檔案、使用 Java 安裝 Oracle APEX Listener，然後啟動接聽程式。

安裝與設定 Oracle APEX Listener

1. 根據 Oracle APEX Listener 建立新目錄，並開啟接聽程式檔案。

執行下列程式碼：

```
mkdir /home/apexuser/apexlistener  
cd /home/apexuser/apexlistener  
unzip ../apex_listener.version.zip
```

2. 執行下列程式碼。

```
java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -  
jar ./apex.war
```

3. 輸入資訊讓程式提示下列：

- APEX Listener Administrator 使用者名稱。預設值為 adminlistener。
- APEX Listener Administrator 的密碼。
- APEX Listener Manager 使用者名稱。預設值為 managerlistener。
- APEX Listener Administrator 的密碼。

程式會印出您完成組態所需的 URL，如下所示。

```
INFO: Please complete configuration at: http://localhost:8080/apex/  
listenerConfigure  
Database is not yet configured
```

4. 讓 Oracle APEX Listener 保持執行，使得您可以使用 Oracle Application Express。當您完成此設定程序時，就可以換成在背景中執行接聽程式。
5. 在 web 瀏覽器中，移至 APEX Listener 程式所提供的 URL。Oracle Application Express Listener 管理視窗隨即出現。輸入下列資訊：
 - 使用者名稱 – APEX_PUBLIC_USER
 - 密碼 – APEX_PUBLIC_USER 的密碼。這是您稍早設定 APEX 儲存庫時所指定的密碼。如需更多詳細資訊，請參閱 [解除鎖定公有使用者帳戶](#)。
 - 連線類型 – 基本
 - 主機名稱 – Amazon RDS 資料庫執行個體的端點，例如 mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com。
 - 連接埠 – 1521
 - SID – Amazon RDS 資料庫執行個體上的資料庫名稱，例如 mydb。
6. 選擇 Apply (套用)。APEX 管理視窗隨即出現。
7. 設定 APEX admin 使用者的密碼。若要這麼做，請使用 SQL*Plus 以主要使用者身分連線至資料庫執行個體，然後執行以下命令。

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;  
grant APEX_ADMINISTRATOR_ROLE to master;  
@/home/apexuser/apex/apxchpwd.sql
```

將 *master* 改為您的主要使用者名稱。當 apxchpwd.sql 指令碼提示您時，請輸入新的 admin 密碼。

8. 在瀏覽器中返回 APEX 管理視窗，然後選擇 Administration (管理)。接著選擇 Application Express Internal Administration (Application Express 內部管理)。當提示您提供登入資料時，請輸入下列資訊：

- 使用者名稱 – admin
- 密碼 – 使用 apxchpwd.sql 指令碼所設定的密碼

選擇 Login (登入)，然後設定 admin 使用者的新密碼。

您的接聽程式現在可供使用。

升級 APEX 版本

Important

升級 APEX 之前，請備份資料庫執行個體。如需更多詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#) 及 [測試 Oracle 資料庫升級](#)。

若要連同資料庫執行個體一起升級 APEX，請執行下列動作：

- 為資料庫執行個體的升級版本建立新的選項群組。
- 將 APEX 和 APEX-DEV 的升級版本新增至新的選項群組。務必包括資料庫執行個體使用的其他任何選項。如需更多詳細資訊，請參閱 [選項群組考量](#)。
- 升級資料庫執行個體時，請為升級的資料庫執行個體指定新的選項群組。

升級 APEX 的版本之後，舊版的 APEX 結構描述可能仍存在於資料庫中。如果不再需要，您可以在升級之後從資料庫中刪除舊的 APEX 結構描述。

如果您升級 APEX 版本，但舊版 APEX 中未設定 RESTful 服務，建議您設定 RESTful 服務。如需更多詳細資訊，請參閱 [設定適用於 Oracle APEX 的 RESTful 服務](#)。

在某些情況下，當您打算升級資料庫執行個體的主要版本時，您可能會發現使用的 APEX 版本與目標資料庫版本不相容。在這類情況下，您可先升級 APEX 版本，再升級資料庫執行個體。先升級 APEX 可以縮短升級資料庫執行個體所需的時間。

Note

在升級 APEX 後，請安裝並設定接聽程式以使用升級後的版本。如需說明，請參閱「[設定 Oracle APEX Listener](#)」。

移除 APEX 選項

您可以從資料庫執行個體中移除 Amazon RDS APEX 選項。若要從一個資料庫執行個體中移除 APEX 選項，請執行下列其中一個動作：

- 若要從多個資料庫執行個體中移除 APEX 選項，請將 APEX 選項從所屬的選項群組中移除。此變更會影響使用選項群組的所有資料庫執行個體。當您從附加至多個資料庫執行個體的選項群組中移除 APEX 選項時，在所有資料庫執行個體都重新啟動期間，會發生短暫停機。

如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。

- 若要從單一資料庫執行個體中移除 APEX 選項，請修改資料庫執行個體，並指定不包括 APEX 選項的不同選項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。當您移除 APEX 選項時，因為資料庫執行個體會自動重新啟動，所以會發生短暫停機。

如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

當您從資料庫執行個體中移除 APEX 選項時，APEX 結構描述也會從資料庫中移除。

Amazon EFS 整合

Amazon Elastic File System (Amazon EFS) 提供無伺服器、完全彈性的檔案儲存功能，讓您無需佈建或管理儲存容量和效能，即可分享檔案資料。使用 Amazon EFS，您可以建立檔案系統，然後透過 NFS 4.0 和 4.1 版 (NFSv4) 通訊協定將其掛載到虛擬私人雲端中。然後，您可以像使用任何其他 POSIX 相容檔案系統一樣使用 EFS 檔案系統。如需一般資訊，請參閱[什麼是 Amazon 彈性檔案系統？](#)和 AWS 部落格文章[整合 Amazon RDS for Oracle 與 Amazon EFS](#)。

主題

- [Amazon EFS 整合概觀](#)
- [設定網路許可以進行 RDS for Oracle 與 Amazon EFS 的整合](#)
- [設定 IAM 許可以進行 RDS for Oracle 與 Amazon EFS 的整合](#)
- [新增 EFS_INTEGRATION 選項](#)
- [設定 Amazon EFS 檔案系統許可](#)
- [在 RDS for Oracle 與 Amazon EFS 檔案系統之間傳輸檔案](#)
- [移除 EFS_INTEGRATION 選項](#)
- [針對 Amazon EFS 整合進行疑難排解](#)

Amazon EFS 整合概觀

透過 Amazon EFS，您可以在 RDS for Oracle 資料庫執行個體與 Amazon EFS 檔案系統之間傳輸檔案。例如，您可以使用 EFS 支援下列使用案例：

- 在應用程式和多個資料庫伺服器之間共用檔案系統。
- 建立共用目錄，用於與移轉任務相關的檔案 (包括可傳輸的資料表空間資料檔案)。如需詳細資訊，請參閱 [使用 Oracle 可傳輸資料表空間進行移轉](#)。
- 儲存和共用已封存的重做日誌檔，無需在伺服器上額外配置儲存空間。
- 使用 Oracle 資料庫公用程式 (例如 UTL_FILE) 來讀取和寫入檔案。

Amazon EFS 整合的優勢

選擇 EFS 檔案系統而非其他資料傳輸解決方案的效益如下：

- 您可以在 RDS for Oracle 資料庫執行個體與 Amazon EFS 檔案系統之間傳輸檔案。您不需要在本機複製這些檔案，因為 Data Pump 會直接從 EFS 檔案系統匯入。如需詳細資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle](#)。
- 相較於使用資料庫連結，資料移轉速度更快。
- 您可以避免在 RDS for Oracle 資料庫執行個體上配置儲存空間來儲存檔案。
- EFS 檔案系統可以自動擴充儲存空間，您無需動手佈建。
- Amazon EFS 整合沒有最低費用或設定成本。您僅需按實際用量付費。
- Amazon EFS 整合支援兩種加密形式：加密傳輸中的資料和靜態加密。依預設，使用 TLS 1.2 版啟用傳輸中資料的加密功能。您可在 Amazon EFS 檔案系統建立時啟用靜態資料加密。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的 [靜態資料加密](#)。

Amazon EFS 整合的需求

請確定您符合下列需求：

- 您的資料庫執行的資料庫版本為 19.0.0.0.ru-2022-07.rur-2022-07.r1 或更新版本。
- 您的資料庫執行個體和 EFS 檔案系統位於相同 AWS 區域、VPC 和 AWS 帳戶。適用於 Oracle 的 RDS 不支援 EFS 的跨帳戶和跨區域存取。
- 您的 VPC 已啟用 enableDnsSupport 屬性。如需詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPC 中的 DNS 屬性](#)。
- 您的 EFS 檔案系統使用標準或標準 IA 儲存體類別。
- 若要在 mount 命令中使用 DNS 名稱，必須符合下列條件：
 - 連線資料庫執行個體位於 VPC 內，且設定為使用 Amazon 提供的 DNS 伺服器。不支援自訂 DNS 伺服器。
 - 連線執行個體的 VPC 必須同時已啟用 DNS 解析和 DNS 主機名稱。
 - 連線執行個體必須與 EFS 檔案系統位在相同的 VPC 內。
- 您可以使用非 RDS 解決方案來備份 EFS 檔案系統。RDS for Oracle 不支援 EFS 檔案系統的自動備份或手動資料庫快照。如需詳細資訊，請參閱 [備份您的 Amazon EFS 檔案系統](#)。

設定網路許可以進行 RDS for Oracle 與 Amazon EFS 的整合

若要讓 RDS for Oracle 與 Amazon EFS 整合，請確定您的資料庫執行個體具有 EFS 檔案系統的網路存取權。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的 [控制 NFS 用戶端對 Amazon EFS 檔案系統的網路存取](#)。

主題

- [使用安全群組控制網路存取](#)
- [使用檔案系統政策控制網路存取](#)

使用安全群組控制網路存取

您可以使用網路層安全機制 (例如 VPC 安全群組) 控制資料庫執行個體對 EFS 檔案系統的存取。若要允許資料庫執行個體存取 EFS 檔案系統，請確定您的 EFS 檔案系統符合下列需求：

- EFS 掛載目標存在於 RDS for Oracle 資料庫執行個體使用的每個可用區域中。

EFS 掛載目標會提供 NFSv4 端點的 IP 地址，您可以在該端點上掛載 EFS 檔案系統。您將使用其 DNS 名稱掛載您的檔案系統，該名稱會解析為 EFS 掛載目標 IP 地址，而該目標位於資料庫執行個體所使用的可用區域中。

您可以設定不同可用區域中的資料庫執行個體，以使用相同的 EFS 檔案系統。針對多可用區，您需要部署中每個可用區域的掛載點。您可能需要將資料庫執行個體移動至不同的可用區域。基於這些原因，建議您在 VPC 中的每個可用區域建立 EFS 掛載點。根據預設，當您使用主控台建立新的 EFS 檔案系統時，RDS 會為所有可用區域建立掛載目標。

- 安全群組連接至掛載目標。
- 安全群組具有傳入規則，允許在 TCP/2049 (NFS 類型) 上 RDS for Oracle 資料庫執行個體的網路子網路或安全群組。

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[建立 Amazon EFS 檔案系統](#)和[建立和管理 EFS 掛載目標和安全群組](#)。

使用檔案系統政策控制網路存取

Amazon EFS 與 RDS for Oracle 整合會使用預設 (空白) EFS 檔案系統政策。預設政策不會使用 IAM 進行身分驗證。相反地，它會將完全存取權授予任何可以使用掛載目標連線至檔案系統的匿名用戶端。每當使用者設定的檔案系統政策未生效時，預設政策就會生效，包括在建立檔案系統時。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[預設 EFS 檔案系統政策](#)。

若要加強所有用戶端 (包括 RDS for Oracle) 對 EFS 檔案系統的存取，您可以設定 IAM 許可。在此方法中，您可以建立檔案系統政策。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[建立檔案系統政策](#)。

設定 IAM 許可進行 RDS for Oracle 與 Amazon EFS 的整合

根據預設，Amazon EFS 整合功能不使用 IAM 角色：選 `USE_IAM_ROLE` 項設定為 `FALSE`。若要將 RDS 版 Oracle 與 Amazon EFS 和 IAM 角色整合，您的資料庫執行個體必須具有 IAM 許可才能存取 Amazon EFS 檔案系統。

主題

- [步驟 1：針對您的資料庫執行個體建立 IAM 角色並附加您的政策](#)
- [步驟 2：針對您的 Amazon EFS 檔案系統建立檔案系統政策](#)
- [步驟 3：將 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯](#)

步驟 1：針對您的資料庫執行個體建立 IAM 角色並附加您的政策

在此步驟中，您會針對 RDS for Oracle 資料庫執行個體建立一個角色，允許 Amazon RDS 存取您的 EFS 檔案系統。

主控台

建立 IAM 角色以允許 Amazon RDS 存取 EFS 檔案系統

1. 開啟 [IAM 管理主控台](#)。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 針對 AWS 服務，請選擇 RDS。
5. 針對 Select your use case (選擇使用案例) 選擇 RDS – Add Role to Database (新增角色至資料庫)。
6. 選擇下一步。
7. 請不要新增任何許可政策。選擇下一步。
8. 將 Role Name (角色名稱) 設為您的 IAM 角色名稱，例如 `rds-efs-integration-role`。您也可以新增選用的 Description (描述) 值。
9. 選擇建立角色。

AWS CLI

若要將服務的許可限制為特定資源，建議您在資源型信任關係中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰。這是防止 [混淆代理人問題](#) 最有效的方式。

您可以同時使用全域條件內容索引鍵和包含帳號 ID 的 `aws:SourceArn` 值。在此情況下，當在相同陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 裡的帳戶時，兩者必須使用同樣的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在信任關係中，請務必使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含存取角色的資源之完整 Amazon 資源名稱 (ARN)。

下列 AWS CLI 命令會建立 `rds-efs-integration-role` 為此目的命名的角色。

Example

對於 Linux macOS、或 Unix：

```
aws iam create-role \  
  --role-name rds-efs-integration-role \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",  
        "Condition": {  
          "StringEquals": {  
            "aws:SourceAccount": my_account_ID,  
            "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"  
          }  
        }  
      }  
    ]  
  }'
```

在 Windows 中：

```
aws iam create-role ^  
  --role-name rds-efs-integration-role ^
```

```
--assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": my_account_ID,
          "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
        }
      }
    }
  ]
}'
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

步驟 2：針對您的 Amazon EFS 檔案系統建立檔案系統政策

在此步驟中，您會針對 EFS 檔案系統建立檔案系統政策。

建立或編輯 EFS 檔案系統政策

1. 開啟 [EFS 管理主控台](#)。
2. 選擇 File Systems (檔案系統)。
3. 在 File systems (檔案系統) 頁面上，選擇您要編輯或針對其建立檔案系統政策的檔案系統。即會顯示該檔案系統的詳細資訊頁面。
4. 選擇 File system policy (檔案系統政策) 索引標籤。

如果政策空白，表示正在使用預設 EFS 檔案系統政策。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[預設 EFS 檔案系統政策](#)。

5. 選擇編輯。File system policy (檔案系統政策) 頁面隨即顯示。
6. 在 Policy editor (原則編輯器) 中，輸入如下的政策，然後選擇 Save (儲存)。

```
{
  "Version": "2012-10-17",
```

```
"Id": "ExamplePolicy01",
"Statement": [
  {
    "Sid": "ExampleStatement01",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/rds-efs-integration-role"
    },
    "Action": [
      "elasticfilesystem:ClientMount",
      "elasticfilesystem:ClientWrite",
      "elasticfilesystem:ClientRootAccess"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:123456789012:file-
system/fs-1234567890abcdef0"
  }
]
```

步驟 3：將 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯

在此步驟中，您會將 IAM 角色與資料庫執行個體建立關聯。請注意下列需求：

- 您必須有權存取必要 Amazon EFS 許可政策附加至其中的 IAM 角色。
- 您一次只能將一個 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯。
- 執行個體的狀態必須為 Available (可用)。

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[適用於 Amazon EFS 的 Identity and Access Management](#)。

主控台

如要建立 IAM 角色與 RDS for Oracle 資料庫執行個體的關聯

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 Databases (資料庫)。
3. 如果您的資料庫執行個體無法使用，請選擇 Actions (動作)，然後選擇 Start (開始)。當執行個體狀態顯示 Started (已啟動) 時，請移至下一個步驟。

4. 選擇 Oracle 資料庫執行個體名稱以顯示其詳細資訊。
5. 在 Connectivity & security (連線和安全) 索引標籤上，向下捲動至頁面底部的 Manage IAM roles (管理 IAM 角色) 區段。
6. 在 Add IAM roles to this instance (新增 IAM 角色到此執行個體) 區段中選擇要新增的角色。
7. 針對 Feature (功能)，選擇 EFS_INTEGRATION。
8. 選擇 Add role (新增角色)。

AWS CLI

下列 AWS CLI 命令會將角色新增至名為的 Oracle 資料庫執行個體 *mydbinstance*。

Example

對於 Linux macOS、或 Unix：

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier mydbinstance \  
  --feature-name EFS_INTEGRATION \  
  --role-arn your-role-arn
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --feature-name EFS_INTEGRATION ^  
  --role-arn your-role-arn
```

將 *your-role-arn* 替換為您前個步驟記下的角色 ARN。EFS_INTEGRATION 必須在 --feature-name 選項中指定。

新增 EFS_INTEGRATION 選項

若要整合 Amazon RDS for Oracle 與 Amazon EFS，您的資料庫執行個體必須與包含 EFS_INTEGRATION 選項的選項群組建立關聯。

屬於相同選項群組的多個 Oracle 資料庫執行個體共用相同的 EFS 檔案系統。不同的資料庫執行個體可以存取相同的資料，但存取可以透過使用不同的 Oracle 目錄來進行劃分。如需更多資訊，請參閱在 [RDS for Oracle 與 Amazon EFS 檔案系統之間傳輸檔案](#)。

主控台

設定選項群組進行 Amazon EFS 整合

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 EFS_INTEGRATION 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 EFS_INTEGRATION 選項新增至選項群組。您需要指定 EFS_ID 檔案系統 ID 並設定 USE_IAM_ROLE 旗標。

如需詳細資訊，請參閱 [將選項新增至選項群組](#)。

3. 請採取下其中一種方式將選項群組與您的資料庫執行個體建立關聯：

- 建立新的 Oracle 資料庫執行個體，並將選項群組與其建立關聯。如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改 Oracle 資料庫執行個體，以將選項群組與其建立關聯。如需修改 Oracle 資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

設定選項群組進行 EFS 整合

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 EFS_INTEGRATION 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 EFS_INTEGRATION 選項新增至選項群組。

例如，下列 AWS CLI 指令會將 EFS_INTEGRATION 選項新增至名為的選項群組 **myoptiongroup**。

Example

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name myoptiongroup \  
  --options "OptionName=EFS_INTEGRATION,OptionSettings=\  
  [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
```

在 Windows 中：

```
aws rds add-option-to-option-group ^
  --option-group-name myoptiongroup ^
  --options "OptionName=EFS_INTEGRATION,OptionSettings=^
  [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
```

3. 請採取下其中一種方式將選項群組與您的資料庫執行個體建立關聯：

- 建立新的 Oracle 資料庫執行個體，並將選項群組與其建立關聯。如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 修改 Oracle 資料庫執行個體，以將選項群組與其建立關聯。如需修改 Oracle 資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

設定 Amazon EFS 檔案系統許可

根據預設，只有根使用者 (UID 0) 對新建立的 EFS 檔案系統具備讀取、寫入和執行許可。若要讓其他使用者能夠修改檔案系統，根使用者必須明確地授予存取權給這些使用者。RDS for Oracle 資料庫執行個體的使用者位於 `others` 類別中。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的 [使用網路檔案系統 \(NFS\) 層級的使用者、群組和許可](#)。

若要允許 RDS for Oracle 資料庫執行個體讀取和寫入 EFS 檔案系統上的檔案，請執行下動作：

- 在本機於 Amazon EC2 或內部部署執行個體上掛載 EFS 檔案系統。
- 設定精細許可。

例如，若要授予 `other` 使用者寫入 EFS 檔案系統根目錄的許可，請在此目錄上執行 `chmod 777`。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的 [範例 Amazon EFS 檔案系統使用案例和許可](#)。

在 RDS for Oracle 與 Amazon EFS 檔案系統之間傳輸檔案

若要在 RDS for Oracle 執行個體與 Amazon EFS 檔案系統之間傳輸檔案，請至少建立一個 Oracle 目錄，然後設定 EFS 檔案系統許可，以控制資料庫執行個體存取。

主題

- [建立 Oracle 目錄](#)
- [在 EFS 檔案系統之間傳輸資料：範例](#)

建立 Oracle 目錄

若要建立 Oracle 目錄，請使用程序 `rdsadmin.rdsadmin_util.create_directory_efs`。程序具有下列參數。

參數名稱	資料類型	預設	必要	描述
<code>p_directory_name</code>	VARCHAR	–	是	Oracle 目錄的名稱。
<code>p_path_on_efs</code>	VARCHAR	–	是	<p>EFS 檔案系統上的路徑。路徑名稱的字首會使用模式 <code>/rdsefs-<i>fsid</i>/</code>，其中 <i>fsid</i> 是 EFS 檔案系統 ID 的預留位置。</p> <p>例如，如果您的 EFS 檔案系統已命名為 <code>fs-1234567890abcdef0</code>，且您在名為 <code>mydir</code> 的這個檔案系統上建立了一個子目錄，則您可以指定下列值：</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block;">/rdsefs-fs-1234567890abcdef0/mydir</pre>

假設您在 EFS 檔案系統 `fs-1234567890abcdef0` 上建立了名為 `/datapump1` 的子目錄。下列範例會建立一個 Oracle 目錄 `DATA_PUMP_DIR_EFS`，指向 EFS 檔案系統上的 `/datapump1` 目錄。`p_path_on_efs` 參數的檔案系統路徑值以字串 `/rdsefs-` 為字首。

```
BEGIN
  rdsadmin.rdsadmin_util.create_directory_efs(
    p_directory_name => 'DATA_PUMP_DIR_EFS',
    p_path_on_efs    => '/rdsefs-fs-1234567890abcdef0/datapump1');
END;
/
```

在 EFS 檔案系統之間傳輸資料：範例

下列範例使用 Oracle Data Pump 將名為 `MY_TABLE` 的資料表匯出至檔案 `datapump.dmp`。此檔案位於 EFS 檔案系統上。

```
DECLARE
```

```
v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'TABLE',
job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdn1,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdn1,
    filename  => 'datapump-exp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdn1,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/
```

下列範例使用 Oracle Data Pump，從檔案 datapump.dmp 中匯入名為 MY_TABLE 的資料表。此檔案位於 EFS 檔案系統上。

```
DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'TABLE',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdn1,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdn1,
    filename  => 'datapump-imp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdn1,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/
```

如需詳細資訊，請參閱 [將資料匯入 Amazon RDS 上的 Oracle](#)。

移除 EFS_INTEGRATION 選項

移除 EFS_INTEGRATION 選項的步驟取決於您要從多個資料庫執行個體或單一執行個體移除選項。

資料庫執行個體的數目	動作	相關資訊
多個	從資料庫執行個體所屬的選項群組中移除選項。EFS_INTEGRATION 此變更會影響使用選項群組的所有執行個體。	從選項群組移除選項
單一	修改資料庫執行個體，並指定不包括 EFS_INTEGRATION 選項的不同選項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。	修改 Amazon RDS 資料庫執行個體

移除此 EFS_INTEGRATION 選項後，您可以選擇性地刪除連線到資料庫執行個體的 EFS 檔案系統。

針對 Amazon EFS 整合進行疑難排解

您的 RDS for Oracle 資料庫執行個體會監控與 EFS 檔案系統的連線。當監控偵測到問題時，它可能會嘗試修正該問題，並在 RDS 主控台中發佈事件。如需詳細資訊，請參閱 [檢視 Amazon RDS 事件](#)。

請使用此節中的資訊，協助您診斷並修正在您使用 Amazon EFS 整合時常見的問題。

通知	描述	動作
The EFS for RDS Oracle instance <i>instance_name</i> isn't available on the primary host. NFS port 2049 of your EFS isn't reachable.	資料庫執行個體無法與 EFS 檔案系統通訊。	請確定下列情況： <ul style="list-style-type: none"> EFS 檔案系統存在。 連接至 EFS 掛載目標的安全群組具有傳入規則，允許在 TCP/2049 (NFS 類型) 上 RDS for Oracle 資料庫執行個體的安全群組或網路子網路。

通知	描述	動作
The EFS isn't reachable.	安裝 EFS_INTEGRATION 選項期間發生錯誤。	<p>請確定下列情況：</p> <ul style="list-style-type: none"> • EFS 檔案系統存在。 • 連接至 EFS 掛載目標的安全群組具有傳入規則，允許在 TCP/2049 (NFS 類型) 上 RDS for Oracle 資料庫執行個體的安全群組或網路子網路。 • 已針對您的 VPC 開啟 enableDnsSupport 屬性。 • 您正在 VPC 中使用 Amazon 提供的 DNS 伺服器。Amazon EFS 整合無法使用自訂 DHCP DNS。
The associated role with your DB instance wasn't found.	安裝 EFS_INTEGRATION 選項期間發生錯誤。	請確定您已將 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯。
The associated role with your DB instance wasn't found.	安裝 EFS_INTEGRATION 選項期間發生錯誤。適用於 Oracle 的 RDS 已從資料庫快照集還原，且選 USE_IAM_ROLE 項設定為 TRUE。	請確定您已將 IAM 角色與 RDS for Oracle 資料庫執行個體建立關聯。
The associated role with your DB instance wasn't found.	安裝 EFS_INTEGRATION 選項期間發生錯誤。RDS 適用於甲骨文是從具有 USE_IAM_ROLE 選項設定的 all-in-one CloudFormation 範本建立的 TRUE。	<p>因應措施是完成下列步驟：</p> <ol style="list-style-type: none"> 1. 使用 IAM 角色和預設選項群組建立資料庫執行個體。 2. 在後續的堆疊更新中，使用選項新增自訂 EFS_INTEGRATION 選項群組。

通知	描述	動作
PLS-00302: component 'CREATE_DIRECTORY_EFS' must be declared	當您使用的是不支援 Amazon EFS 的 RDS for Oracle 版本時，可能會發生此錯誤。	請確定您使用的是 RDS for Oracle 資料庫執行個體版本 19.0.0.0.ru-2022-07.rur-2022-07.r1 或更新版本。
Read access of your EFS is denied. Check your file system policy.	您的資料庫執行個體無法讀取 EFS 檔案系統。	請確定您的 EFS 檔案系統允許透過 IAM 角色或在 EFS 檔案系統層級進行讀取存取。
N/A	您的資料庫執行個體無法寫入至 EFS 檔案系統。	<p>採取下列步驟：</p> <ol style="list-style-type: none"> 1. 請確定已在 Amazon EC2 執行個體上掛載 EFS 檔案系統。 2. 將 <code>others</code> 群組寫入存取權授予您的 RDS 使用者。最簡單的技術是在 EFS 檔案系統的頂端目錄上執行 <code>chmod 777</code> 命令。
<p><code>host -s</code> 命令會傳回 <code>hostname</code> not found: 3(NXDOMAIN)。</p>	您正在使用自訂 DNS 伺服器。	<p>若要在 <code>mount</code> 命令中使用 DNS 名稱，必須符合下列條件：</p> <ul style="list-style-type: none"> • 連線資料庫執行個體位於 VPC 內，且設定為使用 Amazon 提供的 DNS 伺服器。不支援自訂 DNS 伺服器。 • 連線執行個體的 VPC 必須同時已啟用 DNS 解析和 DNS 主機名稱。 • 連線執行個體必須與 EFS 檔案系統位在相同的 VPC 內。

Oracle Java 虛擬機器

Amazon RDS 會使用 JVM 選項，藉此支援 Oracle Java 虛擬機器 (JVM)。Oracle Java 提供 SQL 結構描述和函數，可加速 Oracle 資料庫的 Oracle Java 功能。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle 資料庫中的 Java 簡介](#)。您可以將 Oracle JVM 與所有版本的 Oracle 資料庫 21c (21.0.0) 和 Oracle 資料庫 19c (19.0.0) 搭配使用。

甲骨文 JVM 的注意事項

Amazon RDS 中的 Java 實作擁有有限的一組許可。主要使用者已授予 RDS_JAVA_ADMIN 角色，可授予由 JAVA_ADMIN 角色授予的權限子集。如要列出授予給 RDS_JAVA_ADMIN 角色的權限，請在您的資料庫執行個體上執行以下查詢：

```
SELECT * FROM dba_java_policy
WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
AND enabled = 'ENABLED'
ORDER BY type_name, name, grantee;
```

Oracle JVM 的先決條件

下列是使用 Oracle Java 的先決條件：

- 您的資料庫執行個體必須具有夠大的類別。Oracle Java 不支援 db.t3.micro 或 db.t3.small 資料庫執行個體類別。如需更多詳細資訊，請參閱 [資料庫執行個體類別](#)。
- 您的資料庫執行個體必須將自動次要版本升級啟用。此選項可讓資料庫執行個體自動接收可用的次要資料庫引擎版本升級。Amazon RDS 使用此選項將您的資料庫執行個體更新為最新版 Oracle 修補程式集更新 (PSU) 或版本更新 (RU)。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle JVM 的最佳實務

下列是使用 Oracle Java 的最佳實務：

- 為了將安全性提升至最高，請使用 JVM 選項搭配 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。
- 設定資料庫執行個體以限制網路存取。如需更多詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 及 [在 VPC 中使用資料庫執行個體](#)。
- 如果您符合下列條件，請更新 HTTPS 端點的組態以支援 TLSv1.2：

- 您可以使用 Oracle Java 虛擬機器 (JVM) 透過 TLSv1 或 TLSv1.1 通訊協定來連線 HTTPS 端點。
- 您的端點不支援 TLSv1.2 通訊協定。
- 您尚未將 2021 年 4 月版本更新套用至您的 Oracle 資料庫。

透過更新端點組態，您可以確保 JVM 與 HTTPS 端點的連線能夠繼續運作。如需 Oracle JRE 和 JDK 中 TLS 變更的詳細資訊，請參閱 [Oracle JRE 和 JDK 密碼編譯藍圖](#)。

新增 Oracle JVM 選項

下列是將 JVM 選項新增至資料庫執行個體的一般程序：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

新增 JVM 選項時，會有短暫的停機。在您新增選項之後，就不需要重新啟動資料庫執行個體。只要選項群組為作用中，Oracle Java 就可使用。

Note

在中斷過程中，密碼驗證功能將會短暫停用。在中斷過程中，您也可以預期將會看到與密碼驗證功能相關的事件。在 Oracle 資料庫執行個體可用之前，密碼驗證功能將會再次啟用。

將 JVM 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - 對於 Engine (引擎)，請選擇資料庫執行個體使用的資料庫引擎 (oracle-ee、oracle-se、oracle-se1 或 oracle-se2)。
 - 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 JVM 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：

- 針對新的資料庫執行個體，在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
4. 將所需的許可授予使用者。

Amazon RDS 主要使用者預設有使用 JVM 選項的許可。如果其他使用者需要這些許可，請以主要使用者的身分連線到資料庫執行個體，並且將該許可授予使用者。

下列範例將使用 JVM 選項的許可授予 test_proc 使用者。

```
create user test_proc identified by password;  
CALL dbms_java.grant_permission('TEST_PROC',  
  'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

對使用者授予許可之後，下列查詢應該會傳回輸出。

```
select * from dba_java_policy where grantee='TEST_PROC';
```

Note

Oracle 使用者名稱會區分大小寫，且通常全部是大寫字元。

移除 Oracle JVM 選項

您可以從資料庫執行個體移除 JVM 選項。移除選項時，會有短暫的停機。移除 JVM 選項之後，就不需要重新啟動資料庫執行個體。

⚠ Warning

如果資料庫執行個體使用的是做為選項一部分而啟用的資料類型，移除 JVM 選項可能會造成資料遺失。備份資料之後再繼續。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

若要從資料庫執行個體中移除 JVM 選項，請執行下列其中一個動作：

- 從其所屬的選項群組中移除 JVM 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
- 修改資料庫執行個體，並指定不包括 JVM 選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Enterprise Manager

Amazon RDS 支援 Oracle Enterprise Manager (OEM)。OEM 是 Oracle 的整合企業資訊技術管理產品線。

Amazon RDS 僅支持甲骨文數據庫 19c 非 CDB 上的 OEM。下表說明支援的 OEM 選項。

選項	選項 ID	支援的 OEM 版本
OEM Database Express	OEM	OEM Database Express 12c
OEM Management Agent	OEM_AGENT	OEM Cloud Control for 13c OEM Cloud Control for 12c

Note

您可以使用 OEM Database 或 OEM Management Agent，但無法同時使用。

Note

多租戶架構不支援這些選項。

Oracle Enterprise Manager Database Express

Amazon RDS 會使用 OEM 選項，藉此支援 Oracle Enterprise Manager (OEM) Database Express。Amazon RDS 僅使用非 CDB 架構支持 Oracle 數據庫 19c 的甲骨文企業管理器數據庫快遞。

OEM Database Express 和資料庫控制是類似工具，都具有用於 Oracle 資料庫管理的 Web 型界面。如需這些工具的詳細資訊，請參閱 Oracle 文件中的[存取 Enterprise Manager Database Express 18c](#)和[存取 Enterprise Manager Database Express 12c](#)。

Note

db.t3. 小型資料庫執行個體類別不支援 OEM 資料庫快速運算。如需資料庫執行個體類別的詳細資訊，請參閱[RDS for Oracle 執行個體類別](#)。

OEM 資料庫選項設定

Amazon RDS 支援 OEM 選項的下列設定。

選項設定	有效值	描述
連接埠	整數值	資料庫執行個體上接聽 OEM 資料庫的連接埠。OEM Database Express 的預設值為 5500。
安全群組	—	有權存取 Port (連接埠) 的安全群組。

新增 OEM 資料庫選項

將 OEM 選項新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

當您新增 OEM 選項時，資料庫執行個體自動重新啟動時會發生短暫中斷。

將 OEM 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle 版本。
 - b. 針對 Major engine version (主要引擎版本)，請選擇適用於資料庫執行個體版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 OEM 選項新增至選項群組，並設定選項設定。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。如需每項設定的詳細資訊，請參閱 [OEM 資料庫選項設定](#)。

Note

如果您將 OEM 選項新增至已連接至一或多個資料庫執行個體的現有選項群組，則會在自動重新啟動所有資料庫執行個體時發生短暫中斷。

3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。當您新增 OEM 選項時，資料庫執行個體自動重新啟動時會發生短暫中斷。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Note

您也可以使用 AWS CLI 來新增 OEM 選項。如需範例，請參閱 [將選項新增至選項群組](#)。

透過瀏覽器存取 OEM

在啟用 OEM 選項之後，您可以從 Web 瀏覽器開始使用 OEM 資料庫工具。

您可以從 Web 瀏覽器存取 OEM 資料庫控制或 OEM Database Express。例如，如果 Amazon RDS 資料庫執行個體的端點為 `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`，且 OEM 連接埠為 1158，則用來存取 OEM 資料庫控制的 URL 將如下所示。

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

當您從 web 瀏覽器存取任一工具時，登入視窗會出現，提示您輸入使用者名稱和密碼。輸入資料庫執行個體的主要使用者名稱和主要密碼。您現已準備好可管理您的 Oracle 資料庫。

修改 OEM 資料庫設定

在啟用 OEM 資料庫之後，您可以修改選項的安全群組設定。

在使選項群組與資料庫執行個體產生關聯之後，就無法修改 OEM 連接埠號碼。若要變更資料庫執行個體的 OEM 連接埠號碼，請執行下列動作：

1. 建立新的選項群組。
2. 將 OEM 選項與新的連接埠號碼加入至新的選項群組。
3. 從資料庫執行個體中移除現有的選項群組。
4. 將新的選項群組加入至資料庫執行個體。

如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。如需每項設定的詳細資訊，請參閱[OEM 資料庫選項設定](#)。

執行 OEM Database Express 任務

您可以使用 Amazon RDS 程序來執行某些 OEM Database Express 任務。透過執行這些程序，您可以完成下方列出的任務。

Note

OEM Database Express 任務會以非同步方式執行。

任務

- [將 OEM Database Express 網站前端切換為 Adobe Flash](#)
- [將 OEM Database Express 的網站前端切換為 Oracle JET](#)

將 OEM Database Express 網站前端切換為 Adobe Flash

Note

此任務僅適用於 Oracle Database 19c 非 CDB。

從 Oracle 資料庫 19c 開始，Oracle 已取代以前以 Adobe Flash 為基礎的 OEM Database Express 使用者界面。相反地，「OEM Database Express」現在使用 Oracle JET 內建的介面。如果您在使用新介面時遇到困難，可以切換回已取代的 Flash 版本介面。您可能會遇到的新介面使用困難，包括登入 OEM Database Express 後卡在 Loading 螢幕上。您也可能會找不到 OEM Database Express 的 Flash 版本中存在的某些功能。

若要將 OEM Database Express 網站前端切換為 Adobe Flash，請執行此 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash`。此程序相當於 `execemx emx SQL` 命令。

安全最佳實務不建議使用 Adobe Flash。雖然您可以還原至 Flash 版本的 OEM Database Express，但我們建議您儘可能使用 JET 版本的 OEM Database Express 網站。如果您還原至使用 Adobe Flash，並想要切換回使用 Oracle JET，請使用 `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` 程序。在 Oracle 資料庫升級之後，較新版的 Oracle JET 可能會解決 OEM Database Express 中與 JET 相關的問題。如需切換至 Oracle JET 的詳細資訊，請參閱 [將 OEM Database Express 的網站前端切換為 Oracle JET](#)。

Note

從僅供讀取複本的來源資料庫執行個體執行此任務，也會導致僅供讀取複本將其 OEM Database Express 網站前端切換為 Adobe Flash。

下列程序呼叫會建立任務，將 OEM Database Express 網站切換為 Adobe Flash，並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

您可以透過顯示任務的輸出檔案來檢視結果。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

以程序傳回的任務 ID 取代 *task-id*。如需 Amazon RDS 程序 `rdsadmin.rds_file_util.read_text_file` 的詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#)

您也可以在此「記錄與事件」區段中搜尋 AWS Management Console 的記錄項目，在中檢視工作輸出檔案的內容 `task-id`。

將 OEM Database Express 的網站前端切換為 Oracle JET

Note

此任務僅適用於 Oracle Database 19c 非 CDB。

若要將 OEM Database Express 網站前端切換至 Oracle JET，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`。此程序相當於 `execemx omx SQL` 命令。

依照預設，執行 19c 或更新版本之 Oracle 資料庫執行個體的 OEM Database Express 網站會使用 Oracle JET。如果您使用 `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` 程序將 OEM Database Express 網站前端切換為 Adobe Flash，則可以切換回 Oracle JET。若要執行這項操作，請使用 `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` 程序。如需切換至 Adobe Flash 的詳細資訊，請參閱 [將 OEM Database Express 網站前端切換為 Adobe Flash](#)。

Note

從僅供讀取複本的來源資料庫執行個體執行此工作，也會導致僅供讀取複本將其 OEM Database Express 網站前端切換為 Oracle JET。

下列程序呼叫會建立將 OEM Database Express 網站切換至 Oracle JET 的任務，並傳回任務的 ID。

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

您可以透過顯示任務的輸出檔案來檢視結果。

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```


以程序傳回的任務 ID 取代 *task-id*。如需 Amazon RDS 程序 `rdsadmin.rds_file_util.read_text_file` 的詳細資訊，請參閱 [讀取資料庫執行個體目錄中的檔案](#)

您也可以在此「記錄與事件」區段中搜尋 AWS Management Console 的記錄項目，在中檢視工作輸出檔案的內容 `task-id`。

移除 OEM 資料庫選項

您可以從資料庫執行個體中移除 OEM 選項。移除 OEM 選項時，系統會在執行個體自動重新啟動時發生短暫中斷。因此，在您移除 OEM 選項之後，您並不需要重新啟動資料庫執行個體。

若要從資料庫執行個體中移除 OEM 選項，請執行下列其中一個動作：

- 從其所屬的選項群組中移除 OEM 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
- 修改資料庫執行個體，並指定不包括 OEM 選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent

Oracle Enterprise Manager (OEM) Management Agent 是一項軟體元件，可監控主機上執行的目標，並將該資訊傳達給中間層 Oracle Management Service (OMS)。Amazon RDS 透過使用 OEM_AGENT 選項支援 Management Agent。

如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Enterprise Manager Cloud Control 12c 的概觀](#) 和 [Oracle Enterprise Manager Cloud Control 13c 的概觀](#)。

主題

- [管理代理程式的需求](#)
- [OMS 主機通訊先決條件](#)
- [Management Agent 的限制](#)
- [Management Agent 的選項設定](#)
- [新增 Management Agent 選項](#)
- [使用 Management Agent](#)
- [修改 Management Agent 設定](#)
- [使用 Management Agent 執行資料庫任務](#)
- [移除 Management Agent 選項](#)

管理代理程式的需求

以下是使用「管理代理程式」的一般需求：

- 您的資料庫執行個體必須使用非 CDB 架構執行 Oracle 資料庫 19c (19.0.0.0)。
- 您必須使用設定為連線到資料庫執行個體的 Oracle 管理服務 (OMS)。請注意下列 OMS 需求：
 - Management Agent 13.5.0.0.v1 版需要 OMS 13.5.0.0 版或更新版本。
 - Management Agent 13.4.0.9.v1 版需要 OMS 13.4.0.9 版或更新版本，以及修補程式 32198287。
- 在多數情況下，您需要將 VPC 設定為允許從 OMS 連線至資料庫執行個體。如果您不熟悉 Amazon Virtual Private Cloud (Amazon VPC)，建議您先完成[教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)中的步驟，再繼續操作。
- 您可以在 12c 和 13c 的 Oracle 企業管理員雲端控制中使用管理代理程式。請確定您有足夠的儲存空間供您的 OEM 版本使用：
 - OEM 13c 版本 5 至少 8.5 GiB
 - OEM 13c 發行版本 4 至少 8.5 GiB

- OEM 13c 發行版本 3 至少 8.5 GiB
 - OEM 13c 發行版本 2 至少 5.5 GiB
 - OEM 13c 發行版本 1 至少 4.5 GiB
 - OEM 12c 至少 2.5 GiB
- 如果您使用的是「管理代理程式」版本 OEM_AGENT 13.2.0.0.v313.3.0.0.v2，而且想要使用 TCPS 連線，請遵循 Oracle 說明文件中[設定協力廠商 CA 憑證與目標資料庫通訊](#)中的指示。另外，遵循 Oracle 文件 (Oracle Doc ID 2241358.1) 中的指示，更新 OMS 上的 JDK。此步驟可確保只要是資料庫支援的密碼套件，OMS 也都支援。

Note

Management Agent OEM_AGENT 13.2.0.0.v3、13.3.0.0.v2、13.4.0.9.v1 和更新版本支援 Management Agent 與資料庫執行個體之間的 TCPS 連線。

OMS 主機通訊先決條件

請確定您的 OMS 主機和 Amazon RDS 資料庫執行個體可以通訊。執行下列操作：

- 若要從 Management Agent 連接至 OMS，如果 OMS 位在防火牆之後，請將資料庫執行個體的 IP 地址新增至 OMS。

確定 OMS 的防火牆會允許源自資料庫執行個體的 IP 地址、來自資料庫接聽程式連接埠 (預設值 1521) 和 OEM Agent 連接埠 (預設值 3872) 的流量。

- 若要從 OMS 連接至 Management Agent，如果 OMS 有可公開解析的主機名稱，請將 OMS 地址新增至安全群組。您的安全群組必須具備可允許存取資料庫接聽程式連接埠和 Management Agent 連接埠的傳入規則。如需建立安全性和新增傳入規則的範例，請參閱[教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。
- 若要從 OMS 連接至 Management Agent，如果 OMS 沒有可公開解析的主機名稱，請使用以下其中一個項目：
 - 如果您是在私有 VPC 的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上託管 OMS，則可將 VPC 對等設定為從 OMS 連接至 Management Agent。如需更多詳細資訊，請參閱[由不同 VPC 中的 EC2 執行個體存取 VPC 中的資料庫執行個體](#)。
 - 如果 OMS 在現場部署中進行託管，您可以將 VPN 連接設定為允許從 OMS 存取 Management Agent。如需詳細資訊，請參閱[由用戶端應用程式透過網際網路存取 VPC 中的資料庫執行個體或 VPN 連接](#)。

Management Agent 的限制

下面是使用 Management Agent 時的一些限制：

- 您無法提供自訂的 Oracle 管理代理程式映像檔。
- 不支援需要主機登入資料的管理任務 (例如：任務執行和資料庫修補)。
- 不保證主機指標和程序清單可反映實際的系統狀態。因此，您不應該使用 OEM 來監控根檔案系統或掛載點檔案系統。如需監控作業系統的詳細資訊，請參閱[使用增強型監控來監控作業系統指標](#)。
- 不支援自動探索。您必須手動新增資料庫目標。
- OMS 模組可用性取決於您的資料庫版本。例如，資料庫效能診斷和調校模組僅供 Oracle Database Enterprise Edition 使用。
- Management Agent 會耗用額外的記憶體和運算資源。如果在啟用 OEM_AGENT 選項之後遇到效能問題，建議您擴展至較大型的資料庫執行個體類別。如需更多詳細資訊，請參閱[資料庫執行個體類別](#)及[修改 Amazon RDS 資料庫執行個體](#)。
- 在 Amazon RDS 主機上執行 OEM_AGENT 的使用者沒有存取提醒日誌的作業系統存取權。因此，無法在 OEM 中收集 DB Alert Log 和 DB Alert Log Error Status 的指標。

Management Agent 的選項設定

Amazon RDS 支援 Management Agent 選項的下列設定。

選項設定	必要	有效值	描述
版本 (AGENT_VERSION)	是	13.5.0.0.v1 13.4.0.9.v1 13.3.0.0.v2 13.3.0.0.v1 13.2.0.0.v3	Management Agent 軟體的版本。支援的最低版本為13.1.0.0.v1 。 選 AWS CLI 項名稱為OptionVersion 。 Note 在地 AWS GovCloud (US) 區中，13.1 版本不可用。

選項設定	必要	有效值	描述
		13.2.0.0. v2 13.2.0.0. v1 13.1.0.0. v1	
連接埠 (AGENT_PORT)	是	整數值	資料庫執行個體上接聽 OMS 主機的连接埠。預設為 3872。您的 OMS 主機必須屬於可存取此連接埠的安全群組。 選 AWS CLI 項名稱為Port。
安全群組	是	現有的安全群組	有權存取 Port (連接埠) 的安全群組。您的 OMS 主機必須屬於此安全群組。 選 AWS CLI 項名稱為VpcSecurityGroupMemberships 或DBSecurityGroupMemberships 。
OMS_HOST	是	字串值， 例如 <i>my.example.oms</i>	OMS 可公開存取的主機名稱或 IP 地址。 選 AWS CLI 項名稱為OMS_HOST。
OMS_PORT	是	整數值	OMS 主機上接聽 Management Agent 的 HTTPS 上傳連接埠。 若要判斷 HTTPS 上傳連接埠，請連接至 OMS 主機並執行下列命令 (此命令需要 SYSMAN 密碼)： <pre>emctl status oms -details</pre> 選 AWS CLI 項名稱為OMS_PORT。

選項設定	必要	有效值	描述
AGENT_REGISTRATION_PASSWORD	是	字串值	Management Agent 用來向 OMS 驗證自己的密碼。建議您在 OMS 中建立一致的密碼，之後再啟用 OEM_AGENT 選項。您可以利用一致的密碼，在多個 Amazon RDS 資料庫間共用單一的 Management Agent 選項群組。 選 AWS CLI 項名稱為 AGENT_REGISTRATION_PASSWORD。
ALLOW_TLS_ONLY	否	true、false (預設)	值：將 OEM 代理程式設定為做為伺服器接聽時，只支援 TLSv1 通訊協定。不再支援此設定。根據預設，管理代理程式 13.1.0.0.v1 及更新版本支援傳輸層安全性 (TLS)。
MINIMUM_TLS_VERSION	否	TLSv1、(預設)TLSv1.2	值，將 OEM 代理程式做為伺服器接聽時，指定代理程式支援的最低 TLS 版本。不支援的代理程式版本僅支援此設定。TLSv1
TLS_CIPHER_SUITE	否	請參閱 Management Agent 選項的 TLS 設定 。	值，將 OEM 代理程式做為伺服器接聽時，指定代理程式使用的 TLS 密碼套件。

下表列出 Management Agent 選項支援的 TLS 密碼套件。

Management Agent 選項的 TLS 設定

密碼套件	支援的代理程式版本	FedRAMP 合規
TLS_RSA_WITH_AES_128_CBC_SHA	全部	否

密碼套件	支援的代理程式版本	FedRAMP 合規
TLS_RSA_WITH_AES_128_CBC_SH A256	一版及更高版本	否
TLS_RSA_WITH_AES_256_CBC_SHA	3.3 及更高版本	否
TLS_RSA_WITH_AES_256_CBC_SH A256	3.3 及更高版本	否
TLS_ECDHE_RSA_WITH_AES_128_ CBC_SHA	3.3 及更高版本	是
TLS_ECDHE_RSA_WITH_AES_256_ CBC_SHA	3.3 及更高版本	是
TLS_ECDHE_RSA_WITH_AES_128_ CBC_SHA256	3.3 及更高版本	是
TLS_ECDHE_RSA_WITH_AES_256_ CBC_SHA384	3.3 及更高版本	是

新增 Management Agent 選項

若要將管理代理程式選項新增至資料庫執行個體，請執行下列動作：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

如果您遇到錯誤，可以查看 [My Oracle 支援](#) 文件，以取得解決特定問題的相關資訊。

新增 Management Agent 選項之後，您並不需要重新啟動資料庫執行個體。選項群組開始作用後，OEM Agent 就會處於作用中狀態。

如果 OMS 主機使用的是不受信任的第三方憑證，則 Amazon RDS 會傳回下列錯誤。

You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted third party certificate.

Configure your OMS host with the trusted certificates from your third party.

系統傳回錯誤後，Management Agent 選項將無法啟用，直到該問題修正為止。如需修正問題的相關資訊，請參閱 My Oracle 支援文件 [2202569.1](#)。

主控台

新增 Management Agent 選項至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle 版本。
 - b. 針對 Major engine version (主要引擎版本)，請選擇適用於資料庫執行個體版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 OEM_AGENT 選項新增至選項群組，並設定選項設定。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。如需每項設定的詳細資訊，請參閱 [Management Agent 的選項設定](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

下列範例會使用 AWS CLI [add-option-to-option-group](#) 命令，將 OEM_AGENT 選項設定新增至名為 myoptiongroup 的選項群組。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name "myoptiongroup" \  
  --options  
  OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456  
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \  
  --apply-immediately
```


在 Windows 中：

```
aws rds add-option-to-option-group ^
  --option-group-name "myoptiongroup" ^
  --options
  OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
  [{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
  --apply-immediately
```

使用 Management Agent

啟用 Management Agent 選項之後，請採用下列步驟來開始使用它。

使用 Management Agent

1. 解除鎖定和重設 DBSNMP 帳戶登入資料。透過在資料庫執行個體的目標資料庫上執行下列程式碼，並使用您的主要使用者帳戶。

```
ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

2. 手動將目標新增至 OMS 主控台：
 - a. 在您的 OMS 主控台中，依序選擇 Setup (設定)、Add Target (新增目標)、Add Targets Manually (手動新增目標)。
 - b. 選擇 Add Targets Declaratively by Specifying Target Monitoring Properties (透過指定目標監控屬性以宣告方式新增目標)。
 - c. 針對 Target Type (目標類型)，選擇 Database Instance (資料庫執行個體)。
 - d. 針對 Monitoring Agent (監控代理程式)，選擇具有與 RDS 資料庫執行個體識別符相同之識別符的代理程式。
 - e. 選擇 Add Manually (手動新增)。
 - f. 輸入 Amazon RDS 資料庫執行個體的端點，或從主機名稱清單中選擇。確定指定的主機名稱符合 Amazon RDS 資料庫執行個體的端點。

如需尋找 Amazon RDS 資料庫執行個體端點的相關資訊，請參閱[尋找您的 RDS for Oracle 資料庫執行個體的端點](#)。

- g. 指定下列資料庫屬性：
 - 針對 Target name (目標名稱)，輸入名稱。
 - 針對 Database system name (資料庫系統名稱)，輸入名稱。

- 針對 Monitor username (監控使用者名稱)，輸入 **dbsnmp**。
 - 針對 Monitor password (監控密碼)，輸入來自步驟 1 的密碼。
 - 針對 Role (角色)，輸入 normal (一般)。
 - 針對 Oracle home path (Oracle 主目錄)，輸入 **/oracle**。
 - 針對 Listener Machine name (接聽程式機器名稱)，代理程式識別符已顯示。
 - 針對 Port (連接埠)，輸入資料庫連接埠。RDS 的預設連接埠為 1521。
 - 針對 Database name (資料庫名稱)，輸入資料庫的名稱。
- h. 選擇 Test Connection (測試連接)。
- i. 選擇下一步。目標資料庫會出現在您監控的資源清單中。

修改 Management Agent 設定

啟用 Management Agent 之後，您可以修改選項的設定。如需如何修改選項設定的詳細資訊，請參閱[修改選項設定](#)。如需每項設定的詳細資訊，請參閱[Management Agent 的選項設定](#)。

使用 Management Agent 執行資料庫任務

您可以使用 Amazon RDS 程序在 Management Agent 上執行特定的 EMCTL 指令。透過執行這些程序，您可以完成下方列出的任務。

Note

任務是以非同步方式執行。

任務

- [取得 Management Agent 的狀態](#)
- [重新啟動 Management Agent](#)
- [列出 Management Agent 監控的目標](#)
- [列出 Management Agent 監控的收集執行緒](#)
- [清除 Management Agent 狀態](#)
- [讓 Management Agent 上傳其 OMS](#)
- [抓取 OMS](#)
- [檢視進行中任務的狀態](#)

取得 Management Agent 的狀態

若要取得 Management Agent 的狀態，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`。此程序相當於 `emctl status agent` 命令。

以下程序會建立任務，取得管理代理程式的狀態並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

重新啟動 Management Agent

若要重新啟動 Management Agent，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent`。此程序相當於執行 `emctl stop agent` 和 `emctl start agent` 命令。

以下程序會建立任務，以重新啟動 Management Agent 並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

列出 Management Agent 監控的目標

若要列出 Management Agent 監控的目標，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent`。此程序相當於執行 `emctl config agent listtargets` 命令。

以下程序會建立任務，以列出 Management Agent 所監控的目標並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

列出 Management Agent 監控的收集執行緒

若要列出 Management Agent 監控的所有執行中、就緒和排定的收集執行緒，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent`。此程序相當於 `emctl status agent scheduler` 命令。

以下程序會建立任務，以列出收集執行緒並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

清除 Management Agent 狀態

若要清除 Management Agent 狀態，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent`。此程序相當於執行 `emctl clearstate agent` 命令。

以下程序會建立任務，清除 Management Agent 的狀態並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

讓 Management Agent 上傳其 OMS

若要讓 Management Agent 上傳與其相關的 Oracle 管理伺服器 (OMS)，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent`。此程序相當於執行 `emctl upload agent` 命令。

以下程序會建立任務，讓 Management Agent 上傳其相關的 OMS，並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

抓取 OMS

若要抓取 Management Agent 的 OMS，請執行 Amazon RDS 程序 `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent`。此程序相當於執行 `emctl pingOMS` 命令。

以下程序會建立任務，取得 Management Agent 的 OMS 並傳回任務 ID。

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

若要透過顯示任務的輸出檔案來檢視結果，請參閱[檢視進行中任務的狀態](#)。

檢視進行中任務的狀態

您可在 bdump 檔案之中，檢視進行中作業的狀態。bdump 檔案位在 `/rdsdbdata/log/trace` 目錄：每個 bdump 檔案的名稱格式如下。

```
dbtask-task-id.log
```

當您想要監控任務時，將 *task-id* 替換為您要監控作業的 ID。

若要檢視 bdump 檔案的內容，請執行 Amazon RDS 程序 `rdsadmin.rds_file_util.read_text_file`。以下查詢會傳回 `dbtask-1546988886389-2444.log` bdump 檔案的內容。

```
SELECT text FROM  
table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

如需 Amazon RDS 程序 `rdsadmin.rds_file_util.read_text_file` 的詳細資訊，請參閱[讀取資料庫執行個體目錄中的檔案](#)。

移除 Management Agent 選項

您可以從資料庫執行個體中移除 OEM Agent。移除 OEM Agent 之後，您並不需要重新啟動資料庫執行個體。

若要從資料庫執行個體中移除 OEM Agent，請執行下列其中一個動作：

- 從其所屬的選項群組中移除 OEM Agent 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱[從選項群組移除選項](#)。
- 修改資料庫執行個體，並指定不包括 OEM Agent 的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

Oracle Label Security

Amazon RDS 透過使用 OLS 選項支援 Oracle Database 的 Oracle Label Security for the Enterprise Edition。

大部分的資料庫安全機制可控制物件層級的存取。Oracle Label Security 可對個別資料表列的存取提供精細分級的控制。例如，您可以使用 Label Security，以政策型管理模型來強制執行法規遵循。您可以使用 Label Security 政策，控制機密資料的存取，並限制僅具有適當許可層級的使用者才能存取。如需詳細資訊，請參閱 Oracle 文件中的[簡介 Oracle Label Security](#)。

主題

- [Oracle Label Security 的先決條件](#)
- [新增 Oracle Label Security 選項](#)
- [使用 Oracle Label Security](#)
- [移除 Oracle Label Security 選項 \(不支援\)](#)
- [故障診斷](#)

Oracle Label Security 的先決條件

熟悉 Oracle Label Security 的以下先決條件：

- 您的資料庫執行個體必須使用「使用自有授權」模型。如需更多詳細資訊，請參閱 [RDS for Oracle 授權選項](#)。
- 您必須對具有軟體更新授權和支援的 Oracle Enterprise Edition 具備有效的授權。
- 您的 Oracle 授權必須包括 Label Security 選項。
- 您必須使用非多租戶 (非 CDB) 資料庫架構。如需詳細資訊，請參閱 [CDB 架構的單一租戶組態](#)。

新增 Oracle Label Security 選項

將 Oracle Label Security 選項新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。

⚠ Important

Oracle Label Security 是永久且持續的選項。

3. 將選項群組與資料庫執行個體建立關聯。

在新增 Label Security 選項之後，只要選項群組為作用中狀態，Label Security 就會為作用中狀態。

將 Label Security 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇 oracle-ee。
 - b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 OLS 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。

⚠ Important

如果您將 Label Security 新增至已附加至一個或多個資料庫執行個體的現有選項群組，則所有資料庫執行個體都會重新啟動。

3. 將選項群組套用至新的或現有的資料庫執行個體：

- 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。當您將 Label Security 選項新增至現有的資料庫執行個體時，在資料庫執行個體自動重新啟動時會發生短暫的停機。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

使用 Oracle Label Security

若要使用 Oracle Label Security，您可以建立政策，控制資料表中特定資料列的存取。如需詳細資訊，請參閱 Oracle 文件中的 [建立 Oracle Label Security 原則](#)。

使用 Label Security 時，您會以 LBAC_DBA 角色執行所有動作。資料庫執行個體的主要使用者會獲授予 LBAC_DBA 角色。您可以將 LBAC_DBA 角色授予其他使用者，以便他們可以管理 Label Security 政策。

對於使用非 CDB 架構的 Oracle 資料庫 19c，請務必將 OLS_ENFORCEMENT 封裝的存取權授與任何需要「Oracle 標籤安全性」存取權的新使用者。

若要授予 OLS_ENFORCEMENT 套件的存取權，請以主要使用者身分連接至資料庫執行個體，然後執行下列 SQL 陳述式：

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

您可以透過 Oracle Enterprise Manager (OEM) Cloud Control 來設定 Label Security。Amazon RDS 可藉由 Management Agent 選項支援 OEM Cloud Control。如需詳細資訊，請參閱 [適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent](#)。

移除 Oracle Label Security 選項 (不支援)

Oracle Label Security 是永久且持續的選項。因為此選項是永久的，所以您無法從選項群組中將其移除。如果您將 Oracle Label Security 新增至選項群組，並將其與資料庫執行個體建立關聯，則稍後可以將不同的選項群組與資料庫執行個體建立關聯，但此群組也必須包含 Oracle Label Security 選項。

故障診斷

下列是您使用 Oracle Label Security 時可能遇到的問題。

問題	故障診斷建議
當您嘗試建立政策時，您會看到如下的錯誤訊息：insufficient authorization for the SYSDBA package。	Oracle Label Security 功能的已知問題會防止使用者名稱有 16 或 24 個字元的使用者執行 Label Security 命令。您可以建立字元數不同的新使用者、將 LBAC_DBA 授予新使用者、以新使用者身分登入，並以新使用者身分執行 OLS 命令。如需其他資訊，請聯絡 Oracle 客戶 Support 部。

Oracle Locator

Amazon RDS 會使用 LOCATOR 選項，藉此支援 Oracle Locator。Oracle Locator 提供通常支援下列項目所需的功能：網際網路和無線服務型應用程式和合作夥伴型 GIS 解決方案。Oracle Locator 是功能有限的 Oracle Spatial 子集。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Locator](#)。

Important

如果您使用 Oracle Locator，若有通用漏洞評估系統 (CVSS) 分數 9 分以上的安全漏洞或其他公布的安全漏洞，Amazon RDS 會將您的資料庫執行個體自動更新為最新的 Oracle PSU。

Oracle 儲位器支援的資料庫版本

適用於甲骨文的 RDS 支持 Oracle 數據庫 19c 的 Oracle 定位器。Oracle Database 21c 不支援 Oracle Locator，但其功能可在 Oracle Spatial 選項中使用。以前，Spatial 選項需要額外的授權。Oracle Locator 過去代表 Oracle Spatial 功能的子集，當時並不需要額外的授權。2019 年，甲骨文宣布所有 Oracle 空間功能都包含在企業版和標準版 2 許可證中，而無需額外費用。因此，Oracle Spatial 選項不再需要額外的授權。如需詳細資訊，請參閱 Oracle Database Insider 部落格中的 [機器學習、Spatial 和 Graph - 無需授權！](#)。

Oracle Locator 的先決條件

下列是使用 Oracle Locator 的先決條件：

- 您的資料庫執行個體必須具有足夠的類別。Oracle Locator 不支援 db.t3.micro 或 db.t3.small 資料庫執行個體類別。如需更多詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。
- 您的資料庫執行個體必須將自動次要版本升級啟用。此選項可讓資料庫執行個體自動接收可用的次要資料庫引擎版本升級，而且是安裝 Oracle Java Virtual Machine (JVM) 之任何選項的必要項目。Amazon RDS 使用此選項將您的資料庫執行個體更新為最新版 Oracle 修補程式集更新 (PSU) 或版本更新 (RU)。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Locator 的最佳實務

下列是使用 Oracle Locator 的最佳實務：

- 為了將安全性提升至最高，請使用 LOCATOR 選項搭配 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。

- 設定資料庫執行個體以限制對資料庫執行個體的存取。如需更多詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 及 [在 VPC 中使用資料庫執行個體](#)。

新增 Oracle Locator 選項

下列是將 LOCATOR 選項新增至資料庫執行個體的一般程序：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在新增 LOCATOR 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。在您新增選項之後，就不需要重新啟動資料庫執行個體。只要選項群組為作用中，Oracle Locator 就可使用。

Note

在中斷過程中，密碼驗證功能將會短暫停用。在中斷過程中，您也可以預期將會看到與密碼驗證功能相關的事件。在 Oracle 資料庫執行個體可用之前，密碼驗證功能將會再次啟用。

將 LOCATOR 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle 版本。
 - b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 LOCATOR 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

- 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

使用 Oracle Locator

在啟用 Oracle Locator 選項之後，您可以開始使用它。您應該只使用 Oracle Locator 功能。不要使用任何 Oracle Spatial 功能，除非您有 Oracle Spatial 的所有授權。

如需 Oracle Locator 支援之功能的清單，請參閱 Oracle 文件中的 [Locator 隨附的功能](#)。

如需 Oracle Locator 不支援之功能的清單，請參閱 Oracle 文件中的 [Locator 未隨附的功能](#)。

移除 Oracle Locator 選項

捨棄使用該 LOCATOR 選項提供之資料類型的所有物件後，您可以從資料庫執行個體移除該選項。如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在移除 LOCATOR 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。移除 LOCATOR 選項之後，就不需要重新啟動資料庫執行個體。

捨棄 LOCATOR 選項

1. 備份您的資料。

Warning

如果執行個體使用已啟用做為選項一部分的資料類型，而且如果您移除該 LOCATOR 選項，則可能會遺失資料。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

2. 檢查是否有任何現有物件參考資料類型或 LOCATOR 選項的功能。

如果 LOCATOR 選項存在，則在套用沒有 LOCATOR 選項的新選項群組時，執行個體可能會停滯。您可以使用下列查詢以識別物件：

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_GEOMETRY')
```

```
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;

SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM    DBA_TAB_COLUMNS
WHERE   DATA_TYPE = 'SDO_GEOMETRY'
AND     OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. 捨棄參考資料類型或 LOCATOR 選項功能的所有物件。
4. 請執行下列其中一項：
 - 從其所屬的選項群組中移除 LOCATOR 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
 - 修改資料庫執行個體，並指定不包括 LOCATOR 選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle 原生網路加密

Amazon RDS 支援 Oracle 原生網路加密 (NNE)。使用 `NATIVE_NETWORK_ENCRYPTION` 此選項，您可以在資料庫執行個體移入和移出資料庫執行個體時加密資料。Amazon RDS 可支援 Oracle 資料庫所有版本的 NNE。

Oracle 原生網路加密的詳細討論超出本指南的範圍，但是在決定要使用哪個解決方案進行部署之前，您應該瞭解每一個演算法和金鑰的優缺點。如需可透過 Oracle 原生網路加密使用之演算法和金鑰的相關資訊，請參閱 Oracle 文件中的 [配置網路資料加密](#)。如需 AWS 安全性的詳細資訊，請造訪 [AWS 安全中心](#)。

Note

您可以使用原生網路加密或 Secure Sockets Layer，但不能同時使用兩者。如需詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。

原生網路加密選項設定

您可以同時在伺服器 and 用戶端上指定加密需求。例如，當資料庫執行個體使用資料庫連結連線到另一個資料庫時，資料庫執行個體可以充當用戶端。您可能想要避免在伺服器端強制加密。例如，您可能不希望因為伺服器要求而強制所有用戶端通訊使用加密。在這種情況下，您可以使用 `SQLNET.*CLIENT` 選項，強制在用戶端上執行加密。

Amazon RDS 支持該 `NATIVE_NETWORK_ENCRYPTION` 選項的以下設置。

Note

當您使用逗號區隔選項設定的值時，請不要在逗號後面加上空格。

選項設定	有效值	預設值	描述
<code>SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS</code>	TRUE, FALSE	TRUE	使用非安全加密的用戶端嘗試連接至資料庫時伺服器的行為。如果為 TRUE，即使沒有使用 2021 年 7 月 PSU 進行修補，用戶端也可以連接。

選項設定	有效值	預設值	描述
			<p>如果設定為 FALSE，則僅當用戶端使用 2021 年 7 月 PSU 進行修補時才能連接至資料庫。在設定 <code>SQLNET.ALLOW_WEAK_CRYPTOClients</code> 為 FALSE 之前，請確保滿足以下條件：</p> <ul style="list-style-type: none"> • <code>SQLNET.ENCRYPTION_TYPES_SERVER</code> 和 <code>SQLNET.ENCRYPTION_TYPES_CLIENT</code> 具有一個相符的加密方法，但該方法不是 DES、3DES 或 RC4 (所有金鑰長度)。 • <code>SQLNET.CHECKSUM_TYPES_SERVER</code> 和 <code>SQLNET.CHECKSUM_TYPES_CLIENT</code> 具有一個相符的檢查總和方法，但該方法不是 MD5。 • 用戶端已使用 2021 年 7 月 PSU 修補。如果用戶端未修補，則用戶端會失去連線並接收到 ORA-12269 錯誤。

選項設定	有效值	預設值	描述
SQLNET.ALLOW_WEAK_CRYPT0	TRUE, FALSE	TRUE	<p>使用非安全加密的用戶端嘗試連接至資料庫時伺服器的行為。以下加密方式被視為不安全：</p> <ul style="list-style-type: none"> • DES 加密方法 (所有金鑰長度) • 3DES 加密方法 (所有金鑰長度) • RC4 加密方法 (所有金鑰長度) • MD5 檢查總和方法 <p>如果設定為 TRUE，用戶端可在使用上述非加密方式時進行連線。</p> <p>如果設定為 FALSE，當用戶端使用上述非安全加密方式時，資料庫會阻止用戶端連線。在設定 SQLNET.ALLOW_WEAK_CRYPT0 為 FALSE 之前，請確保滿足以下條件：</p> <ul style="list-style-type: none"> • SQLNET.ENCRYPTION_TYPES_SERVER 和 SQLNET.ENCRYPTION_TYPES_CLIENT 具有一個相符的加密方法，但該方法不是 DES、3DES 或 RC4 (所有金鑰長度)。 • SQLNET.CHECKSUM_TYPES_SERVER 和 SQLNET.CHECKSUM_TYPES_CLIENT 具有一個相符的檢查總和方法，但該方法不是 MD5。 • 用戶端已使用 2021 年 7 月 PSU 修補。如果用戶端未修補，則用戶端會失去連線並接收到 ORA-12269 錯誤。

選項設定	有效值	預設值	描述
SQLNET.CRYPTO_CHECKSUM_CLIENT	Accepted, Rejected, Requested, Required	Requested	<p>資料庫執行個體連線至用戶端或充當用戶端的伺服器時的資料完整性行為。當資料庫執行個體使用資料庫連結時，它會充當用戶端。</p> <p>Requested 表示用戶端不需要資料庫執行個體執行檢查總和。</p>
SQLNET.CRYPTO_CHECKSUM_SERVER	Accepted, Rejected, Requested, Required	Requested	<p>用戶端或做為用戶端的伺服器連線至資料庫執行個體時的資料完整性。當資料庫執行個體使用資料庫連結時，它會充當用戶端。</p> <p>Requested 表示資料庫執行個體不需要用戶端執行檢查總和。</p>
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512	<p>總和檢查演算法的清單。</p> <p>您可以指定一值或逗號分隔的值清單。如果您使用逗號，請不要在逗號後面插入空格；否則，您會收到 <code>InvalidParameterValue</code> 錯誤。</p> <p>此參數和 <code>SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER</code> 必須有一個共同的密碼。</p>
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512, SHA1, MD5	<p>總和檢查演算法的清單。</p> <p>您可以指定一值或逗號分隔的值清單。如果您使用逗號，請不要在逗號後面插入空格；否則，您會收到 <code>InvalidParameterValue</code> 錯誤。</p> <p>此參數和 <code>SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT</code> 必須有一個共同的密碼。</p>

選項設定	有效值	預設值	描述
SQLNET.ENCRYPTION_CLIENT	Accepted Rejected Requested , Required	Requested	<p>用戶端或充當用戶端的伺服器連線至資料庫執行個體時用戶端的加密行為。當資料庫執行個體使用資料庫連結時，它會充當用戶端。</p> <p>Requested 表示用戶端不需要加密來自伺服器的流量。</p>
SQLNET.ENCRYPTION_SERVER	Accepted Rejected Requested , Required	Requested	<p>用戶端或充當用戶端的伺服器連線至資料庫執行個體時伺服器的加密行為。當資料庫執行個體使用資料庫連結時，它會充當用戶端。</p> <p>Requested 表示資料庫執行個體不需要將來自用戶端的流量進行加密。</p>

選項設定	有效值	預設值	描述
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	<p>用戶端所使用之加密演算法的清單。用戶端會依序嘗試每一種演算法來試圖解密伺服器輸入，直到演算法成功或達到清單結尾為止。</p> <p>Amazon RDS 會使用來自 Oracle 的下列預設清單。RDS 會從 RC4_256 開始，並依序向下處理清單。您可以變更順序或限制資料庫執行個體將接受的演算法。</p> <ol style="list-style-type: none"> 1. RC4_256 : RSA RC4 (256 位元金鑰大小) 2. AES256 : AES (256 位元金鑰大小) 3. AES192 : AES (192 位元金鑰大小) 4. 3DES168 : 3 金鑰三重 DES (112 位元有效金鑰大小) 5. RC4_128 : RSA RC4 (128 位元金鑰大小) 6. AES128 : AES (128 位元金鑰大小) 7. 3DES112 : 2 金鑰三重 DES (80 位元有效金鑰大小) 8. RC4_56 : RSA RC4 (56 位元金鑰大小) 9. DES : 標準 DES (56 位元金鑰大小) 10. RC4_40 : RSA RC4 (40 位元金鑰大小) 11. DES40 : DES40 (40 位元金鑰大小) <p>您可以指定一值或逗號分隔的值清單。如果您使用逗號，請不要在逗號後面插入空格；否則，您會收到 InvalidParameterValue 錯誤。</p>

選項設定	有效值	預設值	描述
			此參數和 SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER 必須有一個共同的密碼。

選項設定	有效值	預設值	描述
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	<p>資料庫執行個體所使用之加密演算法的清單。資料庫執行個體會依序使用每一個演算法，以嘗試解密用戶端輸入，直到演算法成功或直到達到清單結尾。</p> <p>Amazon RDS 會使用來自 Oracle 的下列預設清單。您可以變更順序或限制用戶端將接受的演算法。</p> <ol style="list-style-type: none"> 1. RC4_256 : RSA RC4 (256 位元金鑰大小) 2. AES256 : AES (256 位元金鑰大小) 3. AES192 : AES (192 位元金鑰大小) 4. 3DES168 : 3 金鑰三重 DES (112 位元有效金鑰大小) 5. RC4_128 : RSA RC4 (128 位元金鑰大小) 6. AES128 : AES (128 位元金鑰大小) 7. 3DES112 : 2 金鑰三重 DES (80 位元有效金鑰大小) 8. RC4_56 : RSA RC4 (56 位元金鑰大小) 9. DES : 標準 DES (56 位元金鑰大小) 10. RC4_40 : RSA RC4 (40 位元金鑰大小) 11. DES40 : DES40 (40 位元金鑰大小) <p>您可以指定一值或逗號分隔的值清單。如果您使用逗號，請不要在逗號後面插入空格；否則，您會收到 InvalidParameterValue 錯誤。</p>

選項設定	有效值	預設值	描述
			此參數和 <code>SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER</code> 必須有一個共同的密碼。

新增原生網路加密選項

將 `NATIVE_NETWORK_ENCRYPTION` 選項新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

當選項群組為作用中時，NNE 就會是作用中的狀態。

若要使用 [新增網路加密] 選項至資料庫執行個體 AWS Management Console

1. 針對 Engine (引擎)，選擇您想要使用的 Oracle 版本。所有版本都支援 NNE。
2. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需詳細資訊，請參閱 [建立選項群組](#)。

3. 將「網路加密」選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。

Note

新增 `NATIVE_NETWORK_COMPARY` 選項之後，您不需要重新啟動資料庫執行個體。只要選項群組為作用中，NNE 就會是作用中的狀態。

4. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。新增 `NATIVE_NETWORK_` 加密選項後，您不需要重新啟動資料庫執行個體。只要選項群組為作用中，NNE 就會是作用中的狀態。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

在 sqlnet.ora 中設定 NNE 值

您可以透過 Oracle 原生網路加密，在伺服器端和用戶端設定網路加密。用戶端是用來連線至資料庫執行個體的電腦。您可以在 sqlnet.ora 中指定下列用戶端設定：

- SQLNET.ALLOW_WEAK_CRYPT0
- SQLNET.ALLOW_WEAK_CRYPT0_CLIENTS
- SQLNET.CRYPT0_CHECKSUM_CLIENT
- SQLNET.CRYPT0_CHECKSUM_TYPES_CLIENT
- SQLNET.ENCRYPTION_CLIENT
- SQLNET.ENCRYPTION_TYPES_CLIENT

如需相關資訊，請參閱 Oracle 文件中的[配置 Oracle 伺服器 and 用戶端的網路資料加密和完整性](#)。

有時，資料庫執行個體會拒絕來自應用程式的連線請求。例如，當用戶端和伺服器上的加密演算法不相符時，可能會出現拒絕情況。若要測試 Oracle 原生網路加密，請將下列幾行新增至用戶端上的 sqlnet.ora 檔案：

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

嘗試連線時，上述幾行會在用戶端上產生一個稱為 /tmp/nettrace* 的追蹤檔。追蹤檔案會包含連線的相關資訊。如需您在使用 Oracle 原生網路加密時所發生之連線相關問題的詳細資訊，請參閱 Oracle 資料庫文件中的[關於溝通加密和完整性](#)。

修改原生網路加密選項設定

啟用 NATIVE_NETWORK_ENCRYPTION 選項後，您可以修改其設定。目前，您只能使用 AWS CLI 或 RDS API 修改 NATIVE_NETWORK_ENCRYPTION 選項設定。您無法使用主控台。下列範例會修改選項中的兩個設定。

```
aws rds add-option-to-option-group \
  --option-group-name my-option-group \
  --options
  "OptionName=NATIVE_NETWORK_ENCRYPTION,OptionSettings=[{Name=SQLNET.CRYPT0_CHECKSUM_TYPES_SERVER,Value=SHA256},
  {Name=SQLNET.CRYPT0_CHECKSUM_TYPES_SERVER,Value=SHA256}]" \
```

```
--apply-immediately
```

若要了解如何使用 CLI 修改選項設定，請參閱 [AWS CLI](#)。如需每項設定的詳細資訊，請參閱[原生網路加密選項設定](#)。

主題

- [修改 CRYPTO_CHECKSUM_* 值](#)
- [修改 ALLOW_WEAK_CRYPTOA* 設定](#)

修改 CRYPTO_CHECKSUM_* 值

如果您修改 NATIVE_NETWORK_COMPARY 選項設定，請確定下列選項設定至少具有一個通用密碼：

- SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

下列範例顯示您可以修改 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER 的情況。此組態有效，因為 CRYPTO_CHECKSUM_TYPES_CLIENT 和 CRYPTO_CHECKSUM_TYPES_SERVER 兩者皆使用 SHA256。

選項設定	修改前的值	修改後的值
SQLNET.CRYPTO_CHEC KSUM_TYPES_CLIENT	SHA256 , SHA384, SHA512	未變更
SQLNET.CRYPTO_CHEC KSUM_TYPES_SERVER	SHA256 , SHA384, SHA512, SHA1, MD5	SHA1, MD5, SHA256

另一個範例，假設您想要將 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER 從其預設設定修改為 SHA1, MD5。在這種情況下，請確認您將 SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT 設定為 SHA1 或 MD5。這些演算法不會包含在 SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT 的預設值中。

修改 ALLOW_WEAK_CRYPTOA* 設定

要將 SQLNET.ALLOW_WEAK_CRYPTOA* 選項從預設值設定為 FALSE，請確保符合下列條件：

- `SQLNET.ENCRYPTION_TYPES_SERVER` 和 `SQLNET.ENCRYPTION_TYPES_CLIENT` 具有一個相符的安全加密方法。如果方法不是 DES、3DES 或 RC4 (所有金鑰長度)，則該方法被認為是安全的。
- `SQLNET.CHECKSUM_TYPES_SERVER` 和 `SQLNET.CHECKSUM_TYPES_CLIENT` 具有一個相符的檢查總和方法。如果方法不是 MD5，則該方法被認為是安全的。
- 用戶端已使用 2021 年 7 月 PSU 修補。如果用戶端未修補，則用戶端會失去連線並接收到 ORA-12269 錯誤。

下列範例顯示範例 NNE 設定。假定將 `SQLNET.ENCRYPTION_TYPES_SERVER` 和 `SQLNET.ENCRYPTION_TYPES_CLIENT` 設定為 `FALSE`，從而阻止非安全連線。檢查總和選項設定滿足先決條件，因為它們都具有 SHA256。但是，`SQLNET.ENCRYPTION_TYPES_CLIENT` 和 `SQLNET.ENCRYPTION_TYPES_SERVER` 使用 DES、3DES 以及 RC4 加密方法，這些方法是不安全的。因此，要將 `SQLNET.ALLOW_WEAK_CCRYPTO*` 選項設定為 `FALSE`，請先將 `SQLNET.ENCRYPTION_TYPES_SERVER` 和 `SQLNET.ENCRYPTION_TYPES_CLIENT` 設為安全加密方法，例如 AES256。

選項設定	值
<code>SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT</code>	SHA256, SHA384, SHA512
<code>SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER</code>	SHA1, MD5, SHA256
<code>SQLNET.ENCRYPTION_TYPES_CLIENT</code>	RC4_256, 3DES168, DES40
<code>SQLNET.ENCRYPTION_TYPES_SERVER</code>	RC4_256, 3DES168, DES40

移除原生網路加密選項

您可以從資料庫執行個體中移除 NNE。

若要從資料庫執行個體中移除 `NATIVE_NETWORK_ENCRYPTION` 選項，請執行下列其中一個動作：

- 若要從多個資料庫執行個體移除 `NATIVE_NETWORK_ENCRYPTION` 選項，請從它們所屬的選項群組中移除該選項。此變更會影響使用選項群組的所有資料庫執行個體。移

除NATIVE_NETWORK_ENCRYPTION選項後，您不需要重新啟動資料庫執行個體。如需詳細資訊，請參閱 [從選項群組移除選項](#)。

- 若要從單一資料庫執行個體移除選項，請修改資料庫執行個體，並指定不包含該選項的其他選NATIVE_NETWORK_ENCRYPTION項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。移除 NATIVE_NETWORK_ENCRYPTION 選項之後，就不需要重新啟動資料庫執行個體。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle OLAP

Amazon RDS 會使用 OLAP 選項，藉此支援 Oracle OLAP。此選項提供 Oracle 資料庫執行個體的線上分析處理 (OLAP)。您可以使用 Oracle OLAP，依據 OLAP 標準建立維度物件和立方體來分析大量的資料。如需詳細資訊，請參閱 [Oracle 文件](#)。

Important

如果您使用 Oracle OLAP，如果有通用漏洞評估系統 (CVSS) 分數 9 分以上的安全漏洞或其他公布的安全漏洞，Amazon RDS 會將您的資料庫執行個體自動更新為最新的 Oracle PSU。

Amazon RDS 支援甲骨文資料庫 19c 及更高版本企業版的甲骨文 OLAP。

Oracle OLAP 的先決條件

下列是使用 Oracle OLAP 的先決條件：

- 您必須擁有來自 Oracle 的 Oracle OLAP 授權。如需詳細資訊，請參閱 Oracle 文件中的 [授權資訊](#)。
- 您的資料庫執行個體必須具有足夠的執行個體類別。Oracle OLAP 不支援 db.t3.micro 或 db.t3.small 資料庫執行個體類別。如需更多詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。
- 您的資料庫執行個體必須將自動次要版本升級啟用。此選項可讓資料庫執行個體自動接收可用的次要資料庫引擎版本升級，而且是安裝 Oracle Java Virtual Machine (JVM) 之任何選項的必要項目。Amazon RDS 使用此選項將您的資料庫執行個體更新為最新版 Oracle 修補程式集更新 (PSU) 或版本更新 (RU)。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
- 您的資料庫執行個體不能有名為 OLAPSYS 的使用者。如果是，OLAP 選項安裝就會失敗。

Oracle OLAP 的最佳實務

下列是使用 Oracle OLAP 的最佳實務：

- 為了將安全性提升至最高，請使用 OLAP 選項搭配 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。
- 設定資料庫執行個體以限制對資料庫執行個體的存取。如需更多詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 及 [在 VPC 中使用資料庫執行個體](#)。

新增 Oracle OLAP 選項

下列是將 OLAP 選項新增至資料庫執行個體的一般程序：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在新增 OLAP 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。在您新增選項之後，就不需要重新啟動資料庫執行個體。只要選項群組為作用中，Oracle OLAP 就可使用。

將 OLAP 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle 版本。
 - 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 OLAP 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

使用 Oracle OLAP

在啟用 Oracle OLAP 選項之後，您可以開始使用它。如需 Oracle OLAP 支援的功能清單，請參閱 [Oracle 文件](#)。

移除 Oracle OLAP 選項

捨棄使用該 OLAP 選項提供之資料類型的所有物件後，您可以從資料庫執行個體移除該選項。如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在移除 OLAP 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。移除 OLAP 選項之後，就不需要重新啟動資料庫執行個體。

捨棄 OLAP 選項

1. 備份您的資料。

Warning

如果執行個體使用已啟用做為選項一部分的資料類型，而且如果您移除該 OLAP 選項，則可能會遺失資料。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

2. 檢查是否有任何現有物件參考資料類型或 OLAP 選項的功能。
3. 捨棄參考資料類型或 OLAP 選項功能的所有物件。
4. 請執行下列其中一項：
 - 從其所屬的選項群組中移除 OLAP 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
 - 修改資料庫執行個體，並指定不包括 OLAP 選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Secure Sockets Layer

您可以在與資料庫執行個體相關聯的選項群組中新增 Oracle SSL 選項，就能對 RDS for Oracle 資料庫執行個體啟用 SSL 加密。Amazon RDS 會使用第二個連接埠 (依 Oracle 要求) 作為 SSL 連線用途。此方法可讓資料庫執行個體與 SQL*Plus 之間同時進行純文字和 SSL 加密的通訊。例如，您可以使用連接埠搭配純文字通訊與 VPC 內的其他資源通訊，同時使用連接埠搭配 SSL 加密的通訊與 VPC 外部的資源通訊。

Note

您可以在相同的 RDS for Oracle 資料庫執行個體上使用 SSL 或原生網路加密 (NNE)，但不能同時使用這兩者。如果您使用 SSL 加密，請務必關閉任何其他連線加密。如需詳細資訊，請參閱 [Oracle 原生網路加密](#)。

SSL/TLS 和 NNE 不再是 Oracle 進階安全性的一部分。您可以對下列 Oracle 資料庫版本的所有版本使用 SSL 加密：

- Oracle Database 21c (21.0.0)
- Oracle 資料庫 19c (19.0.0)

Oracle SSL 選項的 TLS 版本

Amazon RDS for Oracle 現已支援 Transport Layer Security (TLS) 版本 1.0 和 1.2。新增 Oracle SSL 選項時，請明確地將 `SQLNET.SSL_VERSION` 設為有效值。此選項設定允許下列值：

- "1.0" – 用戶端只能使用 TLS 1.0 版連線至資料庫執行個體。若為現有的 Oracle SSL 選項，`SQLNET.SSL_VERSION` 會自動設為 "1.0"。您可以視需要變更此設定。
- "1.2" – 用戶端只能使用 TLS 1.2 來連線至資料庫執行個體。
- "1.2 or 1.0" – 用戶端可以使用 TLS 1.2 或 1.0 來連線至資料庫執行個體。

Oracle SSL 選項的密碼套件

Amazon RDS for Oracle 支援多個 SSL 密碼套件。根據預設，Oracle SSL 選項設定為使用 `SSL_RSA_WITH_AES_256_CBC_SHA` 密碼套件。若要指定在 SSL 連線上使用不同的密碼套件，請使用 `SQLNET.CIPHER_SUITE` 選項設定。

下表總結列出所有版本的 Oracle 資料庫 19c 和 21c 中對於 Oracle 的 RDS 版本的 SSL 支援。

密碼套件 (SQLNET.CIPHER_SUITE)	TLS 版本支援 (SQLNET.SSL_VERSION)	FIPS 支援	FedRAMP 合規
SSL_RSA_WITH_AES_256_CBC_SHA (預設)	1.0 和 1.2	是	否
SSL_RSA_WITH_AES_256_CBC_SHA256	1.2	是	否
SSL_RSA_WITH_AES_256_GCM_SHA384	1.2	是	否
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	1.2	是	是
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	1.2	是	是
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	1.2	是	是
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	1.2	是	是
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	1.2	是	是
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	1.2	是	是

FIPS 支援

RDS for Oracle 可讓您針對 140-2 使用美國聯邦資訊處理標準 (FIPS)。FIPS 140-2 是定義加密模組安全性要求的美國政府標準。您可以針對 Oracle SSL 選項，將設定 `FIPS.SSLFIPS_140` 設為 `TRUE`，來開啟 FIPS 標準。針對 SSL 設定 FIPS 140-2 時，密碼編譯程式庫會在用戶端與 RDS for Oracle 資料庫執行個體之間加密資料。

用戶端必須使用符合 FIPS 規定的密碼套件。建立連線時，用戶端和 RDS for Oracle 資料庫執行個體會協商來回傳輸訊息時所使用的密碼套件。[Oracle SSL 選項的密碼套件](#) 中的資料表會顯示適用於每個

TLS 版本的 FIPS 相容 SSL 密碼套件。如需詳細資訊，請參閱 Oracle 資料庫文件中的 [Oracle 資料庫 FIPS 140-2 設定](#)。

新增 SSL 選項

若要使用 SSL，您的 RDS for Oracle 資料庫執行個體必須與包含 SSL 選項的選項群組建立關聯。

主控台

將 SSL 選項新增至選項群組

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 SSL 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 SSL 選項新增至選項群組。

如果您只想針對 SSL 連線使用經 FIPS 驗證的密碼套件，請將選項 FIPS.SSLFIPS_140 設定為 TRUE。如需 FIPS 標準的相關資訊，請參閱 [FIPS 支援](#)。

如需將選項新增至選項群組的相關資訊，請參閱[將選項新增至選項群組](#)。

3. 建立新的 RDS for Oracle 資料庫執行個體，將其與選項群組建立關聯，或修改 RDS for Oracle 資料庫執行個體，使其與選項群組建立關聯。

如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

如需修改資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

將 SSL 選項新增至選項群組

1. 建立新的選項群組，或識別現有的選項群組，以便在其中新增 SSL 選項。

如需建立選項群組的相關資訊，請參閱[建立選項群組](#)。

2. 將 SSL 選項新增至選項群組。

指定下列選項設定：

- Port – SSL 連接埠號碼
- VpcSecurityGroupMemberships – 已啟用選項的 VPC 安全群組

- `SQLNET.SSL_VERSION` – 用戶端可用來連線至資料庫執行個體的 TLS 版本

例如，下列 AWS CLI 指令會將 SSL 選項新增至名為的選項群組 `ora-option-group`。

Example

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group --option-group-name ora-option-group \  
  --options  
  'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

在 Windows 中：

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^  
  --options  
  'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

3. 建立新的 RDS for Oracle 資料庫執行個體，將其與選項群組建立關聯，或修改 RDS for Oracle 資料庫執行個體，使其與選項群組建立關聯。

如需建立資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

如需修改資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

設定 SQL*Plus 以使用 SSL 搭配 RDS for Oracle 資料庫執行個體

在可以連線至使用 Oracle SSL 選項的 RDS for Oracle 資料庫執行個體之前，您必須先設定 SQL*Plus，然後才能連線。

Note

若要允許從適當的用戶端存取資料庫執行個體，請確定您的安全群組已正確設定。如需更多詳細資訊，請參閱 [使用安全群組控制存取](#)。此外，這些指示僅適用於 SQL*Plus 和其他直接使用 Oracle 主目錄的用戶端。如需了解 JDBC 連線，請參閱 [設定透過 JDBC 的 SSL 連線](#)。

設定 SQL*Plus 以使用 SSL 連線至 RDS for Oracle 資料庫執行個體

1. 將 ORACLE_HOME 環境變數設為 Oracle 主目錄的位置。

Oracle 主目錄的路徑取決於您的安裝。下列範例設定 ORACLE_HOME 環境變數。

```
prompt>export ORACLE_HOME=/home/user/app/user/product/19.0.0/dbhome_1
```

如需有關設定 Oracle 環境變數的資訊，請參閱 Oracle 文件中的 [SQL*Plus 環境變數](#)，以及您的作業系統適用的 Oracle 安裝指南。

2. 將 \$ORACLE_HOME/lib 附加至 LD_LIBRARY_PATH 環境變數。

下列範例設定 LD_LIBRARY_PATH 環境變數。

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. 在 \$ORACLE_HOME/ssl_wallet 建立 Oracle 錢包的目錄。

下列範例建立 Oracle 錢包目錄。

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. 下載適用於所有人的憑證組合 .pem 檔案，AWS 區域 並將檔案放在 ssl_wallet 目錄中。如需相關資訊，請參閱。
5. 在 \$ORACLE_HOME/network/admin 目錄中，修改或建立 tnsnames.ora 檔案，並包含下列項目。

```
net_service_name =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS =  
        (PROTOCOL = TCPS)  
        (HOST = endpoint)  
        (PORT = ssl_port_number)  
      )  
    )  
    (CONNECT_DATA =  
      (SID = database_name)  
    )  
    (SECURITY =
```

```
(SSL_SERVER_CERT_DN =
"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=endpoint")
)
)
```

- 在相同的目錄中，修改或建立 `sqlnet.ora` 檔案，並加入下列參數。

Note

若要透過 TLS 安全連線與實體之間進行通訊，Oracle 需要擁有必要憑證的錢包以進行身分驗證。您可以使用 Oracle 的 ORAPKI 公用程式來建立並維護 Oracle 錢包，如步驟 7 所示。如需詳細資訊，請參閱 Oracle 文件中的[使用 ORAPKI 設定 Oracle 錢包](#)。

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
$ORACLE_HOME/ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

Note

若資料庫執行個體支援，您可將 `SSL_VERSION` 設為較高的值。

- 執行下列命令以建立 Oracle 公事包。

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
```

- 使用 OS 公用程式，將 `.pem` 服務包檔案中的每個憑證解壓縮至個別的 `.pem` 檔案中。
- 使用不同的 `orapki` 命令將每個憑證新增至您的公事包，並以 `.pem` 檔案的絕對檔案名稱取代 `certificate-pem-file`。

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
certificate-pem-file -auto_login_only
```

如需詳細資訊，請參閱 [輪換您的 SSL/TLS 憑證](#)。

使用 SSL 連線至 RDS for Oracle 資料庫執行個體

依先前所述將 SQL*Plus 設定為使用 SSL 之後，即可採用 SSL 選項連線至 RDS for Oracle 資料庫執行個體。(選用) 您可以先匯出指向包含 tnsnames.ora 和 sqlnet.ora 檔案目錄的 TNS_ADMIN 值。如此將確保 SQL*Plus 可一直找到這些檔案。以下範例將匯出 TNS_ADMIN 值。

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

連線到資料庫執行個體。例如，您可以使用 SQL*Plus 和 tnsnames.ora 檔案中的 `<net_service_name>` 進行連線。

```
sqlplus mydbuser@net_service_name
```

您也可以使用以下命令，只利用 SQL*Plus 連線至資料庫執行個體，而不需要用到 tnsnames.ora 檔案。

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = endpoint) (PORT = ssl_port_number))(CONNECT_DATA = (SID = database_name)))'
```

您也可以不使用 SSL 就連線至 RDS for Oracle 資料庫執行個體。例如，以下命令透過純文字連接埠來連線至資料庫執行個體，而不使用 SSL 加密。

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = endpoint) (PORT = port_number))(CONNECT_DATA = (SID = database_name)))'
```

如果您要禁止傳輸控制通訊協定 (TCP) 連接埠存取，請建立不含 IP 地址傳入的安全群組，並新增至執行個體。如此新增之後就會關閉透過 TCP 連接埠的連線，但仍允許透過 SSL 連接埠的連線 (從 SSL 選項安全群組所允許範圍內的 IP 地址指定)。

設定透過 JDBC 的 SSL 連線

若要透過 JDBC 使用 SSL 連線，您必須建立金鑰存放區、信任 Amazon RDS 根 CA 憑證，以及使用如下指定的程式碼片段。

要創建 JKS 格式的密鑰庫，可以使用以下命令。如需有關建立金鑰儲存庫的詳細資訊，請參閱 Oracle 說明文件中的 [< 建立金鑰儲存庫 >](#)。有關參考信息，請參閱 Java 平台中的 [密鑰工具](#)，標準版工具參考。

```
keytool -genkey -alias client -validity 365 -keyalg RSA -keystore clientkeystore
```

請執行下列步驟來信任 Amazon RDS 根 CA 憑證。

信任 Amazon RDS 根 CA 憑證

1. 下載適用於所有人的憑證組合 .pem 檔案，AWS 區域 並將檔案放在 `ssl_wallet` 目錄中。
如需有關下載憑證的詳細資訊，請參閱。
2. 使用 OS 公用程式，將 .pem 檔案中的每個憑證解壓縮至不同的檔案。
3. 使用單獨的 `openssl` 命令將每個證書轉換為 .der 格式，用 `## .pem ##### .pem ##` 的名稱（不帶 .pem 擴展名）。

```
openssl x509 -outform der -in certificate-pem-file.pem -out certificate-pem-file.der
```

4. 使用下列命令將每個憑證匯入金鑰儲存庫。

```
keytool -import -alias rds-root -keystore clientkeystore.jks -file certificate-pem-file.der
```

如需詳細資訊，請參閱 [輪換您的 SSL/TLS 憑證](#)。

5. 確認金鑰存放區已成功建立。

```
keytool -list -v -keystore clientkeystore.jks
```

在提示出現時，請輸入金鑰存放區密碼。

下列程式碼範例示範如何使用 JDBC 來設定 SSL 連線。

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "dns-name-provided-by-amazon-rds";
```

```
private static final Integer SSL_PORT = "ssl-option-port-configured-in-option-  
group";  
private static final String DB_SID = "oracle-sid";  
private static final String DB_USER = "user-name";  
private static final String DB_PASSWORD = "password";  
// This key store has only the prod root ca.  
private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";  
private static final String KEY_STORE_PASS = "keystore-password";  
  
public static void main(String[] args) throws SQLException {  
    final Properties properties = new Properties();  
    final String connectionString = String.format(  
        "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=  
%d))(CONNECT_DATA=(SID=%s)))",  
        DB_SERVER_NAME, SSL_PORT, DB_SID);  
    properties.put("user", DB_USER);  
    properties.put("password", DB_PASSWORD);  
    properties.put("oracle.jdbc.J2EE13Compliant", "true");  
    properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);  
    properties.put("javax.net.ssl.trustStoreType", "JKS");  
    properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);  
    final Connection connection = DriverManager.getConnection(connectionString,  
properties);  
    // If no exception, that means handshake has passed, and an SSL connection can  
be opened  
    }  
}
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

使用 SSL 連線強制執行 DN 比對

您可以使用 Oracle 參數 `SSL_SERVER_DN_MATCH`，以強制資料庫伺服器的辨別名稱 (DN) 比對其服務名稱。如果強制使用比對驗證，則 SSL 會確保憑證來自伺服器。如果不強制使用比對驗證，則 SSL 會執行檢查，但無論是否符合都允許連線。如果您不強制執行匹配，則允許伺服器可能偽造其身份。

若要強制使用 DN 比對，請新增 DN 比對屬性，並使用如下指定的連線字串。

將此屬性新增至用戶端連線，以強制使用 DN 比對。

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

使用下列連線字串，即可在使用 SSL 時強制使用 DN 比對。

```
final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s)))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
    \"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",
    DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

對 SSL 連線進行疑難排解

您可能會查詢資料庫並收到 ORA-28860 錯誤。

```
ORA-28860: Fatal SSL error
28860. 00000 - "Fatal SSL error"
*Cause: An error occurred during the SSL connection to the peer. It is likely that this
side sent data which the peer rejected.
*Action: Enable tracing to determine the exact cause of this error.
```

當用戶端嘗試使用伺服器不支援的 TLS 版本進行連線時，就會發生這個錯誤。若要避免此錯誤，請編輯 `sqlnet.ora` 並將 `SSL_VERSION` 設定為正確的 TLS 版本。如需詳細資訊，請參閱「My Oracle 支援」中的 [My Oracle 支援文件 2748438.1](#)。

Oracle Spatial

Amazon RDS 會使用 SPATIAL 選項，藉此支援 Oracle Spatial。Oracle Spatial 提供的 SQL 結構描述和函數，有助於 Oracle 資料庫中空間資料集合的儲存、擷取、更新和查詢。如需詳細資訊，請參閱 Oracle 文件中的 [Spatial 概念](#)。

Important

如果您使用 Oracle Spatial，Amazon RDS 會在下列任一情況存在時，自動將資料庫執行個體更新至最新 Oracle PSU：

- 具有通用漏洞評分系統 (CVSS) 分數的 9+ 的安全漏洞
- 其他已宣布的安全漏洞

Amazon RDS 僅可在 Oracle Enterprise Edition (EE) 和 Oracle Standard Edition 2 (SE2) 中支援 Oracle Spatial。下表顯示支援 EE 和 SE2 的資料庫引擎版本。

甲骨文数据库	企業版本	Standard Edition 2
21.0.0.0，所有版本	是	是
19.0.0.0，所有版本	是	是

Note

在 Oracle 資料庫 19c 中，空間修補程式服務包與資料庫修正程式集更新 (PSU) 和發行版本更新 (RU) 不同。適用於甲骨文的 RDS 不支持應用空間批處理包。

Oracle Spatial 的先決條件

下列是使用 Oracle Spatial 的先決條件：

- 確認您的資料庫執行個體有足夠的執行個體類別。Oracle Spatial 不支援 db.t3.micro 或 db.t3.small 資料庫執行個體類別。如需更多詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。
- 確認您的資料庫執行個體已啟用自動次要版本升級。此選項可讓資料庫執行個體自動接收可用的次要資料庫引擎版本升級，而且是安裝 Oracle Java Virtual Machine (JVM) 之任何選項的必要項

目。Amazon RDS 使用此選項將您的資料庫執行個體更新為最新版 Oracle 修補程式集更新 (PSU) 或版本更新 (RU)。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Spatial 的最佳實務

下列是使用 Oracle Spatial 的最佳實務：

- 為了將安全性提升至最高，請使用 SPATIAL 選項搭配 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。
- 設定資料庫執行個體以限制對資料庫執行個體的存取。如需更多詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#) 及 [在 VPC 中使用資料庫執行個體](#)。

新增 Oracle Spatial 選項

下列是將 SPATIAL 選項新增至資料庫執行個體的一般程序：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在新增 SPATIAL 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。在您新增選項之後，就不需要重新啟動資料庫執行個體。只要選項群組為作用中，Oracle Spatial 就會可用。

Note

在中斷過程中，密碼驗證功能將會短暫停用。在中斷過程中，您也可以預期將會看到與密碼驗證功能相關的事件。在 Oracle 資料庫執行個體可用之前，密碼驗證功能將會再次啟用。

將 SPATIAL 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle 版本。

b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 SPATIAL 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。

3. 將選項群組套用至新的或現有的資料庫執行個體：

- 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
- 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

移除 Oracle Spatial 選項

捨棄使用 SPATIAL 選項所提供之資料類型的所有物件之後，您可以將該選項從資料庫執行個體捨棄。如果 Oracle Java Virtual Machine (JVM) 未安裝在資料庫執行個體上，則在移除 SPATIAL 選項時，會發生短暫當機現象。如果已經在資料庫執行個體上安裝 Oracle Java Virtual Machine (JVM)，則不會發生停機現象。移除 SPATIAL 選項之後，就不需要重新啟動資料庫執行個體。

捨棄 SPATIAL 選項

1. 備份您的資料。

Warning

如果執行個體使用已啟用做為選項一部分的資料類型，而且如果您移除該 SPATIAL 選項，則可能會遺失資料。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

2. 檢查是否有任何現有物件參考資料類型或 SPATIAL 選項的功能。

如果 SPATIAL 選項存在，則在套用沒有 SPATIAL 選項的新選項群組時，執行個體可能會停滯。您可以使用下列查詢以識別物件：

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
```

```
WHERE DATA_TYPE='SDO_GEOMETRY'  
AND OWNER <> 'MDSYS')  
ORDER BY 1,2,3,4;  
  
SELECT OWNER, TABLE_NAME, COLUMN_NAME  
FROM DBA_TAB_COLUMNS  
WHERE DATA_TYPE = 'SDO_GEOMETRY'  
AND OWNER <> 'MDSYS'  
ORDER BY 1,2,3;
```

3. 捨棄參考資料類型或 SPATIAL 選項功能的所有物件。
4. 請執行下列其中一項：
 - 從其所屬的選項群組中移除 SPATIAL 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
 - 修改資料庫執行個體，並指定不包括 SPATIAL 選項的不同選項群組。此變更會影響單一資料庫執行個體。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle SQLT

Amazon RDS 透過使用 SQLT 選項來支援 Oracle SQLTXPLAIN (SQLT)。您可以將 SQLT 與任何版本的甲骨文數據庫 19c 及更高版本一起使用。

Oracle EXPLAIN PLAN 陳述式可以決定 SQL 陳述式的執行計劃。它可以驗證 Oracle 最佳化工具是否選擇特定的執行計劃，例如巢狀迴圈聯結。還可協助您了解最佳化工具的決策，例如，為何選擇巢狀迴圈聯結，而非雜湊聯結。因此，EXPLAIN PLAN 協助您了解陳述式的效能。

SQLT 是可產生報告的 Oracle 公用程式。報告包含物件統計資料、物件中繼資料、最佳化工具相關的初始化參數，以及其他資訊，供資料庫管理員用來微調 SQL 陳述式以獲得最佳效能。SQLT 會產生 HTML 報告，內含報告中所有區段的超連結。

不同於「自動工作負載儲存區域」或 Statspack 報告，SQLT 以個別 SQL 陳述式為處理對象。SQLT 是 SQL、PL/SQL 和 SQL*Plus 檔案的集合，這些檔案收集、儲存並顯示效能資料。

以下是每個 SQLT 版本支援的 Oracle 版本。

SQLT 版本	Oracle Database 21c	Oracle 資料庫 19c
2018-07-25.v1	支援	支援
2018-03-31.v1	不支援	不支援
2016-04-29.v1	不支援	不支援

若要下載 SQLT 並取得使用的說明：

- 登入您的「我的 Oracle 支援」帳戶，並開啟下列文件：
- 若要下載 SQLT：[文件 215187.1](#)
- 關於 SQLT 使用指示：[文件 1614107.1](#)
- 關於 SQLT 的常見問答集：[文件 1454160.1](#)
- 有關如何閱讀 SQLT 輸出的相關資訊：[文件 1456176.1](#)
- 如需解譯主要報告：[文件 1922234.1](#)

Amazon RDS 不支援下列 SQLT 方法：

- XPLORE

- XHUME

SQLT 先決條件

下列是使用 SQLT 的先決條件：

- 您必須移除 SQLT 所需的使用者和角色 (若存在)。

SQLT 選項會在資料庫執行個體上建立下列使用者和角色：

- SQLTXPLAIN 使用者
- SQLTXADMIN 使用者
- SQLT_USER_ROLE 角色

如果您的資料庫執行個體有任何這些使用者或角色，請透過 SQL 用戶端登入資料庫執行個體，然後使用下列陳述式捨棄這些使用者或角色：

```
DROP USER SQLTXPLAIN CASCADE;  
DROP USER SQLTXADMIN CASCADE;  
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- 您必須移除 SQLT 所需的資料表空間 (若存在)。

SQLT 選項會在資料庫執行個體上建立下列資料表空間：

- RDS_SQLT_TS
- RDS_TEMP_SQLT_TS


如果您的資料庫執行個體有這些資料表空間，請使用 SQL 用戶端登入資料庫執行個體，然後捨棄這些資料表空間。

SQLT 選項設定

SQLT 可以搭配「Oracle 調校套件」和「Oracle 診斷套件」所提供的授權功能來使用。Oracle 調校套件包含 SQL Tuning Advisor，Oracle 診斷套件包含「自動工作負載儲存區域」。SQLT 設定可允許或禁止從 SQLT 存取這些功能。

Amazon RDS 支援 SQLT 選項的下列設定。

選項設定	有效值	預設值	描述
LICENSE_PACK	T, D, N	N	<p>您想要透過 SQLT 來存取的 Oracle 管理套件。輸入下列其中一個值：</p> <ul style="list-style-type: none"> • T 表示您有 Oracle 調校套件和 Oracle 診斷套件的授權，而且想要從 SQLT 存取 SQL Tuning Advisor 和「自動工作負載儲存區域」。 • D 表示您有 Oracle 診斷套件的授權，而且想要從 SQLT 存取「自動工作負載儲存區域」。 • N 表示您沒有 Oracle 調校套件和 Oracle 診斷套件的授權，或您有其中一個或兩者的授權，但不想讓 SQLT 存取這些功能。 <div data-bbox="954 982 1510 1732" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Amazon RDS 不提供這些 Oracle 管理套件的授權。如果您指出想要使用的套件不在資料庫執行個體中，您可以對資料庫執行個體使用 SQLT。不過，SQLT 無法存取此套件，而 SQLT 報告也不含此套件的資料。例如，若您指定 T，但資料庫執行個體不含 Oracle 調校套件，則 SQLT 可處理資料庫執行個體，但所產生的報告不含 Oracle 調校套件相關的資料。</p> </div>
VERSION	2016-04-29.v1	2016-04-29.v1	您要安裝的 SQLT 版本。

選項設定	有效值	預設值	描述
	2018-03-3 1.v1 2018-07-2 5.v1		 Note 對於 Oracle Database 19c 和 21c，唯一支援的版本是 2018-07-25.v1。此版本是這些版本的預設版本。

新增 SQLT 選項

下列是將 SQLT 選項新增至資料庫執行個體的一般程序：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將 SQLT 選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

在新增 SQLT 選項之後，只要選項群組為作用中，SQLT 就立即變成作用中。

將 SQLT 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇您想要使用的 Oracle 版本。所有版本都支援 SQLT 選項。
 - b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 SQLT 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

4. (選用) 在每一個含有 SQLT 選項的資料庫執行個體上驗證 SQLT 安裝。

- a. 以主要使用者身分使用 SQL 用戶端來連接到資料庫執行個體。

如需使用 SQL 用戶端連接至 Oracle 資料庫執行個體的相關資訊，請參閱[連線至 RDS for Oracle 資料庫執行個體](#)。

- b. 執行下列查詢：

```
SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```


該查詢會傳回 Amazon RDS 上 SQLT 選項的目前版本。12.1.160429 即為 Amazon RDS 上可用 SQLT 版本的範例。

5. 變更 SQLT 選項所建立之使用者的密碼。

- a. 以主要使用者身分使用 SQL 用戶端來連接到資料庫執行個體。

- b. 執行下列 SQL 陳述式來變更 SQLTXADMIN 使用者的密碼：

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

 Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

- c. 執行下列 SQL 陳述式來變更 SQLTXPLAIN 使用者的密碼：

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

Note

升級 SQLT 需要先解除安裝舊版的 SQLT，然後安裝新版本。因此，升級 SQLT 時會遺失所有 SQLT 中繼資料。資料庫的主要版本升級也會解除安裝並重新安裝 SQLT。主要版本升級的範例是從 Oracle 資料庫 19c 升級到 Oracle 資料庫 21c。

使用 SQLT

SQLT 可搭配 Oracle SQL*Plus 公用程式一起使用。

使用 SQLT

1. 從「我的 Oracle 支援」網站上的[文件 215187.1](#) 下載 SQLT .zip 檔案。

Note

您無法從「我的 Oracle 支援」網站下載 SQLT 12.1.160429。Oracle 已廢除這個較舊的版本。

2. 解壓縮 SQLT .zip 檔案。
3. 從命令列提示字元中，切換到檔案系統上的 sqlt/run 目錄。
4. 從命令提示字元中，開啟 SQL*Plus，並以主要使用者身分連接到資料庫執行個體。

如需使用 SQL*Plus 連接到資料庫執行個體的相關資訊，請參閱[連線至 RDS for Oracle 資料庫執行個體](#)。

5. 取得 SQL 陳述式的 SQL ID：

```
SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
```


輸出類似如下：

```
SQL_ID
-----
chvsmttqjzjkn
```

6. 使用 SQLT 來分析 SQL 陳述式：

```
START sqltextract.sql sql_id sqltexplain_user_password
```


例如，若為 SQL ID `chvsmttqjzjkn`，請輸入下列陳述式：

```
START sqltextract.sql chvsmttqjzjkn sqltexplain_user_password
```

SQLT 會在執行 SQLT 命令的目錄中產生 HTML 報告及相關的資源，並全部壓縮成一個 .zip 檔案。

7. (選用) 若要讓應用程式使用者利用 SQLT 來診斷 SQL 陳述式，請執行下列陳述式，將 `SQLT_USER_ROLE` 授予每個應用程式使用者：

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

 Note

Oracle 不建議以 SYS 使用者或具備 DBA 角色的使用者來執行 SQLT。最好將 `SQLT_USER_ROLE` 授予應用程式使用者，再以應用程式使用者的帳戶來執行 SQLT 診斷。

升級 SQLT 選項

使用 Amazon RDS for Oracle 之後，可以將 SQLT 選項從您現有的版本升級到更新版本。若要升級 SQLT 選項，請為新版 SQLT 完成 [使用 SQLT](#) 中的步驟 1–3。另外，如果您在該節的步驟 7 中授予舊版 SQLT 的權限，請再次對於新版 SQLT 授予該權限。

升級 SQLT 選項會導致失去舊版 SQLT 的中繼資料。舊版 SQLT 的結構描述和相關物件都會被捨棄，並安裝較新版的 SQLT。如需 SQLT 最新版變更的詳細資訊，請參閱「My Oracle Support」網站上的 [文件 1614201.1](#)。

Note

不支援版本降級。

修改 SQLT 設定

在啟用 SQLT 之後，您可以修改該選項的 LICENSE_PACK 和 VERSION 設定。

如需如何修改選項設定的詳細資訊，請參閱 [修改選項設定](#)。如需每項設定的詳細資訊，請參閱 [SQLT 選項設定](#)。

移除 SQLT 選項

您可以從資料庫執行個體中移除 SQLT。

若要從資料庫執行個體中移除 SQLT，請執行下列其中一個動作：

- 若要從多個資料庫執行個體中移除 SQLT，請從資料庫執行個體所屬的選項群組中移除 SQLT 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
- 若要從單一資料庫執行個體中移除 SQLT，請修改資料庫執行個體，並指定不包括 SQLT 選項的不同選項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Oracle Statspack

Oracle Statspack 選項會安裝和啟用 Oracle Statspack 效能統計資料功能。Oracle Statspack 是 SQL、PL/SQL 和 SQL*Plus 指令碼的集合，這些指令碼會收集、存放並顯示效能資料。如需有關使用 Oracle Statspack 的資訊，請參閱 Oracle 文件中的 [Oracle Statspack](#)。

Note

Oracle Statspack 不再由 Oracle 所支援，已替換為更進階的「自動工作負載資料儲存器 (AWR)」。只有已購買診斷套件的 Oracle Enterprise Edition 客戶才能使用 AWR。您可以將 Oracle Statspack 與 Amazon RDS 上的任何 Oracle 資料庫引擎搭配使用。您無法在 Amazon RDS 讀取複本上執行 Oracle Statspack。

設定 Oracle Statspack

若要執行 Statspack 指令碼，您必須新增 Statspack 選項。

設定 Oracle Statspack

1. 在 SQL 用戶端中，使用管理帳戶登入 Oracle 資料庫。
2. 根據是否已安裝 Statspack，執行下列其中一個動作：
 - 如果已安裝 Statspack，且 PERFSTAT 帳戶與 Statspack 相關聯，請跳至步驟 4。
 - 如果沒有安裝 Statspack，且 PERFSTAT 帳戶存在，請按以下方式捨棄帳戶：

```
DROP USER PERFSTAT CASCADE;
```

否則，嘗試新增 Statspack 選項會產生錯誤和 RDS-Event-0058。

3. 將 Statspack 選項新增至選項群組。請參閱[將選項新增至選項群組](#)。

Amazon RDS 會自動在資料庫執行個體上安裝 Statspack 指令碼，然後設定 PERFSTAT 帳戶。

4. 使用以下 SQL 陳述式重設密碼，用新密碼替換 `pwd`：

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

您可以使用 PERFSTAT 使用者帳戶登入並執行 Statspack 指令碼。

5. 使用下列陳述式將 CREATE JOB 權限授與 PERFSTAT 帳戶：

```
GRANT CREATE JOB TO PERFSTAT;
```

6. 確定已填入 PERFSTAT.STATS\$IDLE_EVENT 表格中的閒置等待事件。

由於 Oracle 錯誤 28523746，PERFSTAT.STATS\$IDLE_EVENT 中的閒置等待事件可能不會填入。若要確保所有閒置事件都可用，請執行下列陳述式：

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

產生 Statspack 報告

Statspack 報告會比較兩個快照。

產生 Statspack 報告

1. 在 SQL 用戶端中，使用 PERFSTAT 帳戶登入 Oracle 資料庫。
2. 使用下列其中一種技巧建立快照：
 - 手動建立 Statspack 快照。
 - 建立在指定時間間隔之後建立 Statspack 快照的任務。例如，下列工作每小時會建立一個 Statspack 快照：

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE
+1/24, 'HH24')');
COMMIT;
```

3. 使用下列查詢檢視快照：

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. 執行 Amazon RDS 程序 rdsadmin.rds_run_spreport，以快照 ID 取代 begin_snap 和 end_snap 快照：

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

例如，下列命令會根據 Statspack 快照 1 和 2 之間的時間建立報告：

```
exec rdsadmin.rds_run_spreport(1,2);
```

Statspack 報告的檔案名稱包含兩個快照的數目。例如，使用 Statspack 快照 1 和 2 所建立的報告檔案會命名為 ORCL_spreport_1_2.lst。

5. 監控輸出是否有錯誤。

Oracle Statspack 會在執行報表之前執行檢查。因此，您也可能會在命令輸出中看到錯誤訊息。例如，您可能會嘗試根據無效範圍產生報告，其中 Statspack 快照集的開始值大於結束值。在這種情況下，輸出會顯示錯誤訊息，但資料庫引擎不會產生錯誤檔案。

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

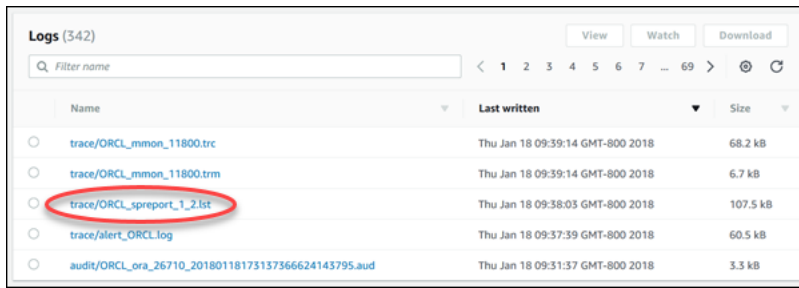
如果您使用 Statspack 快照的無效數字，則輸出會顯示錯誤。例如，如果您嘗試產生快照 1 和 50 的報告，但快照 50 不存在，則輸出會顯示錯誤。

```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
ORA-20000: Could not find both snapshot IDs
```

6. (選用)

若要擷取報表，請呼叫追蹤檔程序，如 [使用 Oracle 追蹤檔案](#) 中所述。

或者，從 RDS 主控台下載 Statspack 報告。移至資料庫執行個體詳細資訊的 Log (日誌) 區段，然後選擇 Download (下載)：



The screenshot shows the AWS Management Console Logs page for an Amazon RDS instance. The page title is 'Logs (342)'. There are buttons for 'View', 'Watch', and 'Download'. A search bar labeled 'Filter name' is present. Below the search bar is a table with columns for 'Name', 'Last written', and 'Size'. The table contains several log files, with 'trace/ORCL_spreport_1_2.lst' circled in red.

Name	Last written	Size
trace/ORCL_mmon_11800.trc	Thu Jan 18 09:39:14 GMT-800 2018	68.2 kB
trace/ORCL_mmon_11800.trm	Thu Jan 18 09:39:14 GMT-800 2018	6.7 kB
trace/ORCL_spreport_1_2.lst	Thu Jan 18 09:38:03 GMT-800 2018	107.5 kB
trace/alert_ORCL.log	Thu Jan 18 09:37:39 GMT-800 2018	60.5 kB
audit/ORCL_ora_26710_201801181751373566624143795.aud	Thu Jan 18 09:31:37 GMT-800 2018	3.3 kB

如果在產生報告時發生錯誤，則資料庫引擎使用相同的命名慣例做為報告，但擁有 .err 的副檔名。例如，若使用 Statspack 快照 1 和 7 建立報告時發生錯誤，則報告檔案會命名為 ORCL_spreport_1_7.err。您可以使用與標準快照報告相同的技巧來下載錯誤報告。

移除 Statspack 快照

若要移除 Oracle Statspack 快照的範圍，請使用下列命令：

```
exec statspack.purge(begin snap, end snap);
```

Oracle 時區

若要變更您 Oracle 資料庫執行個體所使用的系統時區，請使用時區選項。例如，您可以變更資料庫執行個體的時區，以與現場部署環境或舊型應用程式相容。時區選項會變更主機層級的時區。變更時區會影響所有日期欄和值，包括 SYSDATE 和 SYSTIMESTAMP。

時區選項不同於 `rdsadmin_util.alter_db_time_zone` 命令。`alter_db_time_zone` 命令只會變更特定資料類型的時區。時區選項會變更所有日期欄和值的時區。如需有關 `alter_db_time_zone` 的詳細資訊，請參閱 [設定資料庫時區](#)。如需升級考量的詳細資訊，請參閱 [時區考量](#)。

設定時區的限制

時區選項是永久且持續的選項。因此，您無法執行以下動作：

- 新增時區選項後，從選項群組移除選項。
- 在新增選項群組之後，從資料庫執行個體中將其移除。
- 將選項的時區設定修改為其他時區。

設定時區的建議

在您將時區選項新增至生產資料庫之前，強烈建議您執行下列動作：

- 拍攝您資料庫執行個體的快照。如果您不小心設定為錯誤的時區，則必須將您的資料庫執行個體復原為先前的時區設定。如需詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。
- 將時區選項新增至測試資料庫執行個體。新增時區選項可能導致資料表使用系統日期來新增日期或時間的問題。建議您在測試執行個體上分析資料和應用程式。如此一來，您就可以評估變更生產執行個體上時區的影響。

時區選項設定

Amazon RDS 支援時區選項的下列設定。

選項設定	有效值	描述
TIME_ZONE	其中一個可用的時區。如需完整清單，請參閱 可用時區 。	資料庫執行個體的新時區。

新增時區選項

請完成下列步驟，將時區選項新增至您的資料庫執行個體：

1. (建議) 建立資料庫執行個體的快照。
2. 執行下列其中一項工作：
 - 從頭開始建立新的選項群組。如需詳細資訊，請參閱 [建立選項群組](#)。
 - 使用 AWS CLI 或 API 複製現有的選項群組。如需詳細資訊，請參閱 [刪除選項群組](#)。
 - 重複使用現有的非預設選項群組。最佳做法是使用目前未與任何資料庫執行個體或快照相關聯的選項群組。
3. 從上一個步驟將新選項新增至選項群組。
4. 如果目前與資料庫執行個體關聯的選項群組已啟用選項，請將這些選項新增至新的選項群組。此策略可防止在啟用新選項時解除安裝現有選項。
5. 將新的選項群組新增至資料庫執行個體。

當您新增時區選項時，在資料庫執行個體自動重新啟動時會發生短暫的中斷。

主控台

將時區選項新增至選項群組，並將其與資料庫執行個體建立關聯

1. 在 RDS 主控台中，選擇 [選項群組]。
2. 選擇您要新增選項的選項群組名稱。
3. 選擇 Add option (新增選項)。
4. 針對 [選項名稱]，選擇 [時區]，然後設定選項設定。
5. 將選項群組與新的或現有的資料庫執行個體建立關聯：
 - 針對新的資料庫執行個體，在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。當您將新選項新增至現有的資料庫執行個體時，資料庫執行個體自動重新啟動時會發生短暫中斷。如需詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

下列範例會使用新 AWS CLI [增選項至選項群組](#) 命令，將Timezone選項和選項設定新增至名為的TIME_ZONE選項群組。myoptiongroup時區會設為 Africa/Cairo。

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name "myoptiongroup" \  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/  
Cairo}]" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name "myoptiongroup" ^  
  --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/  
Cairo}]" ^  
  --apply-immediately
```

修改時區設定

時區選項是永久且持續的選項。在新增此選項之後，您無法從選項群組中移除它。在新增選項群組之後，您無法從資料庫執行個體中移除它。您無法將選項的時區設定修改為不同的時區。如果您未正確地設定時區，則可以使用您在新增時區選項之前所取得的快照來復原資料庫執行個體。

移除時區選項

時區選項是永久且持續的選項。在新增此選項之後，您無法從選項群組中移除它。在新增選項群組之後，您無法從資料庫執行個體中移除它。若要移除設定時區選項，請使用您在新增時區選項之前所取得的快照來復原資料庫執行個體。

可用時區

您可對時區選項使用下列值。

區域	時區
非洲	非洲/開羅，非洲/卡薩布蘭卡，非洲/哈拉雷，非洲/拉哥斯，非洲/魯安達，非洲/蒙羅維亞，非洲/奈洛比，非洲/的黎波里，非洲/溫吐克

區域	時區
美洲	美洲/阿拉瓜伊納，美洲/阿根廷/布宜諾斯艾利斯，美洲/亞松森，美洲/波哥大，美洲/卡拉卡斯，美洲/芝加哥，美洲/赤瓦瓦州，美洲/古雅巴，美洲/丹佛，美洲/底特律，美洲/福塔力沙，美洲/哥特哈布，美洲/瓜地馬拉，美洲/哈利法克斯，美洲/利馬，美洲/洛杉磯，美洲/瑪瑙斯，美洲/馬塔莫羅斯，美洲/墨西哥市，美洲/蒙特利，美洲/蒙特維多，美洲/紐約州，美洲/鳳凰城，美洲/聖地牙哥，美洲/聖保羅，美洲/提華納，美洲/多倫多
亞洲	亞洲/安曼，亞洲/阿什哈巴德，亞洲/巴格達，亞洲/巴庫，亞洲/曼谷，亞洲/貝魯特，亞洲/加爾各答，亞洲/大馬士革，亞洲/達卡，亞洲/香港，亞洲/伊爾庫次克，亞洲/雅加達，亞洲/耶路撒冷，亞洲/喀布爾，亞洲/喀拉蚩，亞洲/加德滿都，亞洲/加爾各答，亞洲/克拉斯諾雅斯克，亞洲/馬加丹，亞洲/馬尼拉，亞洲/馬斯喀特，亞洲/新西伯利亞，亞洲/仰光，亞洲/利雅德，亞洲/首爾，亞洲/上海，亞洲/新加坡，亞洲/台北，亞洲/德黑蘭，亞洲/東京，亞洲/庫倫，亞洲/海參崴，亞洲/雅庫次克，亞洲/葉里溫
大西洋	大西洋/亞速爾群島，大西洋/維德角群島
澳大利亞	澳洲/阿德萊德，澳洲/布里斯班，澳洲/達爾文，澳洲/尤克拉，澳洲/荷巴特，澳洲/羅豪，澳洲/伯斯，澳洲/雪梨
巴西	巴西/，巴西/東DeNoronha部
加拿大	加拿大/紐芬蘭，加拿大/薩克其萬省
等等	等等/GMT-3
歐洲	歐洲/阿姆斯特丹，歐洲/雅典，歐洲/柏林，歐洲/都柏林，歐洲/赫爾辛基，歐洲/加里寧格勒，歐洲/倫敦，歐洲/馬德里，歐洲/莫斯科，歐洲/巴黎，歐洲/布拉格，歐洲/羅馬，歐洲/塞拉耶佛
太平洋	太平洋/亞庇，太平洋/奧克蘭，太平洋/查塔姆，太平洋/斐濟，太平洋/關島，太平洋/檀香山，太平洋/刻里提瑪斯，太平洋/馬克沙斯，太平洋/薩摩亞，太平洋/東加塔普，太平洋/威克
美國	美國/阿拉斯加，美國/中部，美國/東印第安納州，美國/東部，美國/太平洋
UTC	UTC

Oracle 時區檔案自動升級

您可以 `TIMEZONE_FILE_AUTOUPGRADE` 選擇將目前的時區檔案升級到 RDS for Oracle 資料庫執行個體的最新版本。

主題

- [Oracle 時區檔案的概觀](#)
- [更新時區檔案的策略](#)
- [時區檔案更新期間的停機時間](#)
- [準備更新時區檔案](#)
- [新增時區檔案自動升級選項](#)
- [更新時區檔案後，檢查您的資料](#)

Oracle 時區檔案的概觀

Oracle Database 時區檔案存放下列資訊：

- 與國際標準時間 (UTC) 的偏移
- 日光節約時間 (DST) 的轉換時間
- 標準時間和 DST 的縮寫

Oracle Database 提供多個版本的時區檔案。當您在內部部署環境中建立 Oracle 資料庫時，請選擇時區檔案版本。如需詳細資訊，請參閱《Oracle Database 全球化支援指南》中的[選擇時區檔案](#)。

如果 DST 的規則變更，Oracle 會發佈新的時區檔案。Oracle 會獨立於每季發行版本更新 (RU) 和發行更新修訂 (RU) 的排程，發行這些新時區檔案。時區檔案位於資料庫主機上的目錄 `$ORACLE_HOME/oracore/zoneinfo/` 中。時區檔案名稱使用格式 `DSTvversion`，如 `DSTv35` 中所示。

時區檔案如何影響資料傳輸

在 Oracle 資料庫中，`TIMESTAMP WITH TIME ZONE` 資料類型儲存時間戳記和時區資料。具 `TIMESTAMP WITH TIME ZONE` 資料類型的資料會使用相關聯時區檔案版本中的規則。因此，當您更新時區檔案時，現有 `TIMESTAMP WITH TIME ZONE` 資料會受到影響。

當您在使用不同版本時區檔案的資料庫之間傳輸資料時，可能會發生問題。例如，如果您從時區檔案版本高於目標資料庫的來源資料庫匯入資料，則資料庫就會發出 `ORA-39405` 錯誤。在以前，您必須使用下列任一技術來解決此錯誤：

- 利用所需的時區檔案建立一個 RDS for Oracle 資料庫執行個體，從來源資料庫匯出資料，接著將其匯入新的資料庫。
- 使用 AWS DMS 或邏輯複寫來移轉資料。

使用 TIMEZONE_FILE_AUTOUPGRADE 選項自動更新

當連接至 RDS for Oracle 資料庫執行個體的選項群組包含選 TIMEZONE_FILE_AUTOUPGRADE 項時，RDS 會自動更新您的時區檔案。藉由確保 Oracle 資料庫使用相同的時區檔案版本，您可以避免在不同環境之間移動資料時耗費時間的手動技巧。容器資料庫 (CDB) 和非 CDB 都支援 TIMEZONE_FILE_AUTOUPGRADE 選項。

將 TIMEZONE_FILE_AUTOUPGRADE 選項新增至選項群組時，您可以選擇立即新增選項還是在維護時段新增選項。資料庫執行個體套用新選項後，RDS 會檢查是否可以安裝較新的 DSTV ## 檔案。目標 DSTV`version` 取決於下列項目：

- 您的資料庫執行個體目前正在執行的次要引擎版本
- 您要將資料庫執行個體升級至其中的次要引擎版本

例如，您目前的時區檔案版本可能是 DSTv33。RDS 將更新套用至您的選項群組時，可能會判斷您的資料庫執行個體檔案系統目前有可用的 DSTv34。然後 RDS 會自動將您的時區檔案更新為 DSTv34。

若要在支援的 RDS 版本更新中尋找可用的 DST 版本，請查看 [適用於 Oracle 的 Amazon Relational Database Service \(Amazon RDS\) 版本備註](#) 中的修補程式。例如，[版本 19.0.0.0.ru-2022-10.rur-2022-10.r1](#) 列出修補程式 34533061: RDBMS - DSTV39 UPDATE - TZDATA2022C。

更新時區檔案的策略

升級您的數據庫引擎並將選 TIMEZONE_FILE_AUTOUPGRADE 項添加到選項組是單獨的操作。如果有更新的時區檔案可用，新增 TIMEZONE_FILE_AUTOUPGRADE 選項會啟動時區檔案的更新。您可以立即執行下列指令 (只會顯示相關選項)，或在下一個維護時段執行：

- 僅使用下列 RDS CLI 命令升級您的資料庫引擎：

```
modify-db-instance --engine-version name ...
```

- 僅使用下列 CLI 命令新增 TIMEZONE_FILE_AUTOUPGRADE 選項：

```
add-option-to-option-group --option-group-name name --options  
OptionName=TIMEZONE_FILE_AUTOUPGRADE ...
```

- 使用下列 CLI 命令升級資料庫引擎，並將新的選項群組新增至執行個體：

```
modify-db-instance --engine-version name --option-group-name name ...
```

您的更新策略取決於您是要一起升級資料庫和時區檔案，還是只執行這些作業之一。請記住，如果您更新選項組，然後在單獨的 API 操作中升級資料庫引擎，則當您升級資料庫引擎時，時區文件更新可能正在進行中。

本節中的範例假設如下情況：

- 您尚未新增 TIMEZONE_FILE_AUTOUPGRADE 至目前與資料庫執行個體相關聯的選項群組。
- 您的資料庫執行個體使用資料庫版本 19.0.0.0.ru-2019-07.rur-2019-07.r1 和時區檔案 DSTv33。
- 您的資料庫執行個體檔案系統包含檔案 DSTv34。
- 版本更新 19.0.0.0.ru-2022-10.rur-2022-10.r1 包含 DSTv35。

若要更新您的時區檔案，您可以使用下列策略。

主題

- [更新時區檔案，而不升級引擎](#)
- [升級時區檔案和資料庫引擎版本](#)
- [升級您的資料庫引擎版本，而不更新時區檔案](#)

更新時區檔案，而不升級引擎

在這個案例中，您的資料庫使用的是 DSTv33，但 DSTv34 可在資料庫執行個體檔案系統上使用。您想要將資料庫執行個體所使用的時區檔案從 DSTv33 更新為 DSTv34，但不想要將您的引擎升級至次要版本，其中包括 DSTv35。

在 add-option-to-option-group 命令中，新增 TIMEZONE_FILE_AUTOUPGRADE 至資料庫執行個體使用的選項群組。指定要立即新增選項還是將其延遲至維護時段。套用此 TIMEZONE_FILE_AUTOUPGRADE 選項之後，RDS 會執行下列動作：

1. 檢查是否有新的 DST 版本。

2. 判斷檔案系統上是否可使用 DSTv34。
3. 立即更新時區檔案。

升級時區檔案和資料庫引擎版本

在這個案例中，您的資料庫使用的是 DSTv33，但 DSTv34 可在資料庫執行個體檔案系統上使用。您想要將資料庫引擎升級至次要版本 19.0.0.0.ru-2022-10.rur-2022-10.r1 (其中包括 DSTv35)，並在引擎升級期間將時區檔案更新為 DSTv35。因此，您的目標是跳過 DSTv34 並將您的時區檔案直接更新為 DSTv35。

若要同時升級引擎和時區檔案，請 `modify-db-instance` 使用 `--option-group-name` 和選 `--engine-version` 項執行。您可以立即執行指令，或將其延遲至維護時段。在 `--option-group-name` 中，指定包含選項的 `TIMEZONE_FILE_AUTOUPGRADE` 選項群組。例如：

```
aws rds modify-db-instance
  --db-instance-identifier my-instance \
  --engine-version new-version \
  ---option-group-name og-with-timezone-file-autoupgrade \
  --apply-immediately
```

RDS 會開始將您的引擎升級至 19.0.0.Ru-10。套用此 `TIMEZONE_FILE_AUTOUPGRADE` 選項之後，RDS 會檢查是否有新的 DST 版本，並會看到 DSTv35 是否有提供，並立即開始對 DSTv35 的更新。

若要立即升級引擎，然後升級您的時區檔案，請依序執行作業：

1. 僅使用以下 CLI 命令升級您的數據庫引擎：

```
aws rds modify-db-instance \
  --db-instance-identifier my-instance \
  --engine-version new-version \
  --apply-immediately
```

2. 使用下列 CLI 指令，將選項新增至連接至執行個體的選項群組：`TIMEZONE_FILE_AUTOUPGRADE`

```
aws rds add-option-to-option-group \
  --option-group-name og-in-use-by-your-instance \
  --options OptionName=TIMEZONE_FILE_AUTOUPGRADE \
  --apply-immediately
```

升級您的資料庫引擎版本，而不更新時區檔案

在這個案例中，您的資料庫使用的是 DSTv33，但 DSTv34 可在資料庫執行個體檔案系統上使用。您想要將資料庫引擎升級到版本 19.0.0.0.ru-2022-10.rur-2022-10.r1 (其中包括 DSTv35)，但保留時區檔案 DSTv33。您可能會因下列原因選擇此策略：

- 您的資料不使用 `TIMESTAMP WITH TIME ZONE` 資料類型。
- 您的資料會使用 `TIMESTAMP WITH TIME ZONE` 資料類型，但您的資料不會受到時區變更的影響。
- 您想要延遲更新時區檔案，因為您無法容忍額外的停機時間。

您的策略取決於下列哪個可能性是真的：

- 您的資料庫執行個體並未與包含 `TIMEZONE_FILE_AUTOUPGRADE` 的選項群組相關聯。在您的 `modify-db-instance` 命令中，請勿指定新的選項群組，以便 RDS 不會更新您的時區檔案。
- 您的資料庫執行個體目前與包含的選項群組相關聯 `TIMEZONE_FILE_AUTOUPGRADE`。在單一 `modify-db-instance` 命令中，將您的資料庫執行個體與不包含的選項群組建立關聯，`TIMEZONE_FILE_AUTOUPGRADE` 並將資料庫引擎升級至 19.0.0.0.ru-2022-10.rur-2022-10.r1。

時區檔案更新期間的停機時間

當 RDS 更新您的時區檔案時，使用 `TIMESTAMP WITH TIME ZONE` 的現有資料可能會變更。在此情況下，您的主要考量是停機時間。

Warning

若您新增 `TIMEZONE_FILE_AUTOUPGRADE` 選項，則您的引擎升級可能會延長停機時間。更新大型資料庫的時區資料可能需要花費數小時或數天。

時區檔案更新的長度視下列因素而定：

- 您資料庫中 `TIMESTAMP WITH TIME ZONE` 資料的總量
- 資料庫執行個體組態
- DB instance class (資料庫執行個體類別)
- 儲存組態
- 資料庫組態

- 資料庫參數設定

當您執行下列動作時，可能會出現額外的停機時間：

- 資料庫執行個體使用過期時區檔案時，請將選項新增至選項群組
- 當新引擎版本包含一個新版本的時區檔案時，請升級 Oracle 資料庫引擎

Note

於時區檔案更新期間，RDS for Oracle 會呼叫 PURGE DBA_RECYCLEBIN。

準備更新時區檔案

時區檔案升級有兩個不同的階段：準備和升級。雖然並非必要，但強烈建議您執行準備步驟。於此步驟中，您可以找出將受執行 PL/SQL 程序 DBMS_DST.FIND_AFFECTED_TABLES 影響的資料。如需準備視窗的詳細資訊，請參閱 Oracle 資料庫文件中的[使用時區資料升級時區檔案和時間戳記](#)。

如要準備更新時區檔案

1. 使用 SQL 用戶端連線至您的 Oracle 資料庫。
2. 決定目前使用的時區檔案版本。

```
SELECT * FROM V$TIMEZONE_FILE;
```

3. 決定您資料庫執行個體上可用的最新時區檔案版本。

```
SELECT DBMS_DST.GET_LATEST_TIMEZONE_VERSION FROM DUAL;
```

4. 決定具有類型 TIMESTAMP WITH LOCAL TIME ZONE 或 TIMESTAMP WITH TIME ZONE 資料欄的資料表總大小。

```
SELECT SUM(BYTES)/1024/1024/1024 "Total_size_w_TSTZ_columns_GB"  
FROM   DBA_SEGMENTS  
WHERE  SEGMENT_TYPE LIKE 'TABLE%'  
AND    (OWNER, SEGMENT_NAME) IN  
        (SELECT OWNER, TABLE_NAME  
         FROM   DBA_TAB_COLUMNS
```

```
WHERE DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE');
```

5. 決定具有類型 `TIMESTAMP WITH LOCAL TIME ZONE` 或 `TIMESTAMP WITH TIME ZONE` 資料欄的區段名稱和大小。

```
SELECT OWNER, SEGMENT_NAME, SUM(BYTES)/1024/1024/1024
       "SEGMENT_SIZE_W_TSTZ_COLUMNS_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
         WHERE DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE')
GROUP BY OWNER, SEGMENT_NAME;
```

6. 執行準備步驟。

- 程序 `DBMS_DST.CREATE_AFFECTED_TABLE` 會建立資料表，來儲存任何受影響的資料。您將此資料表的名稱傳遞給 `DBMS_DST.FIND_AFFECTED_TABLES` 程序。如需詳細資訊，請參閱 Oracle 資料庫文件中的 [CREATE_AFFECTED_TABLE 程序](#)。
- 此程序 `CREATE_ERROR_TABLE` 會建立一個資料表來記錄錯誤。如需詳細資訊，請參閱 Oracle 資料庫文件中的 [CREATE_ERROR_TABLE 程序](#)。

下列範例會建立受影響的資料和錯誤資料表，並找出所有受影響的資料表。

```
EXEC DBMS_DST.CREATE_ERROR_TABLE('my_error_table')
EXEC DBMS_DST.CREATE_AFFECTED_TABLE('my_affected_table')

EXEC DBMS_DST.BEGIN_PREPARE(new_version);
EXEC DBMS_DST.FIND_AFFECTED_TABLES('my_affected_table', TRUE, 'my_error_table');
EXEC DBMS_DST.END_PREPARE;

SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

7. 查詢受影響和錯誤資料表。

```
SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

新增時區檔案自動升級選項

當您將選項新增至選項群組時，選項群組會處於下列其中一種狀態：

- 現有的選項群組目前連接至少一個資料庫執行個體。當您新增選項時，使用此選項群組的所有資料庫執行個體都會自動重新啟動。這會導致短暫的中斷。
- 現有的選項群組未連接至任何資料庫執行個體。您計劃新增選項，然後將現有選項群組與現有資料庫執行個體或新的資料庫執行個體建立關聯。
- 您可以建立新的選項群組並新增選項。您計劃將新的選項群組與現有的資料庫執行個體或與新的資料庫執行個體建立關聯。

主控台

如要將時區檔案自動升級新增至資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
 2. 在導覽窗格中，選擇 Option groups (選項群組)。
 3. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇適用於資料庫執行個體的 Oracle Database 版本。
 - b. 針對 Major engine version (主要引擎版本)，請選擇適用於資料庫執行個體版本。
- 如需詳細資訊，請參閱 [建立選項群組](#)。
4. 選擇您要修改的選項群組，然後選擇 Add option (新增選項)。
 5. 在 Add option (新增選項) 視窗中，執行下列作業：
 - a. 選擇 TIMEZONE_FILE_AUTOUPGRADE。
 - b. 若要在所有關聯的資料庫執行個體新增選項時隨即啟用選項，請在 Apply Immediately (立即套用) 選擇 Yes (是)。如果您選擇了 No (不) (預設)，選項將會在下一個維護時段為每個關聯的資料庫執行個體啟用。
 6. 當您滿意設定後，選擇 Add option (新增選項)。

AWS CLI

下列範例會使用新 AWS CLI [增選項至選項群組命令](#)，將選項加入至名為的 `TIMEZONE_FILE_AUTOUPGRADE` 選項群組。myoptiongroup

對於LinuxmacOS、或Unix：

```
aws rds add-option-to-option-group \  
  --option-group-name "myoptiongroup" \  
  --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" \  
  --apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
  --option-group-name "myoptiongroup" ^  
  --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" ^  
  --apply-immediately
```

更新時區檔案後，檢查您的資料

我們建議您於更新時區檔案後檢查您的資料。在準備步驟期間，RDS for Oracle 會自動建立下列表格：

- `rdsadmin.rds_dst_affected_tables` - 列出包含受更新影響之資料的表格
- `rdsadmin.rds_dst_error_table` - 列出更新期間產生的錯誤

這些表格獨立於您在準備視窗中建立的任何表格。如要查看更新的結果，請查詢資料表，如下所示。

```
SELECT * FROM rdsadmin.rds_dst_affected_tables;  
SELECT * FROM rdsadmin.rds_dst_error_table;
```

如需有關受影響資料和錯誤資料表結構描述的詳細資訊，請參閱 Oracle 文件中的 [FIND_AFFECTED_TABLES 程序](#)。

Oracle 透明資料加密

Amazon RDS 支援 Oracle Transparent Data Encryption (TDE)，此為 Oracle Enterprise Edition 中所提供「Oracle 進階安全性」選項的一項功能。此功能會在資料寫入至儲存體之前自動將其加密，並在從儲存體中讀取資料時自動將其解密。此選項僅支援自攜授權 (BYOL) 模型。

TDE 在需要加密敏感資料的情況下非常有用，以防第三方取得資料檔案和備份。當您需要遵守安全相關法規時，TDE 也很有用。

關於 Oracle 資料庫中 TDE 的詳細說明已超出本指南的範圍。如需相關資訊，請參閱下列 Oracle 資料庫資源：

- [使用 Oracle 資料庫文件中的「通透資料加密」來保護儲存資料](#)
- [Oracle 資料庫說明文件中的 Oracle 進階安全性](#)
- [Oracle 進階安全性透明資料加密最佳實務](#)，這是 Oracle 白皮書

如需將 TDE 搭配 Oracle 版 RDS 使用的相關資訊，請參閱下列部落格：

- [Amazon RDS 上的 Oracle 數據庫加密選項](#)
- [遷移具有跨帳戶 TDE 功能的 Amazon RDS for Oracle 資料庫執行個體，以減少停機時間 AWS DMS](#)

TDE 加密模式

「Oracle 透明資料加密」支援兩種加密模式：TDE 資料表空間加密和 TDE 欄加密。TDE 資料表空間加密用於加密全部的應用程式資料表。TDE 欄加密用於加密包含敏感性資料的個別資料元素。您也可以套用混合加密解決方案，即可同時使用 TDE 資料表空間和欄加密。

Note

Amazon RDS 會管理資料庫執行個體的 Oracle 錢包和 TDE 主金鑰。您不需要使用命令 ALTER SYSTEM set encryption key 來設定加密金鑰。

啟用 TDE 此選項之後，您可以使用下列命令來檢查「Oracle 公事包」的狀態：

```
SELECT * FROM v$encryption_wallet;
```

若要建立加密的資料表空間，請使用下列命令：

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

若要指定加密演算法，請使用下列命令：

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

先前用來加密表格空間的陳述式與您在內部部署 Oracle 資料庫中使用的相同。

TDE 選項的限制

TDE 選項是永久且持久的。將資料庫執行個體與已啟用 TDE 選項的選項群組產生關聯後，您無法執行下列動作：

- 停用目前關聯的 TDE 選項群組中的選項。
- 將您的資料庫執行個體與不包含該選項的其他 TDE 選項群組建立關聯。
- 共用使用此 TDE 選項的資料庫快照。如需共用資料庫快照的詳細資訊，請參閱 [共享資料庫快照](#)。

若要取得有關永久和永久選項的更多資訊，請參閱 [持久性與永久性選項](#)。

判斷您的資料庫執行個體是否使用 TDE

您可能想要判斷資料庫執行個體是否與已啟用選項的選 TDE 項群組相關聯。[若要檢視與資料庫執行個體相關聯的選項群組，請使用 RDS 主控台、說明 db 執行個體 AWS CLI 命令或 API 作業 Describe DBInstances。](#)

新增 TDE 選項

若要將 TDE 選項新增至資料庫執行個體，請完成以下步驟：

1. (建議) 建立資料庫執行個體的快照。
2. 執行下列其中一項工作：
 - 從頭開始建立新的選項群組。如需詳細資訊，請參閱 [建立選項群組](#)。
 - 使用 AWS CLI 或 API 複製現有的選項群組。如需詳細資訊，請參閱 [刪除選項群組](#)。
 - 重複使用現有的非預設選項群組。最佳做法是使用目前未與任何資料庫執行個體或快照相關聯的選項群組。
3. 從上一個步驟將新選項新增至選項群組。

4. 如果目前與資料庫執行個體關聯的選項群組已啟用選項，請將這些選項新增至新的選項群組。此策略可防止在啟用新選項時解除安裝現有選項。
5. 將新的選項群組新增至資料庫執行個體。

主控台

將 TDE 選項新增至選項群組，並將其與您的資料庫執行個體建立關聯

1. 在 RDS 主控台中，選擇 [選項群組]。
2. 選擇您要新增選項的選項群組名稱。
3. 選擇 Add option (新增選項)。
4. 在 [選項名稱] 中，選擇 [TDE]，然後設定選項設定。
5. 選擇 Add option (新增選項)。

Important

如果您將 TDE 選項新增至目前連接至一或多個資料庫執行個體的選項群組，則會在自動重新啟動所有資料庫執行個體時發生短暫中斷。

如需新增選項的詳細資訊，請參閱[將選項新增至選項群組](#)。

6. 將選項群組與新的或現有的資料庫執行個體建立關聯：
 - 針對新的資料庫執行個體，在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，可以透過修改執行個體並附加新的選項群組來套用選項群組。當您將新選項新增到現有的資料庫執行個體時，資料庫執行個體自動重新啟動時會發生短暫中斷。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

AWS CLI

在下列範例中，您可以使用新 AWS CLI [增選項至選項群組命令](#)，將選項加入至名為的 TDE 選項群組。myoptiongroup 如需詳細資訊，請參閱[開始使用：Flink 1.13.2](#)。

對於 Linux macOS、或 Unix：

```
aws rds add-option-to-option-group \
```

```
--option-group-name "myoptiongroup" \  
--options "OptionName=TDE" \  
--apply-immediately
```

在 Windows 中：

```
aws rds add-option-to-option-group ^  
--option-group-name "myoptiongroup" ^  
--options "OptionName=TDE" ^  
--apply-immediately
```

將資料複製到不包含 TDE 選項的資料庫執行個體

您無法從資料庫執行個體移除 TDE 選項，也無法將其與不包含 TDE 選項的選項群組產生關聯。若要將資料移轉至不包含 TDE 選項的執行個體，請執行下列動作：

1. 解密資料庫執行個體上的資料。
2. 將資料複製到未與已啟用的選項群組相關聯的新資料庫執行個體。
3. 刪除原始資料庫執行個體。

您可以為新執行個體使用與先前資料庫執行個體相同的名稱。

將 TDE 與 Oracle 資料汲取搭配使用時的考量

您可以使用 Oracle Data Pump 來匯入或匯出加密的傾印檔案。Amazon RDS 支持 Oracle 數據泵的密碼加密模式(ENCRYPTION_MODE=PASSWORD)。Amazon RDS 不支援 Oracle 資料泵(ENCRYPTION_MODE=TRANSPARENT)的透明加密模式。如需更多詳細資訊，請參閱 [使用 Oracle Data Pump 匯入](#)。

Oracle UTL_MAIL

Amazon RDS 透過使用 UTL_MAIL 選項和 SMTP 伺服器支援 Oracle UTL_MAIL。您可以利用 UTL_MAIL 套件，直接從資料庫傳送電子郵件。Amazon RDS 支援下列 Oracle 版本的 UTL_MAIL：

- Oracle Database 21c (21.0.0.0)，所有版本
- Oracle 資料庫 19c (19.0.0.0)，所有版本

下列是使用 UTL_MAIL 時的一些限制：

- UTL_MAIL 不支援 Transport Layer Security (TLS)，因此電子郵件不會加密。

若要建立及上傳自訂 Oracle 錢包，以安全地連線至遠端 SSL/TLS 資源，請遵循 [使用憑證和 Oracle 錢包設定 UTL_HTTP 存取](#) 之中的指示。

您錢包所需的特定憑證視服務而異。對於 AWS 服務，這些通常可以在 [Amazon 信任服務儲存庫](#) 中找到。

- UTL_MAIL 不支援使用 SMTP 伺服器進行身分驗證。
- 您只能在電子郵件中傳送單一附件。
- 您無法傳送大於 32 K 的附件。
- 您只能使用 ASCII 和延伸二進制編碼十進交換碼 (EBCDIC) 字元編碼。
- SMTP 連接埠 (25) 是根據彈性網路界面擁有者的政策進行調節。

當您啟用 UTL_MAIL 時，僅資料庫執行個體的主要使用者會獲授予執行權限。必要時，主要使用者可將執行權限授予其他使用者，以便他們可以使用 UTL_MAIL。

Important

建議您啟用 Oracle 的內建稽核功能來追蹤 UTL_MAIL 程序的使用。

Oracle UTL_MAIL 的先決條件

下列是使用 Oracle UTL_MAIL 的先決條件：

- 一部或多部 SMTP 伺服器，以及對應 IP 位址，或者公有或私有網域名稱伺服器 (DNS) 名稱。如需透過自訂 DNS 伺服器解析之私有 DNS 名稱的詳細資訊，請參閱 [設定自訂 DNS 伺服器](#)。

新增 Oracle UTL_MAIL 選項

將 Oracle UTL_MAIL 選項新增至資料庫執行個體的一般程序如下：

1. 建立新的選項群組，或是複製或修改現有選項群組。
2. 將選項新增至選項群組。
3. 將選項群組與資料庫執行個體建立關聯。

在新增 UTL_MAIL 選項之後，只要選項群組為作用中狀態，UTL_MAIL 就會為作用中狀態。

將 UTL_MAIL 選項新增至資料庫執行個體

1. 判斷要使用的選項群組。您可以建立新的選項群組或使用現有的選項群組。如果您要使用現有的選項群組，請跳到下一個步驟。否則請使用下列設定來建立自訂資料庫選項群組：
 - a. 針對 Engine (引擎)，選擇您想要使用的 Oracle 版本。
 - b. 針對 Major engine version (主要引擎版本)，請選擇您資料庫執行個體的版本。

如需更多詳細資訊，請參閱 [建立選項群組](#)。

2. 將 UTL_MAIL 選項新增至選項群組。如需新增選項的詳細資訊，請參閱 [將選項新增至選項群組](#)。
3. 將選項群組套用至新的或現有的資料庫執行個體：
 - 針對新的資料庫執行個體，您會在啟動執行個體時套用選項群組。如需更多詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。
 - 針對現有的資料庫執行個體，您可以透過修改執行個體並附加新的選項群組來套用選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

使用 Oracle UTL_MAIL

在啟用 UTL_MAIL 選項之後，您必須先設定 SMTP 伺服器，然後才能開始使用該選項。

您可以將 SMTP_OUT_SERVER 參數設定為有效的 IP 位址或公有 DNS 名稱，來設定 SMTP 伺服器。對於 SMTP_OUT_SERVER 參數，您可以指定逗號分隔的清單，列出多個伺服器的位址。如果第一部伺服器無法使用，則 UTL_MAIL 會嘗試下一部伺服器，依此類推。

您可以使用 [資料庫參數群組](#)，來設定資料庫執行個體的預設 SMTP_OUT_SERVER。您可以在資料庫執行個體的資料庫上執行下列程式碼，來設定工作階段的 SMTP_OUT_SERVER 參數。

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

在啟用 UTL_MAIL 選項，且設定 SMTP_OUT_SERVER 之後，您就可以使用 SEND 程序傳送郵件。如需詳細資訊，請參閱 Oracle 文件中的 [UTL_MAIL](#)。

移除 Oracle UTL_MAIL 選項

您可以從資料庫執行個體中移除 Oracle UTL_MAIL。

若要從資料庫執行個體中移除 UTL_MAIL，請執行下列其中一個動作：

- 若要從多個資料庫執行個體中移除 UTL_MAIL，請從其所屬的選項群組中移除 UTL_MAIL 選項。此變更會影響使用選項群組的所有資料庫執行個體。如需更多詳細資訊，請參閱 [從選項群組移除選項](#)。
- 若要從單一資料庫執行個體中移除 UTL_MAIL，請修改資料庫執行個體，並指定不包括 UTL_MAIL 選項的不同選項群組。您可以指定預設的 (空白) 選項群組或不同的自訂選項群組。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

疑難排解

下列是您使用 UTL_MAIL 與 Amazon RDS 搭配時可能遇到的問題。

- 調節。SMTP 連接埠 (25) 是根據彈性網路界面擁有者的政策進行調節。如果您可以使用 UTL_MAIL 成功傳送電子郵件，而且看到錯誤 ORA-29278: SMTP transient error: 421 Service not available，可能是因為正在進行調節。如果您在傳輸電子郵件時經歷調節，則建議您實作退避演算法。如需輪詢演算法的詳細資訊，請參閱 [AWS 中的錯誤重試與指數退避](#) 和 [如何處理「調節 – 超過最大傳送速率」錯誤](#)。

您可以要求移除此調節。如需詳細資訊，請參閱 [如何從我的 EC2 執行個體中移除連接埠 25 上的調節？](#)。

Oracle XML 資料庫

Oracle XML 資料庫將原生 XML 支援新增至您的資料庫執行個體。使用 XML DB，您可以存儲和檢索結構化或非結構化 XML 和關係數據。該 XML 數據庫協議服務器不支持的 RDS 為甲骨文。

XML 數據庫已預先安裝在甲骨文數據庫 12c 及更高版本上。因此，您不需要使用選項群組來明確安裝 XML DB 做為附加功能。

若要瞭解如何設定和使用 XML 資料庫，請參閱 [Oracle 資料庫說明文件中的 Oracle XML 資料庫開發人員指南](#)。

升級 RDS for Oracle 資料庫引擎

當 Amazon RDS 支援新版本的 Oracle 資料庫時，您可以將資料庫執行個體升級為新版本。如需 Amazon RDS 上提供的 Oracle 版本相關資訊，請參閱 [Amazon RDS for Oracle 版本備註](#)。

Important

不再支援 RDS for Oracle 資料庫 11g、12c 和 18c。如果您維護 Oracle 資料庫 11g、12c 或 18c 快照，可以將其升級至較新的版本。如需更多詳細資訊，請參閱 [升級 Oracle 資料庫快照](#)。

主題

- [RDS for Oracle 引擎升級概觀](#)
- [Oracle 主要版本升級](#)
- [Oracle 次要版本升級](#)
- [Oracle 資料庫升級的考量](#)
- [測試 Oracle 資料庫升級](#)
- [升級 Oracle 資料庫執行個體的 RDS 版本](#)
- [升級 Oracle 資料庫快照](#)

RDS for Oracle 引擎升級概觀

升級 RDS for Oracle 資料庫執行個體之前，請先熟悉下列概念。

主題

- [主要和次要版本升級](#)
- [RDS for Oracle 主要版本的預期支援日期](#)
- [Oracle 引擎版本管理](#)
- [引擎升級期間的自動快照](#)
- [異地同步備份部署中的 Oracle 升級](#)
- [僅供讀取複本的 Oracle 升級](#)

主要和次要版本升級

主要版本是 Oracle 資料庫的主要版本，每 1 至 2 年推出一次。主要版本的範例為 Oracle 資料庫 19c 和 Oracle 資料庫 21c。

次要版本也稱為版本更新 (RU)，通常由 Oracle 每季推出一次。次要版本包含小功能增強和錯誤修正。次要版本的例子是 21.0.0.0.Ru-2023-10-羅爾 -2023-10.r1 和 19.0.0.0.0.Ru-2023-10.r。如需更多詳細資訊，請參閱 [適用於 Oracle 的 Amazon Relational Database Service \(Amazon RDS\) 版本備註](#)。

RDS for Oracle 支援下列資料庫執行個體的升級：

升級類型	應用程式相容性	升級方式	升級路徑範例
主要版本	主要版本升級可能引進與現有應用程式不相容的變更。	僅限手動	從 Oracle 資料庫 19c 到 Oracle 資料庫 21c
次要版本	次要版本升級只包含與現有應用程式回溯相容的變更。	自動或手動	從二〇〇〇〇三年七月二十一日至二十一點零零零

Important

當您升級資料庫引擎時，將會發生中斷。中斷的持續時間取決於您的引擎版本和資料庫執行個體大小。

所有升級都務必要經過完整測試，以確認您的應用程式正常運作，再將升級套用至您的生產資料庫。如需詳細資訊，請參閱 [測試 Oracle 資料庫升級](#)。

RDS for Oracle 主要版本的預期支援日期

RDS for Oracle 主要版本至少在對應 Oracle 資料庫發行版本的支援日期結束前都會保持可用。您可以使用下列日期來規劃測試和升級週期。這些日期表示可能需要升級到較新版本的最早日期。如果 Amazon 延長對 RDS for Oracle 版本的支援時間超過原先聲明的時間，我們計劃更新此資料表以反映後來的日期。

Oracle 資料庫主要版本	升級至較新版本的預計日期
Oracle 資料庫 19c	2026 年 4 月 30 日提供 BYOL 頂級 Support 服務 (延伸 Support 豁免費用) 2027 年 4 月 30 日含 BYOL 延伸 Support (額外費用) 或無限制授權合約 2027 年 4 月 30 日, 含授權已包含 (LI)
Oracle Database 21c	2025 年 4 月 30 日 (不適用於延伸 Support)

在我們要求您升級至更新的主要版本之前，我們會至少提前 12 個月前就提醒您。我們會詳細說明升級程序，包括重要里程碑的時間、對資料庫叢集的影響，以及建議動作。您應在升級至主要版本之前，先徹底對應用程式進行新版 RDS for Oracle 的測試。

在這事先通知期間之後，可能會將後續主要版本的自動升級套用至任何仍在執行舊版本的 RDS for Oracle 資料庫執行個體。如果是這樣，則會在排定的維護時段期間啟動升級。

如需詳細資訊，請參閱我的 Oracle 客戶服務部 [中目前資料庫版本的發行排 Support](#)。

Oracle 引擎版本管理

透過資料庫引擎版本管理，您可控制何時以及如何對資料庫引擎軟體進行修補和升級。您可以靈活地維持與資料庫引擎修補程式版本相容的狀態。您也可以測試新的 RDS for Oracle 修補程式版本，以確保它們在生產環境中部署之前，能與您的應用程式搭配使用。此外，您可以根據自己的條件和時間表升級版本。

Note

Amazon RDS 會利用 Amazon RDS 特定的資料庫引擎版本，定期彙總官方 Oracle 資料庫修補程式。如需查看 Amazon RDS Oracle 特定引擎版本中所含的 Oracle 修補程式清單，請前往 [Amazon RDS for Oracle 版本備註](#)。

引擎升級期間的自動快照

在您升級 Oracle 資料庫執行個體期間，快照可提供防範升級問題的保護。如果資料庫執行個體的備份保留期間大於 0，則 Amazon RDS 會在升級期間拍攝下列資料庫快照：

1. 完成任何升級變更之前的資料庫執行個體快照。如果升級失敗，您可以還原此快照，以建立執行舊版本的資料庫執行個體。
2. 升級完成後的資料庫執行個體快照。

Note

若要變更備份保留期，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

升級後，您無法還原至先前的引擎版本。不過，您可以透過還原升級前的快照，來建立新的 Oracle 資料庫執行個體。

異地同步備份部署中的 Oracle 升級

如果資料庫執行個體位於異地同步備份部署中，Amazon RDS 會升級主要和待命複本。如果不需要任何作業系統更新，主要和待命升級會同時進行。在升級完成之後，執行個體才能使用。

如果異地同步備份部署需要作業系統更新，Amazon RDS 會在您請求資料庫升級時套用更新。Amazon RDS 執行以下步驟：

1. 更新目前待命資料庫執行個體上的作業系統。
2. 將主要資料庫執行個體容錯移轉至待命資料庫執行個體。
3. 升級新主要資料庫執行個體 (先前為待命執行個體) 上的資料庫版本。主要資料庫在升級期間無法使用。
4. 更新新待命資料庫執行個體 (以前是主要資料庫執行個體) 上的作業系統。
5. 升級新待命資料庫執行個體上的資料庫版本。
6. 將新的主要資料庫執行個體容錯移轉回原始主要資料庫執行個體，並將新的待命資料庫執行個體容錯移轉回原始備用資料庫執行個體。因此，Amazon RDS 會將複寫組態恢復到其原始狀態。

僅供讀取複本的 Oracle 升級

來源資料庫執行個體及其所有僅供讀取複本的 Oracle 資料庫引擎版本必須相同。Amazon RDS 在下列階段執行升級：

1. 升級來源資料庫執行個體。僅供讀取複本可在此階段使用。
2. 平行升級僅供讀取複本，無論複本維護時段為何。來源資料庫可在此階段使用。

對於跨區域僅供讀取複本的主要版本升級，Amazon RDS 會執行其他動作：

- 自動產生目標版本的選項群組
- 將原始選項群組中的所有選項和選項設定複製到新選項群組
- 將升級的跨區域僅供讀取複本與新選項群組建立關聯

Oracle 主要版本升級

若要執行主要版本升級，請手動修改資料庫執行個體。主要版本升級不會自動進行。

Important

所有升級都務必要經過完整測試，以確認您的應用程式正常運作，再將升級套用至您的生產資料庫。如需詳細資訊，請參閱 [測試 Oracle 資料庫升級](#)。

主題

- [主要升級的支援版本](#)
- [主要升級的支援執行個體類別](#)
- [在主要升級前收集統計資料](#)
- [允許主要升級](#)

主要升級的支援版本

Amazon RDS 支援下列主要版本升級。

目前版本	支援的升級
19.0.0.0 使用 CDB 架構	21.0.0.0

Oracle Database 的主要版本升級必須升級至同月或稍後發行的版本更新 (RU)。不支援任何 Oracle Database 版本的主要版本降級。

主要升級的支援執行個體類別

您目前的 Oracle 資料庫執行個體可能在您要升級到的版本所不支援的資料庫執行個體類別上執行。在此情況下，在升級之前，請將資料庫執行個體移轉至支援的資料庫執行個體類別。如需 Amazon RDS for Oracle 各版本和版次所支援的資料庫執行個體類別詳細資訊，請參閱 [資料庫執行個體類別](#)。

在主要升級前收集統計資料

在執行主要版本升級之前，Oracle 建議您先在您要升級的資料庫執行個體上收集最佳化工具統計資料。此動作可減少在升級期間資料庫執行個體的停機時間。

若要收集最佳化工具統計資料，請以主要使用者身分連接至資料庫執行個體，然後執行 DBMS_STATS.GATHER_DICTIONARY_STATS 程序，如下列範例所示。

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

如需詳細資訊，請參閱 [Oracle 文件中的程序](#)。

允許主要升級

主要引擎版本升級可能與您的應用程式不相容。升級無法復原。如果您為 EngineVersion 參數指定與目前主要版本不同的主要版本，則必須允許主要版本升級。

如果您使用 CLI 命令 [modify-db-instance](#) 升級主要版本，請指定 `--allow-major-version-upgrade`。此設定並非持久性設定，因此您必須在每次執行主要升級時均指定 `--allow-major-version-upgrade`。此參數不會影響次要引擎版本的升級。如需更多詳細資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

如果您使用主控台升級主要版本，您就不需要選擇允許升級的選項。相反地，主控台會顯示主要升級無法復原的警告。

Oracle 次要版本升級

次要版本升級適用於主要引擎版本中 Oracle 資料庫修補程式集更新 (PSU) 或版本更新 (RU)。例如，若您的資料庫執行個體執行了 Oracle Database 21c 主要版本和 21.0.0.0.ru-2022-07.rur-2022-07.r1 次要版本，您可以升級至 21.0.0.0.ru-2022-10.rur-2022-10.r1 次要版本。通常每季都會推出新的次要版本。

Note

RDS for Oracle 不支援次要版本降級。

您可以手動或自動將資料庫引擎升級為次要版本。如需了解如何手動升級，請參閱 [手動升級引擎版本](#)。如需了解如何設定自動升級，請參閱 [自動升級次要引擎版本](#)。無論手動或自動升級，次要版本升級皆需停機時間。請在規劃升級時謹記這點。

Important

所有升級都務必要經過完整測試，以確認您的應用程式正常運作，再將升級套用至您的生產資料庫。如需更多詳細資訊，請參閱 [測試 Oracle 資料庫升級](#)。

主題

- [開啟 Oracle 的自動次要版本升級](#)
- [排程 Oracle 的自動次要版本升級之前](#)
- [RDS 何時為 Oracle 排程自動次要版本升級](#)
- [管理 Oracle 的自動次要版本升級](#)

開啟 Oracle 的自動次要版本升級

進行自動次要版本升級時，RDS 會將最新可用的次要版本套用至您的 Oracle 資料庫，無需您手動介入。在下列情況下，Amazon RDS for Oracle 資料庫執行個體會在下一次維護時段排程升級：

- 您的資料庫執行個體已開啟自動次要版本升級選項。
- 您的資料庫執行個體未執行最新次要資料庫引擎版本。
- 您的資料庫執行個體尚未有排定的擱置升級。

如需了解如何開啟自動升級，請參閱 [自動升級次要引擎版本](#)。

排程 Oracle 的自動次要版本升級之前

RDS 會在排程自動升級之前發佈提前通知。您可以在資料庫詳細資訊頁面的維護與備份標籤中看到通知。訊息採用下列格式：

An automatic minor version upgrade to *engine version* will become available on *availability-date* and will be applied during a subsequent maintenance window.

上一則訊息的 *availability-date* 是 RDS 開始排程 AWS 區域中資料庫執行個體升級的日期。而非資料庫執行個體的升級日期。

您也可以使用 AWS CLI 中的 `describe-pending-maintenance-actions` 命令，取得升級可用日期，如下範例所示：

```
aws rds describe-pending-maintenance-actions

{
  "PendingMaintenanceActions": [
    {
      "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:orclinst1",
      "PendingMaintenanceActionDetails": [
        {
          "Action": "db-upgrade",
          "Description": "Automatic minor version upgrade to
21.0.0.0.ru-2022-10.rur-2022-10.r1",
          "CurrentApplyDate": "2022-12-02T08:10:00Z",
          "OptInStatus": "next-maintenance"
        }
      ]
    }
  ], ...
}
```

下表說明每一種類型的待定維護動作訊息的選項。

待定維護動作訊息	訊息出現時	是否有資格在 下一次維修時 段套用？	是否有資格 立即套用？	是否有資格 復原選擇加 入？
<i>engine-version</i> 的 自動次要版本升級將於 <i>availability-date</i> 推出，並且應在接下來的 維護時段套用。	在排程的自動升級 前 4-6 週。	是	是	是
<i>engine-version</i> 的自 動次要版本升級	在 <i>availabil</i> <i>ity-date</i> 當天或	是	是	否

待定維護動作訊息	訊息出現時	是否有資格在 下一次維修時 段套用？	是否有資格 立即套用？	是否有資格 復原選擇加 入？
	之後。RDS 會在資料庫執行個體的 下一次維護時段中自 動套用此升級。			

如需了解 [describe-pending-maintenance-actions](#)，請參閱《AWS CLI 命令參考》。

RDS 何時為 Oracle 排程自動次要版本升級

到自動升級的可用日期時，RDS 會開始排程升級。對於大部分的 AWS 區域，RDS 會在可用日期約四至六週後，為您排程升級至每季最新 RU。排程日期會有所不同，取決於 AWS 區域 和其他因素。如需有關 RU 和 RUR 詳細資訊，請參閱 [Amazon RDS for Oracle 版本備註](#)。

RDS 排程升級時，下列通知會顯示在資料庫詳細資訊頁面的維護與備份標籤：

```
Automatic minor version upgrade to engine-version
```

上述訊息指出 RDS 已排程您的資料庫引擎在下一個維護時段進行升級。

管理 Oracle 的自動次要版本升級

新的次要版本可用時，您可以手動將資料庫執行個體升級為該版本。以下範例會立即升級名為 `orclinst1` 的資料庫執行個體：

```
aws rds apply-pending-maintenance-action \
  --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
  --apply-action db-upgrade \
  --opt-in-type immediate
```

若要選擇退出尚未排程的次要版本自動升級，請將 `opt-in-type` 設為 `undo-opt-in`，如下列範例所示：

```
aws rds apply-pending-maintenance-action \
  --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
  --apply-action db-upgrade \
  --opt-in-type undo-opt-in
```

```
--opt-in-type undo-opt-in
```

若 RDS 已排定資料庫執行個體升級，您就無法以 `apply-pending-maintenance-action` 取消該升級。但您可以修改資料庫執行個體，並關閉自動次要升級功能，進而取消升級排程。

如需了解如何關閉自動次要版本升級，請參閱 [自動升級次要引擎版本](#)。如需了解 [apply-pending-maintenance-action](#)，請參閱《AWS CLI 命令參考》。

Oracle 資料庫升級的考量

升級 Oracle 執行個體前，請先檢閱下列資訊。

主題

- [Oracle 多重租用考量](#)
- [選項群組考量](#)
- [參數群組考量](#)
- [時區考量](#)

Oracle 多重租用考量

下表說明不同版本支援的 Oracle 資料庫架構。

Oracle Database 版本	RDS 支援狀態	架構
Oracle Database 21c	支援	僅 CDB
Oracle 資料庫 19c	支援	CDB 或非 CDB

下表描述支援和不支援的升級途徑。

升級途徑	支援？
CDB 到 CDB	是
非 CDB 到 CDB	不，但您可以將非 CDB 轉換為 CDB，然後對其進行升級

升級途徑	支援？
CDB 到非 CDB	否

如需 RDS for Oracle 中 Oracle 多租用戶的詳細資訊，請參閱 [CDB 架構的單一租戶組態](#)。

選項群組考量

如果您的資料庫執行個體使用自訂選項群組，有時 Amazon RDS 無法自動指派新的選項群組。例如，在升級至新的主要版本時發生此情況。在此類情況下，請在升級時指定新的選項群組。我們建議您建立新的選項群組，並如同在現有自訂選項群組中般，將相同的選項新增到其中。

如需詳細資訊，請參閱「[建立選項群組](#)」或「[刪除選項群組](#)」。

如果您的資料庫執行個體使用包含 APEX 選項的自訂選項群組，有時您可以縮短升級所需的時間。若要這麼做，請與資料庫執行個體同時升級 APEX 版本。如需更多詳細資訊，請參閱 [升級 APEX 版本](#)。

參數群組考量

如果您的資料庫執行個體使用自訂參數群組，Amazon RDS 有時無法自動為資料庫執行個體指派新的參數群組。例如，在升級至新的主要版本時發生此情況。在此類情況下，請務必在升級時指定新的參數群組。建議您建立新的參數群組，並在其中設定和現有自訂參數群組中相同的參數。

如需詳細資訊，請參閱「[建立資料庫參數群組](#)」或「[複製資料庫參數群組](#)」。

時區考量

您可以使用時區選項，來變更 Oracle 資料庫執行個體所使用的系統時區。例如，您可以變更資料庫執行個體的時區，以與現場部署環境或舊型應用程式相容。時區選項會變更主機層級的時區。Amazon RDS for Oracle 全年自動更新系統時區。如需系統時區的詳細資訊，請參閱 [Oracle 時區](#)。

當您建立 Oracle 資料庫執行個體時，資料庫會自動設定資料庫時區。資料庫時區也稱為「日光節約時間 (DST)」時區。資料庫時區與系統時區不同。

在 Oracle 資料庫版本之間，修補程式集或個別修補程式可能包含新的 DST 版本。這些修補程式反映不同時區區域的轉換規則變更。例如，當 DST 生效時，政府可能會變更。DST 規則的變更可能會影響 `TIMESTAMP WITH TIME ZONE` 資料類型的現有資料。

若您升級 RDS for Oracle 資料庫執行個體，Amazon RDS 不會自動升級資料庫時區檔案。如要自動升級時區檔案，您可在引擎版本升級期間或之後在與資料庫執行個體相關聯的選項組中包含 `TIMEZONE_FILE_AUTOUPGRADE` 選項。如需詳細資訊，請參閱 [Oracle 時區檔案自動升級](#)。

或者，如要手動升級資料庫時區檔案，請建立具有所需 DST 修補程式的新 Oracle 資料庫執行個體。不過，我們建議您使用 `TIMEZONE_FILE_AUTOUPGRADE` 選項升級資料庫時區檔案。

升級時區檔案後，將資料從目前執行個體遷移至新執行個體。您可以使用幾種技術來遷移資料，包括以下各項：

- AWS Database Migration Service
- 甲骨文 GoldenGate
- Oracle Data Pump
- 原始匯出/匯入 (不再支援一般用途)

Note

使用 Oracle Data Pump 遷移資料時，當目標時區版本低於來源時區版本時，公用程式會引發錯誤 ORA-39405。

如需詳細資訊，請參閱 Oracle 文件中的 [TIMESTAMP WITH TIMEZONE 限制](#)。

測試 Oracle 資料庫升級

在將資料庫執行個體升級至主要版本之前，請完整測試資料庫，以及會存取該資料庫的所有應用程式，了解是否與新版本相容。建議您採用下列程序。

測試主要版本升級

1. 檢閱新版本資料庫引擎的 Oracle 升級文件，了解是否有可能影響資料庫或應用程式的相容性問題。如需詳細資訊，請參閱 Oracle 文件中的 [資料庫升級指南](#)。
2. 如果您的資料庫執行個體使用自訂選項群組，請建立與升級後新版本相容的新選項群組。如需更多詳細資訊，請參閱 [選項群組考量](#)。
3. 如果您的資料庫執行個體使用自訂參數群組，請建立與升級後新版本相容的新參數群組。如需更多詳細資訊，請參閱 [參數群組考量](#)。
4. 針對要升級的資料庫執行個體建立資料庫快照。如需詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。
5. 還原資料庫快照，以建立新的測試資料庫執行個體。如需更多詳細資訊，請參閱 [從資料庫快照還原](#)。

6. 使用下列其中一種方法，修改這個新的測試資料庫執行個體，以升級到新版本：
 - [主控台](#)
 - [AWS CLI](#)
 - [RDS API](#)
7. 執行測試：
 - 對升級的資料庫執行個體儘可能進行越多次品質保證測試，以確保資料庫和應用程式搭配新版本可以正常運作。
 - 實作任何必要的新測試，以評估您在步驟 1 發現的任何相容性問題所帶來的影響。
 - 測試所有預存程序、函數和觸發程序。
 - 將應用程式的測試版本指向升級的資料庫執行個體。驗證應用程式可對新版本正確運作。
 - 評估升級的執行個體所使用的儲存體，以決定升級是否需要更多儲存體。您可能需要選擇較大的執行個體類別，才能支援生產環境中的新版本。如需更多詳細資訊，請參閱 [資料庫執行個體類別](#)。
8. 如果通過所有測試，則請升級生產資料庫執行個體。允許對資料庫執行個體進行寫入作業之前，建議您先確認資料庫執行個體運作正常。

升級 Oracle 資料庫執行個體的 RDS 版本

若要手動升級適用於 Oracle 資料庫執行個體之 RDS 的資料庫引擎版本 AWS CLI，請使用、或 RDS API。AWS Management Console 如需 RDS 中資料庫升級的一般資訊，請參閱 [升級 Oracle 資料庫執行個體的 RDS 版本](#)。若要取得有效的升級目標，請使用 AWS CLI [describe-db-engine-versions](#) 指令。

主控台

使用主控台升級 RDS 適用於 Oracle 資料庫執行個體的引擎版本

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中選擇 Databases (資料庫)，然後選擇您要升級的資料庫執行個體。
3. 選擇 Modify (修改)。
4. 對於資料庫引擎版本，請選擇較高的資料庫版本。

5. 選擇 Continue (繼續)，並檢查修改的摘要。請確定您瞭解資料庫版本升級的含意。您無法將升級的資料庫執行個體轉換回先前的版本。在繼續之前，請確保您已經使用新版本測試了數據庫和應用程序。
6. 決定何時排定資料庫執行個體升級。若要立即套用變更，請選擇 Apply immediately (立即套用)。在某些情況下，選擇此選項會導致停機。如需更多詳細資訊，請參閱 [排程修改設定](#)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB instance (修改資料庫執行個體) 以儲存您的變更。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要升級適用於 Oracle 資料庫執行個體的 RDS 引擎版本，您可以使用 CLI [modify-db-instance](#) 命令。指定下列參數：

- `--db-instance-identifier`— Oracle 資料庫執行個體的 RDS 名稱。
- `--engine-version` – 會以此資料庫引擎版本編號為目標進行升級。

如需有效引擎版本的相關資訊，請使用 AWS CLI [describe-db-engine-versions](#) 指令。

- `--allow-major-version-upgrade`-升級數據庫引擎版本。
- `--no-apply-immediately` – 在下次維護時段套用變更。若要立即套用變更，請使用 `--apply-immediately`。

Example

下列範例會將名為的 CDB 執行個體 `myorainst` 從目前版本的升級 `19.0.0.0.ru-2024-01.rur-2024-01.r1` 為版本 `21.0.0.0.ru-2024-04.rur-2024-04.r1`。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier myorainst \  
  --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 \  
  --allow-major-version-upgrade \  
  --no-apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --db-instance-identifier myorainst ^
  --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 ^
  --allow-major-version-upgrade ^
  --no-apply-immediately
```

RDS API

若要升級適用於 Oracle 資料庫執行個體的 RDS，請使用[修改資料庫執行個體動作](#)。指定下列參數：

- `DBInstanceIdentifier` – 資料庫執行個體的名稱，例如 *myorainst*。
- `EngineVersion` – 會以此資料庫引擎版本編號為目標進行升級。如需有效引擎版本的相關資訊，請使用[描述 B 作EngineVersions](#)業。
- `AllowMajorVersionUpgrade` – 是否允許主要版本升級。若要這麼做，請設為 `true` 值。
- `ApplyImmediately` – 指出是否要立即套用變更，或等到下個維護時段再套用。若要立即套用變更，請將值設為 `true`。若要在下一次維護時段套用變更，請將值設為 `false`。

升級 Oracle 資料庫快照

如果您有現有的手動資料庫快照，您可以將它們升級至 Oracle 資料庫引擎的更新版本。

當 Oracle 停止為某個版本提供修補程式，且 Amazon RDS 取代該版本時，您可以升級與已取代版本對應的快照。如需詳細資訊，請參閱[Oracle 引擎版本管理](#)。

Amazon RDS 在所有 AWS 區域中都支援升級快照。

主控台

升級 Oracle 資料庫快照

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)，然後選取您要升級的資料庫快照。
3. 針對 Actions (動作)，選擇 Upgrade snapshot (升級快照)。Upgrade snapshot (升級快照) 頁面隨即出現。
4. 選擇要升級快照的新引擎版本。
5. (選用) 在 Option group (選項群組) 中，選擇升級後資料庫快照的選項群組。升級資料庫快照時的選項群組注意事項，與升級資料庫執行個體時相同。如需更多詳細資訊，請參閱[選項群組考量](#)。

6. 請選擇 Save changes (儲存變更) 儲存您所做的變更。

升級程序期間，此資料庫快照的所有快照動作會停用。同時，資料庫快照狀態會從 available (可用) 變更為 upgrading (升級中)，然後在完成時變更為 active (作用中)。如果資料庫快照因為快照損毀問題而無法升級，狀態會變更為 unavailable (無法使用)。您無法從此狀態復原快照。

Note

如果資料庫升級失敗，快照會轉返至原始版本的原始狀態。

AWS CLI

若要使用升級 Oracle 資料庫快照集AWS CLI，請使用下列參數呼叫[modify-db-snapshot](#)命令：

- `--db-snapshot-identifier` – 資料庫快照的名稱。
- `--engine-version` – 快照會以此版本為目標進行升級。

您可能也需要提供下列參數。升級資料庫快照時的選項群組注意事項，與升級資料庫執行個體時相同。如需更多詳細資訊，請參閱 [選項群組考量](#)。

- `--option-group-name` – 升級後資料庫快照的選項群組。

Example

下列範例會升級資料庫快照。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier mydbsnapshot \  
  --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \  
  --option-group-name default:oracle-se2-19
```

在Windows中：

```
aws rds modify-db-snapshot ^  
  --db-snapshot-identifier mydbsnapshot ^  
  --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 ^
```

```
--option-group-name default:oracle-se2-19
```

RDS API

若要使用 Amazon RDS API 來升級 Oracle 資料庫快照，請搭配下列參數呼叫 [ModifyDBSnapshot](#) 操作：

- `DBSnapshotIdentifier` – 資料庫快照的名稱。
- `EngineVersion` – 快照會以此版本為目標進行升級。

您可能還需要包含 `OptionGroupName` 參數。升級資料庫快照時的選項群組注意事項，與升級資料庫執行個體時相同。如需更多詳細資訊，請參閱 [選項群組考量](#)。

將 RDS for Oracle 資料庫執行個體與第三方軟體搭配使用

您可以託管支援工具和協力廠商軟體的 RDS for Oracle 資料庫執行個體。

主題

- [使用甲骨文 GoldenGate 與 Amazon RDS for Oracle](#)
- [在 RDS for Oracle 上使用 Oracle Repository Creation Utility](#)
- [在 Amazon EC2 執行個體上設定 Oracle Connection Manager](#)
- [在 Amazon RDS 上的 Oracle 上安裝 Siebel 資料庫](#)

使用甲骨文 GoldenGate 與 Amazon RDS for Oracle

Oracle 會在資料庫之間 GoldenGate 收集、複寫和管理交易資料。這是日誌式變更資料擷取 (CDC) 和複寫軟體套件，與線上交易處理 (OLTP) 系統的資料庫搭配使用。Oracle GoldenGate 會建立軌跡檔案，其中包含來源資料庫中最近變更的資料。然後它會將這些檔案推送至伺服器，其中有一個程序會將追蹤檔案轉換成標準 SQL，以套用至目標資料庫。

甲骨文 GoldenGate 與 RDS 為甲骨文支持以下功能：

- 主動-主動式資料庫複寫
- 災難復原
- 資料保護
- 區域內和跨區域複寫
- 零停機時間遷移與升級
- 在 RDS for Oracle 資料庫執行個體與非 Oracle 資料庫之間進行資料複寫

Note

如需支援的資料庫清單，請參閱 Oracle 文件中的 [Oracle Fusion Middleware 支援的系統組態](#)。

您可以使用 Oracle GoldenGate 與 RDS 為甲骨文升級到主要版本的 Oracle 數據庫。例如，您可以使用 Oracle GoldenGate 在 Amazon RDS 資料庫執行個體上從甲骨文資料庫 11g 現場部署資料庫升級到 Oracle 資料庫 19c。

主題

- [Oracle 支援的版本和授權選項 GoldenGate](#)
- [甲骨文的要求和限制 GoldenGate](#)
- [甲骨文 GoldenGate 架構](#)
- [設定甲骨文 GoldenGate](#)
- [使用甲骨文的提取和副本實用程序 GoldenGate](#)
- [監督甲骨文 GoldenGate](#)
- [Oracle 疑難排解 GoldenGate](#)

Oracle 支援的版本和授權選項 GoldenGate

您可以在甲骨文 GoldenGate 文 12c 及更高版本中使用 RDS 的標準版 2 (SE2) 或企業版 (EE)。您可以使用下列 Oracle GoldenGate 功能：

- 支援 Oracle GoldenGate 遠端擷取 (擷取)。
- 使用傳統非 CDB 資料庫架構的 RDS for Oracle 資料庫執行個體支援擷取。Oracle 資料庫 21c 容器資料庫 (CDB) 支援 Oracle GoldenGate 遠端 PDB 擷取。
- 針對使用非 CDB 或 CDB 架構的 Oracle 資料庫執行個體，RDS 支援 Oracle GoldenGate 遠端傳遞 (複本)。遠端交付支援整合式複寫、平行複寫、協調式複寫和傳統複寫。
- 適用於甲骨文的 RDS 支持甲骨文的典型和微服務架構。GoldenGate
- 使用整合式擷取模式時，支援 Oracle GoldenGate DDL 和序列值複寫。

您必須負責管理甲骨文 GoldenGate 授權 (BYOL)，以便與 Amazon RDS 一起使用。AWS 區域如需詳細資訊，請參閱 [RDS for Oracle 授權選項](#)。

甲骨文的要求和限制 GoldenGate

當您使用甲骨文 GoldenGate 和 RDS 適用於甲骨文時，請考慮以下要求和限制：

- 你負責設置和管理甲骨文與 RDS GoldenGate 為甲骨文使用。
- 您必須負責設定已通過來源和目標資料庫認證的 Oracle GoldenGate 版本。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Fusion Middleware 支援的系統組態](#)。
- 您可以針對許多不同的使用案例 GoldenGate 在許多不同的 AWS 環境中使用 Oracle。如果您有與 Oracle 相關的支援相關問題 GoldenGate，請聯絡「Oracle 客戶 Support 務部」。
- 您可以針對使用 Oracle 通透資料加密 (TDE) 的 Oracle 資料庫執行個體使用 Oracle GoldenGate on RDS。若要維護複寫資料的完整性，請使用 Amazon EBS 加密磁碟區或追蹤檔案加密在 Oracle GoldenGate 中樞上設定加密。同時為 Oracle GoldenGate Hub 與來源和目標資料庫執行處理之間傳送的資料設定加密。RDS for Oracle 資料庫執行個體支援使用 [Oracle Secure Sockets Layer](#) 或 [Oracle 原生網路加密](#) 加密。

甲骨文 GoldenGate 架構

與 Amazon RDS 搭配使用的甲骨文 GoldenGate 架構包含以下分離模組：

來源資料庫

來源資料庫可以是內部部署 Oracle 資料庫、Amazon EC2 執行個體上的 Oracle 資料庫，或 Amazon RDS 資料庫執行個體上的 Oracle 資料庫。

甲骨文 GoldenGate 樞紐

Oracle 中 GoldenGate 樞會將交易資訊從來源資料庫移至目標資料庫。中樞可以是以下其中一種：

- GoldenGate 安裝甲骨文數據庫和甲骨文的 Amazon EC2 實例
- 內部部署 Oracle 安裝

您可以有多個 Amazon EC2 中樞。如果您使用 Oracle GoldenGate 進行跨區域複寫，建議您使用兩個中樞。

目標資料庫

您的目標資料庫可以位於 Amazon RDS 資料庫執行個體、Amazon EC2 執行個體或內部部署位置。

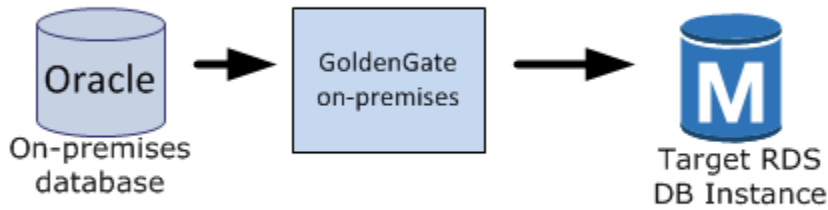
以下各節說明 Amazon RDS GoldenGate 上甲骨文的常見案例。

主題

- [內部部署來源資料庫和 Oracle GoldenGate 中樞](#)
- [內部部署來源資料庫和 Amazon EC2 中樞](#)
- [Amazon RDS 來源資料庫和 Amazon EC2 中樞](#)
- [Amazon EC2 來源資料庫和 Amazon EC2 中樞](#)
- [位於不同 AWS 區域的 Amazon EC2 中樞](#)

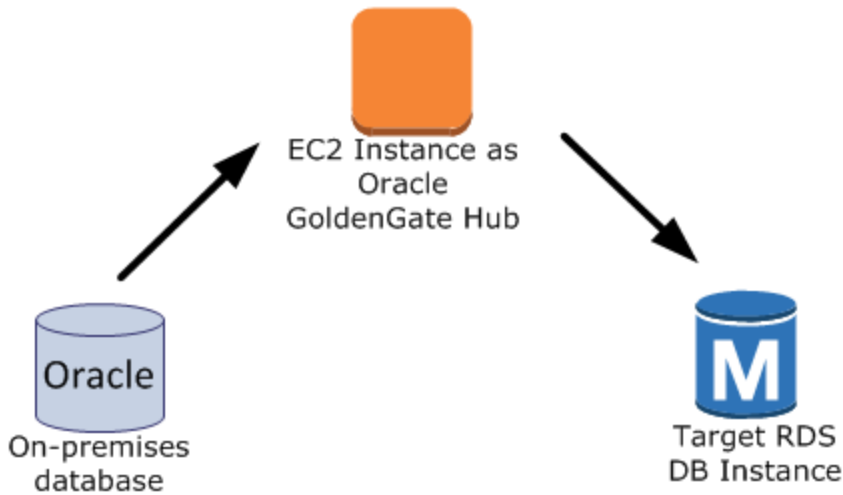
內部部署來源資料庫和 Oracle GoldenGate 中樞

在此案例中，現場部署 Oracle 來源資料庫和現場部署 Oracle 中 GoldenGate 樞會將資料提供給目標 Amazon RDS 資料庫執行個體。



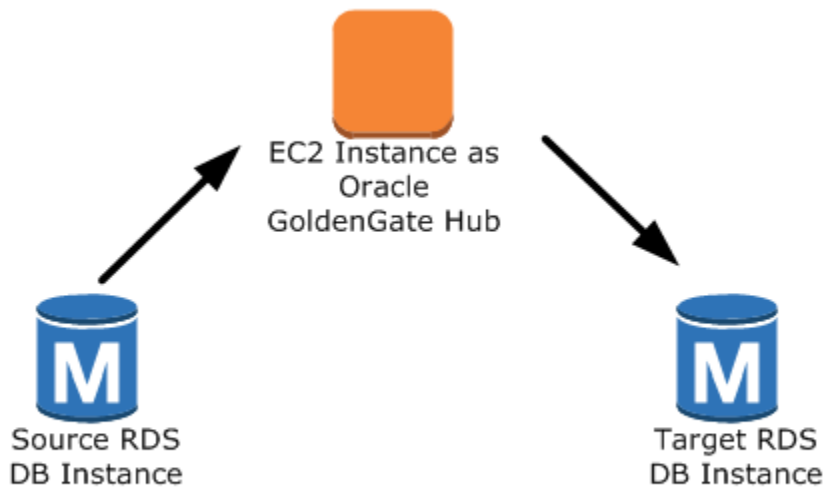
內部部署來源資料庫和 Amazon EC2 中樞

在此案例中，內部部署 Oracle 資料庫做為來源資料庫使用。它連線至一個 Amazon EC2 執行個體。此中樞會向目標 RDS for Oracle 資料庫執行個體提供資料。



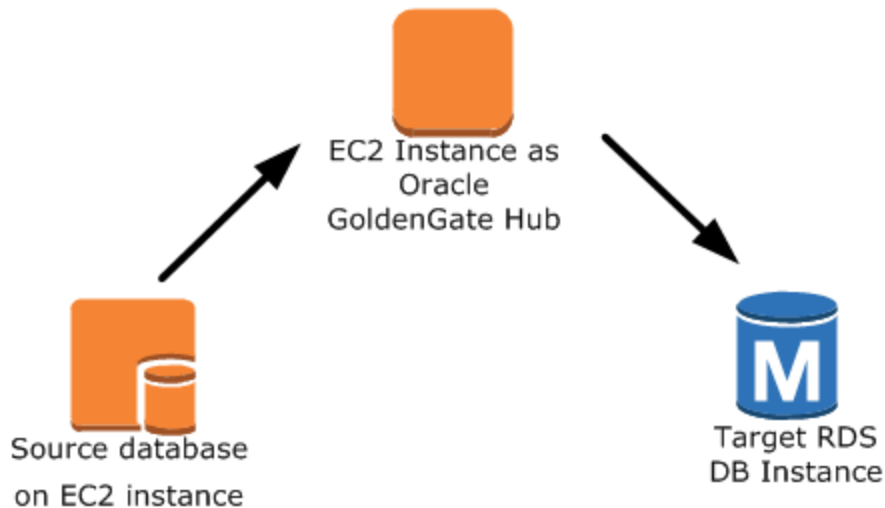
Amazon RDS 來源資料庫和 Amazon EC2 中樞

在此案例中，RDS for Oracle 資料庫執行個體做為來源資料庫使用。它連線至一個 Amazon EC2 執行個體。此中樞會向目標 RDS for Oracle 資料庫執行個體提供資料。



Amazon EC2 來源資料庫和 Amazon EC2 中樞

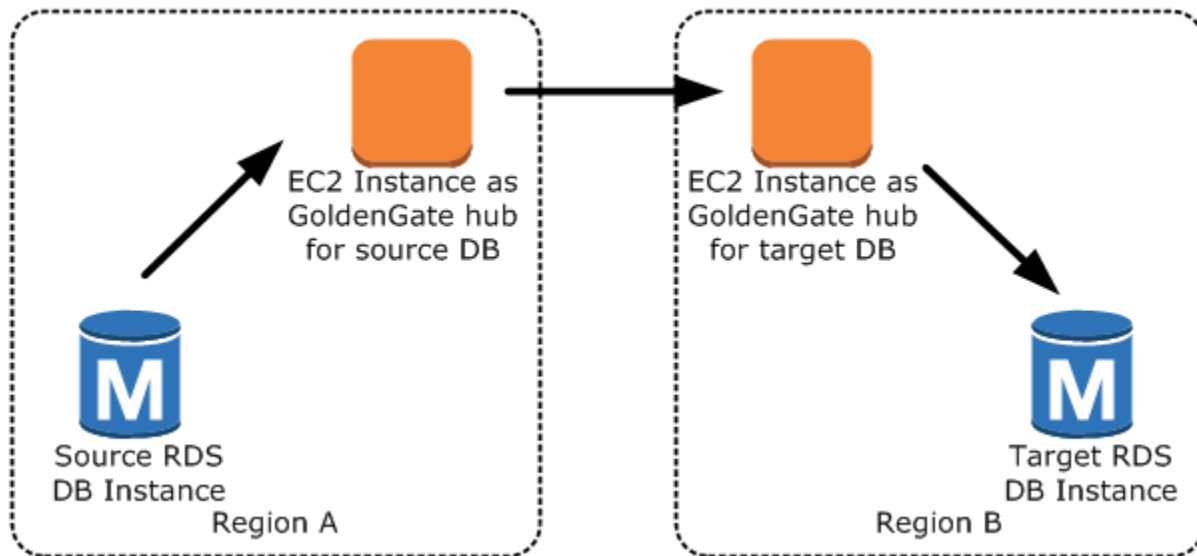
在此案例中，Amazon EC2 執行個體上的 Oracle 資料庫做為來源資料庫使用。它連線至一個 Amazon EC2 執行個體。此中樞會向目標 RDS for Oracle 資料庫執行個體提供資料。



位於不同 AWS 區域的 Amazon EC2 中樞

在這個案例中，Amazon RDS 資料庫執行個體上的 Oracle 資料庫會連接到相同 AWS 區域中的 Amazon EC2 執行個體中樞。集線器已連接至不同 AWS 區域中的 Amazon EC2 執行個體中樞。第二

個中樞將資料提供給與第二個 Amazon EC2 執行個體中樞位於相同 AWS 區域的 Oracle 資料庫執行個體的目標 RDS。



Note

任何影響在內部部署環境中執行 GoldenGate 行 Oracle 的問題，也會影響 GoldenGate 在上執行 Oracle AWS。強烈建議您監視 Oracle 中 GoldenGate 樞，以確保在發生容錯移轉時恢復 EXTRACT 和 REPLICAT 恢復。由於 Oracle 中 GoldenGate 樞是在 Amazon EC2 執行個體上執行，因此 Amazon RDS 不會管理 Oracle 中 GoldenGate 樞，也無法確保其正在執行。

設定甲骨文 GoldenGate

若要 GoldenGate 使用 Amazon RDS 設定 Oracle，請在 Amazon EC2 執行個體上設定中樞，然後設定來源和目標資料庫。以下各節提供了如何設置甲骨文以與亞馬遜 RDS GoldenGate 為甲骨文一起使用的示例。

主題

- [在 Amazon EC2 上設置甲骨文 GoldenGate 中心](#)
- [設定來源資料庫，以便 GoldenGate 在 Amazon RDS 上與甲骨文搭配使用](#)
- [設定目標資料庫，以便 GoldenGate 在 Amazon RDS 上與甲骨文搭配使用](#)

在 Amazon EC2 上設置甲骨文 GoldenGate 中心

若要在 Amazon EC2 執行個體上建立甲骨文 GoldenGate 中樞，您必須先建立具有 Oracle RDBMS 完整用戶端安裝的 Amazon EC2 執行個體。亞馬遜 EC2 執行個體也必須安裝甲骨文 GoldenGate 軟體。Oracle GoldenGate 軟體版本取決於來源和目標資料庫版本。如需有關安裝 Oracle 的詳細資訊 GoldenGate，請參閱 [Oracle GoldenGate 說明文件](#)。

作為 Oracle GoldenGate 中樞使用的 Amazon EC2 執行個體，可存放來源資料庫中的交易資訊，並將其處理為追蹤檔案。若要支援此程序，請確定您符合下列需求：

- 您已針對追蹤檔案配置足夠的儲存體。
- Amazon EC2 執行個體具有足夠的處理能力來管理資料量。
- EC2 執行個體有足夠的記憶體來儲存交易資訊，然後再寫入追蹤檔案。

在 Amazon EC2 執行個體上設定甲骨文 GoldenGate 傳統架構中心

1. 在 Oracle 目錄中建立子目錄 GoldenGate 目錄。

在 Amazon EC2 命令行外殼中 `ggsci`，啟動甲骨文 GoldenGate 命令解釋器。CREATE SUBDIRS 命令會在 /gg 目錄下建立參數、報告和檢查點檔案的子目錄。

```
prompt$ cd /gg
prompt$ ./ggsci

GGSCI> CREATE SUBDIRS
```

2. 設定 mgr.prm 檔案。

下列範例會將下列幾行新增至 \$GGHOME/dirprm/mgr.prm 檔案。

```
PORT 8199
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
```

3. 啟動管理員。

下列範例會啟動 `ggsci` 並執行 `start mgr` 命令。

```
GGSCI> start mgr
```

Oracle 中 GoldenGate 樞現在已準備就緒，可供使用。

設定來源資料庫，以便 GoldenGate 在 Amazon RDS 上與甲骨文搭配使用

請完成下列作業，以設定來源資料庫以搭配 Oracle 使用 GoldenGate。

設定步驟

- [步驟 1：開啟來源資料庫上的補充記錄](#)
- [步驟 2：將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true](#)
- [步驟 3：在來源資料庫上設定日誌保留期間](#)
- [步驟 4：在來源資料庫上建立 Oracle GoldenGate 使用者帳戶](#)
- [步驟 5：在來源資料庫上將帳戶權限授予使用者](#)
- [步驟 6：新增來源資料庫的 TNS 別名](#)

步驟 1：開啟來源資料庫上的補充記錄

若要開啟最低資料庫層級補充記錄，請執行下列 PL/SQL 程序：

```
EXEC rdsadmin.rdsadmin_util.alter_supplemental_logging(p_action => 'ADD')
```

步驟 2：將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true

當您將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true 時，它會允許資料庫服務支援邏輯複寫。如果來源資料庫位於 Amazon RDS 資料庫執行個體，請確定將一個參數群組指派給 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true 的資料庫執行個體。如需 ENABLE_GOLDENGATE_REPLICATION 初始化參數的詳細資訊，請參閱 [Oracle Database 文件](#)。

步驟 3：在來源資料庫上設定日誌保留期間

請確定來源資料庫設定為保留封存的重做日誌。請考量下列準則：

- 指定記錄保留的持續時間 (以小時為單位)。最小值為一小時。
- 將持續時間設定為超過來源資料庫執行個體的任何潛在停機時間、通訊的任何潛在期間，以及來源執行個體聯網問題的任何潛在期間。這樣的持續時間可讓 Oracle 視需要從來源執行環境 GoldenGate 復原記錄。
- 請確定執行個體有足夠的儲存空間可存放這些檔案。

例如，將封存重做日誌的保留期間設為 24 小時。

```
EXEC rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24)
```

如果未啟用日誌保留，或如果保留值太小，您會收到如下訊息：

```
2022-03-06 06:17:27 ERROR OGG-00446 error 2 (No such file or directory)
opening redo log /rdsbdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

因為您的資料庫執行個體會保留您封存的重做日誌，所以請確定您有足夠的空間存放檔案。若要查看您在過去 *num_hours* 小時內已使用多少空間，請使用下列查詢，並將 *num_hours* 改為時數。

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
WHERE NEXT_TIME>=SYSDATE-num_hours/24 AND DEST_ID=1;
```

步驟 4：在來源資料庫上建立 Oracle GoldenGate 使用者帳戶

Oracle 以資料庫使用者身分 GoldenGate 執行，並需要適當的資料庫權限，才能存取來源資料庫的重做和存檔重做日誌。若要提供這些內容，在來源資料庫上建立使用者帳戶。如需 Oracle GoldenGate 使用者帳戶權限的詳細資訊，請參閱 [Oracle 說明文件](#)。

下列陳述式會建立一個名為 oggadm1 的使用者帳戶。

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1 IDENTIFIED BY "password"
DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

步驟 5：在來源資料庫上將帳戶權限授予使用者

在此任務中，您會將必要的帳戶權限授予來源資料庫上的資料庫使用者。

在來源資料庫上授予帳戶權限

1. 使用 SQL 命令 `grant` 和 `rdsadmin.rdsadmin_util` 程序，將必要的權限授與 Oracle 使用 GoldenGate 者帳戶 `grant_sys_object`。下列陳述式會將權限授予一個名為 `oggadm1` 的使用者。

```
GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
GRANT RESOURCE TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT FLASHBACK ANY TABLE TO oggadm1;
GRANT SELECT ANY TABLE TO oggadm1;
GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
EXEC rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
GRANT ALTER ANY TABLE TO oggadm1;
```

2. 授與使用者帳戶成為 Oracle GoldenGate 管理員所需的權限。執行下列 PL/SQL 程式。

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
  grantee           => 'OGGADM1',
  privilege_type    => 'capture',
  grant_select_privileges => true,
  do_grants         => TRUE);
```

若要撤銷權限，請在同一套件中使用程序 `revoke_admin_privilege`。

步驟 6：新增來源資料庫的 TNS 別名

將下列項目新增至 Oracle 主目錄中要由 EXTRACT 程序使用的 `$ORACLE_HOME/network/admin/tnsnames.ora`。如需 `tnsnames.ora` 檔案的詳細資訊，請參閱 [Oracle 文件](#)。

```
OGGSOURCE=
  (DESCRIPTION=
    (ENABLE=BROKEN)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200))
    (CONNECT_DATA=(SERVICE_NAME=ORCL))
  )
```


設定目標資料庫，以便 GoldenGate 在 Amazon RDS 上與甲骨文搭配使用

在此工作中，您可以設定要搭配 Oracle 使用的目標資料庫執行個體 GoldenGate。

設定步驟

- [步驟 1：將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true](#)
- [步驟 2：在目標資料庫上建立 Oracle GoldenGate 使用者帳戶](#)
- [步驟 3：在目標資料庫上授予帳戶權限](#)
- [步驟 4：新增目標資料庫的 TNS 別名](#)

步驟 1：將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true

當您將 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true 時，它會允許資料庫服務支援邏輯複寫。如果來源資料庫位於 Amazon RDS 資料庫執行個體，請確定將一個參數群組指派給 ENABLE_GOLDENGATE_REPLICATION 初始化參數設為 true 的資料庫執行個體。如需 ENABLE_GOLDENGATE_REPLICATION 初始化參數的詳細資訊，請參閱 [Oracle Database 文件](#)。

步驟 2：在目標資料庫上建立 Oracle GoldenGate 使用者帳戶

Oracle 以資料庫使用者身分 GoldenGate 執行，且需要適當的資料庫權限。若要確定它具有這些權限，請在目標資料庫上建立一個使用者帳戶。

下列陳述式會建立一個名為 oggadm1 的使用者。

```
CREATE TABLESPACE administrator;  
CREATE USER oggadm1 IDENTIFIED BY "password"  
  DEFAULT TABLESPACE administrator  
  TEMPORARY TABLESPACE temp;  
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

步驟 3：在目標資料庫上授予帳戶權限

在此任務中，您會將必要的帳戶權限授予目標資料庫上的資料庫使用者。

在目標資料庫上授予帳戶權限

1. 將必要的權限授與目標資料庫上的 Oracle GoldenGate 使用者帳戶。在下列範例中，您會將權限授予 oggadm1。

```
GRANT CREATE SESSION          TO oggadm1;
GRANT ALTER SESSION          TO oggadm1;
GRANT CREATE CLUSTER          TO oggadm1;
GRANT CREATE INDEXTYPE        TO oggadm1;
GRANT CREATE OPERATOR         TO oggadm1;
GRANT CREATE PROCEDURE        TO oggadm1;
GRANT CREATE SEQUENCE         TO oggadm1;
GRANT CREATE TABLE           TO oggadm1;
GRANT CREATE TRIGGER          TO oggadm1;
GRANT CREATE TYPE             TO oggadm1;
GRANT SELECT ANY DICTIONARY   TO oggadm1;
GRANT CREATE ANY TABLE       TO oggadm1;
GRANT ALTER ANY TABLE        TO oggadm1;
GRANT LOCK ANY TABLE         TO oggadm1;
GRANT SELECT ANY TABLE       TO oggadm1;
GRANT INSERT ANY TABLE       TO oggadm1;
GRANT UPDATE ANY TABLE       TO oggadm1;
GRANT DELETE ANY TABLE       TO oggadm1;
```

2. 授與使用者帳戶成為 Oracle GoldenGate 管理員所需的權限。執行下列 PL/SQL 程式。

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
  grantee           => 'OGGADM1',
  privilege_type    => 'apply',
  grant_select_privileges => true,
  do_grants         => TRUE);
```

若要撤銷權限，請在同一套件中使用程序 `revoke_admin_privilege`。

步驟 4：新增目標資料庫的 TNS 別名

將下列項目新增至 Oracle 主目錄中要由 REPLICAT 程序使用的 `$ORACLE_HOME/network/admin/tnsnames.ora`。對於 Oracle 多租戶資料庫，請確定 TNS 別名指向 PDB 的服務名稱。如需 `tnsnames.ora` 檔案的詳細資訊，請參閱 [Oracle 文件](#)。

```
OGGTARGET=
```

```
(DESCRIPTION=
  (ENABLE=BROKEN)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
  (CONNECT_DATA=(SERVICE_NAME=ORCL))
)
```

使用甲骨文的提取和副本實用程序 GoldenGate

Oracle GoldenGate 公用程式 EXTRACT 和共同合 REPLICAT 作，透過使用追蹤檔案的增量交易複寫，讓來源和目標資料庫保持同步。來源資料庫上發生的所有變更都會由 Oracle 現 GoldenGate 場部署或 Amazon EC2 執行個體中樞自動偵測 EXTRACT，然後格式化並傳輸到追蹤檔案。完成初始載入之後，REPLICAT 公用程式會從這些檔案讀取資料並複寫至目標資料庫。

執行甲骨文 GoldenGate 擷取公用程式

EXTRACT 公用程式會從來源資料庫擷取、轉換及輸出資料至追蹤檔案。基本程序如下：

1. EXTRACT 會將交易詳細資訊排入記憶體或暫存磁碟儲存體。
2. 該來源資料庫會遞交該筆交易。
3. EXTRACT 會將交易詳細資訊寫入追蹤檔案。
4. 追蹤檔案會將這些詳細資料路由到 Oracle GoldenGate 現場部署或 Amazon EC2 執行個體中心，然後路由到目標資料庫。

下列步驟會啟動 EXTRACT 公用程式、從來源資料庫 OGGSOURCE 中的 EXAMPLE.TABLE 擷取資料，然後建立追蹤檔案。

執行 EXTRACT 公用程式

1. 在 Oracle GoldenGate 中樞 (現場部署或 Amazon EC2 執行個體) 上設定 EXTRACT 參數檔案。下列清單會顯示名為 \$GGHOME/dirprm/eabc.prm 的範例 EXTRACT 參數檔案。

```
EXTRACT EABC

USERID oggadm1@OGGSOURCE, PASSWORD "my-password"
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPLICATES
```

```
GETAPPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

2. 在 Oracle 中 GoldenGate 樞上，登入來源資料庫並啟動 Oracle GoldenGate 命令列介面 ggsci。以下範例會顯示登入的格式。

```
dblogin oggadm1@OGGSOURCE
```

3. 新增交易資料，開啟資料庫資料表的補充記錄。

```
add trandata EXAMPLE.TABLE
```

4. 使用 ggsci 命令列，搭配下列命令來啟用 EXTRACT 公用程式。

```
add extract EABC tranlog, INTEGRATED tranlog, begin now
add exttrail /path/to/goldengate/dirdat/ab
  extract EABC,
  MEGABYTES 100
```

5. 向資料庫註冊 EXTRACT 公用程式，如此就不會刪除封存日誌。此任務可讓您在必要時復原未遞交的舊交易。若要向資料庫註冊 EXTRACT 公用程式，請使用下列命令。

```
register EXTRACT EABC, DATABASE
```

6. 使用下列命令啟動 EXTRACT 公用程式。

```
start EABC
```

執行 Oracle GoldenGate 複本公用程式

REPLICAT 公用程式會將追蹤檔案中的交易資訊「推送」至目標資料庫。

下列步驟會啟用並啟動 REPLICAT 公用程式，以便它可以將擷取的資料複寫到目標資料庫 OGGTARGET 中的資料表 EXAMPLE.TABLE。

執行 REPLICATE 公用程式

1. 在 Oracle GoldenGate 中樞 (內部部署或 EC2 執行個體) 上設定 REPLICAT 參數檔案。下列清單會顯示名為 \$GGHOME/dirprm/rabc.prm 的範例 REPLICAT 參數檔案。

```
REPLICAT RABC

USERID oggadm1@OGGTARGET, password "my-password"

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

- 登入目標資料庫並啟動 Oracle GoldenGate 命令行介面 (ggsci)。以下範例會顯示登入的格式。

```
dblogin userid oggadm1@OGGTARGET
```

- 使用 ggsci 命令列，新增檢查點資料表。指定的使用者應該是 Oracle GoldenGate 使用者帳戶，而不是目標資料表結構描述擁有者。下列範例會建立名為 gg_checkpoint 的檢查點資料表。

```
add checkpointtable oggadm1.ogchkpt
```

- 若要啟用 REPLICAT 公用程式，請使用下列命令。

```
add replicat RABC EXTTRAIL /path/to/goldengate/dirdat/ab CHECKPOINTTABLE
oggadm1.ogchkpt
```

- 使用下列命令啟動 REPLICAT 公用程式。

```
start RABC
```

監督甲骨文 GoldenGate

當您使用 Oracle 進 GoldenGate 行複製時，請確定 Oracle GoldenGate 處理作業已啟動並執行，而且來源和目標資料庫已同步。您可以使用下列監控工具：

- [Amazon CloudWatch](#) 是一種監控服務，用於此模式來監視 GoldenGate 錯誤日誌。
- [Amazon SNS](#) 是一項訊息通知服務，用於此模式來傳送電子郵件通知。

如需詳細指示，請參閱[使用 Amazon 監控 Oracle GoldenGate 日誌 CloudWatch](#)。

Oracle 疑難排解 GoldenGate

本節說明使用甲骨文 GoldenGate 與 Amazon RDS for Oracle 甲骨文時最常見的問題。

主題

- [開啟線上重做日誌時發生錯誤](#)
- [Oracle GoldenGate 似乎已正確配置，但複製不起作用](#)
- [由於 SYS."_DBA_APPLY_CDR_INFO" 上的查詢，整合式 REPLICAT 執行緩慢。](#)

開啟線上重做日誌時發生錯誤

確定您已將資料庫設定為保留封存的重做日誌。請考量下列準則：

- 指定記錄保留的持續時間 (以小時為單位)。最小值為一小時。
- 將持續時間設定為超過來源資料庫執行個體的任何潛在停機時間、通訊的任何潛在期間，以及來源資料庫執行個體聯網問題的任何潛在期間。這樣的持續時間可讓 Oracle 視需要從來源資料庫執行個體 GoldenGate 復原記錄。
- 請確定執行個體有足夠的儲存空間可存放這些檔案。

如果未啟用日誌保留，或如果保留值太小，您會收到如下訊息：

```
2022-03-06 06:17:27 ERROR OGG-00446 error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Oracle GoldenGate 似乎已正確配置，但複製不起作用

對於預先存在的表格，您必須指定 Oracle 從中 GoldenGate 工作的 SCN。

修正此問題

1. 登入來源資料庫並啟動 Oracle GoldenGate 命令行介面 (ggsci)。以下範例會顯示登入的格式。

```
dblogin userid oggadm1@OGGSOURCE
```

2. 使用 ggsci 命令列，設定 EXTRACT 程序的起始 SCN。下列範例將 EXTRACT 的 SCN 設為 223274。

```
ALTER EXTRACT EABC SCN 223274
start EABC
```

- 登入目標資料庫。以下範例會顯示登入的格式。

```
dblogin userid oggadm1@OGGTARGET
```

- 使用 `ggsci` 命令列，設定 REPLICAT 程序的起始 SCN。下列範例將 REPLICAT 的 SCN 設為 223274。

```
start RABC atcsn 223274
```

由於 SYS."DBA_APPLY_CDR_INFO" 上的查詢，整合式 REPLICAT 執行緩慢。

Oracle GoldenGate 衝突偵測與解決 (CDR) 提供基本的衝突解決常式。例如，CDR 可以解決 INSERT 陳述式的唯一衝突。

當 CDR 解決碰撞時，它可以暫時地將記錄插入到例外資料表 DBA_APPLY_CDR_INFO。整合的 REPLICAT 稍後會刪除這些記錄。在極少數情況下，整合的 REPLICAT 可以處理大量的碰撞，但新的整合 REPLICAT 不會取代它。不是被刪除，而是 DBA_APPLY_CDR_INFO 中的現有資料列是孤立的。任何新的整合 REPLICAT 程序都會變慢，因為它們正在查詢 DBA_APPLY_CDR_INFO 中的孤立資料列。

若要從 DBA_APPLY_CDR_INFO 中移除所有資料列，請使用 Amazon RDS 程序 `rdsadmin.rdsadmin_util.truncate_apply$cdr_info`。此程序是作為 2020 年 10 月發行版本和修補程式更新的一部分發行。此程序可在下列資料庫版本中提供：

- [版本 21.0.0.0.ru-2022-01.rur-2022-01.r1](#) 和更新版本
- [版本 19.0.0.0.ru-2020-10.rur-2020-10.r1](#) 和更新版本

下列範例會截斷資料表 DBA_APPLY_CDR_INFO。

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$cdr_info;
```

在 RDS for Oracle 上使用 Oracle Repository Creation Utility

您可以使用 Amazon RDS 來託管保存結構描述的 RDS for Oracle 資料庫執行個體，以支援您的 Oracle Fusion Middleware 元件。您需在資料庫中為它們建立和填入結構描述，才可以使用 Fusion Middleware 元件。您可以使用 Oracle Repository Creation Utility (RCU) 來建立和填入結構描述。

RCU 支援的版本和授權選項

Amazon RDS 僅支援 Oracle Repository Creation Utility (RCU) 版本 12c。您可以在下列組態中使用 RCU：

- RCU 12c 搭配 Oracle 資料庫 21c
- RCU 12c 搭配 Oracle 資料庫 19c

在您可以使用 RCU 之前，請確認執行下列作業：

- 取得 Oracle Fusion Middleware 的授權。
- 遵循託管儲存庫之 Oracle 資料庫的 Oracle 授權準則。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Fusion Middleware 授權資訊使用者手冊](#)。

融合 MiddleWare 支持甲骨文數據庫企業版和標準版 2 的存儲庫。Oracle 建議將 Enterprise Edition 用於需要分割的生產安裝，和需要重新建置線上索引的安裝。

建立 RDS for Oracle 執行個體之前，請確認您需要的 Oracle 資料庫版本可支援您想要部署的元件。若要尋找您想要部署的 Fusion Middleware 元件和版本的需求，請使用認證矩陣。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Fusion Middleware 支援的系統組態](#)。

Amazon RDS 可視需要支援 Oracle 資料庫版本升級。如需詳細資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

RCU 的要求與限制

若要使用 RCU，您需要 Amazon VPC。Amazon RDS 資料庫執行個體必須只供 Fusion Middleware 元件使用，且不得對公有網際網路公開。因此，請在私有子網路上託管 Amazon RDS 資料庫執行個體，藉此提高安全性。您也需要 RDS for Oracle 資料庫執行個體。如需詳細資訊，請參閱 [建立並連線至 Oracle 資料庫執行個體](#)。

您可以在 Amazon RDS 資料庫執行個體中為任何 Fusion Middleware 元件存放結構描述。下列是已驗證可正確安裝的結構描述：

- Analytics (ACTIVITIES)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Discussions (DISCUSSIONS)
- Metadata Services (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portal and Services (WEBCENTER)
- Portlet Producers (PORTLET)
- Service Table (STB)
- SOA Infrastructure (SOAINFRA)
- User Messaging Service (UCSUMS)
- WebLogic 服務 (維隆斯)

使用 RCU 的指導方針

下列是在此案例中搭配使用資料庫執行個體的一些建議：

- 對於生產工作負載，建議使用多個可用區。如需有關使用多個可用區域的詳細資訊，請參閱[區域、可用區域和 Local Zones](#)。
- 若要獲得額外的安全性，Oracle 建議您使用透明資料加密 (TDE) 來靜態加密您的資料。如果您有包含進階安全性選項的 Enterprise Edition 授權，則可以透過使用 TDE 選項來啟用靜態加密。如需更多詳細資訊，請參閱[Oracle 透明資料加密](#)。

Amazon RDS 也為所有資料庫版本提供靜態加密選項。如需更多詳細資訊，請參閱[加密 Amazon RDS 資源](#)。

- 設定 VPC 安全群組，以允許應用程式伺服器 and Amazon RDS 資料庫執行個體之間的通訊。託管 Fusion Middleware 元件的應用程式伺服器可以採用 Amazon EC2 或現場部署。

執行 RCU

若要建立和填入結構描述以支援 Fusion Middleware 元件，請使用 Oracle Repository Creation Utility (RCU)。您能以不同方式執行 RCU。

主題

- [使用命令列以一個步驟執行 RCU](#)
- [使用命令列以多個步驟執行 RCU](#)
- [在互動式模式中執行 RCU](#)

使用命令列以一個步驟執行 RCU

如果不需要在填入您的任何結構描述之前加以編輯，您可以透過單一步驟執行 RCU。否則，請查看用於以多個步驟執行 RCU 的下列小節。

您可以使用命令列參數 `-silent` 在靜音模式中執行 RCU。當您以無訊息模式執行 RCU 時，您可以建立包含密碼的文字檔案，避免在命令列上輸入密碼。建立一個文字檔案，將 `dbUser` 的密碼放在第一行，以及將每個元件的密碼放在後續的行上。您可以指定密碼檔案的名稱做為對 RCU 命令的最後一個參數。

Example

下列範例會在單一步驟中建立和填入 SOA Infrastructure 元件 (及其相依性) 的結構描述。

對於LinuxmacOS、或Unix：

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
{ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString {dbhost}:{dbport}:{dbname} \
-dbUser {dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix {SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

如需更多詳細資訊，請參閱 Oracle 文件中的[透過命令列執行 Repository Creation Utility](#)。

使用命令列以多個步驟執行 RCU

若要手動編輯結構描述指令碼，可透過多個步驟執行 RCU：

1. 使用 `-generateScript` 命令列參數來為您的結構描述建立指令碼，在 Prepare Scripts for System Load (針對系統負載準備指令碼) 模式中執行 RCU。
2. 手動編輯並執行產生的指令碼 `script_systemLoad.sql`。
3. 使用 `-dataLoad` 命令列參數來填入結構描述，再次於 Perform Product Load (執行產品負載) 模式中執行 RCU。
4. 執行產生的清除指令碼 `script_postDataLoad.sql`。

若要在無提示模式中執行 RCU，請指定命令列參數 `-silent`。在靜音模式中執行 RCU 時，您可以透過建立包含密碼的文字檔案，避免在命令列上輸入密碼。建立一個文字檔案，將 `dbUser` 的密碼放在第一行，以及將每個元件的密碼放在後續的行上。指定密碼檔案的名稱做為傳至 RCU 命令的最後一個參數。

Example

下列範例會建立 SOA Infrastructure 元件及其相依性的結構描述指令碼。

對於LinuxmacOS、或Unix：

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
```

```
-component UCSUMS \  
-component WLS \  
-component SOAINFRA \  
-scriptLocation /tmp/rcuscripts \  
-f < /tmp/passwordfile.txt
```

現在您可以編輯產生的指令碼，請連接至您的 Oracle 資料庫執行個體，並執行指令碼。產生的指令碼名為 `script_systemLoad.sql`。如需連接至 Oracle 資料庫執行個體的詳細資訊，請參閱 [步驟 3：將您的 SQL 連線至 Oracle 資料庫執行個體](#)。

下列範例會填入 SOA Infrastructure 元件 (及其相依性) 的結構描述。

對於LinuxmacOS、或Unix：

```
export JAVA_HOME=/usr/java/jdk1.8.0_65  
${ORACLE_HOME}/oracle_common/bin/rcu \  
-silent \  
-dataLoad \  
-connectString ${dbhost}:${dbport}:${dbname} \  
-dbUser ${dbuser} \  
-dbRole Normal \  
-honorOMF \  
-schemaPrefix ${SCHEMA_PREFIX} \  
-component MDS \  
-component STB \  
-component OPSS \  
-component IAU \  
-component IAU_APPEND \  
-component IAU_VIEWER \  
-component UCSUMS \  
-component WLS \  
-component SOAINFRA \  
-f < /tmp/passwordfile.txt
```

若要完成，您可以連接至 Oracle 資料庫執行個體，並執行清除指令碼。指令碼名為 `script_postDataLoad.sql`。

如需更多詳細資訊，請參閱 Oracle 文件中的 [透過命令列執行 Repository Creation Utility](#)。

在互動式模式中執行 RCU

若要使用 RCU 圖形使用者界面，請在互動式模式中執行 RCU。包括 `-interactive` 參數並省略 `-silent` 參數。如需詳細資訊，請參閱 Oracle 文件中的 [了解 Repository Creation Utility 畫面](#)。

Example

下列範例會在互動式模式中啟動 RCU 並預先填入連接資訊。

對於LinuxmacOS、或Unix：

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

RCU 疑難排解

請注意以下問題。

主題

- [Oracle Managed Files \(OMF\)](#)
- [物件權限](#)
- [Enterprise Scheduler Service](#)

Oracle Managed Files (OMF)

Amazon RDS 會使用 OMF 資料檔案來簡化儲存體管理作業。您可以自訂資料表空間屬性，例如大小和程度管理。不過，如果您在執行 RCU 時指定資料檔案名稱，則資料表空間代碼會失敗，出現 ORA-20900。您可以下列方式，RCU 與 OMF 搭配使用：

- 在 RCU 12.2.1.0 和更新版本中，使用 `-honorOMF` 命令列參數。
- 在 RCU 12.1.0.3 和更新版本中，使用多個步驟並編輯產生的指令碼。如需詳細資訊，請參閱 [使用命令列以多個步驟執行 RCU](#)。

物件權限

因為 Amazon RDS 屬於受管服務，所以您不具備 RDS for Oracle 資料庫執行個體的完整 SYSDBA 存取權。不過，RCU 12c 支援權限較低的使用者。在多數情況下，要建立儲存庫，主要使用者權限即已足夠。

主要帳戶可以直接授與已授與的權限 WITH GRANT OPTION。在某些情況下，當您嘗試授與 SYS 物件權限時，RCU 可能失敗，出現 ORA-01031。您可以重試並執行 `rdsadmin_util.grant_sys_object` 儲存的程序，如下列範例所示：

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','MY_DBA','SELECT');
END;
/
```

如果您嘗試授與 `SCHEMA_VERSION_REGISTRY` 物件的權限 SYS，操作可能會失敗，出現 ORA-20199: Error in `rdsadmin_util.grant_sys_object`。您可以限定資料表 `SCHEMA_VERSION_REGISTRY$` 和檢視 `SCHEMA_VERSION_REGISTRY` 結構描述擁有者的名稱，也就是 SYSTEM，然後重試該作業。或者，您可以建立同義字。以主要使用者身分登入，然後執行下列陳述式：

```
CREATE OR REPLACE VIEW SYSTEM.SCHEMA_VERSION_REGISTRY
  AS SELECT * FROM SYSTEM.SCHEMA_VERSION_REGISTRY$;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY FOR
  SYSTEM.SCHEMA_VERSION_REGISTRY;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY$ FOR SCHEMA_VERSION_REGISTRY;
```

Enterprise Scheduler Service

使用 RCU 卸除 Enterprise Scheduler Service 儲存庫時，RCU 可能會失敗並出現 Error: Component drop check failed。

在 Amazon EC2 執行個體上設定 Oracle Connection Manager

Oracle Connection Manager (CMAN) 是代理伺服器，可將連線請求轉送到資料庫伺服器或其他代理伺服器。您也可以使用以下參數來設定 CMAN：

存取控制

您可以建立規則來篩選掉使用者指定的用戶端請求並接受其他請求。

工作階段多工

您可以透過一個網路連線將多個用戶端工作階段連線傳輸到共用伺服器目標。

通常 CMAN 位於獨立於資料庫伺服器和用戶端主機的主機上。如需詳細資訊，請參閱 Oracle 資料庫說明文件中的[設定 Oracle Connection Manager](#)。

主題

- [CMAN 支援的版本和授權選項](#)
- [CMAN 的要求與限制](#)
- [設定 CMAN](#)

CMAN 支援的版本和授權選項

CMAN 支援 Amazon RDS 所支援的全部版本 Oracle 資料庫企業版。如需詳細資訊，請參閱 [RDS for Oracle 版本](#)。

您可以將 Oracle Connection Manager 安裝在與安裝了 Oracle 資料庫的主機不同的主機上。執行 CMAN 的主機不需要單獨的授權。

CMAN 的要求與限制

為了提供全受管體驗，Amazon RDS 限制作業系統的存取權。您無法修改需要作業系統存取權的資料庫參數。因此，Amazon RDS 不支援要求登入作業系統的 CMAN 功能。

設定 CMAN

設定 CMAN 時，可以在 RDS for Oracle 資料庫之外執行大部分工作。

主題

- [步驟 1：在與 RDS for Oracle 執行個體相同 VPC 中的 Amazon EC2 執行個體上設定 CMAN](#)
- [步驟 2：設定 CMAN 的資料庫參數](#)
- [步驟 3：為參數群組與您的資料庫執行個體建立關聯。](#)

步驟 1：在與 RDS for Oracle 執行個體相同 VPC 中的 Amazon EC2 執行個體上設定 CMAN

若要了解如何設定 CMAN，請按照部落格文章在 [Amazon EC2 for Amazon RDS for Oracle 上設定和使用 Oracle Connection Manager](#) 中的詳細說明操作。

步驟 2：設定 CMAN 的資料庫參數

若使用 Traffic Director Mode 和工作階段多工等 CMAN 功能，請將 REMOTE_LISTENER 參數設為資料庫參數群組中 CMAN 執行個體的地址。請考量下列情況：

- CMAN 執行個體位於 IP 地址為 10.0.159.100 並使用連接埠 1521 的主機上。
- 資料庫 orcla、orclb 以及 orclc 位於獨立的 RDS for Oracle 資料庫執行個體上。

下表顯示如何設定 REMOTE_LISTENER 值。LOCAL_LISTENER 值由 Amazon RDS 自動設定。

資料庫執行個體名稱	資料庫執行個體 IP	本機監聽器值 (自動設定)	遠端監聽器值 (由使用者設定)
orcla	10.0.159.200	(address= (protocol=tcp) (host=10.0.159.200) (port=1521))	10.0.159.100:1521
orclb	10.0.159.300	(address= (protocol=tcp) (host=10.0.159.300) (port=1521))	10.0.159.100:1521
orclc	10.0.159.400	(address= (protocol=tcp) (host=10.0.159.400)	10.0.159.100:1521

資料庫執行個體名稱	資料庫執行個體 IP	本機監聽器值 (自動設定)	遠端監聽器值 (由使用者設定)
		(port=1521))	

步驟 3：為參數群組與您的資料庫執行個體建立關聯。

建立或修改資料庫執行個體以使用您在 [步驟 2：設定 CMAN 的資料庫參數](#) 中設定的參數群組。如需詳細資訊，請參閱 [將資料庫參數群組與資料庫執行個體建立關聯](#)。

在 Amazon RDS 上的 Oracle 上安裝 Siebel 資料庫

您可以使用 Amazon RDS 來託管 Oracle 資料庫執行個體上的 Siebel 資料庫。Siebel 資料庫是 Siebel Customer Relationship Management (CRM) 應用程式架構的一部分。如需說明，請參閱 [Siebel 商業應用程式的一般架構](#)。

使用下列主題可協助您在 Amazon RDS 上的 Oracle 資料庫執行個體上設定 Siebel 資料庫。您也可以了解如何使用 Amazon Web Services，支援 Siebel CRM 應用程式架構所需的其他元件。

Note

若要在 Amazon RDS 上的 Oracle 上安裝 Siebel 資料庫，您需要使用主要使用者帳戶。您不需要 SYSDBA 權限，主要使用者權限就足夠了。如需更多詳細資訊，請參閱 [主要使用者帳戶權限](#)。

授權和版本

若要在 Amazon RDS 上安裝 Siebel 資料庫，您必須使用自己的 Oracle 資料庫授權，以及自己的 Siebel 授權。您必須對資料庫執行個體類別和 Oracle 資料庫版本具有適當的 Oracle 資料庫授權 (含軟體更新授權和支援)。如需更多詳細資訊，請參閱 [RDS for Oracle 授權選項](#)。

在此案例中，Oracle Database Enterprise Edition 是獲得 Siebel 認證的唯一版本。Amazon RDS 支援 Siebel CRM 版本 15.0 或 16.0。

Amazon RDS 支援資料庫版本升級。如需更多詳細資訊，請參閱 [升級資料庫執行個體引擎版本](#)。

開始之前

開始之前，您需要 Amazon VPC。由於 Amazon RDS 資料庫執行個體僅需供 Siebel Enterprise Server 使用，而不用公開至公有網際網路，因此系統會在更安全的私有子網路中託管 Amazon RDS 資料庫執行個體。如需如何建立 Amazon VPC 以搭配 Siebel CRM 使用的相關資訊，請參閱 [建立並連線至 Oracle 資料庫執行個體](#)。

開始之前，您也需要 Oracle 資料庫執行個體。如需建立要與 Siebel CRM 搭配使用之 Oracle 資料庫執行個體的相關資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

安裝並設定 Siebel 資料庫

在建立 Oracle 資料庫執行個體之後，您可以安裝 Siebel 資料庫。安裝資料庫的方式為建立資料表擁有者和管理員帳戶、安裝預存程序和函數，然後執行 Siebel 資料庫設定精靈。如需詳細資訊，請參閱 [在 RDBMS 上安裝 Siebel 資料庫](#)。

若要執行 Siebel 資料庫設定精靈，您需要使用主要使用者帳戶。您不需要 SYSDBA 權限，主要使用者權限就足夠了。如需更多詳細資訊，請參閱 [主要使用者帳戶權限](#)。

將其他 Amazon RDS 功能與 Siebel 資料庫搭配使用

在建立 Oracle 資料庫執行個體之後，您可以使用其他的 Amazon RDS 功能，來協助您自訂 Siebel 資料庫。

利用 Oracle Statspack 選項收集統計資料

您可以透過使用資料庫選項群組中的選項，將功能新增至資料庫執行個體。當您建立了 Oracle 資料庫執行個體時，表示您已使用預設資料庫選項群組。如果您想要將功能新增至資料庫，您可以針對資料庫執行個體建立新的選項群組。

如果想要在 Siebel 資料庫上收集效能統計資料，您可以新增 Oracle Statspack 功能。如需更多詳細資訊，請參閱 [Oracle Statspack](#)。

有些選項變更會立即套用，而有些選項變更則會在資料庫執行個體的下一次維護時段套用。如需更多詳細資訊，請參閱 [使用選項群組](#)。在建立自訂的選項群組之後，請修改您的資料庫執行個體以連接它。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

使用參數調校效能

您可以透過使用資料庫參數群組中的參數，管理資料庫引擎組態。當您建立了 Oracle 資料庫執行個體時，表示您已使用預設資料庫參數群組。如果您想要自訂資料庫組態，您可以針對資料庫執行個體建立新的參數群組。

當您根據參數類型變更參數時，變更會立即套用，或在您手動重新啟動資料庫執行個體之後套用。如需更多詳細資訊，請參閱 [使用參數群組](#)。在建立自訂的參數群組之後，請修改您的資料庫執行個體以連接它。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

若要針對 Siebel CRM 最佳化您的 Oracle，您可以自訂特定參數。下表顯示一些建議的參數設定。如需調整 Siebel CRM 效能的詳細資訊，請參閱 [Siebel CRM Performance Tuning Guide \(Siebel CRM 效能調校指南\)](#)。

參數名稱	預設值	最佳 Siebel CRM 效能的指導
_always_semi_join	CHOOSE	OFF
_b_tree_bitmap_plans	TRUE	FALSE
_like_with_bind_as_equality	FALSE	TRUE
_no_or_expansion	FALSE	FALSE
_optimize_r_join_sel_sanity_check	TRUE	TRUE
_optimize_r_max_permutations	2000	100
_optimize_r_sortmerge_join_enabled	TRUE	FALSE
_partition_view_enabled	TRUE	FALSE
open_cursors	300	至少要 2000 。

建立快照

在建立 Siebel 資料庫之後，您可以使用 Amazon RDS 的快照功能來複製資料庫。如需更多詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#) 及 [從資料庫快照還原](#)。

支援其他 Siebel CRM 元件

除了 Siebel 資料庫外，您也可以使用 Amazon Web Services 來支援 Siebel CRM 應用程式架構的其他元件。您可以在下表中找到 Amazon AWS 針對其他 Siebel CRM 元件所提供之支援的詳細資訊。

Siebel CRM 元件	Amazon Sup AWS port
Siebel Enterprise (含一部或多部 Siebel 伺服器)	<p>您能夠在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上託管 Siebel 伺服器。您可以使用 Amazon EC2，按您的需要啟動任意數量的虛擬伺服器。使用 Amazon EC2，您可以輕鬆地擴展或縮減，以處理需求中的變更。如需詳細資訊，請參閱 什麼是 Amazon EC2 ?</p> <p>您可以將伺服器與資料庫執行個體放在同一個 VPC 中，並使用 VPC 安全群組來存取資料庫。如需更多詳細資訊，請參閱 在 VPC 中使用資料庫執行個體。</p>
Web 伺服器 (含 Siebel Web 伺服器擴充套件)	<p>您可以在多個 EC2 執行個體上安裝多部 Web 伺服器。然後，您可以使用 Elastic Load Balancing，將傳入流量分散至各個執行個體。如需詳細資訊，請參閱 什麼是 Elastic Load Balancing ?</p>
Siebel Gateway Name Server	<p>您可以在 EC2 執行個體上託管 Siebel Gateway Name Server。然後，您可以將伺服器與資料庫執行個體放在同一個 VPC 中，並使用 VPC 安全群組來存取資料庫。如需更多詳細資訊，請參閱 在 VPC 中使用資料庫執行個體。</p>

Oracle Database 引擎版本備註

可讓您的 Amazon RDS for Oracle 資料庫執行個體保持最新的更新項目。如果您套用更新，便可以確信您的資料庫執行個體執行的是經過 Oracle 與 Amazon 測試的資料庫軟體。我們不支援對個別 RDS for Oracle 資料庫執行個體套用一次性修補程式。

建立新的資料庫執行個體時，您可以指定使用目前支援的任何 Oracle Database 版本。您可以指定主要版本 (例如，Oracle Database 19c 版本)，以及特定主要版本支援的任何次要版本。若未指定版本，Amazon RDS 會預設使用支援的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 預設會使用您指定之主要版本的最新發行版本。如需查看支援版本清單和新建資料庫執行個體的預設值，請使用 [describe-db-engine-versions](#) AWS CLI 命令。

如需有關 Amazon RDS 支援的 Oracle Database 版本的詳細內容，請參閱 [Amazon RDS for Oracle 版本備註](#)。

Amazon RDS for PostgreSQL

Amazon RDS 支援執行幾個 PostgreSQL 版本的資料庫執行個體。如需可用版本的清單，請參閱 [可用的 PostgreSQL 資料庫版本](#)。

Note

排定於 2022 年 4 月 26 日棄用 PostgreSQL 9.6。如需詳細資訊，請參閱 [棄用 PostgreSQL 9.6 版](#)。

您可以建立資料庫執行個體和資料庫快照、point-in-time 還原和備份。執行 PostgreSQL 的資料庫執行個體支援多可用區域部署、僅供讀取複本、佈建 IOPS，且可以在 Virtual Private Cloud (VPC) 內建立。您也可以使用 Secure Sockets Layer (SSL) 來連線至執行 PostgreSQL 的資料庫執行個體。

建立資料庫執行個體之前，請務必完成 [設定 Amazon RDS](#) 中的步驟。

您可以從用戶端電腦使用任何標準 SQL 用戶端應用程式對執行個體執行命令。此類應用程式包括 pgAdmin (適用於 PostgreSQL 的常見開放原始碼管理和開發工具) 或 psql (隨 PostgreSQL 一併安裝的命令列公用程式)。為了提供受管理的服務體驗，Amazon RDS 並不會提供資料庫執行個體的主機存取權。此外，其也會將存取權限制在某些需要進階權限的系統程序和資料表。Amazon RDS 可支援使用任何標準 SQL 用戶端應用程式存取資料庫執行個體上的資料庫。Amazon RDS 不允許使用者利用 Telnet 或安全殼層 (SSH) 直接託管資料庫執行個體的存取權。

Amazon RDS for PostgreSQL 符合多項業界標準。舉例來說，您可以使用 Amazon RDS for PostgreSQL 資料庫來建置符合 HIPAA 規範的應用程式並存放醫療保健相關資訊。這包括依據與 AWS 簽署的完整商業夥伴協議 (BAA) 儲存受保護醫療資訊 (PHI)。Amazon RDS for PostgreSQL 亦符合美國聯邦風險與授權管理計劃 (FedRAMP) 的安全性要求。Amazon RDS for PostgreSQL 已獲得 FedRAMP 聯合授權委員會 (JAB) 臨時授權委員會 (JAB) 在區域內的 FedRAMP 高基準運作授權機構 (P-ATO)。AWS GovCloud (US) 如需支援的合規標準的詳細資訊，請參閱 [AWS 雲端合規](#)。

若要將 PostgreSQL 資料匯入資料庫執行個體，請遵循 [將資料匯入 Amazon RDS 上的 PostgreSQL](#) 一節的資訊。

主題

- [Amazon RDS for PostgreSQL 的常用管理任務](#)
- [使用資料庫預覽環境](#)

- [資料庫預覽環境中的第 17 版](#)
- [資料庫預覽環境中的 PostgreSQL 第 16 版](#)
- [可用的 PostgreSQL 資料庫版本](#)
- [支援的 PostgreSQL 擴充功能版本](#)
- [使用 Amazon RDS for PostgreSQL 支援的 PostgreSQL 功能](#)
- [連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)
- [使用 SSL/TLS 保護連接至 RDS for PostgreSQL 的連線](#)
- [搭配 Amazon RDS for PostgreSQL 使用 Kerberos 身分驗證](#)
- [針對傳出網路存取使用自訂 DNS 伺服器。](#)
- [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)
- [升級 PostgreSQL 資料庫快照引擎版本](#)
- [使用 Amazon RDS for PostgreSQL 的僅供讀取複本](#)
- [使用 Amazon RDS Optimized Reads 改善 RDS for PostgreSQL 的查詢效能](#)
- [將資料匯入 Amazon RDS 上的 PostgreSQL](#)
- [將資料從 RDS for PostgreSQL 資料庫執行個體匯出至 Amazon S3](#)
- [從 叫用 AWS Lambda 函數](#)
- [Amazon RDS for PostgreSQL 的常用 DBA 任務](#)
- [調校 RDS for PostgreSQL 的等待事件](#)
- [使用 Amazon DevOps Guru 主動洞察，調校 RDS for PostgreSQL](#)
- [搭配 Amazon RDS for PostgreSQL 使用 PostgreSQL 擴充功能](#)
- [使用 Amazon RDS for PostgreSQL 支援的外部資料包裝函式](#)
- [使用適用於 PostgreSQL 的受信任語言延伸模組](#)

Amazon RDS for PostgreSQL 的常用管理任務

下列是您可在執行 Amazon RDS for PostgreSQL 資料庫執行個體時使用的一般管理任務，並隨附各工作相關說明文件的連結。

任務區域	相關文件
Amazon RDS 首次使用設定	設定 Amazon RDS

任務區域	相關文件
<p>在建立資料庫執行個體前，請務必完成一些先決條件。例如，根據預設，建立資料庫執行個體時也包含可防止存取的防火牆。您需要以正確的 IP 地址和網路組態建立安全群組，才能存取資料庫執行個體。</p>	
<p>Understanding Amazon RDS DB instances (認識 Amazon RDS 資料庫執行個體)</p> <p>如果您要針對生產用途建立資料庫執行個體，您應該了解執行個體類別、儲存體類型和佈建的 IOPS 在 Amazon RDS 中的運作方式。</p>	<p>資料庫執行個體類別</p> <p>Amazon RDS 儲存類型</p> <p>佈建 IOPS SSD 儲存體</p>
<p>查找可用的 PostgreSQL 版本</p> <p>Amazon RDS 支援數個 PostgreSQL 版本。</p>	<p>可用的 PostgreSQL 資料庫版本</p>
<p>設定高可用性和容錯移轉支援</p> <p>生產資料庫執行個體應該使用多個可用區部署。「異地同步備份部署」可提高資料庫執行個體的可用性、資料耐用性和容錯能力。</p>	<p>設定及管理多可用區部署</p>
<p>認識 Amazon Virtual Private Cloud (VPC) 網路</p> <p>如果您的 AWS 帳戶具有預設 VPC，則會在預設 VPC 中自動建立資料庫執行個體。在某些情況下，您的帳戶可能沒有預設 VPC，而且您可能會想要在 VPC 中的資料庫執行個體。在這些情況下，建立資料庫執行個體前，建立 VPC 和子網路群組。</p>	<p>在 VPC 中使用資料庫執行個體</p>
<p>將資料匯入 Amazon RDS PostgreSQL</p> <p>您可使用多種不同的工具，將資料匯入 Amazon RDS 上的 PostgreSQL 資料庫執行個體。</p>	<p>將資料匯入 Amazon RDS 上的 PostgreSQL</p>

任務區域	相關文件
<p>設定唯讀僅供讀取複本 (主要和待命)</p> <p>RDS for PostgreSQL 支援在相同區域以及與主執行個體不同 AWS 區域中的僅供讀取複本。</p>	<p>使用資料庫執行個體僅供讀取複本</p> <p>使用 Amazon RDS for PostgreSQL 的僅供讀取複本</p> <p>以不同的方式建立僅供讀取複本 AWS 區域</p>
<p>了解安全群組</p> <p>根據預設，建立資料庫執行個體時會同時建立防火牆，可防止執行個體遭受不適當的存取。若要透過該防火牆提供存取權，請編輯與託管資料庫執行個體 VPC 相關聯的 VPC 安全群組的傳入規則。</p>	<p>使用安全群組控制存取</p>
<p>設定參數群組和功能</p> <p>若要變更資料庫執行個體的預設參數，請建立自訂資料庫參數群組並將設定變更為如此。如果在建立資料庫執行個體之前執行此作業，您可以在建立執行個體時選擇自訂資料庫參數群組。</p>	<p>使用參數群組</p>
<p>連線至 PostgreSQL 資料庫執行個體</p> <p>建立安全群組並與資料庫執行個體建立關聯後，您可以使用任何標準 SQL 用戶端應用程式 (例如 psql 或 pgAdmin) 來連線至資料庫執行個體。</p>	<p>連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體</p> <p>將 SSL 與 PostgreSQL 資料庫執行個體搭配使用</p>
<p>備份與還原資料庫執行個體</p> <p>您可以設定資料庫執行個體以進行自動備份，或取得手動快照，然後從備份或快照還原執行個體。</p>	<p>備份、還原和匯出資料</p>
<p>監控資料庫執行個體的活動和效能</p> <p>您可以使用 CloudWatch Amazon RDS 指標、事件和增強型監控來監控 PostgreSQL 資料庫執行個體。</p>	<p>在 Amazon RDS 主控台中檢視指標</p> <p>檢視 Amazon RDS 事件</p>

任務區域	相關文件
<p>升級 PostgreSQL 資料庫版本</p> <p>您可以對 PostgreSQL 資料庫執行個體進行主要和次要版本升級。</p>	<p>升級 Amazon RDS 的 PostgreSQL 資料庫引擎</p> <p>選擇 PostgreSQL 的主要版本升級</p>
<p>使用日誌檔案</p> <p>您可以存取 PostgreSQL 資料庫執行個體の日誌檔案。</p>	<p>RDS for PostgreSQL 資料庫日誌檔</p>
<p>了解 PostgreSQL 資料庫執行個體的最佳實務</p> <p>尋找在 Amazon RDS 上使用 PostgreSQL 的一些最佳實務。</p>	<p>使用 PostgreSQL 的最佳實務</p>

以下是本指南中其他章節的列表，這些章節可幫助您了解並使用 RDS for PostgreSQL 的重要功能：

- [了解 PostgreSQL 角色和許可](#)
- [控制使用者對 PostgreSQL 資料庫的存取](#)
- [在 RDS for PostgreSQL 資料庫執行個體上搭配使用參數](#)
- [了解 RDS for PostgreSQL 支援的記錄機制](#)
- [在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理](#)
- [針對傳出網路存取使用自訂 DNS 伺服器。](#)

使用資料庫預覽環境

PostgreSQL 社群會持續發佈新的 PostgreSQL 版本和新的延伸模組。包括 Beta 版本。這讓 PostgreSQL 使用者有機會提早嘗試新的 PostgreSQL 版本。若要進一步了解 PostgreSQL 社群 Beta 發程序，請參閱 PostgreSQL 文件中的 [Beta 資訊](#)。同樣地，Amazon RDS 會使某些 PostgreSQL Beta 版本當作預覽版本提供。這可讓您使用預覽版本建立資料庫執行個體，並在資料庫預覽環境中測試其功能。

資料庫預覽環境中的 RDS for PostgreSQL 資料庫執行個體是類似於其他 RDS for PostgreSQL 執行個體的功能。不過，您無法使用預覽版本進行生產。

請謹記下列重要限制：

- 所有資料庫執行個體在建立後 60 天就會刪除，也會一併刪除任何備份和快照。
- 您只能在以 Amazon VPC 服務為基礎的 Virtual Private Cloud (VPC) 中建立資料庫執行個體。
- 您只能使用一般用途 SSD 和佈建 IOPS SSD 儲存裝置。
- 您無法從資料庫執行個體的 Sup AWS port 取得協助。相反，您可以將您的問題發佈到 AWS 受管理的問答社群 [AWS Re: post](#)。
- 您無法將資料庫執行個體的快照複製到生產環境。

預覽版支援下列選項：

- 您只能使用 M6i、R6i、M6g、M5、T3、R6g 和 R5 執行個體類型建立資料庫執行個體。如需 RDS 執行個體類別的詳細資訊，請參閱 [資料庫執行個體類別](#)。
- 您可以同時使用單一可用區和異地同步備份部署。
- 您可以使用標準 PostgreSQL 傾印和載入函數，從資料庫預覽環境匯出資料庫，或匯入資料庫至資料庫預覽環境。

資料庫預覽環境中不支援的功能

資料庫預覽環境中無法使用下列功能：

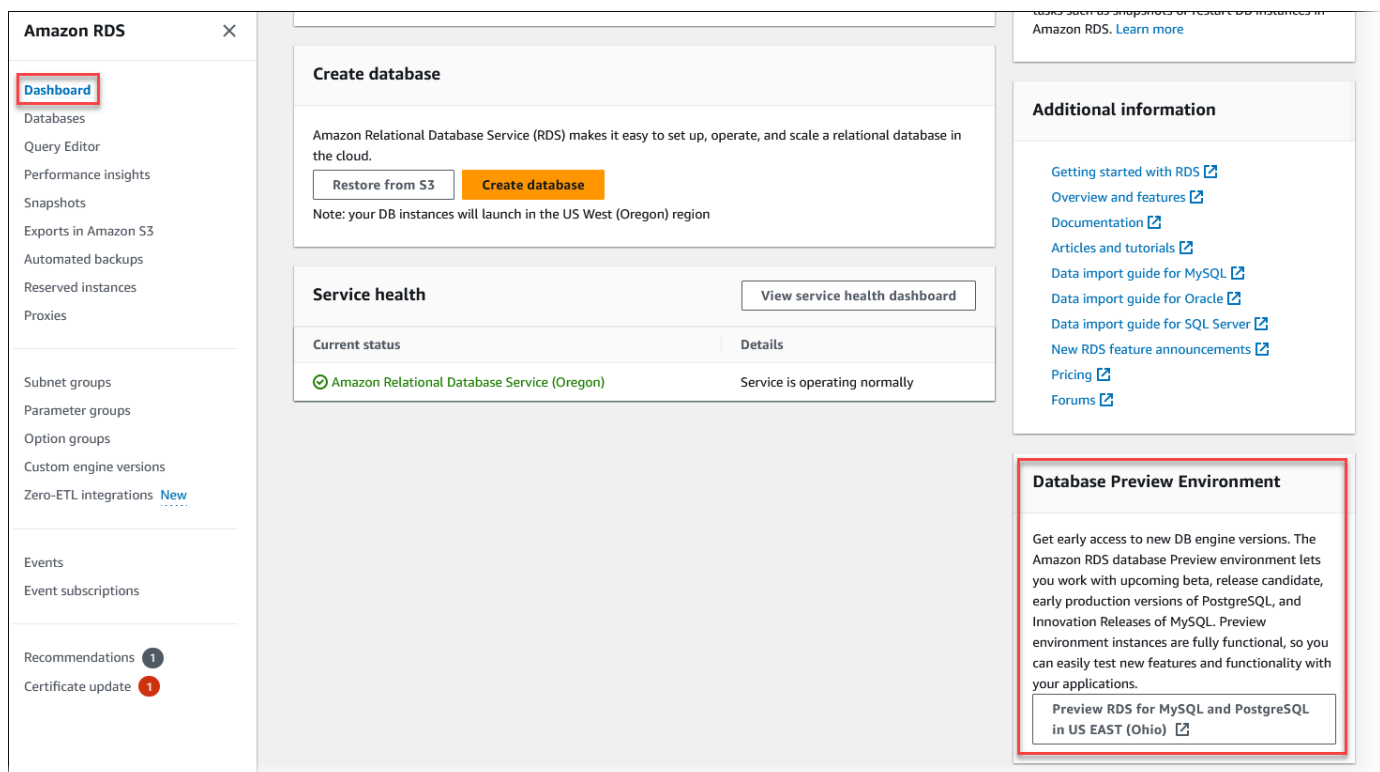
- 跨區域快照複製
- 跨區域僅供讀取複本

在資料庫預覽環境中建立新的資料庫執行個體

使用下列程序在預覽環境中建立資料庫執行個體。


在資料庫預覽環境中建立資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從導覽窗格中選擇 Dashboards (儀表板)。
3. 在儀表板頁面中，找出儀表板頁面上的 Database Preview Environment (資料庫預覽環境) 區段，如下圖所示。



您可以直接導覽至 [資料庫預覽環境](#)。在繼續之前，您必須確認並接受限制。

Database Preview Environment Service Agreement ✕

The Amazon RDS Database Preview Environment is not covered by the Amazon RDS service level agreement (SLA), published at <https://aws.amazon.com/rds/sla> 

Do not use the Amazon RDS Database Preview Environment for production purposes. You should only use this environment for development and testing.

Certain use cases might fail in this environment - for example, upgrading from a previous version is not supported.

I acknowledge this limited service agreement for the Amazon RDS Database Preview Environment and that I should only use this environment for development and testing.

Cancel Accept

- 若要建立 RDS for PostgreSQL 資料庫執行個體，請遵循與建立任何 Amazon RDS 資料庫執行個體相同的程序。如需詳細資訊，請參閱 [建立資料庫執行個體](#) 中的 [主控台](#) 程序。

若要使用 RDS API 或在資料庫預覽環境中建立執行個體 AWS CLI，請使用下列端點。

```
rds-preview.us-east-2.amazonaws.com
```


資料庫預覽環境中的第 17 版

 這是針對第 17 版 PostgreSQL 預覽文件。內容可能變動。

現在可在 Amazon RDS 資料庫預覽環境中使用 PostgreSQL 第 17 版測試版。第 17 版測試版 1 包含了以下 PostgreSQL 文檔中描述的幾個改進：PostgreSQL 17 測試版 1 已發**布**！

如需資料庫預覽環境的資訊，請參閱 [the section called “資料庫預覽環境”](#)。若要從主控台存取預覽環境，請選取 <https://console.aws.amazon.com/rds-preview/>。

資料庫預覽環境中的 PostgreSQL 第 16 版

 這是 Amazon RDS PostgreSQL 第 16 版的預覽文件。內容可能變動。

Note

RDS for PostgreSQL 16.0 版發行後，資料庫預覽環境中將不再支援 RDS for PostgreSQL 16 RC1、16 Beta 3、16 Beta 2 及 16 Beta 1 版。

PostgreSQL 16.0 版現可在 Amazon RDS 資料庫預覽環境中使用。PostgreSQL 第 16 版包含下列 PostgreSQL 文件中所述的數個改善：

- [PostgreSQL 16 已發行](#)
- [PostgreSQL 16 RC1 已發行](#)
- [PostgreSQL 16 Beta 3 已發行！](#)
- [PostgreSQL 16 Beta 2 已發行！](#)
- [PostgreSQL 16 Beta 1 已發行！](#)

如需資料庫預覽環境的資訊，請參閱 [the section called “資料庫預覽環境”](#)。若要從主控台存取預覽環境，請選取 <https://console.aws.amazon.com/rds-preview/>。

可用的 PostgreSQL 資料庫版本

Amazon RDS 支援執行幾個 PostgreSQL 版本的資料庫執行個體。建立新的資料庫執行個體時，您可以指定使用目前可用的任一 PostgreSQL 版本。您可以指定主要版本 (例如 PostgreSQL 14)，以及所指定主要版本的任何可用次要版本。若未指定版本，Amazon RDS 會預設使用可用的版本，通常是最新版本。若已指定主要版本，但未指定次要版本，Amazon RDS 會預設使用您指定主要版本的最新次要版本。

若要查看可用版本的清單以及新建立的資料庫執行個體的預設值，請使用 [describe-db-engine-versions](#) AWS CLI 指令。例如，若要顯示預設 PostgreSQL 引擎版本，請使用下列命令：

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

如需有關 Amazon RDS 支援的 PostgreSQL 版本的詳細內容，請參閱 [Amazon RDS for PostgreSQL 版本備註](#)。

如果您還沒準備好在 RDS 標準支援結束日期之前手動升級到新的主要引擎版本，Amazon RDS 會在 RDS 標準 Support 援結束日期後自動在 Amazon RDS 擴展支援中註冊您的資料庫。然後，您可以繼續執行 RDS 版 PostgreSQL 及更新版本。如需詳細資訊，請參閱 [使用 Amazon RDS 延長支援](#) 和 [Amazon RDS 定價](#)。

棄用 PostgreSQL 第 10 版

2023 年 4 月 17 日，Amazon RDS 預計依照下列排程棄用 PostgreSQL 第 10 版。我們建議您採取行動，並將在主要版本 10 上執行的 PostgreSQL 資料庫升級至更新版本，例如 PostgreSQL 第 14 版。若要從 10.19 以前的 PostgreSQL 版本升級您的 RDS 主要版本 10 資料庫執行個體，建議您先升級至 10.19 版，再升級至第 14 版。如需詳細資訊，請參閱 [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。

動作或建議	日期
PostgreSQL 社群打算棄用 PostgreSQL 10，而且在此日期之後不會提供任何安全性修補程式。	2022 年 11 月 10 日
開始將 RDS for PostgreSQL 10 資料庫執行個體升級至更新的主要版本，例如 PostgreSQL 14。雖然您可以繼續還原 PostgreSQL 10 快照並使用第 10 版建立	2023 年 2 月 14 日截止

動作或建議	日期
僅供讀取複本，但請留意此棄用時間表中的其他重要日期及其影響。	
在此日期之後，您無法使用主要版本 10 從 AWS Management Console 或建立新的 Amazon RDS 執行個體。AWS CLI	2023 年 2 月 14 日
在此日期之後，Amazon RDS 會自動將 PostgreSQL 10 執行個體升級到第 14 版。如果您還原 PostgreSQL 10 資料庫快照，Amazon RDS 會自動將還原的資料庫升級到 PostgreSQL 14。	2023 年 4 月 17 日

如需有關第 10 版淘汰 PostgreSQL RDS 的詳細資訊，請參閱 [\[公告\] : Re: 貼文中的版本取代](#)。AWS

棄用 PostgreSQL 9.6 版

2022 年 3 月 31 日，Amazon RDS 預計依照下列排程取代 PostgreSQL 9.6。之前宣佈的日期為 2022 年 1 月 18 日，現將延長至 2022 年 4 月 26 日。您應盡快將所有 PostgreSQL 9.6 資料庫執行個體升級至 PostgreSQL 12 或更新版本。建議您先升級到次要版本 9.6.20 或更新版本，然後直接升級到 PostgreSQL 12，而不是升級到中間的主要版本。如需詳細資訊，請參閱 [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。

動作或建議	日期
PostgreSQL 社群已終止對 PostgreSQL 9.6 的支援，且不再提供此版本的錯誤修正或安全修補程式。	2021 年 11 月 11 日
請盡快將 RDS for PostgreSQL 9.6 資料庫執行個體升級至 PostgreSQL 12 或更新版本。雖然您可以繼續還原 PostgreSQL 9.6 快照並使用 9.6 版建立僅供讀取複本，但請留意此棄用時間表中的其他重要日期及其影響。	2022 年 3 月 31 日截止
在此日期之後，您無法使用主要版本 9.6 從 AWS Management Console 或建立新的 Amazon RDS 執行個體。AWS CLI	2022 年 3 月 31 日

動作或建議	日期
在此日期之後，Amazon RDS 會自動將 PostgreSQL 9.6 執行個體升級到 12 版。如果您還原 PostgreSQL 9.6 資料庫快照，Amazon RDS 會自動將還原的資料庫升級到 PostgreSQL 12。	2022 年 4 月 26 日

已取代的版本 Amazon RDS for PostgreSQL

RDS for PostgreSQL 9.5 將於 2021 年 3 月棄用。如需有關 RDS 版本 9.5 淘汰 PostgreSQL 詳細資訊，請參閱[從 Amazon RDS for PostgreSQL 9.5 版升級](#)。

若要進一步了解 RDS for PostgreSQL 的棄用政策，請參閱[Amazon RDS 常見問答集](#)。如需 PostgreSQL 版本的詳細資訊，請參閱 PostgreSQL 說明文件中的[版本控制政策](#)。

支援的 PostgreSQL 擴充功能版本

RDS for PostgreSQL 支援許多 PostgreSQL 擴充功能。PostgreSQL 社群有時會將這些稱為模組。擴充功能可以擴充 PostgreSQL 引擎所提供的功能。您可以在該 PostgreSQL 版本的預設資料庫參數群組中找到 Amazon RDS 支援的延伸清單。您也可以使用 `psql` 來顯示 `rds.extensions` 參數，即可看到目前的延伸清單，如下列範例所示。

```
SHOW rds.extensions;
```

Note

在 `rds.extensions` 中使用 `psql` 參數時，次要版本中新增的參數可能不會正確顯示。

從 RDS for PostgreSQL 13 開始，某些擴充功能可由資料庫使用者安裝，而不是 `rds_superuser`。這些稱為受信任的擴充功能。如需進一步了解，請參閱 [PostgreSQL 可信任延伸](#)。

特定 RDS for PostgreSQL 版本支援 `rds.allowed_extensions` 參數。此參數可讓 `rds_superuser` 限制 RDS for PostgreSQL 資料庫執行個體中安裝的擴充功能。如需詳細資訊，請參閱 [限制安裝 PostgreSQL 擴充功能](#)。

如需有關 Amazon RDS 支援的 PostgreSQL 擴充功能的詳細清單，請參閱 Amazon RDS for PostgreSQL 版本備註中的 [Amazon RDS 上支援的 PostgreSQL 擴充功能](#)。

限制安裝 PostgreSQL 擴充功能

您可以限制 PostgreSQL 資料庫執行個體上能安裝哪些擴充功能。依預設，並未設定此參數，因此若使用者具有許可權限，則可新增任何受支援的擴充功能。若要執行這項操作，請將 `rds.allowed_extensions` 參數設定為逗號分隔的副檔名字串。將擴充功能清單新增至此參數，可明確識別 RDS for PostgreSQL 資料庫執行個體可使用的擴充功能。然後，只有這些擴充功能才能安裝在 PostgreSQL 資料庫執行個體中。

`rds.allowed_extensions` 參數的預設字串是 `*`，這表示可以安裝引擎版本可用的任何擴充功能。變更 `rds.allowed_extensions` 參數不需要重新啟動資料庫，因為它是動態參數。

PostgreSQL 資料庫執行個體引擎必須是下列其中一個版本，才能使用 `rds.allowed_extensions` 參數：

- PostgreSQL 本

- 第 15 版及所有更高版本
- PostgreSQL 14 和所有更高的版本
- PostgreSQL 13.3 和更高的次要版本
- PostgreSQL 12.7 和更高的次要版本

若要查看允許安裝哪些擴充功能，請使用下列 `psql` 命令。

```
postgres=> SHOW rds.allowed_extensions;
 rds.allowed_extensions
-----
*
```

如果在將其排除於 `rds.allowed_extensions` 參數清單外之前已經安裝擴充功能，則擴充功能仍然可以正常使用，而且 `ALTER EXTENSION` 和 `DROP EXTENSION` 等命令將繼續運作。但是，在限制擴充功能之後，限制擴充功能的 `CREATE EXTENSION` 命令將會失敗。

`CREATE EXTENSION CASCADE` 擴充功能相依性的安裝也會受到限制。必須在 `rds.allowed_extensions` 中指定擴充功能及其相依性。如果擴充功能相依性安裝失敗，整個 `CREATE EXTENSION CASCADE` 陳述式將會失敗。

如果 `rds.allowed_extensions` 參數中未包含擴充功能，您會在嘗試安裝擴充功能時看到下列錯誤。

```
ERROR: permission denied to create extension "extension-name"
HINT: This extension is not specified in "rds.allowed_extensions".
```

PostgreSQL 可信任延伸

若要安裝大多數 PostgreSQL 延伸，需要 `rds_superuser` 權限。PostgreSQL 13 推出了可信任擴充功能，這減少了向普通使用者授予 `rds_superuser` 權限的需求。使用此功能，使用者可以安裝許多延伸 (如果他們具有目前資料庫的 `CREATE` 權限而不是 `rds_superuser` 角色)。如需詳細資訊，請參閱 PostgreSQL 文件中的 SQL [CREATE EXTENSION](#) 命令。

下面列出了具有目前資料庫 `CREATE` 權限但不需要 `rds_superuser` 角色的使用者可以安裝的延伸：

- [bool_plperl](#)
- [btree_gin](#)
- [btree_gist](#)

- [citext](#)
- [cube](#)
- [dict_int](#)
- [fuzzystrmatch](#)
- [hstore](#)
- [intarray](#)
- [isn](#)
- [jsonb_plperl](#)
- [ltree](#)
- [pg_trgm](#)
- [pgcrypto](#)
- [plperl](#)
- [plpgsql](#)
- [pltcl](#)
- [tablefunc](#)
- [tsm_system_rows](#)
- [tsm_system_time](#)
- [unaccent](#)
- [uuid-osspl](#)

如需有關 Amazon RDS 支援的 PostgreSQL 擴充功能的詳細清單，請參閱 Amazon RDS for PostgreSQL 版本備註中的 [Amazon RDS 上支援的 PostgreSQL 擴充功能](#)。

使用 Amazon RDS for PostgreSQL 支援的 PostgreSQL 功能

Amazon RDS for PostgreSQL 支援許多最常用的 PostgreSQL 功能。例如，PostgreSQL 的自動資料清理功能可對資料庫執行例行維護。根據預設，會使用自動真空功能。雖然您可以關閉此功能，但我們強烈建議保持開啟。了解此功能以及您可以做些什麼來確保它能夠正常工作是任何 DBA 的基本任務。如需有關自動資料清理的詳細資訊，請參閱 [在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理](#)。若要進一步了解其他常見 DBA 任務，請參閱 [Amazon RDS for PostgreSQL 的常用 DBA 任務](#)。

RDS for PostgreSQL 也支援為資料庫執行個體新增重要功能的擴充功能。例如，您可以使用 PostGIS 擴充功能來處理空間資料，或者使用 pg_cron 擴充功能在執行個體內進行維護排程。如需 PostgreSQL 擴充功能的詳細資訊，請參閱 [搭配 Amazon RDS for PostgreSQL 使用 PostgreSQL 擴充功能](#)。

外部資料包裝函式是一種特定類型的擴充功能，旨在讓您的 RDS for PostgreSQL 資料庫執行個體與其他商業資料庫或資料類型搭配使用。如需 RDS PostgreSQL 支援的外部資料包裝函式的詳細資訊，請參閱 [使用 Amazon RDS for PostgreSQL 支援的外部資料包裝函式](#)。

在下文中，您可以了解 RDS for PostgreSQL 支援其他功能的資訊。

主題

- [使用 RDS for PostgreSQL 的自訂資料類型和列舉](#)
- [RDS for PostgreSQL 適用的事件觸發程序](#)
- [RDS for PostgreSQL 的巨型分頁](#)
- [為 Amazon RDS for PostgreSQL 執行邏輯複寫](#)
- [stats_temp_directory 的 RAM 磁碟](#)
- [RDS for PostgreSQL 的資料表空間](#)
- [適用於 EBCDIC 和其他大型機遷移的 RDS for PostgreSQL 定序](#)

使用 RDS for PostgreSQL 的自訂資料類型和列舉

PostgreSQL 支援建立自訂資料類型和使用列舉。如需如何建立並使用列舉及其他資料類型的詳細資訊，請參閱 PostgreSQL 文件中的 [列舉類型](#)。

以下範例示範將類型建立為列舉，然後將值插入到資料表。

```
CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue', 'purple');
```

```
CREATE TYPE
CREATE TABLE t1 (colors rainbow);
CREATE TABLE
INSERT INTO t1 VALUES ('red'), ( 'orange');
INSERT 0 2
SELECT * from t1;
colors
-----
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
-----
crimson
orange
(2 rows)
```

RDS for PostgreSQL 適用的事件觸發程序

所有目前的 PostgreSQL 版本都支援事件觸發程序，所有 RDS for PostgreSQL 版本也支援事件觸發程序。您可以使用主要使用者帳戶 (預設 postgres) 來建立、修改、重新命名和刪除事件觸發程序。事件觸發是在資料庫執行個體層級上運作，可套用至執行個體上的所有資料庫。

例如，下列程式碼建立的事件觸發程序會在每個資料定義語言 (DDL) 命令結束時列印目前使用者。

```
CREATE OR REPLACE FUNCTION raise_notice_func()
    RETURNS event_trigger
    LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
    ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

如需 PostgreSQL 事件觸發的詳細資訊，請參閱 PostgreSQL 文件中的[事件觸發](#)。

在 Amazon RDS 上使用 PostgreSQL 事件觸發時有幾項限制。所需資訊包括下列項目：

- 您無法在僅供讀取複本上建立事件觸發程序。不過，您可以在僅供讀取複本來源建立事件觸發。然後，事件觸發會複製到僅供讀取複本。從來源推送變更時，僅供讀取複本上的事件觸發不會在僅供讀取複本上發動。不過，如果提升僅供讀取複本，則資料庫操作發生時，現有的事件觸發會啟動。
- 若要對使用事件觸發程序的 PostgreSQL 資料庫執行個體執行主要版本升級，就必須在升級執行個體之前刪除事件觸發程序。

RDS for PostgreSQL 的巨型分頁

巨型分頁是一項記憶體管理功能，可減少資料庫執行個體處理大型連續記憶體區塊 (如共用緩衝區使用的記憶體區塊) 時的額外負荷。所有目前可用的 RDS for PostgreSQL 版本都支援此 PostgreSQL 功能。您可以呼叫 `mmap` 或 `SYSV` 共用記憶體，以分配巨型分頁給您的應用程式。RDS for PostgreSQL 同時支援 4 KB 和 2 MB 分頁大小。

您可以開啟或關閉巨型分頁功能，方法是變更 `huge_pages` 參數的值。除了微型、小型和中型資料庫執行個體類之外，所有資料庫執行個體類都預設開啟該功能。

RDS for PostgreSQL 會根據可用的共用記憶體使用巨型分頁。如果資料庫執行個體因為共用記憶體限制而無法使用巨型分頁，Amazon RDS 會阻止資料庫執行個體啟動。在此情況下，Amazon RDS 會將資料庫執行個體的狀態設為不相容的參數狀態。如果發生這種情況，您可以將 `huge_pages` 參數設為「off」，以允許 Amazon RDS 啟動資料庫執行個體。

`shared_buffers` 參數是為了使用巨型分頁而設定所需之共用記憶體集區的關鍵。`shared_buffers` 參數的預設值會使用資料庫參數巨集。此巨集會設定可用於資料庫執行個體記憶體的總計 8 KB 分頁的百分比。當您使用巨型分頁時，這些頁面會與巨型分頁放在一起。如果共用記憶體參數設為需要 90% 以上的資料庫執行個體記憶體，Amazon RDS 會將資料庫執行個體設為不相容的參數狀態。

若要進一步了解 PostgreSQL 記憶體管理，請參閱 PostgreSQL 說明文件中的[資源耗用](#)。

為 Amazon RDS for PostgreSQL 執行邏輯複寫

從 10.4 版開始，RDS for PostgreSQL 支援 PostgreSQL 10 中引進的發佈與訂閱 SQL 語法。如需進一步了解，請參閱 PostgreSQL 文件中的[邏輯複寫](#)。

Note

除了在 PostgreSQL 10 中引進的原生 PostgreSQL 邏輯複寫功能之外，RDS for PostgreSQL 也支援 `pglogical` 延伸模組。如需更多詳細資訊，請參閱 [使用 pglogical 跨執行個體同步資料](#)。

在下文中，您可以了解如何設定 RDS for PostgreSQL 資料庫執行個體邏輯複寫的資訊。

主題

- [了解邏輯複寫和邏輯解碼](#)
- [使用邏輯複寫槽](#)

了解邏輯複寫和邏輯解碼

RDS for PostgreSQL 支援使用 PostgreSQL 的邏輯複寫槽來串流預寫日誌 (WAL) 變更。它也支援使用邏輯解碼。您可以在執行個體上設定邏輯複寫槽，然後透過這些槽將資料庫變更串流至用戶端，例如 `pg_recvlogical`。邏輯複寫插槽在資料庫層級建立，且支援對單一資料庫的多個複寫連線。

PostgreSQL 邏輯複寫最常見的用戶端是 Amazon EC2 執行個體上的 AWS Database Migration Service 或自訂管理的主機。邏輯複寫插槽沒有關於串流接收者的資訊。此外，不要求目標必須是複本資料庫。如果您設定邏輯複寫槽，但未讀取這個槽，則資料會寫入並快速填滿資料庫執行個體的儲存體。

Amazon RDS 的 PostgreSQL 邏輯複寫與邏輯解碼是以參數、複寫連線類型和安全角色來開啟。任何用戶端只要能夠在 PostgreSQL 資料庫執行個體上的資料庫建立複寫連線，都可以當作邏輯解碼的用戶端。

開啟 RDS for PostgreSQL 資料庫執行個體的邏輯解碼

1. 確保您正在使用的使用者帳戶具有以下角色：
 - 可讓您開啟邏輯複寫的 `rds_superuser` 角色
 - 授權來管理邏輯槽和利用邏輯槽來串流資料的 `rds_replication` 角色
2. 將 `rds.logical_replication` 靜態參數設為 1。套用此參數時，也會設定 `wal_level`、`max_wal_senders`、`max_replication_slots` 和 `max_connections` 參數。這些參數變更會產生更多 WAL，因此請僅在使用邏輯槽時才設定 `rds.logical_replication` 參數。

3. 為了讓靜態 `rds.logical_replication` 參數生效，請重新啟動資料庫執行個體。
4. 建立邏輯複寫槽將在下一章節中說明。此程序要求您指定解碼外掛程式。目前，RDS for PostgreSQL 支援隨附於 PostgreSQL 的 `test_decoding` 和 `wal2json` 輸出外掛程式。

如需有關 PostgreSQL 邏輯解碼的詳細資訊，請參閱 [PostgreSQL 文件](#)。

使用邏輯複寫槽

您可以利用 SQL 命令來使用邏輯槽。例如，下列命令使用預設 PostgreSQL 輸出外掛程式 `test_slot`，建立一個名為 `test_decoding` 的邏輯槽。

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name      | xlog_position
-----+-----
regression_slot | 0/16B1970
(1 row)
```

若要列出邏輯槽，請使用下列命令。

```
SELECT * FROM pg_replication_slots;
```

若要捨棄邏輯槽，請使用下列命令。

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
-----
(1 row)
```

如需有關使用邏輯複寫槽的其他範例，請參閱 PostgreSQL 文件中的 [邏輯解碼範例](#)。

建立邏輯複寫槽之後，即可開始串流。下列範例顯示如何透過串流複寫協定來控制邏輯解碼。此範例會使用包含在 PostgreSQL 發行版中的 `pg_recvlogical` 程式。這會需要將用戶端身分驗證設為允許複寫連線。

```
pg_recvlogical -d postgres --slot test_slot -U postgres
--host -instance-name.111122223333.aws-region.rds.amazonaws.com
-f - --start
```

若要查看檢視 `pg_replication_origin_status` 的內容，請查詢 `pg_show_replication_origin_status` 函式。

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
-----+-----+-----+-----
(0 rows)
```

stats_temp_directory 的 RAM 磁碟

您可以使用 RDS for PostgreSQL 參數 `rds.pg_stat_ramdisk_size` 指定組態給 RAM 磁碟的系統記憶體，以存放 PostgreSQL `stats_temp_directory`。RAM 磁碟參數適用於 Amazon RDS 上的所有 PostgreSQL 版本。

在某些工作負載下，設定此參數可以提高效能和降低輸入/輸出需求。如需 `stats_temp_directory` 的詳細資訊，請參閱 [PostgreSQL 文件](#)。

如要設定 `stats_temp_directory` 的 RAM 磁碟，請於資料庫執行個體所使用的參數群組中，將 `rds.pg_stat_ramdisk_size` 參數設為整數常值。此參數表示 MB，因此您必須使用整數值。表達式、公式和函數對 `rds.pg_stat_ramdisk_size` 參數無效。請務必重新啟動資料庫執行個體，變更才會生效。如需有關設定參數的詳細資訊，請參閱 [使用參數群組](#)。

例如，下列 AWS CLI 命令將 RAM 磁碟參數設為 256 MB。

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name pg-95-ramdisk-testing \
  --parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
  ApplyMethod=pending-reboot"
```

重新啟動之後，執行以下命令來查看 `stats_temp_directory` 的狀態。

```
postgres=> SHOW stats_temp_directory;
```

此命令應該會傳回下列結果。

```
stats_temp_directory
-----
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

RDS for PostgreSQL 的資料表空間

RDS for PostgreSQL 支援資料表空間以實現相容性。因為所有儲存區都位於單一邏輯磁碟區上，所以您無法使用資料表空間進行輸入/輸出分割或隔離。我們的基準和經驗指出，單一邏輯磁碟區是多數使用案例的最佳設定。

若要建立資料表空間並搭配 RDS for PostgreSQL 資料庫執行個體一起使用，需要 `rds_superuser` 角色。RDS for PostgreSQL 資料庫執行個體的主使用者帳戶 (預設名稱 `postgres`) 是此角色的成員。如需更多詳細資訊，請參閱 [了解 PostgreSQL 角色和許可](#)。

如果您在建立資料表空間時指定檔案名稱，路徑前綴為 `/rdsdbdata/db/base/tablespace`。以下範例會將資料表空間檔案置於 `/rdsdbdata/db/base/tablespace/data` 中。此範例假設 `dbadmin` 使用者 (角色) 存在，且已獲派處理資料表空間所需的 `rds_superuser` 角色。

```
postgres=> CREATE TABLESPACE act_data
  OWNER dbadmin
  LOCATION '/data';
CREATE TABLESPACE
```

若要進一步了解 PostgreSQL 資料表空間，請參閱 PostgreSQL 說明文件中的 [資料表空間](#)。

適用於 EBCDIC 和其他大型機遷移的 RDS for PostgreSQL 定序

RDS for PostgreSQL 10 版及更高版本包括基於 Unicode 10.0 的 ICU 版本 60.2，其中包含來自 Unicode 通用語言環境資料儲存庫 (CLDR 32) 的定序。這些軟體國際化程式庫可確保字元編碼以一致的方式呈現，不論作業系統或平台為何。如需 Unicode CLDR-32 的詳細資訊，請參閱 Unicode CLDR 網站上的 [CLDR 32 版本備註](#)。您可以在 [ICU Technical Committee \(ICU-TC\)](#) 網站了解有關 Unicode (ICU) 的國際化元件的更多資訊。如需有關 ICU-60 的資訊，請參閱 [下載 ICU 60](#)。

從 14.3 版開始，RDS for PostgreSQL 也包含有助於從 EBCDIC 系統進行資料整合和轉換的定序。擴展的二進制編碼十進制交換碼或 EBCDIC 編碼通常由大型機作業系統使用。這些 Amazon RDS 提供的定序定義很小，只能夠直接對應至 EBCDIC 字碼頁的 Unicode 字元。這些字元按 EBCDIC 代碼點順序進行排序，以便在轉換後進行資料驗證。這些定序不包含部正常的格式，也不包含未直接對應至來源 EBCDIC 字碼頁的 Unicode 字元。

EBCDIC 字碼頁與 Unicode 字碼點之間的字元對應是以 IBM 發佈的表格為基礎。完整的設定可從 IBM 的 [壓縮檔案](#) 下載。RDS for PostgreSQL 會將這些對應與 ICU 提供的工具搭配使用，以建立本節中表格中列出的定序。定序名稱包括 ICU 要求的語言和國家/地區。但是，EBCDIC 字碼頁不會指定語言，有些 EBCDIC 字碼頁涵蓋多個國家/地區。這表示資料表中定序名稱的語言和國家/地區部分是任意的，而

且不需要與目前的地區設定相符。換句話說，字碼頁編號是此表格中定序名稱最重要的部分。您可以在任何 RDS for PostgreSQL 資料庫中使用下列表格中列出的任何定序。

- [Unicode to EBCDIC collations table](#) - 部分大型主機資料移轉工具在內部使用 LATIN1 或 LATIN9 來編碼及處理資料。這類工具使用往返結構描述來保留資料完整性並支援反向轉換。此資料表中的定序可以由使用 LATIN1 編碼來處理資料的工具使用，這不需要特殊處理。
- [Unicode to LATIN9 collations table](#) - 您可以在任何 RDS for PostgreSQL 資料庫中使用這些定序。

在下表中，您會發現 RDS 版 PostgreSQL 中可用的定序，這些定序會將 EBCDIC 字碼頁對應至 Unicode 字碼點。我們建議您使用此表格中的定序來進行應用程式開發，這些應用程式開發需要根據 IBM 字碼頁的順序進行排序。

PostgreSQL 定序名稱	字碼頁對應和排序順序的說明
da-DK-cp277-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 277 (每個轉換表) 按照 IMB CP 277 字碼點順序排序
de-DE-cp273-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 273 (每個轉換表) 按照 IMB CP 273 字碼點順序排序
en-GB-cp285-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 285 (每個轉換表) 按照 IMB CP 285 字碼點順序排序
en-US-cp037-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 037 (每個轉換表) 按照 IMB CP 37 字碼點順序排序
es-ES-cp284-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 284 (每個轉換表) 按照 IMB CP 284 字碼點順序排序
fi-FI-cp278-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 278 (每個轉換表) 按照 IMB CP 278 字碼點順序排序

PostgreSQL 定序名稱	字碼頁對應和排序順序的說明
fr-FR-cp297-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 297 (每個轉換表) 按照 IMB CP 297 字碼點順序排序
it-IT-cp280-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 280 (每個轉換表) 按照 IMB CP 280 字碼點順序排序
nl-BE-cp500-x-icu	Unicode 字元直接對應至 IBM EBCDIC 字碼頁 500 (每個轉換表) 按照 IMB CP 500 字碼點順序排序

Amazon RDS 提供一組額外的定序，可根據來源資料的 EBCDIC 字碼頁，使用 IBM 發佈的表格，依原始程式碼點的順序，對應至 LATIN9 字元的 Unicode 程式碼點進行排序。

PostgreSQL 定序名稱	字碼頁對應排序順序的說明
da-DK-cp1142m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1142 (每個轉換表) 按照 IMB CP 1142 字碼點順序排序
de-DE-cp1141m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1141 (每個轉換表) 按照 IMB CP 1141 字碼點順序排序
en-GB-cp1146m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1146 (每個轉換表) 按照 IMB CP 1146 字碼點順序排序
en-US-cp1140m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1140 (每個轉換表) 按照 IMB CP 1140 字碼點順序排序

PostgreSQL 定序名稱	字碼頁對應排序順序的說明
es-ES-cp1145m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1145 (每個轉換表) 按照 IBM CP 1145 字碼點順序排序
fi-FI-cp1143m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1143 (每個轉換表) 按照 IBM CP 1143 字碼點順序排序
fr-FR-cp1147m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1147 (每個轉換表) 按照 IBM CP 1147 字碼點順序排序
it-IT-cp1144m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1144 (每個轉換表) 按照 IBM CP 1144 字碼點順序排序
nl-BE-cp1148m-x-icu	Unicode 字元對應至 LATIN9 字元，原始轉換自 IBM EBCDIC 字碼頁 1148 (每個轉換表) 按照 IBM CP 1148 字碼點順序排序

在下列各項中，您可以找到使用 RDS for PostgreSQL 定序的範例。

```
db1=> SELECT pg_import_system_collations('pg_catalog');
pg_import_system_collations
-----
                                36
db1=> SELECT 'a' < 'a' coll1;
coll1
-----
t
db1=> SELECT 'a' < 'a' COLLATE "da-DK-cp277-x-icu" coll1;
coll1
-----
f
```

我們建議您使用 [Unicode to EBCDIC collations table](#) 中的定序並在 [Unicode to LATIN9 collations table](#) 中用於需要根據 IBM 字碼頁排序進行排序的應用程式開發。下列定序 (後綴為字母「b」) 也可以在

pg_collation 中看見，但適用於以下位置的大型主機資料整合和移轉工具，AWS 對應具有特定字碼點位移的字碼頁，並且需要在定序中進行特殊處理。換言之，不建議使用下列定序。

- da-DK-277b-x-icu
- da-DK-1142b-x-icu
- de-DE-cp273b-x-icu
- de-DE-cp1141b-x-icu
- en-GB-cp1146b-x-icu
- en-GB-cp285b-x-icu
- en-US-cp037b-x-icu
- en-US-cp1140b-x-icu
- es-ES-cp1145b-x-icu
- es-ES-cp284b-x-icu
- fi-FI-cp1143b-x-icu
- fr-FR-cp1147b-x-icu
- fr-FR-cp297b-x-icu
- it-IT-cp1144b-x-icu
- it-IT-cp280b-x-icu
- nl-BE-cp1148b-x-icu
- nl-BE-cp500b-x-icu

若要深入了解如何將應用程式從大型主機環境移轉至 AWS，請參閱[什麼是 AWS 大型機現代化？](#)。

若要進一步了解管理 PostgreSQL 中的定序，請參閱 PostgreSQL 文件中的[定序支援](#)。

連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體

在 Amazon RDS 佈建資料庫執行個體之後，您就可以使用任何標準 SQL 用戶端應用程式來連線至執行個體。連線到資料庫執行個體之前，資料庫執行個體必須可用且可存取。您是否可以從 VPC 外部連線到執行個體，取決於您建立 Amazon RDS 資料庫執行個體的方式：

- 如果您將資料庫執行個體建立為公有，VPC 外部的裝置和 Amazon EC2 執行個體都可以連線到您的資料庫。
- 如果您將資料庫執行個體建立為私有，只有 Amazon EC2 執行個體和 Amazon VPC 內的裝置可以連線到您的資料庫。

若要檢查您的資料庫執行個體是公用還是私有，請使 AWS Management Console 用檢視執行個體的 [連線與安全性] 索引標籤。在 Security (安全性) 底下會顯示 "Publicly accessible" (可公開存取) 值，其中為 No (否) 表示私有，Yes (是) 表示公有。

若要進一步了解不同的 Amazon RDS 和 Amazon VPC 組態，以及這些組態如何影響可存取性，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

內容

- [安裝 psql 客戶端](#)
- [尋找 RDS 資料庫執行個體的 PostgreSQL 資訊](#)
- [使用 pgAdmin 連線至 RDS for PostgreSQL 資料庫執行個體](#)
- [使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體](#)
- [使用 Amazon Web Services \(AWS\) JDBC 驅動程式連線至 RDS 版](#)
- [使用 Amazon Web Services \(AWS\) Python 驅動程式連接到 RDS 版](#)
- [針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解](#)
 - [錯誤 – 嚴重：資料庫名稱不存在](#)
 - [錯誤 – 無法連線至伺服器：連線逾時](#)
 - [安全性群組存取規則發生錯誤](#)

安裝 psql 客戶端

若要從 EC2 執行個體連線到資料庫執行個體，您可以在 EC2 執行個體上安裝 PostgreSQL 用戶端。若要在 Amazon Linux 2023 上安裝 psql 用戶端，請執行下列命令：

```
sudo dnf install postgresql15
```

若要在 Amazon Linux 2 上安裝 psql 用戶端，請執行下列命令：

```
sudo amazon-linux-extras install postgresql14
```

若要在 Ubuntu 上安裝 psql 用戶端，請執行下列命令：

```
sudo apt-get install -y postgresql14
```

尋找 RDS 資料庫執行個體的 PostgreSQL 資訊

如果資料庫執行個體可用且可存取，您可以透過提供下列資訊給 SQL 用戶端應用程式來連線：

- 資料庫執行個體端點，做為執行個體的主機名稱 (DNS 名稱)。
- 資料庫執行個體接聽所在的連接埠。PostgreSQL 的預設連接埠為 5432。
- 資料庫執行個體的使用者名稱和密碼。PostgreSQL 的預設「主要使用者名稱」是 postgres。
- 資料庫的名稱和密碼 (資料庫名稱)。

您可以使用 AWS Management Console、AWS CLI [describe-db-instances](#) 命令或 Amazon RDS API [DescribeDBInstances](#) 操作來取得這些詳細資訊。

若要尋找端點、連接埠號碼和資料庫名稱，請使用 AWS Management Console

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 開啟 RDS 主控台，然後選擇 Databases (資料庫)，以顯示資料庫執行個體清單。
3. 選擇 PostgreSQL 資料庫執行個體名稱以顯示其詳細資訊。
4. 在 Connectivity & security (連線能力和安全性) 索引標籤上，複製該端點。另外，請記下連接埠號碼。您需要同時有端點和連接埠號碼，才能連線至資料庫執行個體。

RDS > Databases > database-test1

database-test1

Summary

DB identifier database-test1	CPU 5.82%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port	Networking
Endpoint database-test1.123456789012.us-east-1.rds.amazonaws.com	Availability Zone us-east-1c
Port 5432	VPC vpc-
	Subnet group default

5. 在 Configuration (組態) 索引標籤上，記下資料庫名稱。如果您在建立 RDS for PostgreSQL 執行個體時建立資料庫，您會看到名稱列在資料庫名稱底下。如果您沒有建立資料庫，資料庫名稱會顯示破折號 (-)。

Connectivity & security	Monitoring	Logs & events	Configuration
Instance			
Configuration			
DB instance ID	database-test1	Engine version	14.6
DB name	labdb		

以下是連線至 PostgreSQL 資料庫執行個體的兩個方式。第一個範例使用 pgAdmin，這是常見的 PostgreSQL 開放式原始碼管理和開發工具。第二個範例使用 psql，這是屬於 PostgreSQL 安裝一部分的命令列公用程式。

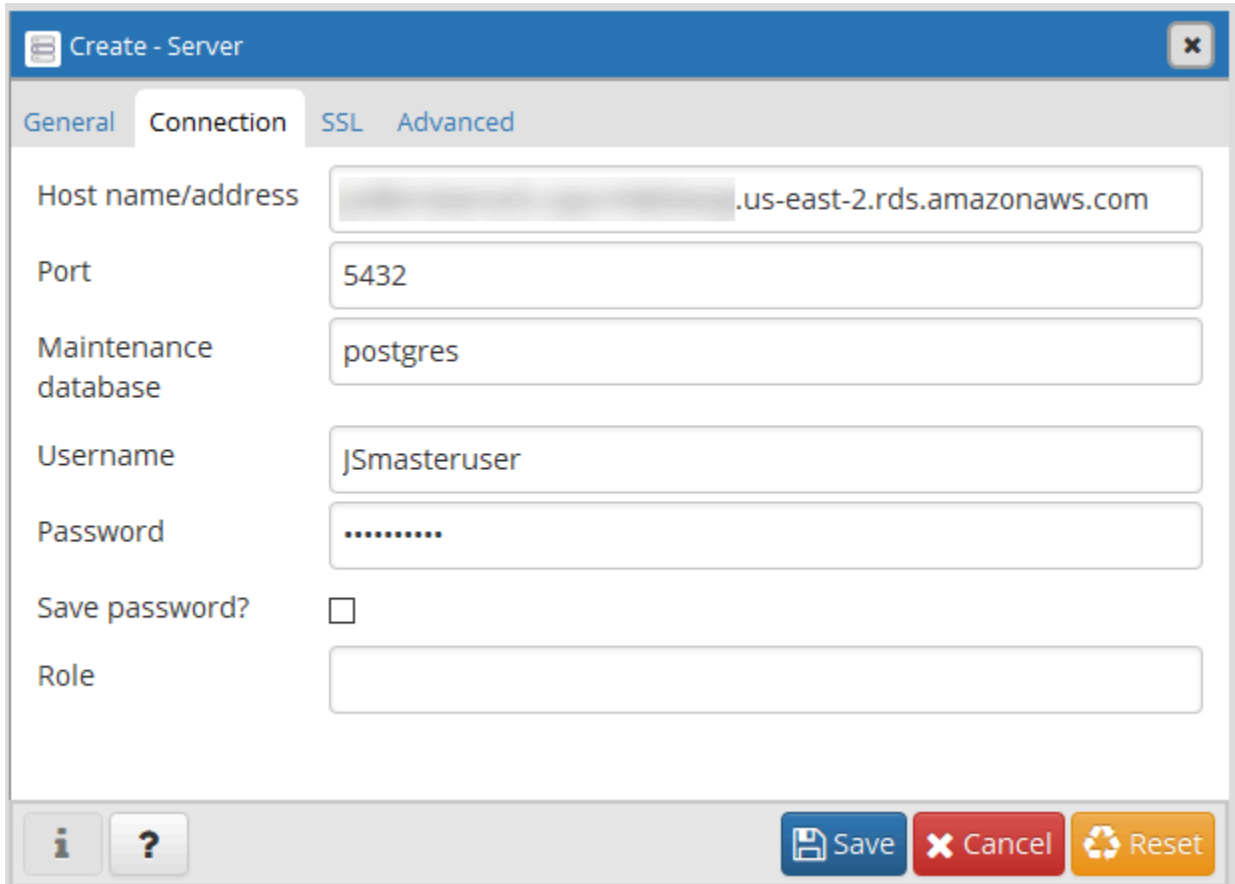
使用 pgAdmin 連線至 RDS for PostgreSQL 資料庫執行個體

您可以使用開放原始碼工具 pgAdmin 來連線至 RDS for PostgreSQL 資料庫執行個體。您可以從 <http://www.pgadmin.org/> 下載及安裝 pgAdmin，不需要在您的用戶端電腦上有 PostgreSQL 的本機執行個體。

若要使用 pgAdmin 連線至 RDS for PostgreSQL 資料庫執行個體

1. 在您的用戶端電腦上啟動 pgAdmin 應用程式。
2. 在 Dashboard (儀表板) 標籤上，選擇 Add New Server (新增伺服器)。
3. 在 Create - Server (建立 - 伺服器) 對話方塊中，於 General (一般) 標籤上輸入名稱，以識別 pgAdmin 中的伺服器。
4. 在 Connection (連線) 標籤上，輸入來自資料庫執行個體的下列資訊：
 - 針對 Host (主機)，輸入端點，例如 `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`。

- 針對 Port (連接埠)，輸入指派的連接埠。
- 針對 Username (使用者名稱)，輸入您在建立資料庫執行個體時輸入的使用者名稱 (如果從預設值 postgres 變更「主要使用者名稱」)。
- 針對 Password (密碼)，輸入建立資料庫執行個體時所輸入的密碼。



The screenshot shows a 'Create - Server' dialog box with the 'Connection' tab selected. The fields are as follows:

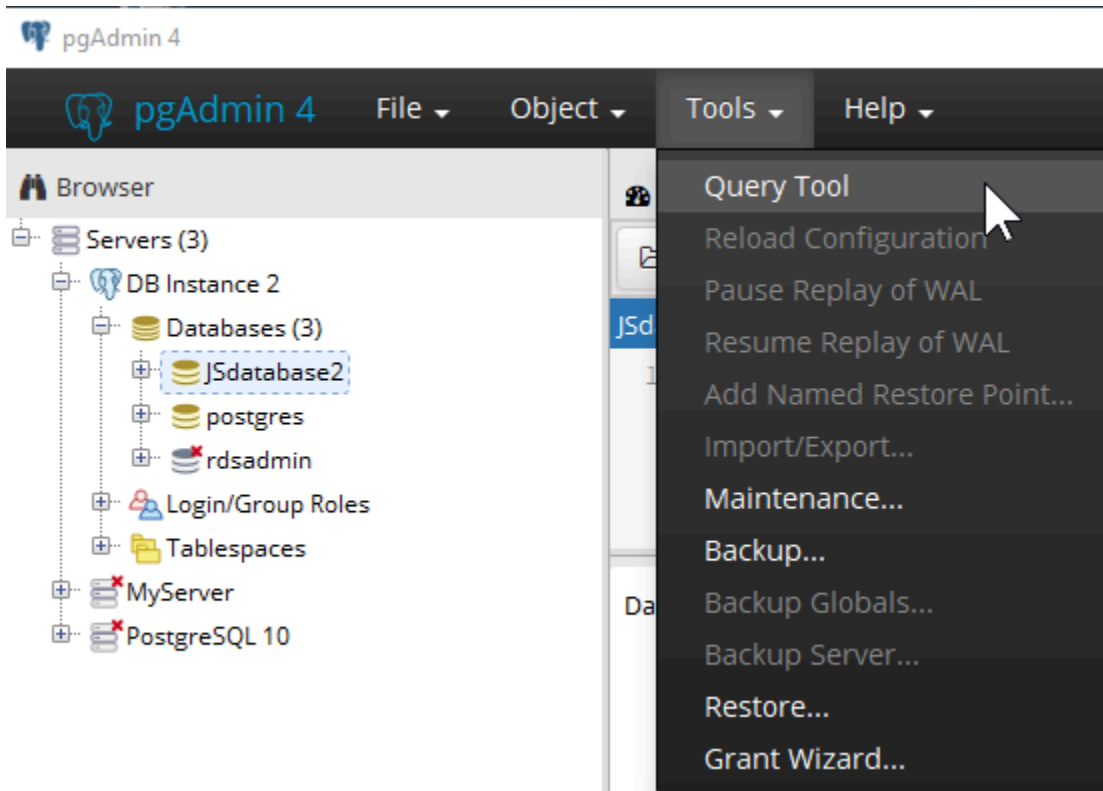
Field	Value
Host name/addressus-east-2.rds.amazonaws.com
Port	5432
Maintenance database	postgres
Username	jSmasteruser
Password
Save password?	<input type="checkbox"/>
Role	

Buttons at the bottom: Save (blue), Cancel (red), Reset (yellow).

5. 選擇 Save (儲存)。

如果有連線問題，請參閱 [針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解](#)。

6. 若要在 pgAdmin 瀏覽器中存取資料庫，請展開 Servers (伺服器)、資料庫執行個體，和 Databases (資料庫)。選擇資料庫執行個體的資料庫名稱。



7. 若要開啟您可以在其中輸入 SQL 命令的面板，請選擇 Tools (工具) 然後選擇 Query Tool (查詢工具)。

使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體

您可以使用 psql 命令列公用程式的本機執行個體來連線至 RDS for PostgreSQL 資料庫執行個體。您需要在用戶端電腦上安裝 PostgreSQL 或 psql 用戶端。

您可以從 [PostgreSQL](#) 網站下載 PostgreSQL 用戶端。請遵循適用於您作業系統版本的指示，以安裝 psql。

若要使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體，您必須提供主機 (DNS) 資訊、存取憑證和資料庫的名稱。

請使用下列其中一個格式來連線至 RDS for PostgreSQL 資料庫執行個體。連線時，系統會提示您輸入密碼。如需批次工作或指令碼，請使用 `--no-password` 選項。此選項是為整個工作階段設定。

Note

當伺服器需要密碼驗證，且無法從其他來源取得密碼時，使用 `--no-password` 的連線嘗試會失敗。如需詳細資訊，請參閱 [psql 文件](#)。

如果這是您第一次連線至此資料庫執行個體，或者尚未針對此 RDS for PostgreSQL 執行個體建立資料庫，則可以使用「主要使用者名稱」和密碼連線至 postgres 資料庫。

針對 Unix，使用下列格式。

```
psql \  
  --host=<DB instance endpoint> \  
  --port=<port> \  
  --username=<master username> \  
  --password \  
  --dbname=<database name>
```

針對 Windows，使用下列格式。

```
psql ^  
  --host=<DB instance endpoint> ^  
  --port=<port> ^  
  --username=<master username> ^  
  --password ^  
  --dbname=<database name>
```

例如，下列命令會在名為 mypgdb 的 PostgreSQL 資料庫執行個體上使用虛擬登入資料連線至名為 mypostgres1 的資料庫。

```
psql --host=mypostgres1.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --  
username=awsuser --password --dbname=mypgdb
```

使用 Amazon Web Services (AWS) JDBC 驅動程式連線至 RDS 版

Amazon Web Services (AWS) JDBC 驅動程序被設計為一個高級 JDBC 包裝。這個包裝是補充和擴展現有 JDBC 驅動程序的功能。該驅動程序與社區 pgJDBC 驅動程序兼容。

若要安裝 AWS JDBC 驅動程式，請附加 AWS JDBC 驅動程式 .jar 檔案 (位於應用程式中 CLASSPATH)，並保留對個別社群驅動程式的參考。更新相應的連接 URL 前綴，如下所示：

- `jdbc:postgresql://` 設定為 `jdbc:aws-wrapper:postgresql://`

如需有關 AWS JDBC 驅動程式的詳細資訊以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS\) JDBC 驅動程式 GitHub 儲存庫](#)。

使用 Amazon Web Services (AWS) Python 驅動程式連接到 RDS 版

Amazon Web Services (AWS) Python 驅動程序被設計為一個先進的 Python 包裝。此包裝器與開源 Psycopg 驅動程序的功能互補並擴展。AWS Python 驅動程式支援 3.8 及更高版本。您可以使用 pip 命令以及 psycopg 開放原始碼套件來安裝套件。aws-advanced-python-wrapper

有關 AWS Python 驅動程序的更多信息以及使用它的完整說明，請參閱 [Amazon Web Services \(AWS \) Python 驅動程序 GitHub 儲存庫](#)。

針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解

主題

- [錯誤 – 嚴重：資料庫名稱不存在](#)
- [錯誤 – 無法連線至伺服器：連線逾時](#)
- [安全性群組存取規則發生錯誤](#)

錯誤 – 嚴重：資料庫##不存在

如果您在連線時收到類似 FATAL: database *name* does not exist 的錯誤，請在 `--dbname` 選項中嘗試使用預設資料庫名稱 postgres。

錯誤 – 無法連線至伺服器：連線逾時

如果您無法連線至資料庫執行個體，常見的錯誤是 `Could not connect to server: Connection timed out.`，如果遇到此錯誤，請檢查以下事項：

- 檢查使用的主機名稱是否是資料庫執行個體端點，且使用的連接埠號碼是否正確。
- 確定該資料庫執行個體的公開存取性是設為 Yes (是) 以允許外部連線。若要修改 Public access (公開存取) 設定，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
- 確定連線到資料庫的使用者對它具有 CONNECT 存取權。您可以使用下列查詢來提供資料庫的連線存取權。


```
GRANT CONNECT ON DATABASE database name TO username;
```

- 檢查指派給資料庫執行個體的安全群組是否具有以下規則：允許透過您的連線可能行經之防火牆的存取。例如，資料庫執行個體是使用預設連接埠 5432 來建立，而公司的防火牆規則會封鎖從外部公司裝置到該連接埠的連線。

若要修復此情況，請將資料庫執行個體修改為使用不同的連接埠。同時，確定套用至資料庫執行個體的安全群組，可允許對該新連接埠的連線。若要修改 Database port (資料庫連接埠) 設定，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

- 另請參閱 [安全性群組存取規則發生錯誤](#)。

安全性群組存取規則發生錯誤

顯然地，最常見的連線問題在於對資料庫執行個體指派之安全群組的存取規則。如果建立資料庫執行個體時使用的是預設的安全群組，安全群組很可能沒有可允許您存取該執行個體的存取規則。

為了讓連線運作，您在建立時指派給資料庫執行個體的安全群組，必須允許對該資料庫執行個體的存取。例如，如果資料庫執行個體是在 VPC 中建立，則它必須具有可授權連線的 VPC 安全群組。檢查建立資料庫執行個體所用的安全群組，是否未授權從執行應用程式所在的裝置或 Amazon EC2 執行個體的連線。

您可以新增或編輯安全群組中的傳入規則。針對 Source (來源)，選擇 My IP (我的 IP) 允許透過您的瀏覽器中偵測到的 IP 位址存取資料庫執行個體。如需更多詳細資訊，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

或者，如果資料庫執行個體是在 VPC 之外建立，則它必須具有可授權那些連線的安全群組。

如需 Amazon RDS 安全群組的詳細資訊，請參閱 [使用安全群組控制存取](#)。

使用 SSL/TLS 保護連接至 RDS for PostgreSQL 的連線

RDS for PostgreSQL 支援 PostgreSQL 資料庫執行個體的 Secure Sockets Layer (SSL) 加密。您可以使用 SSL 將應用程式與 PostgreSQL 資料庫執行個體之間的 PostgreSQL 連線加密。您也可以強制 PostgreSQL 資料庫執行個體的所有連線皆使用 SSL。RDS PostgreSQL 也支援 Transport Layer Security (TLS)，也即 SSL 的後繼通訊協定。

若要進一步了解 Amazon RDS 和資料保護 (包括使用 SSL/TLS 加密連線)，請參閱 [Amazon RDS 中的資料保護](#)。

主題

- [將 SSL 與 PostgreSQL 資料庫執行個體搭配使用](#)
- [更新應用程式使用新的 SSL/TLS 憑證來連線至 PostgreSQL 資料庫執行個體](#)

將 SSL 與 PostgreSQL 資料庫執行個體搭配使用

對於 PostgreSQL 資料庫執行個體，Amazon RDS 支援 Secure Sockets Layer (SSL) 加密。您可以使用 SSL 將應用程式與 PostgreSQL 資料庫執行個體之間的 PostgreSQL 連線加密。依預設，RDS for PostgreSQL 使用並期望所有用戶端皆使用 SSL/TLS 進行連線，但您也可加以要求。RDS 版支 PostgreSQL 輸層安全性 (TLS) 1.1、1.2 和 1.3 版。

如需 SSL 支援和 PostgreSQL 資料庫的一般資訊，請參閱 PostgreSQL 文件中的 [SSL 支援](#)。如需透過 JDBC 使用 SSL 連線的資訊，請參閱 PostgreSQL 文件中的 [設定用戶端](#)。

所有 AWS 區域均提供 PostgreSQL。建立執行個體時，Amazon RDS 會為 PostgreSQL 資料庫執行個體建立 SSL 憑證。如果您啟用 SSL 憑證驗證，則 SSL 憑證會包含資料庫執行個體端點做為 SSL 憑證的一般名稱 (CN)，以防止詐騙攻擊。

主題

- [透過 SSL 連接至 PostgreSQL 資料庫執行個體](#)
- [需要以 SSL 連線至 PostgreSQL 資料庫執行個體](#)
- [判斷 SSL 連線狀態](#)
- [RDS for PostgreSQL 中的 SSL 密碼套件](#)

透過 SSL 連接至 PostgreSQL 資料庫執行個體

透過 SSL 連接至 PostgreSQL 資料庫執行個體

1. 下載憑證。

如需有關下載憑證的詳細資訊，請參閱。

2. 透過 SSL 連接至 PostgreSQL 資料庫執行個體。

當您使用 SSL 來連線時，用戶端可以選擇是否驗證憑證鏈。如果您的連線參數指定 `sslmode=verify-ca` 或 `sslmode=verify-full`，則用戶端在信任存放區必須有 RDS CA 憑證，或在連線 URL 中必須參考這些憑證。此需求是為了驗證用於簽署資料庫憑證的憑證鏈。

當用戶端 (例如 `psql` 或 `JDBC`) 設有 SSL 支援時，依預設，用戶端會先嘗試以 SSL 連線至資料庫。如果用戶端無法以 SSL 連線，則回復為不以 SSL 來連線。以 `libpq` 為基礎的用戶端 (例如 `psql`) 和 `JDBC` 使用的預設 `sslmode` 模式不同。以 `libpq` 為基礎的用戶端預設使用 `prefer`，而 `JDBC` 用戶端預設使用 `verify-full`。

使用 `sslrootcert` 參數來參考憑證，例如 `sslrootcert=rds-ssl-ca-cert.pem`。

下列是使用 `psql` 連線至使用具認證驗證 SSL 之 PostgreSQL 資料庫執行個體的範例。

```
$ psql "host=db-name.555555555555.ap-southeast-1.rds.amazonaws.com
port=5432 dbname=testDB user=testuser sslrootcert=rds-ca-rsa2048-g1.pem
sslmode=verify-full"
```

需要以 SSL 連線至 PostgreSQL 資料庫執行個體

您可以使用 `rds.force_ssl` 參數來要求對 PostgreSQL 資料庫執行個體的連線使用 SSL。對於 RDS for PostgreSQL 第 15 版，預設 `rds.force_ssl` 參數會設定為 1 (開啟)。所有其他 RDS for PostgreSQL 主要版本 14 及更舊版本都會將 `rds.force_ssl` 參數的預設值設定為 0 (關閉)。您可以將 `rds.force_ssl` 參數設為 1 (開啟)，以要求對資料庫執行個體的連線使用 SSL。

如要變更此參數值，您需要建立自訂資料庫參數群組。然後，您可變更您自訂資料庫參數群組中的 `rds.force_ssl` 值為 1，以開啟此功能。若於建立 RDS for PostgreSQL 資料庫執行個體之前準備了自訂資料庫參數群組，則可於建立程序期間加以選擇 (而非預設參數群組)。若您於 RDS for PostgreSQL 資料庫執行個體已執行之後進行此作業，則需要重新啟動執行個體，如此您的執行個體便可使用自訂參數群組。如需詳細資訊，請參閱 [使用參數群組](#)。

當 `rds.force_ssl` 功能在資料庫執行個體上處於活動狀態時，未使用 SSL 的連線嘗試將遭到拒絕，並顯示下列訊息：

```
$ psql -h db-name.555555555555.ap-southeast-1.rds.amazonaws.com port=5432 dbname=testDB
user=testuser
psql: error: FATAL: no pg_hba.conf entry for host "w.x.y.z", user "testuser", database
"testDB", SSL off
```

判斷 SSL 連線狀態

當您連接至資料庫執行個體時，登入橫幅中會顯示連線的加密狀態：

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
postgres=>
```

您也可以載入 `sslinfo` 延伸，然後呼叫 `ssl_is_used()` 函數，以判斷是否正在使用 SSL。如果連線正在使用 SSL，此函數會傳回 `t`，否則傳回 `f`。

```
postgres=> CREATE EXTENSION sslinfo;
CREATE EXTENSION
postgres=> SELECT ssl_is_used();
ssl_is_used
-----
t
(1 row)
```

如需更多詳細資訊，您可使用下列查詢，從 `pg_settings` 取得資訊：

```
SELECT name as "Parameter name", setting as value, short_desc FROM pg_settings WHERE
name LIKE '%ssl%';
```

Parameter name	value	short_desc
ssl	on	Enables SSL connections.
ssl_ca_file	/rdsdbdata/rds-metadata/ca-cert.pem	Location of the SSL certificate authority file.

```

ssl_cert_file          | /rdsdbdata/rds-metadata/server-cert.pem |
Location of the SSL server certificate file.
ssl_ciphers            | HIGH:!aNULL:!3DES                      |
Sets the list of allowed SSL ciphers.
ssl_crl_file          |                                          |
Location of the SSL certificate revocation list file.
ssl_dh_params_file    |                                          |
Location of the SSL DH parameters file.
ssl_ecdh_curve        | prime256v1                              |
Sets the curve to use for ECDH.
ssl_key_file          | /rdsdbdata/rds-metadata/server-key.pem  |
Location of the SSL server private key file.
ssl_library           | OpenSSL                                  |
Name of the SSL library.
ssl_max_protocol_version |                                          |
Sets the maximum SSL/TLS protocol version to use.
ssl_min_protocol_version | TLSv1.2                                  |
Sets the minimum SSL/TLS protocol version to use.
ssl_passphrase_command |                                          |
Command to obtain passphrases for SSL.
ssl_passphrase_command_supports_reload | off                                     |
Also use ssl_passphrase_command during server reload.
ssl_prefer_server_ciphers | on                                       |
Give priority to server ciphersuite order.
(14 rows)

```

您還可使用下列查詢，依程序、用戶端和應用程式收集有關 RDS for PostgreSQL 資料庫執行個體之 SSL 使用情況的所有資訊：

```

SELECT datname as "Database name", username as "User name", ssl, client_addr,
application_name, backend_type
FROM pg_stat_ssl
JOIN pg_stat_activity
ON pg_stat_ssl.pid = pg_stat_activity.pid
ORDER BY ssl;
Database name | User name | ssl | client_addr | application_name |
backend_type
-----+-----+----+-----+-----
+-----+
launcher      |          | f   |            |                  | autovacuum
              | rdsadmin | f   |            |                  | logical
replication launcher

```

writer		f			background
checkpointer		f			
rdsadmin backend	rdsadmin	t	127.0.0.1		walwriter client
rdsadmin backend	rdsadmin	t	127.0.0.1	PostgreSQL JDBC Driver	client
postgres backend	postgres	t	204.246.162.36	psql	client
(8 rows)					

如要識別用於 SSL 連線的密碼，可依如下方式查詢：

```
postgres=> SELECT ssl_cipher();
ssl_cipher
-----
DHE-RSA-AES256-SHA
(1 row)
```

若要進一步了解 `sslmode`，請參閱 PostgreSQL 文件中的 [資料庫連線控制函數](#)。

RDS for PostgreSQL 中的 SSL 密碼套件

PostgreSQL 組態參數 [ssl_cipher](#) 指定 SSL 連線允許的密碼套件類別。下表列出 RDS for PostgreSQL 中使用的預設密碼套件。

PostgreSQL 引擎版本	密碼套件
16	HIGH:!aNULL:!3DES
15	HIGH:!aNULL:!3DES
14	HIGH:!aNULL:!3DES
13	HIGH:!aNULL:!3DES
12	HIGH:!aNULL:!3DES
11.4 及更高次要版本	HIGH:MEDIUM:+3DES:!aNULL:!RC4

PostgreSQL 引擎版本	密碼套件
11.1、11.2	HIGH:MEDIUM:+3DES:!aNULL
10.9 及更高次要版本	HIGH:MEDIUM:+3DES:!aNULL:!RC4
10.7 及較低次要版本	HIGH:MEDIUM:+3DES:!aNULL

更新應用程式使用新的 SSL/TLS 憑證來連線至 PostgreSQL 資料庫執行個體

用於 Secure Socket Layer 或 Transport Layer Security (SSL/TLS) 的憑證通常具有設定生命週期。當服務供應商更新其憑證授權機構 (CA) 憑證時，用戶端必須更新其應用程式以使用新憑證。您可以在下文找到相關資訊，了解如何判斷用戶端應用程式是否使用 SSL/TLS 連線至 Amazon RDS for PostgreSQL 資料庫執行個體。您還可以找到如何檢查這些應用程式在連線時是否驗證伺服器憑證的相關資訊。

Note

設定為在 SSL/TLS 連線之前驗證伺服器憑證的用戶端應用程式，必須在用戶端的信任存放區中具有有效的 CA 憑證。在需要時更新用戶端信任存放區以取得新憑證。

更新用戶端應用程式信任存放區中的 CA 憑證之後，您就可以在資料庫執行個體輪換憑證。強烈建議先非生產環境中測試這些步驟，再於生產環境中實作。

如需憑證輪換的詳細資訊，請參閱[輪換您的 SSL/TLS 憑證](#)。如需下載憑證的詳細資訊，請參閱。如需對 PostgreSQL 資料庫執行個體使用 SSL/TLS 的資訊，請參閱[將 SSL 與 PostgreSQL 資料庫執行個體搭配使用](#)。

主題

- [判斷任何應用程式是否使用 SSL 連線至 PostgreSQL 資料庫執行個體](#)
- [判斷用戶端是否需要驗證憑證才能連線](#)
- [更新應用程式信任存放區](#)
- [針對不同類型的應用程式使用 SSL/TLS 連線](#)

判斷任何應用程式是否使用 SSL 連線至 PostgreSQL 資料庫執行個體

在資料庫執行個體組態中檢查 `rds.force_ssl` 參數的值。對於使用 PostgreSQL 版本 15 之前版本的資料庫執行個體，`rds.force_ssl` 參數預設會設定為 0 (關閉)。對於使用 PostgreSQL 版本 15 及更新主要版本的資料庫執行個體，`rds.force_ssl` 預設會設定為 1 (開啟)。如果 `rds.force_ssl` 參數設為 1 (開啟)，則用戶端需要使用 SSL/TLS 進行連線。如需參數群組的詳細資訊，請參閱[使用參數群組](#)。

如果您使用 RDS PostgreSQL 9.5 版或更新的主要版本，而 `rds.force_ssl` 未設為 1 (開啟)，請查詢 `pg_stat_ssl` 檢視，以檢查使用 SSL 的連線。例如，下列查詢只傳回 SSL 連線和關於使用 SSL 的用戶端的資訊。

```
SELECT datname, username, ssl, client_addr
   FROM pg_stat_ssl INNER JOIN pg_stat_activity ON pg_stat_ssl.pid =
pg_stat_activity.pid
  WHERE ssl is true and username <> 'rdsadmin';
```

只有使用 SSL/TLS 連線的列才會顯示連線的相關資訊。下列為範例輸出。

```
datname | username | ssl | client_addr
-----+-----+----+-----
benchdb | pgadmin  | t   | 53.95.6.13
postgres | pgadmin  | t   | 53.95.6.13
(2 rows)
```

此查詢只顯示查詢當時的連線。沒有結果不代表沒有應用程式使用 SSL 連線。其他 SSL 連線可能在不同時間建立。

判斷用戶端是否需要驗證憑證才能連線

當用戶端 (例如 `psql` 或 `JDBC`) 設有 SSL 支援時，依預設，用戶端會先嘗試以 SSL 連線至資料庫。如果用戶端無法以 SSL 連線，則回復為不以 SSL 來連線。以 `libpq` 為基礎的用戶端 (例如 `psql`) 和 `JDBC` 使用的預設 `sslmode` 模式不同。以 `libpq` 為基礎的用戶端預設使用 `prefer`，而 `JDBC` 用戶端預設使用 `verify-full`。只有在提供 `sslmode` 設為 `verify-ca` 或時，才會 `sslrootcert` 驗證伺服器上的憑證 `verify-full`。如果憑證無效，則擲出錯誤。

用於 `PGSSLROOTCERT` 使用 `PGSSLMODE` 環境變數驗證憑證，並將其 `PGSSLMODE` 設定為 `verify-ca` 或 `verify-full`。


```
PGSSLMODE=verify-full PGSSLROOTCERT=/fullpath/ssl-cert.pem psql -h  
pgdbidentifier.cxvxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

使用 `sslrootcert` 引數以連接字串格式驗證憑證，並 `sslmode` 設定為 `verify-ca` 或 `verify-full` 來驗證憑證。 `sslmode`

```
psql "host=pgdbidentifier.cxvxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=verify-full  
sslrootcert=/full/path/ssl-cert.pem user=masteruser dbname=postgres"
```

例如，在上述案例中，如果您使用無效根憑證，則在用戶端會看到類似以下的錯誤。

```
psql: SSL error: certificate verify failed
```

更新應用程式信任存放區

如需為 PostgreSQL 應用程式更新信任存放區的資訊，請參閱 PostgreSQL 文件中的 [使用 SSL 保護 TCP/IP 連線的安全](#)。

如需下載根憑證的資訊，請參閱。

如需匯入憑證的範例指令碼，請參閱 [將憑證匯入信任存放區的範例指令碼](#)。

Note

更新信任存放區時，除了新增憑證，您還可以保留舊憑證。

針對不同類型的應用程式使用 SSL/TLS 連線

以下提供針對不同類型的應用程式使用 SSL/TLS 連線的資訊：

- psql

以連線字串或環境變數指定選項，從命令列叫用此用戶端。若為 SSL/TLS 連線，相關選項為 `sslmode` (環境變數 `PGSSLMODE`)、`sslrootcert` (環境變數 `PGSSLROOTCERT`)。

如需完整的選項清單，請參閱 PostgreSQL 文件中的 [參數關鍵字](#)。如需完整的環境變數清單，請參閱 PostgreSQL 文件中的 [環境變數](#)。

- pgAdmin

這個以瀏覽器為基礎的用戶端是更方便連線至 PostgreSQL 資料庫的介面。

如需設定連線的資訊，請參閱 [pgAdmin 文件](#)。

- JDBC


JDBC 可讓 Java 應用程式連線至資料庫。

如需使用 JDBC 連線至 PostgreSQL 資料庫的一般資訊，請參閱 PostgreSQL JDBC 驅動程式文件中的 [連線至資料庫](#)。如需使用 SSL/TLS 來連線的資訊，請參閱 PostgreSQL JDBC 驅動程式文件中的 [設定用戶端](#)。

- Python

常用來連線至 PostgreSQL 資料庫的 Python 程式庫是 psycopg2。

如需使用 psycopg2 的資訊，請參閱 [psycopg2 文件](#)。如需如何連線至 PostgreSQL 資料庫的簡短教學課程，請參閱 [Psycopg2 教學](#)。您可以在 [psycopg2 模組內容](#) 中找到 connect 命令接受的選項的相關資訊。

 Important

判斷資料庫連線使用 SSL/TLS 並更新應用程式信任存放區之後，您可以更新資料庫以使用 rds-ca-rsa 2048-g1 憑證。如需說明，請參閱 [透過修改資料庫執行個體或叢集來更新 CA 憑證](#) 中的步驟 3。

搭配 Amazon RDS for PostgreSQL 使用 Kerberos 身分驗證

在使用者連線至執行 PostgreSQL 的資料庫執行個體時，您可以使用 Kerberos 驗證用戶身分。若要這麼做，請將資料庫執行個體設定為使用 AWS Directory Service for Microsoft Active Directory 進行 Kerberos 身分驗證。AWS Directory Service for Microsoft Active Directory 也被稱為 AWS Managed Microsoft AD。其是一個可與 AWS Directory Service 搭配使用的功能。如需進一步了解，請參閱《AWS Directory Service 管理指南》中的[什麼是 AWS Directory Service ?](#)。

請先建立 AWS Managed Microsoft AD 目錄來存放使用者登入資料。然後，將 Active Directory 網域和其他資訊提供給 PostgreSQL 資料庫執行個體。當使用者向 PostgreSQL 資料庫執行個體進行驗證時，身分驗證請求會轉送到 AWS Managed Microsoft AD 目錄。

將您的所有登入資料保留在相同目錄可以節省您的時間和精力。這樣您就有一個集中的位置來存放及管理多個資料庫執行個體的登入資料。使用目錄也可以改善您的整體安全性描述檔。

除此之外，您也可以從自己的內部部署 Microsoft Active Directory 存取登入資料。若要執行這項操作，請建立信任網域關聯，讓 AWS Managed Microsoft AD 目錄信任您的內部部署 Microsoft Active Directory。如此一來，使用者就可以透過在存取您內部部署網路的工作負載時的相同 Windows 單一登入 (SSO) 體驗，來存取 PostgreSQL 執行個體。

資料庫可以使用密碼身分驗證，也可以使用密碼身分驗證搭配 Kerberos 或 AWS Identity and Access Management (IAM) 身分驗證。如需 IAM 身分驗證的詳細資訊，請參閱[適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

主題

- [區域和版本可用性](#)
- [對 PostgreSQL 資料庫執行個體進行 Kerberos 身分驗證的概觀](#)
- [為 PostgreSQL 資料庫執行個體設定 Kerberos 身分驗證](#)
- [管理網域中的資料庫執行個體](#)
- [使用 Kerberos 身分驗證連線至 PostgreSQL](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域 而有所不同。如需有關 RDS for PostgreSQL 搭配 Kerberos 驗證的版本和區域可用性的詳細資訊，請參閱[Amazon RDS 中的 Kerberos 身分驗證支援的區域和資料庫引擎](#)。

對 PostgreSQL 資料庫執行個體進行 Kerberos 身分驗證的概觀

若要為 PostgreSQL 資料庫執行個體設定 Kerberos 身分驗證，請遵循下列步驟，稍後會提供更詳細的說明：

1. 使用 AWS Managed Microsoft AD 來建立 AWS Managed Microsoft AD 目錄。您可以使用 AWS Management Console、AWS CLI 或 AWS Directory Service API 來建立目錄。請務必開啟目錄安全性群組上的相關輸出連接埠，以便目錄可以與執行個體通訊。
2. 建立可提供 Amazon RDS 存取權以對 AWS Managed Microsoft AD 目錄進行呼叫的角色。若要這麼做，需建立一個使用受管 IAM 政策 AmazonRDSDirectoryServiceAccess 的 AWS Identity and Access Management (IAM) 角色。

針對可允許存取的 IAM 角色，您需要在 AWS Security Token Service 帳戶的正確 AWS STS 區域中啟用 AWS (AWS) 端點，如此 AWS STS 端點會在所有 AWS 區域中預設為啟用，接下來無須更進一步的動作，就可以直接使用。如需詳細資訊，請參閱《IAM 使用者指南》中的在 [AWS STS 區域內啟用和停用 AWS](#)。

3. 使用 Microsoft Active Directory 工具，在 AWS Managed Microsoft AD 目錄中建立和設定使用者。如需在 Active Directory 建立使用者的詳細資訊，請參閱《AWS 管理指南》中的 [管理 AWS Directory Service Managed Microsoft AD 中的使用者和群組](#)。
4. 如果您打算在不同 AWS 帳戶或 Virtual Private Cloud (VPC) 中尋找目錄和資料庫執行個體，請設定 VPC 互連。如需詳細資訊，請參閱《Amazon VPC Peering Guide》中的 [什麼是 VPC 互連？](#)。
5. 使用下列其中一種方法，從主控台、CLI 或 RDS API 建立或修改 PostgreSQL 資料庫執行個體：
 - [建立 Amazon RDS 資料庫執行個體](#)
 - [修改 Amazon RDS 資料庫執行個體](#)
 - [從資料庫快照還原](#)
 - [將資料庫執行個體還原至指定的時間](#)

您可以在與目錄相同的 Amazon Virtual Private Cloud (VPC) 中或在不同的 AWS 帳戶或 VPC 中尋找執行個體。當您建立或修改 PostgreSQL 資料庫執行個體時，請執行下列動作：

- 請提供您建立目錄時產生的網域識別符 (d-* 識別符)。
 - 請提供所建立的 IAM 角色名稱。
 - 確保資料庫執行個體安全群組可以從目錄的安全群組接收傳入流量。
6. 使用 RDS 主要使用者登入資料來連接至 PostgreSQL 資料庫執行個體。然後，在 PostgreSQL 中建立要在外部識別的使用者。外部識別的使用者可以使用 Kerberos 身分驗證來登入 PostgreSQL 資料庫執行個體。

為 PostgreSQL 資料庫執行個體設定 Kerberos 身分驗證

若要設定 Kerberos 身分驗證，請遵循下列步驟。

主題

- [第 1 步：使用創建一個目錄 AWS Managed Microsoft AD](#)
- [步驟 2：\(選擇性\) 建立內部部署作用中目錄之間的信任關係，AWS Directory Service](#)
- [步驟 3：為 RDS 創建 IAM 角色以訪問 AWS Directory Service](#)
- [步驟 4：建立和設定使用者](#)
- [步驟 5：啟用目錄和資料庫執行個體之間的跨 VPC 流量](#)
- [步驟 6：建立或修改 PostgreSQL 資料庫執行個體](#)
- [步驟 7：針對您的 Kerberos 主體建立 PostgreSQL 使用者](#)
- [步驟 8：設定 PostgreSQL 用戶端](#)

第 1 步：使用創建一個目錄 AWS Managed Microsoft AD

AWS Directory Service 在 AWS 雲端中建立完全受管理的作用中目錄。當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會為您建立兩個網域控制站和 DNS 伺服器。目錄伺服器是在 VPC 的不同子網路中建立。此備援有助於確保即使發生故障，您仍然可以存取目錄。

當您建立 AWS Managed Microsoft AD 目錄時，AWS Directory Service 會代表您執行下列工作：

- 在 VPC 內設定 Active Directory。
- 建立含有使用者名稱 Admin 與指定密碼的目錄管理員帳戶。您可以使用此帳戶來管理目錄。

Important

請務必儲存此密碼。AWS Directory Service 不會儲存此密碼，也無法擷取或重設密碼。

- 建立目錄控制器的安全群組。安全群組必須允許與 PostgreSQL 資料庫執行個體進行通訊。

啟動時 AWS Directory Service for Microsoft Active Directory，AWS 會建立包含所有目錄物件的組織單位 (OU)。此 OU 有您在建立目錄時所輸入的 NetBIOS 名稱，位於根網域中。網域根目錄擁有及管理 AWS。

使用您的 AWS Managed Microsoft AD 目錄建立的 Admin 帳戶具有 OU 最常見系統管理活動的權限：

- 建立、更新或刪除使用者
- 將資源 (例如檔案或列印伺服器) 新增至您的網域，然後對您 OU 中的使用者指派這些資源的許可
- 建立額外的 OU 和容器
- 委派授權
- 從 Active Directory 資源回收筒還原已刪除的物件
- PowerShell 在活動目錄 Web 服務上運行 Windows 的活動目錄和域名服務 (DNS) 模塊

Admin 帳戶也有權執行下列全網域活動：

- 管理 DNS 組態 (新增、移除或更新記錄、區域和轉寄站)
- 檢視 DNS 事件日誌
- 檢視安全事件日誌

若要建立目錄 AWS Managed Microsoft AD

1. 在 [AWS Directory Service 主控台](#) 中，選擇 Directories (目錄)，然後選擇 Set up directory (設定目錄)。
2. 選擇 AWS Managed Microsoft AD。AWS Managed Microsoft AD 為目前支援與 Amazon RDS 搭配使用的唯一選項。
3. 選擇 Next (下一步)。
4. 在 Enter directory information (輸入目錄資訊) 頁面上，提供下列資訊：

版本

選擇滿足您需求的版本。

目錄 DNS 名稱

目錄的完全合格名稱，例如 **corp.example.com**。

目錄 NetBIOS 名稱

目錄的簡短名稱，例如：CORP。

目錄描述

選擇填寫其他目錄說明。

管理員密碼

目錄管理員的密碼。目錄建立程序會建立含有使用者名稱 Admin 與這組密碼的管理者帳戶。

目錄管理員密碼不得包含 "admin" 字組。密碼區分大小寫，長度須為 8 至 64 個字元。至少須有一位字元屬於以下四種類型中的三類：

- 小寫字母 (a–z)
- 大寫字母 (A–Z)
- 數字 (0–9)
- 非英數字元 (~!@#%&* _+=`|\(){}[]:;'"<>.,?/)

Confirm password (確認密碼)

重新輸入管理員密碼。

Important

請確定您已儲存此密碼。AWS Directory Service 不會儲存此密碼，也無法擷取或重設密碼。

5. 選擇 Next (下一步)。
6. 在 Choose VPC and subnets (選擇 VPC 和子網路) 頁面上，提供下列資訊：

VPC

選擇目錄的 VPC。您可在此相同 VPC 或不同 VPC 中建立 PostgreSQL 資料庫執行個體。

子網路

選擇目錄伺服器的子網路。這兩個子網路必須位於不同的可用區域。

7. 選擇 Next (下一步)。
8. 檢閱目錄資訊。如果需要變更，請選擇 Previous (上一步)，然後進行變更。若資訊無誤，請選擇 Create directory (建立目錄)。

Review & create

Review

Directory type Microsoft AD	VPC vpc-8b6b78e9 ()
Directory DNS name corp.example.com	Subnets subnet-75128d10 (, us-east-1a) subnet-f51665dd (, us-east-1b)
Directory NetBIOS name CORP	
Directory description My directory	

Pricing

Edition Standard	Free trial eligible Learn more 30-day limited trial
~USD () *	
* Includes two domain controllers, USD ()/mo for each additional domain controller.	

Cancel Previous **Create directory**

建立目錄需要幾分鐘的時間。成功建立時，Status (狀態) 值會變更為 Active (作用中)。

若要查看目錄的資訊，請選擇目錄清單中的目錄 ID。請記下 Directory ID (目錄 ID) 值，建立或修改 PostgreSQL 資料庫執行個體時需要此值。

The screenshot shows the AWS Directory Service console for a directory with ID d-90670a8d36. The 'Directory details' section is displayed, with the 'Directory ID' field circled in red. The details include:

Property	Value	Property	Value
Directory type	Microsoft AD	Status	Active
Edition	Standard	Last updated	Tuesday, January 7, 2020
Directory ID	d-90670a8d36	Launch time	Tuesday, January 7, 2020
Directory DNS name	corp.example.com		
Directory NetBIOS name	CORP		
Description	My directory		
VPC	vpc-6594f31c		
Subnets	subnet-7d36a227, subnet-a2ab49c6		
Availability zones	us-east-1c, us-east-1d		
DNS address	[Redacted]		

Navigation tabs at the bottom include: Application management, Scale & share, Networking & security, and Maintenance.

步驟 2 : (選擇性) 建立內部部署作用中目錄之間的信任關係，AWS Directory Service

如果您不打算使用自己的內部部署 Microsoft Active Directory，請跳至[步驟 3 : 為 RDS 創建 IAM 角色](#)以訪問 [AWS Directory Service](#)。

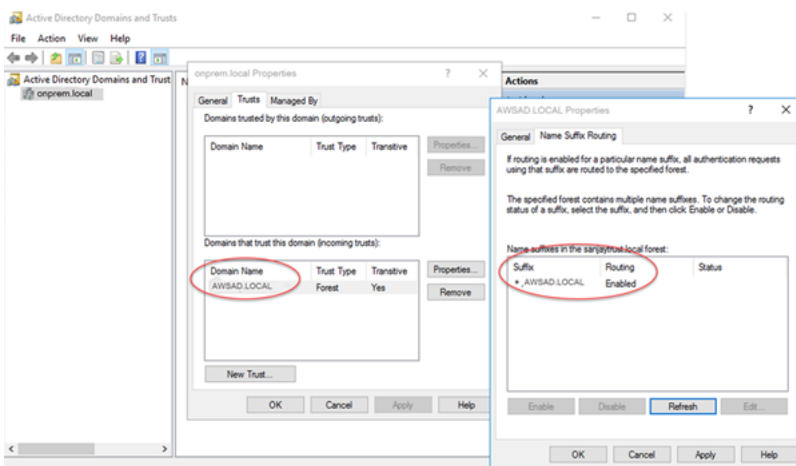
若要使用內部部署 Active Directory 取得 Kerberos 驗證，您必須使用內部部署 Microsoft Active Directory 與目 AWS Managed Microsoft AD 錄 (在中建立) 之間使用樹系信任來建立信任網域關係。[第 1 步 : 使用創建一個目錄 AWS Managed Microsoft AD](#)信任可以是單向的，其中 AWS Managed Microsoft AD 目錄信任內部部署 Microsoft 活動目錄。信任也可以是雙向，其中兩個 Active Directory

互相信任。如需有關使用設定信任的詳細資訊 AWS Directory Service，請參閱《AWS Directory Service 管理指南》中的[建立信任關係的時機](#)。

Note

如果您使用內部部署的 Microsoft 活動目錄，Windows 用戶端會使用的網域名稱在端點 AWS Directory Service 中連線，而不是 `rds.amazonaws.com`。如需進一步了解，請參閱[使用 Kerberos 身分驗證連線至 PostgreSQL](#)。

請確定您的內部部署 Microsoft Active Directory 網域名稱包含對應至新建立之信任關係的 DNS 尾碼路由。以下螢幕擷取畫面顯示了一個範例。



步驟 3：為 RDS 創建 IAM 角色以訪問 AWS Directory Service

若要讓 AWS Directory Service 為您撥打電話，您的 AWS 帳戶需要使用受管 IAM 政策的 IAM 角色 AmazonRDSDirectoryServiceAccess。此角色允許 Amazon RDS 呼叫 AWS Directory Service。

當您使用建立資料庫執行個體，AWS Management Console 且主控台使用者帳戶具有 `iam:CreateRole` 權限時，主控台會自動建立所需的 IAM 角色。在此情況下，角色名為 `rds-directoryservice-kerberos-access-role`。否則，您必須手動建立 IAM 角色。建立此 IAM 角色時 Directory Service，請選擇 AWS 受管政策並將其附加 AmazonRDSDirectoryServiceAccess 到該角色。

如需為服務建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 AWS 服務](#)。

Note

用於 Windows Authentication for RDS for Microsoft SQL Server 的 IAM 角色不可用於 Amazon RDS for PostgreSQL。

作為使用 AmazonRDSDirectoryServiceAccess 受管政策的替代方案，您可以建立具有必要許可的政策。在此情況下，IAM 角色必須有以下 IAM 信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

此角色也須具有下列 IAM 角色政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
}
```

步驟 4：建立和設定使用者

您可以透過使用 Active Directory 使用者和運算集區來建立使用者。這是 Active Directory Domain Services 和 Active Directory 輕量型目錄服務工具之一。如需詳細資訊，請參閱 Microsoft 文件中的[將使用者和電腦新增至 Active Directory 網域](#)。在此情況下，使用者是個人或其他實體，例如其電腦屬於網域，而其身分在目錄中維護。

若要在 AWS Directory Service 目錄中建立使用者，您必須連線至作為 AWS Directory Service 目錄成員的 Windows Amazon EC2 執行個體。同時，您必須以具有建立使用者之許可的使用者身分來登入。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[建立使用者](#)。

步驟 5：啟用目錄和資料庫執行個體之間的跨 VPC 流量

如果您打算在相同 VPC 中尋找目錄和資料庫執行個體，請略過本步驟，並移至[步驟 6：建立或修改 PostgreSQL 資料庫執行個體](#)。

如果您打算在不同 VPC 中尋找目錄和資料庫執行個體，請使用 VPC 互連或[AWS Transit Gateway](#) 來設定跨 VPC 流量。

下列程序會使用 VPC 互連來啟用 VPC 之間的流量。請遵循《Amazon Virtual Private Cloud 互連指南》中[什麼是 VPC 互連？](#)的指示。

使用 VPC 互連以啟用跨 VPC 流量

1. 設定適當的 VPC 路由規則，以確保網路流量可以雙向對流。
2. 確保資料庫執行個體安全群組可以從目錄的安全群組接收傳入流量。
3. 確保沒有網路存取控制清單 (ACL) 規則來封鎖流量。

如果目錄擁有不同的 AWS 帳戶，您必須共用該目錄。

在 AWS 帳戶之間共用目錄

1. 按照 AWS Directory Service 《管理指南》中的[教學課程：共用 AWS 受管 Microsoft AD 目錄以進行無縫 EC2 網域加入](#)中的指示，開始與將在其中建立資料庫執行個體的 AWS 帳戶共用目錄。
2. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台，並確保網域具有 SHARED 狀態，然後再繼續操作。

3. 使用資料庫執行個體的帳戶登入 AWS Directory Service 主控台時，請記下目錄 ID 值。您可以使用此目錄 ID，將資料庫執行個體加入網域。

步驟 6：建立或修改 PostgreSQL 資料庫執行個體

建立或修改要搭配目錄使用的 PostgreSQL 資料庫執行個體。您可以使用主控台、CLI 或 RDS API，將資料庫執行個體與目錄建立關聯。您可採用下列其中一種方式來這麼做：

- 使用主控台、[create-db-instance](#) CLI 命令或建立資料庫執行個體 RDS API [作業](#)，[建立新的 PostgreSQL 資料庫執行個體](#)。如需說明，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[modify-db-instance](#) CLI 命令或修改資料庫執行個體 RDS API 作業來[修改現有的 PostgreSQL 資料庫執行個體](#)。如需說明，請參閱[修改 Amazon RDS 資料庫執行個體](#)。
- 使用主控台、[restore-db-instance-from-db-快照](#) CLI 命令或還原資料庫快照 RDS API 作業，[從資料庫快照還原 PostgreSQL 資料庫執行個體](#)。InstanceFrom 如需說明，請參閱[從資料庫快照還原](#)。
- point-in-time 使用主控台、[restore-db-instance-to-point-in-time](#) CLI 命令或還原 RDS API 作業，將 PostgreSQL 資料庫執行個體還原至 InstanceToPointInTime 如需說明，請參閱[將資料庫執行個體還原至指定的時間](#)。

僅有 VPC 中的 PostgreSQL 資料庫執行個體支援 Kerberos 身分驗證。資料庫執行個體可在與目錄相同的 VPC 中，或在不同 VPC 中。資料庫執行個體必須使用允許在目錄 VPC 內傳入和傳出的安全群組，如此資料庫執行個體才能與目錄通訊。

主控台

使用主控台建立、修改或還原資料庫執行個體時，請在 Database authentication (資料庫身分驗證) 區段中選擇 Password and Kerberos authentication (密碼和 Kerberos 身分驗證)。然後選擇 Browse Directory (瀏覽目錄)。選取目錄或選擇 Create a new directory (建立新目錄) 以使用 Directory Service。

Database authentication

Database authentication options [Info](#)

- Password authentication
Authenticates using database passwords.
- Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Directory

docs-lab-active-dir.com (d-9...)

Browse Directory

AWS CLI

使用時 AWS CLI，資料庫執行個體需要下列參數才能使用您建立的目錄：

- 針對 `--domain` 參數，使用您建立目錄時產生的網域識別符 ("d-*" 識別符)。
- 針對 `--domain-iam-role-name` 參數，使用您建立的規則，其會使用受管 IAM 政策 `AmazonRDSDirectoryServiceAccess`。

例如，下列 CLI 命令會修改資料庫執行個體來使用目錄。

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

⚠ Important

如果您修改資料庫執行個體，以啟用 Kerberos 身分驗證，請在進行變更後重新啟動資料庫執行個體。

步驟 7：針對您的 Kerberos 主體建立 PostgreSQL 使用者

此時，您的 RDS for PostgreSQL 資料庫執行個體已加入 AWS Managed Microsoft AD 網域中。您在 [步驟 4：建立和設定使用者](#) 的目錄中建立的使用者必須設定為 PostgreSQL 資料庫使用者，並授予登入資料庫的權限。您可以透過以具有 `rds_superuser` 權限的資料庫使用者身分登入來執行此操作。

例如，如果您在建立 RDS for PostgreSQL 資料庫執行個體時接受預設值，請使用 `postgres`，如下列步驟中所示。

針對 Kerberos 主體建立 PostgreSQL 資料庫使用者

1. 使用 `psql` 搭配 `psql` 以連線至 RDS for PostgreSQL 資料庫執行個體端點。下列範例使用 `rds_superuser` 角色的預設 `postgres` 帳戶。

```
psql --host=cluster-instance-1.111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres --password
```

2. 針對您想要其有權存取資料庫之每個 Kerberos 主體建立資料庫使用者名稱 (Active Directory 使用者名稱)。針對該使用者名稱使用 Active Directory 執行個體中定義的標準使用者名稱 (身分) (亦即小寫 `alias` (Active Directory 中的使用者名稱))，以及 Active Directory 網域的大寫名稱。Active Directory 使用者名稱是經過外部驗證的使用者，因此請使用引號括住名稱，如下所示。

```
postgres=> CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;  
CREATE ROLE
```

3. 將 `rds_ad` 角色授予資料庫使用者。

```
postgres=> GRANT rds_ad TO "username@CORP.EXAMPLE.COM";  
GRANT ROLE
```

在您完成了針對 Active Directory 使用者身分建立所有 PostgreSQL 使用者之後，使用者就可以使用其 Kerberos 憑證存取 RDS for PostgreSQL 資料庫執行個體。

使用 Kerberos 進行驗證的資料庫使用者必須從屬於 Active Directory 網域成員的用戶端電腦執行這項作業。

已獲授予 `rds_ad` 角色的資料庫使用者也無法具有 `rds_iam` 角色。這也適用於巢狀成員資格。如需詳細資訊，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

步驟 8：設定 PostgreSQL 用戶端

若要設定 PostgreSQL 用戶端，請執行下列步驟：

- 建立可指向網域的 `krb5.conf` 檔案 (或同等檔案)。
- 確認流量可以在用戶端主機和之間流動 AWS Directory Service。請使用 Netcat 等網路公用程式檢查以下各項：

- 確認透過 DNS 傳送至連接埠 53 的流量。
- 確認透過 TCP/UDP 傳送至連接埠 53 和 Kerberos 的流量，其中包括用於 AWS Directory Service 的連接埠 88 和 464。
- 確定流量可透過資料庫連接埠在用戶端主機和資料庫執行個體之間往來。例如，您可以使用 psql 來連接和存取資料庫。

以下是的範例內容。AWS Managed Microsoft AD

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
    kdc = example.com
    admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

下列是內部部署 Microsoft Active Directory 的範例 krb5.conf 內容。

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
    kdc = example.com
    admin_server = example.com
}
ONPREM.COM = {
    kdc = onprem.com
    admin_server = onprem.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
.onprem.com = ONPREM.COM
onprem.com = ONPREM.COM
.rds.amazonaws.com = EXAMPLE.COM
.amazonaws.com.cn = EXAMPLE.COM
.amazon.com = EXAMPLE.COM
```


管理網域中的資料庫執行個體

您可以使用主控台、CLI 或 RDS API，來管理資料庫執行個體，以及其與 Microsoft Active Directory 的關係。例如，您可以使 Active Directory 產生關聯，以啟用 Kerberos 身分驗證。您也可以移除 Active Directory 的關聯，以停用 Kerberos 身分驗證。您可以將要由某個 Microsoft Active Directory 外部識別的資料庫執行個體移至另一個 Microsoft Active Directory。

例如，使用 CLI，您可以執行下列動作：

- 若要對失敗的成員資格重新嘗試啟用 Kerberos 身分驗證，請使用 [modify-db-instance](#) CLI 命令。並針對 `--domain` 選項指定目前成員資格的目錄 ID。
- 若要在資料庫執行個體上停用 Kerberos 身分驗證，請使用 [modify-db-instance](#) CLI 命令。並針對 `none` 選項指定 `--domain`。
- 若要將資料庫執行個體從某個網域移至另一個網域，請使用 [modify-db-instance](#) CLI 命令。並針對 `--domain` 選項指定新網域的網域識別符。

了解網域成員資格

在您建立或修改資料庫執行個體之後，網域的成員。您可以在主控台中或執行 [describe-db-instances](#) CLI 命令，來檢視網域成員資格狀態。資料庫執行個體的状态可以是下列其中一個：

- `kerberos-enabled` – 資料庫執行個體已啟用 Kerberos 身分驗證。
- `enabling-kerberos` – AWS 正在此資料庫執行個體上啟用 Kerberos 身分驗證。
- `pending-enable-kerberos` – 在此資料庫執行個體上擱置 Kerberos 身分驗證的啟用。
- `pending-maintenance-enable-kerberos` – AWS 將在下次排定的維護時段嘗試在資料庫執行個體上啟用 Kerberos 身分驗證。
- `pending-disable-kerberos` – 在此資料庫執行個體上擱置 Kerberos 身分驗證的停用。
- `pending-maintenance-disable-kerberos` – AWS 將在下次排定的維護時段嘗試在資料庫執行個體上停用 Kerberos 身分驗證。
- `enable-kerberos-failed` – 發生組態問題，導致 AWS 無法在資料庫執行個體上啟用 Kerberos 身分驗證。請更正問題，然後重新發出命令來修改資料庫執行個體。
- `disabling-kerberos` – AWS 正在此資料庫執行個體上停用 Kerberos 身分驗證。

由於網路連線問題或 IAM 角色不正確，請求啟用 Kerberos 身分驗證可能失敗。在某些情況下，當您建立或修改資料庫執行個體時，嘗試啟用 Kerberos 身分驗證可能會失敗。若是如此，請確定您使用正確的 IAM 角色，然後修改要加入網域的資料庫執行個體。

Note

只有搭配使用 Kerberos 身分驗證與 RDS for PostgreSQL 時，流量才會傳送至網域的 DNS 伺服器。所有其他 DNS 請求都會被視為執行 PostgreSQL 之資料庫執行個體上的傳出網路存取。如需傳出網路存取搭配 RDS for PostgreSQL 的詳細資訊，請參閱[針對傳出網路存取使用自訂 DNS 伺服器](#)。

使用 Kerberos 身分驗證連線至 PostgreSQL

您可以使用 pgAdmin 界面或 psql 等命令列界面搭配 Kerberos 身分驗證連接至 PostgreSQL。如需連線的詳細資訊，請參閱[連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)。如需取得端點、連接埠號碼和其他連線所需詳細資訊的資訊，請參閱[步驟 3：連線至 PostgreSQL 資料庫執行個體](#)。

pgAdmin

若要使用 Kerberos 身分驗證搭配 pgAdmin 連接至 PostgreSQL，請遵循下列步驟：

1. 在您的用戶端電腦上啟動 pgAdmin 應用程式。
2. 在 Dashboard (儀表板) 標籤上，選擇 Add New Server (新增伺服器)。
3. 在 Create - Server (建立 - 伺服器) 對話方塊中，於 General (一般) 標籤上輸入名稱，以識別 pgAdmin 中的伺服器。
4. 在 Connection (連線) 標籤上，輸入來自 RDS for PostgreSQL 資料庫的下列資訊：
 - 對於 Host (託管)，輸入的端點。RDS for PostgreSQL 資料庫執行個體。端點看起來類似下列：

```
RDS-DB-instance.111122223333.aws-region.rds.amazonaws.com
```

若要從 Windows 用戶端連線至內部部署 Microsoft Active Directory，使用 AWS Managed Active Directory 中的網域名稱，而不是託管端點中的 `rds.amazonaws.com`。例如，假設 AWS 受管 Active Directory 的網域名稱為 `corp.example.com`。然後對於 Host (託管)，端點的指定方式如下：

```
RDS-DB-instance.111122223333.aws-region.corp.example.com
```

- 針對 Port (連接埠)，輸入指派的連接埠。
- 針對 Maintenance database (維護資料庫)，輸入用戶端將連接的初始資料庫名稱。
- 針對 Username (使用者名稱)，輸入您在 [步驟 7：針對您的 Kerberos 主體建立 PostgreSQL 使用者](#) 中為 Kerberos 身分驗證輸入的使用者名稱。

5. 選擇 Save (儲存)。

Psql

若要使用 Kerberos 身分驗證搭配 psql 連接至 PostgreSQL，請遵循下列步驟：

1. 在命令提示中，執行下列命令。

```
kinit username
```

請將 *username* 換成使用者名稱。在提示字元中，輸入 Microsoft Active Directory 中為使用者存放的密碼。

2. 如果 PostgreSQL 資料庫執行個體是使用可公開存取的 VPC，請將資料庫執行個體端點的 IP 地址放入 EC2 用戶端上的 `/etc/hosts` 檔案。例如，使用下列命令取得 IP 地址，然後將該地址放入 `/etc/hosts` 檔案。

```
% dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
34.210.197.118

% echo " 34.210.197.118 PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/
hosts
```

如果使用來自 Windows 用戶端的內部部署 Microsoft Active Directory，則需要使用特殊化的端點進行連線。請使用 `rds.amazonaws.com` 受管 Active Directory 的網域名稱，而不是在主機端點中使用 Amazon 網域 AWS。

例如，假設您的 AWS 受管 Active Directory 的網域名稱為 `corp.example.com`。則為端點使用 `PostgreSQL-endpoint.AWS-Region.corp.example.com` 格式並將其放入 `/etc/hosts` 檔案中。

```
% echo " 34.210.197.118 PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/hosts
```

3. 使用下列 `psql` 命令來登入與 Active Directory 整合的 PostgreSQL 資料庫執行個體。

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com postgres
```

若要使用內部部署 Active Directory 從 Windows 用戶端登入 PostgreSQL 資料庫叢集，請使用下列 `psql` 命令搭配上一步驟的網域名稱 (`corp.example.com`)：

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-Region.corp.example.com postgres
```

針對傳出網路存取使用自訂 DNS 伺服器。

RDS for PostgreSQL 支援資料庫執行個體上的傳出網路存取，且允許來自客戶擁有的自訂 DNS 伺服器的網域名稱服務 (DNS) 解析。您僅可以透過您的自訂 DNS 伺服器，解析來自您 RDS for PostgreSQL 資料庫執行個體的完整網域名稱。

主題

- [開啟自訂 DNS 解析](#)
- [關閉自訂 DNS 解析](#)
- [設定自訂 DNS 伺服器](#)

開啟自訂 DNS 解析

若要在您的客戶 VPC 中開啟 DNS 解析，請先將自訂資料庫參數群組與您的 RDS for PostgreSQL 執行個體建立關聯。然後將參數設定為 1 來開啟 `rds.custom_dns_resolution`，接下來將資料庫執行個體重新啟動，讓變更生效。

關閉自訂 DNS 解析

若要在客戶 VPC 中關閉 DNS 解析，請首先關閉自訂資料庫參數群組的 `rds.custom_dns_resolution` 參數，方法是將資料庫參數群組設定為 0。然後重新啟動資料庫執行個體，讓變更生效。

設定自訂 DNS 伺服器

設定您的自訂 DNS 名稱伺服器之後，需要最多 30 分鐘的時間，才能將變更傳播到您的資料庫執行個體。變更傳播到您的資料庫執行個體之後，所有傳出網路流量都需要透過連接埠 53 對您的 DNS 伺服器進行 DNS 查閱查詢。

Note

如果您未設定自訂 DNS 伺服器，且設定 `rds.custom_dns_resolution` 為 1，則會使用 Amazon Route 53 私有區域來解析主機。如需詳細資訊，請參閱[使用私有託管區域](#)。

為 RDS for PostgreSQL 資料庫執行個體設定自訂 DNS 伺服器

1. 從連接至您 VPC 的動態主機組態協定 (DHCP) 選項集，將 `domain-name-servers` 選項設定為您 DNS 名稱伺服器的 IP 地址。如需詳細資訊，請參閱 [DHCP 選項集](#)。

Note

`domain-name-servers` 選項會接受最多四個值，但您的 Amazon RDS 資料庫執行個體只會使用第一個值。

2. 請確保您的 DNS 伺服器可以解析所有查閱查詢，包括公有 DNS 名稱、Amazon EC2 私有 DNS 名稱，以及客戶特定的 DNS 名稱。如果傳出網路流量包含您的 DNS 伺服器無法處理的任何 DNS 查閱，您的 DNS 伺服器就必須設定適當的上游 DNS 提供者。
3. 設定您的 DNS 伺服器以產生 512 個位元組或更少的使用者資料包通訊協定 (UDP) 回應。
4. 設定您的 DNS 伺服器以產生 1,024 個位元組或更少的傳輸控制通訊協定 (TCP) 回應。
5. 設定您的 DNS 伺服器以允許來自您的 Amazon RDS 資料庫執行個體透過連接埠 53 的傳入流量。如果您的 DNS 伺服器位於 Amazon VPC 中，VPC 必須具有一個安全群組，其中包含允許連接埠 53 上的 UDP 和 TCP 流量的傳入規則。如果您的 DNS 伺服器不在 Amazon VPC 中，則必須具有適當的防火牆設定，才能允許連接埠 53 上的 UDP 和 TCP 傳入流量。

如需更多詳細資訊，請參閱 [VPC 安全群組](#) 和 [新增與移除規則](#)。

6. 設定您 Amazon RDS 資料庫執行個體的 VPC，以允許透過連接埠 53 的傳出流量。您的 VPC 必須具有一個安全群組，其中包含允許連接埠 53 上的 UDP 和 TCP 流量的傳出規則。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [VPC 的安全群組](#) 和 [新增與移除規則](#)。

7. 必須正確設定 Amazon RDS 資料庫執行個體和 DNS 伺服器之間的路由路徑，才能允許 DNS 流量。

此外，如果 Amazon RDS 資料庫執行個體和 DNS 伺服器不在相同的 VPC 中，請務必在兩者之間設定對等連線。如需詳細資訊，請參閱《Amazon VPC 對等互連指南》中的 [什麼是 VPC 對等互連？](#)

升級 Amazon RDS 的 PostgreSQL 資料庫引擎

您可以為 PostgreSQL 資料庫管理兩種類型的升級：

- 作業系統更新 - Amazon RDS 偶爾可能需要更新資料庫的基礎作業系統，才能套用安全修正或作業系統變更。您可以使用 RDS 主控台、AWS Command Line Interface (AWS CLI) 或 RDS API 來決定 Amazon RDS 何時套用作業系統更新。如需作業系統更新的詳細資訊，請參閱[套用資料庫執行個體的更新](#)。
- 資料庫引擎升級 - 當 Amazon RDS 支援新版本的資料庫引擎時，您可以將資料庫升級為新版本。

此環境中的資料庫是 RDS for PostgreSQL 資料庫執行個體或多可用區域資料庫叢集。

PostgreSQL 資料庫有兩種引擎升級：主要版本升級和次要版本升級。

主要版本升級

主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。因此，您必須手動執行資料庫的主要版本升級。您可以透過修改資料庫執行個體或多可用區域資料庫叢集來啟動主要版本升級。在您執行主要版本升級之前，我們建議您遵循中所述的步驟[選擇 PostgreSQL 的主要版本升級](#)。

如果要升級具有區域內僅供讀取複本的資料庫執行個體，Amazon RDS 會將複本與主要資料庫執行個體一起升級。

Amazon RDS 不會升級多可用區域資料庫叢集僅供讀取複本。如果您執行異地同步備份資料庫叢集的主要版本升級，則其僅供讀取複本的複寫狀態會變更為已終止。升級完成後，您必須手動刪除並重新建立僅供讀取複本。

Tip

您可以使用藍/綠部署，將主要版本升級所需的停機時間降至最低。如需詳細資訊，請參閱[使用藍/綠部署進行資料庫更新](#)。

次要版本升級

反之，次要版本升級只包含與現有應用程式回溯相容的變更。您可以藉由修改資料庫來手動啟動次要版本升級。或者，您可以在建立或修改資料庫時啟用自動次要版本升級選項。這樣做表示 Amazon RDS 會在測試和核准新版本後自動升級您的資料庫。如果 PostgreSQL 資料庫使用僅供讀取複本，則必須先升級所有僅供讀取複本，然後再升級來源執行個體或叢集。

如果您的資料庫是異地同步備份資料庫執行個體部署，Amazon RDS 會同時升級主要和任何待命執行個體。因此，在升級完成之前，您的資料庫可能無法使用。如果您的資料庫是異地同步備份資料庫叢集部署，Amazon RDS 會一次升級一個讀取器資料庫執行個體。然後，其中一個讀取器資料庫執行個體會切換為新的寫入器資料庫執行個體。然後，Amazon RDS 會升級舊的寫入器執行個體 (現在是讀取器執行個體)。

Note

異地同步備份資料庫執行個體部署次要版本升級的停機時間可能會持續數分鐘。異地同步備份資料庫叢集通常會將次要版本升級的停機時間縮短至約 35 秒。與 RDS Proxy 搭配使用時，您可以進一步將停機時間減少到一秒或更短的時間。如需詳細資訊，請參閱 [使用 RDS Proxy](#)。或者，您可以使用開放原始碼資料庫代理伺服器 (例如 [ProxSQL](#)) 或 MySQL 的 [AWS JDBC 驅動程式](#)。 [PgBouncer](#)

如需詳細資訊，請參閱 [PostgreSQL 的自動次要版本升級](#)。如需有關手動執行次要版本升級的詳細資訊，請參閱 [手動升級引擎版本](#)。

如需有關資料庫引擎版本和取代資料庫引擎版本政策的詳細資訊，請參閱 Amazon RDS 常見問答集中的 [資料庫引擎版本](#)。

主題

- [PostgreSQL 升級概觀](#)
- [PostgreSQL 版本編號](#)
- [RDS 版本編號](#)
- [選擇 PostgreSQL 的主要版本升級](#)
- [如何執行主要版本升級](#)
- [PostgreSQL 的自動次要版本升級](#)
- [升級 PostgreSQL 延伸](#)

PostgreSQL 升級概觀

為了安全地升級您的資料庫，Amazon RDS 會使用 [PostgreSQL 文件](#) 中所述的 `pg_upgrade` 公用程式。

當您使用升級 AWS Management Console 資料庫時，它會顯示資料庫的有效升級目標。您也可以使用下列 AWS CLI 命令來識別資料庫的有效升級目標：

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine postgres \  
  --engine-version version-number \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --engine postgres ^  
  --engine-version version-number ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

例如，若要識別 PostgreSQL 12.13 版資料庫的有效升級目標，請執行下列命令：AWS CLI

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
  --engine postgres \  
  --engine-version 12.13 \  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
  --engine postgres ^  
  --engine-version 12.13 ^  
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --  
  output text
```

如果您的備份保留期大於 0，則 Amazon RDS 會在升級程序期間建立兩個資料庫快照。第一個資料庫快照是屬於完成任何升級變更之前的資料庫。如果資料庫升級失敗，您可以還原此快照，以建立執行舊版本的資料庫。升級完成後會建立第二個資料庫快照。

Note

您必須將資料庫的備份保留期設定為大於 0 的數字，Amazon RDS 才會在升級過程中拍攝資料庫快照。若要變更資料庫執行個體的備份保留期，請參閱 [the section called “修改資料庫執行個體”](#)。您無法為多可用區域資料庫叢集設定自訂備份保留期。

當您執行資料庫執行個體的主要版本升級時，任何區域內僅供讀取複本也都會自動升級。升級工作流程開始後，僅供讀取複本會等待主要資料庫執行個體上的 `pg_upgrade` 順利完成。然後，主要資料庫執行個體升級會等待僅供讀取複本升級完成。在升級完成之前，可能會遭遇停機。當您執行多可用區域資料庫叢集的主要版本升級時，其僅供讀取複本的複寫狀態會變更為已終止。

升級完成後，您就無法還原為舊版的資料庫引擎。若您想要恢復前一版本，請從升級前拍攝的資料庫快照進行還原，以建立新的資料庫。

PostgreSQL 版本編號

PostgreSQL 資料庫引擎的版本編號順序如下：

- 對於 PostgreSQL 第 10 版及更新版本，引擎版本號碼的格式為 `major.minor`。主要版本號是版本號碼的整數部分。次要版本號碼是版本號碼的小數部分。

主要版本升級會增加版本號碼的整數部分，例如從 `10.minor` 升級為 `11.minor`。

- 對於第 10 版之前的 PostgreSQL 版本，引擎版本號碼的格式為 `major.major.minor`。主要引擎版本號碼是整數和版本號碼的第一個小數部分。例如，`9.6` 是主要版本。次要版本號碼是版本號碼的第三部分。例如，對於第 `9.6.12` 版而言，`12` 是次要版本號碼。

主要版本升級會增加版本號碼的主要部分。例如，從 `9.6.12` 升級至 `11.14` 是主要版本升級，其中 `9.6` 和 `11` 是主要版本編號。

如需 RDS 延伸 Support 版本編號的相關資訊，請參閱 [Amazon RDS 擴展 Support 版本命名](#)。

RDS 版本編號

RDS 版本編號會使用 `major.minor.patch` 命名方案。RDS 修補程式版本中包含發行後新增到次要版本的重要錯誤修正。如需 RDS 延伸 Support 版本編號的相關資訊，請參閱 [Amazon RDS 擴展 Support 版本命名](#)。

若要識別資料庫的 Amazon RDS 版本編號，您必須先使用下列命令建立 `rds_tools` 延伸模組：

```
CREATE EXTENSION rds_tools;
```

從 PostgreSQL 15.2-R2 版開始，您可以使用以下 SQL 查詢，找出 RDS for PostgreSQL 資料庫的 RDS 版本編號：

```
postgres=> SELECT rds_tools.rds_version();
```

例如，查詢 RDS for PostgreSQL 15.2 資料庫會傳回以下內容：

```
rds_version
-----
 15.2.R2
(1 row)
```

選擇 PostgreSQL 的主要版本升級

主要版本升級可能包含無法與舊版資料庫回溯相容的變更。新功能可能導致現有的應用程式停止正確運作。因此，Amazon RDS 不會自動套用主要版本升級。若要執行主要版本升級，請手動修改資料庫。所有升級都務必要經過完整測試，以確認您的應用程式正常運作，再將升級套用至您的生產資料庫。進行 PostgreSQL 主要版本升級時，建議您遵循 [如何執行主要版本升級](#) 中描述的步驟。

當您將 PostgreSQL 單一可用區域資料庫執行個體或多可用區域資料庫執行個體部署升級到下一個主要版本時，任何與資料庫相關聯的僅供讀取複本也會升級到相同的下一個主要版本。在某些情況下，您可以在升級時跳到更高的主要版本。如果您的升級略過主要版本，僅供讀取複本也會升級至該目標主要版本。跳過其他主要版本直接升級至第 11 版存在某些限制。您可以按照 [如何執行主要版本升級](#) 所述的步驟中找到詳細資訊。

PostgreSQL 引擎升級時並不會升級大多數的 PostgreSQL 擴充功能。這些擴充功能必須單獨升級。如需詳細資訊，請參閱 [升級 PostgreSQL 延伸](#)。

您可以執行下列查詢，找出適用於 RDS for PostgreSQL 資料庫的主要版本：AWS CLI

```
aws rds describe-db-engine-versions --engine postgres --engine-version your-version
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

下表總結了所有可用版本的查詢結果。版本編號上的星號 (*) 表示該版本已棄用。如果您目前的版本已棄用，建議您升級到最新的次要版本升級目標，或升級到該版本的其他可用升級目標之一。如需 RDS

for PostgreSQL 9.6 版取代的詳細資訊，請參閱 [棄用 PostgreSQL 9.6 版](#)。如需 RDS for PostgreSQL 10 版取代的詳細資訊，請參閱 [棄用 PostgreSQL 第 10 版](#)。

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標									
16.2	16										
16.1	16	16									
15.7	16										
15.6	16	16	15								
15.5	16	16	16	15	15						
15.4	16	16	16	15	15	15					
15.3	16	16	16	15	15	15	15				
15.2	16	16	16	15	15	15	15	15			
14.10	16	15									
14.1	16	15	15	14							
14.10	16	15	15	15	14	14					
14.9	15	15	15	15	14	14	14				
14.8	15	15	15	15	15	14	14	14	14	14	

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標																			
14.7.*	15	15	15	15	15	15	14	14	14	14											
14.6*	15	15	15	15	15	15	14	14	14	14	14										
14.5*	15	15	15	15	15	15	14	14	14	14	14	14	14								
14.4*	15	15	15	15	15	15	14	14	14	14	14	14	14	14							
14.3*	15	15	15	15	15	15	14	14	14	14	14	14	14	14	14						
14.2*	15	15	15	15	15	15	14	14	14	14	14	14	14	14	14	14					
14.1*	15	15	15	15	15	15	14	14	14	14	14	14	14	14	14	14	14				
13.1*	16	15	14																		
13.1*	16	15	14	14	13																
13.1*	16	15	14	14	14	13	13	13													
13.1*	15	14	14	14	14	14	13	13	13	13											
13.1*	15	14	14	14	14	14	14	13	13	13	13										
13.9*	14	14	14	14	14	14	14	13	13	13	13	13									

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標																																								
13.8	14	14	14	14	14	14	14	14	14	13	13	13	13	13																												
13.7	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13																										
13.6	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13																							
13.5	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13																					
13.4	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13																				
13.3	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13	13	13																		
13.2 *	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13		
12.16	16	15	14	13																																						
12.15	16	15	14	13	13	12																																				
12.14	16	15	14	13	13	13	12	12	12																																	
12.13	15	14	13	13	13	13	13	13	12	12	12	12																														
12.12	15	14	13	13	13	13	13	13	12	12	12	12																														
12.11	14	13	13	13	13	13	13	12	12	12	12	12																														

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標																			
12.10	14	13	13	13	13	13	13	13	13	12	12	12	12	12							
12.10	14	14	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12			
12.10	14	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12		
12.9	14	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	
12.8	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	
12.7	13	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12.8
12.6 5*、 12.3 2*	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12	12	12			
11.2	16	15	14	13	12	11															
						RC															
						.20															
11.2	15	14	13	12	12	11															
11.2	15	14	13	12	12	12	11	11													
11.1	15	14	13	12	12	12	12	11	11	11											

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標																		
11.10	14	13	12	12	12	12	12	12	11	11	11	11								
11.11	14	13	12	12	12	12	12	12	12	11	11	11	11	11						
11.10	14	14	13	12	12	12	12	12	12	12	11	11	11	11	11	11				
11.10	14	13	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11			
11.14	14	13	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11
11.10	13	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11
11.10	13	12	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11
10.20	14	13	12	11	11	11	11	11												
10.20	14	13	12	11	11	11	11	11	11	11	10	10								
10.20	14	14	13	12	11	11	11	11	11	11	11	10	10							
10.20	14	13	12	11	11	11	11	11	11	11	11	10	10	10	10					
10.19	14	13	12	11	11	11	11	11	11	11	11	10	10	10	10					
10.18	13	12	11	11	11	11	11	11	11	11	11	10	10	10	10	10				
10.17	13	12	11	11	11	11	11	11	11	11	11	11	10	10	10	10	10	10	10	10

目前原始碼版本 (* 已取代)	最新主要版本升級目標	其他可用的升級目標																		
9.6.2	14	13	12	11	10	10														
9.6.2	13	12	11	10	10	10	9.6													
9.6.2	13	12	11	10	10	10	10	9.6	9.6											
9.6.1	9.6	14	13	12	11	10	10	9.6	9.6											
.6.18																				
6.17																				
.16*																				
15*																				
4*、																				
*、9.																				
、9.6																				
9.6.9																				
6.8*																				
6*、																				
、9.6																				
.6.2*																				
.1*																				

如何執行主要版本升級

在 Amazon RDS for PostgreSQL 資料庫上執行主要版本升級時，我們建議採取以下程序：

1. 準備好版本相容的參數群組 – 如果您使用的是自訂參數群組，您有兩個選擇。您可以為新資料庫引擎版本指定預設參數群組。您也可以為新資料庫引擎版本建立自己的自訂參數群組。如需詳細資訊，請參閱 [the section called “使用參數群組”](#) 及 [the section called “使用資料庫叢集參數群組”](#)。
2. 檢查不支援的資料庫類別 - 檢查資料庫的執行個體類別是否與您要升級的目標 PostgreSQL 版本相容。如需詳細資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。
3. 檢查是否有不支援的使用：
 - 備妥交易 – 嘗試升級之前，遞交或轉返所有開啟的備妥交易。

您可以使用下列查詢來確認資料庫上沒有開啟的預備交易。

```
SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
```

- Reg* 資料類型 – 嘗試升級之前，請移除所有 reg* 資料類型的使用。除了 regtype 和 regclass，您無法升級 reg* 資料類型。pg_upgrade 公用程式無法保留此資料類型，Amazon RDS 會使用該資料類型進行升級。

您可以使用下列查詢，確認在每個資料庫中未使用不支援的 reg* 資料類型：

```
SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
pg_catalog.pg_attribute a
WHERE c.oid = a.attrelid
AND NOT a.attisdropped
AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
                  'pg_catalog.regprocedure'::pg_catalog.regtype,
                  'pg_catalog.regoper'::pg_catalog.regtype,
                  'pg_catalog.regoperator'::pg_catalog.regtype,
                  'pg_catalog.regconfig'::pg_catalog.regtype,
                  'pg_catalog.regdictionary'::pg_catalog.regtype)
AND c.relnamespace = n.oid
AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

4. 處理邏輯複寫槽 - 如果資料庫具有任何邏輯複寫槽，就無法進行升級。邏輯複寫槽通常用於 AWS DMS 遷移，以及將資料表從資料庫複寫到資料湖、BI 工具和其他目標。升級之前，請確保您知道任何使用中邏輯複寫槽的用途，並確認可以刪除它們。如果邏輯複寫槽仍在使用的，則不應刪除它們，也無法繼續升級。

如果不需要邏輯複寫槽，可以使用以下 SQL 加以刪除：

```
SELECT * FROM pg_replication_slots;  
SELECT pg_drop_replication_slot(slot_name);
```

使用 `pglogical` 延伸模組的邏輯複寫設定也必須具有捨棄的插槽，才能成功進行主要版本升級。如需如何識別和捨棄使用 `pglogical` 延伸模組所建立之插槽的相關資訊，請參閱 [管理 RDS for PostgreSQL 的邏輯複寫槽](#)。

5. 處理僅供讀取複本 - 升級單一可用區域資料庫執行個體或多可用區域資料庫執行個體部署時，也會升級區域內僅供讀取複本以及主要資料庫執行個體。Amazon RDS 不會升級多可用區域資料庫叢集讀取複本。

您無法個別升級僅供讀取複本。如果可以，它可能會導致主要資料庫和複本資料庫具有不同 PostgreSQL 主要版本的情況。不過，僅供讀取複本升級可能會增加主要資料庫執行個體的停機時間。若要防止僅供讀取複本升級，請將複本提升為獨立執行個體，或在開始升級程序之前刪除複本。

升級程序會根據僅供讀取複本的目前參數群組，重新建立僅供讀取複本的參數群組。只有在升級完成後，您才能修改僅供讀取複本，以便將自訂參數群組套用至僅供讀取複本。如需僅供讀取複本的詳細資訊，請參閱 [使用 Amazon RDS for PostgreSQL 的僅供讀取複本](#)。

6. 執行備份 – 建議您先執行備份再執行主要版本升級，以便您的資料庫有已知的還原點。如果您的備份保留期大於 0，在升級前和升級後，升級程序會建立資料庫的資料庫快照。若要變更備份保留期，請參閱 [修改 Amazon RDS 資料庫執行個體](#) 和 [the section called “修改多可用區域資料庫叢集”](#)。

若要手動執行備份，請參閱 [the section called “為單一可用區資料庫執行個體建立資料庫快照”](#) 和 [the section called “建立多可用區域資料庫叢集快照”](#)。

7. 在主要版本升級之前更新特定延伸套件 – 如果您計劃在升級時跳過主要版本，您必須在執行主要版本升級之前更新特定延伸套件。例如，從 9.5.x 或 9.6.x 版升級為 11.x 版會跳過主要版本。要更新的擴充功能包括 PostGIS 和用於處理空間資料的相關擴充功能。
 - `address_standardizer`
 - `address_standardizer_data_us`
 - `postgis_raster`
 - `postgis_tiger_geocoder`
 - `postgis_topology`

針對您使用的每個擴充功能執行下列命令：

```
ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version';
```

如需詳細資訊，請參閱 [升級 PostgreSQL 延伸](#)。若要進一步了解 PostGIS 升級的詳細資訊，請參閱 [步驟 6：升級 PostGIS 擴充功能](#)。

8. 在主要版本升級之前捨棄特定延伸套件 – 跳過主要版本升級為 11.x 版本時，不支援更新 pgRouting 延伸套件。從 9.4.x、9.5.x 或 9.6.x 版升級為 11.x 版會跳過主要版本。安全的做法是捨棄 pgRouting 延伸套件，然後在升級之後再重新安裝該延伸套件的相容版本。有關您可以更新的延伸套件版本，請參閱 [支援的 PostgreSQL 擴充功能版本](#)。

PostgreSQL 11 或以上版本不再支援 tsearch2 和 chkpass 延伸套件。如果您升級為 11.x 版，請在升級之前捨棄 tsearch2 和 chkpass 延伸套件。

9. 刪除未知的資料類型 – 會根據目標版本丟棄 unknown 資料類型。

PostgreSQL 第 10 版已停止支援 unknown 資料類型。如果第 9.6 版資料庫使用 unknown 資料類型，升級至版本 10 會顯示錯誤訊息，如下所示：

```
Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:  
The instance could not be upgraded because the 'unknown' data type is used in user  
tables.  
Please remove all usages of the 'unknown' data type and try again."
```

若要尋找資料庫中的 unknown 資料類型，以便移除違規資料行或將其變更為支援的資料類型，請使用下列 SQL：


```
SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE  
'unknown';
```

- 10 執行升級停用試轉 – 強烈建議您先在生產資料庫的複本上測試主要版本升級，然後才在生產資料庫上嘗試此升級。您可以監控複寫測試資料庫上的執行計劃，以查看任何可能的執行計劃迴歸，並評估其效能。若要建立重複的測試執行個體，您可以從最近的快照還原資料庫，或將資料庫 point-in-time 還原到最新的可還原時間。

如需詳細資訊，請參閱 [the section called “從快照還原”](#) 或 [the section called “Point-in-time 回收”](#)。若是多可用區域資料庫叢集，請參閱 [the section called “從快照還原至多可用區域資料庫叢集”](#) 或 [the section called “將多可用區域資料庫叢集還原至指定時間”](#)。

如需執行升級的詳細資訊，請參閱 [the section called “手動升級引擎版本”](#)。


將 9.6 版資料庫升級到第 10 版時，請注意 PostgreSQL 10 會預設啟用平行查詢。升級之前，您可以在測試資料庫上將 `max_parallel_workers_per_gather` 參數變更為 2，藉此測試平行處理的影響。

 Note

在 `default.postgresql10` 資料庫參數群組中，`max_parallel_workers_per_gather` 參數的預設值為 2。

如需詳細資訊，請參閱 PostgreSQL 說明文件中的[平行查詢](#)。若要在第 10 版上停用平行處理，請將 `max_parallel_workers_per_gather` 參數設為 0。

在主要版本升級期間，`public` 和 `template1` 資料庫，以及每個資料庫中的 `public` 結構描述都會暫時重新命名。這些物件將連同其原始名稱及附加的隨機字串出現在日誌中。字串會被附加上去，以便在主要版本升級期間保留 `locale` 和 `owner` 之類的自訂設定。升級完成之後，物件就會改回其原始名稱。

 Note

在主要版本升級過程中，您無法 `point-in-time` 還原資料庫執行個體或異地同步備份資料庫叢集。Amazon RDS 執行升級後，會自動進行資料庫備份。您可以在升級開始之前和資料庫的自動備份完成之後執行 `point-in-time` 還原至次數。

11 如果預先檢查程序發生錯誤而升級失敗，請解決這些問題 – 在主要版本升級過程中，Amazon RDS for PostgreSQL 會先執行預先檢查程序，以找出可能導致升級失敗的任何問題。預先檢查程序會檢查執行個體中所有資料庫的所有潛在不相容的狀況。

如果預先檢查發現問題，會建立日誌事件以表示升級預先檢查失敗。資料庫的所有資料庫預先檢查程序詳細資訊位於名為 `pg_upgrade_precheck.log` 的升級日誌中。Amazon RDS 會將時間戳記附加至檔案名稱。如需檢視日誌檔案的詳細資訊，請參閱[監控 Amazon RDS 日誌檔案](#)。

如果僅供讀取複本升級在預先檢查時失敗，失敗的僅供讀取複本上的複寫會中斷，而且僅供讀取複本會處於終止狀態。刪除僅供讀取複本並根據升級的主要資料庫執行個體重新建立新的僅供讀取複本。

解決預先檢查日誌中列出的所有問題，然後重新嘗試主要版本升級。以下是預先檢查日誌範例。

```
-----  
Upgrade could not be run on Wed Apr 4 18:30:52 2018  
-----  
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.  
Please take appropriate action on databases that have usage incompatible with the  
requested major engine version upgrade and try the upgrade again.  
  
* There are uncommitted prepared transactions. Please commit or rollback all prepared  
transactions.* One or more role names start with 'pg_'. Rename all role names that  
start with 'pg_'.  
  
* The following issues in the database 'my"million$"db' need to be corrected before  
upgrading:** The ["line","reg*"] data types are used in user tables. Remove all  
usage of these data types.  
** The database name contains characters that are not supported by RDS for  
PostgreSQL. Rename the database.  
** The database has extensions installed that are not supported on the target  
database version. Drop the following extensions from your database: ["tsearch2"].  
  
* The following issues in the database 'mydb' need to be corrected before  
upgrading:** The database has views or materialized views that depend on  
'pg_stat_activity'. Drop the views.
```

12 如果升級資料庫時僅供讀取複本升級失敗，請解決此問題 - 失敗的僅供讀取複本處於 `incompatible-restore` 狀態，且資料庫上的複寫終止。刪除僅供讀取複本並根據升級的主要資料庫執行個體重新建立新的僅供讀取複本。

Note

Amazon RDS 不會升級多可用區域資料庫叢集的僅供讀取複本。如果您在異地同步備份資料庫叢集上執行主要版本升級，則其僅供讀取複本的複寫狀態會變更為已終止。

僅供讀取複本升級可能會因下列原因而失敗：

- 即使在等待時間之後，它也無法趕上主要資料庫執行個體。
- 它處於終止或不相容的生命週期狀態，例如儲存已滿、不相容還原等。
- 當主要資料庫執行個體升級開始時，僅供讀取複本上正在執行個別的次要版本升級。
- 僅供讀取複本使用的參數不相容。
- 僅供讀取複本無法與主要資料庫執行個體通訊，以同步處理資料的資料夾。

13. 升級生產資料庫 - 如果停用試轉主要版本升級成功，您應能放心升級生產資料庫。如需詳細資訊，請參閱 [手動升級引擎版本](#)。

14. 執行 ANALYZE 操作以重新整理 pg_statistic 資料表。您應為所有 PostgreSQL 資料庫上的每個資料庫執行此操作。主要版本升級期間不會傳輸最佳化工具統計數字，因此您需要重新產生所有統計數字以避免效能問題。在沒有任何參數的情況下執行命令，以產生目前資料庫中所有一般資料表的統計數字，如下所示：

```
ANALYZE VERBOSE;
```

您可選用 VERBOSE 旗標，但使用時會顯示進度。如需詳細資訊，請參閱 PostgreSQL 說明文件中的 [ANALYZE](#)。

Note

升級後在系統上執行 ANALYZE，以避免效能問題。

主要版本升級完成後，建議您執行以下動作：

- PostgreSQL 升級並不會升級任何 PostgreSQL 延伸套件。若要升級延伸模組，請參閱 [升級 PostgreSQL 延伸](#)。
- 或者，使用 Amazon RDS 來檢視 pg_upgrade 公用程式產生的兩個日誌。這兩個記錄是 pg_upgrade_internal.log 和 pg_upgrade_server.log。Amazon RDS 會將時間戳記附加至這些日誌的檔案名稱。您可以如同檢視任何其他日誌一般檢視這些日誌。如需詳細資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。

您也可以將升級日誌上傳到 Amazon CloudWatch 日誌。如需詳細資訊，請參閱 [將 PostgreSQL 日誌發佈到 Amazon 日誌 CloudWatch](#)。

- 為了確認所有作業皆符合預期，請在升級後的資料庫上，以類似的工作負載測試您的應用程式。確認升級之後，您可以刪除該測試執行個體。

PostgreSQL 的自動次要版本升級

建立或修改資料庫執行個體或多可用區域資料庫叢集時，如果啟用了自動次要版本升級選項，則可讓資料庫自動升級。

針對每個 RDS for PostgreSQL 主要版本，RDS 會將一個次要版本指定為自動升級版本。次要版本經過測試，並由 Amazon RDS 核准之後，在您的維護時段期間，會自動發生次要版本升級。RDS 不會自動將較新發行的次要版本設定為自動升級版本。在 RDS 指派較新的自動升級版本之前，會考慮數個準則，例如下列：

- 已知安全性問題
- PostgreSQL 社群版本中的錯誤
- 從發行次要版本以來的整體機群穩定性

您可以使用下列 AWS CLI 命令來判斷特定中指定 PostgreSQL 次要版本的目前自動次要升級目標版本。AWS 區域

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
--engine postgres \  
--engine-version minor-version \  
--region region \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output text
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
--engine postgres ^  
--engine-version minor-version ^  
--region region ^  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^  
--output text
```

例如，下列 AWS CLI 命令會決定美國東部 (俄亥俄州) (俄亥俄州) (us-east-2) AWS 區域 PostgreSQL 次要版本 12.13 的自動次要升級目標。

對於LinuxmacOS、或Unix：

```
aws rds describe-db-engine-versions \  
--engine postgres \  
--engine-version minor-version \  
--region region \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output text
```



```
--engine-version 12.13 \  
--region us-east-2 \  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \  
--output table
```

在 Windows 中：

```
aws rds describe-db-engine-versions ^  
--engine postgres ^  
--engine-version 12.13 ^  
--region us-east-2 ^  
--query "DBEngineVersions[*].ValidUpgradeTarget[*].  
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^  
--output table
```

輸出類似如下。

```
-----  
| DescribeDBEngineVersions |  
+-----+-----+  
| AutoUpgrade | EngineVersion |  
+-----+-----+  
| True      | 12.14      |  
| False      | 12.15      |  
| False      | 13.9       |  
| False      | 13.10      |  
| False      | 13.11      |  
| False      | 14.6       |  
+-----+-----+
```

在此範例中，PostgreSQL 12.14 版的此 AutoUpgrade 值是 True。因此，自動次要升級目標是 PostgreSQL 12.14 版，並已在輸出中反白顯示該版本。

如果符合下列準則，PostgreSQL 資料庫會在您的維護時段期間自動升級：

- 資料庫的自動次要版本升級選項已啟用。
- 資料庫執行的次要資料庫引擎版本小於目前的自動升級次要版本。

如需詳細資訊，請參閱 [自動升級次要引擎版本](#)。

Note

PostgreSQL 升級並不會升級 PostgreSQL 延伸套件。若要升級延伸模組，請參閱 [升級 PostgreSQL 延伸](#)。

升級 PostgreSQL 延伸

PostgreSQL 引擎升級並不會升級大多數 PostgreSQL 延伸套件。若要在版本升級後更新延伸套件，請使用 ALTER EXTENSION UPDATE 命令。

Note

如需如何更新 PostGIS 擴充功能的相關資訊，請參閱 [使用 PostGIS 擴充功能管理空間資料 \(步驟 6：升級 PostGIS 擴充功能\)](#)。

若要更新 pg_repack 延伸模組，請捨棄該延伸模組，然後在升級的資料庫中建立新版本。如需詳細資訊，請參閱 pg_repack 說明文件中的 [pg_repack 安裝](#)。

若要升級擴充功能，請使用下列命令。

```
ALTER EXTENSION extension_name UPDATE TO 'new_version';
```

如需 PostgreSQL 延伸套件支援版本的清單，請參閱 [支援的 PostgreSQL 擴充功能版本](#)。

若要列出目前已安裝的擴充功能，請在下列命令中使用 PostgreSQL [pg_extension](#) 目錄。

```
SELECT * FROM pg_extension;
```

若要檢視您的安裝可用的特定擴充功能版本的清單，請在下列命令中使用 PostgreSQL [pg_available_extension_versions](#) 檢視。

```
SELECT * FROM pg_available_extension_versions;
```

升級 PostgreSQL 資料庫快照引擎版本

利用 Amazon RDS，您可以為您的 PostgreSQL 資料庫執行個體建立儲存體磁碟區資料庫快照。建立資料庫快照時，該快照會基於您的 Amazon RDS 執行個體使用的引擎版本。除了升級您的資料庫執行個體的資料庫引擎版本，您也可以升級您的資料庫快照的引擎版本。

還原升級至新引擎版本的資料庫快照之後，務必測試升級已成功。如需主要版本升級的詳細資訊，請參閱[升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。若要了解如何還原資料庫快照，請參閱[從資料庫快照還原](#)。

您可以升級已加密或未加密的手動資料庫快照。

如需可用於升級資料庫快照的引擎版本清單，請參閱[針對 Amazon RDS 升級 PostgreSQL 資料庫引擎](#)。

Note

您不可以升級在自動備份程序期間建立的自動資料庫快照。

主控台

升級資料庫快照

1. 登入 AWS Management Console，開啟位於 <https://console.aws.amazon.com/rds/> 的 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇您要升級的快照。
4. 針對 Actions (動作)，選擇 Upgrade snapshot (升級快照)。Upgrade snapshot (升級快照) 頁面隨即出現。
5. 選擇要升級到的 New engine version (新引擎版本)。
6. 選擇 Save changes (儲存變更) 以升級快照。

升級程序期間，此資料庫快照的所有快照動作會停用。同時，資料庫快照狀態會從 available (可用) 變更為 upgrading (升級中)，然後在完成時變更為 active (作用中)。如果資料庫快照因為快照損毀問題而無法升級，狀態會變更為 unavailable (無法使用)。您無法從此狀態復原快照。

Note

如果資料庫升級失敗，快照會轉返至原始版本的原始狀態。

AWS CLI

若要將資料庫快照集升級為新的資料庫引擎版本，請使用AWS CLI [modify-db-snapshot](#)指令。

參數

- `--db-snapshot-identifier` – 要升級的資料庫快照識別符。識別符必須是唯一的 Amazon Resource Name (ARN)。如需更多詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。
- `--engine-version` – 資料庫快照會以此引擎版本為目標進行升級。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier my_db_snapshot \  
  --engine-version new_version
```

在Windows中：

```
aws rds modify-db-snapshot ^  
  --db-snapshot-identifier my_db_snapshot ^  
  --engine-version new_version
```

RDS API

若要將資料庫快照升級為新的資料庫引擎版本，請呼叫 Amazon RDS API [ModifyDBSnapshot](#) 操作。

- `DBSnapshotIdentifier` – 要升級的資料庫快照識別符。識別符必須是唯一的 Amazon Resource Name (ARN)。如需更多詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。
- `EngineVersion` – 資料庫快照會以此引擎版本為目標進行升級。

使用 Amazon RDS for PostgreSQL 的僅供讀取複本

您可以將僅供讀取複本新增至執行個體，以擴展 Amazon RDS for PostgreSQL 資料庫執行個體的讀取。與其他 Amazon RDS 資料庫引擎一樣，RDS for PostgreSQL 使用 PostgreSQL 的原生複寫機制，讓僅供讀取複本隨著來源資料庫的變更保持在最新狀態。如需僅供讀取複本與 Amazon RDS 的一般資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

有關如何利用 RDS for PostgreSQL 處理僅供讀取複本，以下提供具體資訊。

僅供讀取複本上的邏輯解碼

RDS 支援使用 PostgreSQL 從待機型進行邏輯複寫。這可讓您從唯讀待命模式建立邏輯解碼，以減少主要資料庫執行個體的負載。您可以為需要跨多個系統同步處理資料的應用程式達到更高的可用性。此功能可提升資料倉儲和資料分析的效能。

此外，指定待命狀態上的複寫插槽會持續將該待命提升至主要的備用狀態。這表示，在主要資料庫執行個體容錯移轉或將待命提升為新的主要執行個體時，複寫插槽將持續存在，而前一個待命訂閱者也不會受到影響。

若要在僅供讀取複本上建立邏輯解碼

1. 開啟邏輯複寫 — 若要在待命時建立邏輯解碼，您必須在來源資料庫執行個體及其實體複本上開啟邏輯複寫。如需詳細資訊，請參閱 [使用 PostgreSQL 的僅供讀取複本組態](#)。
 - 若要為新建立的 RDS for PostgreSQL 資料庫執行個體開啟邏輯複寫，請建立新的資料庫自訂參數群組，並將靜態參數 `rds.logical_replication` 設定為 1。然後，將此資料庫參數群組與來源資料庫執行個體及其實體僅供讀取複本建立關聯。如需詳細資訊，請參閱 [將資料庫參數群組與資料庫執行個體建立關聯](#)。
 - 若要為現有 RDS for PostgreSQL 資料庫執行個體開啟邏輯複寫，請將來源資料庫執行個體的資料庫自訂參數群組及其實體僅供讀取複本修改為 `rds.logical_replication 1`。如需詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

Note

您必須重新啟動資料庫執行個體才能套用這些參數變更。

您可以使用下列查詢來驗證來源資料庫執行個體wal_level及rds.logical_replication其實體僅供讀取複本的和值。

```
Postgres=>SELECT name,setting FROM pg_settings WHERE name IN
('wal_level','rds.logical_replication');
```

```

name                | setting
-----+-----
rds.logical_replication | on
wal_level            | logical
(2 rows)
```

2. 在來源資料庫中建立表格 — Connect 至來源資料庫執行個體中的資料庫。如需詳細資訊，請參閱 [連線至執行 PostgreSQL 資料庫引擎的資料庫執行個體](#)。

使用下列查詢在來源資料庫中建立表格並插入值：

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

```
Postgres=>INSERT INTO LR_test VALUES (generate_series(1,10000));
INSERT 0 10000
```

3. 為來源資料表建立發行集 — 使用下列查詢為來源資料庫執行個體上的表格建立發行集。

```
Postgres=>CREATE PUBLICATION testpub FOR TABLE LR_test;
CREATE PUBLICATION
```

使用 SELECT 查詢來驗證在來源資料庫執行個體和實體僅供讀取複本執行個體上建立的發行集詳細資訊。

```
Postgres=>SELECT * from pg_publication;
```

```

oid      | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
pubtruncate | pubviaroot
-----+-----+-----+-----+-----+-----+-----+-----
+-----+-----
```

```

16429 | testpub | 16413 | f | t | t | t | t
      | f
(1 row)

```

4. 從邏輯複本執行個體建立訂閱 — 建立另一個 RDS for PostgreSQL 資料庫執行個體做為邏輯複本執行個體。請確定已正確設定 VPC，以確保此邏輯複本執行個體可存取實體僅供讀取複本執行個體。如需詳細資訊，請參閱 [Amazon VPC](#) 和 [Amazon RDS](#)。如果來源資料庫執行個體閒置，可能會發生連線問題，且主要執行個體不會將資料傳送至待命狀態。

```

Postgres=>CREATE SUBSCRIPTION testsub CONNECTION 'host=Physical replica host name
port=port
          dbname=source_db_name user=user password=password
PUBLICATION testpub;
NOTICE: created replication slot "testsub" on publisher
CREATE SUBSCRIPTION

```

```

Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE

```

使用 SELECT 查詢來驗證邏輯複本執行個體上的訂閱詳細資訊。

```

Postgres=>SELECT oid,subname,subenabled,subslotname,subpublications FROM
pg_subscription;

oid      | subname | subenabled | subslotname | subpublications
-----+-----+-----+-----+-----
 16429 | testsub | t          | testsub     | {testpub}
(1 row)
postgres=> select count(*) from LR_test;
count
-----
 10000
(1 row)

```

5. 檢查邏輯複寫插槽狀態 — 您只能在來源資料庫執行個體上看到實體複寫插槽。

```

Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn

```



```
-----+-----+-----
rds_us_west_2_db_dhqfsmo5wbbjqrn3m6b6ivdhu4 | physical |
(1 row)
```

不過，在僅供讀取複本執行個體上，您可以看到邏輯複寫插槽，而且 `confirmed_flush_lsn` 值會隨著應用程式主動使用邏輯變更而變更。

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
```

```
slot_name | slot_type | confirmed_flush_lsn
-----+-----+-----
testsub   | logical   | 0/500002F0
(1 row)
```

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
pg_replication_slots;
```

```
slot_name | slot_type | confirmed_flush_lsn
-----+-----+-----
testsub   | logical   | 0/5413F5C0
(1 row)
```

使用 PostgreSQL 的僅供讀取複本限制

下列為 PostgreSQL 僅供讀取複本的限制：

Note

適用於執行 PostgreSQL 版本 12 及更早版本的 RDS 異地同步備 PostgreSQL 和單一可用區資料庫執行個體的僅供讀取複本，可在 60 到 90 天維護期間自動重新啟動以套用密碼輪替。

- PostgreSQL 僅供讀取複本皆為唯讀。雖然僅供讀取複本不是可寫入資料庫執行個體，但您可將其提升為獨立的 RDS for PostgreSQL 資料庫執行個體。但此程序是不可逆的。
- 若 RDS for PostgreSQL 資料庫執行個體執行早於 14.1 的 PostgreSQL 版本，則無法從另一個僅供讀取複本建立僅供讀取複本。RDS for PostgreSQL 僅支援 RDS for PostgreSQL 14.1 版和更新版本

上的階層式僅供讀取複本。如需詳細資訊，請參閱 [使用具 RDS for PostgreSQL 的階層式僅供讀取複本](#)。

- 若您提升 PostgreSQL 僅供讀取複本，則會成為可寫入資料庫執行個體。其會停止從來源資料庫執行個體接收預寫日誌 (WAL) 檔案，且不再是個唯讀執行個體。您可從提升的資料庫執行個體建立新的僅供讀取複本，如同對任何 RDS for PostgreSQL 資料庫執行個體所做的一樣。如需詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。
- 如果您從複寫鏈中升級 PostgreSQL 僅供讀取複本 (一系列串聯僅供讀取複本)，則任何現有的下游僅供讀取複本都會繼續自動從升級的執行個體接收 WAL 檔案。如需詳細資訊，請參閱 [使用具 RDS for PostgreSQL 的階層式僅供讀取複本](#)。
- 如果來源資料庫執行個體上未發生使用者交易，則 PostgreSQL 僅供讀取複本會回報最多五分鐘的複寫延遲。複本滯後的計算方式為 $\text{currentTime} - \text{lastCommittedTransactionTimestamp}$ ，亦即未處理任何交易時，複本滯後的值會增加一段時間，直到預寫日誌 (WAL) 區段切換為止。根據預設，RDS for PostgreSQL 每 5 分鐘切換一次 WAL 區段，這會導致交易記錄和報告的延遲減少。
- 您無法為早於 14.1 的 RDS for PostgreSQL 版本開啟 PostgreSQL 僅供讀取複本的自動備份。RDS for PostgreSQL 14.1 及更新版本僅支援僅供讀取複本的自動備份。對於 RDS for PostgreSQL 13 及更早版本，若您想要備份僅供讀取複本，則請從僅供讀取複本建立快照。
- 僅供讀取複本不支援 Point-in-time 復原 (PITR)。您僅可使用具主 (寫入器) 執行個體的 PITR，而非僅供讀取複本。如需進一步了解，請參閱 [將資料庫執行個體還原至指定的時間](#)。

使用 PostgreSQL 的僅供讀取複本組態

RDS for PostgreSQL 會使用 PostgreSQL 原生串流複寫來建立來源資料庫執行個體的唯讀複本。此僅供讀取複本資料庫執行個體是個非同步建立之來源資料庫執行個體的實體複寫。此由將預寫日誌 (WAL) 資料從來源資料庫執行個體傳輸至僅供讀取複本的特殊連線所建立。如需詳細資訊，請參閱 PostgreSQL 文件中的 [串流複寫](#)。

PostgreSQL 會將資料庫變更非同步串流至此安全連接，如同其在來源資料庫執行個體上所進行般。您可透過將 `ssl` 參數設定為 1，對用戶端應用程式至來源資料庫執行個體或任何僅供讀取複本的通訊進行加密。如需詳細資訊，請參閱 [將 SSL 與 PostgreSQL 資料庫執行個體搭配使用](#)。

PostgreSQL 會使用複寫角色來執行串流複寫。角色有優先權，但無法用來修改任何資料。PostgreSQL 使用單一程序來處理複寫。

您可在不影響來源資料庫執行個體的作業或使用者的狀況下建立 PostgreSQL 僅供讀取複本。Amazon RDS 會為來源資料庫執行個體和僅供讀取複本設定必要的參數和權限，而不會影響服務。將會拍攝來源資料庫執行個體的快照，而此快照會用來建立僅供讀取複本。如於未來的某個時間點刪除僅供讀取複本，則不會發生停機。

您可以從相同區域內的一個來源資料庫執行個體建立至多 15 個僅供讀取複本。從 RDS for PostgreSQL 14.1 開始，您還可從來源資料庫執行個體以鏈結 (階層式) 的形式建立最多三個層級的僅供讀取複本。如需詳細資訊，請參閱 [使用具 RDS for PostgreSQL 的階層式僅供讀取複本](#)。在所有情況下，來源資料庫執行個體皆需設定自動備份。您可將資料庫執行個體上的備份保留期設定為 0 以外的任何值以進行此作業。如需詳細資訊，請參閱 [建立僅供讀取複本](#)。

您可以在與來源資料庫執行個體相同 AWS 區域的位置，為 RDS for PostgreSQL 資料庫執行個體建立僅供讀取複本。此稱為區域內複寫。您也可以在不同 AWS 區域於來源資料庫執行個體中建立僅供讀取複本。此稱為跨區域複寫。如需設定跨區域僅供讀取複本的相關資訊，請參閱 [以不同的方式建立僅供讀取複本 AWS 區域](#)。支援「區域內」和「跨區域」複寫程序的各種機制會依 RDS for PostgreSQL 版本而略有不同，此說明於 [串流複寫如何用於不同的 RDS for PostgreSQL 版本](#) 中。

若希望複寫作業順利運作，每個僅供讀取複本具備的運算和儲存資源數量應與來源資料庫執行個體相同。若您擴展來源資料庫執行個體，也請務必擴展僅供讀取複本。

若 Amazon RDS 阻止僅供讀取複本啟動，則期將會覆寫僅供讀取複本上任何不相容的參數。舉例來說，假設在資料庫執行個體的 `max_connections` 參數值高於僅供讀取複本上的值。在此情況下，Amazon RDS 會更新僅供讀取複本上的參數為與來源資料庫執行個體上相同的值。

RDS for PostgreSQL 僅供讀取複本可存取外部資料庫，這些資料庫可經由來源資料庫執行個體上的外部資料包裝函式 (FDW) 取得。例如，假設 RDS for PostgreSQL 資料庫執行個體使用 `mysql_fdw` 包裝函式來存取 RDS for MySQL 的資料。若是如此，您的僅供讀取複本也可存取該資料。其他受支援的 FDW 包括 `oracle_fdw`、`postgres_fdw` 和 `tds_fdw`。如需詳細資訊，請參閱 [使用 Amazon RDS for PostgreSQL 支援的外部資料包裝函式](#)。

使用具多可用區域組態的 RDS for PostgreSQL 僅供讀取複本

您可以從單一可用區域或多可用區域的資料庫執行個體建立僅供讀取複本。您可使用異地同步備份部署，利用備用複本來改善重要資料的耐用性和可用性。若來源資料庫進行容錯移轉，則備用複本是個可承擔工作負載的專用僅供讀取複本。您無法使用備用複本來提供讀取流量。但是，您可從高流量的多可用區域資料庫執行個體建立僅供讀取複本，藉此卸載唯讀查詢。若要進一步了解異地同步備份部署，請參閱 [多可用區域資料庫執行個體部署](#)。

若異地同步備份部署的來源資料庫執行個體容錯移轉至備用複本，則相關聯的僅供讀取複本都會切換為使用備用複本 (現為主要複本) 作為其複寫來源。僅供讀取複本可能需要重新啟動，視 RDS for PostgreSQL 版本而定，如下所示：

- PostgreSQL 13 及更新版本 – 無須重新啟動。僅供讀取複本將會自動與新的主要複本同步。但在某些狀況下，您的用戶端應用程式可能會快取僅供讀取複本的網域名稱服務 (DNS) 詳細資訊。如果是

這樣，請將 time-to-live (TTL) 值設定為小於 30 秒。這麼做可以防止僅供讀取複本保留過時的 IP 地址 (如此，可防止其與新的主要複本同步)。如需進一步了解有關此及其他最佳實務的詳細資訊，請參閱 [Amazon RDS 基本操作準則](#)。

- PostgreSQL 12 及所有早期版本 – 僅供讀取複本在容錯移轉至備用複本後會自動重新啟動，因為備用 (現為主要) 具有不同的 IP 地址和不同的執行個體名稱。重新啟動僅供讀取複本與新的主要複本同步。

如需進一步了解容錯移轉，請參閱 [Amazon RDS 的容錯移轉程序](#)。若要進一步了解僅供讀取複本如何在異地同步備份部署中運作，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

如要提供僅供讀取複本的容錯移轉支援，您可將僅供讀取複本建立為多可用區域資料庫執行個體，則 Amazon RDS 會在另一個可用區域 (AZ) 中建立您複本的待命複本。建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。

使用具 RDS for PostgreSQL 的階層式僅供讀取複本

從 14.1 版開始，RDS for PostgreSQL 支援階層式僅供讀取複本。利用階層式僅供讀取複本，您可以擴展讀取，無須增加來源 RDS for PostgreSQL 資料庫執行個體的額外負荷。來源資料庫執行個體不會將 WAL 日誌的更新傳送至每個僅供讀取複本。反之，階層式系列中的每個僅供讀取複本都會將 WAL 日誌更新傳送至系列中的下一個僅供讀取複本。此會減少來源資料庫執行個體的負擔。

利用階層式僅供讀取複本，您的 RDS for PostgreSQL 資料庫執行個體會將 WAL 資料傳送至鏈結中的第一個僅供讀取複本。之後，該僅供讀取複本會將 WAL 資料傳送至鏈結中的第二個複本，依此類推。最終結果是鏈結中的所有僅供讀取複本皆具有來自 RDS for PostgreSQL 資料庫執行個體的變更，但並無僅在來源資料庫執行個體上的額外負荷。

您可從來源 RDS for PostgreSQL 資料庫執行個體的鏈結中建立一系列最多三個僅供讀取複本。例如，假設您有一個 RDS for PostgreSQL 14.1 資料庫執行個體 `rpg-db-main`。您可以執行下列作業：

- 從 `rpg-db-main` 開始，建立鏈結中的第一個僅供讀取複本 `read-replica-1`。
- 接下來，從 `read-replica-1`，建立鏈結中的下一個僅供讀取複本 `read-replica-2`。
- 最後，從 `read-replica-2`，建立鏈結中的第三個僅供讀取複本 `read-replica-3`。

除了 `rpg-db-main` 系列中的第三個階層式僅供讀取複本之外，您無法建立另一個僅供讀取複本。從 RDS for PostgreSQL 來源資料庫執行個體至一系列階層式僅供讀取複本尾端的完整執行個體系列最多可包含四個資料庫執行個體。

若要使階層式僅供讀取複本正常運作，請開啟 RDS for PostgreSQL 上的自動備份。首先建立僅供讀取複本，然後開啟 RDS for PostgreSQL 資料庫執行個體上的自動備份。此程序與其他 Amazon RDS 資料庫引擎相同。如需詳細資訊，請參閱 [建立僅供讀取複本](#)。

與任何僅供讀取複本一樣，您可提升作為階層式一部分的僅供讀取複本。從僅供讀取複本鏈結中提升僅供讀取複本將會從鏈結中移除該僅供讀取複本。例如，假設您想要將部分工作負載從 rpg-db-main 資料庫執行個體移至新的執行個體，僅供會計部門使用。假設範例中的三個僅供讀取複本鏈結，您決定提升 read-replica-2。該鏈或受到下列影響：

- 提升 read-replica-2 會將其從複寫鏈結中移除。
 - 其現在是一個完整的讀取/寫入資料庫執行個體。
 - 這會持續複寫至 read-replica-3，就像在提升之前所做的一樣。
- 您的 rpg-db-main 會持續複寫至 read-replica-1。

如需提升僅供讀取複本的相關詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。

Note

對於階層式僅供讀取複本，RDS for PostgreSQL 在第一個複寫層支援每個來源資料庫執行個體 15 個僅供讀取複本，而在第二和第三複寫層支援每個來源資料庫執行個體 5 個僅供讀取複本。

使用 RDS 建立跨區域串聯僅供讀取複本

RDS 版支援跨區域串聯僅供讀取複本。您可以從來源資料庫執行個體建立跨區域複本，然後從其建立相同區域複本。您也可以從來源資料庫執行個體建立相同區域複本，然後從該執行個體建立跨區域複本。

建立跨區域複本，然後建立相同區域複本

您可以使用 14.1 或更 PostgreSQL 版本的 RDS 資料庫執行個體 rpg-db-main，執行下列動作：

1. 從 rpg-db-main (US-EAST-1) 開始，建立鏈結中的第一個跨區域僅供讀取複本 read-replica-1 (US-WEST-2)。
2. 使用第一個跨區域 read-replica-1 (US-WEST-2)，在鏈結中建立第二個僅供讀取複本 read-replica-2 (US-WEST-2)。

3. 使用 `read-replica-2`，建立鏈結中的第三個僅供讀取複本 `read-replica-3 (US-WEST-2)`。

建立相同區域複本，然後建立跨區域複本

您可以使用 14.1 或更 PostgreSQL 版本的 RDS 資料庫執行個體 `rpg-db-main`，執行下列動作：

1. 從 `rpg-db-main (US-EAST-1)` 開始，建立鏈結中的第一個僅供讀取複本 `read-replica-1 (US-EAST-1)`。
2. 使用 `read-replica-1 (US-EAST-1)`，建立鏈結中的第一個跨區域僅供讀取複本 `read-replica-2 (US-WEST-2)`。
3. 使用 `read-replica-2 (US-WEST-2)`，建立鏈結中的第三個僅供讀取複本 `read-replica-3 (US-WEST-2)`。

建立跨區域僅供讀取複本的限制

- 跨區域串聯式資料庫複本鏈最多可跨兩個區域，最多四個層級。四個層級包括資料庫來源和三個僅供讀取複本。

使用串聯僅供讀取複本的優點

- 改善讀取延展性 — 透過將讀取查詢分散到多個複本，串聯式複寫有助於平衡負載。這可以減少寫入器資料庫的壓力，進而提升效能，尤其是在需要大量讀取的應用程式中。
- 地理分佈 — 串聯複本可以位於不同的地理位置。如此可減少遠離主要資料庫的使用者延遲，並提供本機僅供讀取複本，進而提升效能和使用者的體驗。
- 高可用性和嚴重損壞復原 — 萬一主要伺服器故障，可以將複本升級為主要伺服器，以確保連續性。串聯式複寫可提供多層容錯移轉選項，進一步增強這一點，改善系統的整體彈性。
- 彈性與模組化成長 — 隨著系統的成長，可以在不同層級新增新的複本，而不需要重新設定主要資料庫。這種模組化方法可讓複寫設定達到可擴充且可管理的成長。

如需有關使用複寫之優點的詳細資訊，請參閱[關於 Cloud SQL 中的複寫](#)。

使用跨區域僅供讀取複本的最佳作法

- 升級複本之前，請先建立其他複本。這將節省時間，並提供有效的工作負載處理。

串流複寫如何用於不同的 RDS for PostgreSQL 版本

如 [使用 PostgreSQL 的僅供讀取複本組態](#) 中所討論，RDS for PostgreSQL 會使用 PostgreSQL 的原生串流複寫協定，從來源資料庫執行個體傳送 WAL 資料。此將來源 WAL 資料傳送至區域內和跨區域僅供讀取複本的僅供讀取複本。在 9.4 版中，PostgreSQL 引入了實體複寫槽，作為複寫程序的支援機制。

實體複寫槽會防止來源資料庫執行個體在所有僅供讀取複本使用 WAL 資料之前移除 WAL 資料。每個僅供讀取複本在來源資料庫執行個體上都有自己的實體槽。該插槽會追蹤複本可能需要的最舊 WAL (按邏輯序號，LSN)。在所有插槽和資料庫連接皆超出給定的 WAL (LSN) 之後，該 LSN 會成為下一個檢查點移除的候選項。

Amazon RDS 使用 Amazon S3 來封存 WAL 資料。對於區域內僅供讀取複本，您可於需要時使用此封存資料復原僅供讀取複本。若來源資料庫和僅供讀取複本之間的連接因任何原因而遭中斷，您可能會這麼做的範例。

於下表中，您可以找到 PostgreSQL 版本之間的差異，及 RDS for PostgreSQL 所使用之區域內和跨區域的支援機制摘要。

區域內

跨區域

PostgreSQL 14.1 and higher versions

- 複寫槽
- Amazon S3 封存

- 複寫槽

PostgreSQL 13 and lower versions

- Amazon S3 封存

- 複寫槽

如需詳細資訊，請參閱 [監控和調校複寫程序](#)。

了解控制 PostgreSQL 複寫的參數

下列參數影響複寫程序，並決定僅供讀取複本與來源資料庫執行個體保持於最新狀態的程度：

max_wal_senders

`max_wal_senders` 參數指定來源資料庫執行個體可透過串流複寫協定同時支援的最大連接數。RDS for PostgreSQL 13 版及更新版本的預設值為 20。此參數應設定為略高於僅供讀取複本的實際數量。若此參數對僅供讀取複本數量設定過低，則複寫會停止。

如需詳細資訊，請參閱 PostgreSQL 文件的 [max_wal_senders](#)。

wal_keep_segments

`wal_keep_segments` 參數指定來源資料庫執行個體保存於 `pg_wal` 目錄中預寫日誌 (WAL) 檔案的數量。預設設定為 32。

若 `wal_keep_segments` 未對您的部署設定足夠大的值，則僅供讀取複本可能遠遠落後於串流複寫停止。若該狀況發生，Amazon RDS 會產生複寫錯誤，並開始在僅供讀取複本上進行復原。透過重複播放來自 Amazon S3 的來源資料庫執行個體的封存 WAL 資料來進行此作業。此還原程序會繼續進行，直到僅供讀取複本跟上進度，以繼續串流複寫。您可於 [範例：僅供讀取複本如何從複寫中斷復原](#) 中看到 PostgreSQL 日誌所擷取的這個程序。

Note

於 PostgreSQL 13 版中，`wal_keep_segments` 參數稱為 `wal_keep_size`。其用途與 `wal_keep_segments` 相同，但其預設值以 MB (2048 MB) 為單位，而非檔案數。如需詳細資訊，請參閱 PostgreSQL 文件中的 [wal_keep_segments](#) 和 [wal_keep_size](#)。

max_slot_wal_keep_size

`max_slot_wal_keep_size` 參數控制保留於 `pg_wal` 目錄中 RDS for PostgreSQL 資料庫執行個體的 WAL 資料量來處理插槽。此參數用於使用複寫插槽的組態。此參數的預設值為 -1，這表示來源資料庫執行個體上保留的 WAL 資料量並無限制。如需監控複寫槽的相關資訊，請參 [監控 RDS for PostgreSQL 資料庫執行個體的複寫槽](#)。

如需此參數的詳細資訊，請參閱 PostgreSQL 文件中的 [max_slot_wal_keep_size](#)。

當提供 WAL 資料給僅供讀取複本的串流中斷時，PostgreSQL 將切換為復原模式。它會使用來自 Amazon S3 的已封存 WAL 資料或使用與複寫插槽關聯的 WAL 資料來還原僅供讀取複本。此程序完成時，PostgreSQL 會重新建立串流複寫。

範例：僅供讀取複本如何從複寫中斷復原

於下列範例中，您可以找到示範僅供讀取複本復原程序的日誌詳細資訊。範例來自於執行 PostgreSQL 12.9 版的 RDS 資料庫執行個體，與來源資料庫相同 AWS 區域，因此不會使用複寫插槽。對於執行早於 14.1 版且具有區域內僅供讀取複本之 PostgreSQL 的其他 RDS for PostgreSQL 資料庫執行個體，復原程序是相同的。

當僅供讀取複本與來源資料庫執行個體失去連線時，Amazon RDS 會在日誌中將問題記錄為 **FATAL: could not receive data from WAL stream** 訊息與 **ERROR: requested WAL segment ... has already been removed**。如粗體行中所示，Amazon RDS 透過重複播放封存的 WAL 檔案來還原複本。

```
2014-11-07 19:01:10 UTC::@[23180]:DEBUG: switched WAL source from archive to stream
after failure
2014-11-07 19:01:10 UTC::@[11575]:LOG: started streaming WAL from primary at 1A/
D3000000 on timeline 1
2014-11-07 19:01:10 UTC::@[11575]:FATAL: could not receive data from WAL stream:
ERROR: requested WAL segment 000000010000001A000000D3 has already been removed
2014-11-07 19:01:10 UTC::@[23180]:DEBUG: could not restore file "00000002.history"
from archive: return code 0
2014-11-07 19:01:15 UTC::@[23180]:DEBUG: switched WAL source from stream to archive
after failure recovering 000000010000001A000000D3
2014-11-07 19:01:16 UTC::@[23180]:LOG: restored log file "000000010000001A000000D3"
from archive
```

當 Amazon RDS 在複本上重複播放足夠的封存 WAL 資料以補足空間，讓僅供讀取複本再次開始串流。恢復串流時，Amazon RDS 會將一個項目寫入日誌檔案中，類似下列內容。

```
2014-11-07 19:41:36 UTC::@[24714]:LOG:started streaming WAL from primary at 1B/
B6000000 on timeline 1
```

設定控制共用記憶體之參數

您設定的參數會決定用於追蹤交易 ID、鎖定和預備交易的共用記憶體大小。待命執行個體的共用記憶體結構必須等於或大於主要執行個體的共用記憶體結構。這樣可以確保前者在復原過程中不會耗盡共用記憶體。如果複本上的參數值小於主要複本上的參數值，Amazon RDS 將自動調整複本參數並重新啟動引擎。

受影響的參數包括：

- `max_connections`

- `max_worker_processes`
- `max_wal_senders`
- `max_prepared_transactions`
- `max_locks_per_transaction`

為避免 RDS 因記憶體不足而將複本重新開機，我們建議採取滾動式重新開機的方式將參數變更套用至每個複本。當您設定參數時，必須套用下列規則：

- 增加參數值：
 - 您應一律先增加所有僅供讀取複本的參數值，並執行所有複本的滾動式重新開機。然後再將參數變更套用至主要執行個體，並重新開機。
- 增加參數值：
 - 您應該先減少主要執行個體的參數值，並執行重新開機。然後再將參數變更套用至所有相關聯的僅供讀取複本，並執行滾動式重新開機。

監控和調校複寫程序

我們強烈建議您定期監控 RDS for PostgreSQL 資料庫執行個體和僅供讀取複本。您需要確保您的僅供讀取複本與來源資料庫執行個體上的變更保持一致。當複寫程序發生中斷時，Amazon RDS 會通透地復原您的僅供讀取複本。但，最好完全避免需要復原。使用複寫槽進行復原比使用 Amazon S3 封存更快，但任何復原程序皆會影響讀取效能。

如要確定僅供讀取複本與來源資料庫執行個體保持一致的程度，可執行下列作業：

- 檢查來源資料庫執行個體和複本間的 **ReplicaLag** 數量。複本延遲指的是讀取複本落後於其來源資料庫執行個體的時間量 (以秒為單位)。此指標會報告下列查詢結果。

```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS "ReplicaLag";
```

複本延遲可表示僅供讀取複本與來源資料庫執行個體保持一致的程度。這是來源資料庫執行個體與特定讀取執行個體之間的延遲量。複本延遲值較高可能表示來源資料庫執行個體與其僅供讀取複本所使用的資料庫執行個體類別或儲存體類型 (或兩者) 不相符。資料庫來源執行個體和所有僅供讀取複本的資料庫執行個體類別和儲存體類型應該相同。

複本延遲也可為儲存體類型連線問題的結果。您可以 CloudWatch 透過檢視 Amazon RDS ReplicaLag 指標來監控 Amazon 中的複寫延遲。如需進一步了解 ReplicaLag 和 Amazon RDS 的其他指標，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。

- 檢查 PostgreSQL 日誌，以取得可用來調整設定的資訊。PostgreSQL 日誌會在每個檢查點擷取回收的交易日誌檔案數量，如下列範例所示。

```
2014-11-07 19:59:35 UTC::@[26820]:LOG: checkpoint complete: wrote 376 buffers
(0.2%);
0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s,
total=35.703 s;
sync files=10, longest=0.013 s, average=0.001 s
```

您可以使用這些資訊來確定指定時段內會回收多少個交易檔案。隨後，您可視需要變更 wal_keep_segments 的設定。例如，假設 PostgreSQL 日誌在 checkpoint complete 中顯示 35 recycled，間隔為 5 分鐘。於本案例中，wal_keep_segments 預設值 32 不足以跟上串流活動的進度，因此您應提高此參數的值。

- 使用 Amazon CloudWatch 監控可預測複寫問題的指標。您可以使用 Amazon CloudWatch 檢查已收集的指標，而不是直接分析 PostgreSQL 日誌。例如，您可以檢查 TransactionLogsGeneration 指標值，來查看來源資料庫執行個體所產生的 WAL 資料量。在某些狀況下，資料庫執行個體上的工作負載可能會產生大量 WAL 資料。若是如此，您可能需要變更來源資料庫執行個體和僅供讀取複本的資料庫執行個體類別。使用具有高 (10 Gbps) 網路效能的執行個體類別可減少複本延遲。

監控 RDS for PostgreSQL 資料庫執行個體的複寫槽

RDS for PostgreSQL 的所有版本皆會對跨區域僅供讀取複本使用複寫槽。RDS for PostgreSQL 14.1 版和較新版本皆會對區域內僅供讀取複本使用複寫槽。區域內僅供讀取複本還會使用 Amazon S3 來封存 WAL 資料。換言之，若您的資料庫執行個體和僅供讀取複本執行 PostgreSQL 14.1 版或更新版本，則複寫槽和 Amazon S3 封存皆可用於復原僅供讀取複本。使用其複寫槽復原僅供讀取複本比從 Amazon S3 封存復原更快。因此，我們建議您監控複寫槽和及相關的指標。

您可檢視 RDS for PostgreSQL 資料庫執行個體上的複寫槽，方法是查詢 pg_replication_slots 檢視，如下所示。

```
postgres=> SELECT * FROM pg_replication_slots;
```

```

slot_name          | plugin | slot_type | datoid | database | temporary |
active | active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn |
wal_status | safe_wal_size | two_phase
-----+-----+-----+-----+-----+-----
+-----+-----+-----+-----+-----+-----
+-----+-----+-----+-----+-----+-----
rds_us_west_1_db_555555555 |      | physical |      |      | f      | t
|      13194 |      |      | 23/D8000060 |      | reserved |
|      | f
(1 row)

```

reserved 值的 wal_status 表示插槽持有的 WAL 資料量在 max_wal_size 參數的範圍內。換言之，複寫槽的大小合適。其他可能的狀態值如下：

- extended – 插槽超過 max_wal_size 設定，但會保留 WAL 資料。
- unreserved – 插槽不再具有所有必要的 WAL 資料。其中一些將於下一個檢查點移除。
- lost – 部分必要 WAL 資料已遭移除。該插槽不再可用。

只有在非負數時，才 wal_status 會 max_slot_wal_keep_size 看到的 unreserved 和 lost 狀態。

pg_replication_slots 檢視顯示了複寫槽的目前狀態。若要評估複寫插槽的效能，您可以使用 Amazon CloudWatch 並監控下列指標：

- **OldestReplicationSlotLag** – 列出延遲最多的插槽，也就是落後主要插槽最多的插槽。此延遲可與僅供讀取複本相關聯，但亦可與連線相關聯。
- **TransactionLogsDiskUsage** – 顯示 WAL 資料正在使用多少儲存空間。僅供讀取複本大幅延遲時，此指標的值可能會顯著提高。

若要進一步了解如何將 Amazon CloudWatch 及其指標用於 PostgreSQL 的資訊，請參閱 [使用 Amazon CloudWatch 監控 Amazon RDS 指標](#) 如需有關監視 RDS for PostgreSQL 資料庫執行個體上串流複寫的詳細資訊，請參閱 AWS 資料庫部落格上的 [Amazon RDS PostgreSQL 複寫的最佳實務](#)。

適用於 RDS 僅供讀取複本的疑難排解

接下來，您可以找到一些常見 RDS for PostgreSQL 僅供讀取複本問題的疑難排解想法。

終止導致僅供讀取複本延遲的查詢

在資料庫中長時間執行的交易狀態下處於使用中或閒置的作業事件可能會干擾 WAL 複寫程序，進而增加複寫延遲。因此，請務必使用 PostgreSQL `pg_stat_activity` 檢視來監視這些交易的執行階段。

在類似下列的主要執行個體上執行查詢，以尋找長時間執行之查詢的處理序 ID (PID)：

```
SELECT datname, pid, username, client_addr, backend_start,
xact_start, current_timestamp - xact_start AS xact_runtime, state,
backend_xmin FROM pg_stat_activity WHERE state='active';
```

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
xact_duration,*
FROM pg_stat_activity
WHERE state = 'idle in transaction'
AND xact_start is not null
ORDER BY 1 DESC;
```

識別查詢的 PID 之後，您可以選擇結束查詢。

在類似下列的主要執行個體上執行查詢，以終止長時間執行的查詢：

```
SELECT pg_terminate_backend(PID);
```

使用 Amazon RDS Optimized Reads 改善 RDS for PostgreSQL 的查詢效能

您可以使用 Amazon RDS Optimized Reads，為 RDS for PostgreSQL 實現更快的查詢處理。使用 RDS Optimized Reads 的 RDS for PostgreSQL 資料庫執行個體或多可用區域資料庫叢集，相較於不使用的情况，查詢處理速度最多可提高 50%。

主題

- [PostgreSQL 中 RDS Optimized Reads 的概觀](#)
- [RDS Optimized Reads 的使用案例](#)
- [RDS Optimized Reads 的最佳實務](#)
- [使用 RDS Optimized Reads](#)
- [監控使用 RDS Optimized Reads 的資料庫執行個體](#)
- [PostgreSQL 中 RDS Optimized Reads 的限制](#)

PostgreSQL 中 RDS Optimized Reads 的概觀

根據預設，Optimized Reads 可在 RDS for PostgreSQL 15.2 及更高版本、14.7 及更高版本以及 13.10 及更新版本上使用。

當您使用已開啟 RDS Optimized Reads 的 RDS for PostgreSQL 資料庫執行個體或多可用區域資料庫叢集時，其可透過使用本機非揮發性記憶體儲存裝置 (NVMe) 固態硬碟 (SSD) 區塊層級儲存體，將查詢效能最多提高 50%。您可以將 PostgreSQL 產生的暫存資料表放置在本機儲存體上，藉此減少透過網路傳送至 Elastic Block Storage (EBS) 的流量，以達到更快的查詢處理速度。

在 PostgreSQL 中，臨時物件被指派到暫時命名空間，該命名空間會在工作階段結束時自動丟棄。丟棄時的暫時命名空間會移除任何與工作階段相關的物件，包括結構描述限定物件，例如資料表、函數、運算子，甚至延伸。

在 RDS for PostgreSQL 中，會針對儲存臨時物件的暫存工作區域設定 `temp_tablespace` 參數。

下列查詢會傳回資料表空間的名稱及其位置。

```
postgres=> show temp_tablespace;
temp_tablespace
-----
rds_temp_tablespace
(1 row)
```

`rds_temp_tablespace` 是 RDS 設定的資料表空間，指向 NVMe 本機儲存體。您隨時可以使用 AWS Management Console 修改 Parameter group 中的這個參數為指向 `rds_temp_tablespace` 以外的任何資料表空間，來切換回 Amazon EBS 儲存體。如需詳細相關資訊，請參閱 [修改資料庫參數群組中的參數](#)。您也可以使用 SET 命令，在工作階段層級使用 SET 命令將 `temp_tablespaces` 參數的值修改為 `pg_default`。修改參數會將暫存工作區域重新導向至 Amazon EBS。當 RDS 執行個體或叢集的本機儲存空間不足以執行特定 SQL 操作時，切換回 Amazon EBS 會有所幫助。

```
postgres=> SET temp_tablespaces TO 'pg_default';
SET
```

```
postgres=> show temp_tablespaces;

temp_tablespaces
-----
pg_default
```

RDS Optimized Reads 的使用案例

下列是可從 Optimized Reads 中受益的一些使用案例：

- 分析查詢，包括一般資料表表達式 (CTE)、衍生資料表和群組操作。
- 處理應用程式未最佳化查詢的僅供讀取複本。
- 具有複雜操作的隨需或動態報告查詢 (例如 GROUP BY 和 ORDER BY)，無法始終使用適當的索引。
- 使用內部暫存資料表的其他工作負載
- CREATE INDEX 或用於排序的 REINDEX 操作。

RDS Optimized Reads 的最佳實務

請使用 RDS Optimized Reads 的下列最佳實務：

- 針對唯讀查詢新增重試邏輯，以防這些查詢在執行期間由於執行個體儲存體已滿而失敗。
- 使用 CloudWatch 指標監視執行個體存放區上的可用儲存空間 `FreeLocalStorage`。如果執行個體儲存體由於資料庫執行個體或多可用區域資料庫叢集的工作負載而達到其限制，請對其修改以使用較大的資料庫執行個體類別。

使用 RDS Optimized Reads

當您在單一可用區域資料庫執行個體部署、多可用區域資料庫執行個體部署或多可用區域資料庫叢集部署中，佈建具有 NVMe 基礎其中一個資料庫執行個體類別的 RDS for PostgreSQL 資料庫執行個體時，資料庫執行個體會自動使用 RDS Optimized Reads：

如需多可用區部署的詳細資訊，請參閱[設定及管理多可用區部署](#)。

若要開啟 RDS Optimized Reads，請執行下列其中一項：

- 使用其中一個 NVMe 為基礎的資料庫執行個體類別，建立 RDS for PostgreSQL 資料庫執行個體或多可用區域資料庫叢集。如需詳細資訊，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 修改現有的 RDS for PostgreSQL 資料庫執行個體或多可用區資料庫叢集，以使用其中一個 NVMe 為基礎的資料庫執行個體類別。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

RDS Optimized Reads 適用於支援其中一或多個具有本機 NVMe SSD 儲存體的資料庫執行個體類別之所有 AWS 區域。如需詳細資訊，請參閱[資料庫執行個體類別](#)。

若要切換回非最佳化讀取 RDS 執行個體，請將 RDS 執行個體或叢集的資料庫執行個體類別修改為僅支援資料庫工作負載 EBS 儲存體的類似執行個體類別。例如，如果目前的資料庫執行個體類別是 db.r6gd.4xlarge，請選擇 db.r6g.4xlarge 以切換回來。如需詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

監控使用 RDS Optimized Reads 的資料庫執行個體

您可以使用下列 CloudWatch 指標監控使用 RDS 最佳化讀取的資料庫執行個體：

- FreeLocalStorage
- ReadIOPSLocalStorage
- ReadLatencyLocalStorage
- ReadThroughputLocalStorage
- WriteIOPSLocalStorage
- WriteLatencyLocalStorage
- WriteThroughputLocalStorage

這些指標提供可用執行個體儲存體、IOPS 和輸送量的相關資料。如需這些指標的詳細資訊，請參閱[Amazon RDS 的 CloudWatch Amazon 實例級指標](#)。

若要監控本機儲存體的目前使用狀況，請使用下列查詢登入資料庫：

```
SELECT
    spcname AS "Name",
    pg_catalog.pg_size_pretty(pg_catalog.pg_tablespace_size(oid)) AS "size"
FROM
    pg_catalog.pg_tablespace
WHERE
    spcname IN ('rds_temp_tablespace');
```

如需暫存檔案及其用法的詳細資訊，請參閱[使用 PostgreSQL 管理暫存檔案](#)。

PostgreSQL 中 RDS Optimized Reads 的限制

下列限制適用於 PostgreSQL 中的 RDS Optimized Reads：

- 執行個體儲存體已滿時，交易會失敗。

將資料匯入 Amazon RDS 上的 PostgreSQL

假設有要移動到 Amazon RDS 的現有 PostgreSQL 部署。任務的複雜性取決於資料庫大小以及要傳輸的資料庫物件類型。例如，假設一個資料庫包含大約數 GB 的資料集，以及預存程序和觸發條件。此類資料庫會比僅含少量 GB 測試資料且沒有觸發條件或預存程序的簡單資料庫更複雜。

在下列情況中，建議您使用原生 PostgreSQL 資料庫遷移工具：

- 您準備進行同質遷移，亦即遷移的來源資料庫使用與目標資料庫相同的資料庫引擎。
- 您想要遷移整個資料庫。
- 原生工具可讓您以最短的停機時間來遷移系統。

在其他大部分情況下，使用 AWS Database Migration Service (AWS DMS) 來執行資料庫遷移程序是最佳的辦法。AWS DMS 可以遷移資料庫而不會導致停機，且系統會持續為許多資料庫引擎進行複寫，直到準備好切換為目標資料庫為止。您可以使用 AWS DMS 來遷移到相同資料庫引擎或不同資料庫引擎。如果您要遷移到與來源資料庫不同的資料庫引擎，可使用 AWS Schema Conversion Tool (AWS SCT)。您可使用 AWS SCT 遷移 AWS DMS 未遷移的結構描述物件。如需 AWS DMS 的詳細資訊，請參閱[什麼是 AWS Database Migration Service ?](#)

僅針對您的匯入，修改資料庫參數群組來包含下列設定。您應該測試參數設定，以根據資料庫執行個體大小找出最有效率的設定。匯入完成之後，您也需要將這些參數回復為生產值。

將資料庫執行個體設定修改為下列：

- 停用資料庫執行個體備份 (將 `backup_retention` 設為 0)。
- 停用多個可用區。

修改資料庫參數群組來包含下列設定。您應該僅在匯入資料時才使用這些設定。您應該測試參數設定，以根據資料庫執行個體大小找出最有效率的設定。匯入完成之後，您也需要將這些參數回復為生產值。

參數	匯入時的建議值	描述
<code>maintenance_work_mem</code>	524288、1048576、2097152 或 4194304 (KB)。這些設定相當於 512	此設定的值取決於主機大小。CREATE INDEX 陳述式期間會使用此參數，每個平行命令皆可使用這麼多的記憶體。計算最佳值，以免將此值設得太高而耗盡記憶體。

參數	匯入時的建議值	描述
	MB、1 GB、2 GB 和 4 GB。	
max_wal_size	256 (適用於 9.6 版)、4096 (適用於 10 版及更新版本)	將尺寸提高至最大，讓 WAL 能在自動檢查點期間成長。增加此參數可增加損毀復原所需的時間。此參數會將 checkpoint_segments 取代為 PostgreSQL 9.6 及更新版本。 若為 PostgreSQL 9.6 版，此數值是以 16 MB 為單位。若為更新的版本，該數值是以 1 MB 為單位。例如，在 9.6 版中，128 表示 128 個達到 16 MB 的區塊。在 12.4 版中，2048 表示 2048 個尺寸各為 1 MB 的區塊。
checkpoint_timeout	1800	此設定的值可減少 WAL 輪換次數。
synchronous_commit	關閉	停用此設定可加速寫入。關閉此參數會在伺服器當機時增加資料遺失的風險 (請勿關閉 FSYNC)。
wal_buffers	8192	此值以 8 KB 為單位。這同樣可加速產生 WAL
autovacuum	0	載入資料時停用 PostgreSQL 自動清理參數，以免浪費資源

使用 `pg_dump -Fc` (壓縮) 或 `pg_restore -j` (平行) 命令搭配這些設定。

Note

PostgreSQL 命令 `pg_dumpall` 需要 `super_user` 許可，但建立資料庫執行個體時並未授予此許可，因此無法用來匯入資料。

主題

- [從 Amazon EC2 執行個體匯入 PostgreSQL 資料庫](#)

- [使用 \copy 命令將資料匯入 PostgreSQL 資料庫執行個體上的資料表](#)
- [將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體](#)
- [在資料庫執行個體之間傳輸 PostgreSQL 資料庫](#)

從 Amazon EC2 執行個體匯入 PostgreSQL 資料庫

如果您在 Amazon EC2 執行個體上的 PostgreSQL 伺服器中有資料，且想要將資料移至 PostgreSQL 資料庫執行個體，則可以使用下列程序。下列清單顯示要採取的步驟。以下幾節將詳細討論各步驟。

1. 使用 `pg_dump` 建立檔案，其中包含要載入的資料
2. 建立目標資料庫執行個體
3. 使用 `psql` 在資料庫執行個體上建立資料庫並載入資料
4. 建立資料庫執行個體的資料庫快照

步驟 1：使用 `pg_dump` 建立檔案，其中包含要載入的資料

`pg_dump` 公用程式會使用 `COPY` 命令建立 PostgreSQL 資料庫的結構描述和資料傾印。`pg_dump` 產生的傾印指令碼會將資料載入相同名稱的資料庫中，並重新建立資料表、索引和外部索引鍵。您可以使用 `pg_restore` 命令和 `-d` 參數將資料還原至不同名稱的資料庫。

建立資料傾印之前，您應該查詢要傾印的資料表來取得資料列計數，以確認目標資料庫執行個體上的計數。

下列命令為資料庫 `mydb2` 建立名為 `mydb2dump.sql` 的傾印檔案。

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

步驟 2：建立目標資料庫執行個體

您可以使用 Amazon RDS 主控台、AWS CLI 或 API 來建立目標 PostgreSQL 資料庫執行個體。建立將備份保留期設為 0 的執行個體，並停用多個可用區。這麼做可以加速匯入資料。您必須先在執行個體上建立資料庫，才能傾印資料。此資料庫與包含傾印資料的資料庫可以具有相同名稱。或者，您可以建立不同名稱的資料庫。在此情況下，您可以使用 `pg_restore` 命令和 `-d` 參數將資料還原至新命名的資料庫。

例如，下列命令可用來傾印、還原和重新命名資料庫。

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database]
> [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
name] [database].dump
```

步驟 3：使用 psql 在資料庫執行個體上建立資料庫並載入資料

您可以使用原本用來執行 pg_dump 命令的相同連線，以連線至目標資料庫執行個體並重新建立資料庫。透過 psql，您可以使用主要使用者名稱和主要密碼在資料庫執行個體上建立資料庫。

下列範例使用 psql 和名為 mydb2dump.sql 的傾印檔案，在稱為 mypginstance 的 PostgreSQL 資料庫執行個體上建立名為 mydb2 的資料庫：

對於LinuxmacOS、或Unix：

```
psql \  
-f mydb2dump.sql \  
--host mypginstance.555555555555.aws-region.rds.amazonaws.com \  
--port 8199 \  
--username myawsuser \  
--password password \  
--dbname mydb2
```

在Windows中：

```
psql ^  
-f mydb2dump.sql ^  
--host mypginstance.555555555555.aws-region.rds.amazonaws.com ^  
--port 8199 ^  
--username myawsuser ^  
--password password ^  
--dbname mydb2
```

Note

指定此處所顯示提示以外的密碼，作為安全最佳實務。

步驟 4：建立資料庫執行個體的資料庫快照

在確認資料已載入至資料庫執行個體後，建議您建立目標 PostgreSQL 資料庫執行個體的資料庫快照。資料庫快照是您資料庫執行個體的完整備份，可將您的資料庫執行個體還原至已知狀態。若能在載入後立即拍攝資料庫快照，萬一發生事故，就不必重新載入資料。您也可使用快照植入新的資料庫執行個體。如需有關建立資料庫快照的資訊，請參閱[為單一可用區資料庫執行個體建立資料庫快照](#)。

使用 \copy 命令將資料匯入 PostgreSQL 資料庫執行個體上的資料表

PostgreSQL \copy 命令是中繼命令，可從 psql 互動式用戶端工具使用。您可以使用 \copy 將資料匯入至 RDS for PostgreSQL 資料庫執行個體上的資料表。若要使用 \copy 命令，首先需在目標資料庫執行個體上建立資料表結構，讓 \copy 具有複製資料的目的地。

您可以使用 \copy 從逗點分隔值 (CSV) 檔案 (例如已匯出並儲存到用戶端工作站的檔案) 載入資料。

若要將 CSV 資料匯入至目標 RDS for PostgreSQL 資料庫執行個體，需先使用 psql 連線至目標資料庫執行個體。

```
psql --host=db-instance.111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password --dbname=target-db
```

然後執行 \copy 命令搭配下列參數，以識別資料的目標及其格式。

- `target_table` – 應接收從 CSV 檔案所複製資料的資料表之名稱。
- `column_list` – 資料表的資料欄規格。
- `'filename'` – 您本機工作站上 CSV 檔案的完整路徑。

```
\copy target_table from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV;
```

如果 CSV 檔案包含資料欄標題資訊，您可以使用此版本的命令和參數。

```
\copy target_table (column-1, column-2, column-3, ...)  
from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV HEADER;
```

如果 \copy 命令失敗，PostgreSQL 會輸出錯誤訊息。

使用命令搭配下列範例所示的中 \copy 繼 psql 指令，在資料庫預覽環境中建立新的資料庫執行個體。此範例使用 `source-table` 做為來源資料表名稱、`source-table.csv` 做為 `.csv` 檔案，而 `target-db` 做為目標資料庫：

對於Linux macOS、或Unix：

```
$psql target-db \  
-U <admin user> \  
-p <port> \  
-h <DB instance name> \  
-c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

在Windows中：

```
$psql target-db ^  
-U <admin user> ^  
-p <port> ^  
-h <DB instance name> ^  
-c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

如需有關 `\copy` 命令的完整詳細資訊，請參閱 PostgreSQL 說明文件裡中繼命令中區段的 [psql](#) 頁面。

將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體

您可以將使用 Amazon Simple Storage Service 儲存的資料匯入 RDS for PostgreSQL 資料庫執行個體。要執行此操作，您首先安裝 RDS for PostgreSQL `aws_s3` 擴充功能。此擴充功能提供可用於從 Amazon S3 儲存貯體匯入資料的函數。儲存貯體是物件或檔案的 Amazon S3 容器。資料可以是逗號分隔值 (CSV) 檔案、文字檔案或壓縮 (gzip) 檔案。從下文中，您可以了解如何安裝擴充功能，以及如何將資料從 Amazon S3 匯入資料表。

您的資料庫必須執行 PostgreSQL 10.7 版或更高版本，才能從 Simple Storage Service (Amazon S3) 匯入 RDS for PostgreSQL。

如果您沒有資料儲存於 Amazon S3 上，您需要先建立儲存貯體並儲存資料。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的下列主題：

- [建立儲存貯體](#)
- [將物件新增到儲存貯體](#)

支援從 Amazon S3 匯入跨帳戶。如需更多詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [授予跨帳戶許可](#)。

從 S3 匯入資料時，您可以使用客戶受管金鑰進行加密。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [儲存在 AWS KMS 中的 KMS 金鑰](#)。

Note

Aurora Serverless v1 不支援從 Amazon S3 匯入資料。Aurora Serverless v2 則支援。

主題

- [安裝 aws_s3 擴充功能](#)
- [從 Amazon S3 資料匯入資料的概觀](#)
- [設定對 Amazon S3 儲存貯體的存取權](#)
- [將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體](#)
- [函數參考](#)

安裝 aws_s3 擴充功能

您需要先安裝 aws_s3 擴充功能，才能將 Amazon S3 與 RDS for PostgreSQL 資料庫執行個體搭配使用。此擴充功能提供從 Amazon S3 匯入資料的函數。它還提供了從 RDS for PostgreSQL 資料庫執行個體中匯出資料到 Amazon S3 儲存貯體的功能。如需詳細資訊，請參閱 [將資料從 RDS for PostgreSQL 資料庫執行個體匯出至 Amazon S3](#)。aws_s3 擴充功能取決於 aws_commons 擴充功能中的一些輔助函數，需要時會自動安裝。

安裝 aws_s3 擴充功能

1. 使用 psql (或 pgAdmin) 以具有 rds_superuser 權限的使用者身分連接到 RDS for PostgreSQL 資料庫執行個體。若您在安裝程序期間保留預設名稱，則連接為 postgres。

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password
```

2. 若要安裝擴充功能，請執行下列命令。

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;  
NOTICE: installing required extension "aws_commons"  
CREATE EXTENSION
```

3. 若要驗證是否已經安裝擴充功能，可以使用 psql \dx 中繼命令。

```
postgres=> \dx  
List of installed extensions
```


Name	Version	Schema	Description
aws_commons	1.2	public	Common data types across AWS services
aws_s3	1.1	public	AWS S3 extension for importing data from S3
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language

(3 rows)

現在已提供從 Amazon S3 匯入資料以及將資料匯出至 Amazon S3 的函數。

從 Amazon S3 資料匯入資料的概觀

將 S3 資料匯入至 Amazon RDS

首先，收集您需要提供給函數的詳細資訊。AWS 區域 其中包括 適用於 PostgreSQL 的 RDS 資料庫執行個體，以及儲存貯體名稱、檔案路徑、檔案類型，以及 Amazon S3 資料的存放位置。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的[檢視物件](#)。

Note

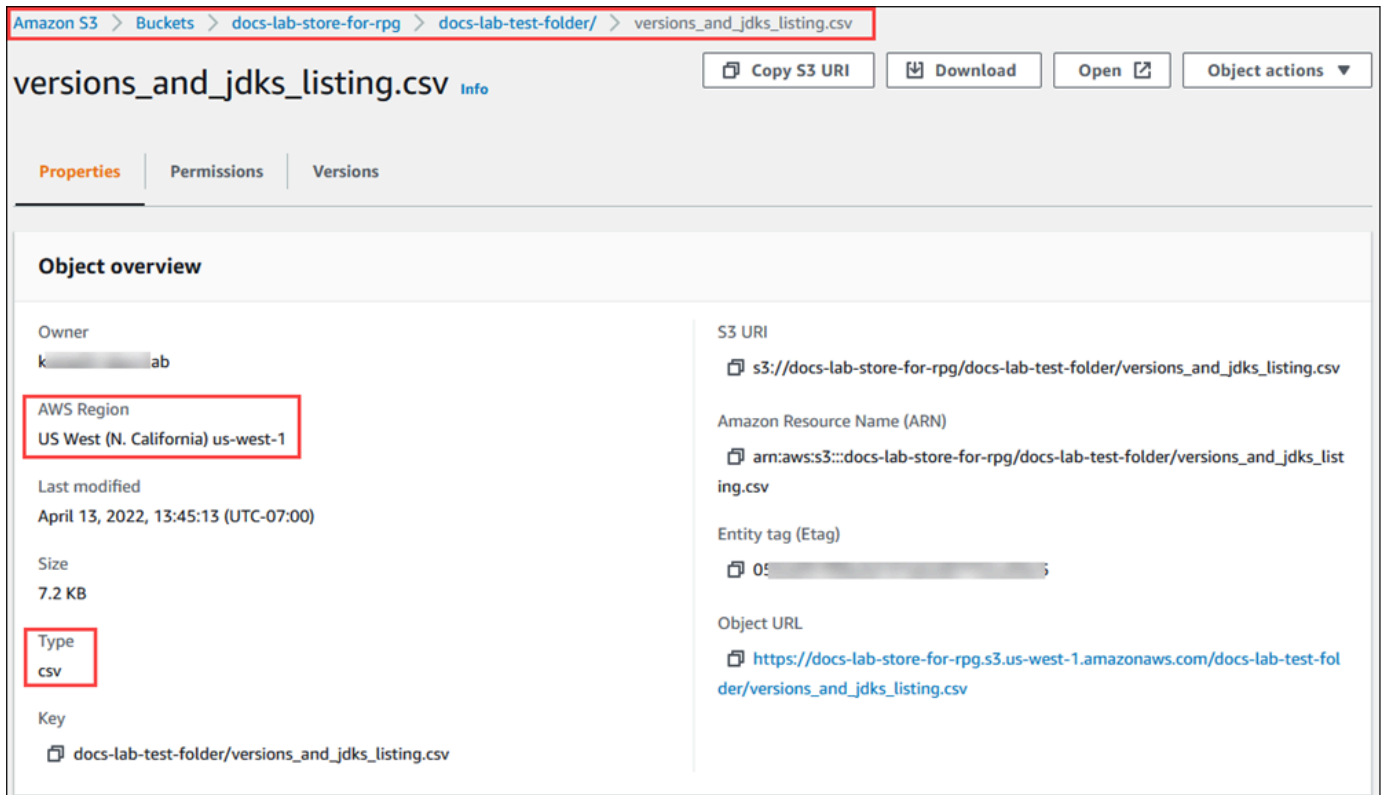
目前不支援從 Amazon S3 匯入多重部分資料。

1. 取得 `aws_s3.table_import_from_s3` 函數要匯入資料的資料表名稱。舉例來說，下列命令建立的資料表 `t1`，可用於之後的步驟中。

```
postgres=> CREATE TABLE t1
  (col1 varchar(80),
   col2 varchar(80),
   col3 varchar(80));
```

2. 取得 Amazon S3 儲存貯體以及要匯入資料的詳細資料。要執行此操作，請在以下網址開啟 Amazon S3 主控台：<https://console.aws.amazon.com/s3/>，然後選擇 Buckets (儲存貯體)。在清單中尋找包含您資料的儲存貯體。選擇儲存貯體，開啟其物件概觀頁面，然後選擇 Properties (屬性)。

記下值區名稱、路徑 AWS 區域、和檔案類型。您稍後需要 Amazon Resource Name (ARN)，以設定透過 IAM 角色對 Amazon S3 的存取權限。如需詳細資訊，請參閱 [設定對 Amazon S3 儲存貯體的存取權](#)。下圖顯示範例。



3. 您可以使用 AWS CLI 命令驗證 Amazon S3 儲存貯體上資料的路徑 `aws s3 cp`。如果資訊正確，此命令會下載 Amazon S3 檔案的副本。

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/sample_file_path ./
```

4. 設定 RDS for PostgreSQL 資料庫執行個體的許可，以允許存取 Amazon S3 儲存貯體上的檔案。若要這麼做，您可以使用 AWS Identity and Access Management (IAM) 角色或安全登入資料。如需詳細資訊，請參閱 [設定對 Amazon S3 儲存貯體的存取權](#)。
5. 將收集到的路徑和其他 Amazon S3 物件詳細資訊 (請參閱步驟 2) 提供給 `create_s3_uri` 函數，以建構 Amazon S3 URI 物件。若要進一步了解此函數，請參閱 [aws_commons.create_s3_uri](#)。以下是在 `psql` 工作階段中建構此物件的範例。

```
postgres=> SELECT aws_commons.create_s3_uri(
    'docs-lab-store-for-rpg',
    'versions_and_jdks_listing.csv',
    'us-west-1'
) AS s3_uri \gset
```

在下一個步驟中，您將此物件 (`aws_commons._s3_uri_1`) 傳遞到 `aws_s3.table_import_from_s3` 函數，將資料匯入資料表。

6. 調用 `aws_s3.table_import_from_s3` 函數，將資料從 Amazon S3 匯入資料表。如需參考資訊，請參閱 [aws_s3.table_import_from_s3](#)。如需範例，請參閱「[將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體](#)」。

設定對 Amazon S3 儲存貯體的存取權

若要從 Amazon S3 檔案匯入資料，請為 RDS for PostgreSQL 資料庫執行個體提供許可，以便存取檔案所在的 Amazon S3 儲存貯體。您可以透過以下兩個方式中的一個提供存取給 Amazon S3 儲存貯體，如下列主題中所述。

主題

- [使用 IAM 角色存取 Amazon S3 儲存貯體](#)
- [使用安全登入資料存取 Amazon S3 儲存貯體](#)
- [對 Amazon S3 的存取進行故障診斷](#)

使用 IAM 角色存取 Amazon S3 儲存貯體

在您由 Amazon S3 檔案載入資料之前，請為 RDS for PostgreSQL 資料庫執行個體提供許可，以便存取檔案所在的 Amazon S3 儲存貯體。這樣您就不必管理額外的登入資料資訊或在 [aws_s3.table_import_from_s3](#) 函數呼叫中提供它。

若要執行此動作，請建立可提供 Amazon S3 儲存貯體存取的 IAM 政策。建立 IAM 角色，並將政策連接到該角色。然後將 IAM 角色指派到您的資料庫執行個體。

Note

您無法建立 IAM 角色與 Aurora Serverless v1 資料庫叢集的關聯，因此以下步驟不適用。

透過 IAM 角色將 RDS for PostgreSQL 資料庫執行個體的存取權授予 Amazon S3

1. 建立 IAM 政策。

此政策可提供儲存貯體及物件許可，讓 RDS for PostgreSQL 資料庫執行個體能夠存取 Amazon S3。

在政策中納入下列必要動作，以允許由 Amazon S3 儲存貯體傳輸檔案至 Amazon RDS：

- `s3:GetObject`

- `s3:ListBucket`

在政策中包含下列資源，以識別 Amazon S3 儲存貯體和儲存貯體中的物件。這會顯示用於存取 Amazon S3 的 Amazon Resource Name (ARN) 格式。

- `ARN: AWS::: #####`
- `ARN: AWS::: ##### /*`

如需建立 Amazon RDS for PostgreSQL IAM 政策的詳細資訊，請參閱 [建立並使用 IAM 政策進行 IAM 資料庫存取](#)。另請參閱《IAM 使用者指南》中的 [教學：建立和連接您的第一個客戶受管原則](#)。

下列 AWS CLI 命令會建立以這些選項命名 `rds-s3-import-policy` 的 IAM 政策。它授予一個名為 `##` 示例桶的訪問權限。

Note

請記下此命令所傳回政策的 Amazon Resource Name (ARN)。在後續步驟中將政策連接至 IAM 角色時，您會需要此 ARN。

Example

對於 Linux/macOS、或 Unix：

```
aws iam create-policy \  
  --policy-name rds-s3-import-policy \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "s3import",  
        "Action": [  
          "s3:GetObject",  
          "s3:ListBucket"  
        ],  
        "Effect": "Allow",  
        "Resource": [  
          "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
```

```

        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
}
]
}'

```

在 Windows 中：

```

aws iam create-policy ^
--policy-name rds-s3-import-policy ^
--policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "s3import",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}'

```

2. 建立 IAM 角色。

這麼做是為了讓 Amazon RDS 擔任此 IAM 角色，以存取您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

建議您在資源型政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰，將服務的許可限定於特定資源。這是防止[混淆代理人問題](#)最有效的方式。

如果同時使用這兩個全域條件內容索引鍵，且 `aws:SourceArn` 值包含帳戶 ID，則在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的帳戶時，必須使用相同的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在政策中，請務必搭配資源的完整 ARN 來使用 `aws:SourceArn` 全域條件內容金鑰。下列範例示範如何使用 AWS CLI 命令來建立名為的角色 `rds-s3-import-role`。

Example

對於LinuxmacOS、或Unix：

```
aws iam create-role \  
  --role-name rds-s3-import-role \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",  
        "Condition": {  
          "StringEquals": {  
            "aws:SourceAccount": "111122223333",  
            "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"  
          }  
        }  
      }  
    ]  
  }'
```

在 Windows 中：

```
aws iam create-role ^  
  --role-name rds-s3-import-role ^  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",
```

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "111122223333",
    "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
  }
}
]
```

3. 將您建立的 IAM 政策附加至您建立的 IAM 角色。

下列 AWS CLI 命令會將在上一個步驟中建立的原則附加至您在先前步驟中記錄 *your-policy-arn* 的名為「以原則 ARN *rds-s3-import-role* 取代」的角色。

Example

對於 Linux macOS、或 Unix：

```
aws iam attach-role-policy \
  --policy-arn your-policy-arn \
  --role-name rds-s3-import-role
```

在 Windows 中：

```
aws iam attach-role-policy ^
  --policy-arn your-policy-arn ^
  --role-name rds-s3-import-role
```

4. 將 IAM 角色新增至資料庫執行個體。

您可以使用 AWS Management Console 或來執行此操作 AWS CLI，如下所述。

主控台

使用主控台為 PostgreSQL 資料庫執行個體新增 IAM 角色

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 PostgreSQL 資料庫執行個體名稱以顯示其詳細資訊。

3. 在 Connectivity & security (連線能力與安全) 標籤上，在 Manage IAM roles (管理 IAM 角色) 區段中，在 Add IAM roles to this instance (將 IAM 角色新增到此叢集執行個體) 下方，選擇要新增的角色。
4. 請在 Feature (功能) 下，選擇 s3Import。
5. 選擇 Add role (新增角色)。

AWS CLI

使用 CLI 為 PostgreSQL 資料庫執行個體新增 IAM 角色

- 使用下列命令將角色新增至名為 my-db-instance 的 PostgreSQL 資料庫執行個體。將 *your-role-arn* 替換為您前個步驟記下的角色 ARN。使用 s3Import 作為 --feature-name 選項的值。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier my-db-instance \  
  --feature-name s3Import \  
  --role-arn your-role-arn \  
  --region your-region
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier my-db-instance ^  
  --feature-name s3Import ^  
  --role-arn your-role-arn ^  
  --region your-region
```

RDS API

若要使用 Amazon RDS API 為 PostgreSQL 資料庫執行個體新增 IAM 角色，請呼叫「[資料庫叢集資料 AddRoleTo](#)」。

使用安全登入資料存取 Amazon S3 儲存貯體

如果想要，可以使用安全登入資料來提供 Amazon S3 儲存貯體的存取，而非使用 IAM 角色提供存取。要執行此操作，您可以指定 [aws_s3.table_import_from_s3](#) 函數呼叫中的 `credentials` 參數。

`credentials` 參數是類型的結構 `aws_commons._aws_credentials_1`，其中包含 AWS 認證。請使用 [aws_commons.create_aws_credentials](#) 函數在 `aws_commons._aws_credentials_1` 結構設定存取金鑰及秘密金鑰，如下所示。

```
postgres=> SELECT aws_commons.create_aws_credentials(  
    'sample_access_key', 'sample_secret_key', '')  
AS creds \gset
```

建立 `aws_commons._aws_credentials_1` 結構之後，請使用 [aws_s3.table_import_from_s3](#) 函數搭配 `credentials` 參數來匯入資料，如下所示。

```
postgres=> SELECT aws_s3.table_import_from_s3(  
    't', '', '(format csv)',  
    :s3_uri,  
    :creds  
);
```

或是您可在 [aws_commons.create_aws_credentials](#) 函數呼叫之中內嵌 `aws_s3.table_import_from_s3` 函數呼叫。

```
postgres=> SELECT aws_s3.table_import_from_s3(  
    't', '', '(format csv)',  
    :s3_uri,  
    aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')  
);
```

對 Amazon S3 的存取進行故障診斷

如果您在嘗試從 Amazon S3 匯入資料時遇到問題，請參閱下文以取得建議：

- [對 Amazon RDS 身分與存取進行故障診斷](#)
- 《Amazon Simple Storage Service 使用者指南》中的 [針對 Amazon S3 進行故障診斷](#)。
- 《IAM 使用者指南》中的 [針對 Amazon S3 和 IAM 進行故障診斷](#)

將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體

您可以使用 `aws_s3` 擴充功能的 `table_import_from_s3` 函數，由 Amazon S3 儲存貯體匯入資料。如需參考資訊，請參閱 [aws_s3.table_import_from_s3](#)。

Note

下列範例使用 IAM 角色方法，允許對 Amazon S3 儲存貯體的存取。因此，`aws_s3.table_import_from_s3` 函數呼叫不包含登入資料參數。

以下顯示一個典型範例。

```
postgres=> SELECT aws_s3.table_import_from_s3(  
    't1',  
    '',  
    '(format csv)',  
    :s3_uri'  
);
```

參數如下：

- `t1` – PostgreSQL 資料庫執行個體的表格名稱，資料會複製到此表格。
- `''` – 資料庫表格之中選用的欄清單。您可使用此參數指出哪些 S3 資料欄要置於哪些表格欄。如果沒有指定欄，所有欄都會複製到表格中。如需使用欄清單的範例，請參閱 [匯入使用自訂分隔符號的 Amazon S3 檔案](#)。
- `(format csv)` – PostgreSQL COPY 引數。複製程序使用 [PostgreSQL COPY](#) 命令的引數及格式匯入資料。格式的選擇包括逗號分隔值 (CSV)、文字和二進位。預設為文字。
- `s3_uri` – 包含識別 Amazon S3 檔案資訊的結構。如需使用 [aws_commons.create_s3_uri](#) 函數來建立 `s3_uri` 結構的範例，請參閱 [從 Amazon S3 資料匯入資料的概觀](#)。

如需此函數狀態的詳細資訊，請參閱 [aws_s3.table_import_from_s3](#)。

`aws_s3.table_import_from_s3` 函數傳回文字。若要指定從 Amazon S3 儲存貯體匯入的其他檔案類型，請參閱下列其中一個範例。

Note

匯入 0 位元組檔案將導致錯誤。

主題

- [匯入使用自訂分隔符號的 Amazon S3 檔案](#)
- [匯入 Amazon S3 壓縮 \(gzip\) 檔案](#)
- [匯入編碼的 Amazon S3 檔案](#)

匯入使用自訂分隔符號的 Amazon S3 檔案

下列範例顯示如何匯入使用自訂分隔符號的檔案。其中也顯示如何使用 `column_list` 函數的 `aws_s3.table_import_from_s3` 參數，控制資料放置於資料庫表格的位置。

我們在此範例假設下列資訊整理至 Amazon S3 檔案之中的縱線分隔欄。

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

匯入使用自訂分隔符號的檔案

1. 在資料庫為匯入資料建立表格。

```
postgres=> CREATE TABLE test (a text, b text, c text, d text, e text);
```

2. 使用以下的 `aws_s3.table_import_from_s3` 函數格式由 Amazon S3 檔案匯入資料。

您可在 `aws_commons.create_s3_uri` 函數呼叫之中內嵌 `aws_s3.table_import_from_s3` 函數呼叫以指定檔案。

```
postgres=> SELECT aws_s3.table_import_from_s3(
  'test',
  'a,b,d,e',
  'DELIMITER '|' ,
  aws_commons.create_s3_uri('DOC-EXAMPLE-BUCKET', 'pipeDelimitedSampleFile', 'us-
east-2')
```

```
);
```

資料目前位於下列欄的表格中。

```
postgres=> SELECT * FROM test;
 a | b | c | d | e
----+-----+----+----+-----+-----
 1 | foo1 | | bar1 | elephant1
 2 | foo2 | | bar2 | elephant2
 3 | foo3 | | bar3 | elephant3
 4 | foo4 | | bar4 | elephant4
```

匯入 Amazon S3 壓縮 (gzip) 檔案

下列範例顯示如何由以 gzip 壓縮的 Amazon S3 匯入檔案。匯入的檔案需具有以下 Amazon S3 中繼資料：

- 索引鍵：Content-Encoding
- 值：gzip

如果您使用上傳檔案 AWS Management Console，系統通常會套用中繼資料。如需使用 AWS Management Console、或 API 將檔案上傳到 Amazon S3 的相關資訊 AWS CLI，請參閱 Amazon 簡單儲存服務使用者指南中的[上傳物件](#)。

如需 Amazon S3 中繼資料的詳細資訊以及系統所提供中繼資料的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[在 Amazon S3 主控台中編輯物件中繼資料](#)。

請依據以下所示內容，將 gzip 檔案匯入 RDS for PostgreSQL 資料庫執行個體。

```
postgres=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_gzip', '', '(format csv)',
 'DOC-EXAMPLE-BUCKET', 'test-data.gz', 'us-east-2'
);
```

匯入編碼的 Amazon S3 檔案

下列範例顯示如何由採用 Windows-1252 編碼的 Amazon S3 匯入檔案。

```
postgres=> SELECT aws_s3.table_import_from_s3(
```

```
'test_table', '', 'encoding ''WIN1252''',  
aws_commons.create_s3_uri('DOC-EXAMPLE-BUCKET', 'SampleFile', 'us-east-2')  
);
```

函數參考

函數

- [aws_s3.table_import_from_s3](#)
- [aws_commons.create_s3_uri](#)
- [aws_commons.create_aws_credentials](#)

aws_s3.table_import_from_s3

將 Amazon S3 資料匯入 Amazon RDS 表。aws_s3 擴充功能提供 aws_s3.table_import_from_s3 函數。傳回值為文字。

語法

必要的參數為 table_name、column_list 及 options。這些參數可識別資料庫表格，並指定資料要如何複製到表格中。

您也可以使用下列參數：

- s3_info 參數指定要匯入的 Amazon S3 檔案。您使用此參數時，IAM 角色會將 Amazon S3 存取權提供給 PostgreSQL 資料庫執行個體。

```
aws_s3.table_import_from_s3 (  
    table_name text,  
    column_list text,  
    options text,  
    s3_info aws_commons._s3_uri_1  
)
```

- credentials 參數指定登入資料以存取 Amazon S3。您使用此項參數時，不必使用 IAM 角色。

```
aws_s3.table_import_from_s3 (  
    table_name text,  
    column_list text,  
    options text,  
    s3_info aws_commons._s3_uri_1,  
    credentials text  
)
```

```
credentials aws_commons._aws_credentials_1
)
```

參數

table_name

必要的文字字串，其中含有要匯入資料的 PostgreSQL 資料庫表格名稱。

column_list

必要的文字字串，其中含有要複製資料的 PostgreSQL 資料庫表格欄選用清單。如果字串為空白，就會使用表格的所有欄。如需範例，請參閱 [匯入使用自訂分隔符號的 Amazon S3 檔案](#)。

options

必要的文字字串，含有 PostgreSQL COPY 命令引數。這些引數指定資料要如何複製到 PostgreSQL 表格中。詳細資訊請參閱 [PostgreSQL COPY 文件](#)。

s3_info

aws_commons._s3_uri_1 複合類型，含有下列 S3 物件相關資訊：

- bucket – 含有檔案的 Amazon S3 儲存貯體名稱。
- file_path – 包括檔案路徑的 Amazon S3 檔案名稱。
- region— 該文件所在的 AWS 區域。如需「AWS 區域」名稱與相關值的清單，請參閱 [區域、可用區域和 Local Zones](#)。

登入資料

aws_commons._aws_credentials_1 複合類型，含有下列登入資料以用於匯入作業：

- 存取金鑰
- 私密金鑰
- 工作階段字符

如需建立 aws_commons._aws_credentials_1 複合結構的詳細資訊，請參閱 [aws_commons.create_aws_credentials](#)。

替代語法

為了協助進行測試，您可使用一組更大的參數取代 s3_info 及 credentials 參數。以下是 aws_s3.table_import_from_s3 函數的額外語法變化：

- 請不要使用 `s3_info` 參數識別 Amazon S3 檔案，而是使用 `bucket`、`file_path` 及 `region` 參數組合進行。使用這種形式的函數，Amazon S3 存取權會由 IAM 角色在 PostgreSQL 資料庫執行個體提供。

```
aws_s3.table_import_from_s3 (  
    table_name text,  
    column_list text,  
    options text,  
    bucket text,  
    file_path text,  
    region text  
)
```

- 請不要使用 `credentials` 參數識別 Amazon S3 存取，而是使用 `access_key`、`session_key` 及 `session_token` 參數組合進行。

```
aws_s3.table_import_from_s3 (  
    table_name text,  
    column_list text,  
    options text,  
    bucket text,  
    file_path text,  
    region text,  
    access_key text,  
    secret_key text,  
    session_token text  
)
```

替代參數

bucket

文字字串，其中含有包含檔案的 Amazon S3 儲存貯體名稱。

file_path

包含 Amazon S3 檔案名稱 (包括檔案路徑) 的文字字串。

region

識別檔案 AWS 區域 位置的文字字串。如需名 AWS 區域 稱與相關值的清單，請參閱 [區域、可用區域和 Local Zones](#)。

access_key

文字字串，其中含有用於匯入作業的存取金鑰。預設值為 NULL。

secret_key

文字字串，其中含有用於匯入作業的秘密金鑰。預設值為 NULL。

session_token

(選用) 文字字串，其中含有用於匯入作業的工作階段金鑰。預設值為 NULL。

aws_commons.create_s3_uri

建立 `aws_commons._s3_uri_1` 結構以保留 Amazon S3 檔案資訊。使用 `aws_commons.create_s3_uri` 函數 `s3_info` 參數的 [aws_s3.table_import_from_s3](#) 函數結果。

語法

```
aws_commons.create_s3_uri(  
    bucket text,  
    file_path text,  
    region text  
)
```

參數

bucket

必要的文字字串，其中含有檔案的 Amazon S3 儲存貯體名稱。

file_path

包含 Amazon S3 檔案名稱 (包括檔案路徑) 的必要文字字串。

region

包含檔案所在的 AWS 區域 必要文字字串。如需名 AWS 區域 稱與相關值的清單，請參閱 [區域、可用區域和 Local Zones](#)。

aws_commons.create_aws_credentials

在 `aws_commons._aws_credentials_1` 結構設定存取金鑰及秘密金鑰。使用 `aws_commons.create_aws_credentials` 函數 `credentials` 參數的 [aws_s3.table_import_from_s3](#) 函數結果。

語法

```
aws_commons.create_aws_credentials(  
    access_key text,  
    secret_key text,  
    session_token text  
)
```

參數

access_key

必要的文字字串，其中含有用於匯入 Amazon S3 檔案的存取金鑰。預設值為 NULL。

secret_key

必要的文字字串，其中含有用於匯入 Amazon S3 檔案的秘密金鑰。預設值為 NULL。

session_token

選用的文字字串，其中含有用於匯入 Amazon S3 檔案的工作階段字符。預設值為 NULL。如果您提供選用的 `session_token`，就可以使用臨時登入資料。

在資料庫執行個體之間傳輸 PostgreSQL 資料庫

針對 Amazon RDS 使用 PostgreSQL 可傳輸資料庫，您可以在兩個資料庫執行個體之間移動 PostgreSQL 資料庫。這是在不同資料庫執行個體之間遷移大型資料庫的一種非常快速的方法。若要使用此方式，您的資料庫執行個體必須皆執行相同的 PostgreSQL 主要版本。

此功能要求您在來源資料庫執行個體和目的地資料庫執行個體上都安裝 `pg_transport` 擴充功能。`pg_transport` 擴充功能提供實體傳輸機制，使用最少的方式移動資料庫檔案。此機制移動資料的速度比傳統的傾印和載入程序更快，而且停機時間更短。

Note

RDS for PostgreSQL 11.5 版和更高版本、RDS for PostgreSQL 10.10 版和更高版本中提供 PostgreSQL 可傳輸資料庫。

若要將 PostgreSQL 資料庫執行個體從一個 RDS for PostgreSQL 資料庫執行個體傳輸到另一個執行個體，請先設定來源和目的地執行個體，如 [設定用於傳輸的資料庫執行個體](#) 中所述。然後，您可以使用 [傳輸 PostgreSQL 資料庫](#) 中描述的函數傳輸資料庫。

主題

- [使用 PostgreSQL 可傳輸資料庫的限制](#)
- [設定傳輸 PostgreSQL 資料庫](#)
- [將 PostgreSQL 資料庫從來源傳輸至目的地](#)
- [資料庫傳輸期間發生的事](#)
- [可傳輸資料庫函數參考](#)
- [可傳輸資料庫參數參考](#)

使用 PostgreSQL 可傳輸資料庫的限制

可傳輸的資料庫有以下限制：

- 僅供讀取複本 – 您無法使用傳送僅供讀取複本或僅供讀取複本的上層執行個體資料庫。
- 不支援的欄位類型 – 您無法在任何計畫使用此方法傳輸的資料庫表格中使用 reg 資料類型。這些類型取決於系統上目錄物件 ID (OIDs)，通常會在傳輸時變更。
- 資料表空間 – 所有來源資料庫物件必須在預設 pg_default 資料表空間中。
- 相容性 – 來源與目的地資料庫執行個體都必須執行相同的 PostgreSQL 主要版本。
- 擴充功能：來源資料庫執行個體只能安裝 pg_transport。
- 角色和 ACL – 來源資料庫的存取權和擁有者資訊，不會一併傳輸至目的地資料庫。所有資料庫物件會由傳輸目的地的使用者建立和擁有。
- 並行傳輸：如果工作者程序設定正確，單個資料庫執行個體最多可支援 32 個並行傳輸，包括匯入和匯出。
- 僅限 RDS for PostgreSQL 資料庫執行個體：僅 RDS for PostgreSQL 資料庫執行個體上支援 PostgreSQL 可傳輸資料庫。您無法將其與內部部署資料庫或在 Amazon EC2 上執行的資料庫搭配使用。

設定傳輸 PostgreSQL 資料庫

開始之前，請確定 RDS for PostgreSQL 資料庫執行個體符合下列要求：

- 來源與目的地的 RDS for PostgreSQL 資料庫執行個體都必須執行相同的 PostgreSQL 版本。
- 目的地資料庫不能具有與要傳輸的來源資料庫名稱相同的資料庫。
- 用於執行傳輸作業的帳戶在來源資料庫和目的地資料庫上均需要 `rds_superuser` 權限。
- 來源資料庫執行個體的安全群組必須允許來自目的地資料庫執行個體的傳入存取。如果來源和目的地資料庫執行個體位於 VPC 中，則可能已經存在這種情況。如需安全群組的詳細資訊，請參閱[使用安全群組控制存取](#)。

將資料庫從來源資料庫執行個體傳輸至目的地資料庫執行個體，需要對與每個執行個體相關聯的資料庫參數群組進行多次變更。這意味著您必須為來源資料庫執行個體建立自訂資料庫參數群組，並為目的地資料庫執行個體建立自訂資料庫參數群組。

Note

如果您的資料庫執行個體已使用自訂資料庫參數群組進行設定，則可從下列程序中的步驟 2 開始。

設定用於傳輸資料庫的自訂資料庫群組參數

若要執行以下步驟，請使用具有 `rds_superuser` 權限的帳戶。

1. 如果來源和目的地資料庫執行個體使用預設資料庫參數群組，則需要使用適合執行個體的版本建立自訂資料庫參數群組。您可以執行此操作，以便變更數個參數的值。如需詳細資訊，請參閱[使用參數群組](#)。
2. 在自訂資料庫參數群組中，變更下列參數的值：
 - `shared_preload_libraries`：將 `pg_transport` 新增至程式庫清單中。
 - `pg_transport.num_workers`：預設值為 3。根據資料庫的需要增加或減少此值。對於 200 GB 的資料庫，建議此值不大於 8。請記住，如果增加此參數的預設值，還應同時增加 `max_worker_processes` 的值。
 - `pg_transport.work_mem`：預設值為 128 MB 或 256 MB，具體取決於 PostgreSQL 版本。預設設定通常可保持不變。
 - `max_worker_processes` – 需要使用以下計算設定此參數值：

```
(3 * pg_transport.num_workers) + 9
```

目的地需要此值來處理傳輸中涉及各種後台工作程序。如要進一步了解 `max_worker_processes`，請參閱 PostgreSQL 文件中的[資源耗用](#)。

如需 `pg_transport` 參數的詳細資訊，請參閱 [可傳輸資料庫參數參考](#)。

- 重新啟動來源 RDS for PostgreSQL 資料庫執行個體和目的地執行個體，以便讓參數的設定生效。
- 連接至 RDS for PostgreSQL 來源資料庫執行個體。

```
psql --host=source-instance.111122223333.aws-region.rds.amazonaws.com --port=5432  
--username=postgres --password
```

- 從資料庫執行個體的公有架構中移除無關的擴充功能。在實際傳輸操作期間僅允許 `pg_transport` 擴充功能。
- 安裝 `pg_transport` 擴充功能，如下所示：

```
postgres=> CREATE EXTENSION pg_transport;  
CREATE EXTENSION
```

- 連接至 RDS for PostgreSQL 目的地資料庫執行個體。刪除任何無關的擴充功能，然後安裝 `pg_transport` 擴充功能。

```
postgres=> CREATE EXTENSION pg_transport;  
CREATE EXTENSION
```

將 PostgreSQL 資料庫從來源傳輸至目的地

在您完成 [設定傳輸 PostgreSQL 資料庫](#) 中的描述程序，您可以開始進行傳輸。若要這麼做，請執行目的地的資料庫執行個體上的 `transport.import_from_server` 功能。您可以在下列語法中找到函數參數。

```
SELECT transport.import_from_server(  
  'source-db-instance-endpoint',  
  'source-db-instance-port',  
  'source-db-instance-user',  
  'source-user-password',  
  'source-database-name',
```

```
'destination-user-password',
false);
```

範例中顯示的 `false` 值告訴函數這並非試轉。若要測試傳輸設定，您可以在呼叫函數時為 `dry_run` 選項指定 `true`，如下所示：

```
postgres=> SELECT transport.import_from_server(
    'docs-lab-source-db.666666666666aws-region.rds.amazonaws.com', 5432,
    'postgres', '*****', 'labdb', '*****', true);
INFO: Starting dry-run of import of database "labdb".
INFO: Created connections to remote database      (took 0.03 seconds).
INFO: Checked remote cluster compatibility      (took 0.05 seconds).
INFO: Dry-run complete                          (took 0.08 seconds total).
import_from_server
-----
(1 row)
```

輸出 INFO 行是因為 `pg_transport.timing` 參數設定為其預設值 `true`。在執行命令時將 `dry_run` 設定為 `false`，來源資料庫即會匯入目的地，如下所示：

```
INFO: Starting import of database "labdb".
INFO: Created connections to remote database      (took 0.02 seconds).
INFO: Marked remote database as read only        (took 0.13 seconds).
INFO: Checked remote cluster compatibility      (took 0.03 seconds).
INFO: Signaled creation of PITR blackout window (took 2.01 seconds).
INFO: Applied remote database schema pre-data   (took 0.50 seconds).
INFO: Created connections to local cluster      (took 0.01 seconds).
INFO: Locked down destination database         (took 0.00 seconds).
INFO: Completed transfer of database files      (took 0.24 seconds).
INFO: Completed clean up                      (took 1.02 seconds).
INFO: Physical transport complete              (took 3.97 seconds total).
import_from_server
-----
(1 row)
```

此功能需要提供資料庫使用者密碼。因此，我們建議您在傳輸完成之後變更您所使用的使用者密碼。或者，您可以使用 SQL 繫結變數建立暫時的使用者角色。使用暫時的角色進行傳輸，並在之後捨棄該角色。

如果傳輸失敗，您可能會看到類似以下內容的錯誤訊息：

```
pg_transport.num_workers=8 25% of files transported failed to download file data
```

"Failed to download file data" (下載檔案資料失敗) 錯誤訊息表示工作者程序的數量未正確設定為資料庫的大小。您可能需要增加或減少為 `pg_transport.num_workers` 設定的值。每次失敗都會報告完成的百分比，以便您可以查看變更的影響。例如，在一種情況下，將設定從 8 變更為 4 會導致以下結果：

```
pg_transport.num_workers=4 75% of files transported failed to download file data
```

請記住，在傳輸過程中也會考慮 `max_worker_processes` 參數。換言之，您可能需要修改 `pg_transport.num_workers` 和 `max_worker_processes` 才能成功傳輸資料庫。當 `pg_transport.num_workers` 設定為 2 時，顯示的範例最終可運作：

```
pg_transport.num_workers=2 100% of files transported
```

如需有關 `transport.import_from_server` 功能及其參數的詳細資訊，請參閱 [可傳輸資料庫函數參考](#)。

資料庫傳輸期間發生的事

PostgreSQL 可傳輸資料庫功能使用提取模式將資料庫從來源匯入目的地資料庫執行個體。`transport.import_from_server` 功能在目的地資料庫執行個體上建立傳輸中資料庫。無法存取傳輸進行期間目的地資料庫執行個體上的傳輸中資料庫。

當傳輸開始時，所有目前來源資料庫上的工作階段皆會終止。任何其他不在來源資料庫執行個體上的資料庫，不會受到傳輸的影響。

來源資料庫會變成特定的唯讀模式。當進入此模式時，您可以連接來源資料庫且執行唯獨模式查詢。然而，會封鎖寫入啟用查詢和其他類型的命令。只有開始進行傳輸的特定來源資料庫會受到此限制的影響。

傳輸期間，您不能重新恢復目的地資料庫執行個體的還原時間點。這是因為傳輸並非交易，且不會使用 PostgreSQL 預寫紀錄來紀錄變更。如果目的地資料庫執行個體有啟用自動備份，會在傳輸完成後自動進行備份。備份完成後，Point-in-time 還原可用於數次。

如果傳輸失敗，`pg_transport` 擴充套件會試圖對來源和目的地資料庫執行個體重做所有變更。這包含移除目的地已經部分傳輸的資料庫。根據錯誤的類型，來源資料庫可能會持續拒絕寫入啟用查詢。若發生這種情況，請使用下列命令以允許寫入啟用查詢。

```
ALTER DATABASE db-name SET default_transaction_read_only = false;
```

可傳輸資料庫函數參考

`transport.import_from_server` 函數透過輸入來源資料庫執行個體至目的地資料庫執行個體，來傳輸 PostgreSQL 資料庫。藉由實體資料庫連接傳輸機制做到此功能。

在開始傳輸之前，此函數將驗證來源和目的地資料庫執行個體的版本是否相同，並且是否與遷移相容。同時還會確認目的地資料庫執行個體有足夠的空間供來源資料庫執行個體使用。

語法

```
transport.import_from_server(  
    host text,  
    port int,  
    username text,  
    password text,  
    database text,  
    local_password text,  
    dry_run bool  
)
```

傳回值

無。

參數

您可以在下表尋找 `transport.import_from_server` 功能參數的說明。

參數	描述
<code>host</code>	來源資料庫執行個體的端點。
<code>port</code>	整數代表來源資料庫執行個體的連接埠。 PostgreSQL 資料庫執行個體通常使用 5432 連接埠。
<code>username</code>	來源資料庫執行個體的使用者。該使用者必須為 <code>rds_superuser</code> 角色的成員。
<code>password</code>	來源資料庫執行個體的使用者密碼。

參數	描述
database	來源資料庫執行個體中要傳輸的資料庫名稱。
local_password	目的地資料庫執行個體目前使用者的本地密碼。該使用者必須為 <code>rds_superuser</code> 角色的成員。
dry_run	選用的布林值指定是否試執行。預設為 <code>false</code> ，代表傳輸繼續。 若要確認來源與目的地資料庫執行個體間的相容性，而不執行實際傳輸，請設定 <code>dry_run</code> 為 <code>true</code> 。

範例

如需範例，請參閱 [將 PostgreSQL 資料庫從來源傳輸至目的地](#)。

可傳輸資料庫參數參考

有幾個參數控制 `pg_transport` 擴充功能的行為。您可以在下面找到這些參數的說明。

`pg_transport.num_workers`

用於傳輸流程的工作者數量。預設為 3。有效值為 1–32。最大的資料庫傳輸通常會用到的工作者也少於 8 個。傳輸期間，目的地資料庫執行個體上的此設定值會由目的地和來源使用。

`pg_transport.timing`

指定是否在傳輸期間回報時間資訊。預設為 `true`，表示報告計時資訊。建議將此參數保留為設定的 `true`，以便您監控進度。如需了解輸出範例，請參閱 [將 PostgreSQL 資料庫從來源傳輸至目的地](#)。

`pg_transport.work_mem`

配置給每個工作者最大的記憶體數量。預設值為 131,072 KB 或 262,144 KB (256 MB)，具體取決於 PostgreSQL 版本。最小值為 64 MB (65,536 KB)。有效的值以 kb (KB) 為單位，且為二進位制，即 1 KB = 1024 位元組。

傳輸可能使用比此參數指定的記憶體更少。即使是大型資料庫傳輸，每個工作者會用到的記憶體通常少於 256 MB (262,144 KB)。

將資料從 RDS for PostgreSQL 資料庫執行個體匯出至 Amazon S3

您可以從 RDS for PostgreSQL 資料庫執行個體中查詢資料，然後將資料直接匯出至 Amazon S3 儲存貯體中存放的檔案中。若要這麼做，首先要安裝 RDS for PostgreSQL `aws_s3` 擴充功能。此擴充功能提供函數，可用於匯出資料至 Amazon S3。接著，您可以了解如何安裝擴充功能，以及如何將資料匯出至 Amazon S3。

Note

不支援跨帳戶匯出至 Amazon S3。

所有目前可用的 RDS for PostgreSQL 版本都支援將快照資料匯出至 Amazon 簡單儲存服務。如需詳細版本資訊，請參閱《Amazon RDS for PostgreSQL 版本資訊》中的 [Amazon RDS for PostgreSQL 版本更新](#)。

如果您沒有為匯出設定儲存貯體，請參閱下列主題 Amazon Storage Service 使用者指南。

- [設定 Amazon S3](#)
- [建立儲存貯體](#)

根據預設，從 RDS 匯出到 Amazon S3 PostgreSQL 料會使用伺服器端加密與 AWS 受管金鑰。如果您使用儲存貯體加密，Amazon S3 儲存貯體必須使用 AWS Key Management Service (AWS KMS) 金鑰 (SSE-KMS) 加密。目前不支援使用 Amazon S3 受管金鑰 (SSE-S3) 加密的儲存貯體。

Note

您可以使用 AWS Management Console、AWS CLI 或 Amazon RDS API 將資料庫快照資料儲存到 Amazon S3。如需詳細資訊，請參閱 [將資料庫快照資料匯出至 Amazon S3](#)。

主題

- [安裝 `aws_s3` 擴充功能](#)
- [將資料匯出至 Amazon S3 的概觀](#)
- [指定要匯出的 Amazon S3 檔案路徑](#)
- [設定對 Amazon S3 儲存貯體的存取權](#)

- [使用 `aws_s3.query_export_to_s3` 函數匯出查詢資料](#)
- [對 Amazon S3 的存取進行故障診斷](#)
- [函數參考](#)

安裝 `aws_s3` 擴充功能

在您可以使用 Amazon Simple Storage Service 搭配 RDS for PostgreSQL 資料庫執行個體之前，您需要安裝 `aws_s3` 擴充功能。此擴充功能提供從 RDS for PostgreSQL 資料庫執行個體匯出資料至 Amazon S3 儲存貯體的功能。它還提供可從 Amazon S3 匯入資料的函數。如需詳細資訊，請參閱 [將資料從 Amazon S3 匯入 RDS for PostgreSQL 資料庫執行個體](#)。 `aws_s3` 擴充功能取決於 `aws_commons` 擴充功能中的一些輔助函數，需要時會自動安裝。

安裝 `aws_s3` 擴充功能

1. 使用 `psql` (或 `pgAdmin`) 以具有 `rds_superuser` 權限的使用者身分連接到 RDS for PostgreSQL 資料庫執行個體。若您在安裝程序期間保留預設名稱，則連接為 `postgres`。

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password
```

2. 若要安裝擴充功能，請執行下列命令。

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;  
NOTICE: installing required extension "aws_commons"  
CREATE EXTENSION
```

3. 若要驗證是否已經安裝擴充功能，可以使用 `psql \dx` 中繼命令。

```
postgres=> \dx  
List of installed extensions  
Name | Version | Schema | Description  
-----+-----+-----+-----  
aws_commons | 1.2 | public | Common data types across AWS services  
aws_s3 | 1.1 | public | AWS S3 extension for importing data from S3  
plpgsql | 1.0 | pg_catalog | PL/pgSQL procedural language  
(3 rows)
```

現在可以使用從 Amazon S3 匯入和匯出資料的功能。

請確認您的 RDS for PostgreSQL 版本支援匯出至 Amazon S3

您可以使用 `describe-db-engine-versions` 命令，驗證您的 RDS for PostgreSQL 版本是否支援匯出至 Amazon S3。下列範例會驗證 10.14 版本的支援。

```
aws rds describe-db-engine-versions --region us-east-1
--engine postgres --engine-version 10.14 | grep s3Export
```

如果輸出包含字串 "s3Export"，則引擎支援 Amazon S3 匯出。否則，引擎不支援它們。

將資料匯出至 Amazon S3 的概觀

如果要將儲存在 RDS for PostgreSQL 資料庫中的資料匯出至 Amazon S3 儲存貯體，請使用以下程序。

將 RDS for PostgreSQL 資料匯出至 S3

1. 識別用於匯出資料的 Amazon S3 檔案路徑。如需此程序的詳細資訊，請參閱[指定要匯出的 Amazon S3 檔案路徑](#)。
2. 提供許可，以存取 Amazon S3 儲存貯體。

如果要將資料匯出至 Amazon S3 檔案，請為 RDS for PostgreSQL 資料庫執行個體提供許可，以便存取匯出將用來儲存的 Amazon S3 儲存貯體。這麼做包括以下步驟：

1. 建立 IAM 政策來為您要匯出的 Amazon S3 儲存貯體提供存取權。
2. 建立 IAM 角色。
3. 請將您建立的政策連接到您建立的角色。
4. 將此 IAM 角色新增至您的 資料庫叢集。

如需此程序的詳細資訊，請參閱[設定對 Amazon S3 儲存貯體的存取權](#)。

3. 識別資料庫查詢以取得資料。呼叫 `aws_s3.query_export_to_s3` 函數來匯出查詢資料。

完成上述的準備工作後，請使用 [aws_s3.query_export_to_s3](#) 函數將查詢結果匯出至 Amazon S3。如需此程序的詳細資訊，請參閱[使用 aws_s3.query_export_to_s3 函數匯出查詢資料](#)。

指定要匯出的 Amazon S3 檔案路徑

指定下列資訊來識別 Amazon S3 中您要匯出資料的位置：

- 儲存貯體名稱 – 儲存貯體是 Amazon S3 物件或檔案的容器。

如需使用 Amazon S3 儲存資料的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[建立儲存貯體](#)和[檢視物件](#)。

- 檔案路徑 – 檔案路徑會識別匯出項目儲存在 Amazon S3 儲存貯體中的位置。檔案路徑由以下項目組成：
 - 識別虛擬資料夾路徑的選擇性路徑字首。
 - 識別一或多個要儲存檔案的檔案字首。較大的匯出項目會儲存在多個檔案中，每個檔案的大小上限約為 6 GB。其他檔案名稱具有相同的檔案字首，但會加上 `_partXX`。XX 代表 2，接著是 3，以此類推

舉例來說，具有 `exports` 資料夾和 `query-1-export` 檔案字首的檔案路徑為 `/exports/query-1-export`。

- AWS 區域 (選用) — Amazon S3 儲存貯體所在的 AWS 區域。如果您未指定 AWS 區域值，則 Amazon RDS 會將您的檔案儲存到與匯出執行個體位於相同 AWS 區域的 Amazon S3。

Note

目前，該 AWS 區域必須與匯出執行個體的區域相同。

如需「AWS 區域」名稱與相關值的清單，請參閱[區域、可用區域和 Local Zones](#)。

如果要保留匯出項目儲存之位置的 Amazon S3 檔案資訊，可以使用 [aws_commons.create_s3_uri](#) 函數建立 `aws_commons._s3_uri_1` 複合結構，如下所示。

```
psql=> SELECT aws_commons.create_s3_uri(  
    'DOC-EXAMPLE-BUCKET',  
    'sample-filepath',  
    'us-west-2'  
) AS s3_uri_1 \gset
```

您稍後可以在對 `s3_uri_1` 函數的呼叫中以參數形式提供此 [aws_s3.query_export_to_s3](#) 值。如需範例，請參閱「[使用 aws_s3.query_export_to_s3 函數匯出查詢資料](#)」。

設定對 Amazon S3 儲存貯體的存取權

如果要將資料匯出至 Amazon S3，請為 PostgreSQL DB 執行個體提供許可，以便存取用來放置檔案的 Amazon S3 儲存貯體。

若要執行此操作，請使用下列程序。

透過 IAM 角色授與 PostgreSQL 資料庫執行個體的 Amazon S3 存取權

1. 建立 IAM 政策。

此政策可提供儲存貯體及物件許可，讓 PostgreSQL 資料庫執行個體能夠存取 Amazon S3。

在建立此原則的過程中，請採取下列步驟：

- a. 在政策中納入下列必要動作，以允許從 PostgreSQL 資料庫叢集執行個體傳輸檔案至 Amazon S3 儲存貯體：
 - `s3:PutObject`
 - `s3:AbortMultipartUpload`
- b. 包含用於識別 Amazon S3 儲存貯體和物件的 Amazon Resource Name (ARN)。存取 Amazon S3 的 ARN 格式為：`arn:aws:s3:::DOC-EXAMPLE-BUCKET/*`

如需如何建立 Amazon RDS for PostgreSQL IAM 政策的詳細資訊，請參閱[建立並使用 IAM 政策進行 IAM 資料庫存取](#)。另請參閱《IAM 使用者指南》中的[教學：建立和連接您的第一個客戶受管原則](#)。

下列 AWS CLI 命令會建立以這些選項命名 `rds-s3-export-policy` 的 IAM 政策。它授予一個名為 `##` 示例桶的訪問權限。

Warning

建議您在設有可存取特定儲存貯體之端點原則的私有 VPC 中設定資料庫。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[對 Amazon S3 使用端點政策](#)。強烈建議您不要建立具有全資源存取權的政策。這個存取權可能會對資料安全構成威脅。如果您建立的原則能為 `s3:PutObject` 提供使用 `"Resource": "*"` 存取所有資源的存取權，則具有匯出許可的使用者可將資料匯出至您帳戶中的所有儲存貯體。此外，使用者可以將資料匯出至您 AWS 區域內可公開寫入的所有儲存貯體。

政策建立後，請記下政策的 Amazon Resource Name (ARN)。在後續步驟中將政策附加至 IAM 角色時，您會需要此 ARN。

```
aws iam create-policy --policy-name rds-s3-export-policy --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "s3export",
      "Action": [
        "s3:PutObject*",
        "s3:ListBucket",
        "s3:GetObject*",
        "s3:DeleteObject*",
        "s3:GetBucketLocation",
        "s3:AbortMultipartUpload"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}'
```

2. 建立 IAM 角色。

您會執行此動作，使得 Amazon RDS 可以代表您擔任此 IAM 角色，以存取您的 Amazon S3 儲存貯體。如需更多詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以將許可委派給 IAM 使用者](#)。

建議您在資源型政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰，將服務的許可限定於特定資源。這是防止[混淆代理人問題](#)最有效的方式。

如果同時使用這兩個全域條件內容索引鍵，且 `aws:SourceArn` 值包含帳戶 ID，則在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的帳戶時，必須使用相同的帳戶 ID。

- 如果您想要跨服務存取單一資源，請使用 `aws:SourceArn`。
- 如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

在政策中，請務必搭配資源的完整 ARN 來使用 `aws:SourceArn` 全域條件內容金鑰。下列範例示範如何使用 AWS CLI 命令來建立名為的角色 `rds-s3-export-role`。

Example

對於LinuxmacOS、或Unix：

```
aws iam create-role \  
  --role-name rds-s3-export-role \  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",  
        "Condition": {  
          "StringEquals": {  
            "aws:SourceAccount": "111122223333",  
            "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"  
          }  
        }  
      }  
    ]  
  }'
```

在 Windows 中：

```
aws iam create-role ^  
  --role-name rds-s3-export-role ^  
  --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Principal": {  
          "Service": "rds.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole",
```

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "111122223333",
    "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
  }
}
]
```

3. 將您建立的 IAM 政策附加至您建立的 IAM 角色。

下列 AWS CLI 命令會將先前建立的原則附加至您在先前步驟中所記錄的名為「以原則 ARN rds-s3-export-role」。取代 *your-policy-arn* 的角色。

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name rds-s3-export-role
```

4. 將 IAM 角色新增至資料庫執行個體。您可以使用 AWS Management Console 或來執行此操作 AWS CLI，如下所述。

主控台

使用主控台為 PostgreSQL 資料庫執行個體新增 IAM 角色

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 選擇 PostgreSQL 資料庫執行個體名稱以顯示其詳細資訊。
3. 在 Connectivity & security (連線能力與安全性) 標籤上的 Manage IAM roles (管理 IAM 角色) 區段中，選擇要在 Add IAM roles to this instance (新增 IAM 角色到此執行個體) 下新增的角色。
4. 請在 Feature (功能) 下，選擇 s3Export。
5. 選擇 Add role (新增角色)。

AWS CLI

使用 CLI 為 PostgreSQL 資料庫執行個體新增 IAM 角色

- 使用下列命令將角色新增至名為 `my-db-instance` 的 PostgreSQL 資料庫執行個體。將 *your-role-arn* 替換為您前個步驟記下的角色 ARN。使用 `s3Export` 作為 `--feature-name` 選項的值。

Example

對於LinuxmacOS、或Unix：

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier my-db-instance \  
  --feature-name s3Export \  
  --role-arn your-role-arn \  
  --region your-region
```

在 Windows 中：

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier my-db-instance ^  
  --feature-name s3Export ^  
  --role-arn your-role-arn ^  
  --region your-region
```

使用 `aws_s3.query_export_to_s3` 函數匯出查詢資料

呼叫 [aws_s3.query_export_to_s3](#) 函數來將 PostgreSQL 資料匯出至 Amazon S3。

主題

- [必要條件](#)
- [正在呼叫 `aws_s3.query_export_to_s3`](#)
- [匯出至使用自訂分隔符號的 CSV 檔案](#)
- [利用編碼匯出至二進位檔案](#)

必要條件

使用 `aws_s3.query_export_to_s3` 函數前，請先完成下列必要條件：

- 安裝需要的 PostgreSQL 延伸，如[將資料匯出至 Amazon S3 的概觀](#) 中所述。
- 決定要將資料匯出至 Amazon S3 的位置，如[指定要匯出的 Amazon S3 檔案路徑](#) 中所述。
- 確認資料庫執行個體如 [設定對 Amazon S3 儲存貯體的存取權](#) 中所述，有對 Amazon S3 的匯出存取權。

下列範例會使用稱為 `sample_table` 的資料庫資料表。這些範例會將資料匯出至名為 *DOC/EXAMPLE* 值區的值區。範例資料表和資料會以下列 `psql` 形式的 SQL 陳述式建立。

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid,name) VALUES (1, 'Monday'), (2, 'Tuesday'), (3,
'Wednesday');
```

正在呼叫 `aws_s3.query_export_to_s3`

以下示範呼叫 [aws_s3.query_export_to_s3](#) 函數的基本方式。

這些範例會使用變數 `s3_uri_1` 來識別包含識別 Amazon S3 檔案資訊的結構。使用 [aws_commons.create_s3_uri](#) 函數來建立結構。

```
psql=> SELECT aws_commons.create_s3_uri(
    'DOC-EXAMPLE-BUCKET',
    'sample-filepath',
    'us-west-2'
) AS s3_uri_1 \gset
```

雖然下列兩個 `aws_s3.query_export_to_s3` 函數呼叫的參數有所不同，但這些範例的結果是相同的。`sample_table` 表中的所有行都導出到一個名為 *DOC/EXAMPLE* 桶的存儲桶中。

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
sample_table', :s3_uri_1);

psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
sample_table', :s3_uri_1, options := 'format text');
```

參數說明如下：

- 'SELECT * FROM sample_table' – 第一個字串是包含 SQL 查詢的文字字串。PostgreSQL 引擎會執行此查詢。查詢結果會複製到其他參數中識別的 S3 儲存貯體。
- ':s3_uri_1' – 此參數是識別 Amazon S3 檔案的結構。此範例使用變數來識別先前建立的結構。您可以改為透過在 `aws_commons.create_s3_uri` 函數呼叫中包含內嵌 `aws_s3.query_export_to_s3` 函數呼叫來建立結構，如下所示。

```
SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
    aws_commons.create_s3_uri('DOC-EXAMPLE-BUCKET', 'sample-filepath', 'us-west-2')
);
```

- `options := 'format text'` – `options` 是包含 PostgreSQL COPY 引數的選用文字字串。複製程序使用 [PostgreSQL COPY](#) 命令的引數及格式。

如果指定的檔案不存在於 Amazon S3 儲存貯體內，就會建立。如果檔案已存在，即會遭到覆寫。以下是存取 Amazon S3 中已匯出資料的語法。

```
s3-region:://bucket-name[/path-prefix]/file-prefix
```

較大的匯出項目會儲存在多個檔案中，每個檔案的大小上限約為 6 GB。其他檔案名稱具有相同的檔案字首，但會加上 `_partXX`。`XX` 代表 2，接著是 3，以此類推。舉例來說，假設您要指定儲存資料檔案的路徑，如下所示。

```
s3-us-west-2://DOC-EXAMPLE-BUCKET/my-prefix
```

如果匯出必須建立三個資料檔案，則 Amazon S3 儲存貯體會包含下列資料檔案。

```
s3-us-west-2://DOC-EXAMPLE-BUCKET/my-prefix
s3-us-west-2://DOC-EXAMPLE-BUCKET/my-prefix_part2
s3-us-west-2://DOC-EXAMPLE-BUCKET/my-prefix_part3
```

如需此函數的完整參考以及其他呼叫方式，請參閱 [aws_s3.query_export_to_s3](#)。如需存取 Amazon S3 中檔案的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [檢視物件](#)。

匯出至使用自訂分隔符號的 CSV 檔案

下列範例示範如何呼叫使用自訂分隔符號來將資料匯入檔案的 [aws_s3.query_export_to_s3](#) 函數。範例使用了 [PostgreSQL COPY](#) 命令的引數，來指定逗號分隔值 (CSV) 格式和冒號 (:) 分隔符號。

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :s3_uri_1',
options := 'format csv, delimiter $$:$$');
```

利用編碼匯出至二進位檔案

下列範例示範如何呼叫 [aws_s3.query_export_to_s3](#) 函數來將資料匯出至使用 Windows-1253 編碼的二進位檔案。

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :s3_uri_1',
options := 'format binary, encoding WIN1253');
```

對 Amazon S3 的存取進行故障診斷

如果您在嘗試將資料匯出至 Amazon S3 時遇到連線問題，請先確認與資料庫執行個體關聯之 VPC 安全群組的輸出存取規則允許網路連線。明確的說，安全群組必須擁有允許資料庫執行個體傳至連接埠 443 和任何 IPv4 地址 (0.0.0.0/0)。如需詳細資訊，請參閱 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

另請參閱以下內容中的建議：

- [對 Amazon RDS 身分與存取進行故障診斷](#)
- 《Amazon Simple Storage Service 使用者指南》中的 [針對 Amazon S3 進行故障診斷](#)。
- 《IAM 使用者指南》中的 [針對 Amazon S3 和 IAM 進行故障診斷](#)

函數參考

函數

- [aws_s3.query_export_to_s3](#)
- [aws_commons.create_s3_uri](#)

aws_s3.query_export_to_s3

將 PostgreSQL 查詢結果匯出至 Amazon S3 儲存貯體。aws_s3 延伸提供 `aws_s3.query_export_to_s3` 函數。

兩個必要參數為 `query` 及 `s3_info`。這些參數會定義要匯出的查詢，以及要匯出至的 Amazon S3 儲存貯體。名為 `options` 的選用參數會提供來定義各種匯出參數。如需使用

`aws_s3.query_export_to_s3` 函數的範例，請參閱 [使用 `aws_s3.query_export_to_s3` 函數匯出查詢資料](#)。

語法

```
aws_s3.query_export_to_s3(  
    query text,  
    s3_info aws_commons._s3_uri_1,  
    options text,  
    kms_key text  
)
```

輸入參數

query

包含 PostgreSQL 引擎執行之 SQL 查詢的必要文字字串。此查詢的結果會複製到 `s3_info` 參數中識別的 S3 儲存貯體。

s3_info

`aws_commons._s3_uri_1` 複合類型，含有下列 S3 物件相關資訊：

- `bucket` – 包含檔案的 Amazon S3 儲存貯體名稱。
- `file_path` – Amazon S3 檔案名稱和路徑。
- `region`— 值 AWS 區所在的區域。如需「AWS 區域」名稱與相關值的清單，請參閱 [區域、可用區域和 Local Zones](#)。

目前，此值必須與匯出執行個體的 AWS 區域相同。預設值為匯出執行個體的 AWS 區域。

如果要建立 `aws_commons._s3_uri_1` 複合結構，請參閱 [aws_commons.create_s3_uri](#) 函數。

options

選用的文字字串，含有 PostgreSQL COPY 命令引數。這些引數指定資料要如何在匯出時複製。詳細資訊請參閱 [PostgreSQL COPY 文件](#)。

替代輸入參數

為了協助進行測試，您可使用一組更大的參數取代 `s3_info` 參數。以下是 `aws_s3.query_export_to_s3` 函數的其他語法變化。

請不要使用 `s3_info` 參數識別 Amazon S3 檔案，而是使用 `bucket`、`file_path` 及 `region` 參數組合進行。

```
aws_s3.query_export_to_s3(  
    query text,  
    bucket text,  
    file_path text,  
    region text,  
    options text,  
)
```

query

包含 PostgreSQL 引擎執行之 SQL 查詢的必要文字字串。此查詢的結果會複製到 `s3_info` 參數中識別的 S3 儲存貯體。

bucket

必要文字字串，其中含有包含檔案的 Amazon S3 儲存貯體名稱。

file_path

包含 Amazon S3 檔案名稱 (包括檔案路徑) 的必要文字字串。

region

包含值區所在 AWS 區域的選擇性文字字串。如需「AWS 區域」名稱與相關值的清單，請參閱[區域、可用區域和 Local Zones](#)。

目前，此值必須與匯出執行個體的 AWS 區域相同。預設值為匯出執行個體的 AWS 區域。

options

選用的文字字串，含有 PostgreSQL COPY 命令引數。這些引數指定資料要如何在匯出時複製。詳細資訊請參閱[PostgreSQL COPY 文件](#)。

輸出參數

```
aws_s3.query_export_to_s3(  
    OUT rows_uploaded bigint,  
    OUT files_uploaded bigint,  
    OUT bytes_uploaded bigint  
)
```

rows_uploaded

指定查詢成功上傳至 Amazon S3 的資料表列數。

files_uploaded

上傳至 Amazon S3 的檔案數。建立的檔案大小約為 6 GB。每個額外建立的檔案，名稱都會加上 `_partXX`。XX 代表 2，接著是 3，視需要以此類推。

bytes_uploaded

上傳至 Amazon S3 的總位元組數。

範例

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'DOC-EXAMPLE-BUCKET', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'DOC-EXAMPLE-BUCKET', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'DOC-EXAMPLE-BUCKET', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

建立 `aws_commons._s3_uri_1` 結構以保留 Amazon S3 檔案資訊。您使用 `aws_commons.create_s3_uri` 函數 `s3_info` 參數之中的 [aws_s3.query_export_to_s3](#) 函數結果。如需使用 `aws_commons.create_s3_uri` 函數的範例，請參閱 [指定要匯出的 Amazon S3 檔案路徑](#)。

語法

```
aws_commons.create_s3_uri(  
    bucket text,  
    file_path text,  
    region text  
)
```

輸入參數

bucket

必要的文字字串，其中含有檔案的 Amazon S3 儲存貯體名稱。

file_path

包含 Amazon S3 檔案名稱 (包括檔案路徑) 的必要文字字串。

region

包含檔案所在 AWS 區域的必要文字字串。如需「AWS 區域」名稱與相關值的清單，請參閱[區域、可用區域和 Local Zones](#)。

從叫用 AWS Lambda 函數

AWS Lambda 是一項事件驅動的運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。它可與許多 AWS 服務搭配使用，包括適用於。例如，您可以使用 Lambda 函數處理來自資料庫的事件通知，或在新檔案上傳到 Simple Storage Service (Amazon S3) 時從檔案中載入資料。若要進一步了解 Lambda，請參閱[什麼是 AWS Lambda？](#) 在 AWS Lambda 開發人員指南中。

Note

以下 RDS 版本支援叫用 AWS Lambda 函數：

- PostgreSQL 本
- 所有 PostgreSQL 15 版本
- PostgreSQL 14.1 版和更高次要版本
- PostgreSQL 13.2 版和更新次要版本
- PostgreSQL 12.6 版和更新次要版本

以下提供必要步驟的摘要。

如需有關 Lambda 函數的詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 入門](#) 和 [AWS Lambda 函數](#)。

主題

- [步驟 1：為輸出連 PostgreSQL 資料庫執行個體 AWS Lambda](#)
- [步驟 2：為您的 RDS 資料庫執行個體和 AWS Lambda](#)
- [步驟 3：為 RDS for PostgreSQL 資料庫執行個體安裝 aws_lambda 擴充功能](#)
- [步驟 4：搭配 RDS for PostgreSQL 資料庫執行個體使用 Lambda helper 函數 \(選用\)](#)
- [步驟 5：從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數。](#)
- [步驟 6：授予其他使用者呼叫 Lambda 函數的許可權限](#)
- [範例：從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數](#)
- [Lambda 函數錯誤訊息](#)
- [AWS Lambda 函數和參數參考](#)

步驟 1：為輸出連 stgreSQL 資料庫執行個體 AWS Lambda

Lambda 函數一律在 AWS Lambda 服務擁有的 Amazon VPC 內執行。Lambda 會將網路存取和安全性規則套用至此 VPC，並自動維護和監控 VPC。您的 RDS for PostgreSQL 資料庫執行個體會將網路流量傳送到 Lambda 服務的 VPC。具體設定取決於 資料庫執行個體是公有的或私有。

- 公有 RDS for PostgreSQL 資料庫執行個體 — 如果個體位於 VPC 的公有子網路中，且執行個體的 `PubliclyAccessible` 內容為 `true`，則該執行個體為公用資料庫執行個體。若要尋找此屬性的值，您可以使用 [描述-db 執行個體](#) AWS CLI 指令。或者可以使用 AWS Management Console 開啟 Connectivity & security (連線與安全性) 索引標籤，檢查 Publicly accessible (可公開存取) 是否為 Yes (是)。若要驗證執行個體是否在您的 VPC 公有子網路中，您可以使用 AWS Management Console 或 AWS CLI。

若要設定 Lambda 的存取權，您可以使 AWS CLI 用 AWS Management Console 或在 VPC 的安全群組上建立輸出規則。傳出規則會指定 TCP 可以使用連接埠 443 將封包傳至任何 IPv4 位址 (0.0.0.0/0)。

- 私有 RDS (適用於 PostgreSQL 資料庫執行個體) — 在此情況下，執行個體的 "PubliclyAccessible" 內容為 `false` 或位於私有子網路中。若要允許執行個體使用 Lambda，您可以使用網路位址轉譯 (NAT) 閘道。如需更多詳細資訊，請參閱 [NAT 閘道](#)。或者，您可以使用 Lambda 的 VPC 端點來設定 VPC。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [VPC 端點](#)。端點將 RDS for PostgreSQL 資料庫執行個體所發出呼叫的回傳至 Lambda 函數。VPC 端點使用自己的私有 DNS 解析。RDS for PostgreSQL 無法使用 Lambda VPC 端點，除非您將 `rds.custom_dns_resolution` 從預設值 0 (未啟用) 改為 1。若要這麼做：

- 建立自訂資料庫參數群組。
- 將參數 `rds.custom_dns_resolution` 從預設值 0 變更為 1。
- 修改資料庫執行個體以使用自訂資料庫參數群組。
- 為了讓修改過的參數生效，請重新啟動執行個體。

您的 VPC 現在可以在網路層級與 AWS Lambda VPC 互動。下一步，您需要使用 IAM 來設定許可。

步驟 2：為您的 RDS 資料庫執行個體和 AWS Lambda

從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數需要某些權限。若要設定必要的權限，建議您建立允許叫用 Lambda 函數的 IAM 政策，將政策指派給一個角色，然後將該角色套用至資料庫執行個體。此做法會提供資料庫執行個體權限，允許代表您叫用指定的 Lambda 函數。下列步驟說明如何在 AWS CLI 中執行此操作。

設定 IAM 許可可以搭配 Lambda 使用 Amazon RDS 執行個體

1. 使用 [建立政策 AWS CLI 命令建立 IAM 政策](#)，以允許您的執行個體叫用指定的 Lambda 函數。(陳述式 ID (Sid) 是政策陳述式的選用描述，不會影響使用。) 此政策為資料庫執行個體提供叫用指定 Lambda 函數所需的最低許可。

```
aws iam create-policy --policy-name rds-lambda-policy --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToExampleFunction",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:444455556666:function:my-function"
    }
  ]
}'
```

或者，您可以使用預先定義的 `AWSLambdaRole` 政策，該政策允許叫用任何 Lambda 函數。如需詳細資訊，請參閱 [適用於 Lambda 的身分型 IAM 政策](#)。

2. 使用 [建立角色 AWS CLI 命令](#) 建立政策可在執行階段承擔的 IAM 角色。

```
aws iam create-role --role-name rds-lambda-role --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

3. 使用 [附件角色原則命令將原則套用至角色 AWS CLI](#)。

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::444455556666:policy/rds-lambda-policy \
  --role-name rds-lambda-role --region aws-region
```

4. AWS CLI 最後這個步驟允許資料庫執行個體的資料庫使用者叫用 Lambda 函數。

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier my-instance-name \  
  --feature-name Lambda \  
  --role-arn arn:aws:iam::444455556666:role/rds-lambda-role \  
  --region aws-region
```

完成 VPC 和 IAM 設定後，現在可以安裝 `aws_lambda` 擴充功能。(請注意，您可以隨時安裝擴充功能，但在設定正確的 VPC 支援和 IAM 權限之前，`aws_lambda` 擴充功能不會為 RDS for PostgreSQL 資料庫執行個體的功能新增任何項目。)

步驟 3：為 RDS for PostgreSQL 資料庫執行個體安裝 `aws_lambda` 擴充功能

若要 AWS Lambda 與您的 RDS 資料庫執行個體搭配使用，請將 PostgreSQL 擴充功能新增至您的資料庫叢集 RDS 資料執行個體。`aws_lambda` 此擴充功能讓 RDS for PostgreSQL 資料庫執行個體能夠從 PostgreSQL 呼叫 Lambda 函數。

在 RDS for PostgreSQL 資料庫執行個體中安裝 `aws_lambda` 擴充功能

使用 PostgreSQL `psql` 命令列或 `pgAdmin` 工具連線到 RDS for PostgreSQL 資料庫執行個體。

1. 以具有 `rds_superuser` 權限的使用者身分連線至 RDS for PostgreSQL 資料庫執行個體。預設 `postgres` 使用者顯示於範例中。

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. 安裝 `aws_lambda` 擴充功能。另外也需要 `aws_commons` 擴充功能。它為 `aws_lambda` 提供了 helper 函數和 PostgreSQL 的許多其他 Aurora 擴充功能。如果尚未裝在 RDS for PostgreSQL 資料庫執行個體上，會透過 `aws_lambda` 進行安裝，如下所示。

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;  
NOTICE: installing required extension "aws_commons"  
CREATE EXTENSION
```

`aws_lambda` 擴充功能已安裝在 資料庫執行個體上。您現在可以建立便利的結構，用於叫用 Lambda 函數。

步驟 4：搭配 RDS for PostgreSQL 資料庫執行個體使用 Lambda helper 函數 (選用)

您可以在 `aws_commons` 擴充功能中使用 helper 函數，準備可更輕鬆從 PostgreSQL 叫用的實體。若要執行此操作，您需要以下有關 Lambda 函數的資訊：

- 函數名稱 – Lambda 函數的名稱、Amazon 資源名稱 (ARN)、版本或別名。在 [步驟 2：為執行個體和 Lambda 設定 IAM](#) 中建立的 IAM 政策需要 ARN，因此建議您使用函數的 ARN。
- AWS 區域 — (選用) 如果 Lambda 函數所在的 AWS 區域與 Aurora PostgreSQL 資料庫叢集 RDS 執行個體不在相同的區域。

若要保存 Lambda 函數名稱資訊，可使用 [aws_commons.create_lambda_function_arn](#) 函數。此 helper 函數會建立一個 `aws_commons._lambda_function_arn_1` 複合結構，其中包含叫用函數所需的詳細資訊。接下來說明設定此複合結構的三種替代做法。

```
SELECT aws_commons.create_lambda_function_arn(  
    'my-function',  
    'aws-region'  
) AS aws_lambda_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(  
    '111122223333:function:my-function',  
    'aws-region'  
) AS lambda_partial_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(  
    'arn:aws:lambda:aws-region:111122223333:function:my-function'  
) AS lambda_arn_1 \gset
```

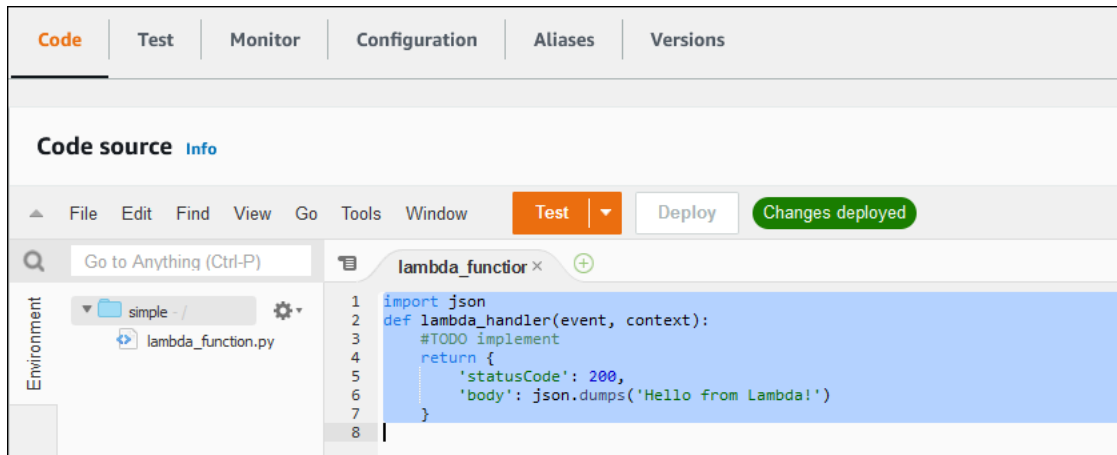
這些值全部都可以用於 [aws_lambda.invoke](#) 函數呼叫。如需範例，請參閱 [步驟 5：從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數](#)。

步驟 5：從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數。

`aws_lambda.invoke` 函數採同步或異步行為，具體取決於 `invocation_type`。此參數的兩個替代項目是 `RequestResponse` (預設值) 和 `Event`，如下所示：

- **RequestResponse** – 此叫用類型為同步。這是在未指定叫用類型的情況下進行呼叫時的預設行為。回應承載包括 `aws_lambda.invoke` 函數的結果。如果您的工作流程需要接收 Lambda 函數的結果才能繼續執行，請使用此叫用類型。
- **Event** – 此叫用類型為非同步。回應不包括含有結果的承載。如果您的工作流程不需要 Lambda 函數的結果即可繼續執行，請使用此叫用類型。

若要簡單測試您的設定，可以使用 `psql` 連線至資料庫執行個體，並從命令列叫用範例函數。假設您在 Lambda 服務上設定了一個基本函數，例如下方螢幕擷取畫面中顯示的簡單 Python 函數。



叫用範例函數

1. 使用 `psql` 或 `pgAdmin` 連線至資料庫執行個體。

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. 使用函數的 ARN 叫用函數。

```
SELECT * from
  aws_lambda.invoke(aws_commons.create_lambda_function_arn('arn:aws:lambda:aws-region:444455556666:function:simple', 'us-west-1'), '{"body": "Hello from
  Postgres!'}'::json );
```

回應如下所示。

```
status_code |                               payload                               |
executed_version | log_result
-----+-----
```

```
200 | {"statusCode": 200, "body": "\"Hello from Lambda!\\""} | $LATEST
|
(1 row)
```

如果您的叫用嘗試未成功，請參閱 [Lambda 函數錯誤訊息](#)。

步驟 6：授予其他使用者呼叫 Lambda 函數的許可權限

在程序中的這一點上，只有身為 `rds_superuser` 的您可以叫用 Lambda 函式。如要允許其他使用者呼叫您建立的任何函數，您需要授予其許可權限。

如要授予叫用 Lambda 函數的許可權限

1. 使用 `psql` 或 `pgAdmin` 連線至資料庫執行個體。

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. 執行下列 SQL 命令：

```
postgres=> GRANT USAGE ON SCHEMA aws_lambda TO db_username;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA aws_lambda TO db_username;
```

範例：從 RDS for PostgreSQL 資料庫執行個體叫用 Lambda 函數

以下提供幾個呼叫 [aws_lambda.invoke](#) 函數的範例。大多數範例都使用您在中建立 `aws_lambda_arn_1` 的複合結構 [步驟 4：搭配 RDS for PostgreSQL 資料庫執行個體使用 Lambda helper 函數 \(選用\)](#) 來簡化傳遞函數詳細資訊。如需非同步叫用的範例，請參閱 [範例：Lambda 函數的非同步 \(Event\) 叫用](#)。列出的所有其他範例都使用同步叫用。

若要進一步了解 Lambda 叫用類型，請參閱《AWS Lambda 開發人員指南》中的 [叫用 Lambda 函數](#)。如需 `aws_lambda_arn_1` 的詳細資訊，請參閱 [aws_commons.create_lambda_function_arn](#)。

範例清單

- [範例：Lambda 函數的同步 \(RequestResponse\) 叫用](#)
- [範例：Lambda 函數的非同步 \(Event\) 叫用](#)
- [範例：在函數回應中擷取 Lambda 執行日誌](#)
- [範例：在 Lambda 函數中包含用戶端內容](#)

- [範例：叫用特定版本的 Lambda 函數](#)

範例：Lambda 函數的同步 (RequestResponse) 叫用

下面是同步 Lambda 函數叫用的兩個範例。這些 `aws_lambda.invoke` 函數的結果相同。

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json);
```

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json, 'RequestResponse');
```

參數說明如下：

- `'aws_lambda_arn_1'` – 此參數識別 [步驟 4：搭配 RDS for PostgreSQL 資料庫執行個體使用 Lambda helper 函數 \(選用\)](#) 中使用 `aws_commons.create_lambda_function_arn` helper 函數建立的複合結構。您也可以在此 `aws_lambda.invoke` 呼叫中建立這個結構，如下所示。

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function',  
'aws-region'),  
'{"body": "Hello from Postgres!"}'::json  
);
```

- `'{"body": "Hello from PostgreSQL!"}'::json` – 要傳遞給 Lambda 函數的 JSON 承載。
- `'RequestResponse'` – Lambda 叫用類型。

範例：Lambda 函數的非同步 (Event) 叫用

以下是非同步 Lambda 函數叫用的範例。Event 叫用類型會使用指定的輸入承載，來排程 Lambda 函數叫用並立即傳回。在某些工作流程中，使用不依賴於 Lambda 函數結果的 Event 叫用類型。

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json, 'Event');
```

範例：在函數回應中擷取 Lambda 執行日誌

在 `aws_lambda.invoke` 函數呼叫中使用 `log_type` 參數，即可讓函數回應中包含執行日誌的最後 4 KB。此參數預設為 `None`，但您可指定 `Tail` 以在回應中擷取 Lambda 執行日誌的結果，如下所示。


```
SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':::json,
'RequestResponse', 'Tail');
```

將 [aws_lambda.invoke](#) log_type 函數的 Tail 參數設定為在回應中包含執行日誌。log_type 參數的預設值為 None。

log_result 傳回的是 base64 編碼字串。您可以使用 decode 和 convert_from PostgreSQL 函數的組合，來解碼內容。

如需 log_type 的詳細資訊，請參閱 [aws_lambda.invoke](#)。

範例：在 Lambda 函數中包含用戶端內容

aws_lambda.invoke 函數具有 context 參數，可用來獨立於承載之外傳遞資訊，如下所示。

```
SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':::json,
'RequestResponse', 'Tail');
```

若要包含用戶端內容，請使用 JSON 物件作為 [aws_lambda.invoke](#) 函數的 context 參數。

如需 context 參數的詳細資訊，請參閱 [aws_lambda.invoke](#) 參考。

範例：叫用特定版本的 Lambda 函數

使用 aws_lambda.invoke 呼叫加入 qualifier 參數，即可指定特定版本的 Lambda 函數。以下提供使用 '*custom_version*' 作為版本別名的範例。

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
Postgres!"}':::json, 'RequestResponse', 'None', NULL, 'custom_version');
```

也可改為提供含有函數名稱的 Lambda 函數限定詞，如下所示。

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-
function:custom_version', 'us-west-2'),
'{"body": "Hello from Postgres!"}':::json);
```

如需 qualifier 和其他參數的詳細資訊，請參閱 [aws_lambda.invoke](#) 參考。

Lambda 函數錯誤訊息

於下列清單中，您可找到有關錯誤訊息的資訊，及可能的原因和解決方案。

- VPC 組態問題

嘗試連線時，VPC 組態問題可能會引發下列錯誤訊息：

```
ERROR: invoke API failed
DETAIL: AWS Lambda client returned 'Unable to connect to endpoint'.
CONTEXT: SQL function "invoke" statement 1
```

此錯誤的常見原因是未正確設定 VPC 安全群組。務必在您的 VPC 安全群組連接埠 443 上開啟 TCP 的傳出規則，讓 VPC 可連線至 Lambda VPC。

如果您的資料庫執行個體是私有的，請檢查 VPC 的私有 DNS 設定。請確定您將 `rds.custom_dns_resolution` 參數設定 AWS PrivateLink 為 1，並按照中所述的方式進行設定 [步驟 1：為輸出連 stgreSQL 資料庫執行個體 AWS Lambda](#)。如需詳細資訊，請參閱 [介面 VPC 端點 \(AWS PrivateLink\)](#)。

- 缺乏叫用 Lambda 函式所需的許可權限

若您看到下列其中一個錯誤訊息，則叫用該函數的使用者 (角色) 並無適當的許可權限。

```
ERROR: permission denied for schema aws_lambda
```

```
ERROR: permission denied for function invoke
```

使用者 (角色) 必須取得特定授權才可叫用 Lambda 函數。如需詳細資訊，請參閱 [步驟 6：授予其他使用者呼叫 Lambda 函數的許可權限](#)。

- 不正確處理 Lambda 函數中的錯誤

如果 Lambda 函數在請求處理期間拋出異常，則 `aws_lambda.invoke` 會失敗並顯示如下所示 PostgreSQL 錯誤。

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
Postgres!"} '::json);
ERROR: lambda invocation failed
```

```
DETAIL: "arn:aws:lambda:us-west-2:555555555555:function:my-function" returned error
"Unhandled", details: "<Error details string>".
```

請務必處理 Lambda 函數或 PostgreSQL 應用程式中的錯誤。

AWS Lambda 函數和參數參考

以下是函數和參數用於調用 Lambda 與 的參考。

函數和參數

- [aws_lambda.invoke](#)
- [aws_commons.create_lambda_function_arn](#)
- [aw_lambda 參數](#)

aws_lambda.invoke

為 RDS for PostgreSQL 資料庫執行個體執行 Lambda 函數。

如需有關叫用 Lambda 函數的詳細資訊，請參閱 AWS Lambda 開發人員指南中的[叫用](#)。

語法

JSON

```
aws_lambda.invoke(
  IN function_name TEXT,
  IN payload JSON,
  IN region TEXT DEFAULT NULL,
  IN invocation_type TEXT DEFAULT 'RequestResponse',
  IN log_type TEXT DEFAULT 'None',
  IN context JSON DEFAULT NULL,
  IN qualifier VARCHAR(128) DEFAULT NULL,
  OUT status_code INT,
  OUT payload JSON,
  OUT executed_version TEXT,
  OUT log_result TEXT)
```

```
aws_lambda.invoke(
```

```
IN function_name aws_commons._lambda_function_arn_1,  
IN payload JSON,  
IN invocation_type TEXT DEFAULT 'RequestResponse',  
IN log_type TEXT DEFAULT 'None',  
IN context JSON DEFAULT NULL,  
IN qualifier VARCHAR(128) DEFAULT NULL,  
OUT status_code INT,  
OUT payload JSON,  
OUT executed_version TEXT,  
OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(  
IN function_name TEXT,  
IN payload JSONB,  
IN region TEXT DEFAULT NULL,  
IN invocation_type TEXT DEFAULT 'RequestResponse',  
IN log_type TEXT DEFAULT 'None',  
IN context JSONB DEFAULT NULL,  
IN qualifier VARCHAR(128) DEFAULT NULL,  
OUT status_code INT,  
OUT payload JSONB,  
OUT executed_version TEXT,  
OUT log_result TEXT)
```

```
aws_lambda.invoke(  
IN function_name aws_commons._lambda_function_arn_1,  
IN payload JSONB,  
IN invocation_type TEXT DEFAULT 'RequestResponse',  
IN log_type TEXT DEFAULT 'None',  
IN context JSONB DEFAULT NULL,  
IN qualifier VARCHAR(128) DEFAULT NULL,  
OUT status_code INT,  
OUT payload JSONB,  
OUT executed_version TEXT,  
OUT log_result TEXT  
)
```

輸入參數

function_name

識別 Lambda 函數的名稱。該值可以是函數名稱、ARN 或部分 ARN。如需適用格式的清單，請參閱 AWS Lambda 開發人員指南中的 [Lambda 函數名稱格式](#)。

payload

Lambda 函數的輸入。格式可以是 JSON 或 JSONB。如需詳細資訊，請參閱 PostgreSQL 文件中的 [JSON 類型](#)。

區域

(選用) 函數的 Lambda 區域。根據預設，RDS 會從 AWS 中的完整 ARN 解析 function_name 區域，或使用 RDS for PostgreSQL 資料庫執行個體區域。如果此區域值與 function_name ARN 中提供的值衝突，則會引發錯誤。

invocation_type

Lambda 函數的叫用類型。值會區分大小寫。可能的值包括以下：

- RequestResponse – 預設值。Lambda 函數的這種叫用類型是同步的，並在結果中傳回回應承載。若您的工作流程依賴於立即接收 Lambda 函數，請使用 RequestResponse 叫用類型。
- Event – Lambda 函數這種叫用類型是非同步的，並且在立即傳回時不含承載。若您在工作流程繼續進行之前不需要 Lambda 函數的結果，請使用 Event 叫用類型。
- DryRun – 這種類型的叫用會測試存取而不執行該 Lambda 函數。

log_type

log_result 輸出參數中要傳回的 Lambda 日誌類型。值會區分大小寫。可能的值包括以下：

- 結尾 – 傳回的 log_result 輸出參數會包含執行日誌的最後 4 KB。
- 無 – 沒有傳回 Lambda 日誌資訊。

context

JSON 或 JSONB 格式的用戶端內容。要使用的欄位包括 custom 和 env。

限定詞

識別待叫用 Lambda 函數版本的限定詞。如果該值與 function_name ARN 中提供的值衝突，則會引發錯誤。

輸出參數

status_code

HTTP 狀態回應代碼。如需更多詳細資訊，請參閱 AWS Lambda 開發人員指南中的 [Lambda 叫用回應元素](#)。

payload

從執行的 Lambda 函數傳回的資訊。格式是 JSON 或 JSONB。

executed_version

Lambda 函數執行的版本。

log_result

如果 log_type 值是叫用 Lambda 函數時的 Tail，則會傳回執行日誌資訊。結果包含以 Base64 編碼的執行日誌的最後 4 KB。

aws_commons.create_lambda_function_arn

建立 aws_commons._lambda_function_arn_1 結構來保存 Lambda 函數名稱資訊。您可以在 aws_lambda.invoke aws_commons.create_lambda_function_arn 函數的 function_name 參數中，使用 [aws_lambda.invoke](#) 函數的結果。

語法

```
aws_commons.create_lambda_function_arn(  
    function_name TEXT,  
    region TEXT DEFAULT NULL  
)  
RETURNS aws_commons._lambda_function_arn_1
```

輸入參數

function_name

包含 Lambda 函數名稱的必要文字字串。該值可以是函數名稱、部分 ARN 或完整 ARN。

區域

選用文字字串，其中含有 Lambda 函數所在的 AWS 區域。如需區域名稱和相關聯值的清單，請參閱 [區域](#)、[可用區域](#)和 [Local Zones](#)。

aws_lambda 參數

在此表格中，您可以找到與aws_lambda函數相關聯的參數。

參數	描述
aws_lambda.connect_timeout_ms	這是一個動態參數，它設置連接到 AWS Lambda 時的最長等待時間。預設值為1000。此參數的允許值為 1-900000。
aws_lambda.request_timeout_ms	這是一個動態參數，它設置等待 AWS Lambda 響應時的最長等待時間。預設值為3000。此參數的允許值為 1-900000。
aws_lambda.endpoint_override	指定可用於連線至 AWS Lambda 的端點。空字串會選取該區域的預設 AWS Lambda 端點。您必須重新啟動資料庫，此靜態參數變更才會生效。

Amazon RDS for PostgreSQL 的常用 DBA 任務

資料庫管理員 (DBA) 在管理 Amazon RDS for PostgreSQL 資料庫執行個體時執行各種任務。若您是已經熟悉 PostgreSQL 的 DBA，則需要了解在硬體上執行 PostgreSQL 和 RDS for PostgreSQL 之間的一些重要差異。例如，由於 Amazon RDS 為受管服務，因此不允許 Shell 存取資料庫執行個體。這表示您無法直接存取 `pg_hba.conf` 和其他組態檔。對於 RDS for PostgreSQL，會對與 RDS for PostgreSQL 資料庫執行個體相關聯的自訂資料庫參數群組，進行通常對內部部署執行個體的 PostgreSQL 組態檔所做的變更。如需詳細資訊，請參閱 [使用參數群組](#)。

您也無法以您對內部部署 PostgreSQL 執行個體所做的相同方式存取日誌檔。如要進一步了解記錄，請參閱 [RDS for PostgreSQL 資料庫日誌檔](#)。

如同另一個範例，您無法存取 PostgreSQL superuser 帳戶。在 RDS for PostgreSQL 上，`rds_superuser` 角色是最具特權的角色，且其在設定時間授予 `postgres`。無論您是熟悉於內部部署使用 PostgreSQL 還是對 RDS for PostgreSQL 完全陌生，我們都建議您了解 `rds_superuser` 角色，及如何使用角色、使用者、群組和許可權限。如需詳細資訊，請參閱 [了解 PostgreSQL 角色和許可](#)。

下文將介紹 RDS for PostgreSQL 的一些常用 DBA 任務。

主題

- [RDS for PostgreSQL 中支援定序](#)
- [了解 PostgreSQL 角色和許可](#)
- [在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理](#)
- [搭配使用 RDS for PostgreSQL 支援的記錄機制](#)
- [使用 PostgreSQL 管理暫存檔案](#)
- [使用 pgBadger 進行 PostgreSQL 的日誌分析](#)
- [使用 PgSNAPPER 監控 PostgreSQL](#)
- [在 RDS for PostgreSQL 資料庫執行個體上搭配使用參數](#)

RDS for PostgreSQL 中支援定序

定序是一組規則，用來決定如何排序和比較存放在資料庫中的字元字串。定序在電腦系統中扮演著基本角色，並包含為作業系統的一部分。隨著新字元加入至語言或排序規則變更時，定序也會變更。

定序程式庫會定義定序的特定規則和演算法。PostgreSQL 內使用的最熱門定序程式庫是 GNU C (glibc)和 Unicode 國際化元件 (ICU)。依預設，RDS for PostgreSQL 會使用 glibc 定序，其中包含多位元組字元序列的 Unicode 字元排序順序。

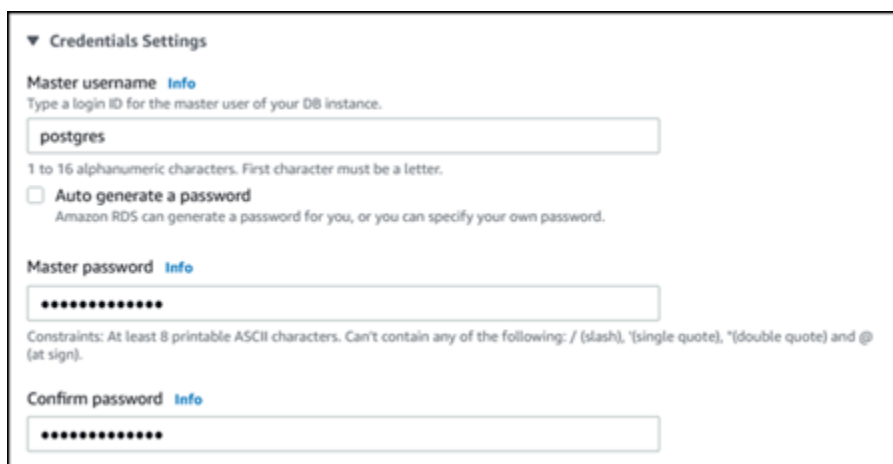
當您在 RDS for PostgreSQL 中建立新的資料庫執行個體時，它會檢查作業系統是否有可用的定序。CREATE DATABASE 命令 LC_COLLATE 和 LC_CTYPE 的 PostgreSQL 參數用來指定一個定序，其作為該資料庫中的預設定序。或者，您也可以使用 CREATE DATABASE 中的 LOCALE 參數來設定這些參數。這會決定資料庫中字元字串的預設定序，以及將字元分類為字母、數字或符號的規則。您也可以選擇要在資料欄、索引或查詢上使用的定序。

RDS for PostgreSQL 取決於作業系統中的 glibc 程式庫是否支援定序。RDS for PostgreSQL 執行個體會定期使用最新版本的作業系統進行更新。這些更新有時會包含較新版本的 glibc 程式庫。較新版本的 glibc 很少會變更某些字元的排序順序或定序，這可能導致資料以不同的方式排序或產生無效的索引項目。如果您在更新期間發現定序的排序順序問題，您可能需要重建索引。

為了 glibc 更新可能造成的影響，RDS for PostgreSQL 現在包含獨立的預設定序程式庫。此定序程式庫可在 RDS for PostgreSQL 14.6、13.9、12.13、11.18、10.23 和較新的次要版本中使用。它與 glibc 2.26-59.amzn2 相容，並提供排序順序穩定性以防止出現不正確的查詢結果。

了解 PostgreSQL 角色和許可

當您使用建立 RDS 資料庫執行個體時 AWS Management Console，系統會同時建立一個管理員帳戶。依預設，其名稱為 postgres，如下列螢幕擷取畫面所示：



▼ Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. First character must be a letter.

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm password [Info](#)

您可以選擇其他名稱，而不是接受預設值 (postgres)。若是如此，則選擇的名稱必須以字母開頭，且介於 1 到 16 個英數字元之間。為簡單起見，我們在本指南中使用主要使用者帳戶的預設值 (postgres)，來指稱主要使用者帳戶。

如果您使用 `create-db-instance` AWS CLI 而不是 AWS Management Console，您可以透過在指令中傳遞 `master-username` 參數來建立名稱。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

無論您使用 AWS Management Console、或 Amazon RDS API AWS CLI，以及使用預設 `postgres` 名稱還是選擇其他名稱，此第一個資料庫使用者帳戶都是該 `rds_superuser` 群組的成員，並具有 `rds_superuser` 權限。

主題

- [了解 `rds_superuser` 角色](#)
- [控制使用者對 PostgreSQL 資料庫的存取](#)
- [委派和控制使用者密碼管理](#)
- [使用 SCRAM 進行 PostgreSQL 密碼加密](#)

了解 `rds_superuser` 角色

於 PostgreSQL 中，角色可定義使用者、群組或授予群組或使用者對資料庫中各種物件的一組特定權限。CREATE USER 和 CREATE GROUP 的 PostgreSQL 命令已為更為通用的 CREATE ROLE 所取代，其具有可區分資料庫使用者的特定屬性。資料庫使用者可視為具有 LOGIN 權限的角色。

Note

CREATE USER 和 CREATE GROUP 命令仍可使用。如需詳細資訊，請參閱 PostgreSQL 文件中的 [資料庫角色](#)。

`postgres` 使用者是 RDS for PostgreSQL 資料庫執行個體上具最高權限的資料庫使用者。其具有下列 CREATE ROLE 陳述式所定義的特性。

```
CREATE ROLE postgres WITH LOGIN NOSUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
VALID UNTIL 'infinity'
```

屬性 NOSUPERUSER、NOREPLICATION、INHERIT，和 VALID UNTIL 'infinity' 為 CREATE ROLE 的預設選項，除非另有指定。

根據預設，`postgres` 具有授與 `rds_superuser` 角色的權限，以及建立角色和資料庫的許可。`rds_superuser` 角色可讓 `postgres` 使用者執行下列動作：

- 新增可與 Amazon RDS 搭配使用的擴充功能。如需詳細資訊，請參閱 [使用 Amazon RDS for PostgreSQL 支援的 PostgreSQL 功能](#)
- 建立使用者的角色，並授予使用者權限。如需詳細資訊，請參閱 PostgreSQL 文件中的 [CREATE ROLE](#) 和 [GRANT](#)。
- 建立資料庫。如需詳細資訊，請參閱 PostgreSQL 文件中的 [CREATE DATABASE](#)。
- 將 `rds_superuser` 權限授予並無這些權限的使用者角色，並視需要撤銷這些權限。建議您僅將此角色授予執行超級使用者任務的使用者。換句話說，您可以將此角色授予資料庫管理員 (DBA) 或系統管理員。
- 對不具 `rds_replication` 角色的資料庫使用者授予 (和撤銷) `rds_superuser` 角色。
- 對不具 `rds_password` 角色的資料庫使用者授予 (和撤銷) `rds_superuser` 角色。
- 使用 `pg_stat_activity` 檢視，取得有關所有資料庫連線的狀態資訊。如有需要，`rds_superuser` 可使用 `pg_terminate_backend` 或 `pg_cancel_backend` 停止任何連線。

於 `CREATE ROLE postgres...` 陳述式中，您可看到 `postgres` 使用者角色明確禁止 PostgreSQL `superuser` 權限。RDS for PostgreSQL 為受管服務，因此您無法存取主機作業系統，也無法使用 PostgreSQL `superuser` 帳戶進行連線。許多需要在獨立 PostgreSQL 上進行 `superuser` 存取的任務是由 Amazon RDS 自動管理。

如需有關授予權限的詳細資訊，請參閱 PostgreSQL 文件中的 [GRANT](#)。

`rds_superuser` 角色是中數個預先定義角色的其中一個。RDS for PostgreSQL 資料庫執行個體。

Note

在 PostgreSQL 13 和更早版本中，預先定義角色稱為預設角色。

於下列清單中，您可以找到為新的自動建立的一些其他預先定義角色。RDS for PostgreSQL 資料庫執行個體。預先定義的角色及其權限無法進行變更。您無法為這些預先定義角色停止、重新命名或修改權限。嘗試這麼做會造成錯誤。

- `rds_password` – 可變更密碼並為資料庫使用者設定密碼約束的角色。依預設，此 `rds_superuser` 角色會授與此角色，而且可以將角色授與資料庫使用者。如需詳細資訊，請參閱 [控制使用者對 PostgreSQL 資料庫的存取](#)。
 - 對於 14 版本以上的 RDS for PostgreSQL，`rds_password` 角色可以為資料庫使用者和具 `rds_superuser` 有角色的使用者變更密碼並設定密碼限制。從 RDS for PostgreSQL 版本 14

及更新版本，`rds_password` 角色可以變更密碼並僅針對資料庫使用者設定密碼限制。只有具有 `rds_superuser` 角色的使用者可對具有角色的其他使用者執行這些動 `rds_superuser` 作。

- `rdsadmin` – 為處理具有 `superuser` 權限的管理員將在獨立 PostgreSQL 資料庫上執行的許多管理任務而建立的角色。此角色由 RDS for PostgreSQL 在內部用於許多管理任務。
- `rdstopmgr`— Amazon RDS 內部使用的一個角色，可支援異地同步備份部署。

如要查看所有預先定義角色，您可連接至 您 RDS for PostgreSQL 資料庫執行個體，並使用 `psql \du` 中繼命令。輸出看似如下：

```
List of roles
Role name | Attributes | Member of
-----+-----+-----
postgres | Create role, Create DB | {rds_superuser}
          | Password valid until infinity |
rds_superuser | Cannot login | {pg_monitor,pg_signal_backend,
          | | rds_replication,rds_password}
...

```

於輸出中，您可看到 `rds_superuser` 並非資料庫使用者角色 (無法登入)，但其具有許多其他角色的權限。您還可以看到資料庫使用者 `postgres` 是 `rds_superuser` 角色的成員。如前所述，`postgres` 是 Amazon RDS 主控台 Create database (建立資料庫) 頁面中的預設值。若選擇其他名稱，則該名稱將顯示於角色清單中。

控制使用者對 PostgreSQL 資料庫的存取

PostgreSQL 中的新資料庫會永遠使用資料庫 `public` 結構描述中的一組預設權限建立，允許所有資料庫使用者 and 角色建立物件。例如，這些權限可讓資料庫使用者連接至資料庫，並在連線時建立暫存資料表。

為了對您在 RDS for PostgreSQL 資料庫執行個體上建立之資料庫執行個體的使用者存取進行更好地控制，我們建議您撤消這些預設的 `public` 權限。完成此作業後，您可更精細地為資料庫使用者授予特定權限，如下列程序所示。

如要設定新資料庫執行個體的角色和權限

假設您正在一個剛建立的 RDS for PostgreSQL 資料庫執行個體上設定資料庫，以供多位研究人員使用，其皆需要資料庫的讀寫存取權。

1. 使用 `psql` (或 `pgAdmin`) 連接至 您的 RDS for PostgreSQL 資料庫執行個體：

```
psql --host=your-db-instance.666666666666.aws-region.rds.amazonaws.com --port=5432  
--username=postgres --password
```

出現提示時，輸入您的密碼。psql 用戶端連接並顯示預設管理連接資料庫 postgres=> 作為提示。

2. 如要防止資料庫使用者在 public 結構描述中建立物件，請執行下列動作：

```
postgres=> REVOKE CREATE ON SCHEMA public FROM PUBLIC;  
REVOKE
```

3. 接下來，您會建立新的資料庫執行個體：

```
postgres=> CREATE DATABASE lab_db;  
CREATE DATABASE
```

4. 從此新資料庫上 PUBLIC 結構描述撤消所有權限。

```
postgres=> REVOKE ALL ON DATABASE lab_db FROM public;  
REVOKE
```

5. 建立一個資料庫使用者的角色。

```
postgres=> CREATE ROLE lab_tech;  
CREATE ROLE
```

6. 使具有此角色的資料庫使用者可連接至資料庫。

```
postgres=> GRANT CONNECT ON DATABASE lab_db TO lab_tech;  
GRANT
```

7. 授予具 lab_tech 角色的所有使用者此資料庫的所有權限。

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_db TO lab_tech;  
GRANT
```

8. 建立資料庫使用者，如下所示：

```
postgres=> CREATE ROLE lab_user1 LOGIN PASSWORD 'change_me';  
CREATE ROLE  
postgres=> CREATE ROLE lab_user2 LOGIN PASSWORD 'change_me';
```

```
CREATE ROLE
```

9. 授予這兩個使用者與 lab_tech 角色關聯的權限：

```
postgres=> GRANT lab_tech TO lab_user1;
GRANT ROLE
postgres=> GRANT lab_tech TO lab_user2;
GRANT ROLE
```

至此，lab_user1 和 lab_user2 便可連接 lab_db 資料庫。此範例並未遵循企業使用的最佳實務，其中可能包括建立多個資料庫執行個體、不同的結構描述，及授予有限的權限。如需更多完整資訊和其他方案，請參閱[管理 PostgreSQL 使用者和角色](#)。

如需有關 PostgreSQL 資料庫中權限的詳細資訊，請參閱 PostgreSQL 文件中的 [GRANT](#) 命令。

委派和控制使用者密碼管理

作為 DBA，您可能想要委派管理使用者密碼。或者，您可能想要防止資料庫使用者變更其密碼或重新設定密碼約束 (例如密碼生命週期)。如要確保只有您選擇的資料庫使用者才可變更密碼設定，您可開啟受限制的密碼管理功能。啟動此功能時，只有那些已被授予 rds_password 角色的資料庫使用者才可管理密碼。

Note

如要使用受限制的密碼管理，您的 RDS for PostgreSQL 資料庫執行個體必須執行 PostgreSQL 10.6 或更新版本。

依預設，此功能為 off，如下列所示：

```
postgres=> SHOW rds.restrict_password_commands;
 rds.restrict_password_commands
-----
off
(1 row)
```

如要開啟此功能，請使用自訂參數群組，並將 rds.restrict_password_commands 的設定變更為 1。請務必重新啟動您的 RDS for PostgreSQL 資料庫執行個體，以使設定生效。

啟用此功能後，下列 SQL 命令需要 rds_password 權限：

```
CREATE ROLE myrole WITH PASSWORD 'mypassword';
CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword';
ALTER ROLE myrole VALID UNTIL '2023-01-01';
ALTER ROLE myrole RENAME TO myrole2;
```

若密碼使用 MD5 雜湊演算法，則重新命名角色 (ALTER ROLE myrole RENAME TO newname) 也會受到限制。

啟用此功能後，嘗試使用任何這些 SQL 命令，而不使用 rds_password 角色權限會產生下列錯誤：

```
ERROR: must be a member of rds_password to alter passwords
```

建議您僅對專用於密碼管理的一些角色授予 rds_password。若您授予不具 rds_superuser 權限之資料庫使用者 rds_password 權限，則還需要授予其 CREATEROLE 屬性。

確定您驗證用戶端上的密碼需求，例如到期時間和需要的複雜度。若您使用自己的用戶端公用程式進行與密碼相關的變更，則該公用程式必須為 rds_password 的成員並具有 CREATE ROLE 權限。

使用 SCRAM 進行 PostgreSQL 密碼加密

Salted Challenge Response Authentication Mechanism (SCRAM) 是 PostgreSQL 預設訊息摘要 (MD5) 演算法的替代選項，用於加密密碼。SCRAM 身分驗證機制被認為比 MD5 更安全。若要進一步了解這兩種不同的密碼保護方法，請參閱 PostgreSQL 文件中的 [Password Authentication](#) (密碼身分驗證)。

建議您使用 SCRAM 作為的密碼加密配置，而不是使用 MD5。RDS for PostgreSQL 資料庫執行個體。這是一種加密的挑戰回應機制，使用 scram-sha-256 演算法進行密碼身分驗證和加密。

您可能需要更新程式庫，用戶端應用程式才會支援 SCRAM。例如，42.2.0 之前的 JDBC 版本不支援 SCRAM。如需詳細資訊，請參閱 PostgreSQL JDBC 驅動程式文件中的 [PostgreSQL JDBC Driver](#) (PostgreSQL JDBC 驅動程式)。如需其他 PostgreSQL 驅動程式和 SCRAM 支援的清單，請參閱 PostgreSQL 文件中的 [List of drivers](#) (驅動程式清單)。

Note

RDS for PostgreSQL 13.1 版和更新版本支援 scram-sha-256。這些版本亦允許您將資料庫執行個體設定為需要 SCRAM，如以下程序所述。

設定 RDS for PostgreSQL 資料庫執行個體為需要 SCRAM

，您可以要求 RDS for PostgreSQL 資料庫執行個體僅接受使用 scram-sha-256 演算法的密碼。

Important

對於 PostgreSQL 資料庫的現有 RDS 代理，如果您將資料庫驗證修改為僅使用 SCRAM，則該代理會變成無法使用，最多持續 60 秒。若要避免發生此問題，請執行下列其中一項：

- 確定資料庫同時允許 SCRAM 和 MD5 身分驗證。
- 若只要使用 SCRAM 身分驗證，請建立新代理、將應用程式流量遷移至新代理，然後刪除先前與資料庫相關聯的代理。

在對系統進行變更之前，請確保您了解完整過程，如下所示：

- 獲得所有資料庫使用者的所有角色與密碼加密相關資訊。
- 針對控制密碼加密的參數，再次檢查 RDS for PostgreSQL 資料庫執行個體的參數設定。
- 如果您的 RDS for PostgreSQL 資料庫執行個體使用預設參數群組，則需要建立自訂資料庫參數群組，並將其套用到 RDS for PostgreSQL 資料庫執行個體，讓您可以在需要時修改參數。如果您的 RDS for PostgreSQL 資料庫執行個體使用自訂參數群組，則您稍後可以視需要在過程中修改必要參數。
- 將 `password_encryption` 參數變更為 `scram-sha-256`。
- 通知所有資料庫使用者他們必須更新密碼。針對您的 `postgres` 帳戶進行相同的動作。系統會使用 `scram-sha-256` 演算法加密與儲存新密碼。
- 驗證確認使用加密類型將所有密碼加密。
- 如果所有密碼都使用 `scram-sha-256`，您可以將 `rds.accepted_password_auth_method` 參數從 `md5+scram` 變更為 `scram-sha-256`。

Warning

在您僅將 `rds.accepted_password_auth_method` 變更為 `scram-sha-256` 後，則任何具有 `md5` 加密密碼的使用者 (角色) 將無法連線。

讓您的 RDS for PostgreSQL 資料庫執行個體準備好需要 SCRAM

對 RDS for PostgreSQL 資料庫執行個體進行任何變更之前，請檢查所有現有的資料庫使用者帳戶。另外，請檢查用於密碼的加密類型。您可以使用 `rds_tools` 擴充功能執行這些任務。RDS for PostgreSQL 13.1 與更新版本支援此擴充功能。

獲得資料庫使用者 (角色) 與密碼加密方法清單

1. 使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體，如下所示。

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. 安裝 `rds_tools` 擴充功能。

```
postgres=> CREATE EXTENSION rds_tools;
CREATE EXTENSION
```

3. 取得角色和加密清單。

```
postgres=> SELECT * FROM
           rds_tools.role_password_encryption_type();
```

您會看到類似下列的輸出。

rolname	encryption_type
pg_monitor	
pg_read_all_settings	
pg_read_all_stats	
pg_stat_scan_tables	
pg_signal_backend	
lab_tester	md5
user_465	md5
postgres	md5

(8 rows)

建立自訂資料庫參數群組

Note

如果您的 RDS for PostgreSQL 資料庫執行個體已使用自訂參數群組，則不需要建立新的群組。

如需 Amazon RDS 的參數群組概觀，請參閱 [在 RDS for PostgreSQL 資料庫執行個體上搭配使用參數](#)。

用於密碼的密碼加密類型是在某一個參數 (亦即 `password_encryption`) 中設定的。RDS for PostgreSQL 資料庫執行個體允許的加密則是在另一個參數 (亦即 `rds.accepted_password_auth_method`) 中設定的。若要從這些預設值變更，則需要您建立自訂資料庫參數群組，並套用到您的執行個體。

您也可以使用 AWS Management Console 或 RDS API 建立自訂資料庫參數群組。如需詳細資訊，請參閱

您可以使用資料庫執行個體與自訂參數群組建立關聯。

建立自訂資料庫參數群組

1. 使用 CLI [create-db-parameter-group](#) 命令建立自訂資料庫參數群組。此範例使用 `postgres13` 作為此自訂參數群組的來源。

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group --db-parameter-group-name 'docs-lab-scam-  
passwords' \  
  --db-parameter-group-family postgres13 --description 'Custom parameter group for  
SCRAM'
```

在Windows中：

```
aws rds create-db-parameter-group --db-parameter-group-name "docs-lab-scam-  
passwords" ^  
  --db-parameter-group-family postgres13 --description "Custom DB parameter group  
for SCRAM"
```

2. 使用 [modify-db-instance](#) CLI 命令將此自訂參數群組套用至您的 RDS for PostgreSQL 資料庫叢集。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance --db-instance-identifier 'your-instance-name' \
  --db-parameter-group-name "docs-lab-scram-passwords"
```

在Windows中：

```
aws rds modify-db-instance --db-instance-identifier "your-instance-name" ^
  --db-parameter-group-name "docs-lab-scram-passwords"
```

若要將 RDS for PostgreSQL 資料庫執行個體與自訂資料庫參數群組重新同步，您需要重新啟動叢集的主要與所有其他執行個體。為了盡量降低對使用者的影響，排程在一般維護時段期間進行此操作。

設定密碼加密以使用 SCRAM

RDS for PostgreSQL 資料庫執行個體使用的密碼加密機制是在資料庫參數群組的 `password_encryption` 參數中設定的。允許的值為未設定、md5 或 scram-sha-256。預設值視 RDS for PostgreSQL 版本而定，如下所示：

- RDS for PostgreSQL 14 和更新版本 - 預設值是 scram-sha-256
- RDS for PostgreSQL 13 – 預設值是 md5

透過將自訂資料庫參數群組連接至 RDS for PostgreSQL 資料庫執行個體，您可以修改密碼加密參數的值。

<input type="checkbox"/>	Name	Values	Allowed values	Modifiable	Source	Apply type
<input type="checkbox"/>	password_encryption	md5	md5, scram-sha-256	true	system	dynamic
<input type="checkbox"/>	rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

將密碼加密設定變更為 scram-sha-256

- 將密碼加密的值變更為 scram-sha-256，如下所示。變更會立即套用，因為參數是動態的，因此不需要重新啟動即可使變更生效。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group --db-parameter-group-name \  
  'docs-lab-scram-passwords' --parameters  
  'ParameterName=password_encryption,ParameterValue=scram-  
sha-256,ApplyMethod=immediate'
```

在Windows中：

```
aws rds modify-db-parameter-group --db-parameter-group-name ^  
  "docs-lab-scram-passwords" --parameters  
  "ParameterName=password_encryption,ParameterValue=scram-  
sha-256,ApplyMethod=immediate"
```

將使用者角色的密碼遷移至 SCRAM

您可以將使用者角色的密碼遷移至 SCRAM，如下說明：

將資料庫使用者 (角色) 密碼從 MD5 遷移至 SCRAM

- 以管理員使用者身分 (預設使用者名稱 postgres) 登入，如下所示。

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password
```

- 使用下列命令檢查 RDS for PostgreSQL 資料庫執行個體上 password_encryption 參數的設定。

```
postgres=> SHOW password_encryption;  
password_encryption  
-----  
md5  
(1 row)
```

- 將此參數的值變更為 `scram-sha-256`。這是動態參數，因此您在進行此變更之後不需要重新啟動執行個體。再次檢查該值以確定它現在已設定為 `scram-sha-256`，如下所示。

```
postgres=> SHOW password_encryption;
password_encryption
-----
scram-sha-256
(1 row)
```

- 通知所有資料庫使用者變更自己的密碼。務必亦為帳戶 `postgres` (具有 `rds_superuser` 權限的資料庫使用者) 變更您自己的密碼。

```
labdb=> ALTER ROLE postgres WITH LOGIN PASSWORD 'change_me';
ALTER ROLE
```

- 對上所有的資料庫重複此程序。RDS for PostgreSQL 資料庫執行個體。

變更參數為需要 SCRAM

這是程序中的最後一個步驟。在以下程序中進行變更後，仍然使用 `md5` 將密碼加密的任何使用者帳戶 (角色) 將無法登入 RDS for PostgreSQL 資料庫執行個體。

`rds.accepted_password_auth_method` 指定 RDS for PostgreSQL 資料庫執行個體在登入程序期間接受的使用者密碼加密方法。預設值為 `md5+scram`，這意味著任何一種方法都接受。在下圖中，您可以找到此參數的預設設定。

<input type="checkbox"/>	Name	Values	Allowed values	Modifiable	Source	Apply type
<input type="checkbox"/>	<code>password_encryption</code>	<code>scram-sha-256</code>	<code>md5, scram-sha-256</code>	true	system	dynamic
<input type="checkbox"/>	<code>rds.accepted_password_auth_method</code>	<code>md5+scram</code>	<code>md5+scram, scram</code>	true	system	dynamic

此參數的允許值為 `md5+scram` 或僅 `scram`。將此參數值變更為 `scram` 會使此成為一個需求。

將參數值變更為需要對密碼進行 SCRAM 身分驗證

- 確認 RDS for PostgreSQL 資料庫執行個體上所有資料庫的所有資料庫使用者密碼使用 `scram-sha-256` 進行密碼加密。若要這麼做，請向 `rds_tools` 查詢角色 (使用者) 和加密類型，如下所示。

```
postgres=> SELECT * FROM rds_tools.role_password_encryption_type();
rolname          | encryption_type
-----+-----
pg_monitor       |
pg_read_all_settings |
pg_read_all_stats |
pg_stat_scan_tables |
pg_signal_backend |
lab_tester       | scram-sha-256
user_465         | scram-sha-256
postgres        | scram-sha-256
( rows)
```

2. 針對您的 中所有的資料庫執行個體重複此查詢。RDS for PostgreSQL 資料庫執行個體。

如果所有密碼都使用 scram-sha-256，您可以繼續進行。

3. 將接受的密碼身分驗證值變更為 scram-sha-256，如下所示。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group --db-parameter-group-name 'docs-lab-scram-
passwords' \
  --parameters
  'ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

在Windows中：

```
aws rds modify-db-parameter-group --db-parameter-group-name "docs-lab-scram-
passwords" ^
  --parameters
  "ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理

強烈建議您使用自動資料清理功能，以維護 PostgreSQL 資料庫執行個體的運作狀態。自動資料清理會自動啟動 VACUUM 和 ANALYZE 指令。此功能會檢查含有大量輸入、更新或刪除元組的資料表。完成檢查後，即會透過從 PostgreSQL 資料庫移除已淘汰的資料或元組回收儲存空間。

預設情況下，使用任何預設 PostgreSQL 資料庫參數群組建立的 Amazon RDS for PostgreSQL 資料庫執行個體上，都開啟了自動資料清理功能。其中包括 `default.postgres10`、`default.postgres11`，以及其他版本。所有預設的 PostgreSQL 資料庫參數群組都有設為 1 的 `rds.adaptive_autovacuum` 參數，因此啟用了該功能。預設情況下，還會設定與自動資料清理功能相關聯的其他組態參數。因為這些預設值相當泛用，針對特定的工作負載調校與自動資料清理功能相關聯的某些參數，對您會有所幫助。

接下來，您可以找到更多有關自動資料清理功能以及如何在您的 RDS for PostgreSQL 資料庫執行個體上調校該功能一些參數的資訊。如需高層級資訊，請參閱 [使用 PostgreSQL 的最佳實務](#)。

主題

- [配置自動資料清理的記憶體](#)
- [降低交易 ID 包圍的可能性](#)
- [判斷資料庫中的資料表是否需要清理](#)
- [判斷哪些資料表目前適合進行自動資料清理](#)
- [判斷自動資料清理目前是否執行中且執行多久時間](#)
- [執行手動清理凍結](#)
- [在自動資料清理執行時重新為資料表建立索引](#)
- [使用大型索引管理自動清空](#)
- [影響自動資料清理的其他參數](#)
- [設定自動資料清理參數資料表層級](#)
- [記錄清理和自動資料清理活動](#)

配置自動資料清理的記憶體

`maintenance_work_mem` 參數是影響自動資料清理效能的最重要參數之一。此參數可決定您為自動資料清理配置多少記憶體，其會用來掃描資料庫資料表，以及保存即將清理的所有資料列 ID。如果您將 `maintenance_work_mem` 參數值設得太低，則清理程序可能必須掃描資料表多次才能完成其工作。多次的掃描可能會對效能產生負面影響。

進行計算以決定 `maintenance_work_mem` 參數值時，請記住兩件事：

- 此參數的預設單位為 kilobytes (KB)。
- `maintenance_work_mem` 參數會搭配 [autovacuum_max_workers](#) 參數運作。如果您有許多小型資料表，請配置較多 `autovacuum_max_workers` 和較少 `maintenance_work_mem`。如果您

有大型資料表 (假設大於 100 GB)，請配置較多記憶體和較少背景工作程序。您必須配置足夠的記憶體，才能在最大的資料表上順利執行作業。每個 `autovacuum_max_workers` 可以使用您配置的記憶體。因此，請確保工作者程序與記憶體的組合等於您想要配置的總記憶體。

一般而言，針對大型主機，將 `maintenance_work_mem` 參數設定為介於 1 與 2 GB 之間的值 (介於 1,048,576 與 2,097,152 KB 之間)。針對極大型主機，將此參數設定為介於 2 與 4 GB 之間的值 (介於 2,097,152 與 4,194,304 KB 之間)。您為此參數設定的值會視工作負載而定。Amazon RDS 已更新此參數的預設值 (以 KB 為單位)，計算方式如下。

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536).
```

降低交易 ID 包圍的可能性

在一些狀況中，自動資料清理相關的參數群組設定可能不夠積極以防止交易 ID 包圍。若要解決此問題，RDS for PostgreSQL 提供自動調整自動資料清理參數值的機制。自動資料清理參數彈性調校是 RDS for PostgreSQL 的功能。PostgreSQL 文件中有詳細的 [交易 ID 包圍](#) 的說明。

預設開啟 RDS for PostgreSQL 執行個體的自動資料清理參數彈性調校功能，且動態參數 `rds.adaptive_autovacuum` 設定為 ON (開啟)。我們強烈建議您開啟此選項。不過，若要關閉參數彈性調校功能，請將參數 `rds.adaptive_autovacuum` 設定為 0 或 OFF (關閉)。

Amazon RDS 調校自動資料清理參數時，交易 ID 迴繞仍可能發生。我們建議您為交易 ID 環繞實作 Amazon CloudWatch 警示。如需詳細資訊，請參閱 AWS 資料庫部落格文章 [Implement an early warning system for transaction ID wraparound in RDS for PostgreSQL](#) (針對 RDS for PostgreSQL 中的交易 ID 迴繞實作預警系統)。

開啟自適應自動真空參數調整後，Amazon RDS 會在 CloudWatch 指標 `MaximumUsedTransactionIDs` 達到參數值或 500,000,000 (以較大者為準) 時開始調整自動真空 `autovacuum_freeze_max_age` 參數。

如果資料表繼續有交易 ID 包圍的趨勢，則 Amazon RDS 會繼續調整自動資料清理的參數。每次調整都提供更多自動資料清理的資源以避免包圍。Amazon RDS 會更新下列自動資料清理相關參數：

- [autovacuum_vacuum_cost_delay](#)
- [autovacuum_vacuum_cost_limit](#)
- [autovacuum_work_mem](#)
- [autovacuum_naptime](#)

RDS 只會在新的值能夠使自動資料清理更為積極時修改這些參數。參數在資料庫執行個體上的記憶體中修改。參數群組中的值不會變更。若要檢視目前的使用中記憶體設定，請使用 PostgreSQL [SHOW \(顯示\)](#) SQL 指令。

當 Amazon RDS 修改任何自動資料清理參數時，即會對受影響的資料庫執行個體產生事件。此事件會顯示在 AWS Management Console 上，也會透過 Amazon RDS API 顯示。指 `MaximumUsedTransactionIDs` CloudWatch 標返回低於閾值後，Amazon RDS 會將記憶體中的自動真空相關參數重設回參數群組中指定的值。系統隨即會產生與此變更相對應的另一個事件。

判斷資料庫中的資料表是否需要清理

您可以使用以下查詢可用來顯示資料庫中未清理的交易數目。資料庫 `datfrozenxid` 列的 `pg_database` 欄是顯示於該資料庫正常交易 ID 的下線。此欄位是資料庫內每個資料表 `relfrozenxid` 最小的值。

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc
limit 20;
```

例如，執行上述查詢的結果可能如下所示。

```
datname      | age
mydb         | 1771757888
template0    | 1721757888
template1    | 1721757888
rdsadmin     | 1694008527
postgres     | 1693881061
(5 rows)
```

當交易 ID 包圍的存留期觸及 20 億，則會發生交易 ID 包圍 (XID)，且資料庫將變成唯獨狀態。您可使用此查詢來產生指標且一天執行數次。根據預設，自動資料清理已將交易的存留期設定為不超過 200,000,000 ([autovacuum_freeze_max_age](#))。

範例監視策略可能如下所示：

- 設定 `autovacuum_freeze_max_age` 值到 2 億交易數目
- 如果資料表達到 5 億的未清理交易，則觸發低安全性警報。這並非不合理的值，但可以表示該自動資料清理並未保持啟動狀態。
- 如果資料表存留達到 10 億，這應視為採取動作的警報。一般而言，您會基於效能理由，而想讓存留期比較接近 `autovacuum_freeze_max_age`。我們建議您使用下列建議調查。

- 如果資料表達到 15 億的未清理交易，則觸發高安全性警報。視資料庫使用交易的速度而定，此警報可指出系統來不及執行自動資料清理。在此情況下，我們建議您立即解決此問題。

如果資料表不斷違反這些閾值，請進一步修改自動資料清理參數。依預設，手動使用 VACUUM (已停用或以成本為基礎的延遲) 比使用預設自動資料清理更積極，但總體上對系統的干擾也更加嚴重。

我們建議下列作法：

- 請注意並開啟監控機制，如此您才會注意最舊交易的存留期。
如需建立程序來警告您有關交易 ID 迴繞的相關資訊，請參閱 AWS 資料庫部落格文章中的[針對 Amazon RDS for PostgreSQL 中的交易 ID 迴繞實作預警系統](#)。
- 對於比較忙碌的資料表，除了依賴自動資料清理以外，請在維護時段定期執行手動清理凍結。如需執行手動清理凍結的資訊，請參閱 [執行手動清理凍結](#)。

判斷哪些資料表目前適合進行自動資料清理

通常有一或兩個資料表需要清理。表單中 relfrozenxid 的值大於 autovacuum_freeze_max_age 中交易的值，總是會成為自動資料清理的目標。否則，如果元組數因為最後一個 VACUUM 超出清理閾值而過時，則會清理資料表。

[自動資料清理閾值](#)已定義如下：

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

在哪 vacuum base threshold 裡 autovacuum_vacuum_threshold, vacuum scale factor 是 autovacuum_vacuum_scale_factor, 並且 number of tuples 是 pg_class.reltuples。

當您連線到資料庫時，執行以下查詢可查看自動資料清理功能認為符合清理條件的資料表清單。

```
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold'),
vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor'),
fma AS (SELECT setting AS autovacuum_freeze_max_age FROM pg_settings WHERE name =
'autovacuum_freeze_max_age'),
sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid, unnest(reloptions)
setting from pg_class) opt)
```

```

SELECT '''||ns.nspname||'."'||c.relname||''' as relation,
pg_size_pretty(pg_table_size(c.oid)) as table_size,
age(relfrozenxid) as xid_age,
coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
  autovacuum_freeze_max_age,
(coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float, autovacuum_vacuum_scale_factor::float) * c.reltuples)
AS autovacuum_vacuum_tuples, n_dead_tup as dead_tuples FROM
pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid join vbt on (1=1) join vsf on (1=1)
  join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and c.oid =
  cvbt.opt_oid
left join sto cvsf on cvsf.param = 'autovacuum_vacuum_scale_factor' and c.oid =
  cvsf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
AND (age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
OR coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float, autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup)
ORDER BY age(relfrozenxid) DESC LIMIT 50;

```

判斷自動資料清理目前是否執行中且執行多久時間

如果您需要手動清理資料表，請判斷目前是否在執行自動資料清理功能。若是，您可能需要調整參數，使其更有效率地執行，或暫時關閉自動資料清理功能，以便您手動執行 VACUUM。

使用以下查詢來判斷自動資料清理是否執行中、已執行多久時間，以及是否正等待另一個工作階段。

```

SELECT datname, username, pid, state, wait_event, current_timestamp - xact_start AS
  xact_runtime, query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;

```

在執行此查詢之後，您應會看到類似底下的輸出：

```

datname | username | pid | state | wait_event |      xact_runtime      | query
-----+-----+----+-----+-----+-----+-----
+-----+-----+----+-----+-----+-----+-----

```

```

mydb      | rdsadmin | 16473 | active |          | 33 days 16:32:11.600656 |
autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
mydb      | rdsadmin | 22553 | active |          | 14 days 09:15:34.073141 |
autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
mydb      | rdsadmin | 41909 | active |          | 3 days 02:43:54.203349 |
autovacuum: VACUUM ANALYZE public.mytable3
mydb      | rdsadmin | 618 | active |          | 00:00:00 |
SELECT datname, username, pid, state, wait_event, current_timestamp - xact_start AS
xact_runtime, query+
          |          |          |          |          |          | FROM
pg_stat_activity
          +
          |          |          |          |          |          | WHERE
query like '%VACUUM%'
          +
          |          |          |          |          |          | ORDER BY
xact_start;
          +

```

有幾個問題可能造成長時間執行自動資料清理階段 (長達數天)。最常見的問題就是資料表大小或更新率的 [maintenance_work_mem](#) 參數值設得太低。

建議您依照下列公式來設定 `maintenance_work_mem` 參數值。

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

短時間執行的自動資料清理工作階段也可以指出問題：

- 它可指出您的工作負載沒有足夠的 `autovacuum_max_workers`。在此情況下，您需要指出工作者數目。
- 它可指出有索引損毀 (自動資料清理會當機並以相同的關聯重新啟動，但沒有任何進度)。在這種情況下，請手動執行 `vacuum freeze verbose table` 來查看確切原因。

執行手動清理凍結

您可以在清理程序已在執行中的資料表上執行手動清理。如果您已辨識出表單的交易將近 20 億 (或超出您所監控的任何閾值) 的資料表，此功能將很實用。

下列步驟是準則，而程序會有多種變化。例如，在測試期間，假設您會發現 [maintenance_work_mem](#) 參數值設定的太小而您需要對資料表採取立即行動。不過，或許您當下不想退回執行個體。使用上個區段所列的查詢，您可判斷哪個資料表有問題，並注意長時間執行的自動資

料清理工作階段。您知道您必須變更 `maintenance_work_mem` 參數設定，但您也需要採取立即動作並清理有問題的資料表。以下程序說明在此情況下的處理方式。

手動執行清理凍結

1. 對包含要清理之資料表的資料庫，開啟兩個工作階段。在第二個工作階段中，如果連線中斷，請使用 "screen" 或其他公用程式來維持此工作階段。
2. 在第一個工作階段中，取得在資料表上執行之自動資料清理工作階段的處理程序 ID (PID)。

執行以下查詢來取得自動資料清理工作階段的 PID。

```
SELECT datname, username, pid, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
xact_start;
```

3. 在第二個工作階段中，計算您需要用於此作業的記憶體數量。在此範例中，我們判斷我們可以將最多 2 GB 的記憶體使用於此作業，所以將目前工作階段的 `maintenance_work_mem` 設定為 2 GB。

```
SET maintenance_work_mem='2 GB';
SET
```

4. 在第二個工作階段中，對表格發佈 `vacuum freeze verbose` 指令。詳細資訊設定很實用，因為即使 PostgreSQL 中目前沒有此工作階段的進度報告，您仍可查看活動。

```
\timing on
Timing is on.
vacuum freeze verbose pgbench_branches;
```

```
INFO: vacuuming "public.pgbench_branches"
INFO: index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
DETAIL: 0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
DETAIL: 0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "pgbench_branches": found 0 removable, 50 nonremovable row versions
in 43 out of 43 pages
```

```
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
VACUUM
Time: 2.765 ms
```

5. 在第一個工作階段，如果自動資料清理封鎖清理工作階段，您會在 `pg_stat_activity` 中看見清理工作階段的等待為 "T"。在此情況下，您需要按如下方式結束自動資料清理程序。

```
SELECT pg_terminate_backend('the_pid');
```

此時，您的工作階段會開始。請特別注意，自動資料清理會立即重新啟動，因為此資料表可能排在其工作清單的最前面。

6. 在第二個工作階段中啟動您的 `vacuum freeze verbose` 命令，然後在第一個工作階段中結束自動資料清理程序。

在自動資料清理執行時重新為資料表建立索引

如果索引毀損，自動資料清理會繼續處理資料表並且失敗。如果您在此情況下嘗試手動清理，則會收到如下錯誤訊息。

```
postgres=> vacuum freeze pgbench_branches;
ERROR:  index "pgbench_branches_test_index" contains unexpected
        zero page at block 30521
HINT:  Please REINDEX it.
```

當索引毀損且自動資料清理嘗試在資料表上執行時，您全力應付已在執行中的自動資料清理工作階段。當您發佈 [REINDEX](#) 指令，您開啟表單上的獨佔鎖定。寫入操作遭到封鎖，也使用特定索引讀取操作。

在資料表上執行自動資料清理時重新為資料表建立索引

1. 對包含要清理之資料表的資料庫，開啟兩個工作階段。在第二個工作階段中，如果連線中斷，請使用 "screen" 或其他公用程式來維持此工作階段。
2. 在第一個工作階段中，取得在資料表上執行之自動資料清理工作階段的 PID。

執行以下查詢來取得自動資料清理工作階段的 PID。

```
SELECT datname, username, pid, current_timestamp - xact_start
```

```
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
xact_start;
```

3. 在第二個工作階段中，發出 `reindex` 命令。

```
\timing on
Timing is on.
reindex index pgbench_branches_test_index;
REINDEX
Time: 9.966 ms
```

4. 在第一個工作階段中，如果自動資料清理封鎖處理程序，您會在 `pg_stat_activity` 中看見清理工作階段的等待為 "T"。在此情況下，您就結束自動資料清理程序。

```
SELECT pg_terminate_backend('the_pid');
```

此時，您的工作階段會開始。請特別注意，自動資料清理會立即重新啟動，因為此資料表可能排在其工作清單的最前面。

5. 在第二個工作階段中啟動您的命令，然後在第一個工作階段中結束自動資料清理程序。

使用大型索引管理自動清空

作為其操作的一部分，自動清空會在資料表上執行時執行數個[清空階段](#)。在清除資料表之前，首先會清空其所有索引。移除多個大型索引時，此階段會耗用大量的時間和資源。因此，最佳實務是務必控制資料表上的索引數目，並清除未使用的索引。

在此程序中，首先檢查整體索引大小。然後，判斷是否有可以移除的潛在未用索引，如下列範例所示。

檢查資料表及其索引的大小

```
postgres=> select pg_size_pretty(pg_relation_size('pgbench_accounts'));
pg_size_pretty
6404 MB
(1 row)
```

```
postgres=> select pg_size_pretty(pg_indexes_size('pgbench_accounts'));
pg_size_pretty
11 GB
```

```
(1 row)
```

在此範例中，索引的大小大於資料表。這種差異可能會導致性能問題，因為索引膨脹或未使用，這會影響自動清空以及插入操作。

檢查是否有未使用的索引

使用 [pg_stat_user_indexes](#) 檢視，您可以檢查索引與 `idx_scan` 資料欄搭配使用的頻率。在下列範例中，未使用的索引的 `idx_scan` 值為 0。

```
postgres=> select * from pg_stat_user_indexes where relname = 'pgbench_accounts' order
by idx_scan desc;
```

relid	indexrelid	schemaname	relname	indexrelname	idx_scan
idx_tup_read	idx_tup_fetch				
16433	16454	public	pgbench_accounts	index_f	6
6	0				
16433	16450	public	pgbench_accounts	index_b	3
199999	0				
16433	16447	public	pgbench_accounts	pgbench_accounts_pkey	0
0	0				
16433	16452	public	pgbench_accounts	index_d	0
0	0				
16433	16453	public	pgbench_accounts	index_e	0
0	0				
16433	16451	public	pgbench_accounts	index_c	0
0	0				
16433	16449	public	pgbench_accounts	index_a	0
0	0				

(7 rows)

```
postgres=> select schemaname, relname, indexrelname, idx_scan from pg_stat_user_indexes
where relname = 'pgbench_accounts' order by idx_scan desc;
```

schemaname	relname	indexrelname	idx_scan
public	pgbench_accounts	index_f	6
public	pgbench_accounts	index_b	3
public	pgbench_accounts	pgbench_accounts_pkey	0


```
public      | pgbench_accounts | index_d      | 0
public      | pgbench_accounts | index_e      | 0
public      | pgbench_accounts | index_c      | 0
public      | pgbench_accounts | index_a      | 0
(7 rows)
```

Note

這些統計資訊是從統計資訊重設時開始累計的。假設您有僅在營業季度結束時使用的索引，或僅用於特定報告的索引。自統計資訊重設以來，可能尚未使用此索引。如需詳細資訊，請參閱[統計資訊函數](#)。用來強制執行唯一性的索引不會執行掃描，也不應將其識別為未使用的索引。若要識別未使用的索引，您應該對應用程式及其查詢有深入的理解。

若要檢查上次何時重設資料庫的統計資訊，請使用 [pg_stat_database](#)

```
postgres=> select datname, stats_reset from pg_stat_database where datname =
'postgres';
```

```
datname  | stats_reset
-----+-----
postgres | 2022-11-17 08:58:11.427224+00
(1 row)
```

盡快清空資料表

RDS for PostgreSQL 12 和更新版本

如果大型資料表中有太多索引，您的資料庫執行個體可能接近交易 ID 包圍 (XID)，也就是當 XID 計數器包圍至零時。若保持取消核取的狀態，此情況可能會導致資料遺失。不過，您可以在不清除索引的情況下快速清空資料表。在 RDS for PostgreSQL 12 及更新版本中，您可以使用 `VACUUM` 搭配 [INDEX_CLEANUP](#) 子句。

```
postgres=> VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) pgbench_accounts;
```

```
INFO: vacuuming "public.pgbench_accounts"
INFO: table "pgbench_accounts": found 0 removable, 8 nonremovable row versions in 1 out
of 819673 pages
```

```
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 7517  
Skipped 0 pages due to buffer pins, 0 frozen pages.  
CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.01 s.
```

如果自動清空工作階段已在執行中，您必須終止它才能開始手動 VACUUM。如需執行手動清空凍結的相關資訊，請參閱 [執行手動清理凍結](#)。

Note

定期略過索引清除可能會導致索引膨脹，這會影響整體掃描效能。作為最佳實務，只使用上述程序來防止交易 ID 包圍。

RDS for PostgreSQL 11 和更舊版本

不過，在 RDS for PostgreSQL 11 和更舊版本中，允許清空更快完成的唯一方法是減少資料表上的索引數目。捨棄索引可能會影響查詢計畫。我們建議您先捨棄未使用的索引，然後在 XID 包圍非常接近時捨棄索引。在清空程序完成之後，您可以重新建立這些索引。

影響自動資料清理的其他參數

以下查詢顯示會直接影響自動資料清理及其行為的某些參數值。PostgreSQL 文件會完整說明 [自動資料清理參數](#)。

```
SELECT name, setting, unit, short_desc  
FROM pg_settings  
WHERE name IN (  
'autovacuum_max_workers',  
'autovacuum_analyze_scale_factor',  
'autovacuum_naptime',  
'autovacuum_analyze_threshold',  
'autovacuum_analyze_scale_factor',  
'autovacuum_vacuum_threshold',  
'autovacuum_vacuum_scale_factor',  
'autovacuum_vacuum_threshold',  
'autovacuum_vacuum_cost_delay',  
'autovacuum_vacuum_cost_limit',  
'vacuum_cost_limit',  
'autovacuum_freeze_max_age',  
'maintenance_work_mem',
```

```
'vacuum_freeze_min_age');
```

雖然這些全都會影響自動資料清理，其中最重要的參數如下：

- [maintenance_work_mem](#)
- [autovacuum_freeze_max_age](#)
- [autovacuum_max_workers](#)
- [autovacuum_vacuum_cost_delay](#)
- [autovacuum_vacuum_cost_limit](#)

設定自動資料清理參數資料表層級

您可以在資料表層級設定自動資料清理相關的[儲存參數](#)，這比改變整個資料庫的行為更理想。對於大型資料表，您可能需要設定積極的設定值，而且不想讓自動資料清理以那種方式對待所有的資料表。

以下查詢顯示哪些資料表目前已備妥資料表層級選項。

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

在比您其餘的資料表大許多的資料表範例上，此查詢可能很實用。假設您有一個 300 GB 的表單，及 30 個少於 1 GB 的其他表單。在此情況下，您可以為大型資料表設定一些特定參數，您就不會改變整個系統的行為。

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

這麼做會針對此資料表關閉成本型自動資料清理延遲，代價是在您系統上使用更多資源。通常情況下，每次達到 `autovacuum_cost_limit` 時，自動資料清理都會暫停 `autovacuum_vacuum_cost_delay`。如需詳細資訊，請參閱 PostgreSQL 文件中的[成本型清理](#)。

記錄清理和自動資料清理活動

有關自動資料清理活動的資訊將根據 `rds.force_autovacuum_logging_level` 參數中指定的層級傳送到 `postgresql.log`。以下是此參數允許的值，以及預設為該值的 PostgreSQL 版本：

- `disabled` (PostgreSQL 10、PostgreSQL 9.6)

- debug5, debug4, debug3, debug2, debug1
- info (PostgreSQL 12、PostgreSQL 11)
- notice
- warning (PostgreSQL 13 及以上)
- error、日誌、fatal、panic

`rds.force_autovacuum_logging_level` 使用 `log_autovacuum_min_duration` 參數。`log_autovacuum_min_duration` 參數的值是記錄自動資料清理動作的閾值 (以毫秒為單位)。設定為 -1，表示不會記錄任何內容；設定為 0，則會記錄所有動作。如同 `rds.force_autovacuum_logging_level`，`log_autovacuum_min_duration` 的預設值取決於版本，如下所示：

- 10000 ms : PostgreSQL 14、PostgreSQL 13、PostgreSQL 12 和 PostgreSQL 11
- (empty) : PostgreSQL 10 和 PostgreSQL 9.6 沒有預設值

建議您將 `rds.force_autovacuum_logging_level` 設定為 WARNING。我們也建議您將 `log_autovacuum_min_duration` 設定為 1000 到 5000 之間的值。設定為 5000，表示會記錄時間超過 5,000 毫秒的活動。如果鎖定衝突或同時刪除關係導致跳過自動資料清理動作，-1 以外的任何設定也會記錄訊息。如需詳細資訊，請參閱 PostgreSQL 文件中的 [自動資料清理](#)。

若要解決問題，您可以將 `rds.force_autovacuum_logging_level` 參數變更為其中一個除錯等級 (從 debug1 至 debug5)，以取得詳盡資訊。我們建議您在短時間內使用除錯設定，並且僅用於疑難排解目的。如需進一步了解，請參閱 PostgreSQL 文件中的 [何時記錄](#)。

Note

PostgreSQL 可讓 `rds_superuser` 帳戶檢視 `pg_stat_activity` 中的自動資料清理工作階段。例如，您可以找出並終止會阻擋命令執行的自動資料清理工作階段，或是執行速度比手動發出的清理命令還要慢的自動資料清理工作階段。

搭配使用 RDS for PostgreSQL 支援的記錄機制

您可以設定數個參數、擴充功能和其他可設定項目來記錄在 PostgreSQL 資料庫執行個體上發生的活動。這些索引標籤包括以下項目：

- `log_statement` 參數可用來記錄 PostgreSQL 資料庫中的使用者活動。若要進一步了解 RDS for PostgreSQL 記錄和如何監控日誌，請參閱 [RDS for PostgreSQL 資料庫日誌檔](#)。
- `rds.force_admin_logging_level` 參數可記錄 Amazon RDS 內部使用者 (rdsadmin) 在資料庫中對資料庫執行個體執行的動作。其會將輸出寫入 PostgreSQL 錯誤日誌。允許的值為：`disabled`、`debug5`、`debug4`、`debug3`、`debug2`、`debug1`、`info`、`notice`、`warning`、`error` 和 `panic`。預設值為 `disabled`。
- `rds.force_autovacuum_logging_level` 參數可設定為在 PostgreSQL 錯誤日誌中擷取各種自動資料清理作業。如需詳細資訊，請參閱 [記錄清理和自動資料清理活動](#)。
- PostgreSQL Audit (pgAudit) 擴充功能可安裝並設定為擷取工作階段層級或物件層級的活動。如需詳細資訊，請參閱 [使用 PgAudit 記錄資料庫活動](#)。
- `log_fdw` 擴充功能可讓您使用 SQL 存取資料庫引擎日誌。如需詳細資訊，請參閱 [使用 log_fdw 擴充功能存取使用 SQL 的資料庫日誌](#)。
- `pg_stat_statements` 程式庫被指定為 RDS for PostgreSQL 第 10 版及更高版本中 `shared_preload_libraries` 參數的預設值。您可以使用這個程式庫來分析正在執行的查詢。請確定已在資料庫參數群組中設定 `pg_stat_statements`。如需使用此程式庫提供的資訊監控 RDS for PostgreSQL 資料庫執行個體的詳細資訊，請參閱 [RDS PostgreSQL 的 SQL 統計資料](#)。
- `log_hostname` 參數會將每個用戶端連線的主機名稱擷取至日誌檔。對於 RDS for PostgreSQL 第 12 版及更新版本，此參數預設會設為 `off`。如果您將其開啟，請務必監控工作階段連線時間。開啟時，服務會使用網域名稱系統 (DNS) 反向查詢請求，來取得進行連線的用戶端主機名稱，並將其新增至 PostgreSQL 日誌。這會在工作階段連線期間產生明顯的影響。建議您開啟此參數，僅用於疑難排解目的。

一般來說，記錄是為了讓 DBA 監控情況、調校效能和排除故障。許多日誌會自動上傳到 Amazon CloudWatch 或 Performance Insights。日誌會在這些地方進行排序和分組，以便為您的資料庫執行個體提供完整的指標。若要進一步了解 Amazon RDS 監控和指標，請參閱 [監控 Amazon RDS 執行個體中的指標](#)。

使用 PostgreSQL 管理暫存檔案

在 PostgreSQL 中，執行排序和雜湊操作的查詢會使用執行個體記憶體來儲存結果，直到 [work_mem](#) 參數中指定的值為止。當執行個體記憶體不足時，會建立暫存檔來儲存結果。這些檔案會寫入磁碟以完成查詢執行。稍後，這些檔案會在查詢完成後自動移除。在 RDS for PostgreSQL 中，這些檔案會儲存在資料磁碟區上的 Amazon EBS 中。如需詳細資訊，請參閱 [Amazon RDS 資料庫執行個體儲存體](#)。您可以監控在 CloudWatch 中發佈的 `FreeStorageSpace` 指標，藉此確保資料庫執行個體有充足的可用儲存空間。如需詳細資訊，請參閱 [FreeStorageSpace](#)。

我們建議針對涉及多個並行查詢的工作負載使用 Amazon RDS Optimized Reads 執行個體，以增加暫存檔案的使用量。這些執行個體會使用本機非揮發性記憶體快速 (NVMe) 為主的固態硬碟 (SSD) 區塊層級儲存體來放置暫存檔案。如需詳細資訊，請參閱 [Amazon RDS Optimized Reads](#)。

您可以使用以下參數和函數來管理執行個體中的暫存檔案。

- **[temp_file_limit](#)** – 此參數會取消任何超過 temp_file 大小的查詢 (以 KB 為單位)。此限制可防止任何查詢無休止地執行，並消耗含有暫存檔的磁碟空間。您可以使用 log_temp_files 參數的結果來估計值。最佳實務是檢查工作負載行為並根據估計值設定限制。以下範例顯示查詢超過限制時的取消方式。

```
postgres=> select * from pgbench_accounts, pg_class, big_table;
```

```
ERROR: temporary file size exceeds temp_file_limit (64kB)
```

- **[log_temp_files](#)** – 此參數會在移除工作階段的暫存檔案時，將訊息傳送至 postgresql.log。這個參數會在查詢順利完成後產生日誌。因此，它可能無助於疑難排解使用中、長時間執行的查詢。

下列範例顯示，當查詢順利完成時，這些項目會在清理暫存檔時記錄在 postgresql.log 檔案中。

```
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
temporary file: path "base/pgsql_tmp/pgsql_tmp31236.5", size 140353536
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
a.bid limit 10;
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
temporary file: path "base/pgsql_tmp/pgsql_tmp31236.4", size 180428800
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
a.bid limit 10;
```

- **[pg_ls_tmpdir](#)** – 此功能可從 RDS for PostgreSQL 13 及以上版本使用，提供目前暫存檔案使用情況的可見性。完成的查詢不會出現在函數的結果中。在下列範例中，您可以檢視此函數的結果。

```
postgres=> select * from pg_ls_tmpdir();
```

name	size	modification
pgsql_tmp8355.1	1072250880	2023-02-06 22:54:56+00
pgsql_tmp8351.0	1072250880	2023-02-06 22:54:43+00
pgsql_tmp8327.0	1072250880	2023-02-06 22:54:56+00
pgsql_tmp8351.1	703168512	2023-02-06 22:54:56+00
pgsql_tmp8355.0	1072250880	2023-02-06 22:54:00+00
pgsql_tmp8328.1	835031040	2023-02-06 22:54:56+00
pgsql_tmp8328.0	1072250880	2023-02-06 22:54:40+00

(7 rows)

```
postgres=> select query from pg_stat_activity where pid = 8355;
```

```
query
```

```
-----
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
a.bid
(1 row)
```

檔案名稱包含產生暫存檔之工作階段的處理 ID (PID)。較進階的查詢 (例如下列範例) 會針對每個 PID 執行暫存檔案的總和。

```
postgres=> select replace(left(name, strpos(name, '.')-1), 'pgsql_tmp', '') as pid,
count(*), sum(size) from pg_ls_tmpdir() group by pid;
```

pid	count	sum
8355	2	2144501760
8351	2	2090770432
8327	1	1072250880
8328	2	2144501760

(4 rows)

- [pg_stat_statements](#) – 如果您啟動 `pg_stat_statements` 參數，則可以檢視每次呼叫的平均暫存檔案使用量。您可以識別查詢的 `query_id`，並使用它來檢查暫存檔的使用情況，如以下範例所示。

```
postgres=> select queryid from pg_stat_statements where query like 'select a.aid from
pgbench%';
```

```
      queryid
-----
-7170349228837045701
(1 row)
```

```
postgres=> select queryid, substr(query,1,25), calls, temp_blks_read/calls
temp_blks_read_per_call, temp_blks_written/calls temp_blks_written_per_call from
pg_stat_statements where queryid = -7170349228837045701;
```

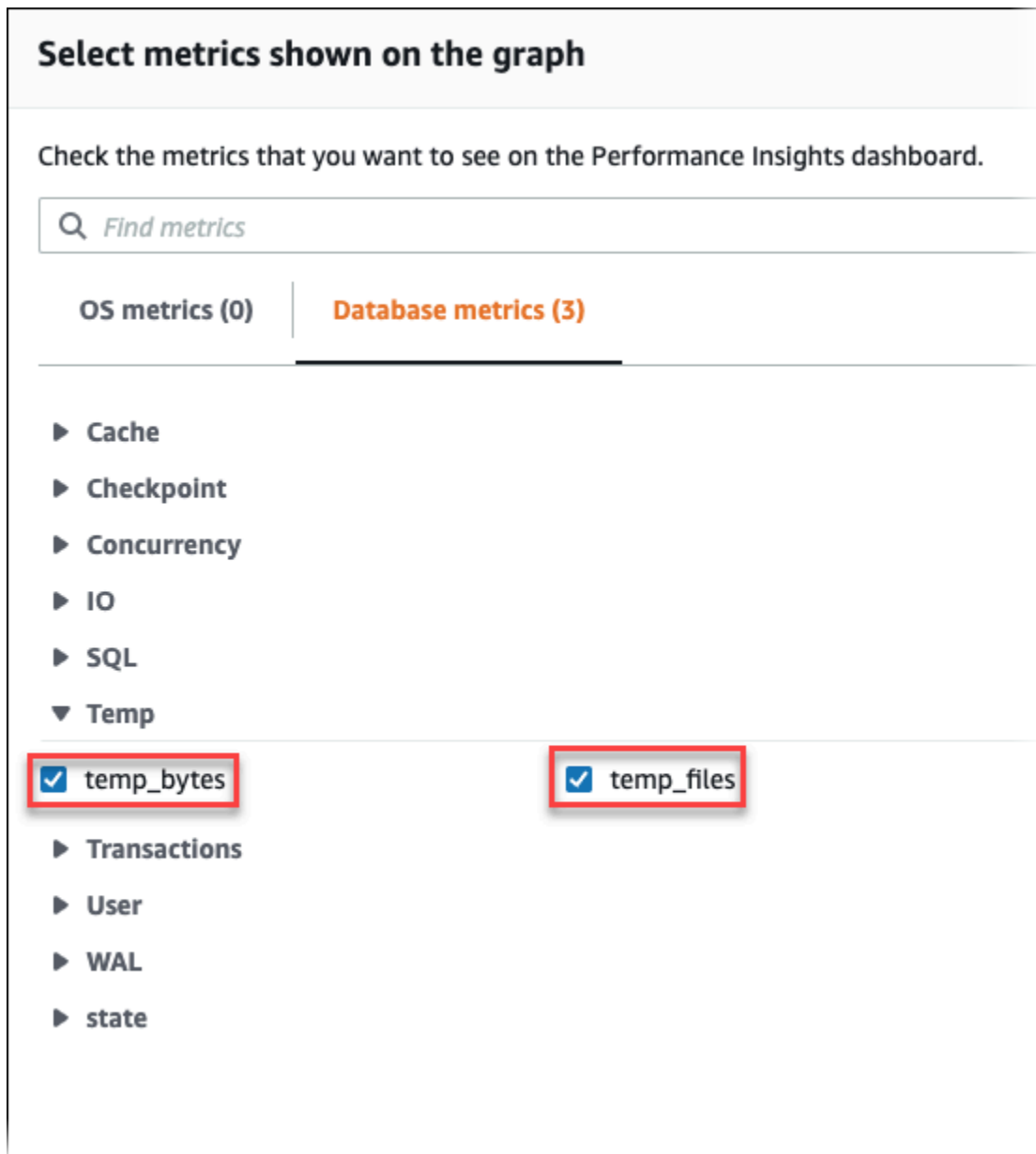
```
      queryid          |          substr          | calls | temp_blks_read_per_call |
temp_blks_written_per_call
-----+-----+-----+-----+-----
-7170349228837045701 | select a.aid from pgbench |    50 |                239226 |
                        388678
(1 row)
```

- **Performance Insights** – 在績效詳情儀表中，您可以透過開啟指標 `temp_bytes` 和 `temp_files` 來檢視暫存檔的使用情況。然後，您可以查看這兩個指標的平均值，並查看它們如何對應至查詢工作負載。績效詳情中的檢視不會明確顯示產生暫存檔的查詢。不過，當您將績效詳情與為 `pg_ls_tmpdir` 顯示的查詢結合使用時，您可以疑難排解、分析及判斷查詢工作負載中的變更。

如需如何使用績效詳情來分析指標和查詢的詳細資訊，請參閱 [使用績效詳情儀表板來分析指標](#)

使用績效詳情來檢視暫存檔使用情況

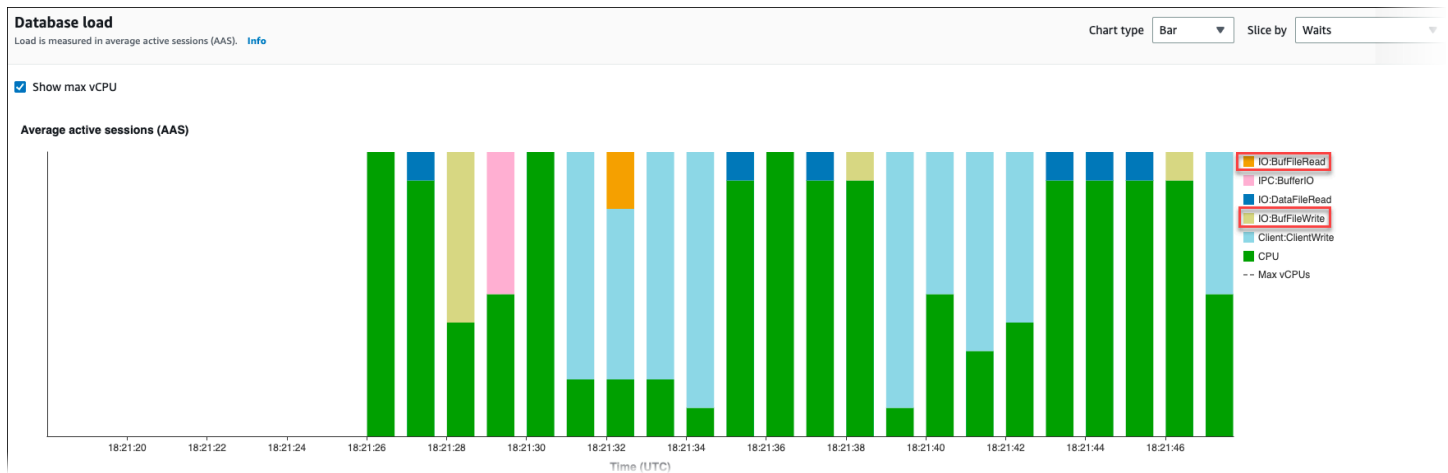
1. 在 [績效詳情] 儀表中選擇管理指標。
2. 選擇資料庫指標，並選取 `temp_bytes` 和 `temp_files` 指標，如下方影像所示。



3. 在最高 SQL 索引標籤中，選擇偏好設定圖示。
4. 在偏好設定視窗中，開啟最高 SQL 索引標籤中顯示的下列統計資料，然後選擇繼續。
 - Temp writes/sec
 - Temp reads/sec
 - Tmp blk write/call
 - Tmp blk read/call
5. 暫存檔在與針對 `pg_ls_tmpdir` 顯示的查詢組合時會被劃分，如以下範例所示。

Top SQL (1) Learn more							
Find SQL statements							
	SQL statements	Calls/sec	Rows/sec	Temp wri...	Temp rea...	Tmp blk ...	Tmp blk r...
11.77	select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order...	0.04	0.43	16589.14	10307.89	381550.15	237081.46

當工作負載中最常用的查詢經常建立暫存檔案時，就會發生 `IO:BufFileRead` 和 `IO:BufFileWrite` 事件。您可以使用 Performance Insights，透過檢閱「資料庫負載」和「最高 SQL」區段中的「平均作用中工作階段」(AAS)，找出最常在 `IO:BufFileRead` 和 `IO:BufFileWrite` 上等待的查詢。



如需如何使用 Performance Insights 來分析各等待事件的最常用查詢和負載的詳細資訊，請參閱 [最高 SQL 索引標籤概觀](#)。您應找出並調整造成暫存檔案使用量及相關等待事件增加的查詢。如需這些等待事件和修補的詳細資訊，請參閱 [IO:BufFileRead](#) 和 [IO:BufFileWrite](#)。

Note

`work_mem` 參數可控制排序操作何時用完記憶體，以及將結果寫入暫存檔。我們建議您不要將此參數的設定變更為高於預設值，因為它會允許每個資料庫工作階段耗用更多記憶體。此外，執行複雜聯結和排序的單一工作階段可以執行平行操作，其中每個操作都會耗用記憶體。最佳實務是，當您有具有多個聯結和排序的大型報表時，請使用 `SET work_mem` 命令在工作階段層級設定此參數。然後，變更僅套用於目前工作階段，不會全域變更該值。

使用 pgBadger 進行 PostgreSQL 的日誌分析

您可以使用 [pgbadger](#) 等日誌分析器來分析 PostgreSQL 日誌。pgBadger 文件會陳述 %l 模式 (工作階段或程序的日誌行) 應該是字首的一部分。不過，如果您將目前的 RDS log_line_prefix 當作參數提供給 pgBadger，它應該仍會產生報告。

例如，下列命令會使用 pgbadger 正確地格式化日期為 2014-02-04 的 Amazon RDS for PostgreSQL 日誌檔案。

```
./pgbadger -f stderr -p '%t:%r:%u@d:[%p]:' postgresql.log.2014-02-04-00
```

使用 PgSNAPPER 監控 PostgreSQL

您可以使用 PgSnapper 協助定期收集與 Amazon RDS for PostgreSQL 效能相關的統計資訊和指標。如需詳細資訊，請參閱[使用 PGSnapper 監控 Amazon RDS for PostgreSQL 效能](#)

在 RDS for PostgreSQL 資料庫執行個體上搭配使用參數

某些情況下，您可能會在沒有指定自訂參數群組的狀況下建立一個 RDS for PostgreSQL 資料庫執行個體。如果是這樣，系統會使用您所選擇 PostgreSQL 版本的預設參數群組來建立資料庫執行個體。例如，假設您使用 PostgreSQL 13.3 建立 RDS for PostgreSQL 資料庫執行個體。在這種情況下，系統會使用 PostgreSQL 第 13 版參數群組 (default.postgres13) 中的值來建立資料庫執行個體。

您也可以建立自己的自訂資料庫參數群組。如果要從預設值中修改 RDS for PostgreSQL 資料庫執行個體的任何設定，則需要執行此操作。如要瞭解如何作業，請參閱[使用參數群組](#)。

您可以使用多種不同的方式追蹤 RDS for PostgreSQL 資料庫執行個體上的設定。您可以使用 AWS Management Console AWS CLI、或 Amazon RDS API。您也可以查詢執行個體 PostgreSQL pg_settings 資料表中的值，如下所示。

```
SELECT name, setting, boot_val, reset_val, unit
FROM pg_settings
ORDER BY name;
```

若要進一步了解此查詢傳回的值，請參閱 PostgreSQL 說明文件中的 [pg_settings](#)。

為 RDS for PostgreSQL 資料庫執行個體更改 max_connections 和 shared_buffers 的設定時請格外小心。例如，假設您修改 max_connections 或 shared_buffers 的設定並且使用的值對於實

際工作負載而言過高。這種情況下，RDS for PostgreSQL 資料庫執行個體將無法啟動。如果發生這種情況，`postgres.log` 中會顯示錯誤，如下所示。

```
2018-09-18 21:13:15 UTC::@[8097]:FATAL: could not map anonymous shared memory: Cannot
allocate memory
2018-09-18 21:13:15 UTC::@[8097]:HINT: This error usually means that PostgreSQL's
request for a shared memory segment
exceeded available memory or swap space. To reduce the request size (currently
3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
max_connections.
```

但是，您無法更改預設 RDS for PostgreSQL 資料庫參數群組中包含的任何設定值。若要變更任何參數的設定，請首先建立自訂資料庫參數群組。然後變更該自訂資料群組中的設定，再將自訂參數群組套用於 RDS for PostgreSQL 資料庫執行個體。如需進一步了解，請參閱[使用參數群組](#)。

有兩種類型的參數在適用於 PostgreSQL 的。

- 靜態參數 – 靜態參數變更後需重新啟動 RDS for PostgreSQL 資料庫執行個體，新值才會生效。
- 動態參數 – 動態參數的設定變更後不需要重新啟動。

Note

如果 RDS for PostgreSQL 資料庫執行個體使用自己的自訂資料庫參數群組，您可以變更正在執行中的資料庫執行個體上的動態參數值。您可以使用 AWS Management Console、AWS CLI 或 Amazon RDS API 來這麼做。

如果您擁有權限，也可以使用 ALTER DATABASE、ALTER ROLE 和 SET 命令來變更參數值。

RDS for PostgreSQL DB 資料庫執行個體參數清單

下表列出 RDS for PostgreSQL 資料庫執行個體中的一些 (但並非全部) 可用參數。若要檢視所有可用的參數，請使用 [describe-db-parameters](#) AWS CLI 指令。例如，若要取得預設參數群組中 RDS for PostgreSQL 第 13 版可用的所有參數清單，請執行下列命令。

```
aws rds describe-db-parameters --db-parameter-group-name default.postgres13
```

您也可以使用主控台。從 Amazon RDS 功能表中選擇 Parameter groups (參數群組)，然後從 AWS 區域中可用的參數群組中選擇參數群組。

參數名稱	Apply_Type	描述
application_name	動態	設定要在統計資訊和日誌中報告的應用程式名稱。
archive_command	動態	設定將呼叫的 shell 命令來封存 WAL 檔案。
array_nulls	動態	能夠在陣列中輸入 NULL 元素。
authentication_timeout	動態	設定要完成用戶端驗證的最大允許時間。
autovacuum	動態	啟動自動資料清理子程序。
autovacuum_analyze_scale_factor	動態	在分析之前的 tuple 插入、更新或刪除數目 (成為 reltuple 的一小部分)。
autovacuum_analyze_threshold	動態	在分析之前的 tuple 插入、更新或刪除數目下限。
autovacuum_freeze_max_age	靜態	要自動資料清理資料表以防止交易 ID 包圍的存留期。
autovacuum_naptime	動態	自動資料清理執行之間的休眠時間。
autovacuum_max_workers	靜態	設定同時執行自動清空工作者程序的數目上限。

參數名稱	Apply_Type	描述
autovacuum_vacuum_cost_delay	動態	自動資料清理的清理成本延遲 (以毫秒為單位)。
autovacuum_vacuum_cost_limit	動態	自動資料清理在小憩前可用的清理成本金額。
autovacuum_vacuum_scale_factor	動態	在清理之前的 tuple 更新或刪除數目 (成為 reltuple 的一小部分)。
autovacuum_vacuum_threshold	動態	在清理之前的 tuple 更新或刪除數目下限。
backslash_quote	動態	設定在字串常值中是否允許反斜線 (\)。
bgwriter_delay	動態	回合之間的背景寫入器休眠時間。
bgwriter_lru_maxpages	動態	每回合要清空之 LRU 頁面的背景寫入器數目上限。
bgwriter_lru_multiplier	動態	每回合要釋放之平均緩衝區使用量的倍數。
bytea_output	動態	設定位元組的輸出格式。
check_function_bodies	動態	在 CREATE FUNCTION 期間檢查函數本文。
checkpoint_completion_target	動態	在檢查點期間用於清空已變更緩衝區的時間 (成為檢查點間隔的一小部分)。
checkpoint_segments	動態	設定日誌區段中自動預寫入日誌 WAL 檢查點之間的最大距離。
checkpoint_timeout	動態	設定自動 WAL 檢查點之間的最長時間。
checkpoint_warning	動態	啟用檢查點區段的填入頻率高於此值時的警告。

參數名稱	Apply_Type	描述
client_connection_check_interval	動態	設定執行查詢時檢查是否中斷連線的時間間隔。
client_encoding	動態	設定用戶端的字元集編碼。
client_min_messages	動態	設定傳送給用戶端的訊息層級。
commit_delay	動態	設定在交易認可與將 WAL 清空至磁碟之間的延遲 (以毫秒為單位)。
commit_siblings	動態	設定在執行 commit_delay 之前同時開啟的交易數目下限。
constraint_exclusion	動態	讓規劃器能夠使用限制條件來最佳化查詢。
cpu_index_tuple_cost	動態	設定規劃器對於在索引掃描期間處理每個索引項目的成本估算。
cpu_operator_cost	動態	設定規劃器對於處理每個運算子或函數呼叫的成本估算。
cpu_tuple_cost	動態	設定規劃器對於處理每個 tuple (資料列) 的成本估算。
cursor_tuple_fraction	動態	設定規劃器對於將擷取之游標的資料列部分估算。
datestyle	動態	設定日期和時間值的顯示格式。
deadlock_timeout	動態	設定在檢查死鎖前等待鎖定的時間。
debug_pretty_print	動態	將剖析和計劃樹狀顯示縮排。
debug_print_parse	動態	記錄每項查詢的剖析樹狀。
debug_print_plan	動態	記錄每項查詢的執行計劃。

參數名稱	Apply_Type	描述
debug_print_rewritten	動態	記錄每項查詢的重寫剖析樹狀。
default_statistics_target	動態	設定預設統計資訊目標。
default_tablespace	動態	設定要在其中建立資料表和索引的預設資料表空間。
default_transaction_deferrable	動態	設定新交易的預設可延遲狀態。
default_transaction_isolation	動態	設定每項新交易的交易隔離層級。
default_transaction_read_only	動態	設定新交易的預設唯讀狀態。
default_with_oids	動態	依預設建立內含物件 ID (OID) 的新資料表。
effective_cache_size	動態	設定規劃器對於磁碟快取大小的假設。
effective_io_concurrency	動態	磁碟子系統可以有效處理的同時要求數目。
enable_bitmapscan	動態	讓規劃器能夠使用點陣圖掃描計劃。
enable_hashagg	動態	讓規劃器能夠使用雜湊彙整計劃。
enable_hashjoin	動態	讓規劃器能夠使用雜湊聯結計劃。
enable_indexscan	動態	讓規劃器能夠使用索引掃描計劃。
enable_material	動態	讓規劃器能夠使用實體化。
enable_mergejoin	動態	讓規劃器能夠使用合併聯結計劃。

參數名稱	Apply_Type	描述
enable_nestloop	動態	讓規劃器能夠使用巢狀迴圈聯結計劃。
enable_seqscan	動態	讓規劃器能夠使用循序掃描計劃。
enable_sort	動態	讓規劃器能夠使用明確排序步驟。
enable_tidscan	動態	讓規劃器能夠使用 TID 掃描計劃。
escape_string_warning	動態	一般字串常值中反斜線 (\) 逸出的相關警告。
extra_float_digits	動態	設定針對浮點值顯示的位數。
from_collapse_limit	動態	設定 FROM-list 大小，超過此大小就不會收合子查詢。
fsync	動態	強制將更新同步至磁碟。
full_page_writes	動態	在檢查點後第一次修改時，將完整頁面寫入至 WAL。
geqo	動態	啟用基因查詢最佳化。
geqo_effort	動態	GEQO：作業是用於設定其他 GEQO 參數的預設值。
geqo_generations	動態	GEQO：演算法的反覆運算次數。
geqo_pool_size	動態	GEQO：人口中的個體數目。
geqo_seed	動態	GEQO：隨機路徑選取的種子。
geqo_selection_bias	動態	GEQO：人口中的選擇壓力。
geqo_threshold	動態	設定 FROM 項目的閾值，超出此閾值時就會使用 GEQO。

參數名稱	Apply_Type	描述
gin_fuzzy_search_limit	動態	設定 GIN 確切搜尋所允許的結果上限。
hot_standby_feedback	動態	決定熱待命是否將意見回饋訊息傳送給主要或上游待命。
intervalstyle	動態	設定間隔值的顯示格式。
join_collapse_limit	動態	設定 FROM-list 大小，超過此大小就不會將 JOIN 結構扁平化。
lc_messages	動態	設定用來顯示訊息的語言。
lc_monetary	動態	設定用於格式化貨幣金額的地區設定。
lc_numeric	動態	設定用於格式化數字的地區設定。
lc_time	動態	設定用於格式化日期和時間值的地區設定。
log_autovacuum_min_duration	動態	設定執行時間下限，超出此時間就會記錄自動資料清理動作。
log_checkpoints	動態	記錄每個檢查點。
log_connections	動態	記錄每個成功連線。
log_disconnections	動態	記錄工作階段的結尾，包括持續時間。
log_duration	動態	記錄每個已完成 SQL 陳述式的持續時間。
log_error_verbosity	動態	設定已記錄訊息的詳細資訊。
log_executor_stats	動態	將執行器效能統計資訊寫入至伺服器日誌。
log_filename	動態	設定日誌檔案的檔案名稱樣式。
log_file_mode	動態	設定日誌檔案的許可。預設值為 0644。

參數名稱	Apply_Type	描述
log_hostname	動態	在連線日誌中記錄主機名稱。從 PostgreSQL 12 及更新版本開始，此參數預設為 'off'。開啟時，連線會使用 DNS 反向查詢來取得擷取至連線日誌的主機名稱。如果開啟此參數，您應該監控它對建立連線所需時間的影響。
log_line_prefix	動態	控制每個日誌行前綴的資訊。
log_lock_waits	動態	記錄長鎖定等待。
log_min_duration_statement	動態	設定執行時間下限，超出此時間就會記錄陳述式。
log_min_error_statement	動態	導致所有陳述式在這個層級或以上產生要記錄的錯誤。
log_min_messages	動態	設定所記錄的訊息層級。
log_parser_stats	動態	將剖析器效能統計資訊寫入至伺服器日誌。
log_planner_stats	動態	將規劃器效能統計資訊寫入至伺服器日誌。
log_rotation_age	動態	自動日誌檔案輪換將在 N 分鐘後發生。
log_rotation_size	動態	自動日誌檔案輪換將在 N KB 後發生。
log_statement	動態	設定已記錄的陳述式類型。
log_statement_stats	動態	將累積效能統計資訊寫入至伺服器日誌。
log_temp_files	動態	記錄大於此 KB 數的暫時檔案使用。
log_timezone	動態	設定要在日誌訊息中使用的時區。
log_truncate_on_rotation	動態	在日誌輪換期間截斷名稱相同的現有日誌檔案。

參數名稱	Apply_Type	描述
logging_collector	靜態	開始子程序，將 stderr 輸出和/或 csvlog 擷取到日誌檔案中。
maintenance_work_mem	動態	設定要用於維護作業的記憶體上限。
max_connections	靜態	設定同時連線的數目上限。
max_files_per_process	靜態	設定每個伺服器程序的同時開啟檔案數目上限。
max_locks_per_transaction	靜態	設定每項交易的鎖定數目上限。
max_pred_locks_per_transaction	靜態	設定每項交易的述詞鎖定數目上限。
max_prepared_transactions	靜態	設定同時備妥交易的數目上限。
max_stack_depth	動態	設定堆疊深度上限 (以 KB 為單位)。
max_standby_archive_delay	動態	當熱待命伺服器正在處理已封存的 WAL 資料時，設定在取消查詢前的延遲上限。
max_standby_streaming_delay	動態	當熱待命伺服器正在處理已串流的 WAL 資料時，設定在取消查詢前的延遲上限。
max_wal_size	動態	設定觸發檢查點的 WAL 大小 (MB)。對於 RDS for PostgreSQL 10 之後的所有版本，預設值至少為 1 GB (1024 MB)。例如，RDS for PostgreSQL 14 的 max_wal_size 設定為 2 GB (2048 MB)。在 RDS for PostgreSQL 資料庫執行個體上使用 SHOW max_wal_size; 命令，以查看其現行值。

參數名稱	Apply_Type	描述
min_wal_size	動態	設定 WAL 縮減到最小的大小。PostgreSQL 9.6 版和更早的版本，min_wal_size 單位是 16 MB。PostgreSQL 10 版和更新版本，min_wal_size 單位是 1 MB。
quote_all_identifiers	動態	在產生 SQL 片段時，將引號 (") 新增至所有識別符。
random_page_cost	動態	設定規劃器對於非循序擷取磁碟頁面的成本估算。除非開啟查詢計劃管理 (QPM)，否則此參數並無任何值。開啟 QPM 時，此參數的預設值為 4。
rds.adaptive_autovacuum	動態	當交易 ID 閾值超過時，隨時自動調整自動資料清理的參數。
rds.force_ssl	動態	需要使用 SSL 連線。對於 RDS for PostgreSQL 第 15 版，預設值會設定為 1 (開啟)。所有其他 RDS for PostgreSQL 主要版本 14 及更舊版本都會將預設值設定為 0 (關閉)。
rds.local_volume_spill_enabled	靜態	允許將邏輯溢出檔案寫入本機磁碟區。
rds.log_retention_period	動態	設定日誌保留期，像是 Amazon RDS 超過 n 分鐘時刪除 PostgreSQL 日誌。
rds.rds_superuser_reserved_connections	靜態	設定為 rds_superuser 保留的連線槽數。此參數僅適用於版本 15 及更早版本。 如需詳細資訊，請參閱保留的 PostgreSQL 文件。
rds.restrict_password_commands	靜態	限制可以管理具有 rds_password 角色之使用者密碼的人員。將此參數設為 1 可啟用密碼限制。預設為 0。
search_path	動態	針對不符合結構描述的名稱設定結構描述搜尋順序。

參數名稱	Apply_Type	描述
seq_page_cost	動態	設定規劃器對於循序擷取磁碟頁面的成本估算。
session_replication_role	動態	設定觸發器和重寫規則的工作階段行為。
shared_buffers	靜態	設定伺服器所用的共享記憶體緩衝區數目。
shared_preload_libraries	靜態	列出要預先載入至 RDS for PostgreSQL 資料庫執行個體的共用程式庫。支援的值包括 auto_explain、orafce、pgaudit、pglogical、pg_bigm、pg_cron、pg_hint_plan、pg_prewarm、pg_similarity、pg_stat_statements、pg_tle、pg_transport、pprofiler 及 plrust。
ssl	動態	啟用 SSL 連線。
sql_inheritance	動態	導致子資料表依預設包含在各種命令中。
ssl_renegotiation_limit	動態	設定在重新商談加密金鑰前要傳送和接收的流量。
standard_conforming_strings	動態	導致 ... 字串逐字地處理反斜線。
statement_timeout	動態	設定任何陳述式允許的持續時間上限。
synchronize_seqscans	動態	啟用已同步的循序掃描。
synchronous_commit	動態	設定目前交易的同步層級。
tcp_keepalives_count	動態	TCP 保持連線重新傳輸的數量上限
tcp_keepalives_idle	動態	發出 TCP 保持連線之間的時間。
tcp_keepalives_interval	動態	TCP 保持連線重新傳輸之間的時間。

參數名稱	Apply_Type	描述
temp_buffers	動態	設定每個工作階段所用的暫時緩衝區數目上限。
temp_file_limit	動態	設定暫時檔案可以增長的大小上限 (以 KB 為單位)。
temp_tablespaces	動態	設定要用於暫存資料表和排序檔案的資料表空間。
timezone	動態	<p>設定可供顯示和解譯時間戳記的時區。</p> <p>網際網路號碼分配局 (IANA) 每年會在 https://www.iana.org/time-zones 發布數次新時區。每次 RDS 發布 PostgreSQL 的新次要維護版本時，它都會在發布時隨附最新的時區資料。當您使用最新的 RDS for PostgreSQL 版本時，您會有來自 RDS 的最新時區資料。為確保您的資料庫執行個體擁有最新的時區資料，建議您升級至更高的資料庫引擎版本。您無法手動修改 PostgreSQL 資料庫執行個體中的時區表。RDS 不會修改或重設執行中資料庫執行個體的時區資料。只有在您執行資料庫引擎版本升級時，才會安裝新的時區資料。</p>
track_activities	動態	收集有關執行命令的資訊。
track_activity_query_size	靜態	設定為 pg_stat_activity.current_query 保留的大小 (以位元組為單位)。
track_counts	動態	收集資料庫活動的統計資訊。
track_functions	動態	收集資料庫活動的功能層級統計資訊。
track_io_timing	動態	收集資料庫輸入/輸出活動的計時統計資訊。
transaction_deferrable	動態	指出是否延遲唯讀可序列化交易，直到可加以啟動為止 (沒有可能的序列化失敗)。

參數名稱	Apply_Type	描述
transaction_isolation	動態	設定目前交易的隔離層級。
transaction_read_only	動態	設定目前交易的唯讀狀態。
transform_null_equals	動態	將 expr=NULL 視為 expr IS NULL。
update_process_title	動態	更新程序標題以顯示作用中的 SQL 命令。
vacuum_cost_delay	動態	清理成本延遲 (以毫秒為單位)。
vacuum_cost_limit	動態	在小憩前可用的清理成本金額。
vacuum_cost_page_dirty	動態	清理所變更之頁面的清理成本。
vacuum_cost_page_hit	動態	在緩衝區快取中找到之頁面的清理成本。
vacuum_cost_page_miss	動態	在緩衝區快取中找不到之頁面的清理成本。
vacuum_defer_cleanup_age	動態	應延遲清理和熱清理 (如果有的話) 的交易數目。
vacuum_freeze_min_age	動態	清理應凍結資料表列的存留期下限。
vacuum_freeze_table_age	動態	清理應掃描整個資料表以凍結 tuple 的存留期。
wal_buffers	靜態	為 WAL 設定共享記憶體中的磁碟分頁緩衝區數目。
wal_writer_delay	動態	WAL 清空之間的 WAL 寫入器休眠時間。

參數名稱	Apply_Type	描述
work_mem	動態	設定要用於查詢工作空間的記憶體上限。
xmlbinary	動態	設定有多少二進位值要在 XML 中編碼。
xmloption	動態	設定要將明確剖析和序列化作業中的 XML 資料視為文件或內容片段。

Amazon RDS 會對所有參數使用預設 PostgreSQL 單位。下表顯示每個參數的 PostgreSQL 預設單位。

參數名稱	單位
archive_timeout	s
authentication_timeout	s
autovacuum_naptime	s
autovacuum_vacuum_cost_delay	ms
bgwriter_delay	ms
checkpoint_timeout	s
checkpoint_warning	s
deadlock_timeout	ms
effective_cache_size	8 KB
lock_timeout	ms
log_autovacuum_min_duration	ms
log_min_duration_statement	ms
log_rotation_age	分鐘

參數名稱	單位
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
max_standby_archive_delay	ms
max_standby_streaming_delay	ms
post_auth_delay	s
pre_auth_delay	s
segment_size	8 KB
shared_buffers	8 KB
statement_timeout	ms
ssl_renegotiation_limit	KB
tcp_keepalives_idle	s
tcp_keepalives_interval	s
temp_file_limit	KB
work_mem	KB
temp_buffers	8 KB
vacuum_cost_delay	ms
wal_buffers	8 KB
wal_receiver_timeout	ms

參數名稱	單位
wal_segment_size	B
wal_sender_timeout	ms
wal_writer_delay	ms
wal_receiver_status_interval	s

調校 RDS for PostgreSQL 的等待事件

等待事件是 RDS for PostgreSQL 的重要調校工具。如果您了解工作階段為何等待資源及其運作情形，就更能夠減少瓶頸。您可以使用本節的資訊來尋找可能的原因和更正動作。本節也討論基本 PostgreSQL 調校概念。

本節中的等待事件專屬於 RDS for PostgreSQL。

主題

- [RDS for PostgreSQL 調校的基本概念](#)
- [RDS for PostgreSQL 等待事件](#)
- [客戶端 : ClientRead](#)
- [客戶端 : ClientWrite](#)
- [CPU](#)
- [IO:BufFileRead 和 IO:BufFileWrite](#)
- [IO : DataFileRead](#)
- [IO:WALWrite](#)
- [Lock:advisory](#)
- [Lock:extend](#)
- [Lock:Relation](#)
- [Lock:transactionid](#)
- [Lock:tuple](#)
- [LWLock:BufferMapping \(LWLock:buffer_mapping\)](#)
- [LWLock:BufferIO \(IPC:BufferIO\)](#)
- [LWLock:buffer_content \(BufferContent\)](#)
- [LWLock:lock_manager \(LWLock:lockmanager\)](#)
- [Timeout:PgSleep](#)
- [Timeout:VacuumDelay](#)

RDS for PostgreSQL 調校的基本概念

調校 RDS for PostgreSQL 資料庫之前，請務必先瞭解什麼是等待事件及其發生原因。也請檢閱 RDS for PostgreSQL 的基本記憶體和磁碟架構。如需實用的架構圖，請參閱 [PostgreSQL](#) 維基教科書。

主題

- [RDS for PostgreSQL 等待事件](#)
- [RDS for PostgreSQL 記憶體](#)
- [RDS for PostgreSQL 程序](#)

RDS for PostgreSQL 等待事件

等待事件表示工作階段正在等待資源。例如，當 RDS for PostgreSQL 等待從用戶端接收資料時會發生等待事件 Client:ClientRead。工作階段通常會等待如下資源。

- 透過單一執行緒存取緩衝區，例如，當工作階段嘗試修改緩衝區時
- 另一個工作階段目前鎖定的資料列
- 資料檔讀取
- 日誌檔寫入

例如，為了滿足查詢，工作階段可能執行完整的資料表掃描。如果資料不在記憶體中，工作階段會等待磁碟輸入/輸出完成。將緩衝區讀入記憶體後，工作階段可能需要等待，因為其他工作階段正在存取這些緩衝區。資料庫使用預先定義的等待事件來記錄等待。這些事件分組為多個類別。

單一等待事件本身不表示效能問題。例如，如果請求的資料不在記憶體中，則需要從磁碟讀取資料。如果一個工作階段鎖定資料列來更新，則另一個工作階段要等待此資料列解除鎖定才能更新。遞交需要等待寫入日誌檔完成。等待是資料庫正常運作所不可或缺。

另一方面，大量等待事件通常顯示效能問題。在這種情況下，您可以使用等待事件資料來判斷工作階段將時間花在何處。例如，如果報告通常執行幾分鐘，但現在執行數小時，您可以識別佔總等待時間最多的等待事件。如果您可以查出最常等待事件的原因，通常就能做些改變來改善效能。例如，如果工作階段等待的資料列被另一個工作階段鎖定，您可以結束該鎖定工作階段。

RDS for PostgreSQL 記憶體

RDS for PostgreSQL 記憶體分成共用和本機。

主題

- [RDS for PostgreSQL 中的共用記憶體](#)
- [RDS for PostgreSQL 中的本機記憶體](#)

RDS for PostgreSQL 中的共用記憶體

RDS for PostgreSQL 在執行個體啟動時配置共用記憶體。共用記憶體分成多個子區域。接下來，您可以看到最重要子區域的描述。

主題

- [共用緩衝區](#)
- [預寫日誌 \(WAL\) 緩衝區](#)

共用緩衝區

共用緩衝集區是一種 RDS for PostgreSQL 記憶體區域，其中保留應用程式連線正在使用或已使用的所有分頁。分頁是記憶體形式的磁碟區塊。共用緩衝集區快取從磁碟讀取的資料區塊。集區可減少從磁碟重新讀取資料的次數，讓資料庫運作更有效率。

每個資料表和索引儲存為一連串固定大小的分頁。每個區塊包含多個元組，對應於資料列。元組可以存放在任何分頁中。

共用緩衝集區的記憶體有限。如果新請求需要的分頁不在記憶體中，且已沒有更多記憶體，RDS for PostgreSQL 會移出較不常用的分頁來容納請求。移出政策以時鐘掃描演算法實作。

`shared_buffers` 參數決定伺服器專用於快取資料的記憶體數量。

預寫日誌 (WAL) 緩衝區

預寫日誌 (WAL) 緩衝區保留交易資料，供 RDS for PostgreSQL 稍後寫入持久性儲存。RDS for PostgreSQL 可以透過 WAL 機制達成下列目標：

- 在故障後復原資料
- 避免頻繁寫入磁碟以減少磁碟輸入/輸出

當用戶端變更資料時，RDS for PostgreSQL 會將變更寫入 WAL 緩衝區。當用戶端發出 COMMIT 時，WAL 寫入器程序會將交易資料寫入 WAL 檔案。

`wal_level` 參數決定將多少資訊寫入 WAL。

RDS for PostgreSQL 中的本機記憶體

每個後端程序會配置本機記憶體來處理查詢。

主題

- [工作記憶體區域](#)
- [維護工作記憶體區域](#)
- [暫時緩衝區域](#)

工作記憶體區域

工作記憶體區域為執行排序和雜湊的查詢保留暫存資料。例如，含有 ORDER BY 子句的查詢執行排序。查詢在雜湊聯結和彙總中使用雜湊表。

`work_mem` 參數指定在寫入暫存磁碟檔案之前，供內部排序操作和雜湊表使用的記憶體數量。預設值為 4 MB。多個工作階段可以同時執行，每個工作階段可以平行執行維護操作。因此，使用的總工作記憶體可能是 `work_mem` 設定的倍數。

維護工作記憶體區域

維護工作記憶體區域快取維護操作的資料。這些操作包括清理、建立索引和新增外部索引鍵。

`maintenance_work_mem` 參數指定供維護操作使用的記憶體數量上限。預設值為 64 MB。一個資料庫工作階段一次只能執行一個維護操作。

暫時緩衝區域

暫時緩衝區域快取每個資料庫工作階段的暫存資料表。

每個工作階段視需要配置暫存緩衝區，以您指定的限制為上限。工作階段結束時，伺服器會清除緩衝區。

`temp_buffers` 參數設定每個工作階段使用的暫存緩衝區數目上限。在工作階段內第一次使用暫存資料表之前，您可以變更 `temp_buffers` 值。

RDS for PostgreSQL 程序

RDS for PostgreSQL 使用多個程序。

主題

- [郵件管理員程序](#)
- [後端程序](#)
- [背景程序](#)

郵件管理員程序

郵件管理員程序是您啟動 RDS for PostgreSQL 時啟動的第一個程序。郵件管理員程序有下列主要責任：

- 分叉和監控背景程序
- 接收來自用戶端程序的身分驗證請求，並在驗證請求之後才允許資料庫處理請求

後端程序

如果郵件管理員驗證用戶端請求，郵件管理員會分叉新的後端程序，也稱為 postgres 程序。一個用戶端程序只連線到一個後端程序。用戶端程序和後端程序直接通訊，無須郵件管理員程序介入。

背景程序

郵件管理員程序分叉幾個程序來執行不同的後端任務。一些較重要的程序包括：

- WAL 寫入器

RDS for PostgreSQL 將 WAL (預寫日誌) 緩衝區中的資料寫入日誌檔。預寫日誌的原則是直到資料庫將描述變更的日誌記錄寫入磁碟之後，資料庫才能將這些變更寫入資料檔。WAL 機制可減少磁碟輸入/輸出，並允許 RDS for PostgreSQL 在故障後使用日誌來復原資料庫。

- 背景寫入器

這個程序定期將已變更 (已修改) 分頁從記憶體緩衝區寫入資料檔。當後端程序在記憶體中修改分頁時，此分頁會變成已變更。

- 自動資料清理常駐程式

此常駐程式由下列組成：

- 自動資料清理啟動器
- 自動資料清理工作者程序

自動資料清理啟用時會檢查已插入、更新或刪除大量元組的資料表。此常駐程式有下列責任：

- 復原或重複使用已更新或刪除的資料列所佔用的磁碟空間
- 更新規劃員使用的統計數字
- 防止因交易 ID 環繞而遺失舊資料

自動資料清理功能可自動執行 VACUUM 和 ANALYZE 命令。VACUUM 有以下變體：標準和完整。標準清理與其他資料庫操作平行執行。VACUUM FULL 需要獨佔鎖定其處理的資料表。因此，無法與存取

同一個資料表的操作平行執行。VACUUM 會建立大量輸入/輸出流量，可能導致其他作用中工作階段的效能不佳。

RDS for PostgreSQL 等待事件

下表列出 RDS for PostgreSQL 中最常表示有效能問題的等待事件，並概述最常見的原因和更正動作。

等待事件	定義
客戶端 : ClientRead	此事件表示 RDS for PostgreSQL 正在等待從用戶端接收資料。
客戶端 : ClientWrite	此事件表示 RDS for PostgreSQL 正在等待將資料寫入用戶端。
CPU	此事件表示執行緒活躍於 CPU 中或正在等待 CPU。
IO:BufFileRead 和 IO:BufFileWrite	這些事件表示 RDS for PostgreSQL 建立暫存檔。
IO : DataFileRead	此事件表示連線等待後端程序從儲存讀取必要分頁，因為共用記憶體中沒有此分頁。
IO:WALWrite	此事件表示 RDS for PostgreSQL 正在等待預寫日誌 (WAL) 緩衝區寫入至 WAL 檔案。
Lock:advisory	此事件表示 PostgreSQL 應用程式使用鎖定在多個工作階段之間協調活動。
Lock:extend	此事件表示後端程序正在等待鎖定關聯來延伸，但另一個程序也基於相同目的而鎖定該關係。
Lock:Relation	此事件表示查詢正等待在目前由另一個交易鎖定的資料表或檢視表上取得鎖定。
Lock:transactionid	此事件表示交易正在等待資料列層級鎖定。
Lock:tuple	此事件表示後端程序正等待在元組上取得鎖定。

等待事件	定義
LWLock:BufferMapping (LWLock:buffer_mapping)	此事件表示工作階段正在等待將資料區塊與共用緩衝集區中的緩衝區建立關聯。
LWLock:BufferIO (IPC:BufferIO)	此事件表示 RDS for PostgreSQL 與其他程序同時嘗試存取分頁，正在等待其他程序完成輸入/輸出 (I/O) 操作。
LWLock:buffer_content (BufferContent)	此事件表示工作階段正等待在記憶體中讀取或寫入資料分頁，但另一個工作階段已鎖定該分頁來寫入。
LWLock:lock_manager (LWLock:lockmanager)	此事件表示因為無法執行快速路徑鎖定，RDS for PostgreSQL 引擎維護共用鎖定的記憶體區域來配置、檢查和解除配置鎖定。
Timeout:PgSleep	此事件表示伺服器程序已呼叫 <code>pg_sleep</code> 函數，正在等待睡眠逾時到期。
Timeout:VacuumDelay	此事件表示清空程序正在休眠，因為已達到預估的成本限制。

客戶端：ClientRead

Client:ClientRead 事件表示 RDS for PostgreSQL 正在等待從用戶端接收資料。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待時間增加的可能原因](#)
- [動作](#)

支援的引擎版本

RDS for PostgreSQL 第 10 版及更新版本支援此等待事件資訊。

Context

RDS for PostgreSQL 資料庫執行個體正在等待從用戶端接收資料。RDS for PostgreSQL 資料庫執行個體必須先從用戶端接收資料，才能將更多資料傳送至用戶端。執行個體從用戶端接收資料之前等待的時間是 `Client:ClientRead` 事件。

等待時間增加的可能原因

`Client:ClientRead` 事件出現在最常等待名單中的常見原因包括：

網路延遲較久

RDS for PostgreSQL 資料庫執行個體與用戶端之間的網路延遲可能變長。網路延遲較久會導致資料庫執行個體需要更多時間從用戶端接收資料。

用戶端的負載增加

用戶端可能面臨 CPU 壓力或網路飽和。用戶端的負載增加可能延遲將資料從用戶端傳輸至 RDS for PostgreSQL 資料庫執行個體。

網路往返太頻繁

如果 RDS for PostgreSQL 資料庫執行個體與用戶端之間網路往返太頻繁，可能會延遲將資料從用戶端傳輸至 RDS for PostgreSQL 資料庫執行個體。

複製操作龐大

在複製操作期間，資料從用戶端的檔案系統傳輸至 RDS for PostgreSQL 資料庫執行個體。如果將大量資料傳送至資料庫執行個體，可能會延遲將資料從用戶端傳輸至資料庫執行個體。

閒置用戶端連線

當用戶端連線至 `idle in transaction` 狀態的 RDS for PostgreSQL 資料庫執行個體時，資料庫執行個體可能會等待用戶端傳送更多資料或發出命令。處於此狀態的連線可能導致 `Client:ClientRead` 事件增加。

PgBouncer 用於連接池

PgBouncer 具有名為的低階網路組態設定 `pkt_buf`，預設設定為 4,096。如果工作負載透過傳送大於 4,096 位元組的查詢封包 PgBouncer，建議將 `pkt_buf` 設定增加到 8,192。如果新的設定未能減少 `Client:ClientRead` 事件，建議將 `pkt_buf` 設定增加為更大的值，例如 16,384 或 32,768。如果查詢文字很大，則較大的設定可能特別有用。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [將用戶端放在與執行個體相同的可用區域和 VPC 子網路中](#)
- [擴展用戶端](#)
- [使用最新一代的執行個體](#)
- [增加網路頻寬](#)
- [監控網路效能的最大值](#)
- [監控處於 "idle in transaction" 狀態的交易](#)

將用戶端放在與執行個體相同的可用區域和 VPC 子網路中

若要縮短網路延遲並增加網路輸送量，請將用戶端放在與 RDS for PostgreSQL 資料庫執行個體相同的可用區域和虛擬私有雲端 (VPC) 子網路中。確保用戶端的地理位置盡可能靠近資料庫執行個體。

擴展用戶端

使用 Amazon CloudWatch 或其他主機指標，判斷您的用戶端目前是否受到 CPU 或網路頻寬的限制，或兩者都受到限制。如果用戶端受到限制，請相應地擴展用戶端。

使用最新一代的執行個體

在某些情況下，您使用的資料庫執行個體類別可能不支援 Jumbo Frame。如果您在 Amazon EC2 上執行應用程式，請考慮為用戶端使用最新一代的執行個體。此外，在用戶端作業系統上設定最大傳輸單位 (MTU)。此技術可以減少網路往返次數並增加網路輸送量。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [巨型框架 \(9001 MTU\)](#)。

如需資料庫執行個體類別的相關資訊，請參閱 [資料庫執行個體類別](#)。若要決定等同於 Amazon EC2 執行個體類型的資料庫執行個體類別，請在 Amazon EC2 執行個體類型名稱前面加上 db.。例如，r5.8xlarge Amazon EC2 執行個體等同於 db.r5.8xlarge 資料庫執行個體類別。

增加網路頻寬

使用 NetworkReceiveThroughput 和 NetworkTransmitThroughput Amazon CloudWatch 指標監控資料庫執行個體上的傳入和傳出網路流量。這些指標可協助您判斷網路頻寬是否足以應付工作負載。

如果網路頻寬不夠，請增加頻寬。如果用 AWS 戶端或您的資料庫執行個體達到網路頻寬限制，增加頻寬的唯一方法是增加資料庫執行個體大小。如需詳細資訊，請參閱 [資料庫執行個體類別的類型](#)。

如需 CloudWatch 測量結果的詳細資訊，請參閱 [Amazon RDS 的 Amazon CloudWatch 指標](#)。

監控網路效能的最大值

如果您使用 Amazon EC2 用戶端，Amazon EC2 提供網路效能指標的最大值，包括傳入和傳出網路總頻寬。也提供連線追蹤來確保如預期傳回封包，還有網域名稱系統 (DNS) 等服務的連結本機服務存取。若要監控這些最大值，請使用目前的增強型網路驅動程式，以監控用戶端的網路效能。

如需詳細資訊，請參閱 Amazon EC2 [使用者指南中的監控 Amazon EC2 執行個體的網路效能](#)，以及在 Amazon EC2 使用者指南 [中監控 Amazon EC2 執行個體的網路效能](#)。

監控處於 "idle in transaction" 狀態的交易

檢查 idle in transaction 連線是否變多。作法是監控 pg_stat_activity 資料表的 state 資料欄。您可以執行類似下列的查詢來識別連線來源。

```
select client_addr, state, count(1) from pg_stat_activity
where state like 'idle in transaction%'
group by 1,2
order by 3 desc
```

客戶端：ClientWrite

Client:ClientWrite 事件表示 RDS for PostgreSQL 正在等待將資料寫入用戶端。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

RDS for PostgreSQL 第 10 版及更新版本支援此等待事件資訊。

Context

用戶端程序必須先讀取從 RDS for PostgreSQL 資料庫叢集收到的所有資料，叢集才能傳送更多資料。叢集將更多資料傳送至用戶端之前等待的時間是 Client:ClientWrite 事件。

RDS for PostgreSQL 資料庫執行個體與用戶端之間的網路輸送量減少可能造成此事件。用戶端的 CPU 壓力和網路飽和也可能造成此事件。CPU 壓力表示 CPU 耗盡，但有任務正在等待 CPU 時間。網路飽和表示資料庫與用戶端之間的網路傳送太多資料，應付不來。

等待變多的可能原因

Client:ClientWrite 事件出現在最常等待名單中的常見原因包括：

網路延遲較久

RDS for PostgreSQL 資料庫執行個體與用戶端之間的網路延遲可能變長。網路延遲較久會導致用戶端需要更多時間來接收資料。

用戶端的負載增加

用戶端可能面臨 CPU 壓力或網路飽和。用戶端的負載增加會延遲從 RDS for PostgreSQL 資料庫執行個體接收資料。

傳送大量資料給用戶端

RDS for PostgreSQL 資料庫執行個體可能將大量資料傳送至用戶端。用戶端可能來不及接收叢集傳送的資料。例如，複製大型資料表這種活動可能導致 Client:ClientWrite 事件增加。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [將用戶端放在與叢集相同的可用區域和 VPC 子網路中](#)
- [使用最新一代的執行個體](#)
- [減少傳送至用戶端的資料量](#)
- [擴展用戶端](#)

將用戶端放在與叢集相同的可用區域和 VPC 子網路中

若要縮短網路延遲並增加網路輸送量，請將用戶端放在與 RDS for PostgreSQL 資料庫執行個體相同的可用區域和虛擬私有雲端 (VPC) 子網路中。

使用最新一代的執行個體

在某些情況下，您使用的資料庫執行個體類別可能不支援 Jumbo Frame。如果您在 Amazon EC2 上執行應用程式，請考慮為用戶端使用最新一代的執行個體。此外，在用戶端作業系統上設定最大傳輸單位 (MTU)。此技術可以減少網路往返次數並增加網路輸送量。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [巨型框架 \(9001 MTU\)](#)。

如需資料庫執行個體類別的相關資訊，請參閱 [資料庫執行個體類別](#)。若要決定等同於 Amazon EC2 執行個體類型的資料庫執行個體類別，請在 Amazon EC2 執行個體類型名稱前面加上 db.。例如，r5.8xlarge Amazon EC2 執行個體等同於 db.r5.8xlarge 資料庫執行個體類別。

減少傳送至用戶端的資料量

可能的話，請調整應用程式，以減少 RDS for PostgreSQL 資料庫執行個體傳送至用戶端的資料量。這樣調整可以減輕用戶端的 CPU 和網路爭用情形。

擴展用戶端

使用 Amazon CloudWatch 或其他主機指標，判斷您的用戶端目前是否受到 CPU 或網路頻寬的限制，或兩者都受到限制。如果用戶端受到限制，請相應地擴展用戶端。

CPU

此事件表示執行緒活躍於 CPU 中或正在等待 CPU。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

此等待事件資訊與所有 RDS for PostgreSQL 版本有關。

Context

中央處理單元 (CPU) 是執行指令的電腦元件。例如，CPU 指令執行算術運算，並在記憶體中交換資料。如果查詢增加更多指令來透過資料庫引擎執行，則查詢會執行更久。CPU 排程將 CPU 時間撥給程序。排程由作業系統的核心協調。

主題

- [如何得知發生這種等待](#)
- [DBLoadCPU 指標](#)
- [os.cpuUtilization 指標](#)
- [CPU 排程的可能原因](#)

如何得知發生這種等待

此 CPU 等待事件表示後端程序活躍於 CPU 中或正在等待 CPU。當查詢顯示下列資訊時，就表示發生此事件：

- `pg_stat_activity.state` 欄具有值 `active`。
- `pg_stat_activity` 中的 `wait_event_type` 和 `wait_event` 欄皆為 `null`。

若要查看正在使用或等待 CPU 的後端程序，請執行下列查詢。

```
SELECT *
FROM   pg_stat_activity
WHERE  state = 'active'
AND    wait_event_type IS NULL
AND    wait_event IS NULL;
```

DBLoadCPU 指標

CPU 的績效詳情指標為 `DBLoadCPU`。`DBLoadCPU` 的值與 Amazon CloudWatch 指標 `CPUUtilization` 的值可能不同。後者是從 HyperVisor 為資料庫執行個體收集的指標。

os.cpuUtilization 指標

績效詳情作業系統指標提供 CPU 使用率的詳細資訊。例如，您可以顯示下列指標：

- `os.cpuUtilization.nice.avg`

- `os.cpuUtilization.total.avg`
- `os.cpuUtilization.wait.avg`
- `os.cpuUtilization.idle.avg`

績效詳情以 `os.cpuUtilization.nice.avg` 報告資料庫引擎的 CPU 使用率。

CPU 排程的可能原因

作業系統核心會處理 CPU 的排程。CPU 作用中時，程序可能需要等待排程。CPU 在執行運算時處於作用中狀態。當它具有未執行的閒置執行緒 (亦即，等待記憶體 I/O 的執行緒) 時，也是處於作用中狀態。此類型的 I/O 主導典型資料庫工作負載。

滿足下列條件時，程序可能等待排定使用 CPU：

- CloudWatch CPUUtilization 指標接近 100%。
- 平均負載大於 vCPU 數目，表示負載過重。您可以在績效詳情的作業系統指標區段中找到 `loadAverageMinute` 指標。

等待變多的可能原因

CPU 等待事件比平時更常發生時，可能表示有效能問題，典型原因包括下列各項。

主題

- [突然激增的可能原因](#)
- [長期頻繁的可能原因](#)
- [罕見情況](#)

突然激增的可能原因

突然激增最可能的原因如下：

- 應用程式對資料庫同時開啟太多連線。這種情況稱為「連線風暴」。
- 應用程式工作負載出現下列任何變化：
 - 新查詢
 - 資料集變大
 - 維護或建立索引

- 新函數
- 新運算子
- 平行執行查詢增加
- 查詢執行計劃已變更。在某些情況下，變更可能造成緩衝區增加。例如，查詢原本使用索引，但現在使用循序掃描。在此情況下，查詢需要更多 CPU 才能完成相同的目標。

長期頻繁的可能原因

事件長期復發最可能的原因：

- CPU 上同時執行太多後端程序。這些程序可能是平行工作者。
- 查詢表現欠佳，因為需要大量緩衝區。

罕見情況

如果可能原因都不是真正原因，可能表示發生下列情況：

- CPU 正在換入換出程序。
- 如果大型頁面功能已關閉，則 CPU 可能正在管理頁面表格項目。除了微型、小型和中型資料庫執行個體類之外，所有資料庫執行個體類都預設開啟此記憶體管理功能。如需更多詳細資訊，請參閱 [RDS for PostgreSQL 的巨型分頁](#)。

動作

如果 CPU 等待事件在資料庫活動中佔多數，則不見得表示有效能問題。效能降低時才需要回應此事件。

主題

- [調查資料庫是否造成 CPU 使用率上升](#)
- [判斷連線數目是否增加](#)
- [回應工作負載變更](#)

調查資料庫是否造成 CPU 使用率上升

在績效詳情中檢查 `os.cpuUtilization.nice.avg` 指標。如果此值遠低於 CPU 使用率，表示佔用 CPU 以非資料庫的程序為主。

判斷連線數目是否增加

在 Amazon CloudWatch 中檢查 DatabaseConnections 指標。您的動作取決於在 CPU 等待事件變多期間，連線數目是增加還是減少。

連線增加

如果連線數目上升，請將耗用 CPU 的後端程序數目與 vCPU 數目做比較。可能的情況如下：

- 耗用 CPU 的後端程序數目小於 vCPU 數目。

在此情況下，連線數目不是問題。但您仍然可以嘗試降低 CPU 使用率。

- 耗用 CPU 的後端程序數目大於 vCPU 數目。

在此情況下，請考慮下列選項：

- 將連線至資料庫的後端程序數目減少。例如，實作連線集區解決方案，例如 RDS Proxy。如需進一步了解，請參閱[使用 Amazon RDS Proxy](#)。
- 加大執行個體以取得更多 vCPU。
- 如適用，請將一些唯讀工作負載重新導向至讀取器節點

連線未增加

在績效詳情中檢查 blks_hit 指標。尋找 blks_hit 增加與 CPU 使用率之間的關聯性。可能的情況如下：

- CPU 使用率與 blks_hit 有關。

在此情況下，請找出與 CPU 使用率最有關的 SQL 陳述式，並尋找計劃變更。您可以使用下列任一技巧：

- 手動解釋計劃，並與預期的執行計劃做比較。
- 查明每秒區塊命中數和每秒本機區塊命中數是否增加。在績效詳情儀表板的 Top SQL (常用 SQL) 區段中，選擇 Preferences (偏好設定)。
- CPU 使用率與 blks_hit 無關。

在此情況下，請判斷是否發生下列任何情形：

- 應用程式與資料庫之間快速連線和斷線。

請開啟 `log_connections` 和 `log_disconnections`，然後分析 PostgreSQL 日誌，以診斷此行為。考慮使用 `pgbadger` 日誌分析器。如需詳細資訊，請參閱 <https://github.com/darold/pgbadger>。

- 作業系統超載。

在此情況下，績效詳情會指出後端程序耗用 CPU 比平常更久。在績效詳情 `os.cpuUtilization` 指標或 CloudWatch `CPUUtilization` 指標中查證。如果作業系統超載，請檢查增強型監控指標以進一步診斷。具體來說，請檢查程序清單及每個程序耗用的 CPU 百分比。

- 常用 SQL 陳述式耗用太多 CPU。

檢查與 CPU 使用率有關的陳述式，了解是否可以少用 CPU。執行 `EXPLAIN` 命令，專注於影響最大的計劃節點上。考慮使用 PostgreSQL execution plan visualizer。若要試用此工具，請參閱 <http://explain.dalibo.com/>。

回應工作負載變更

如果工作負載已變更，請尋找下列類型的變更：

新查詢

檢查是否需要新的查詢。如果是，請確定其執行計劃和每秒執行次數合理。

資料集變大

決定是否應該分割 (如果尚未分割)。此策略可以減少查詢需要擷取的分頁數。

維護或建立索引

檢查是否需要排定維護。最佳實務是將維護活動排定在尖峰活動之外。

新函數

檢查這些函數在測試期間是否正常執行。具體來說，請檢查每秒執行次數是否合理。

新運算子

檢查在測試期間是否正常執行。

平行執行查詢增加

判斷是否發生下列任何情況：

- 涉及的關聯或索引突然變大，明顯不同於 `min_parallel_table_scan_size` 或 `min_parallel_index_scan_size`。
- 最近變更 `parallel_setup_cost` 或 `parallel_tuple_cost`。
- 最近變更 `max_parallel_workers` 或 `max_parallel_workers_per_gather`。

IO:BufFileRead 和 IO:BufFileWrite

IO:BufFileRead 和 IO:BufFileWrite 事件表示 RDS for PostgreSQL 建立暫存檔。如果操作需要的記憶體超過工作記憶體參數目前的定義，則會將暫存資料寫入永久性儲存。此操作有時稱為「溢出到磁碟」。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

IO:BufFileRead 和 IO:BufFileWrite 與工作記憶體區域和維護工作記憶體區域有關。如需這些本機記憶體區域的詳細資訊，請參閱 PostgreSQL 文件中的[資源耗用](#)。

`work_mem` 的預設值為 4 MB。如果一個工作階段平行執行操作，則負責平行處理的每個工作者會使用 4 MB 的記憶體。因此，請小心設定 `work_mem`。如果將此值設得太大，則執行許多工作階段的資料庫可能會耗用太多記憶體。如果將此值設得太低，RDS for PostgreSQL 會在本機儲存體中建立暫存檔。這些暫存檔的磁碟輸入/輸出可能造成效能降低。

如果您看到下列事件序列，表示資料庫可能正在產生暫存檔：

1. 可用性突然遽降
2. 可用空間迅速復原

您也可能看到「鏈鋸」模式。此模式可能表示資料庫不斷建立小檔案。

等待變多的可能原因

一般而言，這些等待事件起因於操作耗用超過 `work_mem` 或 `maintenance_work_mem` 參數所配置的記憶體。為了補償，操作寫入暫存檔。IO:BufFileRead 和 IO:BufFileWrite 事件的常見原因包括：

查詢需要比工作記憶體區域中更多的記憶體

具有下列特性的查詢使用工作記憶體區域：

- 雜湊聯結。
- ORDER BY 子句
- GROUP BY 子句
- DISTINCT
- 範圍函數
- CREATE TABLE AS SELECT
- 具體化檢視表重新整理

陳述式需要比維護工作記憶體區域中更多的記憶體

下列陳述式使用維護工作記憶體區域：

- CREATE INDEX
- CLUSTER

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [識別問題](#)
- [檢查聯結查詢](#)
- [檢查 ORDER BY 和 GROUP BY 查詢](#)
- [避免使用 DISTINCT 操作](#)
- [考慮使用視窗函數代替 GROUP BY 函數](#)
- [調查具體化檢視表和 CTAS 陳述式](#)
- [重建索引時使用 pg_repack](#)

- [叢集化資料表時提高 maintenance_work_mem](#)
- [調整記憶體以防止 IO:BufFileRead 和 IO:BufFileWrite](#)

識別問題

假設績效詳情未開啟，而您懷疑 IO:BufFileRead 和 IO:BufFileWrite 比平時更常發生。若要識別問題的來源，您可以設定 `log_temp_files` 參數來記錄所有產生超過指定閾值 KB 暫存檔的查詢。根據預設，`log_temp_files` 會設為 -1，這會關閉此記錄功能。如果您將此參數設為 0，RDS for PostgreSQL 會記錄所有暫存檔。如果您將其設為 1024，RDS for PostgreSQL 會記錄所有產生大於 1 MB 暫存檔的查詢。如需 `log_temp_files` 的詳細資訊，請參閱 PostgreSQL 文件中的 [錯誤報告和日誌記錄](#)。

檢查聯結查詢

您的查詢很可能使用聯結。例如，下列查詢會聯結四個資料表。

```
SELECT *
  FROM "order"
 INNER JOIN order_item
  ON (order.id = order_item.order_id)
 INNER JOIN customer
  ON (customer.id = order.customer_id)
 INNER JOIN customer_address
  ON (customer_address.customer_id = customer.id AND
      order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

暫存檔使用量激增的可能原因在於查詢本身有問題。例如，不標準的子句可能無法正確篩選聯結。請看下列範例中的第二個內部聯結。

```
SELECT *
  FROM "order"
 INNER JOIN order_item
  ON (order.id = order_item.order_id)
 INNER JOIN customer
  ON (customer.id = customer.id)
 INNER JOIN customer_address
  ON (customer_address.customer_id = customer.id AND
      order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

上述查詢誤將 `customer.id` 連結至 `customer.id`，導致在每個客戶與每筆訂單之間產生笛卡爾乘積。這種意外連結會產生大型暫存檔。根據資料表的大小，笛卡爾查詢甚至可能導致填滿儲存。有下列情況時，表示應用程式可能有笛卡爾連結：

- 您看到儲存可用性大幅遽降，接著迅速復原。
- 未建立索引。
- 未發出 `CREATE TABLE FROM SELECT` 陳述式。
- 未重新整理具體化檢視表。

若要檢查是否以適當索引鍵連結資料表，請檢驗查詢和物件關聯式映射指令。切記，不一定會呼叫應用程式的某些查詢，有些查詢是動態產生。

檢查 ORDER BY 和 GROUP BY 查詢

在某些情況下，ORDER BY 子句可能導致產生過多暫存檔。請考量下列準則：

- 只將需要排序的資料欄放入 ORDER BY 子句中。如果查詢傳回數千個資料列，並在 ORDER BY 子句中指定許多資料欄，此準則尤其重要。
- 當 ORDER BY 子句比對的資料欄有相同遞增或遞減順序時，請考慮建立索引以加速執行。最好是局部索引，因為較小。讀取和周遊較小的索引比較快。
- 如果您為可接受空值的資料欄建立索引，請決定要將空值存放在索引結尾還是開頭。

可能的話，請篩選結果集，以減少需要排序的資料列數。如果您使用 WITH 子句陳述式或子查詢，請記住，內部查詢會產生結果集並傳給外部查詢。查詢篩選掉越多資料列，查詢就越不需要排序。

- 如果不需要取得完整結果集，請使用 LIMIT 子句。例如，如果您只想要前五個資料列，則在查詢中使用 LIMIT 子句就不會一直產生結果。如此，查詢只需要較少的記憶體和暫存檔。

使用 GROUP BY 子句的查詢可能也需要暫存檔。GROUP BY 查詢使用如下函數來彙總值：

- COUNT
- AVG
- MIN
- MAX
- SUM
- STDDEV

若要調校 GROUP BY 查詢，請遵循 ORDER BY 查詢的建議。

避免使用 DISTINCT 操作

可能的話，請避免使用 DISTINCT 操作來移除重複的資料列。查詢傳回不必要和重複的資料列越多，DISTINCT 操作的成本越高。可能的話，請在 WHERE 子句中增加篩選條件，即使不同的資料表使用相同的篩選器也無妨。篩選查詢並正確聯結可改善效能和減少使用資源。還可防止不正確的報告和結果。

如果需要對同一個資料表的多個資料列使用 DISTINCT，請考慮建立複合索引。將多個資料列組合成一個索引，可縮短相異資料列的評估時間。此外，如果您使用 RDS for PostgreSQL 第 10 版或更新版本，則可以使用 CREATE STATISTICS 命令，將多個資料欄之間的統計數字相互關聯。

考慮使用視窗函數代替 GROUP BY 函數

使用 GROUP BY 時，您變更結果集，然後擷取彙總結果。使用視窗函數時，您彙總資料而不變更結果集。視窗函數使用 OVER 子句來跨查詢所定義的集執行計算，使資料列彼此相互關聯。您在視窗函數中可以使用所有 GROUP BY 函數，但也可使用如下函數：

- RANK
- ARRAY_AGG
- ROW_NUMBER
- LAG
- LEAD

若要盡量減少視窗函數產生的暫存檔，當需要兩個相異彙總時，請移除相同結果集的重複部分。請看下列查詢。

```
SELECT sum(salary) OVER (PARTITION BY dept ORDER BY salary DESC) as sum_salary
       , avg(salary) OVER (PARTITION BY dept ORDER BY salary ASC) as avg_salary
FROM empsalary;
```

您可以使用 WINDOW 子句重寫查詢，如下所示。

```
SELECT sum(salary) OVER w as sum_salary
       , avg(salary) OVER w as_avg_salary
FROM empsalary
WINDOW w AS (PARTITION BY dept ORDER BY salary DESC);
```

根據預設，RDS for PostgreSQL 執行規劃工具會合併相似的節點，以免重複操作。不過，使用視窗區塊的明確宣告可以更輕鬆維護查詢。防止重複也可以改善效能。

調查具體化檢視表和 CTAS 陳述式

具體化檢視表重新整理時會執行查詢。此查詢可以包含 GROUP BY、ORDER BY 或 DISTINCT 等操作。在重新整理期間，您可能看到大量暫存檔及等待事件 IO:BufFileWrite 和 IO:BufFileRead。同樣地，當您根據 SELECT 陳述式建立資料表時，CREATE TABLE 陳述式會執行查詢。若要減少所需的暫存檔，請最佳化查詢。

重建索引時使用 pg_repack

當您建立索引時，引擎會排序結果集。隨著資料表變大，以及索引資料欄的值變得更多樣化，暫存檔需要更多空間。在大多數情況下，除非修改維護工作記憶體區域，否則無法阻止為大型資料表建立暫存檔。如需 maintenance_work_mem 的詳細資訊，請參閱 PostgreSQL 文件中的 <https://www.postgresql.org/docs/current/runtime-config-resource.html>。

重新建立大型索引時，可能的解決方法是使用 pg_repack 延伸模組。如需詳細資訊，請參閱 pg_repack 文件中的 [以最少鎖定重組 PostgreSQL 資料庫中的資料表](#)。如需在 RDS for PostgreSQL 資料庫執行個體中設定延伸模組的相關資訊，請參閱 [使用 pg_repack 擴充功能減少資料表和索引膨脹](#)。

叢集化資料表時提高 maintenance_work_mem

CLUSTER 命令根據 index_name 指定的現有索引，以叢集化 table_name 指定的資料表。RDS for PostgreSQL 實際上會重新建立資料表，以符合特定索引的順序。

在磁帶儲存盛行的年代，因為儲存輸送量有限，叢集很普遍。如今 SSD 型儲存很普遍，較不流行叢集。不過，如果將資料表叢集化，還是可稍微提高效能，視資料表大小、索引、查詢等而定。

如果您執行 CLUSTER 命令，然後看到等待事件 IO:BufFileWrite 和 IO:BufFileRead，請調校 maintenance_work_mem。請將記憶體調到很大。較大的值表示引擎可以使用更多記憶體執行叢集操作。

調整記憶體以防止 IO:BufFileRead 和 IO:BufFileWrite

在某些情況下，您需要調整記憶體。您的目標是使用適當的參數來平衡下列消用區域的記憶體，如下所示。

- work_mem 值。
- 折除 shared_buffers 值之後剩餘的記憶體

- 已開啟和使用中的連線數上限，受限於 `max_connections`

如需如何調整記憶體의詳細資訊，請參閱 PostgreSQL 文件中的[資源耗用](#)。

增加工作記憶體區域的大小

在某些情況下，您只能選擇增加工作階段所使用的記憶體。如果查詢撰寫無誤，且使用正確的索引鍵來聯結，請考慮提高 `work_mem` 值。

若要了解查詢產生多少暫存檔，請將 `log_temp_files` 設定為 0。如果將 `work_mem` 值提高到日誌中指出的最大值，就可以防止查詢產生暫存檔。然而，`work_mem` 會針對每個連線或平行工作者，設定每個計劃節點的最大值。如果資料庫有 5,000 個連線，且每個連線各使用 256 MiB 的記憶體，則引擎需要 1.2 TiB 的 RAM。因此，執行個體可能記憶體不足。

為共用緩衝集區保留足夠記憶體

資料庫不只使用工作記憶體區域，還使用共用緩衝集區之類的記憶體區域。提高 `work_mem` 之前，請考慮這些額外記憶體區域的需求。

例如，假設您的 RDS for PostgreSQL 執行個體類別為 `db.r5.2xlarge`。此類別有 64 GiB 的記憶體。預設會保留 25% 的記憶體給共用緩衝集區。減去配置給共用記憶體區域的數量後，剩下 16,384 MB。請勿將剩餘的記憶體全部配置給工作記憶體區域，因為作業系統和引擎也需要記憶體。

可配置給 `work_mem` 的記憶體取決於執行個體類別。使用越大的執行個體類別，可用的記憶體越多。不過，在上述範例中，最多只能使用 16 GiB。否則，當記憶體不足時，就無法使用執行個體。為了讓執行個體從無法使用狀態中復原，RDS for PostgreSQL 自動化服務會自動重新啟動。

管理連線數目

假設您的資料庫執行個體有 5,000 個同時連線。每個連線至少使用 4 MiB 的 `work_mem`。連線耗用大量記憶體可能導致效能降低。因應之道如下：

- 提升為更大的執行個體類別。
- 使用連線代理或集區來減少同時的資料庫連線數。

關於代理，根據您的應用程式而定，請考慮 Amazon RDS Proxy、pgBouncer 或連線集區。此解決方案可減輕 CPU 負載。面臨所有連線都需要工作記憶體區域時，也能降低風險。在只有少數資料庫連線時，您可以提高 `work_mem` 的值。如此就能減少 `IO:BufFileRead` 和 `IO:BufFileWrite` 等待事件。等待工作記憶體區域的查詢也會大幅加速。

IO : DataFileRead

IO:DataFileRead 事件表示因為所需的分頁不在共用記憶體中，連線等待後端程序從儲存讀取該分頁。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待時間增加的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

所有查詢和資料操作 (DML) 作業會存取緩衝集區中的分頁。引起讀取的陳述式包括 SELECT、UPDATE 及 DELETE。例如，UPDATE 可以從資料表或索引讀取分頁。如果所請求或更新的頁面不在共用緩衝集區中，則此讀取可能引起 IO:DataFileRead 事件。

共用緩衝集區有限，可能填滿。在此情況下，請求的分頁不在記憶體中，迫使資料庫從磁碟讀取區塊。如果 IO:DataFileRead 事件經常發生，可能表示共用緩衝集區太小，不足以應付工作負載。這是嚴重問題，因為 SELECT 查詢讀取大量資料列，塞不進緩衝集區。如需緩衝集區的詳細資訊，請參閱 PostgreSQL 文件中的[資源耗用](#)。

等待時間增加的可能原因

IO:DataFileRead 的常見原因包括：

連線尖峰

您可能會發現多個連接生成相同數量的 IO : DataFileRead wait 事件。在此情況下，IO:DataFileRead 事件可能出現尖峰 (突然大幅增加)。

SELECT 和 DML 陳述式執行循序掃描

您的應用程式可能執行新的操作。或者，現有的操作可能因為新的執行計劃而變更。在這種情況下，請尋找 seq_scan 值較大的資料表 (特別是大型資料表)。查詢 pg_stat_user_tables 來尋找。若要追蹤哪些查詢產生較多讀取操作，請使用延伸 pg_stat_statements。

大型資料集的 CTAS 和 CREATE INDEX

CTAS 代表 CREATE TABLE AS SELECT 陳述式。如果您以大型資料集為來源執行 CTAS，或在大型資料表上建立索引，可能會發生 IO:DataFileRead 事件。建立索引時，資料庫可能需要使用循序掃描來讀取整個物件。當分頁不在記憶體中時，CTAS 會產生 IO:DataFile 讀取。

多個清理工作者同時執行

清理工作者是手動或自動觸發。建議採取積極清理策略。不過，當資料表更新或刪除許多資料列時，IO:DataFileRead 等待會變多。回收空間後，花在 IO:DataFileRead 的清理時間就會減少。

擷取大量資料

當應用程式擷取大量資料時，ANALYZE 操作可能更頻繁發生。ANALYZE 程序可以由自動資料清理啟動器觸發，或手動叫用。

ANALYZE 操作讀取資料表的子集。將 30 乘以 default_statistics_target 值，即可算出必須掃描的分頁數。如需詳細資訊，請參閱 [PostgreSQL 文件](#)。default_statistics_target 參數接受 1 到 10,000 之間的值，預設值為 100。

資源耗盡

如果耗用執行個體網路頻寬或 CPU，IO:DataFileRead 事件可能更頻繁發生。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [對產生等待的查詢檢查述詞篩選條件](#)
- [將維護操作的影響降至最低](#)
- [因應大量連線](#)

對產生等待的查詢檢查述詞篩選條件

假設您發現特定的查詢正在產生 IO:DataFileRead 等待事件。請利用下列技巧來識別：

- 績效詳情
- 目錄檢視表，例如延伸 pg_stat_statements 提供的檢視表
- 目錄檢視表 pg_stat_all_tables (如果定期指出實體讀取變多)

- `pg_statio_all_tables` 檢視表 (如果指出 `_read` 計數器上升)

建議您判斷這些查詢的述詞中 (WHERE 子句) 使用哪些篩選條件。請遵守下列準則：

- 執行 EXPLAIN 命令。在輸出中，識別使用的掃描類型。循序掃描並不一定代表有問題。使用循序掃描的查詢，當然比使用篩選條件的查詢產生更多 `IO:DataFileRead` 事件。

查明列在 WHERE 子句中的資料欄是否已編成索引。如果不是，請考慮為此資料欄建立索引。這種方法可避免循序掃描，並減少 `IO:DataFileRead` 事件。如果查詢有嚴格的篩選條件，但仍產生循序掃描，請評估使用的索引是否適當。

- 查明查詢是否存取非常大的資料表。在某些情況下，將資料表分割可以改善效能，讓查詢只讀取必要的分割區。
- 檢查聯結操作中的基數 (總資料列數)。請注意您在 WHERE 子句的過濾條件中傳入的值有多嚴格。可能的話，請調整查詢，以減少在計劃的每個步驟中傳入的資料列數目。

將維護操作的影響降至最低

維護操作很重要，例如 VACUUM 和 ANALYZE。建議不要因為發現這些維護操作相關的 `IO:DataFileRead` 等待事件而關閉維護。下列方法可以將這些操作的影響降至最低：

- 在離峰時段手動執行維護操作。這項技巧可防止資料庫達到自動化操作的閾值。
- 如果資料表非常大，請考慮分割資料表。這項技巧可減少維護操作的額外負荷。資料庫只會存取需要維護的分割區。
- 擷取大量資料時，請考慮停用自動分析功能。

下列公式成立時會自動對資料表觸發自動資料清理功能。

```
pg_stat_user_tables.n_dead_tup > (pg_class.reltuples x autovacuum_vacuum_scale_factor)
+ autovacuum_vacuum_threshold
```

檢視表 `pg_stat_user_tables` 和目錄 `pg_class` 有多個資料列。一個資料列可以對應於資料表中的一個資料列。這個公式假設 `reltuples` 專用於特定資料表。通常是為整個執行個體來整體設定參數 `autovacuum_vacuum_scale_factor` (預設為 0.20) 和 `autovacuum_vacuum_threshold` (預設為 50 個元組)。但您可以針對特定資料表設定不同的值。

主題

- [尋找不必要耗用空間的資料表](#)

- [尋找不必要耗用空間的索引](#)
- [尋找適合自動資料清理的資料表](#)

尋找不必要耗用空間的資料表

若要尋找不必要耗用空間的資料表，您可以使用 PostgreSQL `pgstattuple` 延伸模組中的函數。根據預設，所有 RDS for PostgreSQL 資料庫執行個體都可以使用此延伸模組 (模組)，並可以使用下列命令在執行個體上具體化。

```
CREATE EXTENSION pgstattuple;
```

如需此延伸模組的詳細資訊，請參閱 PostgreSQL 文件中的 [pgstattuple](#)。

您可以在應用程式中檢查資料表和索引膨脹。如需詳細資訊，請參閱[診斷資料表和索引膨脹](#)。

尋找不必要耗用空間的索引

若要尋找膨脹的索引，並預估您對其具有讀取權限之資料表上不必要的空間耗用量，您可以執行下列查詢。

```
-- WARNING: rows with is_na = 't' are known to have bad statistics ("name" type is not
supported).
-- This query is compatible with PostgreSQL 8.2 and later.

SELECT current_database(), nspname AS schemaname, tblname, idxname,
bs*(relpages)::bigint AS real_size,
bs*(relpages-est_pages)::bigint AS extra_size,
100 * (relpages-est_pages)::float / relpages AS extra_ratio,
fillfactor, bs*(relpages-est_pages_ff) AS bloat_size,
100 * (relpages-est_pages_ff)::float / relpages AS bloat_ratio,
is_na
-- , 100-(sub.pst).avg_leaf_density, est_pages, index_tuple_hdr_bm,
-- maxalign, pagehdr, nulldatawidth, nulldatahdrwidth, sub.reltuples, sub.relpages
-- (DEBUG INFO)
FROM (
  SELECT coalesce(1 +
    ceil(reltuples/floor((bs-pageopqdata-pagehdr)/(4+nulldatahdrwidth)::float)), 0
    -- ItemIdData size + computed avg size of a tuple (nulldatahdrwidth)
  ) AS est_pages,
  coalesce(1 +
    ceil(reltuples/floor((bs-pageopqdata-pagehdr)*fillfactor/
(100*(4+nulldatahdrwidth)::float))), 0
```



```

) AS est_pages_ff,
bs, nspname, table_oid, tblname, idxname, relpages, fillfactor, is_na
-- , stattuple.pgstatindex(quote_ident(nspname)||'.'||quote_ident(idxname)) AS
pst,
-- index_tuple_hdr_bm, maxalign, pagehdr, nulldatawidth, nulldatahdrwidth,
reltuples
-- (DEBUG INFO)
FROM (
SELECT maxalign, bs, nspname, tblname, idxname, reltuples, relpages, relam,
table_oid, fillfactor,
( index_tuple_hdr_bm +
maxalign - CASE -- Add padding to the index tuple header to align on MAXALIGN
WHEN index_tuple_hdr_bm%maxalign = 0 THEN maxalign
ELSE index_tuple_hdr_bm%maxalign
END
+ nulldatawidth + maxalign - CASE -- Add padding to the data to align on
MAXALIGN
WHEN nulldatawidth = 0 THEN 0
WHEN nulldatawidth::integer%maxalign = 0 THEN maxalign
ELSE nulldatawidth::integer%maxalign
END
)::numeric AS nulldatahdrwidth, pagehdr, pageopqdata, is_na
-- , index_tuple_hdr_bm, nulldatawidth -- (DEBUG INFO)
FROM (
SELECT
i.nspname, i.tblname, i.idxname, i.reltuples, i.relpages, i.relam, a.attrelid
AS table_oid,
current_setting('block_size')::numeric AS bs, fillfactor,
CASE -- MAXALIGN: 4 on 32bits, 8 on 64bits (and mingw32 ?)
WHEN version() ~ 'mingw32' OR version() ~ '64-bit|x86_64|ppc64|ia64|amd64'
THEN 8
ELSE 4
END AS maxalign,
/* per page header, fixed size: 20 for 7.X, 24 for others */
24 AS pagehdr,
/* per page btree opaque data */
16 AS pageopqdata,
/* per tuple header: add IndexAttributeBitMapData if some cols are null-able */
CASE WHEN max(coalesce(s.null_frac,0)) = 0
THEN 2 -- IndexTupleData size
ELSE 2 + (( 32 + 8 - 1 ) / 8)
-- IndexTupleData size + IndexAttributeBitMapData size ( max num filed per
index + 8 - 1 /8)
END AS index_tuple_hdr_bm,

```



```

        /* data len: we remove null values save space using it fractionnal part from
stats */
        sum( (1-coalesce(s.null_frac, 0)) * coalesce(s.avg_width, 1024)) AS
nulldatawidth,
        max( CASE WHEN a.atttypid = 'pg_catalog.name'::regtype THEN 1 ELSE 0 END ) > 0
AS is_na
FROM pg_attribute AS a
JOIN (
    SELECT nspname, tbl.relname AS tblname, idx.relname AS idxname,
        idx.reltuples, idx.relpages, idx.relam,
        indrelid, indexrelid, indkey::smallint[] AS attnum,
        coalesce(substring(
            array_to_string(idx.reloptions, ' ')
            from 'fillfactor=([0-9]+)'):smallint, 90) AS fillfactor
FROM pg_index
JOIN pg_class idx ON idx.oid=pg_index.indexrelid
JOIN pg_class tbl ON tbl.oid=pg_index.indrelid
JOIN pg_namespace ON pg_namespace.oid = idx.relnamespace
WHERE pg_index.indisvalid AND tbl.relkind = 'r' AND idx.relpages > 0
) AS i ON a.attrelid = i.indexrelid
JOIN pg_stats AS s ON s.schemaname = i.nspname
AND ((s.tablename = i.tblname AND s.attnum =
pg_catalog.pg_get_indexdef(a.attrelid, a.attnum, TRUE))
-- stats from tbl
OR (s.tablename = i.idxname AND s.attnum = a.attnum))
-- stats from functional cols
JOIN pg_type AS t ON a.atttypid = t.oid
WHERE a.attnum > 0
GROUP BY 1, 2, 3, 4, 5, 6, 7, 8, 9
) AS s1
) AS s2
JOIN pg_am am ON s2.relam = am.oid WHERE am.amname = 'btree'
) AS sub
-- WHERE NOT is_na
ORDER BY 2,3,4;

```

尋找適合自動資料清理的資料表

若要尋找適合自動資料清理的資料表，請執行下列查詢。

```

--This query shows tables that need vacuuming and are eligible candidates.
--The following query lists all tables that are due to be processed by autovacuum.
-- During normal operation, this query should return very little.
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold

```

```

        FROM pg_settings WHERE name = 'autovacuum_vacuum_threshold')
, vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor
        FROM pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
, fma AS (SELECT setting AS autovacuum_freeze_max_age
        FROM pg_settings WHERE name = 'autovacuum_freeze_max_age')
, sto AS (SELECT opt_oid, split_part(setting, '=', 1) as param,
        split_part(setting, '=', 2) as value
        FROM (SELECT oid opt_oid, unnest(reloptions) setting FROM pg_class) opt)
SELECT
    '""||ns.nspname||"."||c.relname||"' as relation
, pg_size_pretty(pg_table_size(c.oid)) as table_size
, age(relfrozenxid) as xid_age
, coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
autovacuum_freeze_max_age
, (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
    coalesce(cvsf.value::float, autovacuum_vacuum_scale_factor::float) *
c.reltuples)
    as autovacuum_vacuum_tuples
, n_dead_tup as dead_tuples
FROM pg_class c
JOIN pg_namespace ns ON ns.oid = c.relnamespace
JOIN pg_stat_all_tables stat ON stat.relid = c.oid
JOIN vbt on (1=1)
JOIN vsf ON (1=1)
JOIN fma on (1=1)
LEFT JOIN sto cvbt ON cvbt.param = 'autovacuum_vacuum_threshold' AND c.oid =
cvbt.opt_oid
LEFT JOIN sto cvsf ON cvsf.param = 'autovacuum_vacuum_scale_factor' AND c.oid =
cvsf.opt_oid
LEFT JOIN sto cfma ON cfma.param = 'autovacuum_freeze_max_age' AND c.oid = cfma.opt_oid
WHERE c.relkind = 'r'
AND nspname <> 'pg_catalog'
AND (
    age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
    coalesce(cvsf.value::float, autovacuum_vacuum_scale_factor::float) * c.reltuples
<= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(relfrozenxid) DESC;

```

因應大量連線

監視 Amazon 時 CloudWatch，您可能會發現指 DatabaseConnections 標峰值。此增加表示資料庫的連線數增加。建議採取下列作法：

- 限制應用程式可以對每個執行個體開啟的連線數。如果應用程式有內嵌連線集區功能，請設定合理的連線數目。請以執行個體中的 vCPU 可有效平行處理的數目為準。

如果應用程式不使用連線集區功能，請考慮使用 Amazon RDS Proxy 或替代方案。這種作法可讓應用程式對負載平衡器開啟多個連線。因此，平衡器就能對資料庫開啟較少的連線。由於平行執行的連線較少，資料庫執行個體在核心中就能減少切換環境。查詢應該會進行得更快，使得等待事件變少。如需詳細資訊，請參閱 [使用 Amazon RDS Proxy](#)。

- 盡可能利用 RDS for PostgreSQL 的僅供讀取複本。當您的應用程式執行唯讀操作時，請將這些請求傳送至僅供讀取複本。這項技巧可減少主要 (寫入器) 節點的 I/O 壓力。
- 考慮擴充資料庫執行個體的規模。容量較大的執行個體類別提供更多記憶體，讓 RDS for PostgreSQL 有較大的共用緩衝集區可保留分頁。越大也讓資料庫執行個體有越多 vCPU 來處理連線。當寫入操作產生 IO:DataFileRead 等待事件時，較多 vCPU 會特別有用。

IO:WALWrite

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待時間增加的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 第 10 版和更新版本都支援此等待事件資訊。

Context

資料庫中產生預寫日誌資料的資料庫中的活動會先填滿 WAL 緩衝區，然後以非同步方式寫入至磁碟。等待事件 IO:WALWrite 是在 SQL 工作階段正在等待 WAL 資料，以完成寫入至磁碟時產生的，以便它可以釋出交易的 COMMIT 呼叫。

等待時間增加的可能原因

如果此等待事件經常發生，您應該檢閱工作負載，以及工作負載執行的更新類型及其頻率。尤其，尋找下列類型的活動。

繁重的 DML 活動

變更資料庫資料表中的資料不會立即發生。插入至一個資料表可能需要等待另一個用戶端的相同資料表插入或更新。用於變更資料值的資料處理語言 (DML) 陳述式 (INSERT、UPDATE、DELETE、COMMIT、ROLLBACK TRANSACTION) 可能會產生爭用，因而造成預寫日誌檔等待緩衝區排清。此情況是在下列指出繁重 DML 活動的 Amazon RDS Performance Insights 指標中擷取的。

- `tup_inserted`
- `tup_updated`
- `tup_deleted`
- `xcat_rollback`
- `xact_commit`

如需這些指標的詳細資訊，請參閱 [Amazon RDS for PostgreSQL 的績效詳情計數器](#)。

頻繁的檢查點活動

頻繁的檢查點造就更大的 WAL 大小。在 RDS for PostgreSQL 中，整頁寫入一律「開啟」。整頁寫入有助於防範資料遺失。不過，當檢查點發生過於頻繁時，系統可能會遭受整體效能問題。在 DML 活動繁重的系統上尤其如此。在某些情況下，您可能會在 `postgresql.log` 發現錯誤訊息，說明「檢查點發生過於頻繁」。

建議您在調整檢查點時，仔細平衡效能與異常關機時需要復原的預期時間。

動作

建議採取下列動作來減少此等待事件的數目。

主題

- [減少遞交次數](#)
- [監控檢查點](#)

- [擴增 IO](#)
- [專用記錄檔磁碟區 \(DLV\)](#)

減少遞交次數

若要減少遞交次數，您可以將陳述式合併成交易區塊。使用 Amazon RDS Performance Insights 來檢查正在執行的查詢類型。您也可以將大型維護操作移至離峰時段。例如，建立索引或在非生產時段使用 `pg_repack` 操作。

監控檢查點

有兩個您可以監控的參數，以查看 RDS for PostgreSQL 資料庫執行個體寫入至 WAL 檔案以取得檢查點的頻率。

- `log_checkpoints` – 此參數預設為「開啟」。它會導致訊息傳送至每個檢查點的 PostgreSQL 日誌。這些日誌訊息包含寫入的緩衝區數目、寫入它們所花費的時間，以及針對指定檢查點新增，刪除或回收的 WAL 檔案數目。

如需此參數的詳細資訊，請參閱 PostgreSQL 文件中的[錯誤報告和記錄](#)。

- `checkpoint_warning` – 此參數設定檢查點頻率的閾值 (以秒為單位)，若超過此值，就會產生警告。根據預設，此參數不會在 RDS for PostgreSQL 中設定。您可以設定此參數的值，在 RDS for PostgreSQL 資料庫執行個體中的資料庫變更以 WAL 檔案未調整為可處理大小的速率寫入時收到警告。例如，假設您將此參數設為 30。如果您的 RDS for PostgreSQL 執行個體需要寫入變更的頻率超過每 30 秒一次，則「檢查點發生過於頻繁」的警告會傳送至 PostgreSQL 日誌。這可能指出您的 `max_wal_size` 值應該增加。

如需的詳細資訊，請參閱 PostgreSQL 文件中的[預寫日誌](#)。

擴增 IO

此類型的輸入/輸出 (IO) 等待事件可透過擴展每秒讀寫次數 (IOPS) 來修補，以提供更快的 IO。擴展 IO 優於擴展 CPU，因為擴展 CPU 可能會導致更多的 IO 爭用，因為增加的 CPU 可以處理更多的工作，從而使 IO 瓶頸更糟。一般而言，在執行擴展操作之前，建議您考慮調整工作負載。

專用記錄檔磁碟區 (DLV)

您可以使用 Amazon RDS 主控台、AWS CLI 或 Amazon RDS API，將專用日誌磁碟區 (DLV) 用於使用佈建 IOPS (PIOPS) 儲存體的資料庫執行個體。DLV 會將 PostgreSQL 資料庫交易記錄移至與包含資料庫表格的磁碟區不同的儲存磁碟區。如需更多詳細資訊，請參閱 [專用記錄檔磁碟區 \(DLV\)](#)。

Lock:advisory

Lock:advisory 事件表示 PostgreSQL 應用程式使用鎖定來協調多個工作階段的活動。

主題

- [相關的引擎版本](#)
- [Context](#)
- [原因](#)
- [動作](#)

相關的引擎版本

此等待事件資訊與 RDS for PostgreSQL 9.6 版及更新版本有關。

Context

PostgreSQL 諮詢鎖定是應用程式層級的合作式鎖定，由使用者的應用程式碼明確鎖定和解除鎖定。應用程式可以使用 PostgreSQL 諮詢鎖定來協調多個工作階段的活動。不同於物件層級或資料列層級的一般鎖定，應用程式完全控制鎖定的生命週期。如需詳細資訊，請參閱 PostgreSQL 文件中的[諮詢鎖定](#)。

諮詢鎖定可以在交易結束之前釋放，或由工作階段跨交易保留。但系統強制的隱含鎖定不是如此，例如 CREATE INDEX 陳述式在資料表上取得的存取獨佔鎖定。

如需用來取得 (鎖定) 和釋放 (解除鎖定) 諮詢鎖定的函數描述，請參閱 PostgreSQL 說明文件中的[諮詢鎖定函數](#)。

諮詢鎖定是在一般 PostgreSQL 鎖定系統上實作，在 pg_locks 系統檢視表中可見。

原因

明確使用此鎖定類型的應用程式可完全控制鎖定。如果查詢為每個資料列都取得諮詢鎖定，可能會造成鎖定激增或長期累積。

當查詢取得的鎖定比查詢傳回的資料列更多時，就會出現這些現象。應用程式最終必須釋放每個鎖定，但如果是在未傳回的資料列上取得鎖定，則應用程式無法找齊全部的鎖定。

下列範例來自 PostgreSQL 文件中的[諮詢鎖定](#)。

```
SELECT pg_advisory_lock(id) FROM foo WHERE id > 12345 LIMIT 100;
```

在此範例中，只有在內部選取資料列並鎖定其 ID 值之後，LIMIT 子句才能停止查詢的輸出。突然發生這種情況表示資料量不斷增加，導致規劃工具選擇另一個未經過開發期間測試的執行計劃。在此情況下，發生累積起因於應用程式對每個已鎖定的 ID 值，明確呼叫 `pg_advisory_unlock`。但是，在此情況下，找不到在未傳回的資料列上取得的鎖定集。因為是在工作階段層級取得鎖定，交易結束時不會自動釋放鎖定。

鎖定嘗試受阻次數激增的另一個可能原因是意外衝突。在這些衝突中，應用程式的無關聯部分不慎共用相同的鎖定 ID 空間。

動作

檢閱諮詢鎖定的應用程式用量，並詳述在應用程式流程中何處和何時取得和釋放每一種諮詢鎖定。

查明究竟是工作階段取得太多鎖定，還是長時間執行的工作階段未及早釋放鎖定，導致鎖定逐漸堆積。您可以使用 `pg_terminate_backend(pid)` 結束工作階段，以免工作階段層級鎖定逐漸堆積。

等待諮詢鎖定的用戶端以 `wait_event_type=Lock` 和 `wait_event=advisory` 出現在 `pg_stat_activity` 中。您可以在 `pg_locks` 系統檢視表中查詢相同的 `pid`，尋找 `locktype=advisory` 和 `granted=f`，以取得特定的鎖定值。

然後，您可以在 `pg_locks` 中查詢 `granted=t` 的同一個諮詢鎖定，以識別引起封鎖的工作階段，如下列範例所示。

```
SELECT blocked_locks.pid AS blocked_pid,  
       blocking_locks.pid AS blocking_pid,  
       blocked_activity.username AS blocked_user,  
       blocking_activity.username AS blocking_user,  
       now() - blocked_activity.xact_start AS blocked_transaction_duration,  
       now() - blocking_activity.xact_start AS blocking_transaction_duration,  
       concat(blocked_activity.wait_event_type, ':', blocked_activity.wait_event) AS  
blocked_wait_event,  
       concat(blocking_activity.wait_event_type, ':', blocking_activity.wait_event) AS  
blocking_wait_event,  
       blocked_activity.state AS blocked_state,  
       blocking_activity.state AS blocking_state,  
       blocked_locks.locktype AS blocked_locktype,  
       blocking_locks.locktype AS blocking_locktype,  
       blocked_activity.query AS blocked_statement,
```

```

        blocking_activity.query AS blocking_statement
FROM pg_catalog.pg_locks blocked_locks
JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
blocked_locks.pid
JOIN pg_catalog.pg_locks blocking_locks
    ON blocking_locks.locktype = blocked_locks.locktype
    AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
    AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
    AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
    AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
    AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
    AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
    AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
    AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
    AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
    AND blocking_locks.pid != blocked_locks.pid
JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
WHERE NOT blocked_locks.GRANTED;

```

所有諮詢鎖定 API 函數都有兩組引數，可能是一個 bigint 引數或兩個 integer 引數：

- 如果 API 函數有一個 bigint 引數，則前段 32 位元在 `pg_locks.classid` 中，後段 32 位元在 `pg_locks.objid` 中。
- 如果 API 函數有兩個 integer 引數，則第一個引數為 `pg_locks.classid`，第二個引數為 `pg_locks.objid`。

`pg_locks.objsubid` 值表示使用何種 API 形式：1 表示一個 bigint 引數；2 表示二個 integer 引數。

Lock:extend

Lock:extend 事件表示後端程序正在等待鎖定關聯來延伸，但另一個程序也基於相同目的而鎖定該關聯。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)

- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

Lock:extend 事件表示後端程序正在等待延伸關聯，但另一個後端程序已鎖定該關聯而正在延伸。因為一次只有一個程序可以延伸關聯，所以系統產生 Lock:extend 等待事件。INSERT、COPY 及 UPDATE 操作可能產生此事件。

等待變多的可能原因

Lock:extend 事件比平時更常出現時，可能表示有效能問題，典型原因包括：

突增並行插入或更新同一個資料表

以查詢來插入或更新同一個資料表的並行工作階段可能變多。

網路頻寬不足

資料庫執行個體上的網路頻寬可能不足以滿足目前工作負載的儲存通訊需求。這可能會引起儲存延遲，導致 Lock:extend 事件增加。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [減少並行插入和更新同一個關聯](#)
- [增加網路頻寬](#)

減少並行插入和更新同一個關聯

首先，判斷 tup_inserted 和 tup_updated 指標是否增加，以及此等待事件是否也隨之增加。如果是，請檢查哪些關聯激烈爭用插入和更新操作。若要查明，請在 pg_stat_all_tables 檢視表中查詢 n_tup_ins 和 n_tup_upd 欄位的值。如需 pg_stat_all_tables 檢視表的相關資訊，請參閱 PostgreSQL 文件中的 [pg_stat_all_tables](#)。

關於引起封鎖的查詢和被封鎖的查詢，如需詳細資訊，請查詢 `pg_stat_activity`，如下列範例所示：

```
SELECT
    blocked.pid,
    blocked.username,
    blocked.query,
    blocking.pid AS blocking_id,
    blocking.query AS blocking_query,
    blocking.wait_event AS blocking_wait_event,
    blocking.wait_event_type AS blocking_wait_event_type
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking ON blocking.pid = ANY(pg_blocking_pids(blocked.pid))
where
blocked.wait_event = 'extend'
and blocked.wait_event_type = 'Lock';
```

pid	username	query	blocking_id	blocking_query	blocking_wait_event	blocking_wait_event_type
7143	myuser	insert into tab1 values (1);	4600	INSERT INTO tab1 (a)	DataFileExtend	IO

在找出導致 `Lock:extend` 事件增加的關聯之後，請使用下列技巧來減少爭用：

- 查明您是否可以使用分割來減少爭用同一個資料表。將插入或更新的元組分成不同分割區可以減少爭用。如需分割的相關資訊，請參閱[使用 `pg_partman` 擴充功能來管理 PostgreSQL 分割區](#)。
- 如果等待事件主要是由於更新活動，請考慮降低關聯的 `fillfactor` 值。這樣可以減少在更新期間請求新區塊。`fillfactor` 是資料表的儲存參數，決定可供壓縮資料表分頁的最大空間。以分頁總空間的百分比表示。如需 `fillfactor` 參數的詳細資訊，請參閱 PostgreSQL 文件中的 [CREATE TABLE](#)。

Important

如果您變更 `fillfactor`，強烈建議您測試系統，因為變更此值可能對效能造成負面影響，視工作負載而定。

增加網路頻寬

若要檢查寫入延遲是否變長，請在 CloudWatch 中檢查 WriteLatency 指標。如果是，請使用 WriteThroughput 和 ReadThroughput Amazon CloudWatch 指標來監控資料庫執行個體上的儲存體相關流量。這些指標可協助您判斷網路頻寬是否足以應付工作負載的儲存活動。

如果網路頻寬不夠，請增加頻寬。如果資料庫執行個體快達到網路頻寬限制，則增加頻寬的唯一辦法就是加大資料庫執行個體。

如需 CloudWatch 指標的詳細資訊，請參閱 [Amazon RDS 的 CloudWatch Amazon 實例級指標](#)。關於每個資料庫執行個體類別的網路效能，如需相關資訊，請參閱 [Amazon RDS 的 CloudWatch Amazon 實例級指標](#)。

Lock:Relation

Lock:Relation 事件表示查詢正在等待取得鎖定的資料表或檢視表 (關聯)，目前由另一個交易鎖定。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

大多數 PostgreSQL 命令隱含地使用鎖定來控制並行存取資料表中的資料。您也可以從應用程式碼中使用 LOCK 命令，以明確使用這些鎖定。許多鎖定模式彼此不相容，在嘗試存取同一個物件時，可能封鎖交易。發生這種情況時，RDS for PostgreSQL 會產生 Lock:Relation 事件。以下是一些常見例子：

- 獨佔鎖定 (例如 ACCESS EXCLUSIVE) 會封鎖所有並行存取。資料定義語言 (DDL) 操作 (例如 DROP TABLE、TRUNCATE、VACUUM FULL 及 CLUSTER) 隱含地取得 ACCESS EXCLUSIVE 鎖定。對於未明確指定模式的 LOCK TABLE 陳述式，ACCESS EXCLUSIVE 也是預設鎖定模式。

- 在資料表上使用 CREATE INDEX (without CONCURRENT) 時，與取得 ROW EXCLUSIVE 鎖定的資料處理語言 (DML) 陳述式 UPDATE、DELETE 及 INSERT 會發生衝突。

如需表格層級鎖定和衝突鎖定模式的詳細資訊，請參閱 PostgreSQL 文件中的[明確鎖定](#)。

引起封鎖的查詢和交易通常透過下列其中一種方法解除封鎖：

- 引起封鎖的查詢 — 由應用程式取消查詢，或由使用者結束程序。透過工作階段的陳述式逾時或死鎖偵測機制，引擎也可以強制結束查詢。
- 引起封鎖的交易 — 交易執行 ROLLBACK 或 COMMIT 陳述式來停止封鎖。當工作階段由用戶端或因為網路問題而中斷連線時，或工作階段結束時，也會自動復原。資料庫引擎關閉、系統記憶體不足等原因會結束工作階段。

等待變多的可能原因

當 Lock:Relation 事件發生頻率高於正常情況時，則可能表示效能問題。典型原因包括：

資料表鎖定發生衝突的並行工作階段變多

以查詢來鎖定同一個資料表但鎖定模式衝突的並行工作階段可能變多。

維護操作

運作狀態維護操作 (例如 VACUUM 和 ANALYZE) 可能大幅增加衝突鎖定的數量。VACUUM FULL 取得 ACCESS EXCLUSIVE 鎖定，ANALYZE 取得 SHARE UPDATE EXCLUSIVE 鎖定。這兩種鎖定都可能引起 Lock:Relation 等待事件。應用程式資料維護操作 (例如重新整理具體化檢視表) 也會使鎖定的查詢和交易增加。

鎖定讀取器執行個體

寫入器和讀取器保有的關係鎖之間可能存在衝突。目前，只有 ACCESS EXCLUSIVE 關係鎖被複製到讀取器執行個體。但是，ACCESS EXCLUSIVE 關係鎖將與讀取器所持有的任何 ACCESS SHARE 關係鎖發生衝突。這可能會造成讀取器上的鎖定關係等待事件增加。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [減少封鎖 SQL 陳述式的影響](#)

- [將維護操作的影響降至最低](#)

減少封鎖 SQL 陳述式的影響

若要減少封鎖 SQL 陳述式的影響，請盡可能修改應用程式碼。以下是減少封鎖的兩種常用技巧：

- 使用 NOWAIT 選項 — 某些 SQL 命令支援此選項，例如 SELECT 和 LOCK 陳述式。如果無法立即獲得鎖定，NOWAIT 指令會取消提出鎖定請求的查詢。這項技巧有助於避免引起封鎖的工作階段背後堆積被封鎖的工作階段。

例如：假設交易 A 等待的鎖定由交易 B 持有。現在，如果 B 請求鎖定的資料表由交易 C 鎖定，則可能封鎖交易 A，直到交易 C 完成為止。但是，如果交易 B 請求鎖定 C 時使用 NOWAIT，則會很快失敗，以確保交易 A 不必無限期等待。

- 使用 SET lock_timeout - 設定 lock_timeout 值來限制 SQL 陳述式在關聯上取得鎖定所等待的時間。如果在指定的逾時內未獲得鎖定，則會取消提出鎖定請求的交易。請在工作階段層級設定此值。

將維護操作的影響降至最低

維護操作很重要，例如 VACUUM 和 ANALYZE。建議不要因為發現這些維護操作相關的 Lock:Relation 等待事件而關閉維護。下列方法可以將這些操作的影響降至最低：

- 在離峰時段手動執行維護操作。
- 若要減少 Lock:Relation 等待，請執行任何所需的自動資料清理調校。如需調整自動清理的相關資訊，請參閱《Amazon RDS 使用者指南》中的[在 Amazon RDS 上使用 PostgreSQL 自動資料清理](#)。

Lock:transactionid

Lock:transactionid 事件表示交易正在等待資料列層級鎖定。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

Lock:transactionid 事件表示交易嘗試取得的資料列層級鎖定已授予同一時間執行的另一個交易。因為此鎖定，已封鎖出現 Lock:transactionid 等待事件的工作階段。當引起封鎖的交易在 COMMIT 或 ROLLBACK 陳述式中結束時，被封鎖的交易就可以繼續進行。

RDS for PostgreSQL 的多版本並行控制語意保證讀取器不會封鎖寫入器，而寫入器也不會封鎖讀取器。引起封鎖和被封鎖的交易必須發出下列類型的衝突陳述式，才會發生資料列層級衝突：

- UPDATE
- SELECT ... FOR UPDATE
- SELECT ... FOR KEY SHARE

SELECT ... FOR KEY SHARE 陳述式是特殊情況。資料庫使用 FOR KEY SHARE 子句來最佳化參考完整性的效能。如果資料列上有資料列層級鎖定，則會封鎖其他資料表上參考此資料列的 INSERT、UPDATE 及 DELETE 命令。

等待變多的可能原因

此事件比平時更常出現時，通常是因為 UPDATE、SELECT ... FOR UPDATE 或 SELECT ... FOR KEY SHARE 陳述式兼具下列情況。

主題

- [高度並行](#)
- [交易閒置](#)
- [長時間執行的交易](#)

高度並行

RDS for PostgreSQL 可以使用精細的資料列層級鎖定語意。有下列情況時，較可能發生資料列層級衝突：

- 高度並行工作負載爭用相同的資料列。
- 並行增加。

交易閒置

有時候 `pg_stat_activity.state` 資料欄會顯示 `idle in transaction` 值。已啟動交易但尚未發出 `COMMIT` 或 `ROLLBACK` 的工作階段會出現此值。如果 `pg_stat_activity.state` 值不是 `active`，`pg_stat_activity` 中會顯示最近要完成執行的查詢。因為開啟的交易持有鎖定，引起封鎖的工作階段目前未處理查詢。

如果閒置交易已獲得資料列層級鎖定，可能會阻止其他工作階段取得鎖定。這種情況導致頻繁發生等待事件 `Lock:transactionid`。若要診斷問題，請檢查 `pg_stat_activity` 和 `pg_locks` 的輸出。

長時間執行的交易

長時間執行的交易會長時間持有鎖定。這些長期持有的鎖定可能阻止其他交易執行。

動作

資料列鎖定會造成 `UPDATE`、`SELECT ... FOR UPDATE` 或 `SELECT ... FOR KEY SHARE` 陳述式之間發生衝突。嘗試解決之前，請查明這些陳述式是否在同一個資料列上執行。使用此資訊來選擇以下各節描述的策略。

主題

- [因應高度並行](#)
- [因應閒置的交易](#)
- [因應長時間執行的交易](#)

因應高度並行

如果問題在於並行，請嘗試下列其中一項技巧：

- 降低應用程式中的並行。例如，減少作用中工作階段的數目。
- 實作連線集區。若要了解如何使用 `RDS Proxy` 來建立連線集區，請參閱 [使用 Amazon RDS Proxy](#)。
- 將應用程式或資料模型設計成避免爭用 `UPDATE` 和 `SELECT ... FOR UPDATE` 陳述式。您也可以減少 `SELECT ... FOR KEY SHARE` 陳述式存取的外部索引鍵數目。

因應閒置的交易

如果 `pg_stat_activity.state` 顯示 `idle in transaction`，請使用下列策略：

- 盡可能開啟自動遞交。這種方法可以防止交易在等待 `COMMIT` 或 `ROLLBACK` 時封鎖其他交易。

- 搜尋缺少 COMMIT、ROLLBACK 或 END 的程式碼路徑。
- 確保應用程式中的異常處理邏輯一定有路徑通往有效 end of transaction。
- 確保應用程式以 COMMIT 或 ROLLBACK 結束交易後處理查詢結果。

因應長時間執行的交易

如果長時間執行的交易導致頻繁出現 Lock:transactionid，請嘗試下列策略：

- 避免長時間執行的交易使用資料列鎖定。
- 盡可能實作自動遞交來限制查詢的長度。

Lock:tuple

Lock:tuple 事件表示後端程序正等待在元組上取得鎖定。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

Lock:tuple 事件表示後端正等待在元組上取得鎖定，但另一個後端在同一個元組上持有衝突鎖定。下表說明工作階段產生 Lock:tuple 事件的情節。

時間	工作階段 1	工作階段 2	工作階段 3
t1	開始交易。		
t2	更新資料列 1。		

時間	工作階段 1	工作階段 2	工作階段 3
t3		更新資料列 1。工作階段在元組上取得獨佔鎖定，然後等待工作階段 1 遞交或復原來釋放鎖定。	
t4			更新資料列 1。工作階段等待工作階段 2 釋放元組上的獨佔鎖定。

或者，您可以使用基準化分析工具 `pgbench` 來模擬此等待事件。使用自訂 SQL 檔案，將大量並行工作階段設定成在資料表中更新相同資料列。

若要進一步了解衝突鎖定模式，請參閱 PostgreSQL 文件中的[明確鎖定](#)。若要進一步了解 `pgbench`，請參閱 PostgreSQL 文件中的[pgbench](#)。

等待變多的可能原因

此事件比平時更常出現時，可能表示有效能問題，典型原因包括：

- 大量並行工作階段執行 UPDATE 或 DELETE 陳述式，嘗試取得同一個元組的衝突鎖定。
- 高度並行工作階段使用 FOR UPDATE 或 FOR NO KEY UPDATE 鎖定模式來執行 SELECT 陳述式。
- 各種因素迫使應用程式或連線集區開啟更多工作階段來執行相同的操作。由於新的工作階段嘗試修改相同的資料行，資料庫負載會激增，並出現 `Lock:tuple`。

如需詳細資訊，請參閱 PostgreSQL 文件中的[資料列層級鎖定](#)。

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [調查應用程式邏輯](#)
- [尋找引起封鎖的工作階段](#)
- [減少高度並行](#)

- [瓶頸疑難排解](#)

調查應用程式邏輯

查明引起封鎖的工作階段是否長時間處於 `idle in transaction` 狀態。如果是，請考慮結束引起封鎖的工作階段，當作短期的解決辦法。您也可以使用 `pg_terminate_backend` 函數。如需此函數的詳細資訊，請參閱 PostgreSQL 文件中的[伺服器訊號函數](#)。

如需長期解決方案，請執行下列動作：

- 調整應用程式邏輯。
- 使用 `idle_in_transaction_session_timeout` 參數。任何工作階段中，如果開啟的交易已閒置超過一段指定的時間，此參數會結束工作階段。如需詳細資訊，請參閱 PostgreSQL 文件中的[用戶端連線預設值](#)。
- 盡可能使用自動遞交。如需詳細資訊，請參閱 PostgreSQL 文件中的[SET AUTOCOMMIT](#)。

尋找引起封鎖的工作階段

`Lock:tuple` 等待事件發生時，請查明哪些鎖定彼此相依，以找出引起封鎖和被封鎖的工作階段。如需詳細資訊，請參閱 PostgreSQL Wiki 中的[鎖定相依性資訊](#)。

下列範例查詢所有工作階段，篩選 `tuple` 並依據 `wait_time` 排序。

```
SELECT blocked_locks.pid AS blocked_pid,
       blocking_locks.pid AS blocking_pid,
       blocked_activity.username AS blocked_user,
       blocking_activity.username AS blocking_user,
       now() - blocked_activity.xact_start AS blocked_transaction_duration,
       now() - blocking_activity.xact_start AS blocking_transaction_duration,
       concat(blocked_activity.wait_event_type, ':', blocked_activity.wait_event) AS
blocked_wait_event,
       concat(blocking_activity.wait_event_type, ':', blocking_activity.wait_event) AS
blocking_wait_event,
       blocked_activity.state AS blocked_state,
       blocking_activity.state AS blocking_state,
       blocked_locks.locktype AS blocked_locktype,
       blocking_locks.locktype AS blocking_locktype,
       blocked_activity.query AS blocked_statement,
       blocking_activity.query AS blocking_statement
FROM pg_catalog.pg_locks blocked_locks
```

```
JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
blocked_locks.pid
JOIN pg_catalog.pg_locks blocking_locks
  ON blocking_locks.locktype = blocked_locks.locktype
  AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
  AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
  AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
  AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
  AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
  AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
  AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
  AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
  AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
  AND blocking_locks.pid != blocked_locks.pid
JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
WHERE NOT blocked_locks.GRANTED;
```

減少高度並行

Lock:tuple 事件可能持續發生，尤其在忙碌的工作負載時段。在此情況下，對於非常忙碌的資料列，請考慮減少高度並行。通常只有少數資料列控制佇列或布林邏輯，使得這些資料列非常忙碌。

您可以根據商業需求、應用程式邏輯和工作負載類型，使用不同方法來減少並行。例如，您可以執行下列動作：

- 重新設計資料表和資料邏輯來減少高度並行。
- 變更應用程式邏輯來減少資料列層級的高度並行。
- 善用並重新設計含有資料列層級鎖定的查詢。
- 對重試操作使用 NOWAIT 子句。
- 考慮使用樂觀和混合鎖定邏輯並行控制。
- 考慮變更資料庫隔離層級。

瓶頸疑難排解

Lock:tuple 可能發生瓶頸，例如 CPU 不足或 Amazon EBS 頻寬耗盡。若要減少瓶頸，請考慮下列方法：

- 擴充執行個體類別類型的規模。

- 將消耗大量資源的查詢最佳化。
- 變更應用程式邏輯。
- 封存不常存取的資料。

LWLock:BufferMapping (LWLock:buffer_mapping)

此事件表示工作階段正在等待將資料區塊與共用緩衝集區中的緩衝區建立關聯。

Note

對於 RDS for PostgreSQL 第 13 版及更新版本，此事件命名為 `LWLock:BufferMapping`。
對於 RDS for PostgreSQL 第 12 版及更舊版本，此事件命名為 `LWLock:buffer_mapping`。

主題

- [支援的引擎版本](#)
- [Context](#)
- [原因](#)
- [動作](#)

支援的引擎版本

此等待事件資訊與 RDS for PostgreSQL 9.6 版及更新版本有關。

Context

共用緩衝集區是一種 PostgreSQL 記憶體區域，其中保留程序正在使用或已使用的所有分頁。程序需要分頁時會將分頁讀入共用緩衝集區。`shared_buffers` 參數設定共用緩衝區大小，並保留記憶體區域來存放資料表和索引分頁。如果您變更此參數，請務必重新啟動資料庫。

下列情況會發生 `LWLock:buffer_mapping` 等待事件：

- 程序在緩衝區資料表中搜尋分頁，並取得共用緩衝區映射鎖定。
- 程序將分頁載入緩衝集區，並取得獨佔緩衝區映射鎖定。
- 程序從集區移除頁面，並取得獨佔緩衝區映射鎖定。

原因

此事件比平時更常出現時，可能表示有效能問題，資料庫正在共用緩衝集區中頁進頁出。典型原因包括：

- 大型查詢
- 膨脹的索引和資料表
- 完整資料表掃描
- 小於工作集的共用集區大小

動作

根據等待事件的原因，我們會建議不同的動作。

主題

- [監控緩衝區相關指標](#)
- [評估索引策略](#)
- [減少必須快速配置的緩衝區數目](#)

監控緩衝區相關指標

LWLock:buffer_mapping 等待激增時，請調查緩衝區命中率。您可以使用這些指標，以更加了解緩衝區快取中的情況。檢查下列指標：

blks_hit

此績效詳情計數器指標指出從共用緩衝集區擷取的區塊數目。在 LWLock:buffer_mapping 等待事件出現之後，您可能發現 blks_hit 激增。

blks_read

此績效詳情計數器指標指出需要將輸入/輸出讀入共用緩衝集區的區塊數目。在 LWLock:buffer_mapping 等待事件發生之前，您可能發現 blks_read 激增。

評估索引策略

若要確認索引策略不會降低效能，請檢查下列各項：

索引膨脹

請確定索引和資料表膨脹不會導致將不必要的分頁讀入共用緩衝區。如果資料表包含未使用的資料列，請考慮封存資料並從資料表中移除資料列。然後，您可以對已調整大小的資料表重建索引。

常用查詢的索引

若要判斷您是否有最佳索引，請在績效詳情中監控資料庫引擎指標。tup_returned 指標顯示讀取的資料列數。tup_fetched 指標顯示傳回給用戶端的資料列數。如果 tup_returned 明顯大於 tup_fetched，表示資料可能未正確編製索引。此外，資料表統計數字可能不是最新。

減少必須快速配置的緩衝區數目

若要減少 LWLock:buffer_mapping 等待事件，請嘗試減少必須快速配置的緩衝區數目。有一種策略是執行較小的批次操作。您可以分割資料表來獲得較小的批次。

LWLock:BufferIO (IPC:BufferIO)

LWLock:BufferIO 事件表示 RDS for PostgreSQL 與其他程序同時嘗試存取分頁，正在等待其他程序完成輸入/輸出 (I/O) 操作。目的是為了將該分頁讀入共用緩衝區。

主題

- [相關的引擎版本](#)
- [Context](#)
- [原因](#)
- [動作](#)

相關的引擎版本

此等待事件資訊與所有 RDS for PostgreSQL 版本有關。對於適用於 PostgreSQL 12 的 RDS 和更早版本，此等待事件被命名為 lwlock:buffer_io，而在 RDS 中為 PostgreSQL 13 版本，它被命名為 lwlock:bufferio。從 RDS for PostgreSQL 14 版開始，此等待事件從 LWLock 移動至 IPC 等待事件類型 (IPC:BufferIO)。

Context

每次必須在共用緩衝集區外擷取區塊 (或分頁) 時，每個共用緩衝區都有與 LWLock:BufferIO 等待事件相關聯的輸入/輸出鎖定。

此鎖定用於處理全都需要存取同一個區塊的多個工作階段。必須從 `shared_buffers` 參數定義的共用緩衝集區外讀取此區塊。

在共用緩衝集區內讀取分頁後，就會立刻釋放 `LWLock:BufferIO` 鎖定。

Note

`LWLock:BufferIO` 等待事件在 [IO : DataFileRead](#) 等待事件之前發生。從儲存讀取資料時會發生 `IO:DataFileRead` 等待事件。

如需輕量級鎖定的詳細資訊，請參閱[鎖定概觀](#)。

原因

`LWLock:BufferIO` 事件出現在最常等待名單中的常見原因包括：

- 多個後端或連線嘗試存取同一個分頁，而此分頁也擱置輸入/輸出操作
- 共用緩衝集區 (由 `shared_buffers` 參數定義) 的大小與目前工作負載所需緩衝區數目之間的比率
- 共用緩衝集區的大小與其他操作移出的分頁數目不太相稱
- 大型或膨脹的索引迫使引擎將過多分頁讀入共用緩衝集區
- 缺少索引迫使資料庫引擎從資料表讀取過多分頁
- 檢查點太頻繁出現或需要清除太多已修改的分頁
- 嘗試對同一分頁執行操作的資料庫連線突然激增

動作

我們根據等待事件的原因，建議不同的動作：

- 觀察 Amazon CloudWatch 指標，以瞭解 `BufferCacheHitRatio` 和 `LWLock:BufferIO` 等待事件遽降之間的關聯。這可能表示共用緩衝區設定太小。您可能需要提高此設定，或擴充資料庫執行個體類別的規模。您可以將工作負載分割成更多讀取器節點。
- 如果您發現與 `BufferCacheHitRatio` 指標一致的 `LWLock:BufferIO` 下降，請根據工作負載尖峰時段來調校 `max_wal_size` 和 `checkpoint_timeout`。然後查明哪個查詢造成此狀況。
- 驗證是否有未使用的索引，然後移除。
- 使用分割的資料表 (也有分割的索引)。這樣做有助於盡量避免索引重新排序，並減少其影響。
- 避免不必要地編製資料欄的索引。

- 使用連線集區來防止資料庫連線突然激增。
- 在最佳實務上限制資料庫的連線數目上限。

LWLock:buffer_content (BufferContent)

LWLock:buffer_content 事件表示工作階段正等待在記憶體中讀取或寫入資料分頁，但另一個工作階段已鎖定該分頁來寫入。在 RDS for PostgreSQL 13 及更新版本中，此等待事件稱為 BufferContent。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

為了讀取或操作資料，PostgreSQL 透過共用記憶體緩衝區來存取資料。為了讀取緩衝區，程序以共用模式在緩衝區內容上取得輕量級鎖定 (LWLock)。為了寫入緩衝區，程序以獨佔模式取得該鎖定。共用鎖定允許其他程序同時在該內容上取得共用鎖定。獨佔鎖定阻止其他程序在該內容上獲得任何類型的鎖定。

LWLock:buffer_content (BufferContent) 事件表示多個程序正嘗試在特定緩衝區的內容上取得鎖定。

等待變多的可能原因

LWLock:buffer_content (BufferContent) 事件比平時更常出現時，可能表示有效能問題，典型原因包括：

更常並行更新相同資料

以查詢來更新相同緩衝區內容的並行工作階段可能變多。在有大量索引的資料表上，這種爭用可能更明顯。

工作負載資料不在記憶體中

當作用中工作負載處理的資料不在記憶體中時，這些等待事件可能增加。這是因為程序在執行磁碟輸入/輸出操作時持有鎖定更久。

過度使用外部索引鍵限制

外部索引鍵限制會延長程序持有緩衝區內容鎖定的時間。這是因為讀取操作在更新參考的索引鍵時，在該索引鍵上需要共用緩衝區內容鎖定。

動作

根據等待事件的原因，我們會建議不同的動作。您可以使用 Amazon RDS 績效詳情或查詢 `LWLock:buffer_content` 檢視表來識別 BufferContent (pg_stat_activity) 事件。

主題

- [改善記憶體內效率](#)
- [減少使用外部索引鍵限制](#)
- [移除未使用的索引](#)
- [使用序列時增加快取大小](#)

改善記憶體內效率

若要讓作用中工作負載資料更有機會留在記憶體中，請分割資料表或擴充執行個體類別的規模。如需資料庫執行個體類別的相關資訊，請參閱 [資料庫執行個體類別](#)。

減少使用外部索引鍵限制

調查遇到大量 `LWLock:buffer_content` (BufferContent) 等待事件的工作負載如何使用外部索引鍵限制。刪除不必要的外部索引鍵限制。

移除未使用的索引

對於遇到大量 `LWLock:buffer_content` (BufferContent) 等待事件的工作負載，請識別未使用的索引並移除。

使用序列時增加快取大小

如果您的資料表使用序列，請增加快取大小，以移除序列頁面和索引頁面上的爭用。每個序列都是共用記憶體中的單一頁面。預先定義的快取根據每個連線。當許多並行工作階段取得序列值時，此快取可能不足以處理工作負載。

LWLock:lock_manager (LWLock:lockmanager)

此事件表示因為無法執行快速路徑鎖定，RDS for PostgreSQL 引擎維護共用鎖定的記憶體區域來配置、檢查和解除配置鎖定。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

此等待事件資訊與 RDS for PostgreSQL 9.6 版及更新版本有關。對於比第 13 版舊的 RDS for PostgreSQL 版本，此等待事件的名稱為 LWLock:lock_manager。對於 RDS for PostgreSQL 第 13 版及更新版本，此等待事件的名稱為 LWLock:lockmanager。

Context

當您發出 SQL 陳述式時，RDS for PostgreSQL 會記錄鎖定，在並行操作期間保護資料庫的結構、資料和完整性。引擎可以使用快速路徑鎖定或不快速的路徑鎖定來達成此目標。不快速的路徑鎖定比快速路徑鎖定花更多成本，還會產生更多額外負荷。

快速路徑鎖定

如果鎖定經常取得和釋放但很少衝突，為了減少額外負荷，後端程序可以使用快速路徑鎖定。資料庫使用此機制來處理符合下列條件的鎖定：

- 使用 DEFAULT 鎖定方法。
- 代表鎖定資料庫關聯，而不是共用關聯。
- 弱鎖定，不太可能衝突。
- 引擎可以快速確認不可能有衝突鎖定。

有下列任一情況時，引擎無法使用快速路徑鎖定：

- 鎖定不符合上述條件。
- 已無插槽可供後端程序使用。

若要調整您的查詢以進行快速路徑鎖定，您可以使用下列查詢。

```
SELECT count(*), pid, mode, fastpath
   FROM pg_locks
  WHERE fastpath IS NOT NULL
  GROUP BY 4,3,2
  ORDER BY pid, mode;
count | pid |      mode      | fastpath
-----+-----+-----+-----
  16 | 9185 | AccessShareLock | t
 336 | 9185 | AccessShareLock | f
   1 | 9185 | ExclusiveLock   | t
```

下列查詢只顯示整個資料庫的總計。

```
SELECT count(*), mode, fastpath
   FROM pg_locks
  WHERE fastpath IS NOT NULL
  GROUP BY 3,2
  ORDER BY mode,1;
count |      mode      | fastpath
-----+-----+-----
  16 | AccessShareLock | t
 337 | AccessShareLock | f
   1 | ExclusiveLock   | t
(3 rows)
```

如需快速路徑鎖定的詳細資訊，請參閱 PostgreSQL 鎖定管理員 README 中的[快速路徑](#)和 PostgreSQL 文件中的[pg-locks](#)。

鎖定管理員的擴展問題範例

在此範例中，名為 purchases 的資料表存放五年的資料，並按日分割。每個分割區都有兩個索引。發生下列事件序列：

1. 您查詢很多天的資料，使得資料庫需要讀取許多分割區。
2. 資料庫為每個分割區建立鎖定項目。如果分割區索引出現在最佳化工具存取路徑中，則資料庫也為這種索引建立鎖定項目。
3. 對同一個後端程序所請求的鎖定項目數大於 16 時，即 FP_LOCK_SLOTS_PER_BACKEND 的值，鎖定管理員會使用不快速的路徑鎖定方法。

現代化應用程式可能有數百個工作階段。如果並行工作階段查詢父項，但沒有適當的分割區剔除，則資料庫可能會建立數百甚至數千個不快速的路徑鎖定。當此並行高於 vCPU 數目時，通常會出現 `LWLock:lock_manager` 等待事件。

Note

`LWLock:lock_manager` 等待事件與資料庫結構描述中的分割區或索引數目無關。但與資料庫必須控制的不快速路徑鎖定數目有關。

等待變多的可能原因

`LWLock:lock_manager` 等待事件比平常更常發生時，可能表示有效能問題，突然激增最可能的原因如下：

- 並行作用中工作階段正在執行的查詢未使用快速路徑鎖定。這些工作階段也超過 vCPU 上限。
- 大量並行作用中工作階段正在存取高度分割的資料表。每個分割區有多個索引。
- 資料庫遇到連線風暴。根據預設，當資料庫變慢時，某些應用程式和連線集區軟體會建立更多連線。這種做法使問題變得更糟。請調校連線集區軟體，以免發生連線風暴。
- 大量工作階段查詢父資料表但未剔除分割區。
- 資料定義語言 (DDL)、資料處理語言 (DML) 或維護命令獨佔鎖定忙碌關聯，或是經常存取或修改的元組。

動作

發生 CPU 等待事件不見得表示有效能問題。只有在效能下降且此等待事件主宰資料庫負載時，才需要因應此事件。

主題

- [使用分割區剔除](#)
- [移除不必要的索引](#)
- [調校查詢以使用快速路徑鎖定](#)
- [調校其他等待事件](#)
- [降低硬體瓶頸](#)
- [使用連線集區](#)

- [升級 RDS for PostgreSQL 版本](#)

使用分割區剔除

剔除分割區是一種宣告式分割資料表的查詢最佳化策略，可將不需要的分割區排除在資料表掃描外，進而改善效能。分割區剔除預設為啟用。如果已停用，請如下啟用。

```
SET enable_partition_pruning = on;
```

當 WHERE 子句包含用於分割的資料欄時，查詢可以利用分割區剔除。如需詳細資訊，請參閱 PostgreSQL 文件中的[分割區剔除](#)。

移除不必要的索引

資料庫可能包含未使用或很少使用的索引。若是如此，請考慮刪除這些索引。執行下列任何一項：

- 請參閱 PostgreSQL Wiki 中的[未使用的索引](#)，以了解如何尋找不必要的索引。
- 執行 PG Collector。此 SQL 指令碼會收集資料庫資訊，並顯示在合併的 HTML 報告中。請檢查「Unused indexes (未使用的索引)」區段。如需詳細資訊，請參閱 AWS Labs GitHub 儲存庫中的[pg-collector](#)。

調校查詢以使用快速路徑鎖定

若要查明查詢是否使用快速路徑鎖定，請查詢 pg_locks 資料表的 fastpath 資料欄。如果查詢未使用快速路徑鎖定，請嘗試將每個查詢的關聯數量減少到 16 以下。

調校其他等待事件

如果 LWLock:lock_manager 在最常等待名單中排行前兩名，請檢查下列等待事件是否也出現在名單中：

- Lock:Relation
- Lock:transactionid
- Lock:tuple

如果上述事件在名單中排行前幾名，請考慮先調校這些等待事件。這些事件可能引發 LWLock:lock_manager。

降低硬體瓶頸

您可能遇到硬體瓶頸，例如 CPU 不足或 Amazon EBS 頻寬耗盡。在這些情況下，請考慮降低硬體瓶頸。考慮下列動作：

- 擴充執行個體類別的規模。
- 將耗用大量 CPU 和記憶體查詢最佳化。
- 變更應用程式邏輯。
- 封存資料。

如需 CPU、記憶體和 EBS 網路頻寬的詳細資訊，請參閱 [Amazon RDS 執行個體類型](#)。

使用連線集區

如果作用中連線總數超過 vCPU 上限，表示需要 CPU 的作業系統程序超過執行個體類型可支援的數量。在此情況下，請考慮使用或調校連線集區。關於執行個體類型的 vCPU，如需詳細資訊，請參閱 [Amazon RDS 執行個體類型](#)。

如需連線集區的詳細資訊，請參閱下列資源：

- [使用 Amazon RDS Proxy](#)
- [pgbouncer](#)
- 《PostgreSQL 文件》中的 [連線集區和資料來源](#)

升級 RDS for PostgreSQL 版本

如果您目前的 RDS for PostgreSQL 版本低於 12，請升級至第 12 版或更新版本。PostgreSQL 第 12 版和更新版本已改善分割機制。如需第 12 版的詳細資訊，請參閱 [PostgreSQL 12.0 版本備註](#)。如需升級 RDS for PostgreSQL 的詳細資訊，請參閱 [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。

Timeout:PgSleep

Timeout:PgSleep 事件表示伺服器程序已呼叫 pg_sleep 函數，正在等待睡眠逾時到期。

主題

- [支援的引擎版本](#)
- [等待變多的可能原因](#)

- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

等待變多的可能原因

此等待事件表示應用程式、預存函數或使用者發出的 SQL 陳述式呼叫下列其中一個函數：

- `pg_sleep`
- `pg_sleep_for`
- `pg_sleep_until`

上述函數會延遲執行，直到經過指定的秒數。例如，`SELECT pg_sleep(1)` 會暫停 1 秒。如需詳細資訊，請參閱 PostgreSQL 文件中的[延遲執行](#)。

動作

找出執行 `pg_sleep` 函數的陳述式。判斷使用此函數是否適當。

Timeout:VacuumDelay

`Timeout:VacuumDelay` 事件表示已超過清空 I/O 的成本限制，並且清空程序已進入休眠狀態。清空操作會停止一段時間，長度為各自成本延遲參數中指定的持續時間，然後恢復其運作。對於手動清空命令，延遲是在 `vacuum_cost_delay` 參數中指定。對於自動清空常駐程式，延遲是在 `autovacuum_vacuum_cost_delay parameter` 中指定。

主題

- [支援的引擎版本](#)
- [Context](#)
- [等待變多的可能原因](#)
- [動作](#)

支援的引擎版本

所有 RDS for PostgreSQL 版本都支援此等待事件資訊。

Context

PostgreSQL 同時具有自動清空常駐程式和手動清空命令。對於 RDS for PostgreSQL 資料庫執行個體，自動清空程序預設為「開啟」。手動清空命令是根據需要而使用，例如，用來清除失效元組的資料表或產生新的統計資料。

進行清空時，PostgreSQL 會在系統執行各種 I/O 操作時，使用內部計數器來追蹤預估的成本。當計數器達到成本限制參數所指定的值時，執行操作的程序休眠一段時間，長度為成本延遲參數中指定的短暫持續時間。然後，它會重設計數器並繼續操作。

清空程序具有可以用來調節資源耗用的參數。自動清空和手動清空命令具有自己的參數，用於設定成本限制值。它們還有自己的參數來指定成本延遲，這是達到限制時將清空置於休眠狀態的時間量。如此一來，成本延遲參數會當作資源耗用的限流機制運作。在下列清單中，您可以找到這些參數的描述。

影響自動清空常駐程式限流的參數

- [autovacuum_vacuum_cost_limit](#) – 指定用於自動清空操作的成本限制值。增加此參數的設定可讓清空程序使用更多資源並減少 Timeout:VacuumDelay 等待事件。
- [autovacuum_vacuum_cost_delay](#) – 指定用於自動清空操作的成本延遲值。預設值為 2 毫秒。將延遲參數設為 0 會關閉限流機制，因此 Timeout:VacuumDelay 等待事件不會出現。

如需詳細資訊，請參閱 PostgreSQL 文件中的[自動資料清理](#)。

影響手動清空程序限流的參數

- `vacuum_cost_limit` – 清空程序進入休眠狀態的閾值。根據預設，限制為 200。此數字代表各種資源所需額外 I/O 的累計成本預估值。增加此值可減少 Timeout:VacuumDelay 等待事件的數目。
- `vacuum_cost_delay` – 達到清空成本限制時，清空程序休眠的時間量。預設設定為 0，表示此功能已關閉。您可以將此值設為整數值，以指定開啟此功能的毫秒數，但建議您將其保留為預設設定。

如需 `vacuum_cost_delay` 參數的詳細資訊，請參閱 PostgreSQL 文件中的[資源耗用](#)。

若要進一步了解如何設定和使用自動清空搭配 RDS for PostgreSQL，請參閱 [在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理](#)。

等待變多的可能原因

Timeout:VacuumDelay 會受到成本限制參數設定

(`vacuum_cost_limit`、`autovacuum_vacuum_cost_limit`) 與控制清空休眠持續時間的成本延遲

參數 (`vacuum_cost_delay`、`autovacuum_vacuum_cost_delay`) 之間的平衡影響。提高成本限制參數值允許清空使用更多資源，然後再進入休眠狀態。這會導致更少的 `Timeout:VacuumDelay` 等待事件。增加其中一個延遲參數會導致 `Timeout:VacuumDelay` 等待事件更頻繁地發生，時段更長。

`autovacuum_max_workers` 參數設定也可以增加 `Timeout:VacuumDelay` 的數目。每個額外的自動清空工作者程序都有助於內部計數器機制，因此可以比使用單一自動清空工作者程序更快地達到限制。由於達到成本限制的速度更快，成本延遲會更頻繁地生效，造成更多的 `Timeout:VacuumDelay` 等待事件。如需詳細資訊，請參閱 PostgreSQL 文件中的 [autovacuum_max_worker](#)。

大型物件 (例如 500 GB 或更大) 也會引發此等待事件，因為清空可能需要一些時間才能完成處理大型物件。

動作

如果清空操作如預期完成，則不需要修補。換言之，此等待事件不一定指出問題。其表示清空將進入休眠一段時間，長度為延遲參數中指定的時段，以便資源可以套用到需要完成的其他程序。

如果您想要清空操作更快地完成，則可以降低延遲參數。這可以縮短清空休眠的時間。

使用 Amazon DevOps Guru 主動洞察，調校 RDS for PostgreSQL

DevOps Guru 主動洞察會偵測可能在 RDS for PostgreSQL 資料庫執行個體 上造成問題的狀況，並在發生問題前即讓您了解狀況。DevOps Guru 可以執行下列動作：

- 透過交叉檢查一般建議設定與您的資料庫設定，避免許多常見的資料庫問題。
- 警告您機群內的重大的問題，若未勾選，可能導致更嚴重的問題。
- 提醒您新發現的問題。

每個主動洞察都包含問題原因分析和修正動作建議。

主題

- [資料庫在交易連線中長時間閒置](#)

資料庫在交易連線中長時間閒置

資料庫的連線已經超過 1800 秒都處在 idle in transaction 狀態。

主題

- [支援的引擎版本](#)
- [Context](#)
- [造成此問題的可能原因](#)
- [動作](#)
- [相關指標](#)

支援的引擎版本

所有版本的 RDS for PostgreSQL 皆支援此洞察資訊。

Context

idle in transaction 狀態的交易可以擁有封鎖其他查詢的鎖定。也可以防止 VACUUM (包含自動清空) 清理無效資料列，導致索引或資料表膨脹，或導致交易 ID 包圍。

造成此問題的可能原因

尚未使用 COMMIT、ROLLBACK 或 END 命令，關閉以 BEGIN 或 START TRANSACTION 在互動式工作階段中啟動的交易。這會導致交易移至 idle in transaction 狀態。

動作

您可以透過查詢 pg_stat_activity，找出閒置的交易。

請在您的 SQL 用戶端中執行下列查詢，以列出 idle in transaction 狀態的所有連線，並按持續時間排序：

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
  xact_duration,*
FROM pg_stat_activity
WHERE state = 'idle in transaction'
AND xact_start is not null
ORDER BY 1 DESC;
```

根據洞察的原因，我們會建議不同的動作。

主題

- [End 交易](#)
- [終止連線](#)
- [設定 idle_in_transaction_session_timeout 參數](#)
- [檢查 AUTOCOMMIT 狀態](#)
- [檢查應用程式程式碼中的交易邏輯](#)

End 交易

使用 BEGIN 或 START TRANSACTION 在互動式工作階段中啟動交易時，該筆交易會移至 idle in transaction 狀態。交易會保持在此狀態，直到您發出 COMMIT、ROLLBACK、END 命令結束交易，或完全斷開連線以轉返結束交易。

終止連線

使用以下查詢，終止與閒置交易的連線：

```
SELECT pg_terminate_backend(pid);
```

pid 是連線的程序 ID。

設定 `idle_in_transaction_session_timeout` 參數

在新的參數群組中設定 `idle_in_transaction_session_timeout` 參數。設定此參數的優點在於，不需要手動介入即可終止長時間閒置的交易。如需此參數的詳細資訊，請參閱 [PostgreSQL 文件](#)。

當交易處於 `idle_in_transaction` 狀態超過指定的時間時，在連線終止後便會在 PostgreSQL 日誌中報告下列訊息。

```
FATAL: terminating connection due to idle in transaction timeout
```

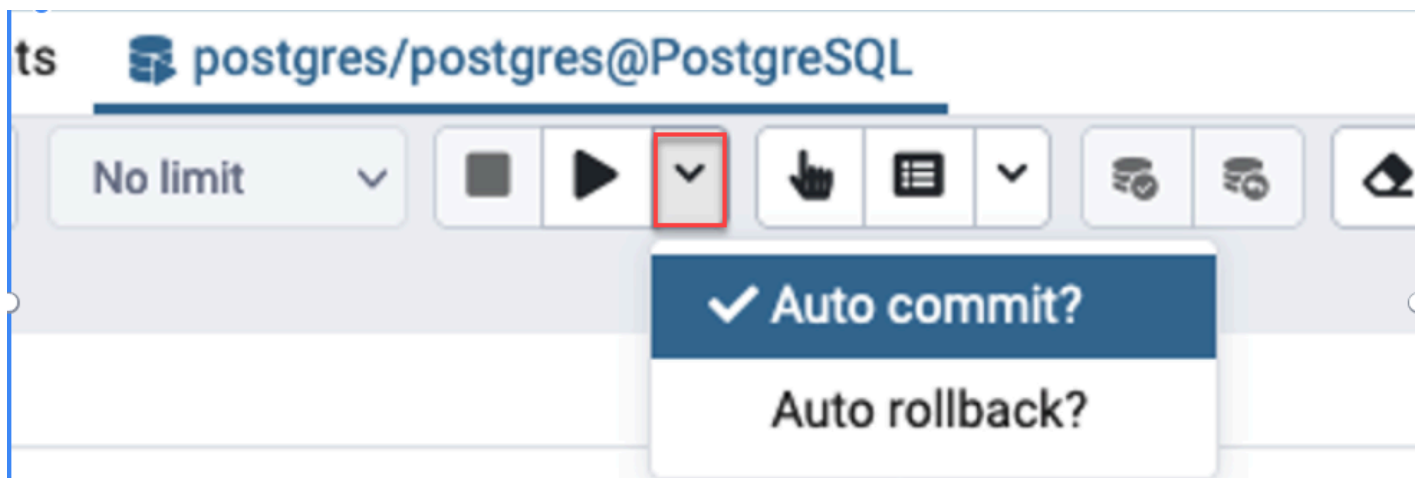
檢查 AUTOCOMMIT 狀態

根據預設，AUTOCOMMIT 為啟用狀態。但是，若客戶端意外將其關閉，請確認重啟。

- 在 psql 用戶端執行下列命令：

```
postgres=> \set AUTOCOMMIT on
```

- 在 pgadmin 中，從向下箭頭選擇 AUTOCOMMIT 選項以將其開啟。



檢查應用程式程式碼中的交易邏輯

調查應用程式邏輯，找出可能的問題。考慮下列動作：

- 檢查 JDBC 自動遞交是否在您的應用程式中設為 true。另外，請考慮在程式碼中使用明確的 COMMIT 命令。
- 檢查錯誤處理邏輯，確認其是否會在發生錯誤後關閉交易。
- 檢查交易開啟時，您的應用程式是否需要長時間處理查詢傳回的資料列。若是如此，請考慮對應用程式進行編碼，在處理資料列之前關閉交易。
- 檢查交易是否包含許多長時間執行的操作。若是如此，請將單一交易分割為多筆交易。

相關指標

下列 PI 指標與此洞察相關：

- `idle_in_transaction_count` - 處於 `idle in transaction` 狀態的工作階段數量。
- `idle_in_transaction_max_time` - 處於 `idle in transaction` 狀態的最長時間執行交易。

搭配 Amazon RDS for PostgreSQL 使用 PostgreSQL 擴充功能

您可安裝各種擴充功能和模組來擴展 PostgreSQL 的功能。例如，若要使用空間資料，您可以安裝和使用 PostGIS 擴充功能。如需詳細資訊，請參閱 [使用 PostGIS 擴充功能管理空間資料](#)。另一個例子是，如果要改進極大型資料表的資料輸入，則可以考慮使用 pg_partman 擴充功能對資料進行分區。如需進一步了解，請參閱 [使用 pg_partman 擴充功能來管理 PostgreSQL 分割區](#)。

Note

從 RDS for PostgreSQL 14.5 開始，RDS for PostgreSQL 支援適用於 PostgreSQL 的受信任語言延伸模組。此功能會實作為延伸模組 pg_tle，您可以將其新增至 RDS for PostgreSQL 資料庫執行個體。透過使用此延伸模組，開發人員可以在安全環境中建立自己的 PostgreSQL 延伸模組，這會簡化設定和組態需求。如需詳細資訊，請參閱 [使用適用於 PostgreSQL 的受信任語言延伸模組](#)。

在某些情況下，您可以將特定模組新增至 RDS for PostgreSQL 資料庫叢集的自訂資料庫叢集參數群組中的 shared_preload_libraries 清單，而不是安裝延伸模組。一般而言，預設資料庫叢集參數群組只會載入 pg_stat_statements，但有數個其他模組可供新增至清單。例如，您可以新增 pg_cron 模組來新增排程功能，如 [使用 PostgreSQL pg_cron 擴充功能排程維護](#) 中所詳述。另一個範例是，您可以載入 auto_explain 模組來記錄查詢執行計劃。若要深入了解，請參閱 AWS 知識中心中的 [記錄查詢執行計畫](#)。

根據 RDS for PostgreSQL 版本，安裝擴充功能可能需要 rds_superuser 許可權限，如下：

- 若為 RDS for PostgreSQL 第 12 版和更早版本，安裝擴充功能需要 rds_superuser 權限。
- 若為 RDS for PostgreSQL 第 13 版和更新版本，在給定資料庫執行個體上具建立許可權限的使用者 (角色) 可以安裝並使用任何信任擴充功能。如需信任擴充功能的清單，請參閱 [PostgreSQL 可信任延伸](#)。

您還可在 rds.allowed_extensions 參數中列出擴充功能，精確指定可在 RDS for PostgreSQL 資料庫執行個體上安裝的擴充功能。如需詳細資訊，請參閱 [限制安裝 PostgreSQL 擴充功能](#)。

若要進一步了解 rds_superuser 角色，請參閱 [了解 PostgreSQL 角色和許可](#)。

主題

- [使用 Orafce 擴充功能中的函數](#)

- [使用 pg_partman 擴充功能來管理 PostgreSQL 分割區](#)
- [使用 PgAudit 記錄資料庫活動](#)
- [使用 PostgreSQL pg_cron 擴充功能排程維護](#)
- [使用 pglogical 跨執行個體同步資料](#)
- [使用 pgactive 來支援主動-主動式複寫](#)
- [使用 pg_repack 擴充功能減少資料表和索引膨脹](#)
- [升級和使用 PLV8 擴充功能](#)
- [使用 PL/Rust 以 Rust 語言撰寫 PostgreSQL 函數](#)
- [使用 PostGIS 擴充功能管理空間資料](#)

使用 Orafce 擴充功能中的函數

Orafce 擴充功能提供的函數和運算子，可以模擬來自 Oracle 資料庫的函數和軟體套件的子集合。Orafce 擴充功能可讓您更輕易地將 Oracle 應用程式移植到 PostgreSQL。RDS for PostgreSQL 9.6.6 版及更高版本支援此擴充功能。如需有關 orafce 的更多資訊，請參閱上的[章節](#)。GitHub

Note

RDS for PostgreSQL 不支援 utl_file 套件，該套件屬於 Orafce 擴充功能的一部分。這是因為 utl_file 結構描述函式雖能夠在作業系統的文字檔上進行讀取及寫入操作，但必須擁有基礎主機的超級使用者存取權，才能執行此類操作。RDS for PostgreSQL 為受管服務，不提供主機存取權。

使用 Orafce 擴充功能

1. 使用您用來建立資料庫執行個體的主要使用者名稱，來連線至資料庫執行個體。

如果您想要在相同的資料庫執行個體中為不同的資料庫開啟 Orafce，請使用 `/c dbname psql` 命令。使用此命令，您可以在啟動連線後從主要資料庫進行變更。

2. 使用 `CREATE EXTENSION` 陳述式開啟 Orafce 擴充功能。

```
CREATE EXTENSION orafce;
```

3. 使用 `ALTER SCHEMA` 陳述式將 oracle 結構描述的擁有權移轉至 `rds_superuser` 角色。

```
ALTER SCHEMA oracle OWNER TO rds_superuser;
```

如果您想要查看 oracle 結構描述的擁有者清單，請使用 `\dn psql` 命令。

使用 pg_partman 擴充功能來管理 PostgreSQL 分割區

PostgreSQL 表格分割區提供了用於高性能處理資料輸入和報告的框架。對需要非常快速輸入大量資料的資料庫使用分割區。分割區還提供了更快的大型表格查詢。分割區有助於維護資料，而不會影響資料庫執行個體，因為它需要較少的輸入/輸出資源。

透過使用分割區，您可以將資料分割為自訂大小的區塊進行處理。例如，您可以分割時間序列資料，例如每小時、每日、每週、每月、每季、年度、自訂或以上任何組合。對於時間序列資料範例，如果您依小時分割資料表，則每個分割區都將包含一小時的資料。如果您依每日分割時間序列資料表，則每個分割區將保存一天的資料，依此類推。分割區索引鍵控制分割區的大小。

當您在分割表格上使用 INSERT 或 UPDATE SQL 命令時，資料庫引擎會將資料路由至適當的分割區。儲存資料的 PostgreSQL 表格分割區是主表格的子表格。

在資料庫查詢讀取期間，PostgreSQL 最佳化器會檢查查詢的 WHERE 子句，如果可能的話，將資料庫掃描導向僅相關的分割區。

從 10 版開始，PostgreSQL 使用宣告式分割來實作資料表分割區。這也被稱為原生 PostgreSQL 分割區。在 PostgreSQL 10 版之前，您已使用觸發器來實作分割區。

PostgreSQL 表格分割區提供下列功能：

- 隨時建立新的分割區。
- 可變的分割區範圍。
- 使用資料定義語言 (DDL) 陳述式的可分離和可重新連接分割區。

例如，可拆分的分割區對於從主磁碟分割區移除歷史資料，但保留歷史資料以供分析來說十分實用。

- 新的分割區會繼承父資料庫表格屬性，包括以下各項：
 - 索引
 - 主索引鍵，其中必須包含分割區索引鍵資料欄
 - 外部索引鍵
 - 檢查限制
 - 參考
- 建立完整資料表或每個特定分割區的索引。

您無法變更個別分割區的結構描述。不過，您可以變更父資料表格 (例如新增新資料欄)，此表格會傳播到分割區。

主題

- [PostgreSQL pg_partman 擴充功能概述](#)
- [啟用 pg_partman 擴充功能](#)
- [使用 create_parent 函數設定分割區](#)
- [使用 run_maintenance_proc 函數來設定分割區維護](#)

PostgreSQL pg_partman 擴充功能概述

您可以使用 PostgreSQL pg_partman 擴充功能，以自動化資料表分割區的建立和維護。如需更多一般資訊，請參閱 pg_partman 文件中的 [PG 分割區管理員](#)。

Note

RDS for PostgreSQL 版本 12.5 及更新版本支援此 pg_partman 擴充功能。

您可以使用下列設定來設定 pg_partman，而不必手動建立每個分割區：

- 要分割的表格
- 分割區類型
- 分割區索引鍵
- 分割區間隔
- 分割區預先建立與管理選項

建立 PostgreSQL 分割區資料表之後，您可以透過呼叫 create_parent 函數，使用 pg_partman 進行註冊。這樣做會根據您傳遞給函數的參數來建立必要的分割區。

pg_partman 擴充功能還提供 run_maintenance_proc 函數，您可以安排程呼叫它以自動管理分割區。為了確保視需要建立適當的分割區，可以排程此函數定期執行 (例如每小時)。您還可以確保自動捨棄分割區。

啟用 pg_partman 擴充功能

如果您要管理相同 PostgreSQL 資料庫執行個體內多個資料庫的分割區，則必須分別為每個資料庫啟用 pg_partman 擴充功能。若要啟用特定資料庫的 pg_partman 擴充功能，請建立分割區維護結構描述，然後建立 pg_partman 擴充功能，如下所示。

```
CREATE SCHEMA partman;  
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

Note

若要建立 `pg_partman` 擴充功能，請確定您具有 `rds_superuser` 權限。

如果您收到下列錯誤訊息，請將 `rds_superuser` 權限授與帳戶或使用您的超級使用者帳戶。

```
ERROR: permission denied to create extension "pg_partman"  
HINT: Must be superuser to create this extension.
```

若要授予 `rds_superuser` 權限，請連線至您的超級使用者帳戶並執行下列命令。

```
GRANT rds_superuser TO user-or-role;
```

以顯示使用 `pg_partman` 擴充功能舉例，我們使用下面的範例資料庫表和分割區。此資料庫使用以時間戳記為基礎的分割表格。結構描述 `data_mart` 包含一個名為 `events` 的表格和一個名為 `created_at` 的欄。下列設定包含在 `events` 表格中：

- 主索引鍵 `event_id` 和 `created_at`，必須有用於引導分割區的欄。
- 強制執行 `ck_valid_operation` 表格欄值的檢查約束 `operation`。
- 兩個外鍵，其中一個 (`fk_orga_membership`) 指向外部表格 `organization`，另一個 (`fk_parent_event_id`) 是自引用的外鍵。
- 兩個索引，其中一個 (`idx_org_id`) 用於外鍵，另一個 (`idx_event_type`) 用於事件類型。

下列 DDL 陳述式會建立這些物件，這些物件會自動包含在每個分割區上。

```
CREATE SCHEMA data_mart;  
CREATE TABLE data_mart.organization ( org_id BIGSERIAL,  
    org_name TEXT,  
    CONSTRAINT pk_organization PRIMARY KEY (org_id)  
);  
  
CREATE TABLE data_mart.events(  
    event_id          BIGSERIAL,
```

```

operation      CHAR(1),
value          FLOAT(24),
parent_event_id BIGINT,
event_type     VARCHAR(25),
org_id         BIGSERIAL,
created_at     timestamp,
CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
CONSTRAINT fk_orga_membership
    FOREIGN KEY(org_id)
    REFERENCES data_mart.organization (org_id),
CONSTRAINT fk_parent_event_id
    FOREIGN KEY(parent_event_id, created_at)
    REFERENCES data_mart.events (event_id,created_at)
) PARTITION BY RANGE (created_at);

```

```

CREATE INDEX idx_org_id      ON data_mart.events(org_id);
CREATE INDEX idx_event_type ON data_mart.events(event_type);

```

使用 create_parent 函數設定分割區

啟用 `pg_partman` 擴充功能之後，可以使用此 `create_parent` 函數來設定在分割維護結構描述內的分割區。以下範例使用在 `events` 中建立的 [啟用 pg_partman 擴充功能](#) 資料表範例。如下所示呼叫 `create_parent` 函數。

```

SELECT partman.create_parent( p_parent_table => 'data_mart.events',
    p_control => 'created_at',
    p_type => 'native',
    p_interval=> 'daily',
    p_premake => 30);

```

參數如下：

- `p_parent_table` – 父項分割表格。此表必須已經存在，並且完全符合資格 (包括結構描述)。
- `p_control` – 分割區依據的欄。資料類型必須是整數或以時間為基礎。
- `p_type` – 類型可以是 'native' 或 'partman'。您通常使用 `native` 類型來改善效能和靈活性。`partman` 類型依賴繼承。
- `p_interval` – 每個分割區的時間間隔或整數範圍。範例值包括 `daily`、每小時等。
- `p_premake` – 預先建立以支援新插入的分割區數目。

如需 `create_parent` 函數的完整描述，請參閱 `pg_partman` 文件中的[建立函數](#)。

使用 `run_maintenance_proc` 函數來設定分割區維護

您可以執行分割區維護作業以自動建立新的分割區、分離分割區或移除舊的分割區。分割區維護依賴 `pg_partman` 擴充功能的 `run_maintenance_proc` 函數和 `pg_cron` 擴充功能，它們可以啟動內部排程器。`pg_cron`排程器會自動執行資料庫中定義的 SQL 陳述式、函數和程序。

下列範例使用中建立於 `events` 中的[啟用 `pg_partman` 擴充功能](#)表格範例，將分割區維護作業設定為自動執行。作為必要條件，將 `pg_cron` 新增至資料庫執行個體的參數群組中的 `shared_preload_libraries` 參數。

```
CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

您可以在下面找到上述範例的逐步說明：

1. 修改與資料庫執行個體關聯的參數群組，並新增 `pg_cron` 至 `shared_preload_libraries` 參數值。此變更需要重新啟動資料庫執行個體才能生效。如需詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。
2. 使用具有 `CREATE EXTENSION pg_cron;` 許可的帳戶執行命令 `rds_superuser`。這會啟用 `pg_cron` 擴充功能。如需詳細資訊，請參閱 [使用 PostgreSQL `pg_cron` 擴充功能排程維護](#)。
3. 執行命令 `UPDATE partman.part_config` 以調整 `data_mart.events` 資料表的 `pg_partman` 設定。
4. 執行 `SET ...` 命令 以使用這些子句設定 `data_mart.events` 資料表：
 - a. `infinite_time_partitions = true`，– 將表格設定為能夠沒有任何限制，自動建立新的分割區。
 - b. `retention = '3 months'`，– 將表格設定為最多保留三個月。
 - c. `retention_keep_table=true` – 設定資料表，以便在保留期限到期時，資料表不會自動刪除。相反地，比保留期間還舊的分割區只會從父表格分離。

5. 執行 `SELECT cron.schedule ...` 命令 進行 `pg_cron` 函數呼叫。此呼叫定義排程器執行 `pg_partman` 維護程序 `partman.run_maintenance_proc` 的頻率。在此範例中，程序會每小時執行一次。

如需 `run_maintenance_proc` 函數的完整描述，請參閱 `pg_partman` 文件中的[維護函數](#)。

使用 PgAudit 記錄資料庫活動

金融機構、政府機構和許多產業都需要保留「稽核日誌」，以符合法規要求。透過使用 PostgreSQL 稽核擴充功能 (PgAudit) 搭配 RDS for PostgreSQL 資料庫執行個體，您可以擷取稽核人員通常需要的詳細記錄，以符合法規要求。例如，您可以設定 pgAudit 擴充功能來追蹤對特定資料庫和資料表所做的變更、記錄進行變更的使用者，以及許多其他詳細資訊。

pgAudit 擴充功能建置在原生 PostgreSQL 記錄基礎結構的功能之上，以更多的詳細資訊擴充日誌訊息。換言之，您可以使用與檢視任何日誌訊息相同的方法來檢視稽核日誌。如需 PostgreSQL 記錄的詳細資訊，請參閱 [RDS for PostgreSQL 資料庫日誌檔](#)。

pgAudit 擴充功能會從日誌中刪減敏感資料，例如純文字密碼。如果您的 RDS for PostgreSQL 資料庫執行個體設定為記錄資料操作語言 (DML) 陳述式 (如[針對您的 RDS for PostgreSQL 資料庫執行個體開啟查詢記錄](#)中所述)，您可以使用 PostgreSQL 稽核擴充功能避免純文字密碼問題。

您可以非常具體地設定對資料庫執行個體的稽核。您可以稽核所有資料庫和所有使用者。或者，您可以選擇僅稽核某些資料庫、使用者和其他物件。您也可以明確排除特定使用者和資料庫，使其不受稽核。如需詳細資訊，請參閱[從稽核記錄中排除使用者或資料庫](#)。

鑑於可以擷取的詳細資訊量，我們建議您，如果的確使用 pgAudit，請監控您的儲存耗用量。

所有可用的 RDS for PostgreSQL 版本。如需可用的 RDS for PostgreSQL 版本支援的 pgSQL 版本清單，請參閱《Amazon RDS for PostgreSQL 版本資訊》中的 [Amazon RDS for PostgreSQL 的擴充功能版本](#)。

主題

- [設定 pgAudit 擴充功能](#)
- [稽核資料庫物件](#)
- [從稽核記錄中排除使用者或資料庫](#)
- [pgAudit 擴充功能的參考](#)

設定 pgAudit 擴充功能

若要在 RDS for PostgreSQL 資料庫執行個體上設定 PgAudit 擴充功能，您必須先將 PgAudit 新增至 RDS for PostgreSQL 資料庫執行個體的自訂資料庫參數群組上的共用程式庫。如需建立自訂資料庫叢集參數群組的相關資訊，請參閱 [使用參數群組](#)。接下來，您會安裝 pgAudit 擴充功能。最後，您會指定要稽核的資料庫和物件。本節中的程序展示做法。您可以使用 AWS Management Console 或 AWS CLI。

您必須具有做為 `rds_superuser` 角色的許可，才能執行所有這些任務。

以下步驟假設您的 RDS for PostgreSQL 資料庫執行個體與自訂資料庫參數群組相關聯。

主控台

設定 pgAudit 擴充功能

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇您的 RDS for PostgreSQL 資料庫執行個體。
3. 針對您的 開啟 Configuration (組態) 索引標籤。RDS for PostgreSQL 資料庫執行個體。在執行個體詳細資訊之間，尋找 Parameter group (參數群組) 連結。
4. 選擇連結以開啟與 相關聯的自訂參數。RDS for PostgreSQL 資料庫執行個體。
5. 在 Parameters (參數) 搜尋欄位中，輸入 `shared_pre` 以尋找 `shared_preload_libraries` 參數。
6. 選擇 Edit parameters (編輯參數) 以存取屬性值。
7. 在 Values (值) 欄位中，將 `pgaudit` 新增至清單。使用逗號區隔值清單中的項目。



RDS > Parameter groups > docs-lab-rpg-14-custom-db-parameters

docs-lab-rpg-14-custom-db-parameters

Parameters

Q shared_pre X

<input type="checkbox"/>	Name	Values	Allowed values
<input type="checkbox"/>	shared_preload_libraries	pgaudit,pg_stat_statements	auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_transport, plprofiler

8. 重新啟動 RDS for PostgreSQL 資料庫執行個體，以便您對 `shared_preload_libraries` 參數所做的變更生效。
9. 當執行個體可用時，請驗證 pgAudit 是否已初始化。使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;
```



```
shared_preload_libraries
-----
rdsutils,pgaudit
(1 row)
```

10. 在初始化 pgAudit 之後，您現在可以建立擴充功能。您必須在初始化程式庫之後建立擴充功能，因為 pgaudit 擴充功能會安裝事件觸發條件，以稽核資料定義語言 (DDL) 陳述式。

```
CREATE EXTENSION pgaudit;
```

11. 關閉 psql 工作階段。

```
labdb=> \q
```

12. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
13. 在清單中尋找 pgaudit.log 參數，並設定為適合您使用案例的適當值。例如，將 pgaudit.log 參數設定為 write (如下圖所示) 會擷取日誌的插入、更新、刪除，以及其他一些類型的變更。

The screenshot shows the AWS Management Console interface for a custom DB parameter group named 'docs-lab-rpg-14-custom-db-parameters'. The 'Parameters' section is visible, with a search bar containing 'pgau'. A table lists the parameters, with 'pgaudit.log' selected. The value is set to 'write', and the allowed values are 'ddl, function, misc, read, role, write, none, all, -ddl, -function, -misc, -read, -role, -write'. The 'Modifiable' status is 'true'.

<input type="checkbox"/>	Name	Values	Allowed values	Modifiable
<input type="checkbox"/>	pgaudit.log	write	ddl, function, misc, read, role, write, none, all, -ddl, -function, -misc, -read, -role, -write	true

您也可以針對 pgaudit.log 參數選擇下列其中一個值。

- none – 這是預設值。不會記錄任何資料庫變更。
- all – 記錄一切 (read、write、function、role、ddl、misc)。
- ddl – 記錄未包含在 ROLE 類別中的所有資料定義語言 (DDL) 陳述式。
- function – 記錄函數呼叫和 DO 區塊。

- misc – 記錄其他命令，例如 DISCARD、FETCH、CHECKPOINT、VACUUM 和 SET。
 - read – 來源為關聯 (例如資料表) 或查詢時，記錄 SELECT 和 COPY。
 - role – 記錄與角色和權限相關的陳述式，例如 GRANT、REVOKE、CREATE ROLE、ALTER ROLE 和 DROP ROLE。
 - write – 目的地為關聯 (資料表) 時，記錄 INSERT、UPDATE、DELETE、TRUNCATE 和 COPY。
14. 選擇儲存變更。
 15. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
 16. 從資料庫清單中選擇 RDS for PostgreSQL 資料庫執行個體以選取它，然後從 Actions (動作) 功能表中選擇 Reboot (重新啟動)。

AWS CLI

設定 pgAudit

若要使用設定 pgAudit AWS CLI，請呼叫 [modify-db-parameter-group](#) 作業來修改自訂參數群組中的稽核記錄參數，如下列程序所示。

1. 使用下列 AWS CLI 命令，將 pgaudit 新增至 shared_preload_libraries 參數。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-  
reboot" \  
  --region aws-region
```

2. 使用下列 AWS CLI 命令重新啟動 RDS for PostgreSQL 資料庫執行個體，以便初始化 pgaudit 程式庫。

```
aws rds reboot-db-instance \  
  --db-instance-identifier your-instance \  
  --region aws-region
```

3. 當執行個體可用時，您可以驗證 pgaudit 是否已初始化。使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;  
shared_preload_libraries
```

```
-----
rdsutils,pgaudit
(1 row)
```

在初始化 PgAudit 之後，您現在可以建立擴充功能。

```
CREATE EXTENSION pgaudit;
```

- 關閉 psql 工作階段，以便您可以使用 AWS CLI。

```
labdb=> \q
```

- 使用下列 AWS CLI 命令，來指定要由工作階段稽核記錄所記錄的陳述式類別。此範例會將 pgaudit.log 參數設定為 write，此參數會擷取日誌的插入、更新和刪除操作。

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name custom-param-group-name \
  --parameters
  "ParameterName=pgaudit.log,ParameterValue=write,ApplyMethod=pending-reboot" \
  --region aws-region
```

您也可以針對 pgaudit.log 參數選擇下列其中一個值。

- none – 這是預設值。不會記錄任何資料庫變更。
- all – 記錄一切 (read、write、function、role、ddl、misc)。
- ddl – 記錄未包含在 ROLE 類別中的所有資料定義語言 (DDL) 陳述式。
- function – 記錄函數呼叫和 DO 區塊。
- misc – 記錄其他命令，例如 DISCARD、FETCH、CHECKPOINT、VACUUM 和 SET。
- read – 來源為關聯 (例如資料表) 或查詢時，記錄 SELECT 和 COPY。
- role – 記錄與角色和權限相關的陳述式，例如 GRANT、REVOKE、CREATE ROLE、ALTER ROLE 和 DROP ROLE。
- write – 目的地為關聯 (資料表) 時，記錄 INSERT、UPDATE、DELETE、TRUNCATE 和 COPY。

使用下列 AWS CLI 命令，重新啟動 RDS for PostgreSQL 資料庫執行個體。

```
aws rds reboot-db-instance \
  --db-instance-identifier your-instance \
```

```
--region aws-region
```

稽核資料庫物件

在您的 RDS for PostgreSQL 資料庫執行個體上設定 pgAudit，並針對您的要求進行設定後，會在 PostgreSQL 日誌中擷取更詳細的資訊。例如，雖然預設 PostgreSQL 記錄組態會識別在資料庫資料表中進行變更的日期和時間，但使用 pgAudit 擴充功能時，日誌項目可以包括結構描述、進行變更的使用者，以及其他詳細資訊，視擴充功能參數的設定方式而定。您可以將稽核設定為以下列方式追蹤變更。

- 對於每個工作階段，依使用者。對於工作階段層級，您可以擷取完整的命令文字。
- 對於每個物件，依使用者和資料庫。

在系統上建立 `rds_pgaudit` 角色，然後將此角色新增至自訂參數群組中的 `pgaudit.role` 參數時，便會啟用物件稽核功能。根據預設，未設定 `pgaudit.role` 參數，且唯一允許的值为 `rds_pgaudit`。下列步驟假設 `pgaudit` 已初始化，且您已遵循[設定 pgAudit 擴充功能](#)中的程序建立 `pgaudit` 擴充功能。

```
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: statement: SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: AUDIT: SESSION,2,1,READ,SELECT,TABLE,public.support,"SELECT
feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;",<none>
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: QUERY STATISTICS
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:DETAIL: ! system usage stats:
! 0.009494 s user, 0.007442 s system, 0.141985 s elapsed
! [0.022327 s user, 0.007442 s system total]
```

如此範例所示，「LOG: AUDIT: SESSION」一行提供資料表及其結構描述的相關資訊，以及其他詳細資訊。

設定物件稽核

1. 使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體。。

```
psql --host=your-instance-name.aws-region.rds.amazonaws.com --port=5432 --
username=postgrespostgres --password --dbname=labdb
```

2. 使用下列命令建立名為 `rds_pgaudit` 的資料庫角色。

```
labdb=> CREATE ROLE rds_pgaudit;
CREATE ROLE
```

```
labdb=>
```

- 關閉 psql 工作階段。

```
labdb=> \q
```

在接下來的幾個步驟中，使用 AWS CLI 修改自訂參數群組中的稽核日誌參數。

- 使用下列 AWS CLI 命令，將 `pgaudit.role` 參數設定為 `rds_pgaudit`。根據預設，此參數是空的，且 `rds_pgaudit` 是唯一允許的值。

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name custom-param-group-name \
  --parameters
  "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
  \
  --region aws-region
```

- 使用下列 AWS CLI 命令，重新啟動 RDS for PostgreSQL 資料庫執行個體，以便您對參數所做的變更生效。

```
aws rds reboot-db-instance \
  --db-instance-identifier your-instance \
  --region aws-region
```

- 執行下列命令來確認 `pgaudit.role` 已設定為 `rds_pgaudit`。

```
SHOW pgaudit.role;
pgaudit.role
-----
rds_pgaudit
```

如要測試 `pgAudit` 記錄功能，您可以執行幾個要稽核的範例命令。例如，您可以執行下列命令。

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;
id
----
(0 rows)
```

資料庫記錄應包含類似以下的項目。

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

如需有關檢視日誌的資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。

[要了解有關 PGAudit 擴展的更多信息，請參閱上的。GitHub](#)

從稽核記錄中排除使用者或資料庫

如 [RDS for PostgreSQL 資料庫日誌檔](#) 中所述，PostgreSQL 日誌會取用儲存空間。使用 pgAudit 擴充功能會在不同程度上增加日誌中收集的資料量，取決於您追蹤的變更。您可能不需要稽核中的每個使用者或資料庫。RDS for PostgreSQL 資料庫執行個體。

若要將對儲存的影響降到最低，並避免不必要地擷取稽核記錄，您可以將使用者和資料庫排除在稽核之外。您也可以給定的工作階段中變更記錄。下列範例向您展示做法。

Note

優先處理工作階段層級的參數設定，再處理 RDS for PostgreSQL 資料庫執行個體中的設定。如果您不想要資料庫使用者略過稽核記錄組態設定，請務必變更其許可。

假設您的 RDS for PostgreSQL 資料庫執行個體已設定為稽核所有使用者和資料庫的相同層級活動。然後，您決定不想要稽核使用者 myuser。您可以使用下列 SQL 命令來關閉 myuser 的稽核。

```
ALTER USER myuser SET pgaudit.log TO 'NONE';
```

然後，您可以使用下列查詢來檢查 pgaudit.log 的 user_specific_settings 欄，以確認參數已設定為 NONE。

```
SELECT
    username AS user_name,
    useconfig AS user_specific_settings
FROM
    pg_user
WHERE
```

```
username = 'myuser';
```

您會看到如下輸出。

```
user_name | user_specific_settings
-----+-----
myuser    | {pgaudit.log=NONE}
(1 row)
```

您可以使用下列命令，在資料庫的工作階段當中關閉特定使用者的記錄。

```
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'none';
```

使用下列查詢，針對特定使用者和資料庫組合檢查 pgaudit.log 的 settings 欄。

```
SELECT
    username AS "user_name",
    datname AS "database_name",
    pg_catalog.array_to_string(setconfig, E'\n') AS "settings"
FROM
    pg_catalog.pg_db_role_setting s
    LEFT JOIN pg_catalog.pg_database d ON d.oid = setdatabase
    LEFT JOIN pg_catalog.pg_user r ON r.usesysid = setrole
WHERE
    username = 'myuser'
    AND datname = 'mydatabase'
ORDER BY
    1,
    2;
```

您會看到類似下列的輸出。

```
user_name | database_name | settings
-----+-----+-----
myuser    | mydatabase    | pgaudit.log=none
(1 row)
```

在關閉 myuser 的稽核之後，您決定不想要追蹤 mydatabase 的變更。您可以使用下列命令來關閉該特定資料庫的稽核。

```
ALTER DATABASE mydatabase SET pgaudit.log to 'NONE';
```

然後，使用下列查詢來檢查 `database_specific_settings` 欄，以確認 `pgaudit.log` 已設定為 `NONE`。

```
SELECT
a.datname AS database_name,
b.setconfig AS database_specific_settings
FROM
pg_database a
FULL JOIN pg_db_role_setting b ON a.oid = b.setdatabase
WHERE
a.datname = 'mydatabase';
```

您會看到如下輸出。

```
database_name | database_specific_settings
-----+-----
mydatabase    | {pgaudit.log=NONE}
(1 row)
```

若要將設定回復為 `myuser` 的預設設定，請使用下列命令：

```
ALTER USER myuser RESET pgaudit.log;
```

若要將設定回復為資料庫的預設設定，請使用下列命令。

```
ALTER DATABASE mydatabase RESET pgaudit.log;
```

若要將使用者和資料庫重設為預設設定，請使用下列命令。

```
ALTER USER myuser IN DATABASE mydatabase RESET pgaudit.log;
```

您也可以將 `pgaudit.log` 設定為 `pgaudit.log` 參數的其他允許值之一，將特定事件擷取至日誌。如需詳細資訊，請參閱[允許的 pgaudit.log 參數設定清單](#)。

```
ALTER USER myuser SET pgaudit.log TO 'read';
ALTER DATABASE mydatabase SET pgaudit.log TO 'function';
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'read,function'
```

pgAudit 擴充功能的參考

您可以變更本節中列出的一或多個參數，針對稽核日誌指定您所需的詳細資訊層級。

控制 pgAudit 行為

您可以變更下表中列出的一或多個參數來控制稽核記錄。

參數	描述
<code>pgaudit.log</code>	指定工作階段稽核記錄將記錄哪些陳述式類別。允許的值包括 <code>ddl</code> 、 <code>function</code> 、 <code>misc</code> 、 <code>read</code> 、 <code>role</code> 、 <code>write</code> 、 <code>none</code> 、 <code>all</code> 。如需詳細資訊，請參閱 允許的 <code>pgaudit.log</code> 參數設定清單 。
<code>pgaudit.log_catalog</code>	開啟 (設定為 1) 時，如果陳述式中的所有關聯都在 <code>pg_catalog</code> 中，則會將陳述式新增至稽核線索。
<code>pgaudit.log_level</code>	指定要用於日誌項目的日誌層級。允許的值： <code>disable</code> 、 <code>debug5</code> 、 <code>debug4</code> 、 <code>debug3</code> 、 <code>debug2</code> 、 <code>debug1</code> 、 <code>info</code> 、 <code>notice</code> 、 <code>warning</code> 、 <code>log</code>
<code>pgaudit.log_parameter</code>	開啟 (設定為 1) 時，會在稽核日誌中擷取隨陳述式傳遞的參數。
<code>pgaudit.log_relation</code>	開啟 (設定為 1) 時，工作階段的稽核日誌會為 <code>SELECT</code> 或 <code>DML</code> 陳述式中參考的每個關聯 (<code>TABLE</code> 、 <code>VIEW</code> 等) 建立個別の日誌項目。
<code>pgaudit.log_statement_once</code>	指定日誌記錄包含的陳述式文字和參數，具有陳述式/子陳述式組合的第一個日誌項目，還是具有每個項目。
<code>pgaudit.role</code>	指定用於物件稽核日誌記錄的主要角色。唯一允許的項目為 <code>rds_pgaudit</code> 。

允許的 `pgaudit.log` 參數設定清單

值	描述
無	此為預設值。不會記錄任何資料庫變更。
全部	記錄一切 (<code>read</code> 、 <code>write</code> 、 <code>function</code> 、 <code>role</code> 、 <code>ddl</code> 、 <code>misc</code>)。

值	描述
ddl	記錄未包含在 ROLE 類別中的所有資料定義語言 (DDL) 陳述式。
函數	記錄函數呼叫和 DO 區塊。
misc	記錄其他命令，例如 DISCARD、FETCH、CHECKPOINT、VACUUM 和 SET。
讀取	來源為關聯 (例如資料表) 或查詢時，記錄 SELECT 和 COPY。
role	記錄與角色和權限相關的陳述式，例如 GRANT、REVOKE、CREATE ROLE、ALTER ROLE 和 DROP ROLE。
write	目的地為關聯 (資料表) 時，記錄 INSERT、UPDATE、DELETE、TRUNCATE 和 COPY。

若要使用工作階段稽核功能記錄多個事件類型，請使用逗號分隔清單。若要記錄所有事件類型，請將 `pgaudit.log` 設定為 ALL。重新啟動資料庫執行個體以套用變更。

您可以透過物件稽核，調整稽核記錄以使用特定關聯。例如，您可以指定您想要針對一或多個資料表上的 READ 操作稽核記錄。

使用 PostgreSQL pg_cron 擴充功能排程維護

您可以使用 PostgreSQL pg_cron 擴充功能，來排程 PostgreSQL 資料庫內的維護命令。如需擴充功能的詳細資訊，請參閱 pg_cron 文件中的[什麼是 pg_cron？](#)。

RDS for PostgreSQL 引擎 12.5 版及更新版本支援 pg_cron 擴充功能。

若要深入了解如何使用 pg_cron，請參閱[在 RDS for PostgreSQL 或 Aurora PostgreSQL 相容版本資料庫上使用 pg_cron 排程任務](#)

主題

- [設定 pg_cron 擴充功能](#)
- [授與資料庫使用者使用 pg_cron 的許可](#)
- [排定 pg_cron 任務](#)
- [pg_cron 擴充功能的參考](#)

設定 pg_cron 擴充功能

設定 pg_cron 擴充功能，如下所示：

1. 修改與 PostgreSQL 資料庫執行個體關聯的自訂參數群組，並將 pg_cron 新增至 shared_preload_libraries 參數值。
 - 如果 RDS for PostgreSQL 資料庫執行個體使用 rds.allowed_extensions 參數顯示列可以安裝的擴展名清單，您需要將 pg_cron 擴展功能新增至清單。只有特定版本的 RDS for PostgreSQL 支援 rds.allowed_extensions 參數。根據預設，允許所有可用的擴充功能。如需詳細資訊，請參閱 [限制安裝 PostgreSQL 擴充功能](#)。

重新啟動 PostgreSQL 資料庫執行個體，使參數群組的變生效。若要深入了解如何使用參數群組，請參閱 [修改資料庫參數群組中的參數](#)。

2. 重新啟動 PostgreSQL 資料庫執行個體之後，請使用具有 rds_superuser 許可的帳戶執行下列命令。例如，如果您在為 RDS for PostgreSQL 資料庫執行個體建立 RDS 時使用了預設設定，請以使用者 postgres 身分連接並建立擴充功能。

```
CREATE EXTENSION pg_cron;
```

pg_cron 排程器設定於名為 postgres 的預設 PostgreSQL 資料庫內。pg_cron 物件會在此 postgres 資料庫中建立，而且所有排程動作都會在此資料庫中執行。

- 您可以使用預設設定，或將任務排程在 PostgreSQL 資料庫執行個體的其他資料庫中執行。若要將任務排程至 PostgreSQL 資料庫執行個體中的其他資料庫，請參閱 [為預設資料庫以外的資料庫排程 cron 任務](#) 中的範例。

授與資料庫使用者使用 pg_cron 的許可

安裝 pg_cron 擴充功能需要 rds_superuser 權限。然而，使用 pg_cron 的許可能夠授予 (由 rds_superuser 群組/角色的成員) 其他資料庫使用者，讓他們可以安排自己的任務。建議您只有在能改善生產環境中的作業時，才視需要授予對 cron 結構描述的許可。

若要授予資料庫使用者對 cron 結構描述的許可，請執行以下命令：

```
postgres=> GRANT USAGE ON SCHEMA cron TO db-user;
```

這樣會給予 *db-user* 存取 cron 結構描述的許可，以針對它們有權存取的物件排程 cron 作業。如果資料庫使用者沒有許可，則作業會在將錯誤訊息發佈至 postgresql.log 檔案後失敗，如下所示：

```
2020-12-08 16:41:00 UTC::@[30647]:ERROR: permission denied for table table-name
2020-12-08 16:41:00 UTC::@[27071]:LOG: background worker "pg_cron" (PID 30647) exited
with exit code 1
```

換句話說，請確定被授與 cron 結構描述權限的資料庫使用者也擁有他們計劃排程之物件 (表格、結構描述等) 的權限。

Cron 工作的詳細資訊及其成功或失敗也會擷取在資料 cron.job_run_details 表中。如需詳細資訊，請參閱 [用於排程工作和擷取狀態的資料表](#)。

排定 pg_cron 任務

以下各節示範如何使用 pg_cron 任務排程各種管理任務。

Note

建立 pg_cron 工作時，請檢查 max_worker_processes 設定大於數目 cron.max_running_jobs。如果 pg_cron 任務耗盡背景工作者程序，則會失敗。pg_cron 任務的預設數字為 5。如需詳細資訊，請參閱 [用於管理 pg_cron 擴充功能的參數](#)。

主題

- [正在清理資料表](#)
- [正在清除 pg_cron 歷史記錄資料表](#)
- [僅將錯誤記錄到 postgresql.log 檔案](#)
- [為預設資料庫以外的資料庫排程 cron 任務](#)

正在清理資料表

自動資料清理功能可處理大多數情況下的清理維護。但是，您可能想要在您選定的時間來排定清理特定的資料表。

另請參閱 [在 Amazon RDS for PostgreSQL 上使用 PostgreSQL 自動資料清理](#)。

以下為使用 `cron.schedule` 函數來設定一個任務，以在每天 22:00 (GMT) 於特定資料表上使用 `VACUUM FREEZE` 的範例。

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
schedule
-----
1
(1 row)
```

執行上述範例之後，您可以如下所示檢查 `cron.job_run_details` 資料表中的歷史記錄。

```
postgres=> SELECT * FROM cron.job_run_details;
jobid | runid | job_pid | database | username | command | status | return_message | start_time | end_time
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | 1 | 3395 | postgres | adminuser | vacuum freeze pgbench_accounts | succeeded | VACUUM | 2020-12-04 21:10:00.050386+00 | 2020-12-04 21:10:00.072028+00
(1 row)
```

以下是 `cron.job_run_details` 表的查詢，看到失敗的作業。

```
postgres=> SELECT * FROM cron.job_run_details WHERE status = 'failed';
jobid | runid | job_pid | database | username | command | status | return_message | start_time | end_time
```

```

-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5      | 4      | 30339   | postgres | adminuser| vacuum freeze pgbench_account | failed
| ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 |
2020-12-04 21:48:00.029567+00
(1 row)

```

如需詳細資訊，請參閱 [用於排程工作和擷取狀態的資料表](#)。

正在清除 pg_cron 歷史記錄資料表

cron.job_run_details 資料表包含 cron 任務的歷史記錄，它可能會隨著時間的推移而變得非常大。建議您排程清除此資料表的任務。例如，保留一週的項目可能足以進行疑難排解。

下列範例會使用 [cron.schedule](#) 函數來排程每天在午夜執行的任務，以清除 cron.job_run_details 資料表。這項任務只保留過去七天。如下所示使用您的 rds_superuser 帳戶來排程任務。

```

SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);

```

如需詳細資訊，請參閱 [用於排程工作和擷取狀態的資料表](#)。

僅將錯誤記錄到 postgresql.log 檔案

若要防止寫入至 cron.job_run_details 資料表，請修改與 PostgreSQL 資料庫執行個體關聯的參數群組，並將 cron.log_run 參數設定為關閉。pg_cron 延伸模組不再寫入至資料表，而且只會將錯誤擷取至 postgresql.log 檔案。如需詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

使用下列命令來檢查 cron.log_run 參數的值。

```

postgres=> SHOW cron.log_run;

```

如需詳細資訊，請參閱 [用於管理 pg_cron 擴充功能的參數](#)。

為預設資料庫以外的資料庫排程 cron 任務

pg_cron 的中繼資料都保留在名為 postgres 的 PostgreSQL 預設資料庫中。由於使用背景工作者來執行維護 cron 任務，因此您可以在 PostgreSQL 資料庫執行個體內的任何資料庫中排程任務：

1. 在 cron 資料庫中，使用 [cron.schedule](#) 如常排程任務。

```
postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum
freeze test_table');
```

2. 作為使用 rds_superuser 角色的使用者，請為您剛建立的任務更新資料庫欄，以便在 PostgreSQL 資料庫執行個體內的另一個資料庫中執行。

```
postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
```

3. 透過查詢 cron.job 資料表進行驗證。

```
postgres=> SELECT * FROM cron.job;
jobid | schedule      | command                                     | nodename | nodeport |
database | username  | active | jobname
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
106   | 29 03 * * * | vacuum freeze test_table                   | localhost | 8192     |
database1 | adminuser | t      | database1 manual vacuum
1     | 59 23 * * * | vacuum freeze pgbench_accounts            | localhost | 8192     |
postgres | adminuser | t      | manual vacuum
(2 rows)
```

Note

在某些情況下，您可能會新增一個打算在其他資料庫上執行的 cron 任務。在此情況下，任務可能會嘗試在預設資料庫 (postgres) 中執行，然後再更新正確的資料庫資料欄。如果使用者名稱具有許可，則任務會在預設資料庫中成功執行。

pg_cron 擴充功能的參考

您可以使用下列參數、函數和資料表搭配 pg_cron 擴充功能。如需詳細資訊，請參閱 pg_cron 文件中的 [什麼是 pg_cron?](#)

主題

- [用於管理 pg_cron 擴充功能的參數](#)
- [函數參考：cron.schedule](#)
- [函數參考：cron.unschedule](#)

- [用於排程工作和擷取狀態的資料表](#)

用於管理 pg_cron 擴充功能的參數

以下是用來控制 pg_cron 擴充功能行為的參數清單。

參數	描述
cron.database_name	在其中保留 pg_cron 中繼資料的資料庫。
cron.host	要連線至 PostgreSQL 的主機名稱。您無法修改此值。
cron.log_run	記錄 job_run_details 資料表執行的每個任務。值為 on 或 off。如需詳細資訊，請參閱 用於排程工作和擷取狀態的資料表 。
cron.log_statement	在執行之前記錄所有 cron 陳述式。值為 on 或 off。
cron.max_running_jobs	可同時執行的任務數量上限。
cron.use_background_workers	使用背景工作者而不是用戶端工作階段。您無法修改此值。

使用下列 SQL 命令來顯示這些參數及其值。

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%'  
ORDER BY name;
```

函數參考：cron.schedule

這個函數會排程一個 cron 任務。任務最初是在預設 postgres 資料庫中進行排程。該函數會返回表示任務識別符的 bigint 值。若要將任務排程在 PostgreSQL 資料庫執行個體中的其他資料庫中執行，請參閱 [為預設資料庫以外的資料庫排程 cron 任務](#) 中的範例。

該函數有兩種語法格式。

語法

```
cron.schedule (job_name,
               schedule,
               command
            );

cron.schedule (schedule,
               command
            );
```

參數

參數	描述
job_name	cron 任務的名稱。
schedule	指出 cron 任務排程的文字。格式為標準 cron 格式。
command	要執行的命令文字。

範例

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
 schedule
-----
          145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
-----
          146
(1 row)
```

函數參考：cron.unschedule

這個函數會刪除 cron 任務。您可以指定 job_name 或 job_id，兩者之一。政策可確保您是移除任務排程的擁有者。該函數會返回一個表示成功或失敗的布林值。

此函數的語法格示如下。

語法

```
cron.unschedule (job_id);  
  
cron.unschedule (job_name);
```

參數



參數	描述
job_id	排程 cron 任務時，從 <code>cron.schedule</code> 函數傳回的任務識別符。
job_name	使用 <code>cron.schedule</code> 函數排程的 cron 任務的名稱。

範例

```
postgres=> SELECT cron.unschedule(108);  
  unschedule  
-----  
  t  
(1 row)  
  
postgres=> SELECT cron.unschedule('test');  
  unschedule  
-----  
  t  
(1 row)
```

用於排程工作和擷取狀態的資料表

下表用於排程 cron 任務以及記錄任務完成的方式。

表格	描述
<code>cron.job</code>	<p>包含每個排程任務的中繼資料。大部分與此資料表的互動應透過使用 <code>cron.schedule</code> 和 <code>cron.unschedule</code> 函數來完成。</p> <div data-bbox="591 369 1507 638"><p> Important</p><p>我們建議您不要將更新或插入權限直接提供給此資料表。這樣做將允許使用者更新 <code>username</code> 資料欄，以 <code>rds-superuser</code> 身分執行任務。</p></div>
<code>cron.job_run_details</code>	<p>包含過去排程任務執行的歷史記錄資訊。這對於調查任務執行的狀態、傳回訊息以及開始和結束時間非常有用。</p> <div data-bbox="591 802 1507 1020"><p> Note</p><p>若要防止此資料表無限期增長，請定期清除。如需範例，請參閱 正在清除 pg_cron 歷史記錄資料表。</p></div>

使用 pglogical 跨執行個體同步資料

所有目前可用的 RDS for PostgreSQL 版本都支援 pglogical 延伸模組。pglogical 延伸模組早於 PostgreSQL 在第 10 版中引入的功能，其類似於邏輯複寫功能。如需詳細資訊，請參閱 [Amazon RDS for PostgreSQL 執行邏輯複寫](#)。

pglogical 延伸模組支援在兩個或以上之間進行邏輯複寫。RDS for PostgreSQL 資料庫執行個體。它也支援在不同的 PostgreSQL 版本之間，以及在 RDS for PostgreSQL 資料庫執行個體和 Aurora PostgreSQL 資料庫叢集上執行的資料庫之間進行複寫。pglogical 延伸模組會使用發佈-訂閱模型，將資料表和其他物件 (例如序列) 的變更從發佈者複製到訂閱者。它依賴複寫槽，以確保變更從發佈者節點同步到訂閱者節點，其定義如下。

- 發佈者節點是 RDS for PostgreSQL 資料庫執行個體，其是要複寫到其他節點的資料來源。發佈者節點定義要在發佈集中複製的資料表。
- 訂閱者節點是 RDS for PostgreSQL 資料庫執行個體，其會從發佈者接收 WAL 更新。訂閱者會建立訂閱以連線至發佈者，並取得解碼的 WAL 資料。訂閱者建立訂閱時，系統會在發佈者節點上建立複寫槽。

您可以在以下內容中找到如何設定 pglogical 延伸模組的相關資訊。

主題

- [pglogical 延伸模組的需求和限制](#)
- [設定 pglogical 延伸模組](#)
- [針對 RDS for PostgreSQL 資料庫執行個體設定邏輯複寫](#)
- [在重大升級之後重新建立邏輯複寫](#)
- [管理 RDS for PostgreSQL 的邏輯複寫槽](#)
- [pglogical 延伸模組的參數參考](#)

pglogical 延伸模組的需求和限制

所有目前可用的 RDS for PostgreSQL 版本都支援 pglogical 延伸模組。

發佈者節點和訂閱者節點都必須針對邏輯複寫進行設定。

您想要從訂閱者複寫到發佈者的資料表必須具有相同的名稱和相同的結構描述。這些資料表亦須包含相同的資料欄，而且這些資料欄必須使用相同的資料類型。發佈者和訂閱者資料表都必須具有相同的主索引鍵。建議您僅使用 PRIMARY KEY 作為唯一限制條件。

對於 CHECK 限制條件和 NOT NULL 限制條件，訂閱者節點上的資料表與發佈者節點上的資料表相比之下，可以具有更寬鬆的限制條件。

pglogical 延伸模組會提供 PostgreSQL (第 10 版及更新版本) 內建邏輯複寫功能不支援的功能，例如雙向複寫。如需詳細資訊，請參閱[使用 pglogical 進行 PostgreSQL 雙向複寫](#)。

設定 pglogical 延伸模組

若要在 RDS for PostgreSQL 資料庫執行個體上設定 pglogical 延伸模組，請將 pglogical 新增至下列群組上的共用程式庫：RDS for PostgreSQL 資料庫執行個體的自訂資料庫參數群組。您亦需將 `rds.logical_replication` 參數的值設為 1，以開啟邏輯解碼。最後，您可以在資料庫中建立延伸模組。您可以針對這些任務使用 AWS Management Console 或 AWS CLI。

您必須具有做為 `rds_superuser` 角色的許可，才能執行這些任務。

以下步驟假設您的 RDS for PostgreSQL 資料庫執行個體與自訂資料庫參數群組相關聯。如需建立自訂資料庫叢集參數群組的相關資訊，請參閱[使用參數群組](#)。

主控台

設定 pglogical 延伸模組

1. 登入 AWS Management Console，並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇您的 RDS for PostgreSQL 資料庫執行個體。
3. 針對您的 開啟 Configuration (組態) 索引標籤。RDS for PostgreSQL 資料庫執行個體。在執行個體詳細資訊之間，尋找 Parameter group (參數群組) 連結。
4. 選擇連結以開啟與 相關聯的自訂參數。RDS for PostgreSQL 資料庫執行個體。
5. 在 Parameters (參數) 搜尋欄位中，輸入 `shared_pre` 以尋找 `shared_preload_libraries` 參數。
6. 選擇 Edit parameters (編輯參數) 以存取屬性值。
7. 在 Values (值) 欄位中，將 pglogical 新增至清單。使用逗號區隔值清單中的項目。

RDS > Parameter groups > docs-lab-rpg-12-parameter-group

docs-lab-rpg-12-parameter-group

Parameters

Q shared_pre X

<input type="checkbox"/>	Name	Values	Allowed values
<input type="checkbox"/>	shared_preload_libraries	pglogical,pg_stat_statements	auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_transport, plprofiler

8. 尋找 `rds.logical_replication` 參數並將其設為 1，以開啟邏輯複寫。
9. 重新啟動 RDS for PostgreSQL 資料庫執行個體，讓您的變更生效。
10. 當執行個體可用時，您可以使用 `psql` (或 `pgAdmin`) 連線至 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

11. 若要驗證 `pglogical` 是否已初始化，請執行下列命令。

```
SHOW shared_preload_libraries;
shared_preload_libraries
-----
rdsutils,pglogical
(1 row)
```

12. 驗證啟用邏輯解碼的設定，如下所示。

```
SHOW wal_level;
wal_level
-----
logical
(1 row)
```

13. 建立延伸模組，如下所示。

```
CREATE EXTENSION pglogical;  
EXTENSION CREATED
```

14. 選擇儲存變更。

15. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。

16. 從資料庫清單中選擇 RDS for PostgreSQL 資料庫執行個體以選取它，然後從 Actions (動作) 功能表中選擇 Reboot (重新啟動)。

AWS CLI

設定 pglogical 延伸模組

若要使用設定 pglogical AWS CLI，請呼叫 [modify-db-parameter-group](#) 作業來修改自訂參數群組中的某些參數，如下列程序所示。

1. 使用下列 AWS CLI 命令，將 pglogical 新增至 shared_preload_libraries 參數。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=shared_preload_libraries,ParameterValue=pglogical,ApplyMethod=pending-  
reboot" \  
  --region aws-region
```

2. 使用下列 AWS CLI 命令將 rds.logical_replication 設為 1，以開啟的邏輯解碼功能。RDS for PostgreSQL 資料庫執行個體。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=rds.logical_replication,ParameterValue=1,ApplyMethod=pending-  
reboot" \  
  --region aws-region
```

3. 使用下列 AWS CLI 命令重新啟動 RDS for PostgreSQL 資料庫執行個體，以便初始化 pglogical 程式庫。

```
aws rds reboot-db-instance \  

```

```
--db-instance-identifier your-instance \  
--region aws-region
```

- 當執行個體可用時，請使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password --dbname=labdb
```

- 建立延伸模組，如下所示。

```
CREATE EXTENSION pglogical;  
EXTENSION CREATED
```

- 使用下列 AWS CLI 命令，重新啟動 RDS for PostgreSQL 資料庫執行個體。

```
aws rds reboot-db-instance \  
--db-instance-identifier your-instance \  
--region aws-region
```

針對 RDS for PostgreSQL 資料庫執行個體設定邏輯複寫

下列程序說明如何在兩個 RDS for PostgreSQL 資料庫執行個體之間啟動邏輯複寫。這些步驟假設來源 (發佈者) 和目標 (訂閱者) 都已設定 `pglogical` 延伸模組，如 [設定 pglogical 延伸模組](#) 中所述。

建立發佈者節點並定義要複製的資料表

這些步驟假設您的 RDS for PostgreSQL 資料庫執行個體具有一個資料庫，其中包含一或多個您要複製到另一個節點的資料表。您必須在訂閱者上重新建立來自發佈者的資料表結構，因此必要時先取得資料表結構。您可以使用 `psql` 中繼命令 `\d tablename`，然後在訂閱者執行個體上建立相同的資料表，以執行該操作。下列程序會在發佈者 (來源) 上建立範例資料表，以供示範之用。

- 使用 `psql` 連線至具有資料表的執行個體，您想要將該資料表用作訂閱者的來源。

```
psql --host=source-instance.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password --dbname=labdb
```

如果您沒有要複製的現有資料表，則可以建立如下的範例資料表。

- 使用下列 SQL 陳述式建立範例資料表。


```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

- b. 使用下列 SQL 陳述式，將產生的資料填入資料表中。

```
INSERT INTO docs_lab_table VALUES (generate_series(1,5000));
INSERT 0 5000
```

- c. 使用下列 SQL 陳述式，驗證資料是否存在於資料表中。

```
SELECT count(*) FROM docs_lab_table;
```

2. 將此 RDS for PostgreSQL 資料庫執行個體識別為發佈者節點，如下所示。

```
SELECT pglogical.create_node(
  node_name := 'docs_lab_provider',
  dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
  dbname=labdb');
create_node
-----
 3410995529
(1 row)
```

3. 將您要複寫的資料表新增至預設複寫集。如需複寫集的詳細資訊，請參閱 pglogical 文件中的 [Replication sets](#) (複寫集)。

```
SELECT pglogical.replication_set_add_table('default', 'docs_lab_table', 'true',
NULL, NULL);
replication_set_add_table
-----
t
(1 row)
```

發佈者節點設定完成。您現在可以設定訂閱者節點，從發佈者接收更新。

設定訂閱者節點並建立訂閱來接收更新

這些步驟假設 RDS for PostgreSQL 資料庫執行個體已使用 pglogical 延伸模組進行設定。如需詳細資訊，請參閱 [設定 pglogical 延伸模組](#)。

1. 使用 psql 來連線至您想要從發佈者接收更新的執行個體。

```
psql --host=target-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. 在訂閱者 RDS for PostgreSQL 資料庫執行個體上，建立存在於發佈者上的相同資料表。在此範例中，資料表是 docs_lab_table。您可以建立資料表，如下所示。

```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

3. 驗證此資料表是否為空的。

```
SELECT count(*) FROM docs_lab_table;
count
-----
0
(1 row)
```

4. 將此 RDS for PostgreSQL 資料庫執行個體識別為訂閱者節點，如下所示。

```
SELECT pglogical.create_node(
    node_name := 'docs_lab_target',
    dsn := 'host=target-instance.aws-region.rds.amazonaws.com port=5432
sslmode=require dbname=labdb user=postgres password=*****');
create_node
-----
2182738256
(1 row)
```

5. 建立訂閱。

```
SELECT pglogical.create_subscription(
    subscription_name := 'docs_lab_subscription',
    provider_dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
sslmode=require dbname=labdb user=postgres password=*****',
    replication_sets := ARRAY['default'],
    synchronize_data := true,
    forward_origins := '{}');
create_subscription
-----
1038357190
(1 row)
```

完成此步驟時，會在訂閱者上的資料表中建立來自發佈者上資料表的資料。您可以使用下列 SQL 查詢來驗證是否已發生此情況。

```
SELECT count(*) FROM docs_lab_table;
count
-----
 5000
(1 row)
```

從此開始，對發佈者上資料表所做的變更會複寫到訂閱者上的資料表。

在重大升級之後重新建立邏輯複寫

在對設為邏輯複寫之發佈者節點的 RDS for PostgreSQL 資料庫執行個體執行主要版本升級之前，您必須捨棄所有複寫槽，即使是未作用中的複寫槽也一樣。建議您暫時從發佈者節點轉移資料庫交易、捨棄複寫槽、升級 RDS for PostgreSQL 資料庫執行個體，然後重新建立並重新啟動複寫。

複寫槽僅託管於發佈者節點上。邏輯複寫案例中的 RDS for PostgreSQL 訂閱者節點沒有要捨棄的複寫槽，但在其指定為具有發佈者訂閱的訂閱者節點時，無法升級至主要版本。在升級 RDS for PostgreSQL 訂閱者節點之前，請先捨棄訂閱和節點。如需詳細資訊，請參閱 [管理 RDS for PostgreSQL 的邏輯複寫槽](#)。

判斷邏輯複寫是否已中斷

您可以查詢發佈者節點或訂閱者節點，來判斷複寫程序是否已中斷，如下所示。

檢查發佈者節點

- 使用 psql 連線到發佈者節點，然後查詢 pg_replication_slots 函數。請注意作用中資料欄中的值。通常，這會傳回 t (true)，表明複寫作用中。如果查詢傳回 f (false)，表示已停止複寫至訂閱者。

```
SELECT slot_name,plugin,slot_type,active FROM pg_replication_slots;
slot_name | plugin | slot_type | active
-----+-----+-----+-----
pgl_labdb_docs_labcb4fa94_docs_lab3de412c | pglogical_output | logical | f
(1 row)
```

檢查訂閱者節點

在訂閱者節點上，您可以採取三種不同的方式來檢查複寫的狀態。

- 查看訂閱者節點上的 PostgreSQL 日誌，以找出失敗訊息。日誌會以包含結束代碼 1 的訊息識別失敗，如下所示。

```
2022-07-06 16:17:03 UTC::@[7361]:LOG: background worker "pglogical apply
16404:2880255011" (PID 14610) exited with exit code 1
2022-07-06 16:19:44 UTC::@[7361]:LOG: background worker "pglogical apply
16404:2880255011" (PID 21783) exited with exit code 1
```

- 查詢 `pg_replication_origin` 函數。使用 `psql` 連線至訂閱者節點上的資料庫，然後查詢 `pg_replication_origin` 函數，如下所示。

```
SELECT * FROM pg_replication_origin;
roident | roname
-----+-----
(0 rows)
```

空白結果集表示複寫已中斷。通常，您會看到如下輸出。

```
roident | roname
-----+-----
1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

- 查詢 `pglogical.show_subscription_status` 函數，如下列範例所示。

```
SELECT subscription_name,status,slot_name FROM pglogical.show_subscription_status();
subscription_name | status | slot_name
-----+-----+-----
docs_lab_subscription | down | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

此輸出表明複寫已中斷。其狀態為 `down`。通常，輸出會將狀態顯示為 `replicating`。

如果您的邏輯複寫程序已中斷，您可以遵循下列步驟重新建立複寫。

重新建立發佈者與訂閱者節點之間的邏輯複寫

若要重新建立複寫，首先中斷訂閱者與發佈者節點的連線，然後重新建立訂閱，如下列步驟所述。

1. 使用 `psql` 連線至訂閱者節點，如下所示。

```
psql --host=222222222222.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. 使用 `pglogical.alter_subscription_disable` 函數停用訂閱。

```
SELECT pglogical.alter_subscription_disable('docs_lab_subscription',true);
alter_subscription_disable
-----
t
(1 row)
```

3. 透過查詢 `pg_replication_origin` 取得發佈者節點的識別符，如下所示。

```
SELECT * FROM pg_replication_origin;
roident |          roname
-----+-----
1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

4. 使用上一個步驟的回應搭配 `pg_replication_origin_create` 命令，來指派訂閱可在重新建立時使用的識別符。

```
SELECT pg_replication_origin_create('pgl_labdb_docs_labcb4fa94_docs_lab3de412c');
pg_replication_origin_create
-----
1
(1 row)
```

5. 透過傳送訂閱的名稱來開啟訂閱，其狀態為 `true`，如下列範例所示。

```
SELECT pglogical.alter_subscription_enable('docs_lab_subscription',true);
alter_subscription_enable
-----
t
(1 row)
```

檢查節點的狀態。其狀態應為 `replicating`，如這個範例所示。

```
SELECT subscription_name,status,slot_name
FROM pglogical.show_subscription_status();
      subscription_name | status | slot_name
-----+-----+-----
 docs_lab_subscription | replicating |
 pgl_labdb_docs_lab98f517b_docs_lab3de412c
(1 row)
```

檢查發佈者節點上訂閱者複寫槽的狀態。複寫槽的 `active` 資料欄應傳回 `t` (`true`)，表示已重新建立複寫。

```
SELECT slot_name,plugin,slot_type,active
FROM pg_replication_slots;
      slot_name | plugin | slot_type | active
-----+-----+-----+-----
 pgl_labdb_docs_lab98f517b_docs_lab3de412c | pglogical_output | logical | t
(1 row)
```

管理 RDS for PostgreSQL 的邏輯複寫槽

在對充當邏輯複寫案例中發佈者節點的 RDS for PostgreSQL 資料庫執行個體執行主要版本升級之前，您必須捨棄執行個體上的複寫槽。主要版本升級預先檢查程序會通知您，除非捨棄複寫槽，否則升級無法繼續進行。

若要從 RDS for PostgreSQL 資料庫執行個體中捨棄複寫槽，請先捨棄訂閱，然後捨棄複寫槽。

若要識別已使用 `pglogical` 延伸模組建立的複寫槽，請登入每個資料庫並取得節點的名稱。查詢訂閱者節點時，您會在輸出中同時取得發佈者和訂閱者節點，如本範例所示。

```
SELECT * FROM pglogical.node;
 node_id | node_name
-----+-----
 2182738256 | docs_lab_target
 3410995529 | docs_lab_provider
(2 rows)
```

您可以使用下列查詢，取得有關訂閱的詳細資訊。

```
SELECT sub_name,sub_slot_name,sub_target
FROM pglogical.subscription;
```

```

sub_name |          sub_slot_name          | sub_target
-----+-----+-----
docs_lab_subscription | pgl_labdb_docs_labcb4fa94_docs_lab3de412c | 2182738256
(1 row)

```

您現在可以捨棄訂閱，如下所示。

```

SELECT pglogical.drop_subscription(subscription_name := 'docs_lab_subscription');
drop_subscription
-----
                1
(1 row)

```

捨棄訂閱之後，您可以刪除節點。

```

SELECT pglogical.drop_node(node_name := 'docs-lab-subscriber');
drop_node
-----
t
(1 row)

```

您可以驗證節點是否不再存在，如下所示。

```

SELECT * FROM pglogical.node;
node_id | node_name
-----+-----
(0 rows)

```

pglogical 延伸模組的參數參考

在資料表中，您可以尋找與 pglogical 延伸模組相關聯的參數。pglogical.conflict_log_level 和 pglogical.conflict_resolution 這類參數用來處理更新衝突。對從發佈者訂閱變更的相同資料表進行本機變更時，可能會出現衝突。在各種案例期間 (例如雙向複寫或從相同發佈者複寫多個訂閱者時)，也會發生衝突。如需詳細資訊，請參閱[使用 pglogical 進行 PostgreSQL 雙向複寫](#)。

參數	描述
pglogical.batch_inserts	可能情況下批次插入。依預設不會設定。變更為 '1' 以開啟、變更為 '0' 以關閉。

參數	描述
pglogical.conflict_log_level	設定用於記錄已解決衝突的日誌層級。支援的字串值為 debug5、debug4、debug3、debug2、debug1、info、notice、warning、error、log、fatal、panic。
pglogical.conflict_resolution	設定用來在衝突可解決時解決衝突的方法。支援的字串值為 error、apply_remote、keep_local、last_update_wins、first_update_wins。
pglogical.extra_connection_options	要新增到所有對等節點連線的連線選項。
pglogical.synchronous_commit	pglogical 特定的同步遞交值
pglogical.use_spi	使用 SPI (伺服器程式設計介面) 而不是低階 API 來套用變更。設為 '1' 以開啟、設為 '0' 以關閉。如需 SPI 的詳細資訊，請參閱 PostgreSQL 文件中的 Server Programming Interface (伺服器程式設計介面)。

使用 pgactive 來支援主動-主動式複寫

pgactive 延伸模組使用主動-主動式複寫來支援和協調多個 RDS for PostgreSQL 資料庫上的寫入操作。Amazon RDS for PostgreSQL 下版本的pgactive擴充功能：

- 適用於 PostgreSQL 更高版本的 RDS
- 適用 PostgreSQL 及更高版本的 RDS
- RDS 適用於 PostgreSQL 14.10 及更高版本
- 適用於 PostgreSQL 及更高版本的 RDS
- RDS 適用於 PostgreSQL 12.17 及更高版本
- 適用於 PostgreSQL 的 RDS

Note

當複寫組態中的多個資料庫上有寫入操作時，可能會發生衝突。如需更多資訊，請參閱[處理主動-主動式複寫中的衝突](#)

主題

- [初始化 pgactive 延伸模組功能](#)
- [針對 RDS for PostgreSQL 資料庫執行個體設定主動-主動式複寫](#)
- [處理主動-主動式複寫中的衝突](#)
- [處理主動-主動式複寫中的序列](#)
- [pgactive 延伸模組的參數參考](#)
- [測量 pgactive 成員之間的複製延遲](#)
- [pgactive 延伸模組的現制](#)

初始化 pgactive 延伸模組功能

若要在 RDS for PostgreSQL 資料庫執行個體上初始化 pgactive 延伸模組，請將 `rds.enable_pgactive` 參數值設定為 1，然後在資料庫中建立延伸模組。這樣做就會自動開啟參數 `rds.logical_replication` 和 `track_commit_timestamp`，並將 `wal_level` 的值設定為 `logical`。

您必須具有做為 `rds_superuser` 角色的許可，才能執行這些任務。

您可以使用 AWS Management Console 或 AWS CLI 來建立所需的 RDS 資 PostgreSQL 執行個體。下列步驟假設您的 RDS for PostgreSQL 資料庫執行個體與自訂資料庫參數群組相關聯。如需建立自訂資料庫參數群組的相關資訊，請參閱 [使用參數群組](#)。

主控台

若要初始化 `pgactive` 延伸模組功能

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇您的 RDS for PostgreSQL 資料庫執行個體。
3. 針對您的 RDS for PostgreSQL 資料庫執行個體開啟組態索引標籤。在執行個體詳細資訊中，找到資料庫執行個體參數群組連結。
4. 選擇連結以開啟與您的 RDS for PostgreSQL 資料庫執行個體相關聯的自訂參數。
5. 找到 `rds.enable_pgactive` 參數，並將其設定為 1 以初始化 `pgactive` 功能。
6. 選擇儲存變更。
7. 從 Amazon RDS 主控台的導覽窗格中，選擇資料庫。
8. 選取您的 RDS for PostgreSQL 資料庫執行個體，然後從動作選單中選擇重新開機。
9. 確認資料庫執行個體重新開機，以讓您的變更生效。
10. 當資料庫執行個體可用時，您可以使用 `psql` 或任何其他 PostgreSQL 用戶端連線至 RDS for PostgreSQL 資料庫執行個體。

下列範例假設您的 RDS for PostgreSQL 資料庫執行個體擁有名為 `postgres` 的預設資料庫。

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master_username --password --dbname=postgres
```

11. 若要驗證 `pgactive` 是否已初始化，請執行下列命令。

```
postgres=>SELECT setting ~ 'pgactive'
FROM pg_catalog.pg_settings
WHERE name = 'shared_preload_libraries';
```

如果 `pgactive` 在 `shared_preload_libraries` 中，則上述命令將傳回以下內容：

```
?column?  
-----  
t
```

12. 建立延伸模組，如下所示。

```
postgres=> CREATE EXTENSION pgactive;
```

AWS CLI

若要初始化 pgactive 延伸模組功能

若要pgactive使用初始化 AWS CLI，請呼叫[修改-db-參數群組作業來修改自訂參數群組](#)中的某些參數，如下列程序所示。

1. 使用下列 AWS CLI 命令將設定rds.enable_pgactive1為初始化適用於 PostgreSQL 資料庫執行個體的pgactive功能。

```
postgres=>aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=rds.enable_pgactive,ParameterValue=1,ApplyMethod=pending-reboot" \  
  --region aws-region
```

2. 使用下列 AWS CLI 命令將 RDS for PostgreSQL 資料庫執行個體重新開機，以便初始化程式pgactive庫。

```
aws rds reboot-db-instance \  
  --db-instance-identifier your-instance \  
  --region aws-region
```

3. 當執行個體可用時，請使用 psql 連線至 RDS for PostgreSQL 資料庫執行個體。。

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=master user --password --dbname=postgres
```

4. 建立延伸模組，如下所示。

```
postgres=> CREATE EXTENSION pgactive;
```

針對 RDS for PostgreSQL 資料庫執行個體設定主動-主動式複寫

下列程序說明如何在相同區域中，於兩個執行 PostgreSQL 15.4 或更新版本的 RDS for PostgreSQL 資料庫執行個體之間啟動主動-主動式複寫。若要執行多區域高可用性範例，您需要在兩個不同的區域部署 Amazon RDS for PostgreSQL 執行個體，並設定 VPC 對等互連。如需詳細資訊，請參閱 [VPC 對等互連](#)。

Note

在多個地區之間傳送流量可能會產生額外費用。

這些步驟假設 RDS for PostgreSQL 資料庫執行個體已使用 `pgactive` 延伸模組設定完成。如需詳細資訊，請參閱 [初始化 `pgactive` 延伸模組功能](#)。

若要設定第一個具有 `pgactive` 延伸模組的 RDS for PostgreSQL 資料庫執行個體

下列範例說明如何建立 `pgactive` 群組，以及在 RDS for PostgreSQL 資料庫執行個體上建立 `pgactive` 延伸模組所需的其他步驟。

1. 使用 `psql` 或其他用戶端工具連線至您的第一個 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=firstinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --username=master username --password --dbname=postgres
```

2. 使用下列命令在 RDS for PostgreSQL 執行個體上建立資料庫：

```
postgres=> CREATE DATABASE app;
```

3. 使用下列命令將連線切換至新資料庫：

```
\c app
```

4. 若要檢查 `shared_preload_libraries` 參數是否包含 `pgactive`，請執行下列命令：

```
app=>SELECT setting ~ 'pgactive' FROM pg_catalog.pg_settings WHERE name = 'shared_preload_libraries';
```

```
?column?
```

```
-----
t
```

5. 使用下列 SQL 陳述式建立並填入範例資料表：

a. 使用下列 SQL 陳述式建立範例資料表。

```
app=> CREATE SCHEMA inventory;
CREATE TABLE inventory.products (
  id int PRIMARY KEY, product_name text NOT NULL,
  created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP);
```

b. 使用下列 SQL 陳述式在資料表中填入一些範例資料。

```
app=> INSERT INTO inventory.products (id, product_name)
VALUES (1, 'soap'), (2, 'shampoo'), (3, 'conditioner');
```

c. 使用下列 SQL 陳述式，驗證資料是否存在於資料表中。

```
app=>SELECT count(*) FROM inventory.products;

count
-----
3
```

6. 在現有資料庫上建立 pgactive 延伸模組。

```
app=> CREATE EXTENSION pgactive;
```

7. 使用下列命令建立並初始化 pgactive 群組：

```
app=> SELECT pgactive.pgactive_create_group(
  node_name := 'node1-app',
  node_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD');
```

node1-app 是您指派的名稱，用於單獨識別 pgactive 群組中的節點。

Note

若要在可公開存取的資料庫執行個體上成功執行此步驟，您必須將 `rds.custom_dns_resolution` 參數設定為 1 以將它開啟。

- 若要檢查資料庫執行個體是否已就緒，請使用下列命令：

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

如果命令成功，您會看到以下輸出內容：

```
pgactive_wait_for_node_ready
-----
(1 row)
```

若要設定第二個 RDS for PostgreSQL 執行個體，並將其加入 **pgactive** 群組

下列範例說明如何建立將 RDS for PostgreSQL 資料庫執行個體加入 `pgactive` 群組，以及在資料庫執行個體上建立 `pgactive` 延伸模組所需的其他步驟。

這些步驟假設已有另一個 RDS for PostgreSQL 資料庫執行個體使用 `pgactive` 延伸模組設定完成。如需詳細資訊，請參閱 [初始化 pgactive 延伸模組功能](#)。

- 使用 `psql` 來連線至您想要從發佈者接收更新的執行個體。

```
psql --host=secondinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master username --password --dbname=postgres
```

- 使用下列命令在第二個 RDS for PostgreSQL 資料庫執行個體上建立資料庫：

```
postgres=> CREATE DATABASE app;
```

- 使用下列命令將連線切換至新資料庫：

```
\c app
```

- 在現有資料庫上建立 `pgactive` 延伸模組。

```
app=> CREATE EXTENSION pgactive;
```

5. 將 RDS for PostgreSQL 第二個資料庫執行個體加入 pgactive 群組，如下所示。

```
app=> SELECT pgactive.pgactive_join_group(
node_name := 'node2-app',
node_dsn := 'dbname=app host=secondinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD',
join_using_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

node2-app 是您指派的名稱，用於單獨識別 pgactive 群組中的節點。

6. 若要檢查資料庫執行個體是否已就緒，請使用下列命令：

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

如果命令成功，您會看到以下輸出內容：

```
pgactive_wait_for_node_ready
-----
(1 row)
```

如果第一個 RDS for PostgreSQL 資料庫相對較大，您會看到 pgactive.pgactive_wait_for_node_ready() 發出還原操作的進度報告。輸出結果類似如下：

```
NOTICE: restoring database 'app', 6% of 7483 MB complete
NOTICE: restoring database 'app', 42% of 7483 MB complete
NOTICE: restoring database 'app', 77% of 7483 MB complete
NOTICE: restoring database 'app', 98% of 7483 MB complete
NOTICE: successfully restored database 'app' from node node1-app in
00:04:12.274956
pgactive_wait_for_node_ready
-----
(1 row)
```

從這裡開始，pgactive 會在兩個資料庫執行個體之間同步資料。

7. 您可以使用下列命令來驗證第二個資料庫執行個體的資料庫是否有資料：

```
app=> SELECT count(*) FROM inventory.products;
```

如果資料已成功同步，您會看到下列輸出內容：

```
count
-----
3
```

8. 執行下列命令以插入新值：

```
app=> INSERT INTO inventory.products (id, product_name) VALUES ('lotion');
```

9. 連線至第一個資料庫執行個體的資料庫，然後執行下列查詢：

```
app=> SELECT count(*) FROM inventory.products;
```

如果主動-主動式複寫已初始化，則會輸出類似下列內容：

```
count
-----
4
```

從 **pgactive** 群組卸離並移除資料庫執行個體

您可以利用下列步驟將資料庫執行個體從 **pgactive** 群組卸離並移除：

1. 您可以使用下列命令將第二個資料庫執行個體從第一個資料庫執行個體卸離：

```
app=> SELECT * FROM pgactive.pgactive_detach_nodes(ARRAY['node2-app']);
```

2. 使用下列命令從第二個資料庫執行個體移除 **pgactive** 延伸模組：

```
app=> SELECT * FROM pgactive.pgactive_remove();
```

若要強制移除延伸模組：


```
app=> SELECT * FROM pgactive.pgactive_remove(true);
```

3. 使用以下命令刪除延伸模組：

```
app=> DROP EXTENSION pgactive;
```

處理主動-主動式複寫中的衝突

pgactive 延伸模組是在每個資料庫上運作，而不是每個叢集。使用 pgactive 的每個資料庫執行個體都是獨立的執行個體，可接受任何來源的資料變更。將變更傳送至資料庫執行個體時，PostgreSQL 會在本機上遞交該變更，然後使用 pgactive 以非同步方式將變更複寫到其他資料庫執行個體。當兩個 PostgreSQL 資料庫執行個體幾乎同時更新相同的記錄時，可能會發生衝突。

pgactive 延伸模組提供了衝突偵測和自動解決的機制。它會追蹤交易在兩個資料庫執行個體上得到認可的時間戳記，並自動套用具有最新時間戳記的變更。pgactive 延伸模組也會記錄 pgactive.pgactive_conflict_history 資料表中發生的衝突。

pgactive.pgactive_conflict_history 將繼續增長。您可能想要定義清除原則。這可以通過刪除一些記錄定期或定義此關係的分區方案（以及後來分離，刪除，截斷感興趣的分區）來完成。若要定期實作清除原則，其中一個選項是使用 pg_cron 擴充功能。請參閱下列 [pg_cron 歷史記錄資料表範例](#) 的資訊，使用 [PostgreSQL pg_cron 擴充功能排程維護](#)。

處理主動-主動式複寫中的序列

具有 pgactive 延伸模組的 RDS for PostgreSQL 資料庫執行個體使用兩種不同的序列機制來產生唯一值。

全域序列

若要使用全域序列，請使用 CREATE SEQUENCE 陳述式建立本機序列。不要使用 using nextval(seqname)，而是使用 pgactive.pgactive_snowflake_id_nextval(seqname) 來取得序列中的下一個唯一值。

下列範例會建立全域序列：

```
postgres=> CREATE TABLE gstest (  
    id bigint primary key,  
    parrot text
```

```
);
```

```
postgres=>CREATE SEQUENCE gstest_id_seq OWNED BY gstest.id;
```

```
postgres=> ALTER TABLE gstest \  
    ALTER COLUMN id SET DEFAULT \  
    pgactive.pgactive_snowflake_id_nextval('gstest_id_seq');
```

分割序列

在拆分步驟或分割序列中，每個節點上會使用標準 PostgreSQL 序列。每個序列會以相同的量遞增，並從不同的偏移量開始。例如，若是步進 100，節點 1 會產生序列為 101、201、301，依此類推，而節點 2 會產生序列為 102、202、302，依此類推。即使節點無法長時間通訊，此結構仍能正常運作，但是設計人員必須在建立結構描述時指定最大節點數，並且需要每個節點的組態。若發生錯誤，便容易導致序列重疊。

透過在節點上建立所需的序列來對 pgactive 設定此方法相對較為簡單，如下所示：

```
CREATE TABLE some_table (generated_value bigint primary key);
```

```
postgres=> CREATE SEQUENCE some_seq INCREMENT 100 OWNED BY some_table.generated_value;
```

```
postgres=> ALTER TABLE some_table ALTER COLUMN generated_value SET DEFAULT  
    nextval('some_seq');
```

然後在每個節點上呼叫，以提供不同的偏移量起始值，如下所示。

```
postgres=>  
-- On node 1  
SELECT setval('some_seq', 1);  
  
-- On node 2  
SELECT setval('some_seq', 2);
```

pgactive 延伸模組的參數參考

您可以使用下列查詢來檢視與 pgactive 延伸模組相關聯的所有參數。

```
postgres=> SELECT * FROM pg_settings WHERE name LIKE 'pgactive.%';
```

測量 pgactive 成員之間的複製延遲

您可以使用下列查詢來檢視成pgactive員之間的複寫延遲。在每個pgactive節點上執行此查詢以取得完整圖片。

```
postgres=# SELECT *, (last_applied_xact_at - last_applied_xact_committs) AS lag
FROM pgactive.pgactive_node_slots;
-[ RECORD 1 ]-----
+-----+
node_name          | node2-app
slot_name          | pgactive_5_7332551165694385385_0_5__
slot_restart_lsn   | 0/1A898A8
slot_confirmed_lsn | 0/1A898E0
walsender_active   | t
walsender_pid      | 69022
sent_lsn           | 0/1A898E0
write_lsn          | 0/1A898E0
flush_lsn          | 0/1A898E0
replay_lsn         | 0/1A898E0
last_sent_xact_id  | 746
last_sent_xact_committs | 2024-02-06 18:04:22.430376+00
last_sent_xact_at  | 2024-02-06 18:04:22.431359+00
last_applied_xact_id | 746
last_applied_xact_committs | 2024-02-06 18:04:22.430376+00
last_applied_xact_at  | 2024-02-06 18:04:52.452465+00
lag                | 00:00:30.022089
```

pgactive 延伸模組的現制

- 所有資料表都需有主索引鍵，否則不允許更新和刪除操作。主索引鍵欄中的值不應更新。
- 序列可能有間隙，有時可能不按順序。序列未複寫。如需詳細資訊，請參閱 [處理主動-主動式複寫中的序列](#)。
- DDL 和大型物件未複寫。
- 次要唯一索引可能會導致資料差異。
- 群組中所有節點的定序都必須相同。
- 節點之間的負載平衡是一種反模式。

- 大型交易可能造成複寫延遲。

使用 pg_repack 擴充功能減少資料表和索引膨脹

您可以使用pg_repack擴充功能從資料表和索引中移除膨脹，做為的替代方VACUUM FULL案。RDS for PostgreSQL 9.6.3 及更新版本支援此擴充功能。如需有關pg_repack擴充功能和完整表格重新封裝的詳細資訊，請參閱[GitHub 專案文件](#)。

與此不同的是VACUUM FULL，在下列情況下，pg_repack擴充功能只需要在資料表重新建置作業期間短時間內執行獨佔AccessExclusive鎖定 (鎖定)：

- 記錄表格的初始建立 — 建立記錄表格來記錄資料初始複製期間發生的變更，如下列範例所示：

```
postgres=>\dt+ repack.log_*
List of relations
-[ RECORD 1 ]-+-----
Schema          | repack
Name            | log_16490
Type           | table
Owner          | postgres
Persistence    | permanent
Access method  | heap
Size           | 65 MB
Description    |
```

- 最後 swap-and-drop 階段。

對於其餘的重建作業，它只需要ACCESS SHARE鎖定原始資料表，即可將資料列從其複製到新資料表。這有助於插入，更新和刪除操作像往常一樣進行。

建議

當您使用pg_repack擴充功能從資料表和索引中移除膨脹時，下列建議適用：

- 在非上班時間執行重新封裝，或透過維護時段執行重新封裝，將其對其他資料庫活動的效能影響降到最低。
- 密切監控重建活動期間的封鎖工作階段，並確保原始資料表上沒有可能封鎖的活動pg_repack，特別是在需要原始資料表獨佔鎖定的最後 swap-and-drop 階段時。如需詳細資訊，請參閱[識別封鎖查詢的內容](#)。

當您看到阻塞會話時，您可以在仔細考慮後使用以下命令終止它。這有助於繼pg_repack續完成重建：

```
SELECT pg_terminate_backend(pid);
```

- 在交易速率非常高的系統上套用pg_repack's日誌表格中的應計變更時，套用處理作業可能無法跟上變更率。在這種情況下，pg_repack將無法完成套用處理作業。如需詳細資訊，請參閱 [在重新裝箱期間監控新表](#)。如果索引嚴重膨脹，另一種解決方案是僅執行索引重新包裝。這也有助於VACUUM 的索引清理週期更快地完成。

您可以使用 PostgreSQL 第 12 版的手動真空來跳過索引清理階段，並且在 PostgreSQL 第 14 版的緊急自動真空期間會自動跳過索引清理階段。這有助於真空更快地完成，而不會消除指數膨脹，並且僅適用於緊急情況，例如防止環繞式真空。如需詳細資訊，請參閱 Amazon Aurora 使用者指南中的 [避免索引膨脹](#)。

先決條件

- 該表必須有主鍵或非空唯一約束。
- 用戶端和伺服器的延伸版本必須相同。
- 確保 RDS 執行個體的大小FreeStorageSpace超過資料表的總大小，而不會膨脹。作為一個例子，考慮包括 TOAST 和索引在內的表的總大小為 2TB，並在表中的總膨脹為 1TB。FreeStorageSpace必要的值必須大於以下計算所傳回的值：

$$2\text{TB (Table size)} - 1\text{TB (Table bloat)} = 1\text{TB}$$

您可以使用下面的查詢來檢查表的總大小，並使用pgstattuple來導出膨脹。如需詳細資訊，請參閱 Amazon Aurora 使用者 [指南中的診斷表格和索引膨脹](#)

```
SELECT pg_size_pretty(pg_total_relation_size('table_name')) AS total_table_size;
```

這個空間是活動完成後回收的。

- 確定 RDS 執行個體具有足夠的運算和 IO 容量來處理重新封裝作業。您可以考慮擴展實例類別以獲得最佳性能平衡。

若要使用pg_repack擴充功能

- 執行以下命令，在 RDS for PostgreSQL 資料庫執行個體上安裝 pg_repack 擴充功能。

```
CREATE EXTENSION pg_repack;
```

- 執行下列命令，以授與寫入存取權給由建立的暫存記錄資料表pg_repack。

```
ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT INSERT ON TABLES TO PUBLIC;  
ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT USAGE, SELECT ON SEQUENCES TO  
PUBLIC;
```

- 使用pg_repack用戶端公用程式 Connect 至資料庫。使用具有 rds_superuser 權限的帳戶。舉例來說，假設 rds_test 角色具有 rds_superuser 權限。下列語法會pg_repack針對包含資料庫中所有資料表索引的完整postgres資料表執行。

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test  
-k postgres
```

Note

您必須使用-k 選項進行連線。不支援 -a 選項。

pg_repack用戶端的回應會提供重新封裝之資料庫執行個體上資料表的相關資訊。

```
INFO: repacking table "pgbench_tellers"  
INFO: repacking table "pgbench_accounts"  
INFO: repacking table "pgbench_branches"
```

- 下面的語法重新包orders括postgres數據庫中的索引的單個表。

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test  
--table orders -k postgres
```

下列語法只會重新填寫postgres資料庫中資料orders表的索引。

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test  
--table orders --only-indexes -k postgres
```

在重新裝箱期間監控新表

- 數據庫的大小增加了表的總大小減去膨脹，直到重新包裝的 swap-and-drop 階段。您可以監視資料庫大小的成長率、計算重新封裝的速度，並大致估計完成初始資料傳輸所需的時間。

作為一個例子，考慮表的總大小為 2TB，數據庫的大小為 4TB，並在表中的總膨脹為 1TB。重新封裝作業結束時計算所傳回的資料庫總大小值如下：

$$2\text{TB (Table size)} + 4\text{ TB (Database size)} - 1\text{TB (Table bloat)} = 5\text{TB}$$

您可以透過取樣兩個時間點之間的成長率 (以位元組為單位)，大致估計重新封裝作業的速度。如果成長率是每分鐘 1GB，則大約需要 1000 分鐘或 16.6 小時才能完成初始資料表建置作業。除了初始表構建之外，pg_repack還需要應用累積的更改。所花費的時間取決於套用持續變更的速率以及累積變更。

Note

您可以使用pgstattuple擴展來計算表中的膨脹。如需更多詳細資訊，請參閱 [pgstattuple](#)。

- 重新封裝結構描述下的pg_repack's記錄資料表中的資料列數目代表初始載入後要套用至新資料表的擱置變更數量。

您可以簽入pg_repack's記錄表格，pg_stat_all_tables以監視套用至新表格的變更。

pg_stat_all_tables.n_live_tup指示等待套用至新資料表的記錄數目。如需詳細資訊，請參閱[資料表](#)。

```
postgres=>SELECT relname,n_live_tup FROM pg_stat_all_tables WHERE schemaname =
'repack' AND relname ILIKE '%log%';
```

```
-[ RECORD 1 ]-----
relname      | log_16490
n_live_tup   | 2000000
```

- 您可以使用pg_stat_statements擴充功能來找出重新封裝作業中每個步驟所花費的時間。這有助於準備在生產環境中套用相同的重新裝箱作業。您可以調整LIMIT條款以進一步擴展輸出。

```
postgres=>SELECT
    SUBSTR(query, 1, 100) query,
    round((round(total_exec_time::numeric, 6) / 1000 / 60),4)
total_exec_time_in_minutes
FROM
    pg_stat_statements
WHERE
```



```

query ILIKE '%repack%'
ORDER BY
total_exec_time DESC LIMIT 5;

query |
total_exec_time_in_minutes |
-----+-----
CREATE UNIQUE INDEX index_16493 ON repack.table_16490 USING btree (a) |
6.8627 |
INSERT INTO repack.table_16490 SELECT a FROM ONLY public.t1 |
6.4150 |
SELECT repack.repack_apply($1, $2, $3, $4, $5, $6) |
0.5395 |
SELECT repack.repack_drop($1, $2) |
0.0004 |
SELECT repack.repack_swap($1) |
0.0004 |
(5 rows)

```

重新包裝完全是一項 out-of-place 操作，因此原始資料表不會受到影響，而且我們預計不會有任何需要復原原始資料表的未預期挑戰。如果重新裝箱意外失敗，您必須檢查錯誤的原因並加以解決。

解決問題之後，請在資料表所在的資料庫中卸除並重新建立 `pg_repack` 擴充功能，然後重試該 `pg_repack` 步驟。此外，計算資源的可用性和表格的並行存取性在重新封裝作業的及時完成中扮演著至關重要的角色。

升級和使用 PLV8 擴充功能

PLV8 是值得信賴的 JavaScript 語言擴充功能，適用於 PostgreSQL。您可用於存放的程序、觸發程序和可從 SQL 呼叫的其他程序性程式碼。所有目前版本的 PostgreSQL 都支援此語言擴充功能。

如果您使用 [PLV8](#) 並將 PostgreSQL 升級至新的 PLV8 版本，即可立即利用新的擴充功能套件。執行下列步驟，同步目錄中繼資料與新版的 PLV8。這些步驟為選用，但強烈建議您完成步驟以避免中繼資料不符的警告。

升級程序會捨棄您現有的所有 PLV8 函數。因此，我們建議您在升級之前建立 RDS for PostgreSQL 資料庫執行個體的快照。如需詳細資訊，請參閱 [為單一可用區資料庫執行個體建立資料庫快照](#)。

如何同步目錄中繼資料與新版 PLV8

1. 確認您需要更新。若要執行此動作，請在已連接至執行個體的情況下，執行下列命令。

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

如果結果包含一個已安裝之版本的值，而此版本的編號小於預設版本，則請繼續此程序來更新擴充功能套件。例如，下列結果集表示您應該更新。

```
name      | default_version | installed_version |          comment
-----+-----+-----+-----
+-----+-----+-----+-----
plls      | 2.1.0           | 1.5.3             | PL/LiveScript (v8) trusted
procedural language
plcoffee | 2.1.0           | 1.5.3             | PL/CoffeeScript (v8) trusted
procedural language
plv8      | 2.1.0           | 1.5.3             | PL/JavaScript (v8) trusted
procedural language
(3 rows)
```

2. 如果您尚未建立 RDS for PostgreSQL 資料庫執行個體的快照，請建立一個。建立快照時，您可以繼續執行下列步驟。
3. 取得資料庫執行個體中的 PLV8 函數計數，以驗證升級之後函數全部存在。例如，下列 SQL 查詢會傳回 PLV8、plcoffee 和 plls 中寫入的函數數目。

```
SELECT proname, nsprname, lanname
FROM pg_proc p, pg_language l, pg_namespace n
WHERE p.prolang = l.oid
```

```
AND n.oid = p.pronamespace
AND lanname IN ('plv8','plcoffee','p1ls');
```

4. 使用 `pg_dump` 建立只含結構描述的傾印檔案。例如，在 `/tmp` 目錄中建立用戶端機器上的檔案。

```
./pg_dump -Fc --schema-only -U master postgres >/tmp/test.dmp
```

此範例使用下列選項：

- `-Fc` – 自訂格式
- `--schema-only` – 僅傾印建立結構描述所需的命令 (我們案例中的函數)
- `-U` – RDS 主要使用者名稱
- `database` – 資料庫執行個體上的資料庫名稱

如需有關 `pg_dump` 的詳細資訊，請參閱 PostgreSQL 文件中的 [pg_dump](#)。

5. 擷取傾印檔案中存在的「CREATE FUNCTION」DDL 陳述式。下列範例使用 `grep` 命令擷取用於建立函數並儲存到檔案中的 DDL 陳述式。您將於後續步驟中使用此 `ddl` 來重新建立函數。

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list/
```

如需有關 `pg_restore` 的詳細資訊，請參閱 PostgreSQL 文件中的 [pg_restore](#)。

6. 捨棄函數和擴充功能。下列範例會捨棄任何以 PLV8 為基礎的物件。`cascade` 選項可確保捨棄任何相依物件。

```
DROP EXTENSION plv8 CASCADE;
```

如果 PostgreSQL 執行個體包含以 `plcoffee` 或 `p1ls` 為基礎的物件，請對這些擴充功能重複此步驟。

7. 建立擴充功能。下列範例會建立 `plv8`、`plcoffee` 和 `p1ls` 擴充功能。

```
CREATE EXTENSION plv8;
CREATE EXTENSION plcoffee;
CREATE EXTENSION p1ls;
```

8. 使用傾印檔案和「驅動程式」檔案建立函數。

下列範例會重新建立您先前擷取的函數。

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. 使用以下查詢驗證是否已重新建立所有函數。

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

PLV8 第 2 版會將以下額外的資料列新增至結果集：

```

proname      | nspname      | lanname
-----+-----+-----
plv8_version | pg_catalog   | plv8

```

使用 PL/Rust 以 Rust 語言撰寫 PostgreSQL 函數

PL/Rust 是 PostgreSQL 的受信任 Rust 語言延伸模組。您可以將其用於預存程序、函數，以及可從 SQL 呼叫的其他程序性程式碼。下列版本提供 PL/Rust 語言延伸模組：

- 適用於 PostgreSQL 更高版本的 RDS
- RDS for PostgreSQL 15.2-R2 及更高的 15 版本
- RDS for PostgreSQL 14.9 及更高的 14 版本
- RDS for PostgreSQL 13.12 及更高的 13 版本

如需詳細資訊，請參閱上的 [PL/Rust](#)。GitHub

主題

- [設定 PL/Rust](#)
- [使用 PL/Rust 建立函數](#)
- [使用套件搭配 PL/Rust](#)
- [PL/Rust 限制](#)

設定 PL/Rust

若要在資料庫執行個體上安裝 plrust 延伸模組，請將 plrust 新增至與資料庫執行個體相關聯之資料庫參數群組中的 `shared_preload_libraries` 參數。安裝 plrust 延伸模組後，您可以建立函數。

若要修改 `shared_preload_libraries` 參數，您的資料庫執行個體必須與自訂參數群組相關聯。如需建立自訂資料庫參數群組的相關資訊，請參閱 [使用參數群組](#)。

您可以使用 AWS Management Console 或安裝擴充套件。AWS CLI

下列步驟假設您的資料庫執行個體與自訂資料庫參數群組相關聯。

主控台

在 `shared_preload_libraries` 參數中安裝 `plrust` 延伸模組

使用屬於 `rds_superuser` 群組 (角色) 成員的帳戶完成下列步驟。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇資料庫執行個體的名稱以顯示其詳細資訊。
4. 開啟資料庫執行個體的組態索引標籤，然後尋找資料庫執行個體參數群組連結。
5. 選擇連結以開啟與資料庫執行個體相關聯的自訂參數。
6. 在 Parameters (參數) 搜尋欄位中，輸入 `shared_pre` 以尋找 `shared_preload_libraries` 參數。
7. 選擇 Edit parameters (編輯參數) 以存取屬性值。
8. 在值欄位中，將 `plrust` 新增至清單。使用逗號區隔值清單中的項目。
9. 重新啟動資料庫執行個體，以便您對 `shared_preload_libraries` 參數的變更生效。初始重新啟動可能需要額外的時間才能完成。
10. 當執行個體可用時，請驗證 `plrust` 是否已初始化。使用 `psql` 連線至資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;
```

您的輸出應該類似以下內容：

```
shared_preload_libraries
-----
rdsutils,plrust
(1 row)
```

AWS CLI

在 `shared_preload_libraries` 參數中安裝 `plrust` 延伸模組

使用屬於 `rds_superuser` 群組 (角色) 成員的帳戶完成下列步驟。

1. 使用「[修改-db-參數群組](#)」 [AWS CLI 指令將 plrust 加入至參數](#)。 `shared_preload_libraries`

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=shared_preload_libraries,ParameterValue=plrust,ApplyMethod=pending-  
reboot" \  
  --region aws-region
```

2. 使用[重新啟動 db-instance AWS CLI 指令重新啟動資料庫執行個體](#)，並初始化 `plrust` 程式庫。初始重新啟動可能需要額外的時間才能完成。

```
aws rds reboot-db-instance \  
  --db-instance-identifier your-instance \  
  --region aws-region
```

3. 當執行個體可用時，您可以驗證 `plrust` 是否已初始化。使用 `psql` 連線至資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;
```

您的輸出應該類似以下內容：

```
shared_preload_libraries  
-----  
rdsutils,plrust  
(1 row)
```

使用 PL/Rust 建立函數

PL/Rust 會將函數編譯為動態程式庫、載入它，然後執行它。

以下 Rust 函數會從陣列中篩選出倍數。

```
postgres=> CREATE LANGUAGE plrust;
```

CREATE EXTENSION

```
CREATE OR REPLACE FUNCTION filter_multiples(a BIGINT[], multiple BIGINT) RETURNS
BIGINT[]
    IMMUTABLE STRICT
    LANGUAGE PLRUST AS
$$
    Ok(Some(a.into_iter().filter(|x| x.unwrap() % multiple != 0).collect()))
$$;

WITH gen_values AS (
SELECT ARRAY(SELECT * FROM generate_series(1,100)) as arr)
SELECT filter_multiples(arr, 3)
from gen_values;
```

使用套件搭配 PL/Rust

在適用於 PostgreSQL 的 RDS 版本 16.3-R2 及更高版本、15.7-R2 及更高版本 15 個版本、14.12-R2 及更高版本 14 版本，以及 13 版及更高版本中，PL/RUST 支援額外的包裝箱：

- url
- regex
- serde
- serde_json

在適用於 PostgreSQL 的 RDS 版本 15.5-R2 及更高版本、14.10-R2 及更高版本 14 個版本，以及 13 版和更高版本中，PL/RUST 支援兩個額外的包裝箱：

- roaring-rs
- num-bigint

從 Amazon RDS for PostgreSQL 版本 15.4、14.9 和 13.12 版開始，PL/ 鐵鏽支援下列包裝箱：

- aes
- ctr
- rand

這些套件僅支援預設功能。新的 RDS for PostgreSQL 版本可能包含更新的套件版本，而且可能不再支援較舊的套件版本。

請遵循執行主要版本升級的最佳實務來進行測試，了解您的 PL/Rust 函數是否與新的主要版本相容。如需詳細資訊，請參閱部落格 [Best practices for upgrading Amazon RDS to major and minor versions of PostgreSQL](#) (將 Amazon RDS 升級至 PostgreSQL 的主要和次要版本的最佳實務)，以及《Amazon RDS 使用者指南》中的 [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。

[使用相依性](#) 中提供了建立 PL/Rust 函數時使用相依性的範例。

PL/Rust 限制

根據預設，資料庫使用者無法使用 PL/Rust。若要提供 PL/Rust 的存取權，請以具有 `rds_superuser` 權限的使用者身分連線，然後執行下列命令：

```
postgres=> GRANT USAGE ON LANGUAGE PLRUST TO user;
```


使用 PostGIS 擴充功能管理空間資料

PostGIS 是 PostgreSQL 的擴充功能，可用於儲存和管理空間資訊。若要進一步了解 PostGIS，請參閱 [PostGIS.net](https://postgis.net)。

從 10.5 版開始，PostgreSQL 即支援 PostGIS 用於處理 Mapbox 向量圖標資料的 libprotobuf 1.3.0 程式庫。

設定 PostGIS 擴充功能需要 `rds_superuser` 權限。我們建議您建立一使用者 (角色) 來管理 PostGIS 擴充功能及您的空間資料。PostGIS 擴充功能及其相關元件會為 PostgreSQL 新增數千個函數。若這對您的使用案例有意義，請考慮在其自己的結構描述中建立 PostGIS 擴充功能。下列範例會顯示如何在其自己的資料庫中安裝擴充功能，但這並非必要。

主題

- [步驟 1：建立使用者 \(角色\) 來管理 PostGIS 擴充功能](#)
- [步驟 2：載入 PostGIS 擴充功能](#)
- [步驟 3：轉移擴充功能的所有權](#)
- [步驟 4：轉移 PostGIS 物件的所有權](#)
- [步驟 5：測試擴充功能](#)
- [步驟 6：升級 PostGIS 擴充功能](#)
- [PostGIS 擴充功能版本](#)
- [將 PostGIS 2 升級到 PostGIS 3](#)

步驟 1：建立使用者 (角色) 來管理 PostGIS 擴充功能

首先，以具有 `rds_superuser` 權限的使用者身分連線至您的 RDS for PostgreSQL 資料庫執行個體。若您在設定執行個體時保留預設名稱，則以 `postgres` 身分連線：

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres  
--password
```

建立一個單獨的角色 (使用者) 來管理 PostGIS 擴充功能。

```
postgres=> CREATE ROLE gis_admin LOGIN PASSWORD 'change_me';  
CREATE ROLE
```

授予此角色 `rds_superuser` 權限，允許角色安裝擴充功能。

```
postgres=> GRANT rds_superuser TO gis_admin;  
GRANT
```

建立用於 PostGIS 成品的資料庫：此為選擇性步驟。或者，您可以在 PostGIS 擴充功能的使用者資料庫中建立結構描述，但這也並非必要。

```
postgres=> CREATE DATABASE lab_gis;  
CREATE DATABASE
```

授予 `gis_admin` 在 `lab_gis` 資料庫上的所有權限。

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_gis TO gis_admin;  
GRANT
```

退出工作階段，並以 `gis_admin` 身分重新連線至您的 RDS for PostgreSQL 資料庫執行個體。

```
postgres=> psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=gis_admin --password --dbname=lab_gis  
Password for user gis_admin:..  
lab_gis=>
```

按照後續步驟中的詳細說明，繼續設定擴充功能。

步驟 2：載入 PostGIS 擴充功能

PostGIS 擴充功能包含數個共同運作的相關擴充功能，以提供地理空間的功能。根據您的使用案例，可能不需要在此步驟中建立的所有擴充功能。

使用 `CREATE EXTENSION` 陳述式載入 PostGIS 擴充。

```
CREATE EXTENSION postgis;  
CREATE EXTENSION  
CREATE EXTENSION postgis_raster;  
CREATE EXTENSION  
CREATE EXTENSION fuzzystrmatch;  
CREATE EXTENSION  
CREATE EXTENSION postgis_tiger_geocoder;
```

```
CREATE EXTENSION
CREATE EXTENSION postgis_topology;
CREATE EXTENSION
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION
```

您可以執行下列中顯示的 SQL 查詢來確認結果，其中列出擴充功能及其擁有者。

```
SELECT n.nspname AS "Name",
       pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
FROM pg_catalog.pg_namespace n
WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
ORDER BY 1;
```

List of schemas

Name	Owner
public	postgres
tiger	rdsadmin
tiger_data	rdsadmin
topology	rdsadmin

(4 rows)

步驟 3：轉移擴充功能的所有權

使用 ALTER SCHEMA 陳述式將結構描述的擁有權移轉至 gis_admin 角色。

```
ALTER SCHEMA tiger OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA tiger_data OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA topology OWNER TO gis_admin;
ALTER SCHEMA
```

您可執行下列 SQL 查詢，來確認所有權變更。您也可以使用 \dn 中繼命令和 psql 命令列。

```
SELECT n.nspname AS "Name",
       pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
FROM pg_catalog.pg_namespace n
WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
ORDER BY 1;
```

```

      List of schemas
      Name          |      Owner
      -----+-----
 public           | postgres
 tiger            | gis_admin
 tiger_data       | gis_admin
 topology         | gis_admin
 (4 rows)

```

步驟 4：轉移 PostGIS 物件的所有權

使用下列函式將 PostGIS 物件的擁有權移轉至 `gis_admin` 角色。從 `psql` 提示字元執行下列陳述式來建立函式。

```

CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $$ BEGIN EXECUTE
  $1; RETURN $1; END; $$;
CREATE FUNCTION

```

接下來，執行下列查詢來執行 `exec` 函數，該函數會執行陳述式及變更權限。

```

SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname)
 || ' OWNER TO gis_admin;')
FROM (
  SELECT nspname, relname
  FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
  WHERE nspname in ('tiger','topology') AND
  relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;

```

步驟 5：測試擴充功能

為避免需要指定結構描述名稱，請使用下列命令將 `tiger` 結構描述新增至您的搜尋路徑。

```

SET search_path=public,tiger;
SET

```

使用下列 `SELECT` 陳述式來測試 `tiger` 結構描述。

```

SELECT address, streetname, streettypeabbrev, zip

```

```

FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev | zip
-----+-----+-----+-----
      1 | Devonshire | Pl                | 02109
(1 row)

```

要進一步了解此擴充功能，請參閱 PostGIS 文件中的 [Tiger Geocoder](#) (Tiger 地理編碼器)。

使用下列 SELECT 陳述式來測試存取 topology 結構描述。這樣會呼叫 createtopology 函數，以使用指定的空間參考識別碼 (26986) 和預設公差 (0.5) 註冊新拓撲物件 (my_new_topo)。若要進一步了解，請參閱 PostGIS 文件 [Create Topology](#) 中的。

```

SELECT topology.createtopology('my_new_topo',26986,0.5);
createtopology
-----
              1
(1 row)

```

步驟 6：升級 PostGIS 擴充功能

每個新版本的 PostgreSQL 都支援一個或多個與該版本相容的 PostGIS 擴充功能版本。將 PostgreSQL 引擎升級到新版本並不會自動升級 PostGIS 擴充功能。升級 PostgreSQL 引擎之前，您通常會將 PostGIS 升級到目前 PostgreSQL 版本的最新可用版本。如需詳細資訊，請參閱 [PostGIS 擴充功能版本](#)。

PostgreSQL 引擎升級之後，接著再次將 PostGIS 擴充功能升級為支援新升級之 PostgreSQL 引擎版本的版本。如需升級 PostgreSQL 資料庫引擎的詳細資訊，請參閱 [如何執行主要版本升級](#)。

您可以隨時在 RDS for PostgreSQL 資料庫執行個體上檢查可用的 PostGIS 擴充功能版本更新。若要這麼做，請執行下列命令。PostGIS 2.5.0 和更新版本可使用此功能。

```
SELECT postGIS_extensions_upgrade();
```

如果您的應用程式不支援最新的 PostGIS 版本，您仍然可以安裝主要版本中可用的舊版 PostGIS，如下所示。

```
CREATE EXTENSION postgis VERSION "2.5.5";
```

如果您想要從舊版本升級到特定的 PostGIS 版本，也可以使用以下命令。

```
ALTER EXTENSION postgis UPDATE TO "2.5.5";
```

視您要從哪個版本升級而定，您可能需要再次執行此函數。第一次執行函數的結果會決定是否需要額外的升級函數。例如，這是從 PostGIS 2 升級到 PostGIS 3 的情況。如需詳細資訊，請參閱 [將 PostGIS 2 升級到 PostGIS 3](#)。

如果您升級此擴充功能以準備 PostgreSQL 引擎的主要版本升級，您可以繼續執行其他初步工作。如需詳細資訊，請參閱 [如何執行主要版本升級](#)。

PostGIS 擴充功能版本

我們建議您安裝所有擴充功能的版本，例如在《Amazon RDS for PostgreSQL 版本備註》中的 [Amazon RDS for PostgreSQL 擴充功能版本](#) 所列出的 PostGIS。若要取得您的發行版本中有哪些可用版本，請使用下列命令。

```
SELECT * FROM pg_available_extension_versions WHERE name='postgis';
```

您也可以從 Amazon RDS for PostgreSQL 版本備註的以下各節找到版本資訊：

- [Amazon RDS 支援第 16 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 15 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 14 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 13 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 12 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 11 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 10 版擴充功能](#)
- [Amazon RDS 上支援的 PostgreSQL 9.6.x 版擴充功能](#)

將 PostGIS 2 升級到 PostGIS 3

從 3.0 版開始，PostGIS 點陣函數現在是一個單獨的擴充功能，`postgis_raster`。此擴充功能具有自己的安裝和升級路徑。如此一來，可以從核心 `postgis` 擴充功能移除點陣影像處理所需的數十種函數、資料類型和其他成品。這意味著，如果您的使用案例不需要點陣處理，則不需要安裝 `postgis_raster` 擴充功能。

在以下升級範例中，第一個升級命令會將點陣函數擷取至 `postgis_raster` 擴充功能。接著便需要第二個升級命令將 `postgres_raster` 升級到新版本。

從 PostGIS 2 升級到 PostGIS 3

1. 識別可用於 PostgreSQL 版本的 PostGIS 預設版本，而其中 PostgreSQL 位於您的 RDS for PostgreSQL 資料庫執行個體。若要這麼做，請執行下列查詢。

```
SELECT * FROM pg_available_extensions
    WHERE default_version > installed_version;
 name | default_version | installed_version | comment
-----+-----+-----
+-----+-----+-----
 postgis | 3.1.4          | 2.3.7            | PostGIS geometry and geography
 spatial types and functions
(1 row)
```

2. 識別您的 RDS for PostgreSQL 資料庫執行個體上，每個資料庫中安裝的 PostGIS 版本。換句話說，查詢每個使用者資料庫，如下所示。

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
    AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
 Name | Version | Schema | Description
-----+-----+-----
+-----+-----+-----
 postgis | 2.3.7   | public | PostGIS geometry, geography, and raster spatial types
 and functions
(1 row)
```

預設版本 (PostGIS 3.1.4) 與安裝的版本 (PostGIS 2.3.7) 之間不符，代表您需要升級 PostGIS 擴充功能。

```
ALTER EXTENSION postgis UPDATE;
```

```
ALTER EXTENSION
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
```

3. 執行下列查詢，以確認點陣函數現在位於其自己的套件中。

```
SELECT
  probin,
  count(*)
FROM
  pg_proc
WHERE
  probin LIKE '%postgis%'
GROUP BY
  probin;
```

probin	count
\$libdir/rtpostgis-2.3	107
\$libdir/postgis-3	487

(2 rows)

輸出結果顯示版本之間仍然存在差異。PostGIS 函數是第 3 版 (postgis-3)，而點陣函數 (rtpostgis) 是第 2 版 (rtpostgis-2.3)。若要完成升級，請再次執行升級命令，如下所示。

```
postgres=> SELECT postgis_extensions_upgrade();
```

您可以放心忽略警告訊息。再次執行下列查詢以確認升級是否已完成。當 PostGIS 和所有相關的擴充功能均未標記為需要升級時，則升級完成。

```
SELECT postgis_full_version();
```

4. 使用下列查詢來查看已完成的升級程序和個別套裝的擴充功能，並確認其版本是否相符。

```
SELECT
  e.extname AS "Name",
  e.extversion AS "Version",
  n.nspname AS "Schema",
  c.description AS "Description"
FROM
  pg_catalog.pg_extension e
  LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
  LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
```



```
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
   Name      | Version | Schema | Description
-----+-----+-----+-----
postgis      | 3.1.5   | public | PostGIS geometry, geography, and raster
spatial types and functions
postgis_raster | 3.1.5   | public | PostGIS raster types and functions
(2 rows)
```

輸出結果顯示 PostGIS 2 擴充功能已升級到 PostGIS 3，並且 `postgis` 和現在的個別 `postgis_raster` 擴充功能則為 3.1.5 版。

升級完成後，如果您不打算使用點陣函數，您可以按以下方式卸除擴充功能。

```
DROP EXTENSION postgis_raster;
```

使用 Amazon RDS for PostgreSQL 支援的外部資料包裝函式

外部資料包裝函式 (FDW) 是一種特定類型的擴充功能，可提供對外部資料的存取。例如，`oracle_fdw` 擴充功能可讓您的 RDS for PostgreSQL 資料庫叢集使用 Oracle 資料庫。另一個範例是，您可以使用 PostgreSQL 原生 `postgres_fdw` 擴充功能，存取存放在 RDS for PostgreSQL 資料庫執行個體外部的 PostgreSQL 資料庫執行個體中的資料。

在下文中，您可以了解幾個受支援的 PostgreSQL 外部資料包裝函式的資訊。

主題

- [使用 `log_fdw` 擴充功能存取使用 SQL 的資料庫日誌](#)
- [使用 `postgres_fdw` 擴充功能存取外部資料](#)
- [使用 `mysql_fdw` 擴充功能處理 MySQL 資料庫](#)
- [使用 `oracle_fdw` 擴充功能處理 Oracle 資料庫](#)
- [使用 `tds_fdw` 擴充功能處理 SQL 資料庫](#)

使用 `log_fdw` 擴充功能存取使用 SQL 的資料庫日誌

RDS for PostgreSQL 資料庫執行個體支援 `log_fdw` 延伸模組，您可以用來使用 SQL 介面存取資料庫引擎日誌。`log_fdw` 擴充功能推出兩個新函數，可讓您輕鬆為資料庫日誌建立外部資料表：

- `list_postgres_log_files` – 列出資料庫日誌目錄中的檔案和檔案大小 (以位元組為單位)。
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – 在目前資料庫中為指定的檔案建立外部資料表。

`log_fdw` 建立的所有函數皆為 `rds_superuser` 所擁有。`rds_superuser` 角色的成員可以將這些函數的存取權授予其他資料庫使用者。

依預設，日誌檔案由 Amazon RDS 以 `stderr` (標準錯誤) 格式產生，如 `log_destination` 參數中所指定。此參數只有兩個選項：`stderr` 和 `csvlog` (逗號分隔值, CSV)。如果您將 `csvlog` 選項新增至參數，Amazon RDS 會同時產生 `stderr` 和 `csvlog` 日誌。這可能會影響資料庫叢集上的儲存容量，因此您需要了解影響日誌處理的其他參數。如需更多詳細資訊，請參閱 [設定日誌目標 \(`stderr`、`csvlog`\)](#)。

產生 `csvlog` 日誌的一個好處為 `log_fdw` 延伸允許您建置外部資料表，並將資料整齊分割成數個資料欄。為此，您的執行個體需要與自訂資料庫參數群組關聯，則您可變更 `log_destination` 的設定。如需如何執行作業的資訊，請參閱 [在 RDS for PostgreSQL 資料庫執行個體上搭配使用參數](#)。

下列範例假設 `log_destination` 參數包括 `csvlog`。

使用 `log_fdw` 擴充功能

1. 安裝 `log_fdw` 擴充功能。

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. 建立日誌伺服器做為外部資料包裝函數。

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. 叢日誌檔案清單全選。

```
postgres=> SELECT * FROM list_postgres_log_files() ORDER BY 1;
```

範例回應如下所示。

file_name	file_size_bytes
postgresql.log.2023-08-09-22.csv	1111
postgresql.log.2023-08-09-23.csv	1172
postgresql.log.2023-08-10-00.csv	1744
postgresql.log.2023-08-10-01.csv	1102

(4 rows)

4. 針對選取的檔案，建立只有單一 'log_entry' 資料欄的資料表。

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
'log_server', 'postgresql.log.2023-08-09-22.csv');
```

除了目前存在的資料表之外，回應不提供任何其他詳細資訊。

```
-----
(1 row)
```

5. 選取日誌檔案的範例。以下程式碼會擷取日誌時間和錯誤訊息描述。

```
postgres=> SELECT log_time, message FROM my_postgres_error_log ORDER BY 1;
```

範例回應如下所示。

```

          log_time          |          message
-----+-----
+-----+-----
Tue Aug 09 15:45:18.172 2023 PDT | ending log output to stderr
Tue Aug 09 15:45:18.175 2023 PDT | database system was interrupted; last known up
  at 2023-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2023 PDT | checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2023 PDT | redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2023 PDT | next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2023 PDT | next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2023 PDT | oldest unfrozen transaction ID: 1822, in
  database 1
(7 rows)

```

使用 postgres_fdw 擴充功能存取外部資料

您可以使用 [postgres_fdw](#) 擴充功能，存取遠端資料庫伺服器上資料表中的資料。如果您設定來自 PostgreSQL 資料庫執行個體的遠端連線，則也可以存取您的僅供讀取複本。

若要使用 postgres_fdw 來存取遠端資料庫伺服器

1. 安裝 postgres_fdw 擴充功能。

```
CREATE EXTENSION postgres_fdw;
```

2. 使用 CREATE SERVER 建立外部資料伺服器。

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. 建立使用者對應，找出要使用於遠端伺服器的角色。

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. 建立一個資料表，其對應至遠端伺服器上的資料表。

```
CREATE FOREIGN TABLE foreign_table (  
    id integer NOT NULL,  
    data text)  
SERVER foreign_server  
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

使用 mysql_fdw 擴充功能處理 MySQL 資料庫

若要從 RDS for PostgreSQL 資料庫執行個體存取相容於 MySQL 的資料庫，您可以安裝並使用 mysql_fdw 延伸模組。此外部資料包裝函數可讓您使用 MySQL、Aurora MySQL、MariaDB 和其他相容於 MySQL 的資料庫的 RDS。從 RDS for PostgreSQL 資料庫執行個體到 MySQL 資料庫的連線是在最大努力的基礎上加密的，具體取決於用戶端和伺服器的組態。但是，如有需要，您可以強制執行加密。如需更多詳細資訊，請參閱 [搭配此擴充功能使用傳輸中加密](#)。

Amazon RDS for PostgreSQL 14.2、13.6 版和更新版本支援 mysql_fdw 延伸模組。它支援從 RDS for PostgreSQL DB 對相容於 MySQL 的資料庫執行個體上的資料表進行選擇、插入、更新和刪除。

主題

- [將 RDS for PostgreSQL 資料庫設定為使用 mysql_fdw 延伸模組](#)
- [範例：從 RDS for PostgreSQL 使用 RDS for MySQL 資料庫](#)
- [搭配此擴充功能使用傳輸中加密](#)

將 RDS for PostgreSQL 資料庫設定為使用 mysql_fdw 延伸模組

在您的 RDS for PostgreSQL 資料庫執行個體上設定 mysql_fdw 延伸模組涉及在資料庫執行個體中載入延伸模組，然後建立 MySQL 資料庫執行個體的連線點。針對此任務，您必須有以下關於 MySQL 資料庫執行個體的詳細內容：

- 主機名稱或端點。對於 RDS for MySQL 資料庫執行個體，您可以使用主控台尋找端點。選擇 Connectivity & security (連線和安全) 索引標籤，然後查看「端點和連線埠」區段。
- 連線埠號碼。MySQL 的預設連線埠號為 3306。
- 資料庫的名稱。資料庫識別符。

您也必須提供 MySQL 連線埠 3306 的安全群組或存取控制清單 (ACL) 的存取權限。RDS for PostgreSQL 資料庫執行個體和 RDS for MySQL 資料庫執行個體都需要存取連接埠 3306。如果未正確設定存取權限，當您嘗試連線至相容於 MySQL 的資料表時，會看到類似以下的錯誤訊息：

```
ERROR: failed to connect to MySQL: Can't connect to MySQL server on 'hostname.aws-region.rds.amazonaws.com:3306' (110)
```

在以下程序中，您 (作為 `rds_superuser` 帳戶) 建立外部伺服器。然後，您將外部伺服器的存取權限授予特定使用者。然後，這些使用者建立自己的映射到適合的 MySQL 使用者帳戶以使用 MySQL 資料庫執行個體。

使用 `mysql_fdw` 存取 MySQL 資料庫伺服器

1. 使用有 `rds_superuser` 角色的帳戶連線到您的 PostgreSQL 資料庫執行個體。如果您在為 RDS for PostgreSQL 資料庫執行個體建立 RDS 時接受了預設值，則使用者名稱為 `postgres`，而且您可以使用 `psql` 命令列工具進行連線，如下所示：

```
psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password
```

2. 安裝 `mysql_fdw` 擴充功能，如下所示：

```
postgres=> CREATE EXTENSION mysql_fdw;  
CREATE EXTENSION
```

在 RDS for PostgreSQL 資料庫執行個體上安裝延伸模組之後，您可以設定提供 MySQL 資料庫連線的外部伺服器。

建立外部伺服器

在 RDS for PostgreSQL 資料庫執行個體上執行這些任務。這些步驟假設您以具有 `rds_superuser` 權限 (例如 `postgres`) 的使用者進行連線。

1. 在 RDS for PostgreSQL 資料庫執行個體中建立外部伺服器：

```
postgres=> CREATE SERVER mysql-db FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'db-  
name.111122223333.aws-region.rds.amazonaws.com', port '3306');  
CREATE SERVER
```

- 將適當的使用者存取權限授予外部伺服器。這些使用者應該是非管理員使用者，亦即沒有 `rds_superuser` 角色。

```
postgres=> GRANT USAGE ON FOREIGN SERVER mysql-db to user1;  
GRANT
```

PostgreSQL 使用者透過外部伺服器建立和管理自己的 MySQL 資料庫連線。

範例：從 RDS for PostgreSQL 使用 RDS for MySQL 資料庫

假設您在 RDS for PostgreSQL 資料庫執行個體上具有簡易的資料表。您的 RDS for PostgreSQL 使用者想要查詢該資料表上的 (SELECT)、INSERT、UPDATE 和 DELETE 項目。假設 `mysql_fdw` 擴充功能已在 RDS for PostgreSQL 資料庫執行個體上建立，詳情如上述程序所述。作為具有 `rds_superuser` 權限的使用者，在您連線到 RDS 的 PostgreSQL 資料庫執行個體後，可繼續執行以下步驟。

- 在 RDS for PostgreSQL 資料庫執行個體上，建立外部伺服器：

```
test=> CREATE SERVER mysqldb FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'your-DB.aws-region.rds.amazonaws.com', port '3306');  
CREATE SERVER
```

- 授與使用量給沒有 `rds_superuser` 許可的使用者，例如 `user1`：

```
test=> GRANT USAGE ON FOREIGN SERVER mysqldb TO user1;  
GRANT
```

- 以 `user1` 身分連線，然後建立一個 MySQL 使用者的映射：

```
test=> CREATE USER MAPPING FOR user1 SERVER mysqldb OPTIONS (username 'myuser',  
password 'mypassword');  
CREATE USER MAPPING
```

- 建立 MySQL 資料表的外部資料表連結。

```
test=> CREATE FOREIGN TABLE mytab (a int, b text) SERVER mysqldb OPTIONS (dbname  
'test', table_name '');  
CREATE FOREIGN TABLE
```

- 對外部資料表執行簡單查詢：

```
test=> SELECT * FROM mytab;
a | b
---+-----
1 | apple
(1 row)
```

6. 您可以從 MySQL 資料表新增、變更和移除資料。例如：

```
test=> INSERT INTO mytab values (2, 'mango');
INSERT 0 1
```

再次執行 SELECT 查詢以查看結果：

```
test=> SELECT * FROM mytab ORDER BY 1;
a | b
---+-----
1 | apple
2 | mango
(2 rows)
```

搭配此擴充功能使用傳輸中加密

依預設，從 RDS for PostgreSQL 到 MySQL 的連線會使用傳輸中加密 (TLS/SSL)。但是，當用戶端和伺服器組態不同時，連線會回退到未加密。您可以在 RDS for MySQL 使用者帳戶上指定 REQUIRE SSL 選項，強制加密所有的對外連線。此方法也適用於 MariaDB 和 Aurora MySQL 使用者帳戶。

當 MySQL 使用者帳戶設定為 REQUIRE SSL，如果無法建立安全連線，連線嘗試將會失敗。

若要強制加密現有 MySQL 資料庫使用者帳戶，可使用 ALTER USER 命令。語法依據 MySQL 版本而異，如下表所示。如需詳細資訊，請參閱《MySQL 參考手冊》中的 [ALTER USER](#)。

MySQL 5.7、MySQL 8.0	MySQL 5.6
ALTER USER ' <i>user</i> '@'%' REQUIRE SSL;	GRANT USAGE ON *.* to ' <i>user</i> '@'%' REQUIRE SSL;

如需 mysql_fdw 擴充功能的詳細資訊，請參閱 [mysql_fdw](#) 文件。

使用 oracle_fdw 擴充功能處理 Oracle 資料庫

若要從您的 RDS for PostgreSQL 資料庫執行個體 存取 Oracle 資料庫，您可以安裝並使用 oracle_fdw 擴充功能。此擴充功能是 Oracle 資料庫的外部資料包裝函式。若要進一步了解此擴充功能，請參閱 [oracle_fdw](#) 文件。

RDS for PostgreSQL 12.7、13.3 版及更高版本支援此 oracle_fdw 擴充功能。

主題

- [開啟 oracle_fdw 擴充功能](#)
- [範例：使用 Amazon RDS for Oracle Database 的外部伺服器連結](#)
- [在傳輸中使用加密](#)
- [了解 pg_user_mappings 檢視和許可權限](#)

開啟 oracle_fdw 擴充功能

如要使用 oracle_fdw 擴充功能，請執行下列程序。

如要開啟 oracle_fdw 擴充功能

- 使用具有 rds_superuser 許可的帳戶執行下列命令。

```
CREATE EXTENSION oracle_fdw;
```

範例：使用 Amazon RDS for Oracle Database 的外部伺服器連結

下列範例顯示使用連結至 Amazon RDS for Oracle 資料庫的外部伺服器。

若要建立連結至 RDS for Oracle 資料庫的外部伺服器

1. 請注意以下 RDS for Oracle 資料庫執行個體的事項：
 - 端點
 - 連線埠
 - 資料庫名稱
2. 建立外部伺服器。

```
test=> CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS (dbserver
'//endpoint:port/DB_name');
CREATE SERVER
```

3. 授予使用權限給沒有 `rds_superuser` 權限的使用者，例如 `user1`。

```
test=> GRANT USAGE ON FOREIGN SERVER oradb TO user1;
GRANT
```

4. 連線為 `user1` 並建立一個對應至 Oracle 使用者的映射。

```
test=> CREATE USER MAPPING FOR user1 SERVER oradb OPTIONS (user 'oracleuser',
password 'mypassword');
CREATE USER MAPPING
```

5. 建立連結至 Oracle 資料表的外部資料表。

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER oradb OPTIONS (table 'MYTABLE');
CREATE FOREIGN TABLE
```

6. 查詢外部資料表。

```
test=> SELECT * FROM mytab;
a
---
1
(1 row)
```

如果查詢報告下列錯誤，請檢查您的安全群組和存取控制清單 (ACL)，以確定這兩個執行個體可以通訊。

```
ERROR: connection for foreign table "mytab" cannot be established
DETAIL: ORA-12170: TNS:Connect timeout occurred
```

在傳輸中使用加密

傳輸中的 PostgreSQL-to-Oracle 加密是以從用戶端和伺服器組態參數的組合為依據。如需使用 Oracle 21c 的範例，請參閱 Oracle 文件中的[關於溝通加密和完整性的值](#)。在 Amazon RDS 上用於 `oracle_fdw` 的用戶端已設定為 `ACCEPTED`，表示加密取決於 Oracle 資料庫伺服器組態。

如果資料庫位於 RDS for Oracle，請參閱 [Oracle 原生網路加密](#) 以設定加密。

了解 pg_user_mappings 檢視和許可權限

PostgreSQL 目錄 pg_user_mapping 儲存從 RDS-PostgreSQL 使用者新增至外部資料 (遠端) 伺服器上使用者的映射。存取目錄受到限制，但您使用 pg_user_mappings 檢視查看映射。接下來，您可找到一個範例，其顯示許可權限如何套用於範例 Oracle 資料庫，但此資訊通常適用於任何外部資料包裝函式。

在以下輸出中，您可以找到映射至三個不同範例使用者的角色和權限。使用者 rdssu1 和 rdssu2 是 rds_superuser 角色的成員，user1 則不是。此範例使用 psql 中繼命令 \du 來列出現有角色。

```
test=> \du
```

Role name	Member of	Attributes	List of roles
rdssu1	{rds_superuser}		
rdssu2	{rds_superuser}		
user1			{}

所有使用者，包括具有 rds_superuser 權限的使用者，可在 pg_user_mappings 資料表中查看自己的使用者映射 (umoptions)。如以下範例所示，當 rdssu1 嘗試獲取所有使用者映射，即使有 rdssu1rds_superuser 權限，仍會引起錯誤：

```
test=> SELECT * FROM pg_user_mapping;
ERROR: permission denied for table pg_user_mapping
```

下列是一些範例。

```
test=> SET SESSION AUTHORIZATION rdssu1;
SET
test=> SELECT * FROM pg_user_mappings;
```

umid	srvid	srvname	umuser	username	umoptions
16414	16411	oradb	16412	user1	
16423	16411	oradb	16421	rdssu1	{user=oracleuser,password=mypwd}
16424	16411	oradb	16422	rdssu2	

(3 rows)

```

test=> SET SESSION AUTHORIZATION rdssu2;
SET
test=> SELECT * FROM pg_user_mappings;
  umid | srvid | srvname | umuser | username |          umoptions
-----+-----+-----+-----+-----+-----
 16414 | 16411 | oradb   | 16412 | user1    |
 16423 | 16411 | oradb   | 16421 | rdssu1   |
 16424 | 16411 | oradb   | 16422 | rdssu2   | {user=oracleuser,password=mypwd}
(3 rows)

test=> SET SESSION AUTHORIZATION user1;
SET
test=> SELECT * FROM pg_user_mappings;
  umid | srvid | srvname | umuser | username |          umoptions
-----+-----+-----+-----+-----+-----
 16414 | 16411 | oradb   | 16412 | user1    | {user=oracleuser,password=mypwd}
 16423 | 16411 | oradb   | 16421 | rdssu1   |
 16424 | 16411 | oradb   | 16422 | rdssu2   |
(3 rows)

```

由於實作 `information_schema._pg_user_mappings` 和 `pg_catalog.pg_user_mappings` 存在差異，手動建立的 `rds_superuser` 需要額外的許可才能在 `pg_catalog.pg_user_mappings` 上檢視密碼。

`rds_superuser` 不需要額外許可才能在 `information_schema._pg_user_mappings` 中檢視密碼。

沒有 `rds_superuser` 角色的使用者僅能在以下情況下在 `pg_user_mappings` 中檢視密碼：

- 目前使用者是被映射的使用者，且擁有伺服器或擁有伺服器上 USAGE 的權限。
- 目前使用者是伺服器擁有者，且映射適用於 PUBLIC。

使用 tds_fdw 擴充功能處理 SQL 資料庫

您可以使用 PostgreSQL `tds_fdw` 擴充功能來存取支援表格式資料串流 (TDS) 協議的資料庫，如 Sybase 和 Microsoft SQL Server 資料庫。此外部資料包裝函式可讓您從 RDS for PostgreSQL 資料庫執行個體連線到使用 TDS 協議的資料庫，包括 Amazon RDS for Microsoft SQL Server。如需詳細資訊，請參閱 GitHub 上的 [tds-fdw/tds_fdw](#) 文件。

Amazon RDS for PostgreSQL 版本 14.2、13.6 及更新版本支援此 `tds_fdw` 擴充功能。

將 Aurora PostgreSQL 資料庫設定為使用 tds_fdw 延伸模組

在以下程序中，您可以找到設定及使用 tds_fdw 與 RDS for PostgreSQL 資料庫執行個體的範例。您必須取得執行個體的以下詳細內容，然後才能使用 tds_fdw 連線到 SQL Server 資料庫：

- 主機名稱或端點。如果是 RDS for SQL Server 資料庫執行個體，您可以使用主控台找到端點。選擇 Connectivity & security (連線和安全) 索引標籤，然後查看「端點和連線埠」區段。
- 連線埠號碼。Microsoft SQL 伺服器的預設連線埠號是 1433。
- 資料庫的名稱。資料庫識別符。

您也必須提供 SQL 連線埠 1433 的安全群組或存取控制清單 (ACL) 的存取權限。RDS for PostgreSQL 資料庫執行個體和 RDS for SQL 資料庫執行個體都需要存取連線埠 1433。如果存取權限未正確設定，當您嘗試查詢 Microsoft SQL Server 時，您會看到以下錯誤訊息：

```
ERROR: DB-Library error: DB #: 20009, DB Msg: Unable to connect:
Adaptive Server is unavailable or does not exist (mssql2019.aws-
region.rds.amazonaws.com), OS #: 0, OS Msg: Success, Level: 9
```

使用 tds_fdw 連線到 SQL Server 資料庫

1. 使用有 rds_superuser 角色的帳戶連線到您的 PostgreSQL 資料庫執行個體：

```
psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
username=test --password
```

2. 安裝 tds_fdw 擴充功能：

```
test=> CREATE EXTENSION tds_fdw;
CREATE EXTENSION
```

擴充功能安裝到您的 RDS for PostgreSQL 資料庫執行個體之後，您就可以設定外部伺服器。

建立外部伺服器

使用有 rds_superuser 權限的帳戶，在 RDS for PostgreSQL 資料庫執行個體上執行這些任務。

1. 在 RDS for PostgreSQL 資料庫執行個體中建立外部伺服器：

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS
  (servername 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database
  'tds_fdw_testing');
CREATE SERVER
```

如要存取 SQLServer 端上的非 ASCII 資料，請使用 RDS for PostgreSQL 資料庫執行個體中的 `character_set` 選項建立伺服器連結：

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS (servername
  'mssql2019.aws-region.rds.amazonaws.com', port '1433', database 'tds_fdw_testing',
  character_set 'UTF-8');
CREATE SERVER
```

2. 將權限授予沒有 `rds_superuser` 角色權限的使用者，例如 `user1`：

```
test=> GRANT USAGE ON FOREIGN SERVER sqlserverdb TO user1;
```

3. 以 `user1` 身分連線，然後建立一個 SQL Server 使用者的映射：

```
test=> CREATE USER MAPPING FOR user1 SERVER sqlserverdb OPTIONS (username
  'sqlserveruser', password 'password');
CREATE USER MAPPING
```

4. 建立連結至 SQL Server 資料表的外部資料表：

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER sqlserverdb OPTIONS (table
  'MYTABLE');
CREATE FOREIGN TABLE
```

5. 查詢外部資料表：

```
test=> SELECT * FROM mytab;
 a
 ---
 1
(1 row)
```

對連線使用傳輸中加密

從 RDS for PostgreSQL 到 SQL Server 的連線使用傳輸中加密 (TLS/SSL)，具體取決於 SQL Server 資料庫組態。如果 SQL 伺服器未設定為加密，則 RDS for PostgreSQL 用戶端對 SQL Server 資料庫發出的請求將回退為未加密。

您可以透過設定 `rds.force_ssl` 參數，強制加密 RDS for SQL Server 資料庫執行個體的連線。若要了解作法，請參閱[強制使用 SSL 連線至資料庫執行個體](#)。如需有關 RDS for SQL Server 的 SSL/TLS 組態的詳細資訊，請參閱[對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)。

使用適用於 PostgreSQL 的受信任語言延伸模組

PostgreSQL 的受信任語言延伸模組是用於建置 PostgreSQL 延伸模組的開放原始碼開發套件。它可讓您建置高效能 PostgreSQL 延伸模組，並在 RDS for PostgreSQL 資料庫執行個體上安全地執行它們。透過使用適用於 PostgreSQL 的受信任語言延伸模組 (TLE)，您可以建立 PostgreSQL 延伸模組，遵循記載的方法來擴充 PostgreSQL 功能。如需詳細資訊，請參閱 PostgreSQL 文件中的[將相關物件封裝為延伸模組](#)。

TLE 的一個主要優點，就是您可以在未於 PostgreSQL 執行個體之下提供檔案系統存取權的環境中使用它。先前，安裝新的延伸模組需要存取檔案系統。TLE 移除了此條件約束。它提供一個開發環境，可讓您針對任何 PostgreSQL 資料庫 (包括在 RDS for PostgreSQL 資料庫執行個體上執行的資料庫) 建立新的延伸模組。

TLE 的設計旨在防止存取對您使用 TLE 建立的延伸模組而言不安全的資源。其執行時間環境會限制任何延伸模組瑕疵對單一資料庫連線的影響。TLE 還可讓資料庫管理員對可以安裝延伸模組的人員進行精細控制，並提供用於執行延伸模組的許可模型。

下列 RDS for PostgreSQL 版本支援 TLE：

- 版本 16.1 及更高版本 16 個版本
- 版本 15.2 及更高版本 15 個版本
- 版本 14.5 及更高版本 14 個版本
- 版本 13.12 及更高版本 13 個版本

受信任語言延伸模組開發環境和執行時間會封裝成 `pg_tle` PostgreSQL 延伸模組 1.0.1 版。它支持在 JavaScript, Perl, Tcl, PL/PGSQL 和 SQL 中創建擴展。您可以採取您安裝其他 PostgreSQL 延伸模組的同一方式，在 RDS for PostgreSQL 資料庫執行個體中安裝 `pg_tle` 延伸模組。在設定了 `pg_tle` 之後，開發人員可以使用它，建立新的 PostgreSQL 延伸模組，稱為 TLE 延伸模組。

在下列主題中，您可以找到如何設定受信任語言延伸模組，以及如何開始建立您自己的 TLE 延伸模組的相關資訊。

主題

- [術語](#)
- [使用適用於 PostgreSQL 的受信任語言延伸模組的需求](#)
- [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)

- [適用於 PostgreSQL 的受信任語言延伸模組概觀](#)
- [針對 RDS for PostgreSQL 建立 TLE 延伸模組](#)
- [從資料庫中捨棄您的 TLE 延伸模組](#)
- [解除安裝適用於 PostgreSQL 的受信任語言延伸模組](#)
- [搭配您的延伸模組使用 PostgreSQL 掛鉤](#)
- [在 TLE 中使用自訂資料類型](#)
- [適用於 PostgreSQL 的受信任語言延伸模組的函數參考](#)
- [適用於 PostgreSQL 的受信任語言延伸模組的掛鉤參考](#)

術語

為了協助您更好地了解受信任語言延伸模組，請檢視下列詞彙表以取得本主題中使用的術語。

適用於 PostgreSQL 的受信任語言延伸模組

適用於 PostgreSQL 的受信任語言延伸模組是封裝為 `pg_tle` 延伸模組之開放原始碼開發套件的正式名稱。它可以在任何 PostgreSQL 系統上使用。如需詳細資訊，請參閱 [\(詳見\)](#)。GitHub

受信任語言延伸模組

受信任語言延伸模組是適用於 PostgreSQL 的受信任語言延伸模組的簡稱。此簡稱及其縮寫 (TLE) 也會在本文件中使用。

受信任語言

受信任語言是具有特定安全性屬性的程式設計或指令碼語言。例如，受信任語言通常會限制對檔案系統的存取，而且也會限制使用指定的聯網屬性。TLE 開發套件專為支援受信任語言而設計。PostgreSQL 支援數種不同的語言，這些語言用來建立受信任或不受信任的延伸模組。如需範例，請參閱 PostgreSQL 文件中的 [受信任和不受信任的 PL/Perl](#)。當您使用受信任語言延伸模組建立延伸模組時，延伸模組本質上會使用受信任語言機制。

TLE 延伸模組

TLE 延伸模組是已使用受信任語言延伸模組 (TLE) 開發套件所建立的 PostgreSQL 延伸模組。

使用適用於 PostgreSQL 的受信任語言延伸模組的需求

以下是設定和使用 TLE 開發套件時的需求。

- RDS for PostgreSQL 版本 - 僅在 RDS for PostgreSQL 13.12 版和更新的第 13 版、14.5 版和更新的第 14 版，以及 15.2 版及更新版本上支援受信任語言延伸模組。
- 如果您需要升級 RDS for PostgreSQL 執行個體，請參閱 [升級 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。
- 如果您還沒有執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，則可以建立一個。如需詳細資訊，請參閱 RDS for PostgreSQL 資料庫執行個體，請參閱 [建立並連線至 PostgreSQL 資料庫執行個體](#)。
- 需要 **rds_superuser** 權限 - 若要設定 pg_tle 延伸模組，您的資料庫使用者角色必須具有 rds_superuser 角色的許可。根據預設，會將此角色授予建立的 postgres 使用者。RDS for PostgreSQL 資料庫執行個體。
- 需要自訂資料庫參數群組 – 您的 RDS for PostgreSQL 資料庫執行個體必須搭配自訂資料庫參數群組進行設定。
 - 如果您的 RDS for PostgreSQL 資料庫執行個體未搭配自訂資料庫參數群組進行設定，您應該建立一個，並將其與 RDS for PostgreSQL 資料庫執行個體建立關聯。如需步驟的簡短摘要，請參閱 [建立並套用自訂資料庫參數群組](#)。
 - 如果您的 RDS for PostgreSQL 資料庫執行個體已使用自訂資料庫參數群組進行設定，您可以設定受信任語言延伸模組。如需詳細資訊，請參閱 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)。

建立並套用自訂資料庫參數群組

請使用下列步驟建立自訂資料庫參數群組，並設定您的 RDS for PostgreSQL 資料庫執行個體來使用該群組。

主控台

建立自訂資料庫參數群組，並將其與您的 RDS for PostgreSQL 資料庫執行個體搭配使用。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 從 Amazon RDS 功能表中，選擇 Parameter groups (參數群組)。
3. 選擇 Create parameter group (建立參數群組)。
4. 在 Parameter group details (參數群組詳細資訊) 頁面中，輸入下列資訊：
 - 針對 Parameter group family (參數群組系列)，選擇 postgres14。。
 - 針對 Type (類型)，選擇 DB Parameter Group (資料庫參數群組)。

- 針對 Group name (群組名稱)，在操作內容中給與您的參數群組一個有意義的名稱。
 - 針對 Description (描述)，輸入有用的描述，以便您團隊中的其他成員可以輕鬆找到它。
5. 選擇建立。您的自訂資料庫參數群組是在 AWS 區域中建立。您現在可以遵循下列步驟修改 RDS for PostgreSQL 資料庫執行個體以使用它。
 6. 從 Amazon RDS 功能表中選擇 Databases (資料庫)。
 7. 從列出的 RDS for PostgreSQL 資料庫執行個體中選擇您想要哪一個與 TLE 搭配使用，然後選擇 Modify (修改)。
 8. 在 Modify DB instance settings (修改資料庫執行個體設定) 頁面中，於 Additional configuration (其他組態) 區段中尋找 Database options (資料庫選項)，然後從選取器中選擇您的自訂資料庫參數群組。
 9. 選擇 Continue (繼續) 以儲存變更。
 10. 選擇 Apply immediately (立即套用)，以便您可以繼續設定 RDS for PostgreSQL 資料庫執行個體以使用 TLE。

若要繼續針對受信任語言延伸模組設定您的系統，請參閱 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)。

如需深入了解如何使用資料庫參數群組，請參閱 [在資料庫執行個體中使用資料庫參數群組](#)。

AWS CLI

您可以避免在使用 CLI 命令時指定 `--region` 引數，方法是使用您的預設 AWS 區域來設定您的 AWS CLI。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的 [組態基礎概念](#)。

建立自訂資料庫參數群組，並將其與您的 RDS for PostgreSQL 資料庫執行個體搭配使用。

1. 使用此 [create-db-parameter-group](#) AWS CLI 指令建立以 AWS 區域

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --region aws-region \  
  --db-parameter-group-name custom-params-for-pg-tle \  
  --db-parameter-group-family postgres14 \  
  --description "My custom DB parameter group for Trusted Language Extensions"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^
  --region aws-region ^
  --db-parameter-group-name custom-params-for-pg-tle ^
  --db-parameter-group-family postgres14 ^
  --description "My custom DB parameter group for Trusted Language Extensions"
```

您的自訂資料庫參數群組可在 AWS 區域中使用，因此您可以修改 RDS for PostgreSQL 資料庫叢集以使用它。

2. 使用此命 [modify-db-instance](#) AWS CLI 令將自訂資料庫參數群組套用至。您的 RDS 資 PostgreSQL 行個體。此命令會立即重新啟動作用中執行個體。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --region aws-region \  
  --db-instance-identifier your-instance-name \  
  --db-parameter-group-name custom-params-for-pg-tle \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
  --region aws-region ^
  --db-instance-identifier your-instance-name ^
  --db-parameter-group-name custom-params-for-pg-tle ^
  --apply-immediately
```

若要繼續針對受信任語言延伸模組設定您的系統，請參閱 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)。

如需詳細資訊，請參閱 [使用參數群組](#)。

在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組

下列步驟假設您的 RDS for PostgreSQL 資料庫執行個體與自訂資料庫參數群組相關聯。您可以 AWS CLI 針對這些步驟使用 AWS Management Console 或。

當您在 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組時，可以將其安裝在特定資料庫中，供具有該資料庫許可的資料庫使用者使用。

主控台

設定受信任語言延伸模組

使用屬於 `rds_superuser` 群組 (角色) 成員的帳戶執行下列步驟。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇您的 RDS for PostgreSQL 資料庫執行個體。
3. 針對您的 開啟 Configuration (組態) 索引標籤。RDS for PostgreSQL 資料庫執行個體。在執行個體詳細資訊之間，尋找 Parameter group (參數群組) 連結。
4. 選擇連結以開啟與 相關聯的自訂參數。RDS for PostgreSQL 資料庫執行個體。
5. 在 Parameters (參數) 搜尋欄位中，輸入 `shared_pre` 以尋找 `shared_preload_libraries` 參數。
6. 選擇 Edit parameters (編輯參數) 以存取屬性值。
7. 在 Values (值) 欄位中，將 `pg_tle` 新增至清單。使用逗號區隔值清單中的項目。

<input type="checkbox"/>	Name	Values	Allowed values
<input type="checkbox"/>	shared_preload_libraries	pg_tle	auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_tle, pg_transport, pprofiler

8. 重新啟動 RDS for PostgreSQL 資料庫執行個體，以便您對 `shared_preload_libraries` 參數所做的變更生效。
9. 當執行個體可用時，請驗證 `pg_tle` 是否已初始化。使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;  
shared_preload_libraries  
-----
```

```
rdsutils,pg_tle
(1 row)
```

10. 在初始化 `pg_tle` 延伸模組之後，您現在可以建立延伸模組。

```
CREATE EXTENSION pg_tle;
```

您可以使用下列 `psql` 中繼命令，驗證是否已安裝延伸模組。

```
labdb=> \dx
                                List of installed extensions
  Name   | Version | Schema  | Description
-----+-----+-----+-----
 pg_tle  | 1.0.1   | pgtle   | Trusted-Language Extensions for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

11. 在設定時，將 `pgtle_admin` 角色授予您已針對 RDS for PostgreSQL 資料庫執行個體建立的主要使用者名稱。如果您接受預設值，其為 `postgres`。

```
labdb=> GRANT pgtle_admin TO postgres;
GRANT ROLE
```

您可以使用 `psql` 中繼命令來驗證授予是否已發生，如下列範例所示。只有 `pgtle_admin` 和 `postgres` 角色會顯示在輸出中。如需詳細資訊，請參閱 [了解 rds_superuser 角色](#)。

```
labdb=> \du
                                List of roles
  Role name   | Attributes                               | Member of
-----+-----+-----
 pgtle_admin  | Cannot login                             | {}
 postgres    | Create role, Create DB                   +| {rds_superuser,pgtle_admin}
              | Password valid until infinity            |...
```

12. 使用 `\q` 中繼命令關閉 `psql` 工作階段。

```
\q
```

若要開始建立 TLE 延伸模組，請參閱 [範例：使用 SQL 建立受信任語言延伸模組](#)。

AWS CLI

您可以避免在使用 CLI 命令時指定 `--region` 引數，方法是使用您的預設 AWS 區域來設定您的 AWS CLI。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[組態基礎概念](#)。

設定受信任語言延伸模組

1. 使用[modify-db-parameter-group](#) AWS CLI 指令加入 `pg_tle` 至 `shared_preload_libraries` 參數。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name custom-param-group-name \  
  --parameters  
  "ParameterName=shared_preload_libraries,ParameterValue=pg_tle,ApplyMethod=pending-  
reboot" \  
  --region aws-region
```

2. 使用此[reboot-db-instance](#) AWS CLI 命令將入器執行個體重新啟動，並初始化程式庫。 `pg_tle`

```
aws rds reboot-db-instance \  
  --db-instance-identifier your-instance \  
  --region aws-region
```

3. 當執行個體可用時，您可以驗證 `pg_tle` 是否已初始化。使用 `psql` 連線至 RDS for PostgreSQL 資料庫執行個體，然後執行下列命令。

```
SHOW shared_preload_libraries;  
shared_preload_libraries  
-----  
rdsutils,pg_tle  
(1 row)
```

在初始化 `pg_tle` 之後，您現在可以建立延伸模組。

```
CREATE EXTENSION pg_tle;
```

4. 在設定時，將 `pgtle_admin` 角色授予您已針對 RDS for PostgreSQL 資料庫執行個體建立的主要使用者名稱。如果您接受預設值，其為 `postgres`。

```
GRANT pgtle_admin TO postgres;  
GRANT ROLE
```

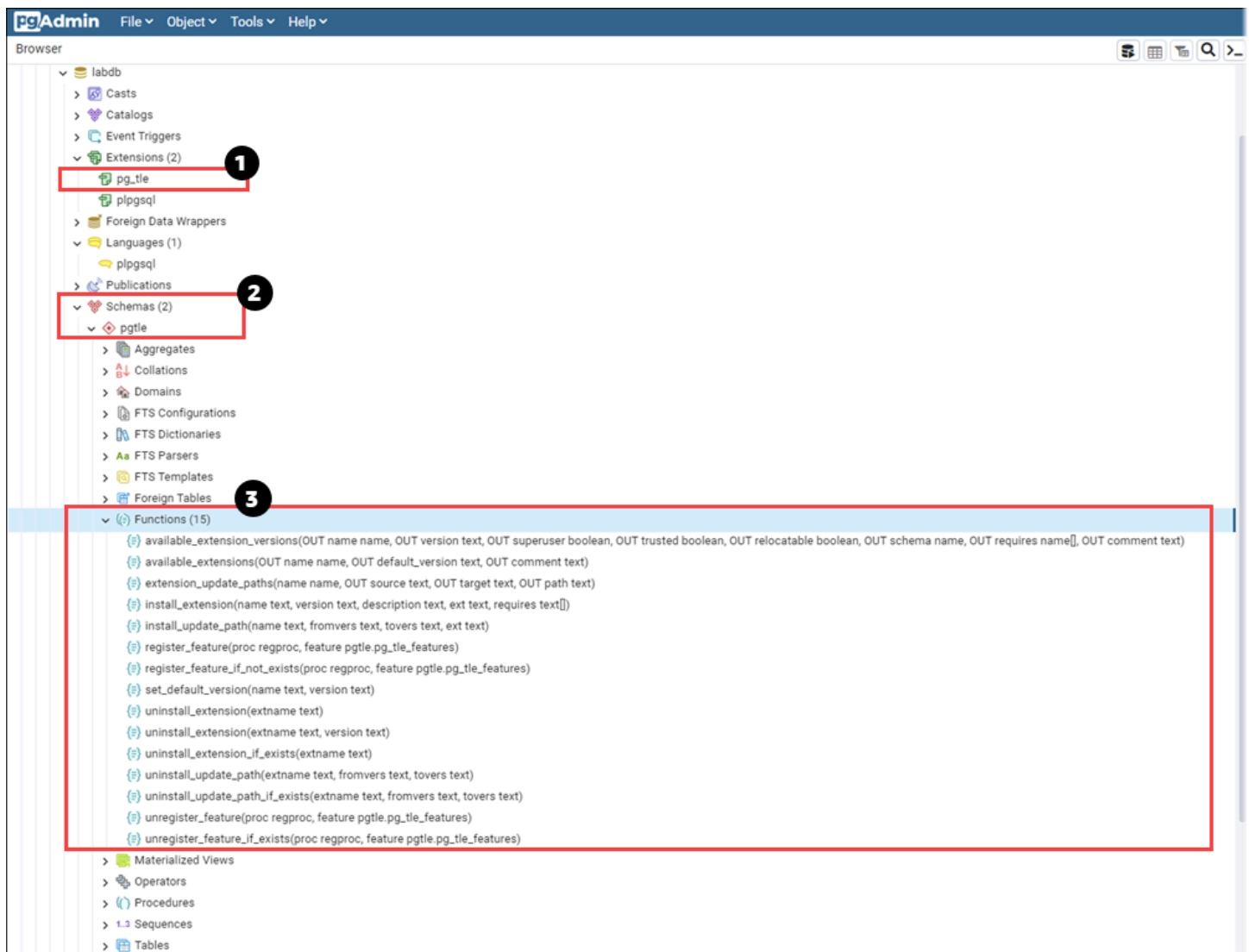
5. 關閉 psql 工作階段，如下所示。

```
labdb=> \q
```

若要開始建立 TLE 延伸模組，請參閱 [範例：使用 SQL 建立受信任語言延伸模組](#)。

適用於 PostgreSQL 的受信任語言延伸模組概觀

適用於 PostgreSQL 的受信任語言延伸模組是一種 PostgreSQL 延伸模組，而您使用您設定其他 PostgreSQL 延伸模組的同一方式，將其安裝在 RDS for PostgreSQLR 資料庫執行個體中。在 pgAdmin 用戶端工具中範例資料庫的下列影像中，您可以檢視構成 pg_tle 延伸模組的一些元件。



您可以查看下列詳細資訊。

1. 適用於 PostgreSQL 的受信任語言延伸模組 (TLE) 開發套件會封裝為 `pg_tle` 延伸模組。因此，`pg_tle` 會新增至其安裝所在資料庫的可用延伸模組。
2. TLE 有其自己的結構描述 (`pgtle`)。此結構描述包含協助程式函數 (3)，用於安裝和管理您建立的延伸模組。
3. TLE 提供了十幾個協助程式函數，用於安裝、註冊和管理您的延伸模組。若要進一步了解這些函數，請參閱 [適用於 PostgreSQL 的受信任語言延伸模組的函數參考](#)。

`pg_tle` 延伸套件的其他元件包含下列項目：

- **`pgtle_admin`** 角色 – 安裝 `pg_tle` 延伸模組時會建立 `pgtle_admin` 角色。此角色具有特殊權限，且應如此對待。強烈建議您在將 `pgtle_admin` 角色授予資料庫使用者時遵循最低權限原則。換句話說，只將 `pgtle_admin` 角色授予資料庫使用者，允許其建立、安裝和管理新的 TLE 延伸模組，例如 `postgres`。
- **`pgtle.feature_info`** 資料表 – `pgtle.feature_info` 資料表是受保護的資料表，其中包含 TLE、掛鉤，以及其使用的自訂預存程序和函數的相關資訊。如果具有 `pgtle_admin` 權限，則您可以使用下列受信任語言延伸模組函數，在資料表中新增和更新該資訊。
 - [pgtle.register_feature](#)
 - [pgtle.register_feature_if_not_exists](#)
 - [pgtle.unregister_feature](#)
 - [pgtle.unregister_feature_if_exists](#)

針對 RDS for PostgreSQL 建立 TLE 延伸模組

您可以在任何已安裝延伸模組的 RDS for PostgreSQL 資料庫執行個體中安裝使用 TLE 建立的任何 `pg_tle` 延伸模組。`pg_tle` 延伸模組的範圍會限定在其安裝所在的 PostgreSQL 資料庫。您使用 TLE 建立的延伸模組，其範圍會限定在相同的資料庫。

使用各種 `pgtle` 函數來安裝構成 TLE 延伸模組的程式碼。下列受信任語言延伸模組函數全都需要 `pgtle_admin` 角色。

- [pgtle.install_extension](#)
- [pgtle.install_update_path](#)
- [pgtle.register_feature](#)
- [pgtle.register_feature_if_not_exists](#)
- [pgtle.set_default_version](#)

- [pgtle.uninstall_extension\(name\)](#)
- [pgtle.uninstall_extension\(name, version\)](#)
- [pgtle.uninstall_extension_if_exists](#)
- [pgtle.uninstall_update_path](#)
- [pgtle.uninstall_update_path_if_exists](#)
- [pgtle.unregister_feature](#)
- [pgtle.unregister_feature_if_exists](#)

範例：使用 SQL 建立受信任語言延伸模組

下列範例說明如何建立名為 `pg_distance` 的 TLE 延伸模組，其中包含一些 SQL 函數，用於使用不同的公式計算距離。在清單中，您可以找到用於計算曼哈頓距離的函數，以及計算歐幾里得距離的函數。如需這些公式之間差異的詳細資訊，請參閱 Wikipedia 中的 [出租車幾何](#) 和 [歐幾里得幾何](#)。

如果您已按照 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#) 中所述設定 `pg_tle` 延伸模組，則可以在自己的 RDS for PostgreSQL 資料庫執行個體中使用此範例。

Note

您必須具有 `pgtle_admin` 角色的權限才能遵循此程序。

建立範例 TLE 延伸模組

下列步驟會使用名為 `labdb` 的範例資料庫。此資料庫是由 `postgres` 主要使用者所擁有。`postgres` 角色也具有 `pgtle_admin` 角色的權限。

1. 使用 `psql` 連線到 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. 複製下列程式碼並將其貼入 `psql` 工作階段主控台中，來建立名為 `pg_distance` 的 TLE 延伸模組。

```
SELECT pgtle.install_extension
(
  'pg_distance',
```

```
'0.1',
'Distance functions for two points',
$_pg_tle_$
CREATE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8, norm int)
RETURNS float8
AS $$
SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
$$ LANGUAGE SQL;

CREATE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2 float8)
RETURNS float8
AS $$
SELECT dist(x1, y1, x2, y2, 1);
$$ LANGUAGE SQL;

CREATE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2 float8)
RETURNS float8
AS $$
SELECT dist(x1, y1, x2, y2, 2);
$$ LANGUAGE SQL;
$_pg_tle_$
);
```

您會看到如下輸出。

```
install_extension
-----
t
(1 row)
```

構成 `pg_distance` 延伸模組的成品現在已安裝在您的資料庫中。這些成品包括延伸模組的控制檔和程式碼，這些是必須存在的項目，如此才能使用 `CREATE EXTENSION` 命令建立延伸模組。換句話說，您仍然需要建立延伸模組，以使其函數可供資料庫使用者使用。

- 若要建立延伸模組，請使用 `CREATE EXTENSION` 命令，如您對任何其他延伸模組所做一樣。與其他延伸模組一樣，資料庫使用者需要在資料庫中具有 `CREATE` 權限。

```
CREATE EXTENSION pg_distance;
```

- 若要測試 `pg_distance` TLE 延伸模組，您可以使用它，計算四點之間的 [曼哈頓距離](#)。

```
labdb=> SELECT manhattan_dist(1, 1, 5, 5);
```

8

若要計算同一組點之間的[歐幾里得距離](#)，您可以使用下列命令。

```
labdb=> SELECT euclidean_dist(1, 1, 5, 5);
5.656854249492381
```

pg_distance 延伸模組會在資料庫中載入函數，並使其可供具有資料庫權限的任何使用者使用。

修改 TLE 延伸模組

若要改善此 TLE 延伸模組中封裝之函數的查詢效能，請將下列兩個 PostgreSQL 屬性新增至其規格。

- IMMUTABLE – IMMUTABLE 屬性確保查詢最佳化工具可以使用最佳化來改善查詢回應時間。如需詳細資訊，請參閱 PostgreSQL 文件中的[函數波動類別](#)。
- PARALLEL SAFE – PARALLEL SAFE 屬性是允許 PostgreSQL 以平行模式執行函數的另一個屬性。如需詳細資訊，請參閱 PostgreSQL 文件中的[CREATE FUNCTION](#)。

在下列範例中，您可以看到如何使用 pgtle.install_update_path 函數將這些屬性新增到每個函數，以建立 pg_distance TLE 延伸模組的版本 0.2。如需此函數狀態的詳細資訊，請參閱[pgtle.install_update_path](#)。您必須具有 pgtle_admin 角色才能執行此任務。

更新現有 TLE 延伸模組並指定預設版本

1. 使用 psql 或其他用戶端工具 (例如 pgAdmin)，連線至 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. 複製下列程式碼並將其貼入 psql 工作階段主控台中，來修改現有的 TLE 延伸模組。

```
SELECT pgtle.install_update_path
(
  'pg_distance',
  '0.1',
  '0.2',
  $_pg_tle_$
  CREATE OR REPLACE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8,
  norm int)
```

```

    RETURNS float8
    AS $$
        SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

CREATE OR REPLACE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2
float8)
    RETURNS float8
    AS $$
        SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

CREATE OR REPLACE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2
float8)
    RETURNS float8
    AS $$
        SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;
$_pg_tle_$
);

```

您會看到如下回應：

```

install_update_path
-----
 t
(1 row)

```

您可以將此版本的延伸模組設為預設版本，這樣資料庫使用者就不必在資料庫中建立或更新延伸模組時指定版本。

- 若要指定 TLE 延伸模組的修改版本 (版本 0.2) 為預設版本，請使用 `pgtle.set_default_version` 函數，如下列範例所示。

```
SELECT pgtle.set_default_version('pg_distance', '0.2');
```

如需此函數狀態的詳細資訊，請參閱 [pgtle.set_default_version](#)。

- 在程式碼就位之後，您可以使用 `ALTER EXTENSION ... UPDATE` 命令，以尋常方式更新已安裝的 TLE 延伸模組，如這裡所示：

```
ALTER EXTENSION pg_distance UPDATE;
```

從資料庫中捨棄您的 TLE 延伸模組

您可以採取您對其他 PostgreSQL 延伸模組所用的同一方式，使用 DROP EXTENSION 命令來捨棄 TLE 延伸模組。捨棄延伸模組並不會移除構成延伸模組的安裝檔案，因而允許使用者重新建立延伸模組。若要移除延伸模組及其安裝檔案，請執行下列兩步驟程序。

捨棄 TLE 延伸模組並移除其安裝檔案

1. 使用 psql 或其他用戶端工具，連線至 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --  
username=postgres --password --dbname=dbname
```

2. 捨棄延伸模組，如同您對任何 PostgreSQL 延伸模組所做一般。

```
DROP EXTENSION your-TLE-extension
```

例如，如果您如[範例：使用 SQL 建立受信任語言延伸模組](#)所述建立 pg_distance 延伸模組，則可以捨棄延伸模組，如下所示。

```
DROP EXTENSION pg_distance;
```

您會看到輸出，確認已捨棄延伸模組，如下所示。

```
DROP EXTENSION
```

此時，延伸模組在資料庫中不再處於作用中狀態。不過，其安裝檔案和控制檔案仍然可在資料庫中使用，因此資料庫使用者可以再次建立延伸模組 (如果想要的話)。

- 如果想要延伸模組檔案保持不變，以便資料庫使用者可以建立您的 TLE 延伸模組，您可以在此停止。
 - 如果想要移除所有構成延伸模組的檔案，請繼續下一個步驟。
3. 若要移除延伸模組的所有安裝檔案，請使用 pgtle.uninstall_extension 函數。此函數會移除延伸模組的所有程式碼和控制檔。

```
SELECT pgtle.uninstall_extension('your-tle-extension-name');
```

例如，若要移除所有 pg_distance 安裝檔案，請使用下列命令。

```
SELECT pgtle.uninstall_extension('pg_distance');
      uninstall_extension
-----
t
(1 row)
```

解除安裝適用於 PostgreSQL 的受信任語言延伸模組

如果您再也不要使用 TLE 建立自己的 TLE 延伸模組，則可以捨棄 `pg_tle` 延伸模組並移除所有成品。此動作包括捨棄資料庫中的任何 TLE 延伸模組，以及捨棄 `pgtle` 結構描述。

從資料庫中捨棄 **pg_tle** 延伸模組及其結構描述

1. 使用 `psql` 或其他用戶端工具，連線至 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. 從資料庫中捨棄 `pg_tle` 延伸模組。如果您自己的 TLE 延伸模組仍在資料庫中執行，您也需要捨棄這些延伸模組。若要這樣做，您可以使用 `CASCADE` 關鍵字，如下所示。

```
DROP EXTENSION pg_tle CASCADE;
```

如果 `pg_tle` 延伸模組在資料庫中仍未作用中，則您不需要使用 `CASCADE` 關鍵字。

3. 捨棄 `pgtle` 結構描述。此動作會移除資料庫中的所有管理函數。

```
DROP SCHEMA pgtle CASCADE;
```

此命令會在程序完成時傳回下列內容。

```
DROP SCHEMA
```

`pg_tle` 延伸模組、其結構描述和函數，以及所有成品都會遭到移除。若要使用 TLE 建立新的延伸模組，請再次完成設定程序。如需詳細資訊，請參閱 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)。

搭配您的延伸模組使用 PostgreSQL 掛鉤

掛鉤是 PostgreSQL 中提供的回呼機制，允許開發人員在一般資料庫操作期間呼叫自訂函數或其他常式。TLE 開發套件支援 PostgreSQL 掛鉤，以便您可以在執行時整合自訂函數與 PostgreSQL 行為。例如，您可以使用掛鉤，將身分驗證程序與您自己的自訂程式碼建立關聯，或因應您的特定需求修改查詢規劃和執行程序。

您的 TLE 延伸模組可以使用掛鉤。如果掛鉤在範圍內是全域的，則其在所有資料庫之中都適用。因此，如果您的 TLE 延伸模組使用全域掛鉤，則您需要在使用者可以存取的所有資料庫中建立 TLE 延伸模組。

當使用 `pg_tle` 延伸模組來建置您自己的受信任語言延伸模組時，您可以使用 SQL API 中可用的掛鉤來建置延伸模組的函數。您應該使用 `pg_tle` 註冊任何掛鉤。對於某些掛鉤，您可能還需要設定各種組態參數。例如，`passcode` 檢查掛鉤可以設為開啟、關閉或需要。如需可用 `pg_tle` 掛鉤之特定需求的詳細資訊，請參閱 [適用於 PostgreSQL 的受信任語言延伸模組的掛鉤參考](#)。

範例：建立使用 PostgreSQL 掛鉤的延伸模組

本節中討論的範例使用 PostgreSQL 掛鉤，來檢查在特定 SQL 操作期間提供的密碼，並防止資料庫使用者將其密碼設為 `password_check.bad_passwords` 資料表中包含的任何密碼。該資料表包含前十大最常用但容易破解的密碼選擇。

若要在您的 RDS for PostgreSQL 資料庫執行個體中設定此範例，您必須已安裝受信任語言延伸模組。如需詳細資訊，請參閱 [在您的 RDS for PostgreSQL 資料庫執行個體中設定受信任語言延伸模組](#)。

設定密碼檢查掛鉤範例

1. 使用 `psql` 連線到 RDS for PostgreSQL 資料庫執行個體。

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. 從 [密碼檢查掛鉤程式碼清單](#) 中複製程式碼，並將其貼入您的資料庫中。

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
  $pgtle_$
CREATE SCHEMA password_check;
REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
```



```
GRANT USAGE ON SCHEMA password_check TO PUBLIC;

CREATE TABLE password_check.bad_passwords (plaintext) AS
VALUES
  ('123456'),
  ('password'),
  ('12345678'),
  ('qwerty'),
  ('123456789'),
  ('12345'),
  ('1234'),
  ('111111'),
  ('1234567'),
  ('dragon');
CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

CREATE FUNCTION password_check.passcheck_hook(username text, password text,
password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
RETURNS void AS $$
  DECLARE
    invalid bool := false;
  BEGIN
    IF password_type = 'PASSWORD_TYPE_MD5' THEN
      SELECT EXISTS(
        SELECT 1
        FROM password_check.bad_passwords bp
        WHERE ('md5' || md5(bp.plaintext || username)) = password
      ) INTO invalid;
      IF invalid THEN
        RAISE EXCEPTION 'Cannot use passwords from the common password
dictionary';
      END IF;
    ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
      SELECT EXISTS(
        SELECT 1
        FROM password_check.bad_passwords bp
        WHERE bp.plaintext = password
      ) INTO invalid;
      IF invalid THEN
        RAISE EXCEPTION 'Cannot use passwords from the common common password
dictionary';
      END IF;
    END IF;
  END IF;
END
```

```

$$ LANGUAGE plpgsql SECURITY DEFINER;

GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);

```

將延伸模組載入資料庫後，您會看到如下輸出。

```

install_extension
-----
t
(1 row)

```

3. 在仍然連線到資料庫時，您現在可以建立延伸模組。

```
CREATE EXTENSION my_password_check_rules;
```

4. 您可以使用下列 `psql` 中繼命令，確認已在資料庫中建立延伸模組。

```

\dx
          List of installed extensions
  Name          | Version | Schema |
  Description
-----+-----+-----+
+-----+-----+-----+
my_password_check_rules | 1.0    | public | Prevent use of any of the top-ten
most common bad passwords
pg_tle          | 1.0.1  | pgtle  | Trusted-Language Extensions for
PostgreSQL
plpgsql        | 1.0    | pg_catalog | PL/pgSQL procedural language
(3 rows)

```

5. 開啟另一個要使用的終端機工作階段 AWS CLI。您需要修改自訂資料庫參數群組，以開啟密碼檢查掛鉤。若要這麼做，請使用 [modify-db-parameter-group](#) CLI 命令，如下列範例所示。

```

aws rds modify-db-parameter-group \
  --region aws-region \
  --db-parameter-group-name your-custom-parameter-group \
  --parameters
  "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"

```

在成功開啟參數時，您會看到如下輸出。

```
(
  "DBParameterGroupName": "docs-lab-parameters-for-tle"
}
```

可能需要幾分鐘，對參數群組設定所做的變更才會生效。不過，此參數是動態參數，因此您不需要重新啟動 RDS for PostgreSQL 資料庫執行個體，設定即可生效。

6. 開啟 `psql` 工作階段並查詢資料庫，以驗證密碼檢查掛鉤是否已開啟。

```
labdb=> SHOW pgtle.enable_password_check;
pgtle.enable_password_check
-----
on
(1 row)
```

密碼檢查掛鉤現在處於作用中狀態。您可以建立新角色並使用其中一個錯誤密碼來對它進行測試，如下列範例所示。

```
CREATE ROLE test_role PASSWORD 'password';
ERROR: Cannot use passwords from the common password dictionary
CONTEXT: PL/pgSQL function
password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
zone,boolean) line 21 at RAISE
SQL statement "SELECT password_check.passcheck_hook(
  $1::pg_catalog.text,
  $2::pg_catalog.text,
  $3::pgtle.password_types,
  $4::pg_catalog.timestampz,
  $5::pg_catalog.bool)"
```

為了方便閱讀，輸出已經過格式化處理。

下列範例顯示 `psql` 互動式中繼命令 `\password` 行為也受密碼檢查掛鉤的影響。

```
postgres=> SET password_encryption TO 'md5';
SET
postgres=> \password
Enter new password for user "postgres":*****
```

```

Enter it again:*****
ERROR: Cannot use passwords from the common password dictionary
CONTEXT: PL/pgSQL function
password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
zone,boolean) line 12 at RAISE
SQL statement "SELECT password_check.passcheck_hook($1::pg_catalog.text,
$2::pg_catalog.text, $3::pgtle.password_types, $4::pg_catalog.timestampz,
$5::pg_catalog.bool)"

```

如果需要，您可以捨棄此 TLE 延伸模組並解除安裝其來源檔案。如需詳細資訊，請參閱 [從資料庫中捨棄您的 TLE 延伸模組](#)。

密碼檢查掛鉤程式碼清單

此處顯示的範例程式碼定義了 my_password_check_rules TLE 延伸模組的規格。當您複製此程式碼並將其貼入資料庫中時，my_password_check_rules 延伸模組的程式碼會載入至資料庫，而且 password_check 掛鉤會進行註冊以供該延伸模組使用。

```

SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
  $_pgtle_$
CREATE SCHEMA password_check;
REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
GRANT USAGE ON SCHEMA password_check TO PUBLIC;

CREATE TABLE password_check.bad_passwords (plaintext) AS
VALUES
  ('123456'),
  ('password'),
  ('12345678'),
  ('qwerty'),
  ('123456789'),
  ('12345'),
  ('1234'),
  ('111111'),
  ('1234567'),
  ('dragon');
CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

CREATE FUNCTION password_check.passcheck_hook(username text, password text,
password_type pgtle.password_types, valid_until timestampz, valid_null boolean)

```

```
RETURNS void AS $$
DECLARE
    invalid bool := false;
BEGIN
    IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
            SELECT 1
            FROM password_check.bad_passwords bp
            WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
            RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
    ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
            SELECT 1
            FROM password_check.bad_passwords bp
            WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
            RAISE EXCEPTION 'Cannot use passwords from the common common password
dictionary';
        END IF;
    END IF;
END
$$ LANGUAGE plpgsql SECURITY DEFINER;

GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

在 TLE 中使用自訂資料類型

PostgreSQL 支援註冊新基底類型 (也稱為純量類型) 的命令，以便有效處理資料庫中複雜的資料結構。基底類型可讓您自訂在內部儲存資料的方式，以及如何將資料與外部文字表示法相互轉換。這些自訂資料類型在擴充 PostgreSQL 以支援功能定義域時很實用，因為這些定義域的內建類型 (如數字或文字) 無法提供足夠的搜尋語意。

RDS for PostgreSQL 可讓您在信任的語言延伸模組中建立自訂資料類型，並定義功能來支援這些新資料類型的 SQL 和索引操作。以下版本提供自訂資料類型：

- RDS for PostgreSQL 15.4 及更高的 15 版
- RDS for PostgreSQL 14.9 及更高的 14 版
- RDS for PostgreSQL 13.12 及更高的 13 版

如需詳細資訊，請參閱[信任的語言基底類型](#)。

適用於 PostgreSQL 的受信任語言延伸模組的函數參考

檢視下列有關適用於 PostgreSQL 的受信任語言延伸模組中提供之函數的參考文件。使用這些函數來安裝、註冊、更新和管理您的 TLE 延伸模組，亦即您使用受信任語言延伸模組開發套件所開發的 PostgreSQL 延伸模組。

主題

- [pgtle.available_extensions](#)
- [pgtle.available_extension_versions](#)
- [pgtle.extension_update_paths](#)
- [pgtle.install_extension](#)
- [pgtle.install_update_path](#)
- [pgtle.register_feature](#)
- [pgtle.register_feature_if_not_exists](#)
- [pgtle.set_default_version](#)
- [pgtle.uninstall_extension\(name\)](#)
- [pgtle.uninstall_extension\(name, version\)](#)
- [pgtle.uninstall_extension_if_exists](#)
- [pgtle.uninstall_update_path](#)
- [pgtle.uninstall_update_path_if_exists](#)
- [pgtle.unregister_feature](#)
- [pgtle.unregister_feature_if_exists](#)

pgtle.available_extensions

`pgtle.available_extensions` 函數是一個設定-傳回函數。它會傳回資料庫中所有可用的 TLE 延伸模組。每一傳回的列都包含單一 TLE 延伸模組的相關資訊。

函數原型

```
pgtle.available_extensions()
```

角色

無。

引數

無。

輸出

- `name` – TLE 延伸模組的名稱。
- `default_version` – 在未指定版本的情況下呼叫 `CREATE EXTENSION` 時使用的 TLE 延伸模組版本。
- `description` – 有關 TLE 延伸模組的詳細描述。

使用範例

```
SELECT * FROM pgtle.available_extensions();
```

pgtle.available_extension_versions

`available_extension_versions` 函數是一個設定-傳回函數。其會傳回一份清單，列出所有可用的 TLE 延伸模組及其版本。每一列都包含指定 TLE 延伸模組之特定版本的相關資訊，包括其是否需要特定角色。

函數原型

```
pgtle.available_extension_versions()
```

角色

無。

引數

無。

輸出

- `name` – TLE 延伸模組的名稱。
- `version` – TLE 延伸模組的版本。
- `superuser` – 對於 TLE 延伸模組，此值一律為 `false`。建立 TLE 延伸模組或更新它所需的許可，與在指定資料庫中建立其他物件所需的許可相同。
- `trusted` – 對於 TLE 延伸模組，此值一律為 `false`。
- `relocatable` – 對於 TLE 延伸模組，此值一律為 `false`。
- `schema` – 指定其中安裝 TLE 延伸模組的結構描述名稱。
- `requires` – 一種陣列，其中包含此 TLE 延伸模組所需之其他延伸模組的名稱。
- `description` – TLE 延伸模組的詳細描述。

如需輸出值的詳細資訊，請參閱 PostgreSQL 文件中的[將相關物件封裝為延伸模組 > 延伸模組檔案](#)。

使用範例

```
SELECT * FROM pgtle.available_extension_versions();
```

pgtle.extension_update_paths

`extension_update_paths` 函數是一個設定-傳回函數。其會傳回一份清單，列出 TLE 延伸模組的所有可能更新路徑。每一列都包含該 TLE 延伸模組的可用升級或降級。

函數原型

```
pgtle.extension_update_paths(name)
```

角色

無。

引數

`name` – 要從中取得升級路徑的 TLE 延伸模組名稱。

輸出

- `source` – 更新的來源版本。
- `target` – 更新的目標版本。

- `path` – 用來將 TLE 延伸模組從 `source` 版本更新至 `target` 版本的升級路徑，例如 `0.1--0.2`。

使用範例

```
SELECT * FROM pgtle.extension_update_paths('your-TLE');
```

pgtle.install_extension

`install_extension` 函數可讓您在資料庫中安裝構成 TLE 延伸模組的成品，然後您可以使用 `CREATE EXTENSION` 命令來建立它。

函數原型

```
pgtle.install_extension(name text, version text, description text, ext text, requires text[] DEFAULT NULL::text[])
```

角色

無。

引數

- `name` – TLE 延伸模組的名稱。呼叫 `CREATE EXTENSION` 時會使用此值。
- `version` – TLE 延伸模組的版本。
- `description` – 有關 TLE 延伸模組的詳細描述。此描述顯示在 `pgtle.available_extensions()` 的 `comment` 欄位中。
- `ext` – TLE 延伸模組的內容。此值包含函數之類的物件。
- `requires` – 針對此 TLE 延伸模組指定相依性的選用參數。pg_tle 延伸模組會自動新增為相依性。

其中許多引數與延伸模組控制檔中包含的引數相同，用於在 PostgreSQL 執行個體的檔案系統上安裝 PostgreSQL 延伸模組。如需詳細資訊，請參閱 PostgreSQL 文件中[將相關物件封裝為延伸模組](#)中的[延伸模組檔案](#)。

輸出

此函數會在成功時傳回 `OK`，以及在發生錯誤時傳回 `NULL`。

- `OK` – TLE 延伸模組已成功安裝在資料庫中。

- NULL – TLE 延伸模組未成功安裝在資料庫中。

使用範例

```
SELECT pgtle.install_extension(  
  'pg_tle_test',  
  '0.1',  
  'My first pg_tle extension',  
  $_pgtle_$  
  CREATE FUNCTION my_test()  
  RETURNS INT  
  AS $$  
    SELECT 42;  
  $$ LANGUAGE SQL IMMUTABLE;  
  $_pgtle_$  
);
```

pgtle.install_update_path

`install_update_path` 函數會提供 TLE 延伸模組的兩個不同版本之間的更新路徑。此函數允許 TLE 延伸模組的使用者透過使用 `ALTER EXTENSION ... UPDATE` 語法更新其版本。

函數原型

```
pgtle.install_update_path(name text, fromvers text, tovers text, ext text)
```

角色

`pgtle_admin`

引數

- `name` – TLE 延伸模組的名稱。呼叫 `CREATE EXTENSION` 時會使用此值。
- `fromvers` – 用於升級之 TLE 延伸模組的來源版本。
- `tovers` – 用於升級之 TLE 延伸模組的目標版本。
- `ext` – 更新的內容。此值包含函數之類的物件。

輸出

無。

使用範例

```
SELECT pgtle.install_update_path('pg_tle_test', '0.1', '0.2',
    $_pgtle_$
    CREATE OR REPLACE FUNCTION my_test()
    RETURNS INT
    AS $$
    SELECT 21;
    $$ LANGUAGE SQL IMMUTABLE;
    $_pgtle_$
);
```

pgtle.register_feature

函數 `register_feature` 會將指定的內部 PostgreSQL 功能新增至 `pgtle.feature_info` 資料表。PostgreSQL 掛鉤是內部 PostgreSQL 功能的範例。受信任語言延伸模組開發套件支援使用 PostgreSQL 掛鉤。目前，此函數支援下列功能。

- `passcheck` – 使用自訂 PostgreSQL 密碼檢查行為的程序或函數註冊密碼檢查掛鉤。

函數原型

```
pgtle.register_feature(proc regproc, feature pg_tle_feature)
```

角色

`pgtle_admin`

引數

- `proc` – 要用於功能的預存程序或函數名稱。
- `feature` – 要向函數註冊的 `pg_tle` 功能 (例如 `passcheck`) 名稱。

輸出

無。

使用範例

```
SELECT pgtle.register_feature('pw_hook', 'passcheck');
```

pgtle.register_feature_if_not_exists

`pgtle.register_feature_if_not_exists` 函數會將指定的 PostgreSQL 功能新增至 `pgtle.feature_info` 資料表，並識別 TLE 延伸模組或其他使用該功能的程序或函數。如需掛鉤和受信任語言延伸模組的詳細資訊，請參閱 [搭配您的延伸模組使用 PostgreSQL 掛鉤](#)。

函數原型

```
pgtle.register_feature_if_not_exists(proc regproc, feature pgtle_feature)
```

角色

`pgtle_admin`

引數

- `proc` – 預存程序或函數的名稱，此預存程序或函數包含可用作 TLE 延伸功能之功能的邏輯 (程式碼)。例如，`pw_hook` 程式碼。
- `feature` – 要針對 TLE 函數註冊的 PostgreSQL 功能名稱。目前，唯一可用的功能為 `passcheck` 掛鉤。如需詳細資訊，請參閱 [密碼檢查掛鉤 \(passcheck\)](#)。

輸出

在針對指定的延伸模組註冊功能之後傳回 `true`。如果此功能已註冊，則會傳回 `false`。

使用範例

```
SELECT pgtle.register_feature_if_not_exists('pw_hook', 'passcheck');
```

pgtle.set_default_version

`set_default_version` 函數可讓您針對 TLE 延伸模組指定 `default_version`。您可以使用此函數來定義升級路徑，並將該版本指定為 TLE 延伸模組的預設版本。當資料庫使用者在 `CREATE EXTENSION` 和 `ALTER EXTENSION ... UPDATE` 命令中指定您的 TLE 延伸模組時，該版本的 TLE 延伸模組就會在該使用者的資料庫中建立。

成功時此函數會傳回 `true`。如果 `name` 引數中指定的 TLE 延伸模組不存在，函數會傳回錯誤。同樣地，如果 TLE 延伸模組的 `version` 不存在，則其會傳回錯誤。

函數原型

```
pgtle.set_default_version(name text, version text)
```

角色

pgtle_admin

引數

- `name` – TLE 延伸模組的名稱。呼叫 CREATE EXTENSION 時會使用此值。
- `version` – 要設定預設值的 TLE 延伸模組版本。

輸出

- `true` – 設定預設版本成功時，函數會傳回 `true`。
- `ERROR` – 如果具有指定名稱或版本的 TLE 延伸模組不存在，則會傳回錯誤訊息。

使用範例

```
SELECT * FROM pgtle.set_default_version('my-extension', '1.1');
```

pgtle.uninstall_extension(name)

`uninstall_extension` 函數會從資料庫中移除 TLE 延伸模組的所有版本。此函數可防止未來呼叫 CREATE EXTENSION 時安裝 TLE 延伸模組。如果 TLE 延伸模組不存在於資料庫中，則會引發錯誤。

`uninstall_extension` 函數不會捨棄資料庫中目前作用中的 TLE 延伸模組。若要移除目前作用中的 TLE 延伸模組，您必須明確地呼叫 DROP EXTENSION 才能將其移除。

函數原型

```
pgtle.uninstall_extension(extname text)
```

角色

pgtle_admin

引數

- `extname` – 要解除安裝的 TLE 延伸模組名稱。此名稱與搭配 `CREATE EXTENSION` 用來載入 TLE 延伸模組，以用於指定資料庫的名稱相同。

輸出

無。

使用範例

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test');
```

`pgtle.uninstall_extension(name, version)`

`uninstall_extension(name, version)` 函數會從資料庫中移除 TLE 延伸模組的指定版本。此函數可防止 `CREATE EXTENSION` 和 `ALTER EXTENSION` 將 TLE 延伸模組安裝或更新為指定版本。此函數也會移除 TLE 延伸模組指定版本的所有更新路徑。如果 TLE 延伸模組目前在資料庫中作用中，則此函數無法將其解除安裝。您必須明確地呼叫 `DROP EXTENSION` 才能移除 TLE 延伸模組。若要解除安裝 TLE 延伸模組的所有版本，請參閱 [pgtle.uninstall_extension\(name\)](#)。

函數原型

```
pgtle.uninstall_extension(extname text, version text)
```

角色

`pgtle_admin`

引數

- `extname` – TLE 延伸模組的名稱。呼叫 `CREATE EXTENSION` 時會使用此值。
- `version` – 要從資料庫中解除安裝的 TLE 延伸模組版本。

輸出

無。

使用範例

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test', '0.2');
```

pgtle.uninstall_extension_if_exists

`uninstall_extension_if_exists` 函數會從指定資料庫中移除 TLE 延伸模組的所有版本。如果 TLE 延伸模組不存在，則函數會以無訊息方式返回 (不會引發任何錯誤訊息)。如果指定的延伸模組目前在資料庫內作用中，則此函數不會捨棄它。您必須明確地呼叫 `DROP EXTENSION` 來移除 TLE 延伸模組，然後再使用此函數來解除安裝其成品。

函數原型

```
pgtle.uninstall_extension_if_exists(extname text)
```

角色

`pgtle_admin`

引數

- `extname` – TLE 延伸模組的名稱。呼叫 `CREATE EXTENSION` 時會使用此值。

輸出

在解除安裝指定的延伸模組之後，`uninstall_extension_if_exists` 函數會傳回 `true`。如果指定的延伸模組不存在，函數會傳回 `false`。

- `true` – 在解除安裝 TLE 延伸模組之後傳回 `true`。
- `false` – 當 TLE 延伸模組不存在於資料庫中時傳回 `false`。

使用範例

```
SELECT * FROM pgtle.uninstall_extension_if_exists('pg_tle_test');
```

pgtle.uninstall_update_path

`uninstall_update_path` 函數會從 TLE 延伸模組中移除特定的更新路徑。這樣可以防止 `ALTER EXTENSION ... UPDATE TO` 將其用作更新路徑。

如果 TLE 延伸模組目前正由這個更新路徑上的其中一個版本使用，則其仍會保留在資料庫中。

如果指定的更新路徑不存在，此函數會引發錯誤。

函數原型

```
pgtle.uninstall_update_path(extname text, fromvers text, tovers text)
```

角色

pgtle_admin

引數

- `extname` – TLE 延伸模組的名稱。呼叫 CREATE EXTENSION 時會使用此值。
- `fromvers` – 更新路徑上使用之 TLE 延伸模組的來源版本。
- `tovers` – 更新路徑上使用之 TLE 延伸模組的目標版本。

輸出

無。

使用範例

```
SELECT * FROM pgtle.uninstall_update_path('pg_tle_test', '0.1', '0.2');
```

pgtle.uninstall_update_path_if_exists

`uninstall_update_path_if_exists` 函數類似於 `uninstall_update_path`，在這裡其會從 TLE 延伸模組中移除指定的更新路徑。不過，如果更新路徑不存在，則此函數不會引發錯誤訊息。反之，此函數會傳回 `false`。

函數原型

```
pgtle.uninstall_update_path_if_exists(extname text, fromvers text, tovers text)
```

角色

pgtle_admin

引數

- `extname` – TLE 延伸模組的名稱。呼叫 `CREATE EXTENSION` 時會使用此值。
- `fromvers` – 更新路徑上使用之 TLE 延伸模組的來源版本。
- `tovers` – 更新路徑上使用之 TLE 延伸模組的目標版本。

輸出

- `true` – 函數已成功更新 TLE 延伸模組的路徑。
- `false` – 函數無法更新 TLE 延伸模組的路徑。

使用範例

```
SELECT * FROM pgtle.uninstall_update_path_if_exists('pg_tle_test', '0.1', '0.2');
```

pgtle.unregister_feature

`unregister_feature` 函數會提供一種方法，來刪除已註冊為使用 `pg_tle` 功能 (例如掛鉤) 的函數。如需註冊功能的相關資訊，請參閱 [pgtle.register_feature](#)。

函數原型

```
pgtle.unregister_feature(proc regproc, feature pg_tle_features)
```

角色

`pgtle_admin`

引數

- `proc` – 要向 `pg_tle` 功能註冊的預存函數名稱。
- `feature` – 要向函數註冊的 `pg_tle` 功能名稱。例如，`passcheck` 是一種功能，可以將其註冊以供您開發的受信任語言延伸模組使用。如需更多詳細資訊，請參閱 [密碼檢查掛鉤 \(passcheck\)](#)。

輸出

無。

使用範例

```
SELECT * FROM pgtle.unregister_feature('pw_hook', 'passcheck');
```

pgtle.unregister_feature_if_exists

`unregister_feature` 函數會提供一種方法，來刪除已註冊為使用 `pg_tle` 功能 (例如掛鉤) 的函數。如需更多詳細資訊，請參閱 [搭配您的延伸模組使用 PostgreSQL 掛鉤](#)。在成功取消註冊功能之後傳回 `true`。如果功能未註冊，則會傳回 `false`。

如需為 TLE 延伸模組註冊 `pg_tle` 功能的相關資訊，請參閱 [pgtle.register_feature](#)。

函數原型

```
pgtle.unregister_feature_if_exists('proc regproc', 'feature pg_tle_features')
```

角色

`pgtle_admin`

引數

- `proc` – 已註冊來包含 `pg_tle` 功能的預存函數名稱。
- `feature` – 已向受信任語言延伸模組註冊的 `pg_tle` 功能名稱。

輸出

傳回 `true` 或 `false`，如下所示。

- `true` – 函數已成功從延伸模組中取消註冊功能。
- `false` – 函數無法從 TLE 延伸模組中取消註冊功能。

使用範例

```
SELECT * FROM pgtle.unregister_feature_if_exists('pw_hook', 'passcheck');
```

適用於 PostgreSQL 的受信任語言延伸模組的掛鉤參考

適用於 PostgreSQL 的受信任語言延伸模組支援 PostgreSQL 掛鉤。掛鉤是開發人員可用於擴充 PostgreSQL 核心功能的內部招呼機制。透過使用掛鉤，開發人員可以實作自己的函數或程序，以在各

種資料庫操作期間使用，從而以某種方式修改 PostgreSQL 的行為。例如，您可以使用 `passcheck` 掛鉤，來自訂 PostgreSQL 在為使用者 (角色) 建立或變更密碼時如何處理所提供的密碼。

檢視下列文件，以了解 TLE 延伸模組可用的掛鉤。

主題

- [密碼檢查掛鉤 \(passcheck\)](#)

密碼檢查掛鉤 (passcheck)

`passcheck` 掛鉤用來針對下列 SQL 命令和 `psql` 中繼命令自訂密碼檢查過程中的 PostgreSQL 行為。

- `CREATE ROLE username ...PASSWORD` – 如需詳細資訊，請參閱 PostgreSQL 文件中的 [CREATE ROLE](#)。
- `ALTER ROLE username ...PASSWORD` – 如需詳細資訊，請參閱 PostgreSQL 文件中的 [ALTER ROLE](#)。
- `\password username` – 此互動式 `psql` 中繼命令透過在透明地使用 `ALTER ROLE ... PASSWORD` 語法之前對密碼進行雜湊處理，來安全地變更所指定使用者的密碼。中繼命令是 `ALTER ROLE ... PASSWORD` 命令的安全包裝函式，因此掛鉤適用於 `psql` 中繼命令的行為。

如需範例，請參閱 [密碼檢查掛鉤程式碼清單](#)。

函數原型

```
passcheck_hook(username text, password text, password_type pgtle.password_types,
               valid_until timestamptz, valid_null boolean)
```

引數

`passcheck` 掛鉤函數採用下列引數：

- `username` – 設定密碼之角色 (使用者名稱) 的名稱 (以文字形式表示)。
- `password` – 純文字或雜湊密碼。輸入的密碼應符合 `password_type` 中指定的類型。
- `password_type` – 指定密碼的 `pgtle.password_type` 格式。此格式可以是下列其中一個選項：
 - `PASSWORD_TYPE_PLAINTEXT` – 純文字密碼。

- `PASSWORD_TYPE_MD5` – 使用 MD5 (訊息摘要 5) 演算法進行雜湊處理的密碼。
- `PASSWORD_TYPE_SCRAM_SHA_256` – 使用 SCRAM-SHA-256 演算法進行雜湊處理的密碼。
- `valid_until` – 指定密碼變成無效的時間。此為選用引數。如果使用此引數，請將時間指定為 `timestampz` 值。
- `valid_null` – 如果將此布林值設為 `true`，則 `valid_until` 選項會設為 `NULL`。

組態

此函數 `pgtle.enable_password_check` 可控制 `passcheck` 掛鉤是否作用中。`passcheck` 掛鉤有三種可能的設定。

- `off` – 關閉 `passcheck` 密碼檢查掛鉤。這是預設值。
- `on` – 開啟 `passcode` 密碼檢查掛鉤，以便針對資料表檢查密碼。
- `require` – 需要定義密碼檢查掛鉤。

使用須知

若要開啟或關閉 `passcheck` 掛鉤，您需要針對 RDS for PostgreSQL 資料庫執行個體修改自訂資料庫參數群組。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --region aws-region \  
  --db-parameter-group-name your-custom-parameter-group \  
  --parameters  
  "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

在Windows中：

```
aws rds modify-db-parameter-group ^  
  --region aws-region ^  
  --db-parameter-group-name your-custom-parameter-group ^  
  --parameters  
  "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

使用 AWS 開發套件的 Amazon RDS 的程式碼範例

下列程式碼範例顯示如何搭配 AWS 軟體開發套件 (SDK) 使用 Amazon RDS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

Scenarios (案例) 是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

Cross-service examples (跨服務範例) 是跨多個 AWS 服務執行的應用程式範例。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含入門相關資訊和舊版 SDK 的詳細資訊。

開始使用

您好 Amazon RDS

下列程式碼範例說明如何開始使用 Amazon RDS。

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
using Amazon.RDS.Model;

namespace RDSActions;

public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();
```

```
        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\\tDB name: {instance.DBName}");
            Console.WriteLine($"\\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [DescribeDBInstances](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

C MakeLists.txt 的 CMake 文件的代碼。

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
```

```
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
  # Copy relevant AWS SDK for C++ libraries into the current binary directory
  for running and debugging.

  # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
  may need to uncomment this
                                # and set the proper subdirectory to the
  executables' location.

  AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
  hello_rds.cpp)

target_link_libraries(${PROJECT_NAME}
  ${AWSSDK_LINK_LIBRARIES})
```

hello_rds.cpp 來源檔案的程式碼。

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
 * A "Hello Rds" starter application which initializes an Amazon Relational
 * Database Service (Amazon RDS) client and
 * describes the Amazon RDS instances.
 *
 * main function
 *
 * Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }

            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
```



```
        instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
    }
    marker = outcome.GetResult().GetMarker();
} else {
    result = 1;
    std::cerr << "Error with RDS::DescribeDBInstances. "
              << outcome.GetError().GetMessage()
              << std::endl;

    break;
}
} while (!marker.empty());


std::cout << instanceDBIDs.size() << " RDS instances found." <<
std::endl;
for (auto &instanceDBID: instanceDBIDs) {
    std::cout << "    Instance: " << instanceDBID << std::endl;
}
}

Aws::ShutdownAPI(options); // Should only be called once.
return result;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [DescribeDBInstances](#)。

Go

SDK for Go V2

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
package main

import (
    "context"
    "fmt"
```

```
"github.com/aws/aws-sdk-go-v2/aws"  
"github.com/aws/aws-sdk-go-v2/config"  
"github.com/aws/aws-sdk-go-v2/service/rds"  
)  
  
// main uses the AWS SDK for Go V2 to create an Amazon Relational Database  
// Service (Amazon RDS)  
// client and list up to 20 DB instances in your account.  
// This example uses the default settings specified in your shared credentials  
// and config files.  
func main() {  
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())  
    if err != nil {  
        fmt.Println("Couldn't load default configuration. Have you set up your AWS  
account?")  
        fmt.Println(err)  
        return  
    }  
    rdsClient := rds.NewFromConfig(sdkConfig)  
    const maxInstances = 20  
    fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)  
    output, err := rdsClient.DescribeDBInstances(context.TODO(),  
        &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})  
    if err != nil {  
        fmt.Printf("Couldn't list DB instances: %v\n", err)  
        return  
    }  
    if len(output.DBInstances) == 0 {  
        fmt.Println("No DB instances found.")  
    } else {  
        for _, instance := range output.DBInstances {  
            fmt.Printf("DB instance %v has database %v.\n",  
*instance.DBInstanceIdentifier,  
                *instance.DBName)  
        }  
    }  
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [DescribeDBInstances](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
            .region(region)
            .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
```

```
        List<DBInstance> instanceList = response.dbInstances();
        for (DBInstance instance : instanceList) {
            System.out.println("Instance ARN is: " +
instance.dbInstanceArn());
            System.out.println("The Engine is " + instance.engine());
            System.out.println("Connection endpoint is" +
instance.endpoint().address());
        }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [DescribeDBInstances](#)。

程式碼範例

- [使用 AWS 開發套件為 Amazon RDS 執行動作](#)
 - [搭CreateDBInstance配 AWS 開發套件或 CLI 使用](#)
 - [搭CreateDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
 - [搭CreateDBSnapshot配 AWS 開發套件或 CLI 使用](#)
 - [搭DeleteDBInstance配 AWS 開發套件或 CLI 使用](#)
 - [搭DeleteDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeAccountAttributes配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeDBEngineVersions配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeDBInstances配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeDBParameterGroups配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeDBParameters配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeDBSnapshots配 AWS 開發套件或 CLI 使用](#)
 - [搭DescribeOrderableDBInstanceOptions配 AWS 開發套件或 CLI 使用](#)
 - [搭GenerateRDSAuthToken配 AWS 開發套件或 CLI 使用](#)
 - [搭ModifyDBInstance配 AWS 開發套件或 CLI 使用](#)

- [搭ModifyDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
- [搭RebootDBInstance配 AWS 開發套件或 CLI 使用](#)
- [使用 AWS 開發套件的 Amazon RDS 案例](#)
 - [使用開 AWS 發套件開始使用 Amazon RDS 資料庫執行個體](#)
- [使 AWS 用開發套件的 Amazon RDS 無伺服器範例](#)
 - [在 Lambda 函數中連接到 Amazon RDS 數據庫](#)
- [使 AWS 用開發套件的 Amazon RDS 跨服務範例](#)
 - [建立 Aurora 無伺服器工作項目追蹤器](#)

使用 AWS 開發套件為 Amazon RDS 執行動作

下列程式碼範例示範如何使用 AWS 開發套件執行個別 Amazon RDS 動作。這些摘錄會呼叫 Amazon RDS API，是必須在內容中執行之大型程式的程式碼摘錄。每個範例都包含一個連結 GitHub，您可以在其中找到設定和執行程式碼的指示。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 [Amazon Relational Database Service \(Amazon RDS\) API 參考](#)。

範例

- [搭CreateDBInstance配 AWS 開發套件或 CLI 使用](#)
- [搭CreateDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
- [搭CreateDBSnapshot配 AWS 開發套件或 CLI 使用](#)
- [搭DeleteDBInstance配 AWS 開發套件或 CLI 使用](#)
- [搭DeleteDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeAccountAttributes配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeDBEngineVersions配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeDBInstances配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeDBParameterGroups配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeDBParameters配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeDBSnapshots配 AWS 開發套件或 CLI 使用](#)
- [搭DescribeOrderableDBInstanceOptions配 AWS 開發套件或 CLI 使用](#)
- [搭GenerateRDSAuthToken配 AWS 開發套件或 CLI 使用](#)

- [搭ModifyDBInstance配 AWS 開發套件或 CLI 使用](#)
- [搭ModifyDBParameterGroup配 AWS 開發套件或 CLI 使用](#)
- [搭RebootDBInstance配 AWS 開發套件或 CLI 使用](#)

搭CreateDBInstance配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateDBInstance。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
/// to determine when the DB instance is ready to use.
/// </summary>
/// <param name="dbName">Name for the DB instance.</param>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
/// <param name="dbEngine">The engine for the DB instance.</param>
/// <param name="dbEngineVersion">Version for the DB instance.</param>
/// <param name="instanceClass">Class for the DB instance.</param>
/// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
/// <param name="adminName">Admin user name.</param>
```

```
/// <param name="adminPassword">Admin user password.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
    string parameterGroupName, string dbEngine, string dbEngineVersion,
    string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
{
    var response = await _amazonRDS.CreateDBInstanceAsync(
        new CreateDBInstanceRequest()
        {
            DBName = dbName,
            DBInstanceIdentifier = dbInstanceIdentifier,
            DBParameterGroupName = parameterGroupName,
            Engine = dbEngine,
            EngineVersion = dbEngineVersion,
            DBInstanceClass = instanceClass,
            AllocatedStorage = allocatedStorage,
            MasterUsername = adminName,
            MasterUserPassword = adminPassword
        });

    return response.DBInstance;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [CreateDBInstance](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";
```

```
Aws::RDS::RDSClient client(clientConfig);

Aws::RDS::Model::CreateDBInstanceRequest request;
request.SetDBName(DB_NAME);
request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
request.SetEngine(engineVersion.GetEngine());
request.SetEngineVersion(engineVersion.GetEngineVersion());
request.SetDBInstanceClass(dbInstanceClass);
request.SetStorageType(DB_STORAGE_TYPE);
request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
request.SetMasterUsername(administratorName);
request.SetMasterUserPassword(administratorPassword);

Aws::RDS::Model::CreateDBInstanceOutcome outcome =
    client.CreateDBInstance(request);

if (outcome.IsSuccess()) {
    std::cout << "The DB instance creation has started."
              << std::endl;
}
else {
    std::cerr << "Error with RDS::CreateDBInstance. "
              << outcome.GetError().GetMessage()
              << std::endl;
    cleanUpResources(PARAMETER_GROUP_NAME, "", client);
    return false;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [CreateDBInstance](#)。

CLI

AWS CLI

若要建立資料庫執行個體

下列create-db-instance範例會使用必要的選項來啟動新的資料庫執行個體。

```
aws rds create-db-instance \  
    --db-instance-identifier test-mysql-instance \  
    --db-instance-class db.t3.micro \  
    --engine mysql \  
    --engine-version 5.7.27 \  
    --db-parameter-group-name default:mysql5.7.27 \  
    --master-username testuser \  
    --master-user-password testpassword \  
    --allocated-storage 5 \  
    --storage-type gp2 \  
    --db-name testdb \  
    --no-promote-to-master \  
    --no-copy-restore-to-instance \  
    --no-copy-restore-to-restore \  
    --no-copy-restore-to-restore-from-instance \  
    --no-copy-restore-to-restore-from-instance-from-instance \  
    --no-copy-restore-to-restore-from-instance-from-instance-from-instance
```



```
--db-instance-class db.t3.micro \  
--engine mysql \  
--master-username admin \  
--master-user-password secret99 \  
--allocated-storage 20
```

輸出：

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "test-mysql-instance",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "creating",  
    "MasterUsername": "admin",  
    "AllocatedStorage": 20,  
    "PreferredBackupWindow": "12:55-13:25",  
    "BackupRetentionPeriod": 1,  
    "DBSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-12345abc",  
        "Status": "active"  
      }  
    ],  
    "DBParameterGroups": [  
      {  
        "DBParameterGroupName": "default.mysql5.7",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "DBSubnetGroup": {  
      "DBSubnetGroupName": "default",  
      "DBSubnetGroupDescription": "default",  
      "VpcId": "vpc-2ff2ff2f",  
      "SubnetGroupStatus": "Complete",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-#####",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2c"  
          },  
          "SubnetStatus": "Active"  
        }  
      ]  
    }  
  }  
}
```

```
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
      "SubnetStatus": "Active"
    }
  ]
},
"PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
"PendingModifiedValues": {
  "MasterUserPassword": "*****"
},
"MultiAZ": false,
"EngineVersion": "5.7.22",
"AutoMinorVersionUpgrade": true,
"ReadReplicaDBInstanceIdentifiers": [],
"LicenseModel": "general-public-license",
"OptionGroupMemberships": [
  {
    "OptionGroupName": "default:mysql-5-7",
    "Status": "in-sync"
  }
],
"PubliclyAccessible": true,
"StorageType": "gp2",
"DbInstancePort": 0,
"StorageEncrypted": false,
"DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
```

```
    "CACertificateIdentifier": "rds-ca-2019",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": []
  }
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的建立 Amazon RDS 資料庫執行個體](#)。

- 如需 API 詳細資訊，請參閱 [命令參考中 AWS CLI 的 CreateDBInstance](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {
  RdsClient *rds.Client
}

// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
  dbEngine string, dbEngineVersion string, parameterGroupName string,
  dbInstanceClass string,
  storageType string, allocatedStorage int32, adminName string, adminPassword
  string) (
  *types.DBInstance, error) {
```

```
output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
&rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:              aws.String(dbEngine),
  EngineVersion:      aws.String(dbEngineVersion),
  DBInstanceClass:    aws.String(dbInstanceClass),
  StorageType:        aws.String(storageType),
  AllocatedStorage:   aws.Int32(allocatedStorage),
  MasterUsername:     aws.String(adminName),
  MasterUserPassword: aws.String(adminPassword),
})
if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
} else {
  return output.DBInstance, nil
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [CreateDBInstance](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
import com.google.gson.Gson;
import
  software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
```

```
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For more details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating\_how-services-use-secrets\_RS.html
 *
 */

public class CreateDBInstance {
    public static long sleepTime = 20;

    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <dbInstanceIdentifier> <dbName> <secretName>

            Where:
                dbInstanceIdentifier - The database instance identifier.\s
                dbName - The database name.\s
    }
}
```

```
        secretName - The name of the AWS Secrets Manager secret that
contains the database credentials."
        """;

    if (args.length != 3) {
        System.out.println(usage);
        System.exit(1);
    }

    String dbInstanceIdentifier = args[0];
    String dbName = args[1];
    String secretName = args[2];
    Gson gson = new Gson();
    User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
    Region region = Region.US_WEST_2;
    RdsClient rdsClient = RdsClient.builder()
        .region(region)
        .build();

    createDatabaseInstance(rdsClient, dbInstanceIdentifier, dbName,
user.getUsername(), user.getPassword());
    waitForInstanceReady(rdsClient, dbInstanceIdentifier);
    rdsClient.close();
}

private static SecretsManagerClient getSecretClient() {
    Region region = Region.US_WEST_2;
    return SecretsManagerClient.builder()
        .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
        .build();
}

private static String getSecretValues(String secretName) {
    SecretsManagerClient secretClient = getSecretClient();
    GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
        .secretId(secretName)
        .build();

    GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
    return valueResponse.secretString();
}
```

```
}

public static void createDatabaseInstance(RdsClient rdsClient,
    String dbInstanceIdentifier,
    String dbName,
    String userName,
    String userPassword) {

    try {
        CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .allocatedStorage(100)
            .dbName(dbName)
            .engine("mysql")
            .dbInstanceClass("db.m4.large")
            .engineVersion("8.0")
            .storageType("standard")
            .masterUsername(userName)
            .masterUserPassword(userPassword)
            .build();

        CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
        System.out.println("The status is " +
response.dbInstance().dbInstanceStatus());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Waits until the database instance is available.
public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
    boolean instanceReady = false;
    String instanceReadyStr;
    System.out.println("Waiting for instance to become available.");
    try {
        DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .build();
```

```
// Loop until the cluster is ready.
while (!instanceReady) {
    DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
    List<DBInstance> instanceList = response.dbInstances();
    for (DBInstance instance : instanceList) {
        instanceReadyStr = instance.dbInstanceStatus();
        if (instanceReadyStr.contains("available"))
            instanceReady = true;
        else {
            System.out.print(".");
            Thread.sleep(sleepTime * 1000);
        }
    }
}
System.out.println("Database instance is available!");

} catch (RdsException | InterruptedException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [CreateDBInstance](#)。

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
suspend fun createDatabaseInstance(
    dbInstanceIdentifierVal: String?,
    dbNameVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
```



```
) {
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNameVal
            engine = "mysql"
            dbInstanceClass = "db.m4.large"
            engineVersion = "8.0"
            storageType = "standard"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the database instance is available.
suspend fun waitForInstanceReady(dbInstanceIdentifierVal: String?) {
    val sleepTime: Long = 20
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        while (!instanceReady) {
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        instanceReady = true
                    } else {
                        println("...$instanceReadyStr")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}
println("Database instance is available!")
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的 AWS SDK API 參考》中的 [CreateDBInstance](#)。

PHP

適用於 PHP 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;

$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$dbClass = 'db.t2.micro';
$storage = 5;
$engine = 'MySQL';
$username = 'MyUser';
$password = 'MyPassword';
```

```
try {
    $result = $rdsClient->createDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBInstanceClass' => $dbClass,
        'AllocatedStorage' => $storage,
        'Engine' => $engine,
        'MasterUsername' => $username,
        'MasterUserPassword' => $password,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for PHP API 參考》中的 [CreateDBInstance](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
```

```
    """
    rds_client = boto3.client("rds")
    return cls(rds_client)

def create_db_instance(
    self,
    db_name,
    instance_id,
    parameter_group_name,
    db_engine,
    db_engine_version,
    instance_class,
    storage_type,
    allocated_storage,
    admin_name,
    admin_password,
):
    """
    Creates a DB instance.

    :param db_name: The name of the database that is created in the DB
instance.
    :param instance_id: The ID to give the newly created DB instance.
    :param parameter_group_name: A parameter group to associate with the DB
instance.
    :param db_engine: The database engine of a database to create in the DB
instance.
    :param db_engine_version: The engine version for the created database.
    :param instance_class: The DB instance class for the newly created DB
instance.
    :param storage_type: The storage type of the DB instance.
    :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
    :param admin_name: The name of the admin user for the created database.
    :param admin_password: The admin password for the created database.
    :return: Data about the newly created DB instance.
    """
    try:
        response = self.rds_client.create_db_instance(
            DBName=db_name,
            DBInstanceIdentifier=instance_id,
            DBParameterGroupName=parameter_group_name,
            Engine=db_engine,
```

```
        EngineVersion=db_engine_version,
        DBInstanceClass=instance_class,
        StorageType=storage_type,
        AllocatedStorage=allocated_storage,
        MasterUsername=admin_name,
        MasterUserPassword=admin_password,
    )
    db_inst = response["DBInstance"]
except ClientError as err:
    logger.error(
        "Couldn't create DB instance %s. Here's why: %s: %s",
        instance_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return db_inst
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [CreateDBInstance](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭 `CreateDBParameterGroup` 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 `CreateDBParameterGroup`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
/// to determine when the DB parameter group is ready to use.
/// </summary>
/// <param name="name">Name of the DB parameter group.</param>
/// <param name="family">Family of the DB parameter group.</param>
/// <param name="description">Description of the DB parameter group.</param>
/// <returns>The new DB parameter group.</returns>
public async Task<DBParameterGroup> CreateDBParameterGroup(
    string name, string family, string description)
{
    var response = await _amazonRDS.CreateDBParameterGroupAsync(
        new CreateDBParameterGroupRequest()
        {
            DBParameterGroupName = name,
            DBParameterGroupFamily = family,
            Description = description
        });
    return response.DBParameterGroup;
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for .NET API 參考ParameterGroup中的[創建數據庫](#)。

C++

適用於 C++ 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

Aws::RDS::Model::CreateDBParameterGroupRequest request;
request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
request.SetDBParameterGroupFamily(dbParameterGroupFamily);
request.SetDescription("Example parameter group.");

Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
    client.CreateDBParameterGroup(request);

if (outcome.IsSuccess()) {
    std::cout << "The DB parameter group was successfully created."
              << std::endl;
}
else {
    std::cerr << "Error with RDS::CreateDBParameterGroup. "
              << outcome.GetError().GetMessage()
              << std::endl;
    return false;
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for C++ API 參考ParameterGroup中的[創建數據庫](#)。

CLI

AWS CLI

建立資料庫參數群組的步驟

下列create-db-parameter-group範例會建立資料庫參數群組。

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL5.6 \  
  --description "My new parameter group"
```

輸出：

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbparametergroup",  
    "DBParameterGroupFamily": "mysql5.6",  
    "Description": "My new parameter group",  
    "DBParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:pg:mydbparametergroup"  
  }  
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的建立資料庫參數群組](#)。

- 有關 API 的詳細信息，請參閱AWS CLI 命令參考ParameterGroup中的 [創建數據庫](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {  
  RdsClient *rds.Client
```



```
}

// CreateParameterGroup creates a DB parameter group that is based on the
// specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
    parameterGroupName string, parameterGroupFamily string, description string) (
    *types.DBParameterGroup, error) {

    output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
        &rds.CreateDBParameterGroupInput{
            DBParameterGroupName:    aws.String(parameterGroupName),
            DBParameterGroupFamily: aws.String(parameterGroupFamily),
            Description:            aws.String(description),
        })
    if err != nil {
        log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
        return nil, err
    } else {
        return output.DBParameterGroup, err
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Go API 參考ParameterGroup中的[創建數據庫](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
public static void createDBParameterGroup(RdsClient rdsClient, String
    dbGroupName, String dbParameterGroupFamily) {
    try {
```

```
        CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
    .dbParameterGroupName(dbGroupName)
    .dbParameterGroupFamily(dbParameterGroupFamily)
    .description("Created by using the AWS SDK for Java")
    .build();

        CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
        System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Java 2.x API 參考ParameterGroup中的[創建數據庫](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client
```

```
@classmethod
def from_client(cls):
    """
    Instantiates this class from a Boto3 client.
    """
    rds_client = boto3.client("rds")
    return cls(rds_client)

def create_parameter_group(
    self, parameter_group_name, parameter_group_family, description
):
    """
    Creates a DB parameter group that is based on the specified parameter
group
    family.

    :param parameter_group_name: The name of the newly created parameter
group.
    :param parameter_group_family: The family that is used as the basis of
the new
                                parameter group.
    :param description: A description given to the parameter group.
    :return: Data about the newly created parameter group.
    """
    try:
        response = self.rds_client.create_db_parameter_group(
            DBParameterGroupName=parameter_group_name,
            DBParameterGroupFamily=parameter_group_family,
            Description=description,
        )
    except ClientError as err:
        logger.error(
            "Couldn't create parameter group %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response
```

- 有關 API 的詳細信息，請參閱ParameterGroup在 AWS SDK 中為 Python (博托 3) API 參考[創建數據庫](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateDBSnapshot配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateDBSnapshot。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Create a snapshot of a DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
/// <returns>DB snapshot object.</returns>
public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
{
    var response = await _amazonRDS.CreateDBSnapshotAsync(
        new CreateDBSnapshotRequest()
        {
            DBSnapshotIdentifier = snapshotIdentifier,
            DBInstanceIdentifier = dbInstanceIdentifier
```

```
    });  
  
    return response.DBSnapshot;  
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [CreateDBSnapshot](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;  
// Optional: Set to the AWS Region (overrides config file).  
// clientConfig.region = "us-east-1";  
  
Aws::RDS::RDSClient client(clientConfig);  
  
    Aws::RDS::Model::CreateDBSnapshotRequest request;  
    request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);  
    request.SetDBSnapshotIdentifier(snapshotID);  
  
    Aws::RDS::Model::CreateDBSnapshotOutcome outcome =  
        client.CreateDBSnapshot(request);  
  
    if (outcome.IsSuccess()) {  
        std::cout << "Snapshot creation has started."  
                  << std::endl;  
    }  
    else {  
        std::cerr << "Error with RDS::CreateDBSnapshot. "  
                  << outcome.GetError().GetMessage()  
                  << std::endl;  
        cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,  
client);
```

```
        return false;
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [CreateDBSnapshot](#)。

CLI

AWS CLI

若要建立資料庫快照

下列 `create-db-snapshot` 範例會建立資料庫快照集。

```
aws rds create-db-snapshot \  
  --db-instance-identifier database-mysql \  
  --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBInstanceIdentifier": "database-mysql",  
    "Engine": "mysql",  
    "AllocatedStorage": 100,  
    "Status": "creating",  
    "Port": 3306,  
    "AvailabilityZone": "us-east-1b",  
    "VpcId": "vpc-6594f31c",  
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",  
    "MasterUsername": "admin",  
    "EngineVersion": "5.6.40",  
    "LicenseModel": "general-public-license",  
    "SnapshotType": "manual",  
    "Iops": 1000,  
    "OptionGroupName": "default:mysql-5-6",  
    "PercentProgress": 0,  
    "StorageType": "io1",  
    "Encrypted": true,  
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/  
AKIAIOSFODNN7EXAMPLE",
```

```

    "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
    "IAMDatabaseAuthenticationEnabled": false,
    "ProcessorFeatures": [],
    "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
  }
}

```

如需詳細資訊，請參閱 Amazon RDS 使用者指南中的[建立資料庫快照](#)。

- 如需 API 詳細資訊，請參閱[命令參考中 AWS CLI 的 CreateDBSnapshot](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

type DbInstances struct {
  RdsClient *rds.Client
}

// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
string) (
 *types.DBSnapshot, error) {
  output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
&rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
})
  if err != nil {
    log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
    return nil, err
  } else {

```

```
    return output.DBSnapshot, nil
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [CreateDBSnapshot](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
// Create an Amazon RDS snapshot.
public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
    try {
        CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .dbSnapshotIdentifier(dbSnapshotIdentifier)
            .build();


        CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
        System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [CreateDBSnapshot](#)。

PHP

適用於 PHP 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;

$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$snapshotName = '<<{{backup_2018_12_25}}>>';

try {
    $result = $rdsClient->createDBSnapshot([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBSnapshotIdentifier' => $snapshotName,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for PHP API 參考》中的 [CreateDBSnapshot](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
                DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
```

```
except ClientError as err:
    logger.error(
        "Couldn't create snapshot of %s. Here's why: %s: %s",
        instance_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return snapshot
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [CreateDBSnapshot](#)。

Ruby

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
    db_snapshot_identifier: id
  })
rescue Aws::Errors::ServiceError => e
```

```
puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [CreateDBSnapshot](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DeleteDBInstance配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DeleteDBInstance。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Delete a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
{
    var response = await _amazonRDS.DeleteDBInstanceAsync(
        new DeleteDBInstanceRequest()
        {
            DBInstanceIdentifier = dbInstanceIdentifier,
```

```
        SkipFinalSnapshot = true,
        DeleteAutomatedBackups = true
    });

    return response.DBInstance;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [DeleteDBInstance](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::DeleteDBInstanceRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);
    request.SetSkipFinalSnapshot(true);
    request.SetDeleteAutomatedBackups(true);

    Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
        client.DeleteDBInstance(request);

    if (outcome.IsSuccess()) {
        std::cout << "DB instance deletion has started."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::DeleteDBInstance. "
                  << outcome.GetError().GetMessage()
```

```
        << std::endl;
        result = false;
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [DeleteDBInstance](#)。

CLI

AWS CLI

若要刪除資料庫執行個體

下列delete-db-instance範例會在建立名為的最終資料庫快照集後刪除指定的資料庫執行個體test-instance-final-snap。

```
aws rds delete-db-instance \
  --db-instance-identifier test-instance \
  --final-db-snapshot-identifier test-instance-final-snap
```

輸出：

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "test-instance",
    "DBInstanceStatus": "deleting",
    ...some output truncated...
  }
}
```

- 如需 API 詳細資訊，請參閱命令參考中AWS CLI 的 [DeleteDBInstance](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {
    RdsClient *rds.Client
}

// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
    _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
        &rds.DeleteDBInstanceInput{
            DBInstanceIdentifier: aws.String(instanceName),
            SkipFinalSnapshot:    true,
            DeleteAutomatedBackups: aws.Bool(true),
        })
    if err != nil {
        log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
        return err
    } else {
        return nil
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [DeleteDBInstance](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
```

```
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteDBInstance {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <dbInstanceIdentifier>\s

                Where:
                dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
            .region(region)
            .build();

        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
                .dbInstanceIdentifier(dbInstanceIdentifier)
                .deleteAutomatedBackups(true)

```



```
        .skipFinalSnapshot(true)
        .build();

        DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
        System.out.print("The status of the database is " +
response.dbInstance().dbInstanceStatus());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [DeleteDBInstance](#)。

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
suspend fun deleteDatabaseInstance(dbInstanceIdentifierVal: String?) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            deleteAutomatedBackups = true
            skipFinalSnapshot = true
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
${response.dbInstance?.dbInstanceStatus}")
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的 AWS SDK API 參考》中的 [DeleteDBInstance](#)。

PHP

適用於 PHP 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;

//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-1'
]);

$dbIdentifier = '<<{{db-identifier}}>>';

try {
    $result = $rdsClient->deleteDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for PHP API 參考》中的 [DeleteDBInstance](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
```

```
except ClientError as err:
    logger.error(
        "Couldn't delete DB instance %s. Here's why: %s: %s",
        instance_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return db_inst
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [DeleteDBInstance](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 DeleteDBParameterGroup 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DeleteDBParameterGroup。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
```

```

    /// Delete a DB parameter group. The group cannot be a default DB parameter
    group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

```

- 有關 API 的詳細信息，請參閱 AWS SDK for .NET API 參考ParameterGroup中的[刪除數據庫](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::DeleteDBParameterGroupRequest request;
    request.SetDBParameterGroupName(parameterGroupName);

    Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =

```

```
client.DeleteDBParameterGroup(request);

if (outcome.IsSuccess()) {
    std::cout << "The DB parameter group was successfully deleted."
              << std::endl;
}
else {
    std::cerr << "Error with RDS::DeleteDBParameterGroup. "
              << outcome.GetError().GetMessage()
              << std::endl;
    result = false;
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for C++ API 參考ParameterGroup中的[刪除數據庫](#)。

CLI

AWS CLI

刪除資料庫參數群組

下列command範例會刪除資料庫參數群組。

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的使用資料庫參數群組](#)。

- 如需 API 的詳細資訊，請參閱AWS CLI 命令參考ParameterGroup中的[刪除資料庫](#)。

Go

SDK for Go V2

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。


```
type DbInstances struct {
    RdsClient *rds.Client
}

// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
error {
    _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
        &rds.DeleteDBParameterGroupInput{
            DBParameterGroupName: aws.String(parameterGroupName),
        })
    if err != nil {
        log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
        return err
    } else {
        return nil
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Go API 參考ParameterGroup中的[刪除數據庫](#)。

Java

適用於 Java 2.x 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
// Delete the parameter group after database has been deleted.
// An exception is thrown if you attempt to delete the para group while
database
// exists.
public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
    throws InterruptedException {
    try {
        boolean isDataDel = false;
        boolean didFind;
        String instanceARN;

        // Make sure that the database has been deleted.
        while (!isDataDel) {
            DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            int listSize = instanceList.size();
            didFind = false;
            int index = 1;
            for (DBInstance instance : instanceList) {
                instanceARN = instance.dbInstanceArn();
                if (instanceARN.compareTo(dbARN) == 0) {
                    System.out.println(dbARN + " still exists");
                    didFind = true;
                }
            }
            if ((index == listSize) && (!didFind)) {
                // Went through the entire list and did not find the
database ARN.

                isDataDel = true;
            }
            Thread.sleep(sleepTime * 1000);
        }
    }
}
```



```
        index++;
    }
}

// Delete the para group.
DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
    .dbParameterGroupName(dbGroupName)
    .build();

rdsClient.deleteDBParameterGroup(parameterGroupRequest);
System.out.println(dbGroupName + " was deleted.");

} catch (RdsException e) {
    System.out.println(e.getLocalizedMessage());
    System.exit(1);
}
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Java 2.x API 參考ParameterGroup中的[刪除數據庫](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client
```

```
@classmethod
def from_client(cls):
    """
    Instantiates this class from a Boto3 client.
    """
    rds_client = boto3.client("rds")
    return cls(rds_client)

def delete_parameter_group(self, parameter_group_name):
    """
    Deletes a DB parameter group.

    :param parameter_group_name: The name of the parameter group to delete.
    :return: Data about the parameter group.
    """
    try:
        self.rds_client.delete_db_parameter_group(
            DBParameterGroupName=parameter_group_name
        )
    except ClientError as err:
        logger.error(
            "Couldn't delete parameter group %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

- 如需 API 的詳細資訊，請參閱ParameterGroup在 AWS SDK 中[刪除資料庫](#) (Boto3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DescribeAccountAttributes配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DescribeAccountAttributes。

CLI

AWS CLI

描述帳號屬性

下列describe-account-attributes範例會擷取目前 AWS 帳號的屬性。

```
aws rds describe-account-attributes
```

輸出：

```
{
  "AccountQuotas": [
    {
      "Max": 40,
      "Used": 4,
      "AccountQuotaName": "DBInstances"
    },
    {
      "Max": 40,
      "Used": 0,
      "AccountQuotaName": "ReservedDBInstances"
    },
    {
      "Max": 100000,
      "Used": 40,
      "AccountQuotaName": "AllocatedStorage"
    },
    {
      "Max": 25,
      "Used": 0,
      "AccountQuotaName": "DBSecurityGroups"
    },
    {
      "Max": 20,
      "Used": 0,
      "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
    },
    {
      "Max": 50,
      "Used": 1,
      "AccountQuotaName": "DBParameterGroups"
    }
  ]
}
```

```
    },
    {
      "Max": 100,
      "Used": 3,
      "AccountQuotaName": "ManualSnapshots"
    },
    {
      "Max": 20,
      "Used": 0,
      "AccountQuotaName": "EventSubscriptions"
    },
    {
      "Max": 50,
      "Used": 1,
      "AccountQuotaName": "DBSubnetGroups"
    },
    {
      "Max": 20,
      "Used": 1,
      "AccountQuotaName": "OptionGroups"
    },
    {
      "Max": 20,
      "Used": 6,
      "AccountQuotaName": "SubnetsPerDBSubnetGroup"
    },
    {
      "Max": 5,
      "Used": 0,
      "AccountQuotaName": "ReadReplicasPerMaster"
    },
    {
      "Max": 40,
      "Used": 1,
      "AccountQuotaName": "DBClusters"
    },
    {
      "Max": 50,
      "Used": 0,
      "AccountQuotaName": "DBClusterParameterGroups"
    },
    {
      "Max": 5,
      "Used": 0,
```

```
        "AccountQuotaName": "DBClusterRoles"
    }
]
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考中的[DescribeAccount](#)屬性。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.AccountQuota;
import software.amazon.awssdk.services.rds.model.RdsException;
import
    software.amazon.awssdk.services.rds.model.DescribeAccountAttributesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DescribeAccountAttributes {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
            .region(region)
            .build();

        getAccountAttributes(rdsClient);
    }
}
```

```
        rdsClient.close();
    }

    public static void getAccountAttributes(RdsClient rdsClient) {
        try {
            DescribeAccountAttributesResponse response =
rdsClient.describeAccountAttributes();
            List<AccountQuota> quotasList = response.accountQuotas();
            for (AccountQuota quotas : quotasList) {
                System.out.println("Name is: " + quotas.accountQuotaName());
                System.out.println("Max value is " + quotas.max());
            }
        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的[DescribeAccount](#)屬性。

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
suspend fun getAccountAttributes() {
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
rdsClient.describeAccountAttributes(DescribeAccountAttributesRequest {})
        response.accountQuotas?.forEach { quotas ->
            val response = response.accountQuotas
            println("Name is: ${quotas.accountQuotaName}")
            println("Max value is ${quotas.max}")
        }
    }
}
```

```
}  
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK 中的 [DescribeAccount屬性](#) 以獲取 Kotlin API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭 DescribeDBEngineVersions 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DescribeDBEngineVersions。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>  
/// Get a list of DB engine versions for a particular DB engine.  
/// </summary>  
/// <param name="engine">Name of the engine.</param>  
/// <param name="dbParameterGroupFamily">Optional parameter group family  
name.</param>  
/// <returns>List of DBEngineVersions.</returns>  
public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string  
engine,  
    string dbParameterGroupFamily = null)  
{
```

```

var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
    new DescribeDBEngineVersionsRequest()
    {
        Engine = engine,
        DBParameterGroupFamily = dbParameterGroupFamily
    });
return response.DBEngineVersions;
}

```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考資料 [EngineVersions](#) 中的說明 B。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,

```



```
        const Aws::String &parameterGroupFamily,

Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
        const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.

    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
            client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
engineVersions.begin(),
                                     engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                << outcome.GetError().GetMessage()
                << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考資料 [EngineVersions](#) 中的說明 B。

CLI

AWS CLI

若要描述 MySQL 資料庫引擎的資料庫引擎版本

下列describe-db-engine-versions範例會顯示有關指定資料庫引擎之每個資料庫引擎版本的詳細資料。

```
aws rds describe-db-engine-versions \  
  --engine mysql
```

輸出：

```
{  
  "DBEngineVersions": [  
    {  
      "Engine": "mysql",  
      "EngineVersion": "5.5.46",  
      "DBParameterGroupFamily": "mysql5.5",  
      "DBEngineDescription": "MySQL Community Edition",  
      "DBEngineVersionDescription": "MySQL 5.5.46",  
      "ValidUpgradeTarget": [  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.53",  
          "Description": "MySQL 5.5.53",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        },  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.54",  
          "Description": "MySQL 5.5.54",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        },  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.57",  
          "Description": "MySQL 5.5.57",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        }  
      ]  
    }  
  ]  
}
```

```

    },
    ...some output truncated...
  ]
}

```

如需詳細資訊，請參閱[什麼是 Amazon Relational Database Service 服務 \(Amazon RDS\)？](#) 在 Amazon RDS 用戶指南中。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[EngineVersions](#)中的說明 B。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

type DbInstances struct {
  RdsClient *rds.Client
}

// GetEngineVersions gets database engine versions that are available for the
// specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
parameterGroupFamily string) (
[]types.DBEngineVersion, error) {
output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
&rds.DescribeDBEngineVersionsInput{
  Engine:          aws.String(engine),
  DBParameterGroupFamily: aws.String(parameterGroupFamily),
})
if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
} else {

```

```
    return output.DBEngineVersions, nil
  }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Go API 參考資料 [EngineVersions](#) 中的說明 B。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
public static void describeDBEngines(RdsClient rdsClient) {
    try {
        DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
            .defaultOnly(true)
            .engine("mysql")
            .maxRecords(20)
            .build();

        DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
        List<DBEngineVersion> engines = response.dbEngineVersions();

        // Get all DBEngineVersion objects.
        for (DBEngineVersion engineOb : engines) {
            System.out.println("The name of the DB parameter group family for
the database engine is "
                + engineOb.dbParameterGroupFamily());
            System.out.println("The name of the database engine " +
engineOb.engine());
            System.out.println("The version number of the database engine " +
engineOb.engineVersion());
        }
    }
}
```

```
    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考資料 [EngineVersions](#) 中的說明 [B](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        """
```

```
and parameter group family.

:param engine: The database engine to look up.
:param parameter_group_family: When specified, restricts the returned
list of
                                engine versions to those that are
compatible with
                                this parameter group family.
:return: The list of database engine versions.
"""
try:
    kwargs = {"Engine": engine}
    if parameter_group_family is not None:
        kwargs["DBParameterGroupFamily"] = parameter_group_family
    response = self.rds_client.describe_db_engine_versions(**kwargs)
    versions = response["DBEngineVersions"]
except ClientError as err:
    logger.error(
        "Couldn't get engine versions for %s. Here's why: %s: %s",
        engine,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return versions
```

- 有關 API 的詳細信息，請參閱 AWS SDK EngineVersions 中的 [說明 B](#)，用於 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭 DescribeDBInstances 配 AWS 開發套件或 CLI 使用


下列程式碼範例會示範如何使用 DescribeDBInstances。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

 Note


還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Returns a list of DB instances.
/// </summary>
/// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
/// <returns>List of DB instances.</returns>
public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
{
    var results = new List<DBInstance>();
    var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
        new DescribeDBInstancesRequest
        {
            DBInstanceIdentifier = dbInstanceIdentifier
        });
    // Get the entire list using the paginator.
    await foreach (var instances in instancesPaginator.DBInstances)
    {
        results.Add(instances);
    }
    return results;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [DescribeDBInstances](#)。

C++

適用於 C++ 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
        client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "

```



```
        << outcome.GetError().GetMessage()
        << std::endl;
    }
    // This example does not log an error if the DB instance does not exist.
    // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [DescribeDBInstances](#)。

CLI

AWS CLI

說明資料庫執行個體

下列 describe-db-instances 範例擷取有關指定資料庫執行個體的詳細資訊。

```
aws rds describe-db-instances \
  --db-instance-identifier mydbinstancecf
```

輸出：

```
{
  "DBInstances": [
    {
      "DBInstanceIdentifier": "mydbinstancecf",
      "DBInstanceClass": "db.t3.small",
      "Engine": "mysql",
      "DBInstanceStatus": "available",
      "MasterUsername": "masterawsuser",
      "Endpoint": {
        "Address": "mydbinstancecf.abcxample.us-
east-1.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "Z2R2ITUGPM61AM"
      },
    },
  ],
}
```

```

        ...some output truncated...
    }
]
}

```

- 如需 API 詳細資訊，請參閱命令 [參考中AWS CLI 的 DescribeDB Instances](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

type DbInstances struct {
    RdsClient *rds.Client
}

// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
    *types.DBInstance, error) {
    output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
        &rds.DescribeDBInstancesInput{
            DBInstanceIdentifier: aws.String(instanceName),
        })
    if err != nil {
        var notFoundError *types.DBInstanceNotFoundFault
        if errors.As(err, &notFoundError) {
            log.Printf("DB instance %v does not exist.\n", instanceName)
            err = nil
        } else {
            log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
        }
        return nil, err
    } else {
        return &output.DBInstances[0], nil
    }
}

```

```
}  
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [DescribeDBInstances](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.rds.RdsClient;  
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;  
import software.amazon.awssdk.services.rds.model.DBInstance;  
import software.amazon.awssdk.services.rds.model.RdsException;  
import java.util.List;  
  
/**  
 * Before running this Java V2 code example, set up your development  
 * environment, including your credentials.  
 *  
 * For more information, see the following documentation topic:  
 *  
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html  
 */  
public class DescribeDBInstances {  
  
    public static void main(String[] args) {  
        Region region = Region.US_EAST_1;  
        RdsClient rdsClient = RdsClient.builder()  
            .region(region)  
            .build();  
  
        describeInstances(rdsClient);  
    }  
}
```

```
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [DescribeDBInstances](#)。

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
suspend fun describeInstances() {
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbInstances(DescribeDbInstancesRequest
        {})
    }
}
```

```
response.dbInstances?.forEach { instance ->
    println("Instance Identifier is ${instance.dbInstanceIdentifier}")
    println("The Engine is ${instance.engine}")
    println("Connection endpoint is ${instance.endpoint?.address}")
}
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的 AWS SDK API 參考》中的 [DescribeDBInstances](#)。

PHP

適用於 PHP 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;

//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

try {
    $result = $rdsClient->describeDBInstances();
    foreach ($result['DBInstances'] as $instance) {
        print('<p>DB Identifier: ' . $instance['DBInstanceIdentifier']);
        print('<br />Endpoint: ' . $instance['Endpoint']['Address']
            . ':' . $instance['Endpoint']['Port']);
        print('<br />Current Status: ' . $instance["DBInstanceStatus"]);
        print('</p>');
    }
}
```

```
}
print(" Raw Result ");
var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for PHP API 參考》中的 [DescribeDBInstances](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def get_db_instance(self, instance_id):
        """
```

```
Gets data about a DB instance.

:param instance_id: The ID of the DB instance to retrieve.
:return: The retrieved DB instance.
"""
try:
    response = self.rds_client.describe_db_instances(
        DBInstanceIdentifier=instance_id
    )
    db_inst = response["DBInstances"][0]
except ClientError as err:
    if err.response["Error"]["Code"] == "DBInstanceNotFound":
        logger.info("Instance %s does not exist.", instance_id)
    else:
        logger.error(
            "Couldn't get DB instance %s. Here's why: %s: %s",
            instance_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
else:
    return db_inst
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [DescribeDBInstances](#)。

Ruby

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'
```

```
# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
      "name": i.id,
      "status": i.db_instance_status
    })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBInstances](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 DescribeDBParameterGroups 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DescribeDBParameterGroups。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。


```
/// <summary>
/// Get descriptions of DB parameter groups.
/// </summary>
/// <param name="name">Optional name of the DB parameter group to describe.</
param>
/// <returns>The list of DB parameter group descriptions.</returns>
public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
name = null)
{
    var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
        new DescribeDBParameterGroupsRequest()
        {
            DBParameterGroupName = name
        });
    return response.DBParameterGroups;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考資料 [ParameterGroups](#) 中的說明 [B](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
```

```
Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
    client.DescribeDBParameterGroups(request);

if (outcome.IsSuccess()) {
    std::cout << "DB parameter group named '" <<
        PARAMETER_GROUP_NAME << "' already exists." << std::endl;
    dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
}

else {
    std::cerr << "Error with RDS::DescribeDBParameterGroups. "
        << outcome.GetError().GetMessage()
        << std::endl;
    return false;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考資料 [ParameterGroups](#) 中的說明 B。

CLI

AWS CLI

描述您的資料庫參數群組

下列 describe-db-parameter-groups 範例會擷取有關資料庫參數群組的詳細資訊。

```
aws rds describe-db-parameter-groups
```

輸出：

```
{
  "DBParameterGroups": [
    {
      "DBParameterGroupName": "default.aurora-mysql5.7",
      "DBParameterGroupFamily": "aurora-mysql5.7",
      "Description": "Default parameter group for aurora-mysql5.7",
      "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
    },
  ],
}
```

```

    {
      "DBParameterGroupName": "default.aurora-postgresql9.6",
      "DBParameterGroupFamily": "aurora-postgresql9.6",
      "Description": "Default parameter group for aurora-postgresql9.6",
      "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
    },
    {
      "DBParameterGroupName": "default.aurora5.6",
      "DBParameterGroupFamily": "aurora5.6",
      "Description": "Default parameter group for aurora5.6",
      "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
    },
    {
      "DBParameterGroupName": "default.mariadb10.1",
      "DBParameterGroupFamily": "mariadb10.1",
      "Description": "Default parameter group for mariadb10.1",
      "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
    },
    ...some output truncated...
  ]
}

```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的使用資料庫參數群組](#)。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參閱 [ParameterGroups](#) 中的說明 B。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

type DbInstances struct {
  RdsClient *rds.Client

```

```
}

// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
    *types.DBParameterGroup, error) {
    output, err := instances.RdsClient.DescribeDBParameterGroups(
        context.TODO(), &rds.DescribeDBParameterGroupsInput{
            DBParameterGroupName: aws.String(parameterGroupName),
        })
    if err != nil {
        var notFoundError *types.DBParameterGroupNotFoundFault
        if errors.As(err, &notFoundError) {
            log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
            err = nil
        } else {
            log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
        }
        return nil, err
    } else {
        return &output.DBParameterGroups[0], err
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Go API 參考 [資料ParameterGroups](#) 中的說明 B。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
    try {
```

```
DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
    .dbParameterGroupName(dbGroupName)
    .maxRecords(20)
    .build();

DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
List<DBParameterGroup> groups = response.dbParameterGroups();
for (DBParameterGroup group : groups) {
    System.out.println("The group name is " +
group.dbParameterGroupName());
    System.out.println("The group description is " +
group.description());
}

} catch (RdsException e) {
    System.out.println(e.getLocalizedMessage());
    System.exit(1);
}
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考資料 [ParameterGroups](#) 中的說明 [B](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 [GitHub](#)。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
```

```
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
                    parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group
```

- 有關 API 的詳細信息，請參閱 AWS SDK ParameterGroups 中的 [說明 B](#)，用於 Python (博托 3) API 參考。

Ruby

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
      "name": p.db_parameter_group_name,
      "description": p.description
    })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [資料ParameterGroups](#) 中的 [說明 B](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 DescribeDBParameters 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DescribeDBParameters。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 [GitHub](#)。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Get a list of DB parameters from a specific parameter group.
/// </summary>
/// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
/// <param name="source">Optional source for selecting parameters.</param>
/// <returns>List of parameter values.</returns>
public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
{
    var results = new List<Parameter>();
    var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
        new DescribeDBParametersRequest()
        {
            DBParameterGroupName = dbParameterGroupName,
            Source = source
        });
    // Get the entire list using the paginator.
    await foreach (var parameters in paginateParameters.Parameters)
    {
        results.Add(parameters);
    }
}
```



```

    }
    return results;
}

```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [DescribeDBParameters](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

    //! Routine which gets DB parameters using the 'DescribeDBParameters' api.
    /*!
    \sa getDBParameters()
    \param parameterGroupName: The name of the parameter group.
    \param namePrefix: Prefix string to filter results by parameter name.
    \param source: A source such as 'user', ignored if empty.
    \param parametersResult: Vector of 'Parameter' objects returned by the routine.
    \param client: 'RDSClient' instance.
    \return bool: Successful completion.
    */
    bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                     const Aws::String &namePrefix,
                                     const Aws::String &source,
                                     Aws::Vector<Aws::RDS::Model::Parameter>
                                     &parametersResult,
                                     const Aws::RDS::RDSClient &client) {

        Aws::String marker;

```

```
do {
    Aws::RDS::Model::DescribeDBParametersRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    if (!marker.empty()) {
        request.SetMarker(marker);
    }
    if (!source.empty()) {
        request.SetSource(source);
    }

    Aws::RDS::Model::DescribeDBParametersOutcome outcome =
        client.DescribeDBParameters(request);

    if (outcome.IsSuccess()) {
        const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
            outcome.GetResult().GetParameters();
        for (const Aws::RDS::Model::Parameter &parameter: parameters) {
            if (!namePrefix.empty()) {
                if (parameter.GetParameterName().find(namePrefix) == 0) {
                    parametersResult.push_back(parameter);
                }
            }
            else {
                parametersResult.push_back(parameter);
            }
        }

        marker = outcome.GetResult().GetMarker();
    }
    else {
        std::cerr << "Error with RDS::DescribeDBParameters. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
} while (!marker.empty());

return true;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [DescribeDBParameters](#)。

CLI

AWS CLI

若要描述資料庫參數群組中的參數

下列describe-db-parameters範例會擷取指定之 DB 參數群組的詳細資訊。

```
aws rds describe-db-parameters \  
  --db-parameter-group-name mydbpg
```

輸出：


```
{  
  "Parameters": [  
    {  
      "ParameterName": "allow-suspicious-udfs",  
      "Description": "Controls whether user-defined functions that have  
only an xxx symbol for the main function can be loaded",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": false,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterName": "auto_generate_certs",  
      "Description": "Controls whether the server autogenerates SSL key and  
certificate files in the data directory, if they do not already exist.",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": false,  
      "ApplyMethod": "pending-reboot"  
    },  
    ...some output truncated...  
  ]  
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的使用資料庫參數群組](#)。

- 如需 API 詳細資訊，請參閱[命令參考中AWS CLI 的說明 B 參數](#)。

Go

SDK for Go V2

 Note

還有更多關於 [GitHub](#)。尋找完整範例，並了解如何在[AWS 設定和執程式碼範例儲存庫](#)。

```
type DbInstances struct {
    RdsClient *rds.Client
}

// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
string) (
[]types.Parameter, error) {

var output *rds.DescribeDBParametersOutput
var params []types.Parameter
var err error
parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
&rds.DescribeDBParametersInput{
    DBParameterGroupName: aws.String(parameterGroupName),
    Source:                 aws.String(source),
})
for parameterPaginator.HasMorePages() {
    output, err = parameterPaginator.NextPage(context.TODO())
    if err != nil {
        log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
        break
    } else {
        params = append(params, output.Parameters...)
    }
}
return params, err
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [DescribeDBParameters](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
// Retrieve parameters in the group.
public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
    try {
        DescribeDbParametersRequest dbParameterGroupsRequest;
        if (flag == 0) {
            dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                .dbParameterGroupName(dbGroupName)
                .build();
        } else {
            dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                .dbParameterGroupName(dbGroupName)
                .source("user")
                .build();
        }

        DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
        List<Parameter> dbParameters = response.parameters();
        String paraName;
        for (Parameter para : dbParameters) {
            // Only print out information about either auto_increment_offset
or
            // auto_increment_increment.
            paraName = para.parameterName();
            if ((paraName.compareTo("auto_increment_offset") == 0)
```

```
        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
            System.out.println("*** The parameter name is " + paraName);
            System.out.println("*** The parameter value is " +
para.parameterValue());
            System.out.println("*** The parameter data type is " +
para.dataType());
            System.out.println("*** The parameter description is " +
para.description());
            System.out.println("*** The parameter allowed values is " +
para.allowedValues());
        }
    }

} catch (RdsException e) {
    System.out.println(e.getLocalizedMessage());
    System.exit(1);
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [DescribeDBParameters](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
self.rds_client = rds_client

@classmethod
def from_client(cls):
    """
    Instantiates this class from a Boto3 client.
    """
    rds_client = boto3.client("rds")
    return cls(rds_client)

def get_parameters(self, parameter_group_name, name_prefix="", source=None):
    """
    Gets the parameters that are contained in a DB parameter group.

    :param parameter_group_name: The name of the parameter group to query.
    :param name_prefix: When specified, the retrieved list of parameters is
    filtered
        to contain only parameters that start with this
    prefix.
    :param source: When specified, only parameters from this source are
    retrieved.
        For example, a source of 'user' retrieves only parameters
    that
        were set by a user.
    :return: The list of requested parameters.
    """
    try:
        kwargs = {"DBParameterGroupName": parameter_group_name}
        if source is not None:
            kwargs["Source"] = source
        parameters = []
        paginator = self.rds_client.get_paginator("describe_db_parameters")
        for page in paginator.paginate(**kwargs):
            parameters += [
                p
                for p in page["Parameters"]
                if p["ParameterName"].startswith(name_prefix)
            ]
    except ClientError as err:
        logger.error(
            "Couldn't get parameters for %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
```

```
        err.response["Error"]["Message"],
    )
    raise
else:
    return parameters
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [DescribeDBParameters](#)。

Ruby

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
      "name": p.db_parameter_group_name,
      "description": p.description
    })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBParameters](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 DescribeDBSnapshots 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DescribeDBSnapshots。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Return a list of DB snapshots for a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>List of DB snapshots.</returns>
public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
{
    var results = new List<DBSnapshot>();
    var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
        new DescribeDBSnapshotsRequest()
        {
            DBInstanceIdentifier = dbInstanceIdentifier
        });
}
```

```
// Get the entire list using the paginator.
await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
{
    results.Add(snapshots);
}
return results;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的 [DescribeDBSnapshots](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::DescribeDBSnapshotsRequest request;
    request.SetDBSnapshotIdentifier(snapshotID);

    Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
        client.DescribeDBSnapshots(request);

    if (outcome.IsSuccess()) {
        snapshot = outcome.GetResult().GetDBSnapshots()[0];
    }
    else {
        std::cerr << "Error with RDS::DescribeDBSnapshots. "
            << outcome.GetError().GetMessage()
            << std::endl;
```

```
        cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
        return false;
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的 [DescribeDBSnapshots](#)。

CLI

AWS CLI

範例 1：說明資料庫執行個體的資料庫快照

下列 describe-db-snapshots 範例會擷取資料庫執行個體的資料庫快照詳細資訊。

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{
  "DBSnapshots": [
    {
      "DBSnapshotIdentifier": "mydbsnapshot",
      "DBInstanceIdentifier": "mysqladb",
      "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
      "Engine": "mysql",
      "AllocatedStorage": 20,
      "Status": "available",
      "Port": 3306,
      "AvailabilityZone": "us-east-1f",
      "VpcId": "vpc-6594f31c",
      "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
      "MasterUsername": "mysqladmin",
      "EngineVersion": "5.6.37",
      "LicenseModel": "general-public-license",
      "SnapshotType": "manual",
      "OptionGroupName": "default:mysql-5-6",
      "PercentProgress": 100,
      "StorageType": "gp2",
      "Encrypted": false,
    }
  ]
}
```

```
        "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
]
}
```

如需詳細資訊，請參閱 Amazon RDS 使用者指南中的[建立資料庫快照](#)。

範例 2：尋找已拍攝的手動快照數量

下列 describe-db-snapshots 範例會使用 --query 選項中的 length 運算子來傳回特定「AWS 區域」中已拍攝的手動快照數目。

```
aws rds describe-db-snapshots \
  --snapshot-type manual \
  --query "length(*[].{DBSnapshots:SnapshotType})" \
  --region eu-central-1
```

輸出：

```
35
```

如需詳細資訊，請參閱 Amazon RDS 使用者指南中的[建立資料庫快照](#)。

- 如需 API 詳細資訊，請參閱命令[參考中AWS CLI 的 DescribeDBSnapshots](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {
    RdsClient *rds.Client
}

// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
(*types.DBSnapshot, error) {
    output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
        &rds.DescribeDBSnapshotsInput{
            DBSnapshotIdentifier: aws.String(snapshotName),
        })
    if err != nil {
        log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
        return nil, err
    } else {
        return &output.DBSnapshots[0], nil
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的 [DescribeDBSnapshots](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
self.rds_client = rds_client

@classmethod
def from_client(cls):
    """
    Instantiates this class from a Boto3 client.
    """
    rds_client = boto3.client("rds")
    return cls(rds_client)

def get_snapshot(self, snapshot_id):
    """
    Gets a DB instance snapshot.

    :param snapshot_id: The ID of the snapshot to retrieve.
    :return: The retrieved snapshot.
    """
    try:
        response = self.rds_client.describe_db_snapshots(
            DBSnapshotIdentifier=snapshot_id
        )
        snapshot = response["DBSnapshots"][0]
    except ClientError as err:
        logger.error(
            "Couldn't get snapshot %s. Here's why: %s: %s",
            snapshot_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return snapshot
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的 [DescribeDBSnapshots](#)。

Ruby

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if
# error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
      "id": s.snapshot_id,
      "status": s.status
    })
  end
  instance_snapshots
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBSnapshots](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭 DescribeOrderableDBInstanceOptions 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 DescribeOrderableDBInstanceOptions。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Get a list of orderable DB instance options for a specific
/// engine and engine version.
/// </summary>
/// <param name="engine">Name of the engine.</param>
/// <param name="engineVersion">Version of the engine.</param>
/// <returns>List of OrderableDBInstanceOptions.</returns>
public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
{
    // Use a paginator to get a list of DB instance options.
    var results = new List<OrderableDBInstanceOption>();
    var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
    new DescribeOrderableDBInstanceOptionsRequest()
    {
        Engine = engine,
        EngineVersion = engineVersion,
    });
    // Get the entire list using the paginator.
```



```
    await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
    {
        results.Add(instanceOptions);
    }
    return results;
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for .NET API 參考 InstanceOptions 中的 [DescribeOrderable 數據庫](#)。

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
```

```
const Aws::String &engineVersion,
        Aws::String &dbInstanceClass,
        const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
            client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
&options =
                outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
instanceClass) ==
instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                << outcome.GetError().GetMessage()
                << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
engine are:"
        << std::endl;
```

```
for (int i = 0; i < instanceClasses.size(); ++i) {
    std::cout << "    " << i + 1 << ": " << instanceClasses[i] << std::endl;
}

int choice = askQuestionForIntRange(
    "Which micro DB instance class do you want to use? ",
    1, static_cast<int>(instanceClasses.size()));
dbInstanceClass = instanceClasses[choice - 1];
return true;
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for C++ API 參考 InstanceOptions 中的 [DescribeOrderable 數據庫](#)。

CLI

AWS CLI

說明可訂購的資料庫執行個體選項

下列 describe-orderable-db-instance-options 範例會擷取有關執行 MySQL 資料庫引擎之資料庫執行個體之可訂購選項的詳細資訊。

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

輸出：


```
{
  "OrderableDBInstanceOptions": [
    {
      "MinStorageSize": 5,
      "ReadReplicaCapable": true,
      "MaxStorageSize": 6144,
      "AvailabilityZones": [
        {
          "Name": "us-east-1a"
        },
        {
          "Name": "us-east-1b"
        }
      ]
    }
  ]
}
```

```
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "SupportsIops": false,
    "AvailableProcessorFeatures": [],
    "MultiAZCapable": true,
    "DBInstanceClass": "db.m1.large",
    "Vpc": true,
    "StorageType": "gp2",
    "LicenseModel": "general-public-license",
    "EngineVersion": "5.5.46",
    "SupportsStorageEncryption": false,
    "SupportsEnhancedMonitoring": true,
    "Engine": "mysql",
    "SupportsIAMDatabaseAuthentication": false,
    "SupportsPerformanceInsights": false
    }
]
...some output truncated...
}
```

- 有關 API 的詳細信息，請參閱AWS CLI 命令參考InstanceOptions中的 [DescribeOrderableDB](#)。

Go

SDK for Go V2

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {
    RdsClient *rds.Client
```

```
}

// GetOrderableInstances uses a paginator to get DB instance options that can be
// used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
string) (
[]types.OrderableDBInstanceOption, error) {

var output *rds.DescribeOrderableDBInstanceOptionsOutput
var instanceOptions []types.OrderableDBInstanceOption
var err error
orderablePaginator :=
rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
&rds.DescribeOrderableDBInstanceOptionsInput{
    Engine:      aws.String(engine),
    EngineVersion: aws.String(engineVersion),
})
for orderablePaginator.HasMorePages() {
    output, err = orderablePaginator.NextPage(context.TODO())
    if err != nil {
        log.Printf("Couldn't get orderable DB instance options: %v\n", err)
        break
    } else {
        instanceOptions = append(instanceOptions,
output.OrderableDBInstanceOptions...)
    }
}
return instanceOptions, err
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Go API 參考 InstanceOptions 中的 [DescribeOrderable 數據庫](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
// Get a list of allowed engine versions.
public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
    try {
        DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
            .dbParameterGroupFamily(dbParameterGroupFamily)
            .engine("mysql")
            .build();

        DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
        List<DBEngineVersion> dbEngines = response.dbEngineVersions();
        for (DBEngineVersion dbEngine : dbEngines) {
            System.out.println("The engine version is " +
dbEngine.engineVersion());
            System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
        }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Java 2.x API 參考 InstanceOptions 中的 [DescribeOrderable 數據庫](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)

    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
        instance.
        :param db_engine_version: The engine version that must be supported by
        the DB instance.
        :return: The list of DB instance options that can be used to create a
        compatible DB instance.
        """
        try:
            inst_opts = []
```

```
paginator = self.rds_client.get_paginator(
    "describe_orderable_db_instance_options"
)
for page in paginator.paginate(
    Engine=db_engine, EngineVersion=db_engine_version
):
    inst_opts += page["OrderableDBInstanceOptions"]
except ClientError as err:
    logger.error(
        "Couldn't get orderable DB instances. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return inst_opts
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件InstanceOptions中的資[DescribeOrderable料庫](#) (Boto3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭GenerateRDSAuthToken配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用GenerateRDSAuthToken。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

使用該[RdsUtilities](#)類生成身份驗證令牌。


```
public class GenerateRDSAuthToken {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <dbInstanceIdentifier> <masterUsername>

            Where:
                dbInstanceIdentifier - The database instance identifier.\s
                masterUsername - The master user name.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUsername = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
            .region(region)
            .build();

        String token = getAuthToken(rdsClient, dbInstanceIdentifier,
masterUsername);
        System.out.println("The token response is " + token);
    }

    public static String getAuthToken(RdsClient rdsClient, String
dbInstanceIdentifier, String masterUsername) {

        RdsUtilities utilities = rdsClient.utilities();
        try {
            GenerateAuthenticationTokenRequest tokenRequest =
GenerateAuthenticationTokenRequest.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .username(masterUsername)
                .port(3306)
                .hostname(dbInstanceIdentifier)
                .build();

            return utilities.generateAuthenticationToken(tokenRequest);
        }
    }
}
```

```
    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
    return "";
}
}
```

- 有關 API 詳細信息，請參閱 API 參考AuthToken中AWS SDK for Java 2.x 的[生成器](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配ModifyDBInstance配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用ModifyDBInstance。

CLI

AWS CLI

範例 1：若要修改資料庫執行個體

下列modify-db-instance範例會將選項群組和參數群組與相容的 Microsoft SQL Server 資料庫執行個體產生關聯。--apply-immediately參數會使選項和參數群組立即關聯，而不是等到下一個維護時段。

```
aws rds modify-db-instance \
  --db-instance-identifier database-2 \
  --option-group-name test-se-2017 \
  --db-parameter-group-name test-sqlserver-se-2017 \
  --apply-immediately
```

輸出：

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "database-2",
    "DBInstanceClass": "db.r4.large",
    "Engine": "sqlserver-se",
```

```
"DBInstanceStatus": "available",

...output omitted...

"DBParameterGroups": [
  {
    "DBParameterGroupName": "test-sqlserver-se-2017",
    "ParameterApplyStatus": "applying"
  }
],
"AvailabilityZone": "us-west-2d",

...output omitted...

"MultiAZ": true,
"EngineVersion": "14.00.3281.6.v1",
"AutoMinorVersionUpgrade": false,
"ReadReplicaDBInstanceIdentifiers": [],
"LicenseModel": "license-included",
"OptionGroupMemberships": [
  {
    "OptionGroupName": "test-se-2017",
    "Status": "pending-apply"
  }
],
"CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
"SecondaryAvailabilityZone": "us-west-2c",
"PubliclyAccessible": true,
"StorageType": "gp2",

...output omitted...

"DeletionProtection": false,
"AssociatedRoles": [],
"MaxAllocatedStorage": 1000
}
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的修改 Amazon RDS 資料庫執行個體](#)。

範例 2：將 VPC 安全群組與資料庫執行個體建立關聯

下列 `modify-db-instance` 範例會關聯特定 VPC 安全群組，並從資料庫執行個體移除資料庫安全群組：

```
aws rds modify-db-instance \  
  --db-instance-identifier dbName \  
  --vpc-security-group-ids sg-ID
```

輸出：


```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "dbName",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "available",  
    "MasterUsername": "admin",  
    "Endpoint": {  
      "Address": "dbName.abcdefghijkl.us-west-2.rds.amazonaws.com",  
      "Port": 3306,  
      "HostedZoneId": "ABCDEFGHIJK1234"  
    },  
    "AllocatedStorage": 20,  
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",  
    "PreferredBackupWindow": "11:57-12:27",  
    "BackupRetentionPeriod": 7,  
    "DBSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-ID",  
        "Status": "active"  
      }  
    ],  
    "... output omitted ..."  
    "MultiAZ": false,  
    "EngineVersion": "8.0.35",  
    "AutoMinorVersionUpgrade": true,  
    "ReadReplicaDBInstanceIdentifiers": [],  
    "LicenseModel": "general-public-license",  
    "... output omitted ..."  
  }  
}
```

如需詳細資訊，請參閱 Amazon RDS 使用者指南中的使用[安全群組控制存取](#)。

- 如需 API 詳細資訊，請參閱[命令參考中AWS CLI 的 ModifyDBInstance](#)。

Java

適用於 Java 2.x 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ModifyDBInstance {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <dbInstanceIdentifier> <dbSnapshotIdentifier>\s
                Where:
                dbInstanceIdentifier - The database instance identifier.\s
                masterUserPassword - The updated password that corresponds to
the master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
    }
}
```

```
String dbInstanceIdentifier = args[0];
String masterUserPassword = args[1];
Region region = Region.US_WEST_2;
RdsClient rdsClient = RdsClient.builder()
    .region(region)
    .build();

updateIntance(rdsClient, dbInstanceIdentifier, masterUserPassword);
rdsClient.close();
}

public static void updateIntance(RdsClient rdsClient, String
dbInstanceIdentifier, String masterUserPassword) {
    try {
        // For a demo - modify the DB instance by modifying the master
password.
        ModifyDbInstanceRequest modifyDbInstanceRequest =
ModifyDbInstanceRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .publiclyAccessible(true)
            .masterUserPassword(masterUserPassword)
            .build();

        ModifyDbInstanceResponse instanceResponse =
rdsClient.modifyDBInstance(modifyDbInstanceRequest);
        System.out.println("The ARN of the modified database is: " +
instanceResponse.dbInstance().dbInstanceArn());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [ModifyDBInstance](#)。

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
suspend fun updateInstance(
    dbInstanceIdentifierVal: String?,
    masterUserPasswordVal: String?,
) {
    val request =
        ModifyDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            publiclyAccessible = true
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val instanceResponse = rdsClient.modifyDbInstance(request)
        println("The ARN of the modified database is
        ${instanceResponse.dbInstance?.dbInstanceArn}")
    }
}
```

- 如需 API 的詳細資訊，請參閱《適用於 Kotlin 的 AWS SDK API 參考》中的 [ModifyDBInstance](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 ModifyDBParameterGroup 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 ModifyDBParameterGroup。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [開始使用資料庫執行個體](#)

.NET

AWS SDK for .NET

Note


還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Update a DB parameter group. Use the action
DescribeDBParameterGroupsAsync
/// to determine when the DB parameter group is ready to use.
/// </summary>
/// <param name="name">Name of the DB parameter group.</param>
/// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
/// <returns>The updated DB parameter group name.</returns>
public async Task<string> ModifyDBParameterGroup(
    string name, List<Parameter> parameters)
{
    var response = await _amazonRDS.ModifyDBParameterGroupAsync(
        new ModifyDBParameterGroupRequest()
        {
            DBParameterGroupName = name,
            Parameters = parameters,
        });
    return response.DBParameterGroupName;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考ParameterGroup中的[修改資料庫](#)。

C++

適用於 C++ 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

Aws::RDS::Model::ModifyDBParameterGroupRequest request;
request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
request.SetParameters(updateParameters);

Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
    client.ModifyDBParameterGroup(request);

if (outcome.IsSuccess()) {
    std::cout << "The DB parameter group was successfully modified."
              << std::endl;
}
else {
    std::cerr << "Error with RDS::ModifyDBParameterGroup. "
              << outcome.GetError().GetMessage()
              << std::endl;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考ParameterGroup中的[修改資料庫](#)。

CLI

AWS CLI

修改資料庫參數群組

下列modify-db-parameter-group範例會變更資料庫clr enabled參數群組中的參數值。此--apply-immediately參數會導致資料庫參數群組立即修改，而不是等到下一個維護時段。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --parameters "ParameterName='clr  
enabled',ParameterValue=1,ApplyMethod=immediate"
```

輸出：

```
{  
  "DBParameterGroupName": "test-sqlserver-se-2017"  
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的修改資料庫參數群組](#) 中的參數。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考ParameterGroup中的 [修改資料庫](#)。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
type DbInstances struct {  
  RdsClient *rds.Client  
}  
  
// UpdateParameters updates parameters in a named DB parameter group.  
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params  
 []types.Parameter) error {  
  _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),  
    &rds.ModifyDBParameterGroupInput{
```

```
    DBParameterGroupName: aws.String(parameterGroupName),
    Parameters:           params,
  })
if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
} else {
  return nil
}
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Go API 參考ParameterGroup中的[修改資料庫](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
// Modify auto_increment_offset and auto_increment_increment parameters.
public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
  try {
    Parameter parameter1 = Parameter.builder()
      .parameterName("auto_increment_offset")
      .applyMethod("immediate")
      .parameterValue("5")
      .build();

    List<Parameter> paraList = new ArrayList<>();
    paraList.add(parameter1);
    ModifyDbParameterGroupRequest groupRequest =
    ModifyDbParameterGroupRequest.builder()
      .dbParameterGroupName(dbGroupName)
      .parameters(paraList)
      .build();
```

```
        ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
        System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考ParameterGroup中的[修改資料庫](#)。

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)
```

```
def update_parameters(self, parameter_group_name, update_parameters):
    """
    Updates parameters in a custom DB parameter group.

    :param parameter_group_name: The name of the parameter group to update.
    :param update_parameters: The parameters to update in the group.
    :return: Data about the modified parameter group.
    """
    try:
        response = self.rds_client.modify_db_parameter_group(
            DBParameterGroupName=parameter_group_name,
            Parameters=update_parameters
        )
    except ClientError as err:
        logger.error(
            "Couldn't update parameters in %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response
```

- 有關 API 的詳細信息，請參閱ParameterGroup在 AWS SDK 中[修改數據庫](#)以獲取 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭RebootDBInstance配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用RebootDBInstance。

CLI

AWS CLI

重新啟動資料庫執行個體

下列 `reboot-db-instance` 範例會啟動指定資料庫執行個體的重新啟動。

```
aws rds reboot-db-instance \  
  --db-instance-identifier test-mysql-instance
```

輸出：

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "test-mysql-instance",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "rebooting",  
    "MasterUsername": "admin",  
    "Endpoint": {  
      "Address": "test-mysql-instance.#####.us-  
west-2.rds.amazonaws.com",  
      "Port": 3306,  
      "HostedZoneId": "Z1PVIF0EXAMPLE"  
    },  
    ... output omitted...  
  }  
}
```

如需詳細資訊，請參閱 [Amazon RDS 使用者指南中的重新啟動資料庫執行個體](#)。

- 有關 API 的詳細信息，請參閱 [命令參考中 AWS CLI 的 RebootDBInstance](#)。

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class RebootDBInstance {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <dbInstanceIdentifier>\s

                Where:
                dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
```

```
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
            .region(region)
            .build();

        rebootInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void rebootInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            RebootDbInstanceRequest rebootDbInstanceRequest =
RebootDbInstanceRequest.builder()
                .dbInstanceIdentifier(dbInstanceIdentifier)
                .build();

            RebootDbInstanceResponse instanceResponse =
rdsClient.rebootDBInstance(rebootDbInstanceRequest);
            System.out.print("The database " +
instanceResponse.dbInstance().dbInstanceArn() + " was rebooted");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [RebootDBInstance](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS 開發套件的 Amazon RDS 案例

下列程式碼範例說明如何在具有 AWS 開發套件的 Amazon RDS 中實作常見案例。這些案例會向您展示如何呼叫 Amazon RDS 中的多個函數來完成特定任務。每個案例都包含一個連結 GitHub，您可以在其中找到如何設定和執程式碼的指示。

範例

- [使用開 AWS 發套件開始使用 Amazon RDS 資料庫執行個體](#)

使用開 AWS 發套件開始使用 Amazon RDS 資料庫執行個體

下列程式碼範例示範如何：

- 建立自訂資料庫參數群組並設定參數值。
- 建立資料庫執行個體，設定為使用參數群組。資料庫執行個體也包含資料庫。
- 擷取執行個體的快照。
- 刪除執行個體和參數群組。

.NET

AWS SDK for .NET

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

在命令提示中執行互動式案例。

```
/// <summary>
/// Scenario for RDS DB instance example.
/// </summary>
public class RDSInstanceScenario
{
    /*
    Before running this .NET code example, set up your development environment,
    including your credentials.

    This .NET example performs the following tasks:
    1. Returns a list of the available DB engine families using the
    DescribeDBEngineVersionsAsync method.
    2. Selects an engine family and creates a custom DB parameter group using
    the CreateDBParameterGroupAsync method.
    3. Gets the parameter groups using the DescribeDBParameterGroupsAsync
    method.
```

```

4. Gets parameters in the group using the DescribeDBParameters method.
5. Parses and displays parameters in the group.
6. Modifies both the auto_increment_offset and auto_increment_increment
parameters
   using the ModifyDBParameterGroupAsync method.
7. Gets and displays the updated parameters using the DescribeDBParameters
method with a source of "user".
8. Gets a list of allowed engine versions using the
DescribeDBEngineVersionsAsync method.
9. Displays and selects from a list of micro instance classes available for
the selected engine and version.
10. Creates an RDS DB instance that contains a MySQL database and uses the
parameter group
    using the CreateDBInstanceAsync method.
11. Waits for DB instance to be ready using the DescribeDBInstancesAsync
method.
12. Prints out the connection endpoint string for the new DB instance.
13. Creates a snapshot of the DB instance using the CreateDBSnapshotAsync
method.
14. Waits for DB snapshot to be ready using the DescribeDBSnapshots method.
15. Deletes the DB instance using the DeleteDBInstanceAsync method.
16. Waits for DB instance to be deleted using the DescribeDbInstances method.
17. Deletes the parameter group using the DeleteDBParameterGroupAsync.
*/

private static readonly string sepBar = new('-', 80);
private static RDSWrapper rdsWrapper = null!;
private static ILogger logger = null!;
private static readonly string engine = "mysql";
static async Task Main(string[] args)
{
    // Set up dependency injection for the Amazon RDS service.
    using var host = Host.CreateDefaultBuilder(args)
        .ConfigureLogging(logging =>
            logging.AddFilter("System", LogLevel.Debug)
                .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
        .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonRDS>()
                .AddTransient<RDSWrapper>()
        )
        .Build();

```

```
logger = LoggerFactory.Create(builder =>
{
    builder.AddConsole();
}).CreateLogger<RDSInstanceScenario>();

rdsWrapper = host.Services.GetRequiredService<RDSWrapper>();

Console.WriteLine(sepBar);
Console.WriteLine(
    "Welcome to the Amazon Relational Database Service (Amazon RDS) DB
instance scenario example.");
Console.WriteLine(sepBar);

try
{
    var parameterGroupFamily = await ChooseParameterGroupFamily();

    var parameterGroup = await
CreateDbParameterGroup(parameterGroupFamily);

    var parameters = await
DescribeParametersInGroup(parameterGroup.DBParameterGroupName,
        new List<string> { "auto_increment_offset",
"auto_increment_increment" });

    await ModifyParameters(parameterGroup.DBParameterGroupName,
parameters);

    await
DescribeUserSourceParameters(parameterGroup.DBParameterGroupName);

    var engineVersionChoice = await
ChooseDbEngineVersion(parameterGroupFamily);

    var instanceChoice = await ChooseDbInstanceClass(engine,
engineVersionChoice.EngineVersion);

    var newInstanceIdentifier = "Example-Instance-" + DateTime.Now.Ticks;

    var newInstance = await CreateRdsNewInstance(parameterGroup, engine,
engineVersionChoice.EngineVersion,
        instanceChoice.DBInstanceClass, newInstanceIdentifier);
    if (newInstance != null)
```

```
        {
            DisplayConnectionString(newInstance);

            await CreateSnapshot(newInstance);

            await DeleteRdsInstance(newInstance);
        }

        await DeleteParameterGroup(parameterGroup);

        Console.WriteLine("Scenario complete.");
        Console.WriteLine(sepBar);
    }
    catch (Exception ex)
    {
        logger.LogError(ex, "There was a problem executing the scenario.");
    }
}

/// <summary>
/// Choose the RDS DB parameter group family from a list of available
options.
/// </summary>
/// <returns>The selected parameter group family.</returns>
public static async Task<string> ChooseParameterGroupFamily()
{
    Console.WriteLine(sepBar);
    // 1. Get a list of available engines.
    var engines = await rdsWrapper.DescribeDBEngineVersions(engine);

    Console.WriteLine("1. The following is a list of available DB parameter
group families:");
    int i = 1;
    var parameterGroupFamilies = engines.GroupBy(e =>
e.DBParameterGroupFamily).ToList();
    foreach (var parameterGroupFamily in parameterGroupFamilies)
    {
        // List the available parameter group families.
        Console.WriteLine(
            $"{i}\t{i}. Family: {parameterGroupFamily.Key}");
        i++;
    }

    var choiceNumber = 0;
```

```
        while (choiceNumber < 1 || choiceNumber > parameterGroupFamilies.Count)
        {
            Console.WriteLine("Select an available DB parameter group family by
entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }
        var parameterGroupFamilyChoice = parameterGroupFamilies[choiceNumber -
1];
        Console.WriteLine(sepBar);
        return parameterGroupFamilyChoice.Key;
    }

    /// <summary>
    /// Create and get information on a DB parameter group.
    /// </summary>
    /// <param name="dbParameterGroupFamily">The DBParameterGroupFamily for the
new DB parameter group.</param>
    /// <returns>The new DBParameterGroup.</returns>
    public static async Task<DBParameterGroup> CreateDbParameterGroup(string
dbParameterGroupFamily)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine($"2. Create new DB parameter group with family
{dbParameterGroupFamily}:");

        var parameterGroup = await rdsWrapper.CreateDBParameterGroup(
            "ExampleParameterGroup-" + DateTime.Now.Ticks,
            dbParameterGroupFamily, "New example parameter group");

        var groupInfo =
            await rdsWrapper.DescribeDBParameterGroups(parameterGroup
                .DBParameterGroupName);

        Console.WriteLine(
            $"3. New DB parameter group: \n\t{groupInfo[0].Description}, \n\tARN
{groupInfo[0].DBParameterGroupArn}");
        Console.WriteLine(sepBar);
        return parameterGroup;
    }

    /// <summary>
    /// Get and describe parameters from a DBParameterGroup.
    /// </summary>
```

```
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameterNames">Optional specific names of parameters to
describe.</param>
    /// <returns>The list of requested parameters.</returns>
    public static async Task<List<Parameter>> DescribeParametersInGroup(string
parameterGroupName, List<string>? parameterNames = null)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("4. Get some parameters from the group.");
        Console.WriteLine(sepBar);

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName);

        var matchingParameters =
            parameters.Where(p => parameterNames == null ||
parameterNames.Contains(p.ParameterName)).ToList();

        Console.WriteLine("5. Parameter information:");
        matchingParameters.ForEach(p =>
            Console.WriteLine(
                $"\\n\\tParameter: {p.ParameterName}." +
                $"\\n\\tDescription: {p.Description}." +
                $"\\n\\tAllowed Values: {p.AllowedValues}." +
                $"\\n\\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);

        return matchingParameters;
    }

    /// <summary>
    /// Modify a parameter from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameters">The parameters to modify.</param>
    /// <returns>Async task.</returns>
    public static async Task ModifyParameters(string parameterGroupName,
List<Parameter> parameters)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("6. Modify some parameters in the group.");

        foreach (var p in parameters)
```

```
        {
            if (p.IsModifiable && p.DataType == "integer")
            {
                int newValue = 0;
                while (newValue == 0)
                {
                    Console.WriteLine(
                        $"Enter a new value for {p.ParameterName} from the
allowed values {p.AllowedValues} ");

                    var choice = Console.ReadLine();
                    Int32.TryParse(choice, out newValue);
                }

                p.ParameterValue = newValue.ToString();
            }
        }

        await rdsWrapper.ModifyDBParameterGroup(parameterGroupName, parameters);

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Describe the user source parameters in the group.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <returns>Async task.</returns>
    public static async Task DescribeUserSourceParameters(string
parameterGroupName)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("7. Describe user source parameters in the group.");

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName, "user");

        parameters.ForEach(p =>
            Console.WriteLine(
                $"{p.ParameterName}." +
                $"{p.Description}." +
                $"{p.AllowedValues}." +
                $"{p.ParameterValue}."));
    }
}
```

```
        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Choose a DB engine version.
    /// </summary>
    /// <param name="dbParameterGroupFamily">DB parameter group family for engine
    choice.</param>
    /// <returns>The selected engine version.</returns>
    public static async Task<DBEngineVersion> ChooseDbEngineVersion(string
    dbParameterGroupFamily)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed engines.
        var allowedEngines =
            await rdsWrapper.DescribeDBEngineVersions(engine,
    dbParameterGroupFamily);

        Console.WriteLine($"Available DB engine versions for parameter group
    family {dbParameterGroupFamily}:");
        int i = 1;
        foreach (var version in allowedEngines)
        {
            Console.WriteLine(
                $"{i}. Engine: {version.Engine} Version
    {version.EngineVersion}.");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedEngines.Count)
        {
            Console.WriteLine("8. Select an available DB engine version by
    entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var engineChoice = allowedEngines[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return engineChoice;
    }
}
```



```
/// <summary>
/// Choose a DB instance class for a particular engine and engine version.
/// </summary>
/// <param name="engine">DB engine for DB instance choice.</param>
/// <param name="engineVersion">DB engine version for DB instance choice.</
param>
/// <returns>The selected orderable DB instance option.</returns>
public static async Task<OrderableDBInstanceOption>
ChooseDbInstanceClass(string engine, string engineVersion)
{
    Console.WriteLine(sepBar);
    // Get a list of allowed DB instance classes.
    var allowedInstances =
        await rdsWrapper.DescribeOrderableDBInstanceOptions(engine,
engineVersion);

    Console.WriteLine($"8. Available micro DB instance classes for engine
{engine} and version {engineVersion}:");
    int i = 1;

    // Filter to micro instances for this example.
    allowedInstances = allowedInstances
        .Where(i => i.DBInstanceClass.Contains("micro")).ToList();

    foreach (var instance in allowedInstances)
    {
        Console.WriteLine(
            $"{i}. Instance class: {instance.DBInstanceClass} (storage type
{instance.StorageType})");
        i++;
    }

    var choiceNumber = 0;
    while (choiceNumber < 1 || choiceNumber > allowedInstances.Count)
    {
        Console.WriteLine("9. Select an available DB instance class by
entering a number from the list above:");
        var choice = Console.ReadLine();
        Int32.TryParse(choice, out choiceNumber);
    }

    var instanceChoice = allowedInstances[choiceNumber - 1];
    Console.WriteLine(sepBar);
}
```

```
        return instanceChoice;
    }

    /// <summary>
    /// Create a new RDS DB instance.
    /// </summary>
    /// <param name="parameterGroup">Parameter group to use for the DB
instance.</param>
    /// <param name="engineName">Engine to use for the DB instance.</param>
    /// <param name="engineVersion">Engine version to use for the DB instance.</
param>
    /// <param name="instanceClass">Instance class to use for the DB instance.</
param>
    /// <param name="instanceIdentifier">Instance identifier to use for the DB
instance.</param>
    /// <returns>The new DB instance.</returns>
    public static async Task<DBInstance?> CreateRdsNewInstance(DBParameterGroup
parameterGroup,
        string engineName, string engineVersion, string instanceClass, string
instanceIdentifier)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine($"10. Create a new DB instance with identifier
{instanceIdentifier}.");
        bool isInstanceReady = false;
        DBInstance newInstance;
        var instances = await rdsWrapper.DescribeDBInstances();
        isInstanceReady = instances.FirstOrDefault(i =>
            i.DBInstanceIdentifier == instanceIdentifier)?.DBInstanceStatus ==
"available";

        if (isInstanceReady)
        {
            Console.WriteLine("Instance already created.");
            newInstance = instances.First(i => i.DBInstanceIdentifier ==
instanceIdentifier);
        }
        else
        {
            Console.WriteLine("Please enter an admin user name:");
            var username = Console.ReadLine();

            Console.WriteLine("Please enter an admin password:");
            var password = Console.ReadLine();
        }
    }
}
```

```
        newInstance = await rdsWrapper.CreateDBInstance(
            "ExampleInstance",
            instanceIdentifier,
            parameterGroup.DBParameterGroupName,
            engineName,
            engineVersion,
            instanceClass,
            20,
            username,
            password
        );

        // 11. Wait for the DB instance to be ready.

        Console.WriteLine("11. Waiting for DB instance to be ready...");
        while (!isInstanceReady)
        {
            instances = await
rdsWrapper.DescribeDBInstances(instanceIdentifier);
            isInstanceReady = instances.FirstOrDefault()?.DBInstanceStatus ==
"available";
            newInstance = instances.First();
            Thread.Sleep(30000);
        }
    }

    Console.WriteLine(sepBar);
    return newInstance;
}

/// <summary>
/// Display a connection string for an RDS DB instance.
/// </summary>
/// <param name="instance">The DB instance to use to get a connection
string.</param>
public static void DisplayConnectionString(DBInstance instance)
{
    Console.WriteLine(sepBar);
    // Display the connection string.
    Console.WriteLine("12. New DB instance connection string: ");
    Console.WriteLine(
        $"{engine} -h {instance.Endpoint.Address} -P
{instance.Endpoint.Port} "
    );
}
```

```
        + $"-u {instance.MasterUsername} -p [YOUR PASSWORD]\n");

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Create a snapshot from an RDS DB instance.
    /// </summary>
    /// <param name="instance">DB instance to use when creating a snapshot.</
param>
    /// <returns>The snapshot object.</returns>
    public static async Task<DBSnapshot> CreateSnapshot(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Create a snapshot.
        Console.WriteLine($"13. Creating snapshot from DB instance
{instance.DBInstanceIdentifier}.");
        var snapshot = await
rdsWrapper.CreateDBSnapshot(instance.DBInstanceIdentifier, "ExampleSnapshot-" +
DateTime.Now.Ticks);

        // Wait for the snapshot to be available
        bool isSnapshotReady = false;

        Console.WriteLine($"14. Waiting for snapshot to be ready...");
        while (!isSnapshotReady)
        {
            var snapshots = await
rdsWrapper.DescribeDBSnapshots(instance.DBInstanceIdentifier);
            isSnapshotReady = snapshots.FirstOrDefault()?.Status == "available";
            snapshot = snapshots.First();
            Thread.Sleep(30000);
        }

        Console.WriteLine(
            $"Snapshot {snapshot.DBSnapshotIdentifier} status is
{snapshot.Status}.");
        Console.WriteLine(sepBar);
        return snapshot;
    }

    /// <summary>
    /// Delete an RDS DB instance.
    /// </summary>
```

```
/// <param name="instance">The DB instance to delete.</param>
/// <returns>Async task.</returns>
public static async Task DeleteRdsInstance(DBInstance newInstance)
{
    Console.WriteLine(sepBar);
    // Delete the DB instance.
    Console.WriteLine($"15. Delete the DB instance
{newInstance.DBInstanceIdentifier}.");
    await rdsWrapper.DeleteDBInstance(newInstance.DBInstanceIdentifier);

    // Wait for the DB instance to delete.
    Console.WriteLine($"16. Waiting for the DB instance to delete...");
    bool isInstanceDeleted = false;

    while (!isInstanceDeleted)
    {
        var instance = await rdsWrapper.DescribeDBInstances();
        isInstanceDeleted = instance.All(i => i.DBInstanceIdentifier !=
newInstance.DBInstanceIdentifier);
        Thread.Sleep(30000);
    }

    Console.WriteLine("DB instance deleted.");
    Console.WriteLine(sepBar);
}

/// <summary>
/// Delete a DB parameter group.
/// </summary>
/// <param name="parameterGroup">The parameter group to delete.</param>
/// <returns>Async task.</returns>
public static async Task DeleteParameterGroup(DBParameterGroup
parameterGroup)
{
    Console.WriteLine(sepBar);
    // Delete the parameter group.
    Console.WriteLine($"17. Delete the DB parameter group
{parameterGroup.DBParameterGroupName}.");
    await
rdsWrapper.DeleteDBParameterGroup(parameterGroup.DBParameterGroupName);

    Console.WriteLine(sepBar);
}
```

資料庫執行個體動作案例所使用的包裝函式方式。

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
/// DB instance operations.
/// </summary>
public partial class RDSWrapper
{
    private readonly IAmazonRDS _amazonRDS;
    public RDSWrapper(IAmazonRDS amazonRDS)
    {
        _amazonRDS = amazonRDS;
    }

    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
    name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
    string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }

    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
```

```
/// </summary>
/// <param name="engine">Name of the engine.</param>
/// <param name="engineVersion">Version of the engine.</param>
/// <returns>List of OrderableDBInstanceOptions.</returns>
public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
{
    // Use a paginator to get a list of DB instance options.
    var results = new List<OrderableDBInstanceOption>();
    var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
    new DescribeOrderableDBInstanceOptionsRequest()
    {
        Engine = engine,
        EngineVersion = engineVersion,
    });
    // Get the entire list using the paginator.
    await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
    {
        results.Add(instanceOptions);
    }
    return results;
}

/// <summary>
/// Returns a list of DB instances.
/// </summary>
/// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
/// <returns>List of DB instances.</returns>
public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
{
    var results = new List<DBInstance>();
    var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
    new DescribeDBInstancesRequest
    {
        DBInstanceIdentifier = dbInstanceIdentifier
    });
    // Get the entire list using the paginator.
    await foreach (var instances in instancesPaginator.DBInstances)
```

```
    {
        results.Add(instances);
    }
    return results;
}

/// <summary>
/// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
/// to determine when the DB instance is ready to use.
/// </summary>
/// <param name="dbName">Name for the DB instance.</param>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
/// <param name="dbEngine">The engine for the DB instance.</param>
/// <param name="dbEngineVersion">Version for the DB instance.</param>
/// <param name="instanceClass">Class for the DB instance.</param>
/// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
/// <param name="adminName">Admin user name.</param>
/// <param name="adminPassword">Admin user password.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
    string parameterGroupName, string dbEngine, string dbEngineVersion,
    string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
{
    var response = await _amazonRDS.CreateDBInstanceAsync(
        new CreateDBInstanceRequest()
        {
            DBName = dbName,
            DBInstanceIdentifier = dbInstanceIdentifier,
            DBParameterGroupName = parameterGroupName,
            Engine = dbEngine,
            EngineVersion = dbEngineVersion,
            DBInstanceClass = instanceClass,
            AllocatedStorage = allocatedStorage,
            MasterUsername = adminName,
            MasterUserPassword = adminPassword
        });
}
```



```
        return response.DBInstance;
    }

    /// <summary>
    /// Delete a particular DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>DB instance object.</returns>
    public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
    {
        var response = await _amazonRDS.DeleteDBInstanceAsync(
            new DeleteDBInstanceRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
}
```

資料庫參數群組案例所使用的包裝函式方式。

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
/// parameter groups.
/// </summary>
public partial class RDSWrapper
{
    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
}
```

```
public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
name = null)
{
    var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
        new DescribeDBParameterGroupsRequest()
        {
            DBParameterGroupName = name
        });
    return response.DBParameterGroups;
}

/// <summary>
/// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
/// to determine when the DB parameter group is ready to use.
/// </summary>
/// <param name="name">Name of the DB parameter group.</param>
/// <param name="family">Family of the DB parameter group.</param>
/// <param name="description">Description of the DB parameter group.</param>
/// <returns>The new DB parameter group.</returns>
public async Task<DBParameterGroup> CreateDBParameterGroup(
    string name, string family, string description)
{
    var response = await _amazonRDS.CreateDBParameterGroupAsync(
        new CreateDBParameterGroupRequest()
        {
            DBParameterGroupName = name,
            DBParameterGroupFamily = family,
            Description = description
        });
    return response.DBParameterGroup;
}

/// <summary>
/// Update a DB parameter group. Use the action
DescribeDBParameterGroupsAsync
/// to determine when the DB parameter group is ready to use.
/// </summary>
/// <param name="name">Name of the DB parameter group.</param>
```

```
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }

    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
    group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get a list of DB parameters from a specific parameter group.
    /// </summary>
    /// <param name="dbParameterGroupName">Name of a specific DB parameter
    group.</param>
    /// <param name="source">Optional source for selecting parameters.</param>
    /// <returns>List of parameter values.</returns>
```

```

public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
{
    var results = new List<Parameter>();
    var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
        new DescribeDBParametersRequest()
        {
            DBParameterGroupName = dbParameterGroupName,
            Source = source
        });
    // Get the entire list using the paginator.
    await foreach (var parameters in paginateParameters.Parameters)
    {
        results.Add(parameters);
    }
    return results;
}

```

資料庫快照動作案例所使用的包裝函式方式。

```

/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
    snapshots.
/// </summary>
public partial class RDSWrapper
{
    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,

```

```
        DBInstanceIdentifier = dbInstanceIdentifier
    });

    return response.DBSnapshot;
}

/// <summary>
/// Return a list of DB snapshots for a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>List of DB snapshots.</returns>
public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
{
    var results = new List<DBSnapshot>();
    var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
        new DescribeDBSnapshotsRequest()
        {
            DBInstanceIdentifier = dbInstanceIdentifier
        });

    // Get the entire list using the paginator.
    await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
    {
        results.Add(snapshots);
    }
    return results;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for .NET API 參考》中的下列主題。

- [CreateDBInstance](#)
- [創建數據庫 ParameterGroup](#)
- [CreateDBSnapshot](#)
- [DeleteDBInstance](#)
- [刪除資料庫 ParameterGroup](#)
- [描述 B EngineVersions](#)

- [DescribeDBInstances](#)
- [描述 B ParameterGroups](#)
- [DescribeDBParameters](#)
- [DescribeDBSnapshots](#)
- [DescribeOrderable資料庫 InstanceOptions](#)
- [修改資料庫 ParameterGroup](#)

C++

適用於 C++ 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

//! Routine which creates an Amazon RDS instance and demonstrates several
operations
//! on that instance.
/*!
 \sa gettingStartedWithDBInstances()
 \param clientConfiguration: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::gettingStartedWithDBInstances(
    const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::RDS::RDSClient client(clientConfig);

    printAsterisksLine();
    std::cout << "Welcome to the Amazon Relational Database Service (Amazon RDS)"
                << std::endl;
    std::cout << "get started with DB instances demo." << std::endl;
    printAsterisksLine();
}
```

```

std::cout << "Checking for an existing DB parameter group named '" <<
    PARAMETER_GROUP_NAME << "'." << std::endl;
Aws::String dbParameterGroupFamily("Undefined");
bool parameterGroupFound = true;
{
    // 1. Check if the DB parameter group already exists.
    Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

    Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
        client.DescribeDBParameterGroups(request);

    if (outcome.IsSuccess()) {
        std::cout << "DB parameter group named '" <<
            PARAMETER_GROUP_NAME << "' already exists." << std::endl;
        dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
    }
    else if (outcome.GetError().GetErrorType() ==
        Aws::RDS::RDSErrors::D_B_PARAMETER_GROUP_NOT_FOUND_FAULT) {
        std::cout << "DB parameter group named '" <<
            PARAMETER_GROUP_NAME << "' does not exist." << std::endl;
        parameterGroupFound = false;
    }
    else {
        std::cerr << "Error with RDS::DescribeDBParameterGroups. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
}

if (!parameterGroupFound) {
    Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

    // 2. Get available engine versions for the specified engine.
    if (!getDBEngineVersions(DB_ENGINE, NO_PARAMETER_GROUP_FAMILY,
        engineVersions, client)) {
        return false;
    }

    std::cout << "Getting available database engine versions for " <<
    DB_ENGINE
        << "."

```

```

        << std::endl;
    std::vector<Aws::String> families;
    for (const Aws::RDS::Model::DBEngineVersion &version: engineVersions) {
        Aws::String family = version.GetDBParameterGroupFamily();
        if (std::find(families.begin(), families.end(), family) ==
            families.end()) {
            families.push_back(family);
            std::cout << " " << families.size() << ": " << family <<
std::endl;
        }
    }

    int choice = askQuestionForIntRange("Which family do you want to use? ",
1,
                                     static_cast<int>(families.size()));
    dbParameterGroupFamily = families[choice - 1];
}
if (!parameterGroupFound) {
    // 3. Create a DB parameter group.
    Aws::RDS::Model::CreateDBParameterGroupRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetDBParameterGroupFamily(dbParameterGroupFamily);
    request.SetDescription("Example parameter group.");

    Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
        client.CreateDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully created."
            << std::endl;
    }
    else {
        std::cerr << "Error with RDS::CreateDBParameterGroup. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
}

printAsterisksLine();
std::cout << "Let's set some parameter values in your parameter group."
    << std::endl;

Aws::String marker;

```



```

Aws::Vector<Aws::RDS::Model::Parameter> autoIncrementParameters;
// 4. Get the parameters in the DB parameter group.
if (!getDBParameters(PARAMETER_GROUP_NAME, AUTO_INCREMENT_PREFIX, NO_SOURCE,
                    autoIncrementParameters,
                    client)) {
    cleanUpResources(PARAMETER_GROUP_NAME, "", client);
    return false;
}

Aws::Vector<Aws::RDS::Model::Parameter> updateParameters;

for (Aws::RDS::Model::Parameter &autoIncParameter: autoIncrementParameters) {
    if (autoIncParameter.GetIsModifiable() &&
        (autoIncParameter.GetDataTypes() == "integer")) {
        std::cout << "The " << autoIncParameter.GetParameterName()
                  << " is described as: " <<
                  autoIncParameter.GetDescription() << "." << std::endl;
        if (autoIncParameter.ParameterValueHasBeenSet()) {
            std::cout << "The current value is "
                      << autoIncParameter.GetParameterValue()
                      << "." << std::endl;
        }
        std::vector<int> splitValues = splitToInts(
            autoIncParameter.GetAllowedValues(), '-');
        if (splitValues.size() == 2) {
            int newValue = askQuestionForIntRange(
                Aws::String("Enter a new value in the range ") +
                autoIncParameter.GetAllowedValues() + ": ",
                splitValues[0], splitValues[1]);
            autoIncParameter.SetParameterValue(std::to_string(newValue));
            updateParameters.push_back(autoIncParameter);
        }
        else {
            std::cerr << "Error parsing " <<
            autoIncParameter.GetAllowedValues()
                          << std::endl;
        }
    }
}

{
    // 5. Modify the auto increment parameters in the group.
    Aws::RDS::Model::ModifyDBParameterGroupRequest request;

```

```

    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetParameters(updateParameters);

    Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
        client.ModifyDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully modified."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
}

std::cout
    << "You can get a list of parameters you've set by specifying a
source of 'user'."
    << std::endl;

    Aws::Vector<Aws::RDS::Model::Parameter> userParameters;
    // 6. Display the modified parameters in the group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, NO_NAME_PREFIX, "user",
userParameters,
                        client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    for (const auto &userParameter: userParameters) {
        std::cout << " " << userParameter.GetParameterName() << ", " <<
            userParameter.GetDescription() << ", parameter value - "
            << userParameter.GetParameterValue() << std::endl;
    }

    printAsterisksLine();
    std::cout << "Checking for an existing DB instance." << std::endl;

    Aws::RDS::Model::DBInstance dbInstance;
    // 7. Check if the DB instance already exists.
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);

```

```

    return false;
}

if (dbInstance.DbInstancePortHasBeenSet()) {
    std::cout << "The DB instance already exists." << std::endl;
}
else {
    std::cout << "Let's create a DB instance." << std::endl;
    const Aws::String administratorName = askQuestion(
        "Enter an administrator username for the database: ");
    const Aws::String administratorPassword = askQuestion(
        "Enter a password for the administrator (at least 8 characters):
");
    Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

    // 8. Get a list of available engine versions.
    if (!getDBEngineVersions(DB_ENGINE, dbParameterGroupFamily,
engineVersions,
                                client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    std::cout << "The available engines for your parameter group are:" <<
std::endl;

    int index = 1;
    for (const Aws::RDS::Model::DBEngineVersion &engineVersion:
engineVersions) {
        std::cout << "  " << index << ": " <<
engineVersion.GetEngineVersion()
                << std::endl;
        ++index;
    }
    int choice = askQuestionForIntRange("Which engine do you want to use? ",
1,
static_cast<int>(engineVersions.size()));
    const Aws::RDS::Model::DBEngineVersion engineVersion =
engineVersions[choice -
1];

    Aws::String dbInstanceClass;
    // 9. Get a list of micro instance classes.

```

```

    if (!chooseMicroDBInstanceClass(engineVersion.GetEngine(),
                                     engineVersion.GetEngineVersion(),
                                     dbInstanceClass,
                                     client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    std::cout << "Creating a DB instance named '" << DB_INSTANCE_IDENTIFIER
              << "' and database '" << DB_NAME << "'.\n"
              << "The DB instance is configured to use your custom parameter
group '"
              << PARAMETER_GROUP_NAME << "',\n"
              << "selected engine version " <<
engineVersion.GetEngineVersion()
              << ",\n"
              << "selected DB instance class '" << dbInstanceClass << "',"
              << " and " << DB_ALLOCATED_STORAGE << " GiB of " <<
DB_STORAGE_TYPE
              << " storage.\nThis typically takes several minutes." <<
std::endl;

    Aws::RDS::Model::CreateDBInstanceRequest request;
    request.SetDBName(DB_NAME);
    request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetEngine(engineVersion.GetEngine());
    request.SetEngineVersion(engineVersion.GetEngineVersion());
    request.SetDBInstanceClass(dbInstanceClass);
    request.SetStorageType(DB_STORAGE_TYPE);
    request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
    request.SetMasterUsername(administratorName);
    request.SetMasterUserPassword(administratorPassword);

    Aws::RDS::Model::CreateDBInstanceOutcome outcome =
        client.CreateDBInstance(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB instance creation has started."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::CreateDBInstance. "
                  << outcome.GetError().GetMessage()

```

```
        << std::endl;
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }
}

std::cout << "Waiting for the DB instance to become available." << std::endl;

int counter = 0;
// 11. Wait for the DB instance to become available.
do {
    std::this_thread::sleep_for(std::chrono::seconds(1));
    ++counter;
    if (counter > 900) {
        std::cerr << "Wait for instance to become available timed out after "
            << counter
            << " seconds." << std::endl;
        cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
        return false;
    }

    dbInstance = Aws::RDS::Model::DBInstance();
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
        return false;
    }

    if ((counter % 20) == 0) {
        std::cout << "Current DB instance status is '"
            << dbInstance.GetDBInstanceStatus()
            << "' after " << counter << " seconds." << std::endl;
    }
} while (dbInstance.GetDBInstanceStatus() != "available");

if (dbInstance.GetDBInstanceStatus() == "available") {
    std::cout << "The DB instance has been created." << std::endl;
}

printAsterisksLine();

// 12. Display the connection string that can be used to connect a 'mysql'
shell to the database.
```

```
displayConnection(dbInstance);

printAsterisksLine();

if (askYesNoQuestion(
    "Do you want to create a snapshot of your DB instance (y/n)? ") {
    Aws::String snapshotID(DB_INSTANCE_IDENTIFIER + "-" +
        Aws::String(Aws::Utils::UUID::RandomUUID()));
    {
        std::cout << "Creating a snapshot named " << snapshotID << "." <<
std::endl;
        std::cout << "This typically takes a few minutes." << std::endl;

        // 13. Create a snapshot of the DB instance.
        Aws::RDS::Model::CreateDBSnapshotRequest request;
        request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
        request.SetDBSnapshotIdentifier(snapshotID);

        Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
            client.CreateDBSnapshot(request);

        if (outcome.IsSuccess()) {
            std::cout << "Snapshot creation has started."
                << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBSnapshot. "
                << outcome.GetError().GetMessage()
                << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
            return false;
        }
    }

    std::cout << "Waiting for snapshot to become available." << std::endl;

    Aws::RDS::Model::DBSnapshot snapshot;
    counter = 0;
    do {
        std::this_thread::sleep_for(std::chrono::seconds(1));
        ++counter;
        if (counter > 600) {
            std::cerr << "Wait for snapshot to be available timed out after "
```

```
        << counter
        << " seconds." << std::endl;
    cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
    return false;
}

// 14. Wait for the snapshot to become available.
Aws::RDS::Model::DescribeDBSnapshotsRequest request;
request.SetDBSnapshotIdentifier(snapshotID);

Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
    client.DescribeDBSnapshots(request);

if (outcome.IsSuccess()) {
    snapshot = outcome.GetResult().GetDBSnapshots()[0];
}
else {
    std::cerr << "Error with RDS::DescribeDBSnapshots. "
        << outcome.GetError().GetMessage()
        << std::endl;
    cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
    return false;
}

if ((counter % 20) == 0) {
    std::cout << "Current snapshot status is '"
        << snapshot.GetStatus()
        << "' after " << counter << " seconds." << std::endl;
}
} while (snapshot.GetStatus() != "available");

if (snapshot.GetStatus() != "available") {
    std::cout << "A snapshot has been created." << std::endl;
}
}

printAsterisksLine();

bool result = true;
if (askYesNoQuestion(
    "Do you want to delete the DB instance and parameter group (y/n)? "))
{
```

```

        result = cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
    }

    return result;
}

//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
\sa getDBParameters()
\param parameterGroupName: The name of the parameter group.
\param namePrefix: Prefix string to filter results by parameter name.
\param source: A source such as 'user', ignored if empty.
\param parametersResult: Vector of 'Parameter' objects returned by the routine.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                const Aws::String &namePrefix,
                                const Aws::String &source,
                                Aws::Vector<Aws::RDS::Model::Parameter>
&parametersResult,
                                const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
            client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {

```



```

        parametersResult.push_back(parameter);
    }
}
else {
    parametersResult.push_back(parameter);
}
}

marker = outcome.GetResult().GetMarker();
}
else {
    std::cerr << "Error with RDS::DescribeDBParameters. "
                << outcome.GetError().GetMessage()
                << std::endl;
    return false;
}
} while (!marker.empty());

return true;
}

//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                       const Aws::String &parameterGroupFamily,

                                       Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                       const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }
}

```

```

engineVersionsResult.clear();
Aws::String marker; // Used for pagination.

do {
    if (!marker.empty()) {
        request.SetMarker(marker);
    }

    Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
        client.DescribeDBEngineVersions(request);

    if (outcome.IsSuccess()) {
        auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
        engineVersionsResult.insert(engineVersionsResult.end(),
engineVersions.begin(),
                                engineVersions.end());
        marker = outcome.GetResult().GetMarker();
    }
    else {
        std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                    << outcome.GetError().GetMessage()
                    << std::endl;
        return false;
    }
} while (!marker.empty());

return true;
}

//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {

```

```

    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
        client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
        Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
            << outcome.GetError().GetMessage()
            << std::endl;
    }
    // This example does not log an error if the DB instance does not exist.
    // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}

//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
    \sa chooseMicroDBInstanceClass()
    \param engineName: The DB engine name.
    \param engineVersion: The DB engine version.
    \param dbInstanceClass: String for DB instance class chosen by the user.
    \param client: 'RDSClient' instance.
    \return bool: Successful completion.
    */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {

```

```

    Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
    request.SetEngine(engine);
    request.SetEngineVersion(engineVersion);
    if (!marker.empty()) {
        request.SetMarker(marker);
    }

    Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
        client.DescribeOrderableDBInstanceOptions(request);

    if (outcome.IsSuccess()) {
        const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
&options =
            outcome.GetResult().GetOrderableDBInstanceOptions();
        for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
options) {
            const Aws::String &instanceClass = option.GetDBInstanceClass();
            if (instanceClass.find("micro") != std::string::npos) {
                if (std::find(instanceClasses.begin(), instanceClasses.end(),
instanceClass) ==
instanceClasses.end()) {
                    instanceClasses.push_back(instanceClass);
                }
            }
        }
        marker = outcome.GetResult().GetMarker();
    }
    else {
        std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
} while (!marker.empty());

std::cout << "The available micro DB instance classes for your database
engine are:"
    << std::endl;
for (int i = 0; i < instanceClasses.size(); ++i) {
    std::cout << "    " << i + 1 << ": " << instanceClasses[i] << std::endl;
}

int choice = askQuestionForIntRange(
    "Which micro DB instance class do you want to use? ",

```

```

        1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}

//! Routine which deletes resources created by the scenario.
/*!
\sa cleanUpResources()
\param parameterGroupName: A parameter group name, this may be empty.
\param dbInstanceIdentifier: A DB instance identifier, this may be empty.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::cleanUpResources(const Aws::String &parameterGroupName,
                                   const Aws::String &dbInstanceIdentifier,
                                   const Aws::RDS::RDSClient &client) {

    bool result = true;
    if (!dbInstanceIdentifier.empty()) {
        {
            // 15. Delete the DB instance.
            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                    << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                    << outcome.GetError().GetMessage()
                    << std::endl;
                result = false;
            }
        }
    }

    std::cout
        << "Waiting for DB instance to delete before deleting the
parameter group."
        << std::endl;

```

```
std::cout << "This may take a while." << std::endl;

int counter = 0;
Aws::RDS::Model::DBInstance dbInstance;
do {
    std::this_thread::sleep_for(std::chrono::seconds(1));
    ++counter;
    if (counter > 800) {
        std::cerr << "Wait for instance to delete timed out after " <<
counter
                << " seconds." << std::endl;
        return false;
    }

    dbInstance = Aws::RDS::Model::DBInstance();
    // 16. Wait for the DB instance to be deleted.
    if (!describeDBInstance(dbInstanceIdentifier, dbInstance, client)) {
        return false;
    }

    if (dbInstance.DBInstanceIdentifierHasBeenSet() && (counter % 20) ==
0) {
        std::cout << "Current DB instance status is '"
                << dbInstance.GetDBInstanceStatus()
                << "' after " << counter << " seconds." << std::endl;
    }
} while (dbInstance.DBInstanceIdentifierHasBeenSet());
}

if (!parameterGroupName.empty()) {
    // 17. Delete the parameter group.
    Aws::RDS::Model::DeleteDBParameterGroupRequest request;
    request.SetDBParameterGroupName(parameterGroupName);

    Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
        client.DeleteDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully deleted."
                << std::endl;
    }
    else {
        std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                << outcome.GetError().GetMessage()

```


```
        << std::endl;
        result = false;
    }
}

return result;
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for C++ API 參考》中的下列主題。
 - [CreateDBInstance](#)
 - [創建數據庫 ParameterGroup](#)
 - [CreateDBSnapshot](#)
 - [DeleteDBInstance](#)
 - [刪除資料庫 ParameterGroup](#)
 - [描述 B EngineVersions](#)
 - [DescribeDBInstances](#)
 - [描述 B ParameterGroups](#)
 - [DescribeDBParameters](#)
 - [DescribeDBSnapshots](#)
 - [DescribeOrderable資料庫 InstanceOptions](#)
 - [修改資料庫 ParameterGroup](#)

Go

SDK for Go V2

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

在命令提示中執行互動式案例。

```
// GetStartedInstances is an interactive example that shows you how to use the
// AWS SDK for Go
// with Amazon Relation Database Service (Amazon RDS) to do the following:
//
// 1. Create a custom DB parameter group and set parameter values.
// 2. Create a DB instance that is configured to use the parameter group. The DB
// instance
// also contains a database.
// 3. Take a snapshot of the DB instance.
// 4. Delete the DB instance and parameter group.
type GetStartedInstances struct {
    sdkConfig aws.Config
    instances actions.DbInstances
    questioner demotools.IQuestioner
    helper      IScenarioHelper
    isTestRun  bool
}

// NewGetStartedInstances constructs a GetStartedInstances instance from a
// configuration.
// It uses the specified config to get an Amazon RDS
// client and create wrappers for the actions used in the scenario.
func NewGetStartedInstances(sdkConfig aws.Config, questioner
    demotools.IQuestioner,
    helper IScenarioHelper) GetStartedInstances {
    rdsClient := rds.NewFromConfig(sdkConfig)
    return GetStartedInstances{
        sdkConfig:  sdkConfig,
        instances:  actions.DbInstances{RdsClient: rdsClient},
        questioner: questioner,
        helper:     helper,
    }
}

// Run runs the interactive scenario.
func (scenario GetStartedInstances) Run(dbEngine string, parameterGroupName
    string,
    instanceName string, dbName string) {
    defer func() {
        if r := recover(); r != nil {
            log.Println("Something went wrong with the demo.")
        }
    }()
}
```



```
log.Println(strings.Repeat("-", 88))
log.Println("Welcome to the Amazon Relational Database Service (Amazon RDS) DB
Instance demo.")
log.Println(strings.Repeat("-", 88))

parameterGroup := scenario.CreateParameterGroup(dbEngine, parameterGroupName)
scenario.SetUserParameters(parameterGroupName)
instance := scenario.CreateInstance(instanceName, dbEngine, dbName,
parameterGroup)
scenario.DisplayConnection(instance)
scenario.CreateSnapshot(instance)
scenario.Cleanup(instance, parameterGroup)

log.Println(strings.Repeat("-", 88))
log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}

// CreateParameterGroup shows how to get available engine versions for a
// specified
// database engine and create a DB parameter group that is compatible with a
// selected engine family.
func (scenario GetStartedInstances) CreateParameterGroup(dbEngine string,
parameterGroupName string) *types.DBParameterGroup {

log.Printf("Checking for an existing DB parameter group named %v.\n",
parameterGroupName)
parameterGroup, err := scenario.instances.GetParameterGroup(parameterGroupName)
if err != nil {
panic(err)
}
if parameterGroup == nil {
log.Printf("Getting available database engine versions for %v.\n", dbEngine)
engineVersions, err := scenario.instances.GetEngineVersions(dbEngine, "")
if err != nil {
panic(err)
}

familySet := map[string]struct{}{}
for _, family := range engineVersions {
familySet[*family.DBParameterGroupFamily] = struct{}{}
}
var families []string
for family := range familySet {
```

```

    families = append(families, family)
}
sort.Strings(families)
familyIndex := scenario.questioner.AskChoice("Which family do you want to use?
\n", families)
log.Println("Creating a DB parameter group.")
_, err = scenario.instances.CreateParameterGroup(
    parameterGroupName, families[familyIndex], "Example parameter group.")
if err != nil {
    panic(err)
}
parameterGroup, err = scenario.instances.GetParameterGroup(parameterGroupName)
if err != nil {
    panic(err)
}
}
log.Printf("Parameter group %v:\n", *parameterGroup.DBParameterGroupFamily)
log.Printf("\tName: %v\n", *parameterGroup.DBParameterGroupName)
log.Printf("\tARN: %v\n", *parameterGroup.DBParameterGroupArn)
log.Printf("\tFamily: %v\n", *parameterGroup.DBParameterGroupFamily)
log.Printf("\tDescription: %v\n", *parameterGroup.Description)
log.Println(strings.Repeat("-", 88))
return parameterGroup
}

// SetUserParameters shows how to get the parameters contained in a custom
parameter
// group and update some of the parameter values in the group.
func (scenario GetStartedInstances) SetUserParameters(parameterGroupName string)
{
    log.Println("Let's set some parameter values in your parameter group.")
    dbParameters, err := scenario.instances.GetParameters(parameterGroupName, "")
    if err != nil {
        panic(err)
    }
    var updateParams []types.Parameter
    for _, dbParam := range dbParameters {
        if strings.HasPrefix(*dbParam.ParameterName, "auto_increment") &&
            dbParam.IsModifiable && *dbParam.DataType == "integer" {
            log.Printf("The %v parameter is described as:\n\t%v",
                *dbParam.ParameterName, *dbParam.Description)
            rangeSplit := strings.Split(*dbParam.AllowedValues, "-")
            lower, _ := strconv.Atoi(rangeSplit[0])
            upper, _ := strconv.Atoi(rangeSplit[1])

```

```
    newValue := scenario.questioner.AskInt(
        fmt.Sprintf("Enter a value between %v and %v:", lower, upper),
        demotools.InIntRange{Lower: lower, Upper: upper})
    dbParam.ParameterValue = aws.String(strconv.Itoa(newValue))
    updateParams = append(updateParams, dbParam)
}
}
err = scenario.instances.UpdateParameters(parameterGroupName, updateParams)
if err != nil {
    panic(err)
}
log.Println("To get a list of parameters that you set previously, specify a
source of 'user'.")
userParameters, err := scenario.instances.GetParameters(parameterGroupName,
"user")
if err != nil {
    panic(err)
}
log.Println("Here are the parameters you set:")
for _, param := range userParameters {
    log.Printf("\t%v: %v\n", *param.ParameterName, *param.ParameterValue)
}
log.Println(strings.Repeat("-", 88))
}

// CreateInstance shows how to create a DB instance that contains a database of a
// specified type. The database is also configured to use a custom DB parameter
// group.
func (scenario GetStartedInstances) CreateInstance(instanceName string, dbEngine
string,
dbName string, parameterGroup *types.DBParameterGroup) *types.DBInstance {

    log.Println("Checking for an existing DB instance.")
    instance, err := scenario.instances.GetInstance(instanceName)
    if err != nil {
        panic(err)
    }
    if instance == nil {
        adminUsername := scenario.questioner.Ask(
            "Enter an administrator username for the database: ", demotools.NotEmpty{})
        adminPassword := scenario.questioner.AskPassword(
            "Enter a password for the administrator (at least 8 characters): ", 7)
        engineVersions, err := scenario.instances.GetEngineVersions(dbEngine,
            *parameterGroup.DBParameterGroupFamily)
    }
```

```
if err != nil {
    panic(err)
}
var engineChoices []string
for _, engine := range engineVersions {
    engineChoices = append(engineChoices, *engine.EngineVersion)
}
engineIndex := scenario.questioner.AskChoice(
    "The available engines for your parameter group are:\n", engineChoices)
engineSelection := engineVersions[engineIndex]
instOpts, err :=
scenario.instances.GetOrderableInstances(*engineSelection.Engine,
    *engineSelection.EngineVersion)
if err != nil {
    panic(err)
}
optSet := map[string]struct{}{}
for _, opt := range instOpts {
    if strings.Contains(*opt.DBInstanceClass, "micro") {
        optSet[*opt.DBInstanceClass] = struct{}{}
    }
}
var optChoices []string
for opt := range optSet {
    optChoices = append(optChoices, opt)
}
sort.Strings(optChoices)
optIndex := scenario.questioner.AskChoice(
    "The available micro DB instance classes for your database engine are:\n",
optChoices)
storageType := "standard"
allocatedStorage := int32(5)
log.Printf("Creating a DB instance named %v and database %v.\n"+
    "The DB instance is configured to use your custom parameter group %v,\n"+
    "selected engine %v,\n"+
    "selected DB instance class %v,"+
    "and %v GiB of %v storage.\n"+
    "This typically takes several minutes.",
instanceName, dbName, *parameterGroup.DBParameterGroupName,
*engineSelection.EngineVersion,
optChoices[optIndex], allocatedStorage, storageType)
instance, err = scenario.instances.CreateInstance(
instanceName, dbName, *engineSelection.Engine, *engineSelection.EngineVersion,
*parameterGroup.DBParameterGroupName, optChoices[optIndex], storageType,
```

```

    allocatedStorage, adminUsername, adminPassword)
if err != nil {
    panic(err)
}
for *instance.DBInstanceStatus != "available" {
    scenario.helper.Pause(30)
    instance, err = scenario.instances.GetInstance(instanceName)
    if err != nil {
        panic(err)
    }
}
log.Println("Instance created and available.")
}
log.Println("Instance data:")
log.Printf("\tDBInstanceIdentifier: %v\n", *instance.DBInstanceIdentifier)
log.Printf("\tARN: %v\n", *instance.DBInstanceArn)
log.Printf("\tStatus: %v\n", *instance.DBInstanceStatus)
log.Printf("\tEngine: %v\n", *instance.Engine)
log.Printf("\tEngine version: %v\n", *instance.EngineVersion)
log.Println(strings.Repeat("-", 88))
return instance
}

// DisplayConnection displays connection information about a DB instance and tips
// on how to connect to it.
func (scenario GetStartedInstances) DisplayConnection(instance *types.DBInstance)
{
    log.Println(
        "You can now connect to your database by using your favorite MySQL client.\n" +
        "One way to connect is by using the 'mysql' shell on an Amazon EC2 instance\n"
    +
        "that is running in the same VPC as your DB instance. Pass the endpoint,\n" +
        "port, and administrator username to 'mysql'. Then, enter your password\n" +
        "when prompted:")
    log.Printf("\n\tmysql -h %v -P %v -u %v -p\n",
        *instance.Endpoint.Address, instance.Endpoint.Port, *instance.MasterUsername)
    log.Println("For more information, see the User Guide for RDS:\n" +
        "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
        CHAP\_GettingStarted.CreatingConnecting.MySQL.html#CHAP\_GettingStarted.Connecting.MySQL")
    log.Println(strings.Repeat("-", 88))
}

// CreateSnapshot shows how to create a DB instance snapshot and wait until it's
available.

```

```

func (scenario GetStartedInstances) CreateSnapshot(instance *types.DBInstance) {
    if scenario.questioner.AskBool(
        "Do you want to create a snapshot of your DB instance (y/n)? ", "y") {
        snapshotId := fmt.Sprintf("%v-%v", *instance.DBInstanceIdentifier,
            scenario.helper.UniqueId())
        log.Printf("Creating a snapshot named %v. This typically takes a few minutes.
\n", snapshotId)
        snapshot, err :=
            scenario.instances.CreateSnapshot(*instance.DBInstanceIdentifier, snapshotId)
        if err != nil {
            panic(err)
        }
        for *snapshot.Status != "available" {
            scenario.helper.Pause(30)
            snapshot, err = scenario.instances.GetSnapshot(snapshotId)
            if err != nil {
                panic(err)
            }
        }
        log.Println("Snapshot data:")
        log.Printf("\tDBSnapshotIdentifier: %v\n", *snapshot.DBSnapshotIdentifier)
        log.Printf("\tARN: %v\n", *snapshot.DBSnapshotArn)
        log.Printf("\tStatus: %v\n", *snapshot.Status)
        log.Printf("\tEngine: %v\n", *snapshot.Engine)
        log.Printf("\tEngine version: %v\n", *snapshot.EngineVersion)
        log.Printf("\tDBInstanceIdentifier: %v\n", *snapshot.DBInstanceIdentifier)
        log.Printf("\tSnapshotCreateTime: %v\n", *snapshot.SnapshotCreateTime)
        log.Println(strings.Repeat("-", 88))
    }
}

// Cleanup shows how to clean up a DB instance and DB parameter group.
// Before the DB parameter group can be deleted, all associated DB instances must
// first be deleted.
func (scenario GetStartedInstances) Cleanup(
    instance *types.DBInstance, parameterGroup *types.DBParameterGroup) {

    if scenario.questioner.AskBool(
        "\nDo you want to delete the database instance and parameter group (y/n)? ",
        "y") {
        log.Printf("Deleting database instance %v.\n", *instance.DBInstanceIdentifier)
        err := scenario.instances.DeleteInstance(*instance.DBInstanceIdentifier)
        if err != nil {
            panic(err)
        }
    }
}

```

```

}
log.Println(
    "Waiting for the DB instance to delete. This typically takes several
minutes.")
for instance != nil {
    scenario.helper.Pause(30)
    instance, err = scenario.instances.GetInstance(*instance.DBInstanceIdentifier)
    if err != nil {
        panic(err)
    }
}
log.Printf("Deleting parameter group %v.",
*parameterGroup.DBParameterGroupName)
err =
scenario.instances.DeleteParameterGroup(*parameterGroup.DBParameterGroupName)
if err != nil {
    panic(err)
}
}
}
}

```

定義案例所呼叫的函數以管理 Amazon RDS 動作。

```

type DbInstances struct {
    RdsClient *rds.Client
}

// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
    *types.DBParameterGroup, error) {
    output, err := instances.RdsClient.DescribeDBParameterGroups(
        context.TODO(), &rds.DescribeDBParameterGroupsInput{
            DBParameterGroupName: aws.String(parameterGroupName),
        })
    if err != nil {
        var notFoundError *types.DBParameterGroupNotFoundFault
        if errors.As(err, &notFoundError) {
            log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
            err = nil
        }
    }
}

```

```
    } else {
        log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
    }
    return nil, err
} else {
    return &output.DBParameterGroups[0], err
}
}

// CreateParameterGroup creates a DB parameter group that is based on the
// specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
    parameterGroupName string, parameterGroupFamily string, description string) (
    *types.DBParameterGroup, error) {

    output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
        &rds.CreateDBParameterGroupInput{
            DBParameterGroupName:    aws.String(parameterGroupName),
            DBParameterGroupFamily: aws.String(parameterGroupFamily),
            Description:            aws.String(description),
        })
    if err != nil {
        log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
        return nil, err
    } else {
        return output.DBParameterGroup, err
    }
}

// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
error {
    _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
        &rds.DeleteDBParameterGroupInput{
            DBParameterGroupName: aws.String(parameterGroupName),
        })
    if err != nil {
        log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
        return err
    }
}
```



```
    } else {
        return nil
    }
}

// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
string) (
[]types.Parameter, error) {

var output *rds.DescribeDBParametersOutput
var params []types.Parameter
var err error
parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
&rds.DescribeDBParametersInput{
    DBParameterGroupName: aws.String(parameterGroupName),
    Source:                 aws.String(source),
})
for parameterPaginator.HasMorePages() {
    output, err = parameterPaginator.NextPage(context.TODO())
    if err != nil {
        log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
        break
    } else {
        params = append(params, output.Parameters...)
    }
}
return params, err
}

// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
[]types.Parameter) error {
_, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
&rds.ModifyDBParameterGroupInput{
    DBParameterGroupName: aws.String(parameterGroupName),
    Parameters:           params,
})
if err != nil {
    log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
}
```

```
    return err
  } else {
    return nil
  }
}

// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
string) (
  *types.DBSnapshot, error) {
  output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
&rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
})
  if err != nil {
    log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
    return nil, err
  } else {
    return output.DBSnapshot, nil
  }
}

// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
(*types.DBSnapshot, error) {
  output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
&rds.DescribeDBSnapshotsInput{
  DBSnapshotIdentifier: aws.String(snapshotName),
})
  if err != nil {
    log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
    return nil, err
  } else {
    return &output.DBSnapshots[0], nil
  }
}
```

```
// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
dbEngine string, dbEngineVersion string, parameterGroupName string,
dbInstanceClass string,
storageType string, allocatedStorage int32, adminName string, adminPassword
string) (
*types.DBInstance, error) {
output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
&rds.CreateDBInstanceInput{
DBInstanceIdentifier: aws.String(instanceName),
DBName:                aws.String(dbName),
DBParameterGroupName: aws.String(parameterGroupName),
Engine:                aws.String(dbEngine),
EngineVersion:        aws.String(dbEngineVersion),
DBInstanceClass:      aws.String(dbInstanceClass),
StorageType:          aws.String(storageType),
AllocatedStorage:     aws.Int32(allocatedStorage),
MasterUsername:       aws.String(adminName),
MasterUserPassword:   aws.String(adminPassword),
})
if err != nil {
log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
return nil, err
} else {
return output.DBInstance, nil
}
}

// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
*types.DBInstance, error) {
output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
&rds.DescribeDBInstancesInput{
DBInstanceIdentifier: aws.String(instanceName),
})
if err != nil {
var notFoundError *types.DBInstanceNotFoundFault
if errors.As(err, &notFoundError) {
log.Printf("DB instance %v does not exist.\n", instanceName)
err = nil
} else {
log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
}
```

```
    }
    return nil, err
} else {
    return &output.DBInstances[0], nil
}
}

// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
    _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
        &rds.DeleteDBInstanceInput{
            DBInstanceIdentifier: aws.String(instanceName),
            SkipFinalSnapshot:   true,
            DeleteAutomatedBackups: aws.Bool(true),
        })
    if err != nil {
        log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
        return err
    } else {
        return nil
    }
}

// GetEngineVersions gets database engine versions that are available for the
// specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
    parameterGroupFamily string) (
    []types.DBEngineVersion, error) {
    output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
        &rds.DescribeDBEngineVersionsInput{
            Engine: aws.String(engine),
            DBParameterGroupFamily: aws.String(parameterGroupFamily),
        })
    if err != nil {
        log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
        return nil, err
    } else {
        return output.DBEngineVersions, nil
    }
}
```

```
}

// GetOrderableInstances uses a paginator to get DB instance options that can be
// used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
string) (
[]types.OrderableDBInstanceOption, error) {

var output *rds.DescribeOrderableDBInstanceOptionsOutput
var instanceOptions []types.OrderableDBInstanceOption
var err error
orderablePaginator :=
rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
&rds.DescribeOrderableDBInstanceOptionsInput{
    Engine:      aws.String(engine),
    EngineVersion: aws.String(engineVersion),
})
for orderablePaginator.HasMorePages() {
    output, err = orderablePaginator.NextPage(context.TODO())
    if err != nil {
        log.Printf("Couldn't get orderable DB instance options: %v\n", err)
        break
    } else {
        instanceOptions = append(instanceOptions,
output.OrderableDBInstanceOptions...)
    }
}
return instanceOptions, err
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Go API 參考》中的下列主題。
 - [CreateDBInstance](#)
 - [創建數據庫 ParameterGroup](#)
 - [CreateDBSnapshot](#)
 - [DeleteDBInstance](#)
 - [刪除資料庫 ParameterGroup](#)

- [描述 B EngineVersions](#)
- [DescribeDBInstances](#)
- [描述 B ParameterGroups](#)
- [DescribeDBParameters](#)
- [DescribeDBSnapshots](#)
- [DescribeOrderable資料庫 InstanceOptions](#)
- [修改資料庫 ParameterGroup](#)

Java

適用於 Java 2.x 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

執行多個操作。

```
import com.google.gson.Gson;
import
    software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotRequest;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotResponse;
import software.amazon.awssdk.services.rds.model.DBEngineVersion;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.DBParameterGroup;
import software.amazon.awssdk.services.rds.model.DBSnapshot;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsRequest;
import
    software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsResponse;
```

```
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import
    software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsResponse;
import
    software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsResponse;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.OrderableDBInstanceOption;
import software.amazon.awssdk.services.rds.model.Parameter;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupRequest;
import
    software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupRequest;
import
    software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbParameterGroupRequest;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating\_how-services-use-secrets\_RS.html

```

```
*
* This Java example performs these tasks:
*
* 1. Returns a list of the available DB engines.
* 2. Selects an engine family and create a custom DB parameter group.
* 3. Gets the parameter groups.
* 4. Gets parameters in the group.
* 5. Modifies the auto_increment_offset parameter.
* 6. Gets and displays the updated parameters.
* 7. Gets a list of allowed engine versions.
* 8. Gets a list of micro instance classes available for the selected engine.
* 9. Creates an RDS database instance that contains a MySQL database and uses
* the parameter group.
* 10. Waits for the DB instance to be ready and prints out the connection
* endpoint value.
* 11. Creates a snapshot of the DB instance.
* 12. Waits for an RDS DB snapshot to be ready.
* 13. Deletes the RDS DB instance.
* 14. Deletes the parameter group.
*/
public class RDSScenario {
    public static long sleepTime = 20;
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");

    public static void main(String[] args) throws InterruptedException {
        final String usage = ""

            Usage:
                <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
<dbName> <dbSnapshotIdentifier> <secretName>

            Where:
                dbGroupName - The database group name.\s
                dbParameterGroupFamily - The database parameter group name
(for example, mysql8.0).
                dbInstanceIdentifier - The database instance identifier\s
                dbName - The database name.\s
                dbSnapshotIdentifier - The snapshot identifier.\s
                secretName - The name of the AWS Secrets Manager secret that
contains the database credentials"
            """;

        if (args.length != 6) {
```



```
        System.out.println(usage);
        System.exit(1);
    }

    String dbGroupName = args[0];
    String dbParameterGroupFamily = args[1];
    String dbInstanceIdentifier = args[2];
    String dbName = args[3];
    String dbSnapshotIdentifier = args[4];
    String secretName = args[5];

    Gson gson = new Gson();
    User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
    String masterUsername = user.getUsername();
    String masterUserPassword = user.getPassword();

    Region region = Region.US_WEST_2;
    RdsClient rdsClient = RdsClient.builder()
        .region(region)
        .build();
    System.out.println(DASHES);
    System.out.println("Welcome to the Amazon RDS example scenario.");
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("1. Return a list of the available DB engines");
    describeDBEngines(rdsClient);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("2. Create a custom parameter group");
    createDBParameterGroup(rdsClient, dbGroupName, dbParameterGroupFamily);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("3. Get the parameter group");
    describeDbParameterGroups(rdsClient, dbGroupName);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("4. Get the parameters in the group");
    describeDbParameters(rdsClient, dbGroupName, 0);
    System.out.println(DASHES);
```

```
System.out.println(DASHES);
System.out.println("5. Modify the auto_increment_offset parameter");
modifyDBParas(rdsClient, dbGroupName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("6. Display the updated value");
describeDbParameters(rdsClient, dbGroupName, -1);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("7. Get a list of allowed engine versions");
getAllowedEngines(rdsClient, dbParameterGroupFamily);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("8. Get a list of micro instance classes available for
the selected engine");
getMicroInstances(rdsClient);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println(
    "9. Create an RDS database instance that contains a MySQL
database and uses the parameter group");
String dbARN = createDatabaseInstance(rdsClient, dbGroupName,
dbInstanceIdentifier, dbName, masterUsername,
    masterUserPassword);
System.out.println("The ARN of the new database is " + dbARN);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("10. Wait for DB instance to be ready");
waitForInstanceReady(rdsClient, dbInstanceIdentifier);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("11. Create a snapshot of the DB instance");
createSnapshot(rdsClient, dbInstanceIdentifier, dbSnapshotIdentifier);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("12. Wait for DB snapshot to be ready");
```

```
        waitForSnapshotReady(rdsClient, dbInstanceIdentifier,
dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Delete the DB instance");
        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Delete the parameter group");
        deleteParaGroup(rdsClient, dbGroupName, dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Scenario has successfully completed.");
        System.out.println(DASHES);

        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
            .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
            .build();
    }

    public static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
            .secretId(secretName)
            .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    // Delete the parameter group after database has been deleted.
    // An exception is thrown if you attempt to delete the para group while
database
```

```
// exists.
public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
    throws InterruptedException {
    try {
        boolean isDataDel = false;
        boolean didFind;
        String instanceARN;

        // Make sure that the database has been deleted.
        while (!isDataDel) {
            DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            int listSize = instanceList.size();
            didFind = false;
            int index = 1;
            for (DBInstance instance : instanceList) {
                instanceARN = instance.dbInstanceArn();
                if (instanceARN.compareTo(dbARN) == 0) {
                    System.out.println(dbARN + " still exists");
                    didFind = true;
                }
                if ((index == listSize) && (!didFind)) {
                    // Went through the entire list and did not find the
database ARN.

                    isDataDel = true;
                }
                Thread.sleep(sleepTime * 1000);
                index++;
            }
        }

        // Delete the para group.
        DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
            .dbParameterGroupName(dbGroupName)
            .build();

        rdsClient.deleteDBParameterGroup(parameterGroupRequest);
        System.out.println(dbGroupName + " was deleted.");

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
    }
}
```

```
        System.exit(1);
    }
}

// Delete the DB instance.
public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
    try {
        DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .deleteAutomatedBackups(true)
            .skipFinalSnapshot(true)
            .build();

        DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
        System.out.println("The status of the database is " +
response.dbInstance().dbInstanceStatus());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Waits until the snapshot instance is available.
public static void waitForSnapshotReady(RdsClient rdsClient, String
dbInstanceIdentifier,
    String dbSnapshotIdentifier) {
    try {
        boolean snapshotReady = false;
        String snapshotReadyStr;
        System.out.println("Waiting for the snapshot to become available.");

        DescribeDbSnapshotsRequest snapshotsRequest =
DescribeDbSnapshotsRequest.builder()
            .dbSnapshotIdentifier(dbSnapshotIdentifier)
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .build();

        while (!snapshotReady) {
            DescribeDbSnapshotsResponse response =
rdsClient.describeDBSnapshots(snapshotsRequest);
```

```
        List<DBSnapshot> snapshotList = response.dbSnapshots();
        for (DBSnapshot snapshot : snapshotList) {
            snapshotReadyStr = snapshot.status();
            if (snapshotReadyStr.contains("available")) {
                snapshotReady = true;
            } else {
                System.out.print(".");
                Thread.sleep(sleepTime * 1000);
            }
        }
    }

    System.out.println("The Snapshot is available!");
} catch (RdsException | InterruptedException e) {
    System.out.println(e.getLocalizedMessage());
    System.exit(1);
}
}

// Create an Amazon RDS snapshot.
public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
    try {
        CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
            .dbInstanceIdentifier(dbInstanceIdentifier)
            .dbSnapshotIdentifier(dbSnapshotIdentifier)
            .build();

        CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
        System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Waits until the database instance is available.
public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
    boolean instanceReady = false;
```

```
String instanceReadyStr;
System.out.println("Waiting for instance to become available.");
try {
    DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
        .dbInstanceIdentifier(dbInstanceIdentifier)
        .build();

    String endpoint = "";
    while (!instanceReady) {
        DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
        List<DBInstance> instanceList = response.dbInstances();
        for (DBInstance instance : instanceList) {
            instanceReadyStr = instance.dbInstanceStatus();
            if (instanceReadyStr.contains("available")) {
                endpoint = instance.endpoint().address();
                instanceReady = true;
            } else {
                System.out.print(".");
                Thread.sleep(sleepTime * 1000);
            }
        }
    }
    System.out.println("Database instance is available! The connection
endpoint is " + endpoint);

} catch (RdsException | InterruptedException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}

// Create a database instance and return the ARN of the database.
public static String createDatabaseInstance(RdsClient rdsClient,
    String dbGroupName,
    String dbInstanceIdentifier,
    String dbName,
    String masterUsername,
    String masterUserPassword) {

    try {
        CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
```

```
        .dbInstanceIdentifier(dbInstanceIdentifier)
        .allocatedStorage(100)
        .dbName(dbName)
        .dbParameterGroupName(dbGroupName)
        .engine("mysql")
        .dbInstanceClass("db.m4.large")
        .engineVersion("8.0")
        .storageType("standard")
        .masterUsername(masterUsername)
        .masterUserPassword(masterUserPassword)
        .build();

        CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
        System.out.println("The status is " +
response.dbInstance().dbInstanceStatus());
        return response.dbInstance().dbInstanceArn();

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }

    return "";
}

// Get a list of micro instances.
public static void getMicroInstances(RdsClient rdsClient) {
    try {
        DescribeOrderableDbInstanceOptionsRequest dbInstanceOptionsRequest =
DescribeOrderableDbInstanceOptionsRequest
            .builder()
            .engine("mysql")
            .build();

        DescribeOrderableDbInstanceOptionsResponse response = rdsClient

        .describeOrderableDBInstanceOptions(dbInstanceOptionsRequest);
        List<OrderableDBInstanceOption> orderableDBInstances =
response.orderableDBInstanceOptions();
        for (OrderableDBInstanceOption dbInstanceOption :
orderableDBInstances) {
            System.out.println("The engine version is " +
dbInstanceOption.engineVersion());
        }
    }
}
```



```
        System.out.println("The engine description is " +
dbInstanceOption.engine());
    }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Get a list of allowed engine versions.
public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
    try {
        DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
            .dbParameterGroupFamily(dbParameterGroupFamily)
            .engine("mysql")
            .build();

        DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
        List<DBEngineVersion> dbEngines = response.dbEngineVersions();
        for (DBEngineVersion dbEngine : dbEngines) {
            System.out.println("The engine version is " +
dbEngine.engineVersion());
            System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
        }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Modify auto_increment_offset and auto_increment_increment parameters.
public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
    try {
        Parameter parameter1 = Parameter.builder()
            .parameterName("auto_increment_offset")
            .applyMethod("immediate")
            .parameterValue("5")
            .build();
```

```
        List<Parameter> paraList = new ArrayList<>();
        paraList.add(parameter1);
        ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
            .dbParameterGroupName(dbGroupName)
            .parameters(paraList)
            .build();

        ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
        System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

// Retrieve parameters in the group.
public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
    try {
        DescribeDbParametersRequest dbParameterGroupsRequest;
        if (flag == 0) {
            dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                .dbParameterGroupName(dbGroupName)
                .build();
        } else {
            dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                .dbParameterGroupName(dbGroupName)
                .source("user")
                .build();
        }

        DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
        List<Parameter> dbParameters = response.parameters();
        String paraName;
        for (Parameter para : dbParameters) {
            // Only print out information about either auto_increment_offset
or
            // auto_increment_increment.
```

```
        paraName = para.parameterName();
        if ((paraName.compareTo("auto_increment_offset") == 0)
            || (paraName.compareTo("auto_increment_increment ") ==
0)) {
            System.out.println("*** The parameter name is " + paraName);
            System.out.println("*** The parameter value is " +
para.parameterValue());
            System.out.println("*** The parameter data type is " +
para.dataType());
            System.out.println("*** The parameter description is " +
para.description());
            System.out.println("*** The parameter allowed values is " +
para.allowedValues());
        }
    }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}

public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
    try {
        DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
            .dbParameterGroupName(dbGroupName)
            .maxRecords(20)
            .build();

        DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
        List<DBParameterGroup> groups = response.dbParameterGroups();
        for (DBParameterGroup group : groups) {
            System.out.println("The group name is " +
group.dbParameterGroupName());
            System.out.println("The group description is " +
group.description());
        }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
```

```
    }  
  }  
  
  public static void createDBParameterGroup(RdsClient rdsClient, String  
dbGroupName, String dbParameterGroupFamily) {  
    try {  
      CreateDbParameterGroupRequest groupRequest =  
CreateDbParameterGroupRequest.builder()  
        .dbParameterGroupName(dbGroupName)  
        .dbParameterGroupFamily(dbParameterGroupFamily)  
        .description("Created by using the AWS SDK for Java")  
        .build();  
  
      CreateDbParameterGroupResponse response =  
rdsClient.createDBParameterGroup(groupRequest);  
      System.out.println("The group name is " +  
response.dbParameterGroup().dbParameterGroupName());  
  
    } catch (RdsException e) {  
      System.out.println(e.getLocalizedMessage());  
      System.exit(1);  
    }  
  }  
  
  public static void describeDBEngines(RdsClient rdsClient) {  
    try {  
      DescribeDbEngineVersionsRequest engineVersionsRequest =  
DescribeDbEngineVersionsRequest.builder()  
        .defaultOnly(true)  
        .engine("mysql")  
        .maxRecords(20)  
        .build();  
  
      DescribeDbEngineVersionsResponse response =  
rdsClient.describeDBEngineVersions(engineVersionsRequest);  
      List<DBEngineVersion> engines = response.dbEngineVersions();  
  
      // Get all DBEngineVersion objects.  
      for (DBEngineVersion engineOb : engines) {  
        System.out.println("The name of the DB parameter group family for  
the database engine is "  
          + engineOb.dbParameterGroupFamily());  
        System.out.println("The name of the database engine " +  
engineOb.engine());  
      }  
    }  
  }  
}
```

```
        System.out.println("The version number of the database engine " +
engine0b.engineVersion());
    }

    } catch (RdsException e) {
        System.out.println(e.getLocalizedMessage());
        System.exit(1);
    }
}
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的下列主題。
 - [CreateDBInstance](#)
 - [創建數據庫 ParameterGroup](#)
 - [CreateDBSnapshot](#)
 - [DeleteDBInstance](#)
 - [刪除資料庫 ParameterGroup](#)
 - [描述 B EngineVersions](#)
 - [DescribeDBInstances](#)
 - [描述 B ParameterGroups](#)
 - [DescribeDBParameters](#)
 - [DescribeDBSnapshots](#)
 - [DescribeOrderable資料庫 InstanceOptions](#)
 - [修改資料庫 ParameterGroup](#)

Kotlin

適用於 Kotlin 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/**  
Before running this code example, set up your development environment, including  
your credentials.
```

For more information, see the following documentation topic:

<https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html>

This example requires an AWS Secrets Manager secret that contains the database credentials. If you do not create a secret, this example will not work. For more details, see:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-services-use-secrets_RS.html

This example performs the following tasks:

1. Returns a list of the available DB engines by invoking the `DescribeDbEngineVersions` method.
2. Selects an engine family and create a custom DB parameter group by invoking the `createDBParameterGroup` method.
3. Gets the parameter groups by invoking the `DescribeDbParameterGroups` method.
4. Gets parameters in the group by invoking the `DescribeDbParameters` method.
5. Modifies both the `auto_increment_offset` and `auto_increment_increment` parameters by invoking the `modifyDbParameterGroup` method.
6. Gets and displays the updated parameters.
7. Gets a list of allowed engine versions by invoking the `describeDbEngineVersions` method.
8. Gets a list of micro instance classes available for the selected engine.
9. Creates an Amazon Relational Database Service (Amazon RDS) database instance that contains a MySQL database and uses the parameter group.
10. Waits for DB instance to be ready and prints out the connection endpoint value.
11. Creates a snapshot of the DB instance.
12. Waits for the DB snapshot to be ready.
13. Deletes the DB instance.
14. Deletes the parameter group.

```
*/
```

```
var sleepTime: Long = 20
```

```
suspend fun main(args: Array<String>) {
```

```
val usage = ""
Usage:
    <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
<dbName> <dbSnapshotIdentifier><secretName>

Where:
    dbGroupName - The database group name.
    dbParameterGroupFamily - The database parameter group name.
    dbInstanceIdentifier - The database instance identifier.
    dbName - The database name.
    dbSnapshotIdentifier - The snapshot identifier.
    secretName - The name of the AWS Secrets Manager secret that contains
the database credentials.
""

if (args.size != 6) {
    println(usage)
    exitProcess(1)
}

val dbGroupName = args[0]
val dbParameterGroupFamily = args[1]
val dbInstanceIdentifier = args[2]
val dbName = args[3]
val dbSnapshotIdentifier = args[4]
val secretName = args[5]

val gson = Gson()
val user = gson.fromJson(getSecretValues(secretName).toString(),
User::class.java)
val username = user.username
val userPassword = user.password

println("1. Return a list of the available DB engines")
describeDBEngines()

println("2. Create a custom parameter group")
createDBParameterGroup(dbGroupName, dbParameterGroupFamily)

println("3. Get the parameter groups")
describeDbParameterGroups(dbGroupName)

println("4. Get the parameters in the group")
describeDbParameters(dbGroupName, 0)
```

```
println("5. Modify the auto_increment_offset parameter")
modifyDBParas(dbGroupName)

println("6. Display the updated value")
describeDbParameters(dbGroupName, -1)

println("7. Get a list of allowed engine versions")
getAllowedEngines(dbParameterGroupFamily)

println("8. Get a list of micro instance classes available for the selected
engine")
getMicroInstances()

println("9. Create an RDS database instance that contains a MySQL database
and uses the parameter group")
val dbARN = createDatabaseInstance(dbGroupName, dbInstanceIdentifier, dbName,
username, userPassword)
println("The ARN of the new database is $dbARN")

println("10. Wait for DB instance to be ready")
waitForDbInstanceReady(dbInstanceIdentifier)

println("11. Create a snapshot of the DB instance")
createDbSnapshot(dbInstanceIdentifier, dbSnapshotIdentifier)

println("12. Wait for DB snapshot to be ready")
waitForSnapshotReady(dbInstanceIdentifier, dbSnapshotIdentifier)

println("13. Delete the DB instance")
deleteDbInstance(dbInstanceIdentifier)

println("14. Delete the parameter group")
if (dbARN != null) {
    deleteParaGroup(dbGroupName, dbARN)
}

println("The Scenario has successfully completed.")
}

suspend fun deleteParaGroup(
    dbGroupName: String,
    dbARN: String,
) {
```



```
var isDataDel = false
var didFind: Boolean
var instanceARN: String

RdsClient { region = "us-west-2" }.use { rdsClient ->
    // Make sure that the database has been deleted.
    while (!isDataDel) {
        val response = rdsClient.describeDbInstances()
        val instanceList = response.dbInstances
        val listSize = instanceList?.size
        isDataDel = false // Reset this value.
        didFind = false // Reset this value.
        var index = 1
        if (instanceList != null) {
            for (instance in instanceList) {
                instanceARN = instance.dbInstanceArn.toString()
                if (instanceARN.compareTo(dbARN) == 0) {
                    println("$dbARN still exists")
                    didFind = true
                }
                if (index == listSize && !didFind) {
                    // Went through the entire list and did not find the
database name.
                    isDataDel = true
                }
                index++
            }
        }
    }

    // Delete the para group.
    val parameterGroupRequest =
        DeleteDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
        }
    rdsClient.deleteDbParameterGroup(parameterGroupRequest)
    println("$dbGroupName was deleted.")
}

suspend fun deleteDbInstance(dbInstanceIdentifierVal: String) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }
}
```

```
        deleteAutomatedBackups = true
        skipFinalSnapshot = true
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
    ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the snapshot instance is available.
suspend fun waitForSnapshotReady(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    var snapshotReady = false
    var snapshotReadyStr: String
    println("Waiting for the snapshot to become available.")

    val snapshotsRequest =
        DescribeDbSnapshotsRequest {
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    while (!snapshotReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbSnapshots(snapshotsRequest)
            val snapshotList: List<DbSnapshot>? = response.dbSnapshots
            if (snapshotList != null) {
                for (snapshot in snapshotList) {
                    snapshotReadyStr = snapshot.status.toString()
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("The Snapshot is available!")
}
```

```
}

// Create an Amazon RDS snapshot.
suspend fun createDbSnapshot(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    val snapshotRequest =
        CreateDbSnapshotRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbSnapshot(snapshotRequest)
        print("The Snapshot id is ${response.dbSnapshot?.dbiResourceId}")
    }
}

// Waits until the database instance is available.
suspend fun waitForDbInstanceReady(dbInstanceIdentifierVal: String?) {
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }
    var endpoint = ""
    while (!instanceReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint?.address.toString()
                        instanceReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}

println("Database instance is available! The connection endpoint is
$endpoint")
}

// Create a database instance and return the ARN of the database.
suspend fun createDatabaseInstance(
    dbGroupNameVal: String?,
    dbInstanceIdentifierVal: String?,
    dbNameVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
): String? {
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNameVal
            dbParameterGroupName = dbGroupNameVal
            engine = "mysql"
            dbInstanceClass = "db.m4.large"
            engineVersion = "8.0"
            storageType = "standard"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
        return response.dbInstance?.dbInstanceArn
    }
}

// Get a list of micro instances.
suspend fun getMicroInstances() {
    val dbInstanceOptionsRequest =
        DescribeOrderableDbInstanceOptionsRequest {
            engine = "mysql"
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
```

```
        val response =
rdsClient.describeOrderableDbInstanceOptions(dbInstanceOptionsRequest)
        val orderableDBInstances = response.orderableDbInstanceOptions
        if (orderableDBInstances != null) {
            for (dbInstanceOption in orderableDBInstances) {
                println("The engine version is
${dbInstanceOption.engineVersion}")
                println("The engine description is ${dbInstanceOption.engine}")
            }
        }
    }
}

// Get a list of allowed engine versions.
suspend fun getAllowedEngines(dbParameterGroupFamilyVal: String?) {
    val versionsRequest =
        DescribeDbEngineVersionsRequest {
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            engine = "mysql"
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(versionsRequest)
        val dbEngines: List<DbEngineVersion>? = response.dbEngineVersions
        if (dbEngines != null) {
            for (dbEngine in dbEngines) {
                println("The engine version is ${dbEngine.engineVersion}")
                println("The engine description is
${dbEngine.dbEngineDescription}")
            }
        }
    }
}

// Modify the auto_increment_offset parameter.
suspend fun modifyDBParas(dbGroupName: String) {
    val parameter1 =
        Parameter {
            parameterName = "auto_increment_offset"
            applyMethod = ApplyMethod.Immediate
            parameterValue = "5"
        }

    val paraList: ArrayList<Parameter> = ArrayList()
    paraList.add(parameter1)
```

```
val groupRequest =
    ModifyDbParameterGroupRequest {
        dbParameterGroupName = dbGroupName
        parameters = paraList
    }

RdsClient { region = "us-west-2" }.use { rdsClient ->
    val response = rdsClient.modifyDbParameterGroup(groupRequest)
    println("The parameter group ${response.dbParameterGroupName} was
successfully modified")
}
}

// Retrieve parameters in the group.
suspend fun describeDbParameters(
    dbGroupName: String?,
    flag: Int,
) {
    val dbParameterGroupsRequest: DescribeDbParametersRequest
    dbParameterGroupsRequest =
        if (flag == 0) {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
            }
        } else {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
                source = "user"
            }
        }
}

RdsClient { region = "us-west-2" }.use { rdsClient ->
    val response = rdsClient.describeDbParameters(dbParameterGroupsRequest)
    val dbParameters: List<Parameter>? = response.parameters
    var paraName: String
    if (dbParameters != null) {
        for (para in dbParameters) {
            // Only print out information about either auto_increment_offset
            or auto_increment_increment.
            paraName = para.parameterName.toString()
            if (paraName.compareTo("auto_increment_offset") == 0 ||
paraName.compareTo("auto_increment_increment ") == 0) {
                println("*** The parameter name is $paraName")
                System.out.println("*** The parameter value is
${para.parameterValue}")
            }
        }
    }
}
```

```
        System.out.println("*** The parameter data type is
${para.dataType}")
        System.out.println("*** The parameter description is
${para.description}")
        System.out.println("*** The parameter allowed values is
${para.allowedValues}")
    }
}
}
}

suspend fun describeDbParameterGroups(dbGroupName: String?) {
    val groupsRequest =
        DescribeDbParameterGroupsRequest {
            dbParameterGroupName = dbGroupName
            maxRecords = 20
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameterGroups(groupsRequest)
        val groups = response.dbParameterGroups
        if (groups != null) {
            for (group in groups) {
                println("The group name is ${group.dbParameterGroupName}")
                println("The group description is ${group.description}")
            }
        }
    }
}

// Create a parameter group.
suspend fun createDBParameterGroup(
    dbGroupName: String?,
    dbParameterGroupFamilyVal: String?,
) {
    val groupRequest =
        CreateDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            description = "Created by using the AWS SDK for Kotlin"
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbParameterGroup(groupRequest)
    }
}
```

```
        println("The group name is
    ${response.dbParameterGroup?.dbParameterGroupName}")
    }
}

// Returns a list of the available DB engines.
suspend fun describeDBEngines() {
    val engineVersionsRequest =
        DescribeDbEngineVersionsRequest {
            defaultOnly = true
            engine = "mysql"
            maxRecords = 20
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(engineVersionsRequest)
        val engines: List<DbEngineVersion>? = response.dbEngineVersions

        // Get all DbEngineVersion objects.
        if (engines != null) {
            for (engineOb in engines) {
                println("The name of the DB parameter group family for the
            database engine is ${engineOb.dbParameterGroupFamily}.")
                println("The name of the database engine ${engineOb.engine}.")
                println("The version number of the database engine
            ${engineOb.engineVersion}")
            }
        }
    }
}

suspend fun getSecretValues(secretName: String?): String? {
    val valueRequest =
        GetSecretValueRequest {
            secretId = secretName
        }

    SecretsManagerClient { region = "us-west-2" }.use { secretsClient ->
        val valueResponse = secretsClient.getSecretValue(valueRequest)
        return valueResponse.secretString
    }
}
```


- 如需 API 詳細資訊，請參閱《AWS 適用於 Kotlin 的 SDK API 參考》中的下列主題。
 - [CreateDBInstance](#)
 - [創建數據庫 ParameterGroup](#)
 - [CreateDBSnapshot](#)
 - [DeleteDBInstance](#)
 - [刪除資料庫 ParameterGroup](#)
 - [描述 B EngineVersions](#)
 - [DescribeDBInstances](#)
 - [描述 B ParameterGroups](#)
 - [DescribeDBParameters](#)
 - [DescribeDBSnapshots](#)
 - [DescribeOrderable資料庫 InstanceOptions](#)
 - [修改資料庫 ParameterGroup](#)

Python

適用於 Python (Boto3) 的 SDK

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

在命令提示中執行互動式案例。

```
class RdsInstanceScenario:
    """Runs a scenario that shows how to get started using Amazon RDS DB
    instances."""

    def __init__(self, instance_wrapper):
        """
        :param instance_wrapper: An object that wraps Amazon RDS DB instance
        actions.
        """
        self.instance_wrapper = instance_wrapper
```

```
def create_parameter_group(self, parameter_group_name, db_engine):
    """
    Shows how to get available engine versions for a specified database
    engine and
    create a DB parameter group that is compatible with a selected engine
    family.

    :param parameter_group_name: The name given to the newly created
    parameter group.
    :param db_engine: The database engine to use as a basis.
    :return: The newly created parameter group.
    """
    print(
        f"Checking for an existing DB instance parameter group named
    {parameter_group_name}."
    )
    parameter_group = self.instance_wrapper.get_parameter_group(
        parameter_group_name
    )
    if parameter_group is None:
        print(f"Getting available database engine versions for {db_engine}.")
        engine_versions =
self.instance_wrapper.get_engine_versions(db_engine)
        families = list({ver["DBParameterGroupFamily"] for ver in
engine_versions})
        family_index = q.choose("Which family do you want to use? ",
families)
        print(f"Creating a parameter group.")
        self.instance_wrapper.create_parameter_group(
            parameter_group_name, families[family_index], "Example parameter
group."
        )
        parameter_group = self.instance_wrapper.get_parameter_group(
            parameter_group_name
        )
        print(f"Parameter group {parameter_group['DBParameterGroupName']}:")
        pp(parameter_group)
        print("-" * 88)
        return parameter_group

def update_parameters(self, parameter_group_name):
    """
    Shows how to get the parameters contained in a custom parameter group and
    update some of the parameter values in the group.
```

```

        :param parameter_group_name: The name of the parameter group to query and
modify.
        """
        print("Let's set some parameter values in your parameter group.")
        auto_inc_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, name_prefix="auto_increment"
        )
        update_params = []
        for auto_inc in auto_inc_parameters:
            if auto_inc["IsModifiable"] and auto_inc["DataType"] == "integer":
                print(f"The {auto_inc['ParameterName']} parameter is described
as:")

                print(f"\t{auto_inc['Description']}")
                param_range = auto_inc["AllowedValues"].split("-")
                auto_inc["ParameterValue"] = str(
                    q.ask(
                        f"Enter a value between {param_range[0]} and
{param_range[1]}: ",
                        q.is_int,
                        q.in_range(int(param_range[0]), int(param_range[1])),
                    )
                )
                update_params.append(auto_inc)
        self.instance_wrapper.update_parameters(parameter_group_name,
update_params)
        print(
            "You can get a list of parameters you've set by specifying a source
of 'user'."
        )
        user_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, source="user"
        )
        pp(user_parameters)
        print("-" * 88)

    def create_instance(self, instance_name, db_name, db_engine,
parameter_group):
        """
        Shows how to create a DB instance that contains a database of a specified
type and is configured to use a custom DB parameter group.

        :param instance_name: The name given to the newly created DB instance.
        :param db_name: The name given to the created database.

```

```
    :param db_engine: The engine of the created database.
    :param parameter_group: The parameter group that is associated with the
DB instance.
    :return: The newly created DB instance.
    """
    print("Checking for an existing DB instance.")
    db_inst = self.instance_wrapper.get_db_instance(instance_name)
    if db_inst is None:
        print("Let's create a DB instance.")
        admin_username = q.ask(
            "Enter an administrator user name for the database: ",
q.non_empty
        )
        admin_password = q.ask(
            "Enter a password for the administrator (at least 8 characters):
",
            q.non_empty,
        )
        engine_versions = self.instance_wrapper.get_engine_versions(
            db_engine, parameter_group["DBParameterGroupFamily"]
        )
        engine_choices = [ver["EngineVersion"] for ver in engine_versions]
        print("The available engines for your parameter group are:")
        engine_index = q.choose("Which engine do you want to use? ",
engine_choices)
        engine_selection = engine_versions[engine_index]
        print(
            "The available micro DB instance classes for your database engine
are:"
        )
        inst_opts = self.instance_wrapper.get_orderable_instances(
            engine_selection["Engine"], engine_selection["EngineVersion"]
        )
        inst_choices = list(
            {
                opt["DBInstanceClass"]
                for opt in inst_opts
                if "micro" in opt["DBInstanceClass"]
            }
        )
        inst_index = q.choose(
            "Which micro DB instance class do you want to use? ",
inst_choices
        )
```

```
        group_name = parameter_group["DBParameterGroupName"]
        storage_type = "standard"
        allocated_storage = 5
        print(
            f"Creating a DB instance named {instance_name} and database
{db_name}.\n"
            f"The DB instance is configured to use your custom parameter
group {group_name},\n"
            f"selected engine {engine_selection['EngineVersion']},\n"
            f"selected DB instance class {inst_choices[inst_index]},\n"
            f"and {allocated_storage} GiB of {storage_type} storage.\n"
            f"This typically takes several minutes."
        )
        db_inst = self.instance_wrapper.create_db_instance(
            db_name,
            instance_name,
            group_name,
            engine_selection["Engine"],
            engine_selection["EngineVersion"],
            inst_choices[inst_index],
            storage_type,
            allocated_storage,
            admin_username,
            admin_password,
        )
        while db_inst.get("DBInstanceStatus") != "available":
            wait(10)
            db_inst = self.instance_wrapper.get_db_instance(instance_name)
        print("Instance data:")
        pp(db_inst)
        print("-" * 88)
        return db_inst

    @staticmethod
    def display_connection(db_inst):
        """
        Displays connection information about a DB instance and tips on how to
        connect to it.

        :param db_inst: The DB instance to display.
        """
        print(
            "You can now connect to your database using your favorite MySQL
client.\n"
```

```

        "One way to connect is by using the 'mysql' shell on an Amazon EC2
instance\n"
        "that is running in the same VPC as your DB instance. Pass the
endpoint,\n"
        "port, and administrator user name to 'mysql' and enter your password
\n"
        "when prompted:\n"
    )
    print(
        f"\n\tmysql -h {db_inst['Endpoint']['Address']} -P
{db_inst['Endpoint']['Port']} "
        f"-u {db_inst['MasterUsername']} -p\n"
    )
    print(
        "For more information, see the User Guide for Amazon RDS:\n"
        "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL"
    )
    print("-" * 88)

def create_snapshot(self, instance_name):
    """
    Shows how to create a DB instance snapshot and wait until it's available.

    :param instance_name: The name of a DB instance to snapshot.
    """
    if q.ask(
        "Do you want to create a snapshot of your DB instance (y/n)? ",
        q.is_yesno
    ):
        snapshot_id = f"{instance_name}-{uuid.uuid4()}"
        print(
            f"Creating a snapshot named {snapshot_id}. This typically takes a
few minutes."
        )
        snapshot = self.instance_wrapper.create_snapshot(snapshot_id,
instance_name)
        while snapshot.get("Status") != "available":
            wait(10)
            snapshot = self.instance_wrapper.get_snapshot(snapshot_id)
        pp(snapshot)
        print("-" * 88)

def cleanup(self, db_inst, parameter_group_name):

```

```
""
Shows how to clean up a DB instance and parameter group.
Before the parameter group can be deleted, all associated DB instances
must first
be deleted.

:param db_inst: The DB instance to delete.
:param parameter_group_name: The DB parameter group to delete.
""
if q.ask(
    "\nDo you want to delete the DB instance and parameter group (y/n)?
",
    q.is_yesno,
):
    print(f"Deleting DB instance {db_inst['DBInstanceIdentifier']}")

self.instance_wrapper.delete_db_instance(db_inst["DBInstanceIdentifier"])
    print(
        "Waiting for the DB instance to delete. This typically takes
several minutes."
    )
    while db_inst is not None:
        wait(10)
        db_inst = self.instance_wrapper.get_db_instance(
            db_inst["DBInstanceIdentifier"]
        )
    print(f"Deleting parameter group {parameter_group_name}.")
    self.instance_wrapper.delete_parameter_group(parameter_group_name)

def run_scenario(self, db_engine, parameter_group_name, instance_name,
db_name):
    logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

    print("-" * 88)
    print(
        "Welcome to the Amazon Relational Database Service (Amazon RDS)\n"
        "get started with DB instances demo."
    )
    print("-" * 88)

    parameter_group = self.create_parameter_group(parameter_group_name,
db_engine)
    self.update_parameters(parameter_group_name)
```

```
        db_inst = self.create_instance(
            instance_name, db_name, db_engine, parameter_group
        )
        self.display_connection(db_inst)
        self.create_snapshot(instance_name)
        self.cleanup(db_inst, parameter_group_name)

        print("\nThanks for watching!")
        print("-" * 88)

if __name__ == "__main__":
    try:
        scenario = RdsInstanceScenario(InstanceWrapper.from_client())
        scenario.run_scenario(
            "mysql",
            "doc-example-parameter-group",
            "doc-example-instance",
            "docexampledb",
        )
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

定義案例所呼叫的函數以管理 Amazon RDS 動作。

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)
```



```
def get_parameter_group(self, parameter_group_name):
    """
    Gets a DB parameter group.

    :param parameter_group_name: The name of the parameter group to retrieve.
    :return: The parameter group.
    """
    try:
        response = self.rds_client.describe_db_parameter_groups(
            DBParameterGroupName=parameter_group_name
        )
        parameter_group = response["DBParameterGroups"][0]
    except ClientError as err:
        if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
            logger.info("Parameter group %s does not exist.",
parameter_group_name)
        else:
            logger.error(
                "Couldn't get parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
    else:
        return parameter_group

def create_parameter_group(
    self, parameter_group_name, parameter_group_family, description
):
    """
    Creates a DB parameter group that is based on the specified parameter
group
family.

    :param parameter_group_name: The name of the newly created parameter
group.
    :param parameter_group_family: The family that is used as the basis of
the new
parameter group.
    :param description: A description given to the parameter group.
    :return: Data about the newly created parameter group.
```

```
    """
    try:
        response = self.rds_client.create_db_parameter_group(
            DBParameterGroupName=parameter_group_name,
            DBParameterGroupFamily=parameter_group_family,
            Description=description,
        )
    except ClientError as err:
        logger.error(
            "Couldn't create parameter group %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response

def delete_parameter_group(self, parameter_group_name):
    """
    Deletes a DB parameter group.

    :param parameter_group_name: The name of the parameter group to delete.
    :return: Data about the parameter group.
    """
    try:
        self.rds_client.delete_db_parameter_group(
            DBParameterGroupName=parameter_group_name
        )
    except ClientError as err:
        logger.error(
            "Couldn't delete parameter group %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def get_parameters(self, parameter_group_name, name_prefix="", source=None):
    """
    Gets the parameters that are contained in a DB parameter group.
```

```

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                                to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                                For example, a source of 'user' retrieves only parameters
that
                                were set by a user.
:return: The list of requested parameters.
"""
try:
    kwargs = {"DBParameterGroupName": parameter_group_name}
    if source is not None:
        kwargs["Source"] = source
    parameters = []
    paginator = self.rds_client.get_paginator("describe_db_parameters")
    for page in paginator.paginate(**kwargs):
        parameters += [
            p
            for p in page["Parameters"]
            if p["ParameterName"].startswith(name_prefix)
        ]
except ClientError as err:
    logger.error(
        "Couldn't get parameters for %s. Here's why: %s: %s",
        parameter_group_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return parameters

def update_parameters(self, parameter_group_name, update_parameters):
    """
    Updates parameters in a custom DB parameter group.

    :param parameter_group_name: The name of the parameter group to update.
    :param update_parameters: The parameters to update in the group.
    :return: Data about the modified parameter group.
    """

```

```
    try:
        response = self.rds_client.modify_db_parameter_group(
            DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
        )
    except ClientError as err:
        logger.error(
            "Couldn't update parameters in %s. Here's why: %s: %s",
            parameter_group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response

def create_snapshot(self, snapshot_id, instance_id):
    """
    Creates a snapshot of a DB instance.

    :param snapshot_id: The ID to give the created snapshot.
    :param instance_id: The ID of the DB instance to snapshot.
    :return: Data about the newly created snapshot.
    """
    try:
        response = self.rds_client.create_db_snapshot(
            DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
        )
        snapshot = response["DBSnapshot"]
    except ClientError as err:
        logger.error(
            "Couldn't create snapshot of %s. Here's why: %s: %s",
            instance_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return snapshot

def get_snapshot(self, snapshot_id):
```

```
"""
Gets a DB instance snapshot.

:param snapshot_id: The ID of the snapshot to retrieve.
:return: The retrieved snapshot.
"""
try:
    response = self.rds_client.describe_db_snapshots(
        DBSnapshotIdentifier=snapshot_id
    )
    snapshot = response["DBSnapshots"][0]
except ClientError as err:
    logger.error(
        "Couldn't get snapshot %s. Here's why: %s: %s",
        snapshot_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return snapshot

def get_engine_versions(self, engine, parameter_group_family=None):
    """
    Gets database engine versions that are available for the specified engine
    and parameter group family.

    :param engine: The database engine to look up.
    :param parameter_group_family: When specified, restricts the returned
list of
                                engine versions to those that are
compatible with
                                this parameter group family.

    :return: The list of database engine versions.
    """
    try:
        kwargs = {"Engine": engine}
        if parameter_group_family is not None:
            kwargs["DBParameterGroupFamily"] = parameter_group_family
        response = self.rds_client.describe_db_engine_versions(**kwargs)
        versions = response["DBEngineVersions"]
    except ClientError as err:
        logger.error(
```

```
        "Couldn't get engine versions for %s. Here's why: %s: %s",
        engine,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return versions

def get_orderable_instances(self, db_engine, db_engine_version):
    """
    Gets DB instance options that can be used to create DB instances that are
    compatible with a set of specifications.

    :param db_engine: The database engine that must be supported by the DB
    instance.
    :param db_engine_version: The engine version that must be supported by
    the DB instance.
    :return: The list of DB instance options that can be used to create a
    compatible DB instance.
    """
    try:
        inst_opts = []
        paginator = self.rds_client.get_paginator(
            "describe_orderable_db_instance_options"
        )
        for page in paginator.paginate(
            Engine=db_engine, EngineVersion=db_engine_version
        ):
            inst_opts += page["OrderableDBInstanceOptions"]
    except ClientError as err:
        logger.error(
            "Couldn't get orderable DB instances. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return inst_opts

def get_db_instance(self, instance_id):
    """
```

```
Gets data about a DB instance.

:param instance_id: The ID of the DB instance to retrieve.
:return: The retrieved DB instance.
"""
try:
    response = self.rds_client.describe_db_instances(
        DBInstanceIdentifier=instance_id
    )
    db_inst = response["DBInstances"][0]
except ClientError as err:
    if err.response["Error"]["Code"] == "DBInstanceNotFound":
        logger.info("Instance %s does not exist.", instance_id)
    else:
        logger.error(
            "Couldn't get DB instance %s. Here's why: %s: %s",
            instance_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
else:
    return db_inst

def create_db_instance(
    self,
    db_name,
    instance_id,
    parameter_group_name,
    db_engine,
    db_engine_version,
    instance_class,
    storage_type,
    allocated_storage,
    admin_name,
    admin_password,
):
    """
    Creates a DB instance.

    :param db_name: The name of the database that is created in the DB
    instance.
    :param instance_id: The ID to give the newly created DB instance.
```

```
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
    try:
        response = self.rds_client.create_db_instance(
            DBName=db_name,
            DBInstanceIdentifier=instance_id,
            DBParameterGroupName=parameter_group_name,
            Engine=db_engine,
            EngineVersion=db_engine_version,
            DBInstanceClass=instance_class,
            StorageType=storage_type,
            AllocatedStorage=allocated_storage,
            MasterUsername=admin_name,
            MasterUserPassword=admin_password,
        )
        db_inst = response["DBInstance"]
    except ClientError as err:
        logger.error(
            "Couldn't create DB instance %s. Here's why: %s: %s",
            instance_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return db_inst

    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.
```



```
:param instance_id: The ID of the DB instance to delete.
:return: Data about the deleted DB instance.
"""
try:
    response = self.rds_client.delete_db_instance(
        DBInstanceIdentifier=instance_id,
        SkipFinalSnapshot=True,
        DeleteAutomatedBackups=True,
    )
    db_inst = response["DBInstance"]
except ClientError as err:
    logger.error(
        "Couldn't delete DB instance %s. Here's why: %s: %s",
        instance_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return db_inst
```

- 如需 API 的詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的下列主題。
 - [CreateDBInstance](#)
 - [創建數據庫 ParameterGroup](#)
 - [CreateDBSnapshot](#)
 - [DeleteDBInstance](#)
 - [刪除資料庫 ParameterGroup](#)
 - [描述 B EngineVersions](#)
 - [DescribeDBInstances](#)
 - [描述 B ParameterGroups](#)
 - [DescribeDBParameters](#)
 - [DescribeDBSnapshots](#)
 - [DescribeOrderable資料庫 InstanceOptions](#)

- [修改資料庫 ParameterGroup](#)

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使 AWS 用開發套件的 Amazon RDS 無伺服器範例

下列程式碼範例說明如何搭配 AWS 開發套件使用 Amazon RDS。

範例

- [在 Lambda 函數中連接到 Amazon RDS 數據庫](#)

在 Lambda 函數中連接到 Amazon RDS 數據庫

下列程式碼範例會示範如何實作連線至 RDS 資料庫的 Lambda 函數。該函數使一個簡單的數據庫請求，並返回結果。

Go

SDK for Go V2

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Go 連線至 Lambda 函數中的 Amazon RDS 資料庫。

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
Golang v2 code here.
*/

package main

import (
    "context"
```

```
"database/sql"
"encoding/json"
"fmt"

"github.com/aws/aws-lambda-go/lambda"
"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/feature/rds/auth"
_ "github.com/go-sql-driver/mysql"
)

type MyEvent struct {
    Name string `json:"name"`
}

func HandleRequest(event *MyEvent) (map[string]interface{}, error) {

    var dbName string = "DatabaseName"
    var dbUser string = "DatabaseUser"
    var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    var dbPort int = 3306
    var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
    var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    defer db.Close()
}
```

```
var sum int
err = db.QueryRow("SELECT ?+? AS sum", 3, 2).Scan(&sum)
if err != nil {
    panic(err)
}
s := fmt.Sprintf(sum)
message := fmt.Sprintf("The selected sum is: %s", s)

messageBytes, err := json.Marshal(message)
if err != nil {
    return nil, err
}

messageString := string(messageBytes)
return map[string]interface{}{
    "statusCode": 200,
    "headers":    map[string]string{"Content-Type": "application/json"},
    "body":      messageString,
}, nil
}

func main() {
    lambda.Start(HandleRequest)
}
```

JavaScript

適用於 JavaScript (v2) 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 JavaScript 連接到一個 Lambda 函數中的 Amazon RDS 數據庫。

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
```

```
Node.js code here.
*/
// ES6+ example
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

async function createAuthToken() {
  // Define connection authentication parameters
  const dbinfo = {

    hostname: process.env.ProxyHostName,
    port: process.env.Port,
    username: process.env.DBUserName,
    region: process.env.AWS_REGION,

  }

  // Create RDS Signer object
  const signer = new Signer(dbinfo);

  // Request authorization token from RDS, specifying the username
  const token = await signer.getAuthToken();
  return token;
}

async function dbOps() {

  // Obtain auth token
  const token = await createAuthToken();
  // Define connection configuration
  let connectionConfig = {
    host: process.env.ProxyHostName,
    user: process.env.DBUserName,
    password: token,
    database: process.env.DBName,
    ssl: 'Amazon RDS'
  }
  // Create the connection to the DB
  const conn = await mysql.createConnection(connectionConfig);
  // Obtain the result of the query
  const [res,] = await conn.execute('select ?+? as sum', [3, 2]);
  return res;
}
```

```
export const handler = async (event) => {
  // Execute database flow
  const result = await dbOps();
  // Return result
  return {
    statusCode: 200,
    body: JSON.stringify("The selected sum is: " + result[0].sum)
  }
};
```

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使 AWS 用開發套件的 Amazon RDS 跨服務範例

下列範例應用程式使用 AWS 開發套件將 Amazon RDS 與其他 AWS 服務應用程式結合使用。每個範例都包含一個連結 GitHub，您可以在其中找到如何設定和執行應用程式的指示。

範例

- [建立 Aurora 無伺服器工作項目追蹤器](#)

建立 Aurora 無伺服器工作項目追蹤器

說明如何建立 Web 應用程式追蹤 Amazon Aurora Serverless 資料庫中的工作項目，並且使用 Amazon Simple Email Service (Amazon SES) 傳送報告。

.NET

AWS SDK for .NET

示範如何使用 Amazon 簡單電子郵件服務 (Amazon SES) 建立追蹤 Amazon Aurora 資料庫中工作項目的 Web 應用程式，以及透過電子郵件傳送報告。AWS SDK for .NET 這個範例使用以 React.js 建置的前端與 RESTful .NET 後端互動。

- 將 React 網頁應用程式與 AWS 服務整合。
- 列出、新增、更新和刪除 Aurora 資料表中的項目。
- 使用 Amazon SES 傳送篩選工作項目的電子郵件報告。

- 使用隨附的 AWS CloudFormation 指令碼部署和管理範例資源。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

C++

適用於 C++ 的 SDK

說明如何建立可追蹤和報告存放在 Amazon Aurora Serverless 資料庫中的工作項目的 Web 應用程式。

如需有關如何設定 C++ REST API 以查詢 Amazon Aurora 無伺服器資料以及供 React 應用程式使用的完整原始程式碼和說明，請參閱上[GitHub](#)的完整範例。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

Java

適用於 Java 2.x 的 SDK

說明如何建立可追蹤和報告存放在 Amazon RDS 資料庫中的工作項目的 Web 應用程式。

如需有關如何設定 Spring REST API 以查詢 Amazon Aurora 無伺服器資料以及供 React 應用程式使用的完整原始程式碼和說明，請參閱上[GitHub](#)的完整範例。

有關如何設置和運行使用 JDBC API 的示例的完整源代碼和說明，請參閱上[GitHub](#)的完整示例。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

JavaScript

適用於 JavaScript (v3) 的開發套件

示範如何使用 AWS SDK for JavaScript (v3) 建立 Web 應用程式，以追蹤 Amazon Aurora 資料庫中的工作項目，以及使用 Amazon 簡易電子郵件服務 (Amazon SES) 傳送電子郵件報告的 Web 應用程式。這個範例使用以 React.js 建置的前端與 Express Node.js 後端互動。

- 將 React.js 網路應用程式與 AWS 服務。
- 列出、新增和更新 Aurora 資料表中的項目。
- 使用 Amazon SES 傳送篩選工作項目的電子郵件報告。
- 使用隨附的 AWS CloudFormation 指令碼部署和管理範例資源。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

Kotlin

適用於 Kotlin 的 SDK

說明如何建立可追蹤和報告存放在 Amazon RDS 資料庫中的工作項目的 Web 應用程式。

如需有關如何設定 Spring REST API 以查詢 Amazon Aurora 無伺服器資料以及供 React 應用程式使用的完整原始程式碼和說明，請參閱上[GitHub](#)的完整範例。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

PHP

適用於 PHP 的開發套件

示範如何使用 Amazon 簡單電子郵件服務 (Amazon SES) 建立追蹤 Amazon RDS 資料庫中工作項目的 Web 應用程式，並以電子郵件傳送報告。AWS SDK for PHP 這個範例使用以 React.js 建置的前端與 RESTful PHP 後端互動。

- 將 React.js 網路應用程式與 AWS 服務整合。
- 列出、新增、更新和刪除 Amazon RDS 資料表中的項目。
- 使用 Amazon SES 傳送篩選工作項目的電子郵件報告。
- 使用隨附的 AWS CloudFormation 指令碼部署和管理範例資源。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

Python

適用於 Python (Boto3) 的 SDK

示範如何使用亞馬遜簡單電子郵件服務 (Amazon SES) 建立 REST 服務，以追蹤 Amazon Aurora 無伺服器資料庫中的工作項目和電子郵件報告。AWS SDK for Python (Boto3) 這個範例使用 Flask Web 框架來處理 HTTP 路由，並與 React 網頁整合以呈現功能完整的 Web 應用程式。

- 建置整合的 AWS 服務燒瓶 REST 服務。

- 讀取、寫入和更新儲存在 Aurora 無伺服器資料庫中的工作項目。
- 建立包含資料庫認證的 AWS Secrets Manager 密碼，並使用它來驗證對資料庫的呼叫。
- 使用 Amazon SES 傳送工作項目的電子郵件報告。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Aurora
- Amazon RDS
- Amazon RDS 資料服務
- Amazon SES

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Amazon RDS 中的安全

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全是 AWS 與您共同的責任。[共同的責任模型](#) 將此描述為雲端本身的安全和雲端內部的安全：

- 雲端本身的安全 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 [AWS 合規計劃](#) 中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon RDS 的合規計劃，請參閱 [AWS 合規計劃的服務範圍](#)。
- 雲端內部的安全 – 您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的敏感度、您組織的需求和適用的法律及法規。

本文件有助於您了解如何在使用 Amazon RDS 時套用共同責任模型。下列主題將示範如何設定 Amazon RDS 以達到您的安全和合規目標。您也會了解如何使用其他 AWS 服務來協助您監控並保護 Amazon RDS 資源。

您可以管理對 Amazon RDS 資源以及資料庫執行個體上資料庫的存取權限。您用來管理存取權限的方法取決於使用者需要利用 Amazon RDS 執行何種類型的任務：

- 在以 Amazon VPC 服務為基礎的 virtual private cloud (VPC) 中執行您的資料庫執行個體，以盡可能地取得最大的網路存取控制能力。如需在 VPC 中建立資料庫執行個體的詳細資訊，請參閱 [Amazon VPC 和 Amazon RDS](#)。
- 使用 AWS Identity and Access Management (IAM) 政策來指派許可，決定可以管理 Amazon RDS 資源的人員。例如，您可以使用 IAM 來決定誰可以建立、描述、修改和刪除資料庫執行個體、標記資源，或修改安全群組。
- 使用安全群組來控制哪些 IP 地址或 Amazon EC2 執行個體可以連線至資料庫執行個體上的資料庫。當您第一次建立資料庫執行個體時，其防火牆會防止任何資料庫存取，但透過相關聯的安全群組所指定規則進行的資料庫存取除外。
- 對執行 Db2、MySQL、MariaDB、PostgreSQL、甲骨文或 Microsoft SQL 伺服器資料庫引擎的資料庫執行個體使用安全通訊端層 (SSL) 或傳輸層安全性 (TLS) 連線。如需搭配資料庫執行個體使用 SSL/TLS 的詳細資訊，請參閱。
- 使用 Amazon RDS 加密來保護資料庫執行個體、和靜態快照。Amazon RDS 加密資料庫執行個體叢集會使用業界標準 AES-256 加密演算法，來加密託管資料庫執行個體伺服器上的資料。如需更多詳細資訊，請參閱 [加密 Amazon RDS 資源](#)。

- 使用網路加密和透明資料加密與 Oracle 資料庫執行個體搭配；如需詳細資訊，請參閱 [Oracle 原生網路加密](#) 和 [Oracle 透明資料加密](#)
- 使用資料庫引擎的安全功能，來控制誰可以登入資料庫執行個體上的資料庫。這項功能的運作方式就好像資料庫位在本機網路上。

Note

您只須針對您的使用案例設定安全。您不需要為 Amazon RDS 管理的程序設定安全存取。這些包括建立備份、在主要資料庫執行個體和僅供讀取複本間複寫資料，以及其他程序。

如需管理對 Amazon RDS 資源及在您資料庫執行個體上資料庫存取權限的詳細資訊，請參閱下列主題。

主題

- [使用 Amazon RDS 進行資料庫身分驗證](#)
- [使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager](#)
- [Amazon RDS 中的資料保護](#)
- [Amazon RDS 的 Identity and access management](#)
- [Amazon RDS 中的記錄和監控](#)
- [Amazon RDS 的合規驗證](#)
- [Amazon RDS 的復原功能](#)
- [Amazon RDS 中的基礎設施安全](#)
- [Amazon RDS API 和界面 VPC 端點 \(AWS PrivateLink\)](#)
- [Amazon RDS 的安全最佳實務](#)
- [使用安全群組控制存取](#)
- [主要使用者帳戶權限](#)
- [使用 Amazon RDS 的服務連結角色](#)
- [Amazon VPC 和 Amazon RDS](#)

使用 Amazon RDS 進行資料庫身分驗證

Amazon RDS 支援驗證資料庫使用者的數種方式。

密碼、Kerberos 和 IAM 資料庫身分驗證會對資料庫使用不同的身分驗證方法。因此，特定使用者只能使用一種身分驗證方法登入資料庫。

對於 PostgreSQL，請僅針對特定資料庫的使用者使用下列其中一個角色設定：

- 若要使用 IAM 資料庫身分驗證，請將 `rds_iam` 角色指派給使用者。
- 若要使用 Kerberos 身分驗證，請將 `rds_ad` 角色指派給使用者。
- 若要使用密碼身分驗證，請勿指派 `rds_iam` 或 `rds_ad` 角色給使用者。

不要同時將 `rds_iam` 和 `rds_ad` 角色指派給 PostgreSQL 資料庫的使用者，無論是直接或間接透過巢狀授權存取。如果 `rds_iam` 角色新增至主要使用者，則 IAM 身分驗證優先於密碼身分驗證，因此主要使用者必須以 IAM 使用者身分登入。

Important

我們強烈建議您不要直接在您的應用程式中使用主要使用者。而是遵循最佳實務，使用以應用程式所需的最低權限建立的資料庫使用者。

主題

- [密碼身分驗證](#)
- [IAM 資料庫身分驗證](#)
- [Kerberos 身分驗證](#)

密碼身分驗證

使用密碼身分驗證，您的資料庫會執行使用者帳戶的所有管理。您可以使用 SQL 陳述式 (例如 `CREATE USER`) 搭配資料庫引擎指定密碼所需的適當子句，來建立使用者。例如，在 MySQL 中，陳述式為 `CREATE USER name IDENTIFIED BY password`，而在 PostgreSQL 中，陳述式為 `CREATE USER name WITH PASSWORD password`。

透過密碼身分驗證，您的資料庫可控制並驗證使用者帳戶。如果資料庫引擎具有強大的密碼管理功能，它們可以增強安全性。當您擁有較小的使用者社群時，使用密碼身分驗證來管理資料庫身分驗證可能會更容易。由於在此情況下會產生純文字密碼，因此整合 AWS Secrets Manager 可增強安全性。

如需有關將 Secrets Manager 與 Amazon RDS 搭配使用的詳細資訊，請參閱 AWS Secrets Manager 使用者指南中的[建立基本密碼](#)和[輪換支援 Amazon RDS 資料庫的密碼](#)。如需以程式設計方式擷取自訂應用程式中的密碼的相關資訊，請參閱 AWS Secrets Manager 使用者指南中的[擷取密碼值](#)。

IAM 資料庫身分驗證

您可以使用 AWS Identity and Access Management (IAM) 資料庫身份驗證向資料庫執行個體進行驗證。透過此身分驗證方法，您連線至資料庫執行個體時不需要使用密碼。而是改用身分驗證字符。

如需有關 IAM 資料庫身分驗證的詳細資訊，包括特定資料庫引擎可用性的相關資訊，請參閱[適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

Kerberos 身分驗證

Amazon RDS 支援透過 Kerberos 和 Microsoft Active Directory 對資料庫使用者進行外部身分驗證。Kerberos 是網路身分驗證通訊協定，使用票證和對稱式金鑰加密技術，免除透過網路傳輸密碼的需要。Kerberos 已內建至 Active Directory，旨在驗證網路資源 (例如資料庫) 的使用者身分。

Amazon RDS 對 Kerberos 和 Active Directory 的支援，提供了資料庫使用者的單一登入和集中式身分驗證優點。您可以在 Active Directory 中保留您的使用者登入資料。Active Directory 提供集中位置存放及管理多個資料庫執行個體登入資料。


啟用資料庫使用者針對資料庫執行個體進行身份認證，您可以有兩種方式。他們可以使用儲存在內部部署 Active Directory 中 AWS Directory Service for Microsoft Active Directory 或內部部署中的認證。

適用於 PostgreSQL 的 RDS 不支援樹系信任中的選擇性驗證類型，只支援整個樹系驗證。

Microsoft SQL Server 和 PostgreSQL 資料庫執行個體支援單向和雙向的樹系信任關係。Oracle 資料庫執行個體支援單向和雙向的外部及樹系信任關係。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[建立信任關係的時機](#)。

如需使用特定資料庫引擎進行 Kerberos 身分驗證的相關資訊，請參閱下列內容：

- [使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄](#)
- [針對 MySQL 使用 Kerberos 身分驗證](#)
- [為 Amazon RDS for Oracle 設定 Kerberos 身分驗證](#)
- [搭配 Amazon RDS for PostgreSQL 使用 Kerberos 身分驗證](#)

 Note

目前，MariaDB 資料庫執行個體不支援 Kerberos 身分驗證。

使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager

Amazon RDS 與 Secrets Manager 整合，以管理資料庫執行個體和多可用區域資料庫叢集的主要使用者密碼。

主題

- [Secrets Manager 與 Amazon RDS 整合的限制](#)
- [管理主要使用者密碼的概觀 AWS Secrets Manager](#)
- [使用 Secrets Manager 管理主要使用者密碼的優點](#)
- [Secrets Manager 整合所需的許可](#)
- [在中強制執行 RDS 管理主要使用者密碼 AWS Secrets Manager](#)
- [使用 Secrets Manager 管理資料庫執行個體的主要使用者密碼](#)
- [使用 Secrets Manager 管理多可用區域資料庫叢集的主要使用者密碼](#)
- [輪換資料庫執行個體的主要使用者密碼機密](#)
- [輪換多可用區域資料庫叢集的主要使用者密碼機密](#)
- [檢視資料庫執行個體機密的詳細資訊](#)
- [檢視多可用區域資料庫叢集之機密的詳細資訊](#)
- [區域和版本可用性](#)

Secrets Manager 與 Amazon RDS 整合的限制

下列功能不支援使用 Secrets Manager 管理主要使用者密碼：

- 當來源資料庫或資料庫叢集使用 Secrets Manager 管理認證時，建立僅供讀取複本。這適用於除 SQL 伺服器的 RDS 以外的所有資料庫引擎。
- Amazon RDS 藍/綠部署
- Amazon RDS Custom
- Oracle Data Guard 切換
- 具有 CDB 的 RDS for Oracle

管理主要使用者密碼的概觀 AWS Secrets Manager

使用 AWS Secrets Manager，您可以透過 API 呼叫 Secrets Manager 來取代程式碼中的硬式編碼認證 (包括資料庫密碼)，以程式設計方式擷取密碼。如需 Secrets Manager 的詳細資訊，請參閱 [AWS Secrets Manager 使用者指南](#)。

當您將資料庫密碼儲存在 Secrets Manager 中時，AWS 帳戶 會產生費用。如需定價的資訊，請參閱 [AWS Secrets Manager 定價](#)。

當您執行下列其中一項操作時，您可以指定 RDS，在 Secrets Manager 中管理 Amazon RDS 資料庫執行個體或多可用區域資料庫叢集的主要使用者密碼：

- 建立資料庫執行個體
- 建立多可用區域資料庫叢集
- 修改資料庫執行個體
- 修改多可用區域資料庫叢集
- 從 Amazon S3 還原資料庫執行個體

當您指定 RDS 在 Secrets Manager 中管理主要使用者密碼時，RDS 會產生密碼並將其存放在 Secrets Manager 中。您可以直接與機密互動，以擷取主要使用者的憑證。您也可以指定客戶受管金鑰來加密機密，或使用 Secrets Manager 提供的 KMS 金鑰。

RDS 會管理機密的設定，並依預設每七天輪換一次密碼。您可以修改某些設定，例如輪換排程。如果您刪除在 Secrets Manager 中管理密碼的資料庫執行個體，秘密及其相關聯的中繼資料也會一併刪除。

若要使用機密中的憑證連線至資料庫執行個體或多可用區域資料庫叢集，您可以從 Secrets Manager 擷取機密。如需詳細資訊，請參閱《AWS Secrets Manager 使用指南》中的使用 [密 AWS Secrets Manager 碼中的認證從 SQL 資料庫擷取密碼 AWS Secrets Manager 和 Connect 到 SQL 資料庫](#)。

使用 Secrets Manager 管理主要使用者密碼的優點

使用 Secrets Manager 管理 RDS 主要使用者密碼可提供下列優點：

- RDS 會自動產生資料庫憑證。
- RDS 會在中自動儲存和管理資料庫認證 AWS Secrets Manager。
- RDS 會定期輪換資料庫憑證，而無需變更應用程式。

- Secrets Manager 會保護資料庫憑證免於人類存取和純文字檢視。
- Secrets Manager 允許擷取機密中用於資料庫連線的資料庫憑證。
- Secrets Manager 允許使用 IAM 對機密中資料庫憑證的存取進行細微控制。
- 您可以選擇性地使用不同 KMS 金鑰將資料庫加密與憑證加密分開。
- 您可以消除資料庫憑證的手動管理和輪換。
- 您可以使用 AWS CloudTrail 和 Amazon 輕鬆監控數據庫憑據 CloudWatch。

如需 Secrets Manager 優點的詳細資訊，請參閱《AWS Secrets Manager 使用者指南》<https://docs.aws.amazon.com/secretsmanager/latest/userguide/>。

Secrets Manager 整合所需的許可

使用者必須具有必要的權限，才能執行與 Secrets Manager 整合相關的操作。您可以建立 IAM 政策，授與對其需要的指定資源執行特定 API 操作的許可。然後，您可以將這些政策連接至需要這些許可的 IAM 許可集或角色。如需詳細資訊，請參閱 [Amazon RDS 的 Identity and access management](#)。

對於建立、修改或還原操作，指定 Amazon RDS 在 Secrets Manager 中管理主要使用者密碼的使用者必須具有執行下列操作的許可：

- kms:DescribeKey
- secretsmanager:CreateSecret
- secretsmanager:TagResource

對於建立、修改或還原操作，指定客戶受管金鑰以加密 Secrets Manager 中機密的使用者必須具有執行下列操作的許可：

- kms:Decrypt
- kms:GenerateDataKey
- kms:CreateGrant

對於修改操作，在 Secrets Manager 中輪換主要使用者密碼的使用者必須具有執行下列操作的許可：

- secretsmanager:RotateSecret

在中強制執行 RDS 管理主要使用者密碼 AWS Secrets Manager

您可以使用 IAM 條件金鑰，在 AWS Secrets Manager 中強制執行主要使用者密碼的 RDS 管理。除非主要使用者密碼是在 Secrets Manager 中由 RDS 管理，否則下列政策不允許使用者建立或還原資料庫執行個體或資料庫叢集。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": ["rds:CreateDBInstance", "rds:CreateDBCluster",
        "rds:RestoreDBInstanceFromS3", "rds:RestoreDBClusterFromS3"],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "rds:ManageMasterUserPassword": false
        }
      }
    }
  ]
}
```

Note

此原則會在建立 AWS Secrets Manager 時強制執行密碼管理。不過，您仍然可以停用 Secrets Manager 整合，並透過修改執行個體手動設定主要密碼。

為了防止這種情況，請在政策的動作區塊中包含

`rds:ModifyDBInstance`、`rds:ModifyDBCluster`。請注意，這可防止使用者對未啟用 Secrets Manager 整合的現有執行個體套用任何進一步的修改。

如需在 IAM 政策中使用條件金鑰的詳細資訊，請參閱 [Amazon RDS 的政策條件金鑰](#) 和 [範例政策：使用條件金鑰](#)。

使用 Secrets Manager 管理資料庫執行個體的主要使用者密碼

執行下列動作時，您可以在 Secrets Manager 中設定主要使用者密碼的 RDS 管理：

- [建立 Amazon RDS 資料庫執行個體](#)

- [修改 Amazon RDS 資料庫執行個體](#)
- [將備份還原至 MySQL 資料庫執行個體](#)

您可以使用 RDS 主控台、AWS CLI、或 RDS API 來執行這些動作。

主控台

請遵循使用 RDS 主控台建立或修改資料庫執行個體的指示：

- [建立資料庫執行個體](#)
- [修改 Amazon RDS 資料庫執行個體](#)
- [將資料從 Amazon S3 匯入新的 MySQL 資料庫執行個體](#)

當使用 RDS 主控台來執行其中一項操作時，您可以指定主要使用者密碼是由 Secrets Manager 中的 RDS 管理。若要在建立或還原資料庫執行個體時這樣做，請在 Credential settings (憑證設定) 中選取 Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證)。修改資料庫執行個體時，請在 Settings (設定) 中選取 Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證)。

以下影像是建立或還原資料庫執行個體時，Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證) 設定的範例。

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

當您選取此選項時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

▼ **Credentials Settings**


Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

Select the encryption key [Info](#)
You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.

aws/secretsmanager (default) ▼

[Add new key](#) 

您可以選擇使用 Secrets Manager 提供的 KMS 金鑰，或您建立的客戶受管金鑰來加密機密。在 RDS 管理資料庫執行個體的資料庫憑證之後，您就無法變更改用來加密機密的 KMS 金鑰。

您可以選擇其他設定來符合您的需求。如需您建立資料庫執行個體時可用設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。如需您修改資料庫執行個體時可用設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

AWS CLI

若要在密碼管理員中使用 RDS 管理主要使用者密碼，請在下列其中一個 AWS CLI 命令中指定 `--manage-master-user-password` 選項：

- [create-db-instance](#)
- [modify-db-instance](#)
- [restore-db-instance-from-S3](#)

當您在這些命令中指定 `--manage-master-user-password` 選項時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

若要加密機密，您可以指定客戶受管金鑰或使用 Secrets Manager 提供的預設 KMS 金鑰。使用 `--master-user-secret-kms-key-id` 選項來指定客戶受管金鑰。AWS KMS 金鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。若要在不同的 KMS 金鑰中使用 AWS 帳戶，請指定金鑰 ARN 或別名 ARN。在 RDS 管理資料庫執行個體的資料庫憑證之後，您就無法變更新用來加密機密的 KMS 金鑰。

您可以選擇其他設定來符合您的需求。如需您建立資料庫執行個體時可用設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。如需您修改資料庫執行個體時可用設定的詳細資訊，請參閱 [資料庫執行個體的設定](#)。

此範例會建立資料庫執行個體，並指定 RDS 在 Secrets Manager 中管理主要使用者密碼。機密會使用 Secrets Manager 所提供的 KMS 金鑰進行加密。

Example

對於 Linux/macOS、或 Unix：

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --engine mysql \  
  --engine-version 8.0.30 \  
  --db-instance-class db.r5b.large \  
  --allocated-storage 200 \  
  --manage-master-user-password
```

```
--manage-master-user-password
```

在 Windows 中：

```
aws rds create-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --engine mysql ^  
  --engine-version 8.0.30 ^  
  --db-instance-class db.r5b.large ^  
  --allocated-storage 200 ^  
  --manage-master-user-password
```

RDS API

若要指定 RDS 在 Secrets Manager 中管理主要使用者密碼，請在下列其中一個 RDS API 操作中將 `ManageMasterUserPassword` 參數設為 `true`：

- [CreateDBInstance](#)
- [ModifyDBInstance](#)
- [恢復 B S3 InstanceFrom](#)

當您在其中一個操作中將 `ManageMasterUserPassword` 參數設為 `true` 時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

若要加密機密，您可以指定客戶受管金鑰或使用 Secrets Manager 提供的預設 KMS 金鑰。使用 `MasterUserSecretKmsKeyId` 參數指定客戶受管金鑰。AWS KMS 金鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。若要在不同的 AWS 帳戶中使用 KMS 金鑰，請指定金鑰 ARN 或別名 ARN。在 RDS 管理資料庫執行個體的資料庫憑證之後，您就無法變更新用來加密機密的 KMS 金鑰。

使用 Secrets Manager 管理多可用區域資料庫叢集的主要使用者密碼

執行下列動作時，您可以在 Secrets Manager 中設定主要使用者密碼的 RDS 管理：

- [建立多可用區域資料庫叢集](#)
- [修改多可用區域資料庫叢集](#)

您可以使用 RDS 主控台、AWS CLI、或 RDS API 來執行這些動作。

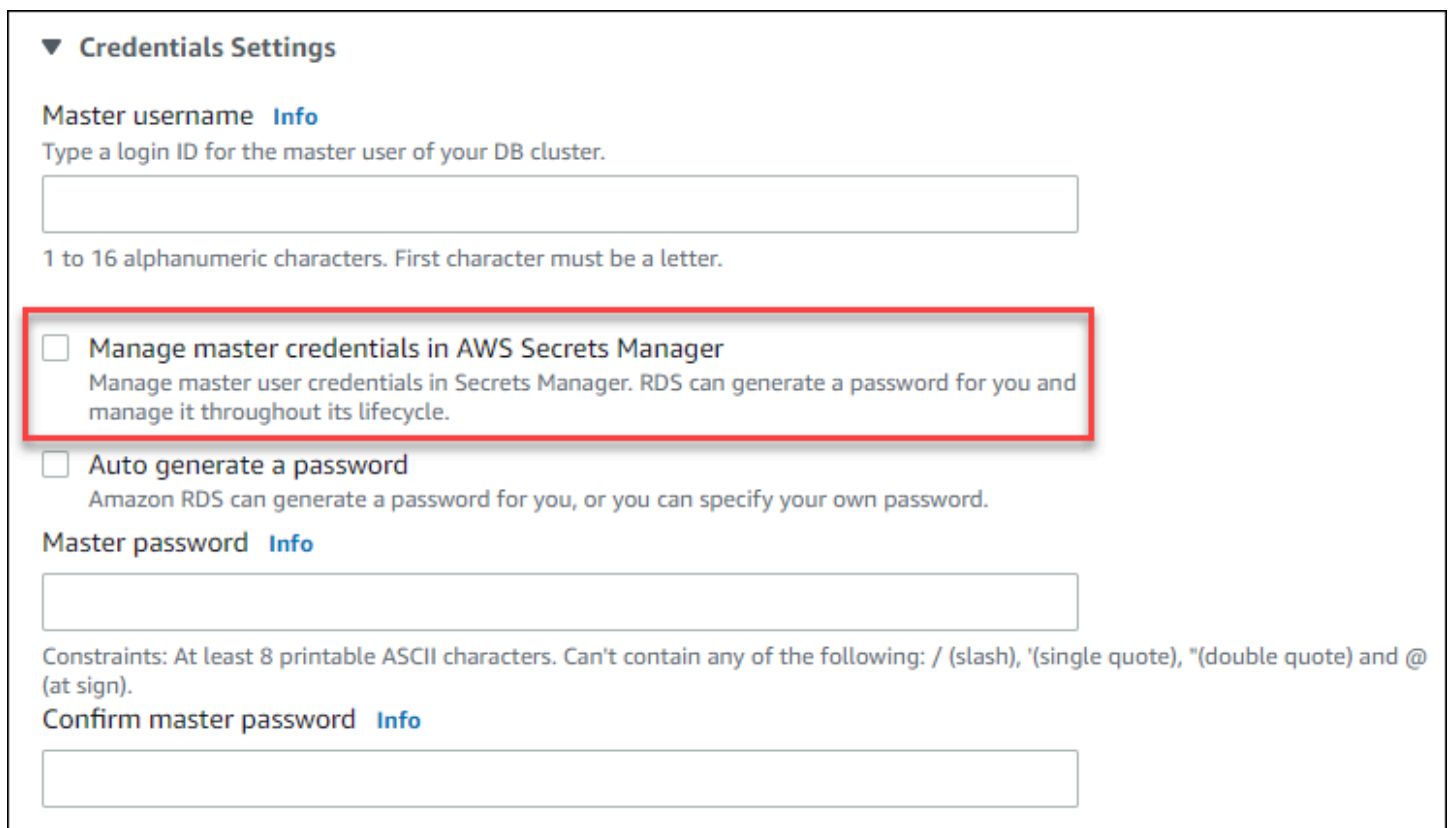
主控台

請遵循使用 RDS 主控台建立或修改多可用區域資料庫叢集的指示：

- [建立資料庫叢集](#)
- [修改多可用區域資料庫叢集](#)

當使用 RDS 主控台來執行其中一項操作時，您可以指定主要使用者密碼是由 Secrets Manager 中的 RDS 管理。若要在建立資料庫叢集時這樣做，請在 Credential settings (憑證設定) 中選取 Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證)。修改資料庫叢集時，請在 Settings (設定) 中選取 Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證)。

以下影像是建立資料庫叢集時，Manage master credentials in AWS Secrets Manager (在 AWS Secrets Manager 中管理主要憑證) 設定的範例。



▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB cluster.

1 to 16 alphanumeric characters. First character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

當您選取此選項時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

▼ **Credentials Settings**


Master username [Info](#)
Type a login ID for the master user of your DB cluster.

1 to 16 alphanumeric characters. First character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

Select the encryption key [Info](#)
You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.

aws/secretsmanager (default)

[Add new key](#) 

您可以選擇使用 Secrets Manager 提供的 KMS 金鑰，或您建立的客戶受管金鑰來加密機密。在 RDS 管理資料庫叢集的資料庫憑證之後，您就無法變更新用來加密機密的 KMS 金鑰。

您可以選擇其他設定來符合您的需求。

如需當您建立多可用區域資料庫叢集時可用設定的詳細資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。如需當您修改多可用區域資料庫叢集時可用設定的詳細資訊，請參閱 [修改多可用區域資料庫叢集的設定](#)。

AWS CLI

若要指定 RDS 在 Secrets Manager 中管理主要使用者密碼，請在下列其中一個命令中指定 `--manage-master-user-password` 選項：

- [create-db-cluster](#)
- [modify-db-cluster](#)

當您在這些命令中指定 `--manage-master-user-password` 選項時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

若要加密機密，您可以指定客戶受管金鑰或使用 Secrets Manager 提供的預設 KMS 金鑰。使用 `--master-user-secret-kms-key-id` 選項來指定客戶受管金鑰。AWS KMS 金鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。若要在不同的 KMS 金鑰中使用 AWS 帳戶，請指

定金鑰 ARN 或別名 ARN。在 RDS 管理資料庫叢集的資料庫認證之後，您無法變更新來加密機密的 KMS 金鑰。

您可以選擇其他設定來符合您的需求。

如需當您建立多可用區域資料庫叢集時可用設定的詳細資訊，請參閱 [建立多可用區域資料庫叢集的設定](#)。如需當您修改多可用區域資料庫叢集時可用設定的詳細資訊，請參閱 [修改多可用區域資料庫叢集的設定](#)。

此範例會建立多可用區域資料庫叢集，並指定 RDS 在 Secrets Manager 中管理密碼。機密會使用 Secrets Manager 所提供的 KMS 金鑰進行加密。

Example

對於LinuxmacOS、或Unix：

```
aws rds create-db-cluster \  
  --db-cluster-identifier mysql-multi-az-db-cluster \  
  --engine mysql \  
  --engine-version 8.0.28 \  
  --backup-retention-period 1 \  
  --allocated-storage 4000 \  
  --storage-type io1 \  
  --iops 10000 \  
  --db-cluster-instance-class db.r6gd.xlarge \  
  --manage-master-user-password
```

在 Windows 中：

```
aws rds create-db-cluster ^  
  --db-cluster-identifier mysql-multi-az-db-cluster ^  
  --engine mysql ^  
  --engine-version 8.0.28 ^  
  --backup-retention-period 1 ^  
  --allocated-storage 4000 ^  
  --storage-type io1 ^  
  --iops 10000 ^  
  --db-cluster-instance-class db.r6gd.xlarge ^  
  --manage-master-user-password
```

RDS API

若要指定 RDS 在 Secrets Manager 中管理主要使用者密碼，請在下列其中一個操作中將 `ManageMasterUserPassword` 參數設為 `true`：

- [CreateDBCluster](#)
- [ModifyDBCluster](#)

當您在其中一個操作中將 `ManageMasterUserPassword` 參數設為 `true` 時，RDS 會產生主要使用者密碼，並在 Secrets Manager 中於整個生命週期管理該密碼。

若要加密機密，您可以指定客戶受管金鑰或使用 Secrets Manager 提供的預設 KMS 金鑰。使用 `MasterUserSecretKmsKeyId` 參數指定客戶受管金鑰。AWS KMS 金鑰識別碼是 KMS 金鑰的金鑰 ARN、金鑰識別碼、別名 ARN 或別名。若要在不同的 AWS 帳戶中使用 KMS 金鑰，請指定金鑰 ARN 或別名 ARN。在 RDS 管理資料庫叢集的資料庫認證之後，您無法變更新用來加密機密的 KMS 金鑰。

輪換資料庫執行個體的主要使用者密碼機密

當 RDS 輪換主要使用者密碼機密時，Secrets Manager 會針對現有機密產生新的機密版本。新版本的機密包含新的主要使用者密碼。Amazon RDS 會變更資料庫執行個體的主要使用者密碼，以符合新機密版本的密碼。

您可以立即輪換機密，而無需等待排程的輪換。若要在 Secrets Manager 中輪換主要使用者密碼機密，請修改資料庫執行個體。如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

您可以使用 RDS 主控台、或 RDS API 立即輪換主要使用者密碼密碼。AWS CLI 新密碼長度一律為 28 個字元，且至少包含一個大小寫字元、一個數字以及一個標點符號。

主控台

若要使用 RDS 主控台輪換主要使用者密碼機密，請修改資料庫執行個體，然後在 Settings (設定) 中選取 `Rotate secret immediately` (立即輪換秘密)。

Settings

DB engine version
Version number of the database engine to be used for this database

8.0.30 ▼

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

- Manage master credentials in AWS Secrets Manager**
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.
- Rotate secret immediately**
When you rotate a secret, you update the credentials in both the secret and the database.

請遵循使用 [修改 Amazon RDS 資料庫執行個體](#) 中的 RDS 主控台修改資料庫執行個體的指示。您必須在確認頁面上選擇 Apply immediately (立即套用)。

AWS CLI

若要使用旋轉主要使用者密碼密碼 AWS CLI，請使用指 [modify-db-instance](#) 令並指定 `--rotate-master-user-password` 選項。您必須在輪換主要密碼時指定 `--apply-immediately` 選項。

此範例會輪換主要使用者密碼機密。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --rotate-master-user-password \  
  --apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^
```

```
--db-instance-identifier mydbinstance ^  
--rotate-master-user-password ^  
--apply-immediately
```

RDS API

您可以使用 [ModifyDBInstance](#) 操作，並將 `RotateMasterUserPassword` 參數設為 `true`，來輪換主要使用者密碼機密。輪換主要密碼時，必須將 `ApplyImmediately` 參數設為 `true`。

輪換多可用區域資料庫叢集的主要使用者密碼機密

當 RDS 輪換主要使用者密碼機密時，Secrets Manager 會針對現有機密產生新的機密版本。新版本的機密包含新的主要使用者密碼。Amazon RDS 會變更多可用區域資料庫叢集的主要使用者密碼，以符合新秘密版本的密碼。

您可以立即輪換機密，而無需等待排程的輪換。若要在 Secrets Manager 中輪換主要使用者密碼機密，請修改多可用區域資料庫叢集。如需修改多可用區域資料庫叢集的詳細資訊，請參閱 [修改多可用區域資料庫叢集](#)。

您可以使用 RDS 主控台、或 RDS API 立即輪換主要使用者密碼密碼。AWS CLI 新密碼長度一律為 28 個字元，且至少包含一個大小寫字元、一個數字以及一個標點符號。

主控台

若要使用 RDS 主控台輪換主要使用者密碼機密，請修改多可用區域資料庫叢集，然後在 Settings (設定) 中選取 `Rotate secret immediately` (立即輪換機密)。

Settings

Engine Version [Info](#)

MySQL 8.0.30

To see more versions, modify the capacity types. [Info](#)

DB cluster identifier [Info](#)

Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

database-2

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

DB cluster identifier

The identifier for the DB cluster.

database-2

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

Rotate secret immediately
When you rotate a secret, you update the credentials in both the secret and the database.

遵循使用 [修改多可用區域資料庫叢集](#) 中的 RDS 主控台修改多可用區域資料庫叢集的指示。您必須在確認頁面上選擇 Apply immediately (立即套用)。

AWS CLI

若要使用旋轉主要使用者密碼密碼 AWS CLI，請使用指 [modify-db-cluster](#) 令並指定 `--rotate-master-user-password` 選項。您必須在輪換主要密碼時指定 `--apply-immediately` 選項。

此範例會輪換主要使用者密碼機密。

Example

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --rotate-master-user-password \  
  --apply-immediately
```

```
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-cluster ^  
  --db-cluster-identifier mydbcluster ^  
  --rotate-master-user-password ^  
  --apply-immediately
```

RDS API

您可以使用 [ModifyDBCluster](#) 操作，並將 `RotateMasterUserPassword` 參數設為 `true`，來輪換主要使用者密碼機密。輪換主要密碼時，必須將 `ApplyImmediately` 參數設為 `true`。

檢視資料庫執行個體機密的詳細資訊

您可以使用控制台 (<https://console.aws.amazon.com/secretsmanager/>) 或 AWS CLI ([get-secret-value](#) 密碼管理器命令) 檢索密碼。

您可以在機密管理器中使用 RDS 主控台、或 RDS API 找到由 RDS 管理的密碼的 Amazon 資源名稱 (ARN)。AWS CLI

主控台

檢視 Secrets Manager 中由 RDS 管理之機密的詳細資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇資料庫執行個體的名稱，以顯示其詳細資訊。
4. 選擇 Configuration (組態) 索引標籤。

在 Master Credentials ARN (主要憑證 ARN) 中，您可以檢視機密 ARN。

The screenshot displays the AWS Management Console interface for an Amazon RDS instance. The 'Configuration' tab is selected, showing various instance details. A red box highlights the 'Master Credentials ARN' field, which contains the following information:

- Master Credentials ARN
- arn:aws:secretsmanager:ap-south-1: [redacted]:secret:rds!db-71d9c43d-4022-44a6-bc18-a67bb156d5a8-RzRqmA
- [Manage in Secrets Manager](#)

Other visible details include:

- Configuration:** DB instance ID (database-1), Engine version (8.0.30), DB name (-), License model (General Public License), Option groups (default:mysql-8-0, In sync), Amazon Resource Name (ARN) (arn:aws:rds:ap-south-1:[redacted]:db:database-1), Resource ID (db-[redacted]), Created time (December 20, 2022, 09:10 (UTC-08:00)), Parameter group (default.mysql8.0, In sync), Deletion protection (Enabled).
- Instance class:** Instance class (db.m6g.large), vCPU (2), RAM (8 GB).
- Availability:** Master username (admin), IAM DB authentication (Not enabled), Multi-AZ (No), Secondary Zone (-).
- Storage:** Encryption (Enabled), AWS KMS key (aws/rds), Storage type (Provisioned), Storage (400 GiB), Provisioned IOPS (3000 IOPS), Storage throughput (-), Storage auto scaling (Enabled), Maximum storage capacity (1000 GiB).

您可以遵循 [Manage in Secrets Manager](#) (在 Secrets Manager 中管理) 連結，在 Secret Manager 主控台中檢視和管理機密。

AWS CLI

您可以使用 [describe-db-instances](#) RDS CLI 命令，在密碼管理員中尋找由 RDS 管 Secrets Manager 碼的下列相關資訊：

- SecretArn – 機密的 ARN
- SecretStatus – 機密的狀態

可能的狀態值包括下列項目：

- creating – 正在建立機密。
- active – 機密可供正常使用和輪換。
- rotating – 正在輪換機密。
- impaired – 機密可以用來存取資料庫憑證，但無法將其輪換。例如，如果變更許可，以便 RDS 再也無法取機密或秘密的 KMS 金鑰，則秘密可能會具有此狀態。

當密碼具有此狀態時，您可以更正導致狀態的條件。如果您更正了導致狀態的條件，則狀態仍會保留 impaired，直到下一次輪換為止。或者，您可以修改資料庫執行個體，以關閉資料庫憑證的自動管理，然後再次修改資料庫執行個體，以開啟資料庫憑證的自動管理。若要修改資料庫執行個體，請使用 [modify-db-instance](#) 指令中的 `--manage-master-user-password` 選項。

- KmsKeyId – 用來加密機密之 KMS 金鑰的 ARN

指定 `--db-instance-identifier` 選項來顯示特定資料庫執行個體的輸出。此範例顯示資料庫執行個體所使用之機密的輸出。

Example

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

以下是機密的範例輸出：

```
"MasterUserSecret": {
    "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
    "SecretStatus": "active",
    "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
}
```

當您擁有秘密 ARN 時，您可以使用秘[get-secret-value](#)密管理員 CLI 命令來檢視有關密碼的詳細資料。

此範例顯示前一個範例輸出中機密的詳細資訊。

Example

對於LinuxmacOS、或Unix：

```
aws secretsmanager get-secret-value \  
  --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!  
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

在 Windows 中：

```
aws secretsmanager get-secret-value ^  
  --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!  
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

RDS API

您可以檢視 Secrets Manager 中由 RDS 管理之機密的 ARN、狀態和 KMS 金鑰，方法為使用 [DescribeDBInstances](#) 操作，並將 DBInstanceIdentifier 參數設為資料庫執行個體識別符。機密的詳細資訊包含在輸出中。

當您擁有秘密 ARN 時，您可以使用「秘[GetSecretValue](#)密管理員」作業來檢視有關密碼的詳細資料。

檢視多可用區域資料庫叢集之機密的詳細資訊

您可以使用控制台 (<https://console.aws.amazon.com/secretsmanager/>) 或 AWS CLI ([get-secret-value](#)密碼管理器命令) 檢索密碼。

您可以使用 RDS 主控台、或 RDS API 在機密管理員中找到由 RDS 所管理的密碼的 Amazon 資源名稱 (ARN)。AWS CLI

主控台

檢視 Secrets Manager 中由 RDS 管理之機密的詳細資訊

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇多可用區域資料庫叢集的名稱，以顯示其詳細資訊。
4. 選擇 Configuration (組態) 索引標籤。

在 Master Credentials ARN (主要憑證 ARN) 中，您可以檢視機密 ARN。

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups
Cluster				
Configuration		Instance class		Storage
DB cluster ID database-2		Instance class db.m5d.large		Encrypti Enabled
DB cluster role Multi-AZ DB cluster		vCPU 2		AWS KM aws/rds
Engine version 8.0.30		RAM 8 GB		Storage Provision
Amazon Resource Name (ARN) arn:aws:rds:ap-south-1: [redacted]:cluster:database-2		Instance Store Info 75 GB		Storage 400 GiB
Resource ID cluster-[redacted]		Availability		Provision 3000 IO
Created time December 20, 2022, 09:08 (UTC-08:00)		Master username admin		Storage -
Parameter group default.mysql8.0		IAM DB authentication Not enabled		Storage Disabled
Deletion protection Enabled		Multi-AZ 3 Zones		
		<div style="border: 2px solid red; padding: 5px;"> Master Credentials ARN arn:aws:secretsmanager:ap-south-1:[redacted]:secret:rds!cluster-701e5459-f820-4a7f-abae-5427f13037af-f8c17f Manage in Secrets Manager </div>		

您可以遵循 [Manage in Secrets Manager](#) (在 Secrets Manager 中管理) 連結，在 Secret Manager 主控台中檢視和管理機密。

AWS CLI

您可以使用 RDS 命 AWS CLI [describe-db-clusters](#) 令，在密碼管理員中尋找由 RDS 所管 Secrets Manager 碼的下列相關資訊：

- SecretArn – 機密的 ARN
- SecretStatus – 機密的狀態

可能的狀態值包括下列項目：

- creating – 正在建立機密。
- active – 機密可供正常使用和輪換。
- rotating – 正在輪換機密。
- impaired – 機密可以用來存取資料庫憑證，但無法將其輪換。例如，如果變更許可，以便 RDS 再也無法取機密或秘密的 KMS 金鑰，則秘密可能會具有此狀態。

當密碼具有此狀態時，您可以更正導致狀態的條件。如果您更正了導致狀態的條件，則狀態仍會保留 `impaired`，直到下一次輪換為止。或者，您可以修改資料庫叢集，以關閉資料庫憑證的自動管理，然後再次修改資料庫叢集，以開啟資料庫憑證的自動管理。若要修改資料庫叢集，請使用 [modify-db-cluster](#) 指令中的 `--manage-master-user-password` 選項。

- KmsKeyId – 用來加密機密之 KMS 金鑰的 ARN

指定 `--db-cluster-identifier` 選項來顯示特定資料庫叢集的輸出。此範例顯示資料庫叢集所使用之機密的輸出。

Example

```
aws rds describe-db-clusters --db-cluster-identifier mydbcluster
```

下列範例顯示機密的輸出：

```
"MasterUserSecret": {
    "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
    "SecretStatus": "active",
    "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
}
```

當您擁有秘密 ARN 時，您可以使用秘[get-secret-value](#)密管理員 CLI 命令來檢視有關密碼的詳細資料。

此範例顯示前一個範例輸出中機密的詳細資訊。

Example

對於LinuxmacOS、或Unix：

```
aws secretsmanager get-secret-value \  
  --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!  
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

在 Windows 中：

```
aws secretsmanager get-secret-value ^  
  --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!  
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

RDS API

您可以檢視 Secrets Manager 中由 RDS 管理之機密的 ARN、狀態和 KMS 金鑰，方法為使用 [DescribeDBClusters](#) RDS 操作，並將 `DBClusterIdentifier` 參數設為資料庫叢集識別符。機密的詳細資訊包含在輸出中。

當您擁有秘密 ARN 時，您可以使用「秘[GetSecretValue](#)密管理員」作業來檢視有關密碼的詳細資料。

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 Secrets Manager 與 Amazon RDS 整合之版本和區域可用性的詳細資訊，請參閱 [Secrets Manager 與 Amazon RDS 整合的支援區域和資料庫引擎](#)。

Amazon RDS 中的資料保護

AWS [共同的責任模型](#)適用於 Amazon Relational Database Service 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的更多相關資訊，請參閱[資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 憑證，並設定個人使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。

- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name (名稱) 欄位。這包括當您使用 Amazon RDS 或其他使用主控台、API、AWS CLI 或 AWS SDK 的 AWS 服務時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

主題

- [使用加密保護資料](#)
- [網際網路流量隱私權](#)

使用加密保護資料

您可以為資料庫資源啟用加密。您也可以加密對資料庫執行個體的連線。

主題

- [加密 Amazon RDS 資源](#)
- [AWS KMS key 管理](#)
-
- [輪換您的 SSL/TLS 憑證](#)

加密 Amazon RDS 資源

Amazon RDS 可加密您的 Amazon RDS 資料庫執行個體。經過加密的待用資料包含資料庫執行個體的基礎儲存體、自動化備份、僅供讀取複本和快照。

Amazon RDS 加密資料庫執行個體會使用業界標準 AES-256 加密演算法，來加密託管 Amazon RDS 資料庫執行個體伺服器上的資料。資料加密後，Amazon RDS 將以透明的方式處理存取身分驗證和資料解密，同時將對效能的影響降至最小。您不需要修改資料庫用戶端應用程式即可使用加密。

Note

對於加密和未加密的資料庫執行個體，即使跨區域進行複寫，在來源和僅供讀取複本之間傳輸中的資料也會加密。AWS

主題

- [加密 Amazon RDS 資源概觀](#)
- [加密資料庫執行個體](#)
- [決定是否為資料庫執行個體開啟加密](#)
- [Amazon RDS 加密的可用性](#)
- [傳輸中加密](#)
- [Amazon RDS 加密資料庫執行個體的限制](#)

加密 Amazon RDS 資源概觀

Amazon RDS 加密資料庫執行個體可以保護您的資料免於發生未經授權的基礎儲存體存取，為資料提供另一層保護。您可以使用 Amazon RDS 加密提高部署於雲端中應用程式的資料保護，以及滿足靜態加密的合規要求。

針對 Amazon RDS 加密資料庫執行個體，所有日誌、備份和快照都會經過加密。Amazon RDS 使用 AWS Key Management Service 金鑰來加密這些資源。如需 KMS 金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》和 [AWS KMS key 管理](#) 中的 [AWS KMS keys](#)。如果您複製加密的快照，您可以使用不同的 KMS 金鑰來加密目標快照，而不是使用用來加密來源快照的 KMS 金鑰。

當 Amazon RDS 加密執行個體的僅供讀取複本位於相同 AWS 區域時，必須使用與主資料庫執行個體相同的 KMS 金鑰加密。如果主資料庫執行個體和僅供讀取複本位於不同的 AWS 區域，您可以使用該 AWS 區域的 KMS 金鑰加密僅供讀取複本。

您可以使用 AWS 受管金鑰，也可以建立客戶管理的金鑰。若要管理用於加密和解密 Amazon RDS 資源的客戶受管金鑰，請使用 [AWS Key Management Service \(AWS KMS\)](#)。AWS KMS 結合了安全且高度可用的軟硬體，可提供針對雲端擴展的金鑰管理系統。您可以使用建立客戶受管金鑰 AWS KMS，並定義政策，以控制這些客戶受管金鑰的使用方式。AWS KMS 支援 CloudTrail，因此您可以稽核 KMS 金鑰使用情況，以確認客戶受管金鑰是否正確使用。您可以將客戶受管金鑰與 Amazon Aurora 和支援的 AWS 服務 (例如 Amazon S3、Amazon EBS 和亞 Amazon Redshift) 搭配使用。如需與之整合的服務清單 AWS KMS，請參閱 [AWS 服務整合](#)。

Amazon RDS 也支援使用透明資料加密 (TDE)，來加密 Oracle 或 SQL Server 資料庫執行個體。TDE 可與 RDS 靜態加密搭配使用，但是同時使用 TDE 與 RDS 靜態加密可能會稍微影響資料庫的效能。您必須管理每種加密方法的不同索引鍵。如需 TDE 的詳細資訊，請參閱 [Oracle 透明資料加密](#) 或 [支援 SQL Server 的透明資料加密](#)。

加密資料庫執行個體

若要加密新資料庫執行個體，請在 Amazon RDS 主控台上選擇 Enable encryption (啟用加密)。如需建立資料庫執行個體的資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

如果您使用 [create-db-instance](#) AWS CLI 命令建立加密的資料庫執行個體，請設定參數。--storage-encrypted 若您使用 [CreateDBInstance](#) API 操作，請將 StorageEncrypted 參數設為 true。

建立加密的資料庫執行個體時，您可以選擇客戶受管金鑰或 AWS 受管金鑰，讓 Amazon RDS 加密資料庫執行個體。如果您沒有為客戶受管金鑰指定金鑰識別碼，Amazon RDS 會 AWS 受管金鑰 為您的新資料庫執行個體使用。Amazon RDS AWS 受管金鑰 為您的 AWS 帳戶創建一個 Amazon RDS。每個 AWS 區域的 Amazon RDS AWS 帳戶都有不同 AWS 受管金鑰 的帳戶。

如需 KMS 金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [AWS KMS keys](#)。

一旦建立了加密資料庫執行個體之後，您就無法變更該資料庫執行個體使用的 KMS 金鑰。因此，請務必在建立加密的資料庫執行個體前，先決定您的 KMS 金鑰要求。

如果您使用 AWS CLI create-db-instance 命令建立具有客戶管理金鑰的加密資料庫執行個體，請將 --kms-key-id 參數設定為 KMS 金鑰的任何金鑰識別碼。如果您使用 Amazon RDS API CreateDBInstance 作業，請將 KmsKeyId 參數設定為 KMS 金鑰的任何金鑰識別碼。若要在不同的 AWS 帳戶中使用客戶受管金鑰，請指定金鑰 ARN 或別名 ARN。

Important

當您停用 KMS 金鑰時，Amazon RDS 可能會失去資料庫執行個體之 KMS 金鑰的存取權。在這些情況下，加密的資料庫執行個體很快就會進入 inaccessible-encryption-credentials-recoverable 狀態。資料庫執行個體會維持此狀態七天，執行個體會在此期間停止。在此期間對資料庫執行個體進行的 API 呼叫可能無法成功。若要復原資料庫執行個體，請啟用 KMS 金鑰並重新啟動此資料庫執行個體。從啟用 KMS 金鑰 AWS Management Console。使用 AWS CLI 指令啟動 DB 執行個體或重新 [啟動資料庫執行個體](#)。AWS Management Console

如果資料庫執行個體未在七天內復原，則會進入終止 `inaccessible-encryption-credentials` 狀態。在此狀態下，資料庫執行個體無法再使用，您只能從備份還原資料庫執行個體。強烈建議您一律開啟加密資料庫執行個體的備份，以免遺失資料庫內的加密資料。在建立資料庫執行個體期間，Amazon RDS 會檢查呼叫主體是否可以存取 KMS 金鑰，並從 KMS 金鑰產生授權，該金鑰在資料庫執行個體的整個生命週期內使用。撤銷呼叫主體對 KMS 金鑰的存取權不會影響執行中的資料庫。在跨帳戶案例 (例如將快照複製到其他帳戶) 中使用 KMS 金鑰時，需要與其他帳戶共用 KMS 金鑰。如果您從快照建立資料庫執行個體，但未指定不同的 KMS 金鑰，則新執行個體會使用來源帳戶中的 KMS 金鑰。建立資料庫執行個體後撤銷金鑰的存取權並不會影響執行個體。不過，停用金鑰會影響使用該金鑰加密的所有資料庫執行個體。若要避免發生這種情況，請在快照複製作業期間指定不同的金鑰。

決定是否為資料庫執行個體開啟加密

您可以使用 AWS Management Console、AWS CLI、或 RDS API 來判斷資料庫執行個體是否開啟靜態加密。

主控台

決定是否為資料庫執行個體開啟靜態加密。

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要檢查的資料庫執行個體名稱，以檢視其詳細資訊。
4. 選擇 Configuration (組態) 索引標籤，然後勾選 Storage (儲存) 下的 Encryption (加密) 值。

其會顯示 Enabled (已啟用) 或 Not enabled (未啟用)。

RDS > Databases > postgres-database-1

postgres-database-1 Modify Actions ▾

Summary

DB identifier postgres-database-1	CPU 4.92%	Status Available	Class db.t3.small
Role Primary	Current activity 0.00 sessions	Engine PostgreSQL	Region & AZ us-east-1f

Connectivity & security | Monitoring | Logs & events | **Configuration** | Maintenance & backups | Tags

Instance

Configuration DB instance ID postgres-database-1	Instance class Instance class db.t3.small	Storage Encryption Enabled	Performance Insights Performance Insights enabled Yes
--	---	---	---

AWS CLI

若要使用判斷資料庫執行個體的靜態加密是否開啟 AWS CLI，請使用下列選項呼叫 [describe-db-instance](#) 命令：

- `--db-instance-identifier` – 資料庫執行個體名稱。

下列範例會使用查詢，為 mydb 資料庫執行個體傳回關於靜態加密的 TRUE 或 FALSE。

Example

```
aws rds describe-db-instances --db-instance-identifier mydb --query "*[].[StorageEncrypted:StorageEncrypted]" --output text
```

RDS API

若要使用 Amazon RDS API 來決定是否為資料庫執行個體開啟靜態加密，請呼叫 [DescribeDBInstances](#) 作業，搭配下列參數：

- `DBInstanceIdentifier` – 資料庫執行個體名稱。

Amazon RDS 加密的可用性

所有資料庫引擎和儲存體類型目前都可以使用 Amazon RDS 加密，但 SQL Server Express Edition 除外。

大多數資料庫執行個體類別可以使用 Amazon RDS 加密。下表列出不支援 Amazon RDS 加密的資料庫執行個體類別：

執行個體類型	執行個體類別
一般用途 (M1)	db.m1.small
	db.m1.medium
	db.m1.large
	db.m1.xlarge
記憶體最佳化 (M2)	db.m2.xlarge
	db.m2.2xlarge
	db.m2.4xlarge
爆量 (T2)	db.t2.micro

傳輸中加密

AWS 在所有類型的資料庫執行個體之間提供安全和私有連線。此外，某些執行個體類型使用基礎 Nitro System 硬體的卸載功能，以自動加密執行個體之間的傳輸中流量。此加密機制使用帶有關聯資料的認證加密 (AEAD) 演算法 (採用 256 位元加密)。這對網路效能沒有影響。若要支援執行個體之間額外的傳輸中流量加密，必須符合下列要求：

- 執行個體使用下列執行個體類型：
 - 一般用途：M6i、M6 英寸、M6 英寸、M7 克
 - 最佳化記憶體：6 英寸、6 英寸、6 英寸、R6 域名、R7 克、X2 IDN、X2 內嵌
- 實例是相同的 AWS 區域。
- 這些執行個體位於相同的 VPC 或對等 VPC 中，且流量不會經過虛擬網路裝置或服務，例如負載平衡器或傳輸閘道。

Amazon RDS 加密資料庫執行個體的限制

Amazon RDS 加密資料庫執行個體具有下列限制：

- 您只能在加密 Amazon RDS 資料庫執行個體時，而不能在建立資料庫執行個體之後為其啟用加密。

不過，因為您可以加密未加密快照的副本，所以可以有效地將加密新增至未加密的資料庫執行個體。亦即，您可以建立資料庫執行個體的快照，然後建立該快照的加密副本。接著，從加密快照中還原資料庫執行個體，因此您有原始資料庫執行個體的加密副本。如需詳細資訊，請參閱 [複製資料庫快照](#)。
- 您無法在加密的資料庫執行個體上關閉加密。
- 您無法建立未加密資料庫執行個體的加密快照。
- 加密資料庫執行個體的快照必須使用與資料庫執行個體相同的 KMS 金鑰進行加密。
- 未加密資料庫執行個體不可以有加密僅供讀取複本，加密資料庫執行個體也不可以有未加密僅供讀取複本。
- 當加密僅供讀取複本位於相同 AWS 區域時，必須使用與來源資料庫執行個體相同的 KMS 金鑰加密。
- 您無法將未加密的備份或快照還原至已加密的資料庫執行個體。
- 若要將加密快照從一個 AWS 區域複製到另一個區域，您必須在目的地 AWS 區域中指定 KMS 金鑰。這是因為 KMS 金鑰專屬於其建立所在的 AWS 區域。

在整個複製過程中來源快照仍會保持加密狀態。Amazon RDS 會在複製過程中使用信封加密來保護資料。如需信封加密的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [信封加密](#)。

- 您無法解密加密的資料庫執行個體。但是，您可以從加密的資料庫執行個體匯出資料，然後將資料匯入未加密的資料庫執行個體。

AWS KMS key 管理

Amazon RDS 會自動與 [AWS Key Management Service \(AWS KMS\)](#) 整合以進行金鑰管理。Amazon RDS 會使用信封加密。如需信封加密的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [信封加密](#)。

您可以使用兩種 AWS KMS 金鑰來加密資料庫執行個體。

- 如果想要完全控制 KMS 金鑰，您必須建立客戶受管金鑰。如需客戶受管金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [客戶受管金鑰](#)。

如果快照使用共用快照的 AWS 帳戶的 AWS 受管金鑰所加密，則您無法共用此快照。

- AWS 受管金鑰 是您帳戶中的 KMS 金鑰，由已與 AWS KMS 整合的 AWS 服務代表您建立、管理和使用。根據預設，RDS AWS 受管金鑰 (aws/rds) 用於加密。您無法管理、輪換或刪除 RDS AWS 受管金鑰。如需 AWS 受管金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [AWS 受管金鑰](#)。

在 [AWS KMS 主控台](#)、AWS CLI 或 AWS KMS API 中使用 [AWS Key Management Service \(AWS KMS\)](#) 管理用於 Amazon RDS 加密資料庫執行個體的 KMS 金鑰。若要檢視透過 AWS 受管或客戶受管金鑰採取的每項動作稽核日誌，請使用 [AWS CloudTrail](#)。如需有關金鑰轉換的詳細資訊，請參閱 [轉換 AWS KMS 金鑰](#)。

Important

如果您關閉或撤銷 RDS 資料庫所使用之 KMS 金鑰的許可，則 RDS 會在需要存取 KMS 金鑰時，將您的資料庫置於終端機狀態。此變更可以是立即或延遲，視需要存取 KMS 金鑰的使用案例而定。在此情況下，您將再也無法使用資料庫執行個體，而且無法復原資料庫的目前狀態。若要還原資料庫執行個體，您必須重新啟用對 RDS 的 KMS 金鑰存取權，然後從最新可用的備份中還原資料庫執行個體。

授權使用客戶受管金鑰

當 RDS 在密碼編譯操作中使用客戶受管金鑰時，它會代表建立或變更 RDS 資源的使用者。

若要使用客戶受管金鑰建立 RDS 資源，使用者必須具備呼叫客戶受管金鑰下列作業的許可：

- kms:CreateGrant
- kms:DescribeKey

您可以在金鑰政策或在 IAM 政策中指定這些必要的許可 (如果金鑰政策允許)。

您可以透過各種方式使 IAM 政策更加嚴格。例如，若要允許客戶受管金鑰僅用於源自 RDS 的請求，您可以搭配 `rds.<region>.amazonaws.com` 值使用 [kms:ViaService 條件索引鍵](#)。您也可以使用 [Amazon RDS 加密內容](#) 中的金鑰或值作為條件，以便將客戶受管金鑰用於加密。

如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [允許其他帳戶使用者使用 KMS 金鑰](#) 和 [AWS KMS 中的金鑰政策](#)。

Amazon RDS 加密內容

當 RDS 使用您的 KMS 金鑰時，或者當 Amazon EBS 代表 RDS 使用 KMS 金鑰時，服務會指定[加密內容](#)。加密內容是[額外驗證資料 \(AAD\)](#)，供 AWS KMS 用來確保資料完整性。為加密操作指定加密內容時，此服務必須為解密操作指定相同的加密內容。否則，解密會失敗。加密內容也會寫入 [AWS CloudTrail](#) 日誌檔，以協助您了解為何使用指定的 KMS 金鑰。您的 CloudTrail 日誌可能包含多個項目來描述 KMS 金鑰使用情形，但每個日誌項目中的加密內容可協助您判斷該特定使用情形的原因。

至少，Amazon RDS 一律使用資料庫執行個體 ID 做為加密內容，如下列 JSON 格式範例：

```
{ "aws:rds:db-id": "db-CQYSMDPBRZ7BPMH7Y3RTDG5QY" }
```

這個加密內容可協助您識別您的 KMS 金鑰所用的資料庫執行個體。

當您的 KMS 金鑰用於特定的資料庫執行個體和特定的 Amazon EBS 磁碟區，資料庫執行個體 ID 和 Amazon EBS 磁碟區 ID 都會用於加密內容，如下列 JSON 格式範例：

```
{
  "aws:rds:db-id": "db-BRG7VYS3SVIFQW7234EJQ0M5RQ",
  "aws:ebs:id": "vol-ad8c6542"
}
```

您可以使用應用程式的安全通訊端層 (SSL) 或傳輸層安全性 (TLS) 來加密與執行 Db2、MariaDB、Microsoft SQL 伺服器、MySQL、甲骨文或 PostgreSQL 之資料庫的連線。

或者，您的 SSL/TLS 連線可以透過驗證資料庫上安裝的伺服器憑證來執行伺服器身分驗證。若需要伺服器身分驗證，請依照此一般程序進行：

1. 為您的資料庫選擇簽署資料庫伺服器憑證的憑證認證機構 (CA)。如需有關憑證認證機構的詳細資訊，請參閱 [憑證授權單位](#)。
2. 下載憑證套件，以便在連線至資料庫時使用。若要下載憑證套件，請參閱 [所有人的憑證套件 AWS 區域](#) 和 [特定的憑證組合 AWS 區域](#)。

Note

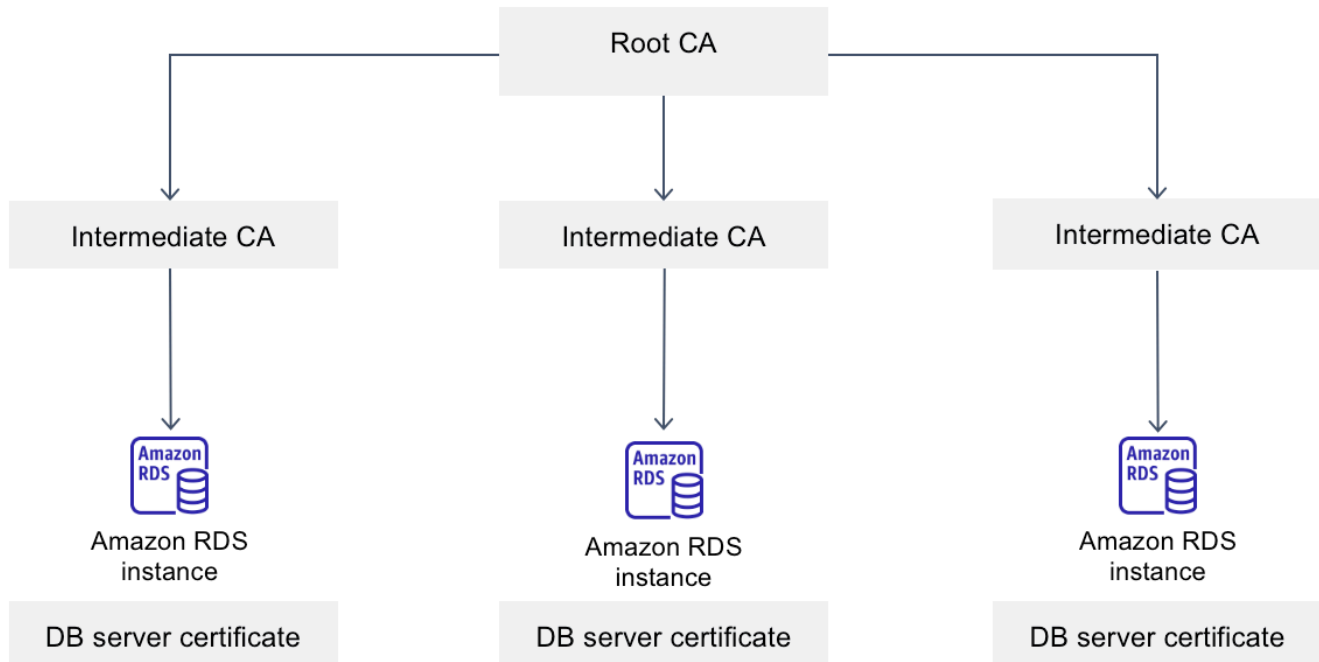
所有憑證都只能使用 SSL/TLS 連線來下載。

3. 使用資料庫引擎實作 SSL/TLS 連線的程序連線至資料庫。每個資料庫引擎都有自己實作 SSL/TLS 的程序。若要了解如何為您的資料庫實作 SSL/TLS，請瀏覽對應您的資料庫引擎的連結：

- [將 SSL/TLS 與 Amazon RDS 用於 Db2 資料庫執行個體搭配使用](#)
- [將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用](#)
- [對 Microsoft SQL Server 資料庫執行個體使用 SSL](#)
- [將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)
- [使用 SSL 搭配 RDS for Oracle 資料庫執行個體](#)
- [將 SSL 與 PostgreSQL 資料庫執行個體搭配使用](#)

憑證授權單位

憑證認證機構 (CA) 是在憑證鏈頂端識別根 CA 的憑證。CA 會簽署安裝在每個資料庫執行個體上的資料庫執行個體憑證。資料庫伺服器憑證會將資料庫執行個體識別為受信任伺服器。



Amazon RDS 提供下列 CA 來簽署資料庫的資料庫伺服器憑證。

憑證授權單位 (CA)	描述
rds-ca-2019	使用搭配 RSA 2048 私密金鑰演算法和 SHA256 簽署演算法的憑證認證機構。此 CA 將於 2024 年到期，且不支援自動伺服器憑證輪換。如果您正在使用此 CA 並

憑證授權單位 (CA)	描述
	且想要保持相同的標準，我們建議您切換至 rds-ca-rsa 2048 g1 CA。
rds-ca-rsa国际集团	<p>在大多數 AWS 區域中，使用搭配 RSA 2048 私密金鑰演算法和 SHA256 簽署演算法的憑證認證機構。</p> <p>在中 AWS GovCloud (US) Regions，此 CA 使用具有 RSA 2048 私密金鑰演算法和 SHA384 簽署演算法的憑證授權單位。</p> <p>此 CA 的有效期限仍超過 rds-ca-2019 CA。此 CA 支援自動伺服器憑證輪換。</p>
rds-ca-rsa国际集团	使用搭配 RSA 4096 私密金鑰演算法和 SHA384 簽署演算法的憑證認證機構。此 CA 支援自動伺服器憑證輪換。
rds-ca-ecc384 国际	使用搭配 ECC 384 私密金鑰演算法和 SHA384 簽署演算法的憑證認證機構。此 CA 支援自動伺服器憑證輪換。

Note

[如果您使用的是 AWS CLI，您可以使用描述憑證來查看上面列出的憑證授權單位的有效性。](#)

這些 CA 憑證包含在區域和全域憑證套件中。當您將 rds-ca-rsa 2048-g1、rds-ca-rsa 4096-g1 或 rds-ca-ecc 384-g1 CA 與資料庫搭配使用時，RDS 會管理資料庫上的資料庫伺服器憑證。RDS 會在資料庫伺服器憑證到期之前自動進行輪換。

設定資料庫的 CA

執行下列任務時，您可以設定資料庫的 CA：

- 建立資料庫執行個體或異地同步備份資料庫叢集 — 您可以在建立資料庫執行個體或叢集時設定 CA。如需相關指示，請參閱 [the section called “建立資料庫執行個體”](#) 或 [the section called “建立多可用區域資料庫叢集”](#)。

- 修改資料庫執行個體或異地同步備份資料庫叢集 — 您可以透過修改資料庫執行個體或叢集來設定 CA。如需相關指示，請參閱 [the section called “修改資料庫執行個體”](#) 或 [the section called “修改多可用區域資料庫叢集”](#)。

Note

預設的 CA 會設定為 rds-ca-rsa 2048-g1。您可以使用 [修改憑證](#) 命令覆寫您的 AWS 帳戶 預設 CA。

可用的 CA 取決於資料庫引擎和資料庫引擎版本。使用 AWS Management Console 時，您可以使用 Certificate authority (憑證授權單位) 設定來選擇 CA，如下圖所示。

Certificate authority - optional [Info](#)

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)

Expiry: May 24, 2061

If you don't select a certificate authority, RDS chooses one for you.

主控台只會顯示資料庫引擎和資料庫引擎版本可用的 CA。如果您使用的是 AWS CLI，可以使用 [create-db-instance](#) 或 [modify-db-instance](#) 命令為資料庫執行個體設定 CA。您可以使用 [create-db-cluster](#) 或 [modify-db-cluster](#) 命令為異地同步備份資料庫叢集設定 CA。

如果您使用的是 AWS CLI，您可以使用 [描述憑證](#) 命令來查看您帳戶的可用 CA。此命令也會在輸出中顯示 ValidTill 中每個 CA 的到期日。您可以使用該 [describe-db-engine-versions](#) 命令找到可用於特定數據庫引擎和數據庫引擎版本的 CA。

下列範例顯示適用於預設 RDS for PostgreSQL 資料庫引擎版本的 CA。

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

輸出類似如下。可用的 CA 列示在 SupportedCACertificateIdentifiers 中。輸出也會顯示 DB 引擎版本是否支援輪換憑證，而不需在 SupportsCertificateRotationWithoutRestart 中重新啟動。

```
{
  "DBEngineVersions": [
```

```
{
  "Engine": "postgres",
  "MajorEngineVersion": "13",
  "EngineVersion": "13.4",
  "DBParameterGroupFamily": "postgres13",
  "DBEngineDescription": "PostgreSQL",
  "DBEngineVersionDescription": "PostgreSQL 13.4-R1",
  "ValidUpgradeTarget": [],
  "SupportsLogExportsToCloudwatchLogs": false,
  "SupportsReadReplica": true,
  "SupportedFeatureNames": [
    "Lambda"
  ],
  "Status": "available",
  "SupportsParallelQuery": false,
  "SupportsGlobalDatabases": false,
  "SupportsBabelfish": false,
  "SupportsCertificateRotationWithoutRestart": true,
  "SupportedCACertificateIdentifiers": [
    "rds-ca-2019",
    "rds-ca-rsa2048-g1",
    "rds-ca-ecc384-g1",
    "rds-ca-rsa4096-g1"
  ]
}
```

資料庫伺服器憑證有效期

資料庫伺服器憑證的有效期取決於資料庫引擎和資料庫引擎版本。如果資料庫引擎版本支援在不重新啟動的情況下輪換憑證，則資料庫伺服器憑證的有效期為 1 年。若不支援的話，則有效期為 3 年。

如需資料庫伺服器憑證輪換的詳細資訊，請參閱 [自動伺服器憑證輪換](#)。

檢視資料庫執行個體的 CA

您可以檢視主控台中的 [連線與安全性] 索引標籤，檢視資料庫 CA 的詳細資料，如下圖所示。

The screenshot displays the 'Connectivity & security' configuration page for an Amazon RDS instance. The page is divided into three main sections: Endpoint & port, Networking, and Security. The 'Certificate authority' section is highlighted with a red box, showing the following details:

Section	Property	Value
Security	VPC security groups	default (sg-062c8f43392f87f49) Active
	Publicly accessible	No
	Certificate authority	rds-ca-2019
	Certificate authority date	August 22, 2024, 19:08 (UTC+02:00)
	DB instance certificate expiration date	August 22, 2024, 19:08 (UTC+02:00)

如果您使用的是 AWS CLI，您可以使用 [describe-db-instances](#) 命令檢視資料庫執行個體 CA 的詳細資料。您可以使用 [describe-db-clusters](#) 指令檢視異地同步備份資料庫叢集的 CA 詳細資料。

若要查看 CA 憑證套件的內容，請使用下列命令：

```
keytool -printcert -v -file global-bundle.pem
```

所有人的憑證套件 AWS 區域

要獲得所有人的證書包 AWS 區域，請從 <https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem> 下載它。

套裝軟體同時包含中 rds-ca-2019 繼憑證和根憑證。套裝軟體也包含 rds-ca-rsa2048-g1、rds-ca-rsa4096-g1、和 rds-ca-ecc384-g1 根 CA 憑證。您的應用程式信任存放區只需要註冊根 CA 憑證。

如果您的應用程式是在 Microsoft 視窗上，並且需要 PKCS7 檔案，您可以從以下位置下載 PKCS7 憑證套件：<https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b>。

Note

Amazon RDS 代理來自 AWS Certificate Manager (ACM) 的證書。如果您使用的是 RDS 代理伺服器，則不需要下載 Amazon RDS 憑證或更新使用 RDS 代理連線的應用程式。如需詳細資訊，請參閱 [搭配 RDS Proxy 使用 TLS/SSL](#)。

特定的憑證組合 AWS 區域

套裝軟體同時包含中rds-ca-2019繼憑證和根憑證。套裝軟體也包含rds-ca-rsa2048-g1rds-ca-rsa4096-g1、和rds-ca-ecc384-g1根 CA 憑證。您的應用程式信任存放區只需要註冊根 CA 憑證。

若要取得的憑證套裝軟體 AWS 區域，請從下表 AWS 區域 中的連結下載。

AWS 區域	憑證套件 (PEM)	憑證套件 (PKCS7)
美國東部 (維吉尼亞北部)	us-east-1-bundle.pem	us-east-1-bundle.p7b
美國東部 (俄亥俄)	us-east-2-bundle.pem	us-east-2-bundle.p7b
美國西部 (加利佛尼亞北部)	us-west-1-bundle.pem	us-west-1-bundle.p7b
美國西部 (奧勒岡)	us-west-2-bundle.pem	us-west-2-bundle.p7b
非洲 (開普敦)	af-south-1-bundle.pem	af-south-1-bundle.p7b
亞太區域 (香港)	ap-east-1-bundle.pem	ap-east-1-bundle.p7b
亞太區域 (海德拉巴)	ap-south-2-bundle.pem	ap-south-2-bundle.p7b
亞太區域 (雅加達)	ap-southeast-3-bundle.pem	ap-southeast-3-bundle.p7b
亞太區域 (墨爾本)	ap-southeast-4-bundle.pem	ap-southeast-4-bundle.p7b
亞太區域 (孟買)	ap-south-1-bundle.pem	ap-south-1-bundle.p7b
亞太區域 (大阪)	ap-northeast-3-bundle.pem	ap-northeast-3-bundle.p7b
亞太區域 (東京)	ap-northeast-1-bundle.pem	ap-northeast-1-bundle.p7b
亞太區域 (首爾)	ap-northeast-2-bundle.pem	ap-northeast-2-bundle.p7b
亞太區域 (新加坡)	ap-southeast-1-bundle.pem	ap-southeast-1-bundle.p7b
亞太區域 (雪梨)	ap-southeast-2-bundle.pem	ap-southeast-2-bundle.p7b
加拿大 (中部)	ca-central-1-bundle.pem	ca-central-1-bundle.p7b

AWS 區域	憑證套件 (PEM)	憑證套件 (PKCS7)
加拿大西部 (卡加利)	鈣-西 1 捆綁 .pem	鈣-西 1 捆綁 .p7b
歐洲 (法蘭克福)	eu-central-1-bundle.pem	eu-central-1-bundle.p7b
歐洲 (愛爾蘭)	eu-west-1-bundle.pem	eu-west-1-bundle.p7b
歐洲 (倫敦)	eu-west-2-bundle.pem	eu-west-2-bundle.p7b
歐洲 (米蘭)	eu-south-1-bundle.pem	eu-south-1-bundle.p7b
歐洲 (巴黎)	eu-west-3-bundle.pem	eu-west-3-bundle.p7b
歐洲 (西班牙)	eu-south-2-bundle.pem	eu-south-2-bundle.p7b
歐洲 (斯德哥爾摩)	eu-north-1-bundle.pem	eu-north-1-bundle.p7b
歐洲 (蘇黎世)	eu-central-2-bundle.pem	eu-central-2-bundle.p7b
以色列 (特拉維夫)	il-central-1-bundle.pem	il-central-1-bundle.p7b
Middle East (Bahrain)	me-south-1-bundle.pem	me-south-1-bundle.p7b
中東 (阿拉伯聯合大公國)	me-central-1-bundle.pem	me-central-1-bundle.p7b
南美洲 (聖保羅)	sa-east-1-bundle.pem	sa-east-1-bundle.p7b

AWS GovCloud (US) 憑證

若要取得同時包含中繼憑證和根憑證的憑證組合，請從 <https://truststore.pki.us-gov-west-1.rds.amazonaws.com/> 下載。AWS GovCloud (US) Region [us-gov-west-1.rds.amazonaws.com](#) /全球/全球包裝.pem。

如果您的應用程式是在 Microsoft 視窗上，並且需要 PKCS7 檔案，您可以從以下位置下載 PKCS7 憑證套件：<https://truststore.pki.us-gov-west-1.rds.amazonaws.com/> 全球/全局組合.p7b。

套裝軟體同時包含中繼憑證和根憑證。套裝軟體也包含 rds-ca-rsa2048-g1、rds-ca-rsa4096-g1、和 rds-ca-ecc384-g1 根 CA 憑證。您的應用程式信任存放區只需要註冊根 CA 憑證。

若要取得的憑證組合 AWS GovCloud (US) Region，請從下表 AWS GovCloud (US) Region 中的連結下載。

AWS GovCloud (US) Region	憑證套件 (PEM)	憑證套件 (PKCS7)
AWS GovCloud (美國東部)	us-gov-east-1 捆綁。PEM	us-gov-east-1 捆綁式
AWS GovCloud (美國西部)	us-gov-west-1 捆綁。PEM	us-gov-west-1 捆綁式

輪換您的 SSL/TLS 憑證

Amazon RDS 憑證授權機構憑證 rds-ca-2019 設定為 2024 年 8 月到期。如果您使用或計劃使用安全通訊端層 (SSL) 或傳輸層安全性 (TLS) 搭配憑證驗證來連線到 RDS 資料庫執行個體或異地同步備份資料庫叢集，請考慮使用其中一個新的 CA 憑證 rds-ca-rsa 2048-g1、4096 g1 或 384-g1。rds-ca-rsa rds-ca-ecc 如果您目前未將 SSL/TLS 與憑證驗證搭配使用，則可能仍有過期的 CA 憑證，而且如果您計劃將 SSL/TLS 與憑證驗證搭配使用以連線至 RDS 資料庫，則必須將其更新為新的 CA 憑證。

請依照下列指示完成更新。在更新資料庫執行個體或異地同步備份資料庫叢集以使用新的 CA 憑證之前，請確定已更新連線至 RDS 資料庫的用戶端或應用程式。

Amazon RDS 提供新的 CA 憑證做為安 AWS 全性最佳實務。如需有關新憑證和支援 AWS 區域的資訊，請參閱。

Note

Amazon RDS 代理來自 AWS Certificate Manager (ACM) 的證書。如果您使用的是 RDS 代理伺服器，當您輪替 SSL/TLS 憑證時，不需要更新使用 RDS 代理連線的應用程式。如需詳細資訊，請參閱 [搭配 RDS Proxy 使用 TLS/SSL](#)。

Note

如果您在 2020 年 7 月 28 日之前建立或更新為 rds-ca-2019 憑證的資料庫執行個體或異地同步備份資料庫叢集使用 Go 版 1.15 應用程式，則必須再次更新憑證。使用新的 modify-db-instance CA 憑證識別碼執行資料庫執行個體的 modify-db-cluster 命令，或針對異地同步備份資料庫叢集執行命令。您可以使用 describe-db-engine-versions 命令，尋找適用於特定資料庫引擎和資料庫引擎版本的 CA。

如果您在 2020 年 7 月 28 日之後建立資料庫或更新了憑證，則不需要採取任何動作。如需詳細資訊，請參閱 [Go GitHub 問題 #39568](#)。

主題

- [透過修改資料庫執行個體或叢集來更新 CA 憑證](#)
- [透過套用維護來更新憑證授權機構憑證](#)
- [自動伺服器憑證輪換](#)
- [將憑證匯入信任存放區的範例指令碼](#)

透過修改資料庫執行個體或叢集來更新 CA 憑證

以下範例會將您的 CA 憑證從 rds-ca-2019 更新為 rds-ca-rsa2048-g1。您可以選擇不同的憑證。如需詳細資訊，請參閱[憑證授權單位](#)。

更新您的應用程式信任存放區，以減少與更新 CA 憑證相關的停機時間。如需與 CA 憑證輪替相關聯之重新啟動的詳細資訊，請參閱[自動伺服器憑證輪換](#)。

修改資料庫執行個體或叢集以更新 CA 憑證

1. 請遵循 所述，下載新的 SSL/TLS 憑證。
2. 更新您的應用程式，以使用新的 SSL/TLS 憑證。

更新應用程式 SSL/TLS 憑證的方法取決於您特定的應用程式。與應用程式開發人員合作更新應用程式的 SSL/TLS 憑證。

如需針對每個資料庫引擎檢查 SSL/TLS 連線和更新應用程式的資訊，請參閱下列主題：

- [更新應用程式，以使用新的 SSL/TLS 憑證連線至 MariaDB 執行個體](#)
- [將應用程式更新為使用新的 SSL/TLS 憑證來連線至 Microsoft SQL Server 資料庫執行個體](#)
- [將應用程式更新為使用新的 SSL/TLS 憑證來連線至 MySQL 資料庫執行個體](#)
- [更新應用程式使用新的 SSL/TLS 憑證來連線至 Oracle 資料庫執行個體](#)
- [更新應用程式使用新的 SSL/TLS 憑證來連線至 PostgreSQL 資料庫執行個體](#)

關於為 Linux 作業系統更新信任存放區的範例指令碼，請參閱[將憑證匯入信任存放區的範例指令碼](#)。

Note

憑證套件同時包含舊 CA 和新 CA 的憑證，讓您可以安全地更新應用程式，並在轉換期間維護連線。如果您使用將資料庫移轉 AWS Database Migration Service 至資料庫執行個體或集，建議您使用憑證服務包，以確保移轉期間的連線能力。

3. 修改資料庫執行個體或異地同步備份資料庫叢集，將 CA 從 RDS-ca-2019 變更為 rds-ca-卡-rsa2048-g1。若要檢查資料庫是否需要重新啟動才能更新 CA 憑證，請使用 [describe-db-engine-versions](#) 命令並查看 SupportsCertificateRotationWithoutRestart 旗標。

Important

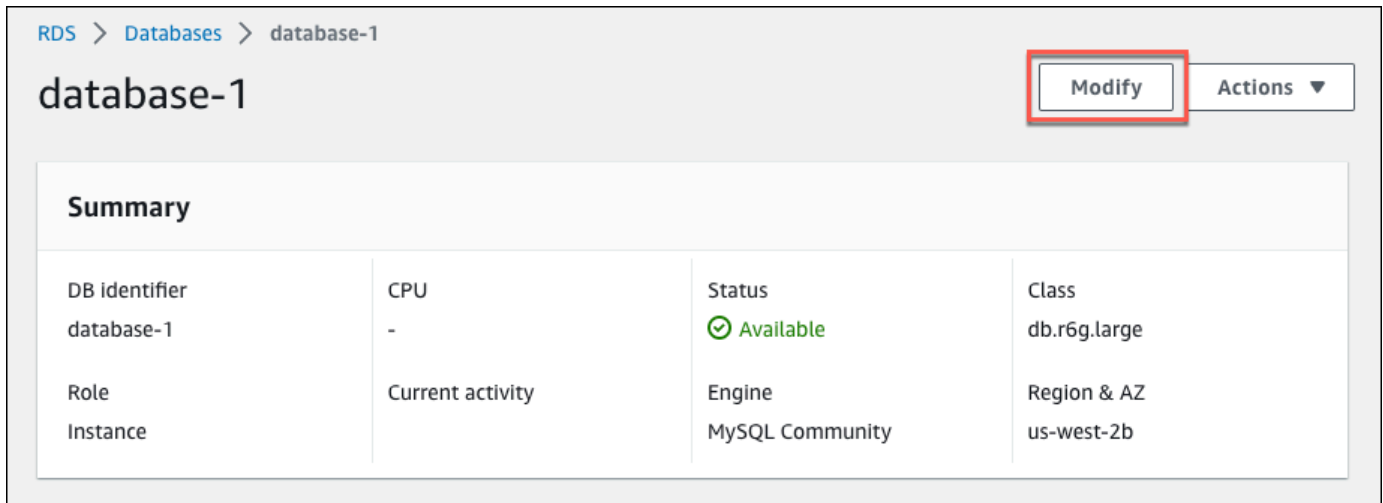
如果您在憑證到期後發生連線問題，請在主控台中指定 Apply immediately (立即套用)，或使用 `--apply-immediately` 來指定 AWS CLI 選項。依預設，此操作排定在下一個維護時段執行。

若要為您的執行個體 CA 設定與預測 RDS CA 不同的覆寫，請使用 [modify-certificates](#) CLI 命令。

對於資料庫執行個體 AWS Management Console 或異地同 AWS CLI 步備份資料庫叢集，您可以使用或將 CA 憑證從 rds-ca-2019 變更為 rds-ca-rsa 2048-g1。

主控台

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在瀏覽窗格中，選擇 [資料庫]，然後選擇要修改的資料庫執行個體或異地同步備份資料庫叢集。
3. 選擇 Modify (修改)。



RDS > Databases > database-1

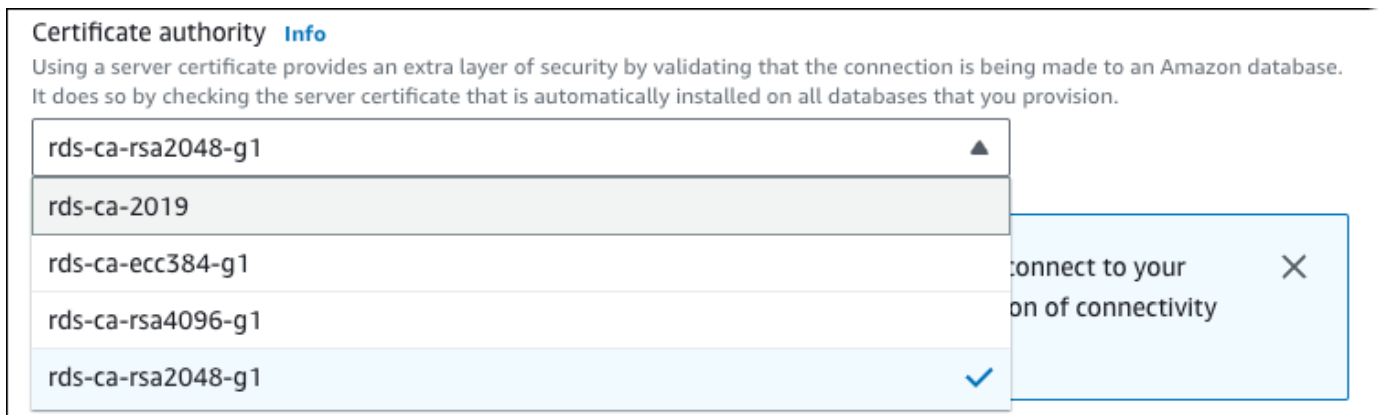
database-1

Modify Actions ▾

Summary

DB identifier database-1	CPU -	Status ✔ Available	Class db.r6g.large
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-west-2b

4. 在連線區段中，選擇 rds-ca-rsa2048-g1。



Certificate authority [Info](#)

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

- rds-ca-rsa2048-g1 ▲
- rds-ca-2019
- rds-ca-ecc384-g1
- rds-ca-rsa4096-g1
- rds-ca-rsa2048-g1 ✓

connect to your
on of connectivity ✕

5. 選擇 Continue (繼續)，並檢查修改的摘要。
6. 若要立即套用變更，請選擇 Apply immediately (立即套用)。
7. 在確認頁面上，檢閱您的變更。如果正確無誤，請選擇 [修改資料庫執行個體] 或 [修改叢集] 以儲存變更。

⚠ Important

排定此操作時，請確定您事先已更新用戶端信任存放區。

或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

AWS CLI

若要針對資料庫執行個體或異地同 AWS CLI 步備份資料庫叢集，使用將 CA 從 `rds-ca-2019` 變更為 `rds-ca-rsa2048-g1`，請呼叫修改的 `db-執行個體` 或修改 `db-叢集` 命令。指定資料庫執行個體或叢集識別碼和選 `--ca-certificate-identifier` 項。

使用 `--apply-immediately` 參數可立即套用更新。依預設，此操作排定在下一個維護時段執行。

Important

排定此操作時，請確定您事先已更新用戶端信任存放區。

Example

資料庫執行個體

下列範例會 `mydbinstance` 透過將 CA 憑證設定為來進行修改 `rds-ca-rsa2048-g1`。

對於 Linux macOS、或 Unix：

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --ca-certificate-identifier rds-ca-rsa2048-g1
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --ca-certificate-identifier rds-ca-rsa2048-g1
```

Note

如果您的執行個體需要重新開機，您可以使用 [修改-db-instance](#) CLI 指令並指定選項。 `--no-certificate-rotation-restart`

Example

異地同步備份 DB 叢集

下列範例會mydbcluster透過將 CA 憑證設定為來進行修改rds-ca-rsa2048-g1。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --ca-certificate-identifier rds-ca-rsa2048-g1
```

在 Windows 中：

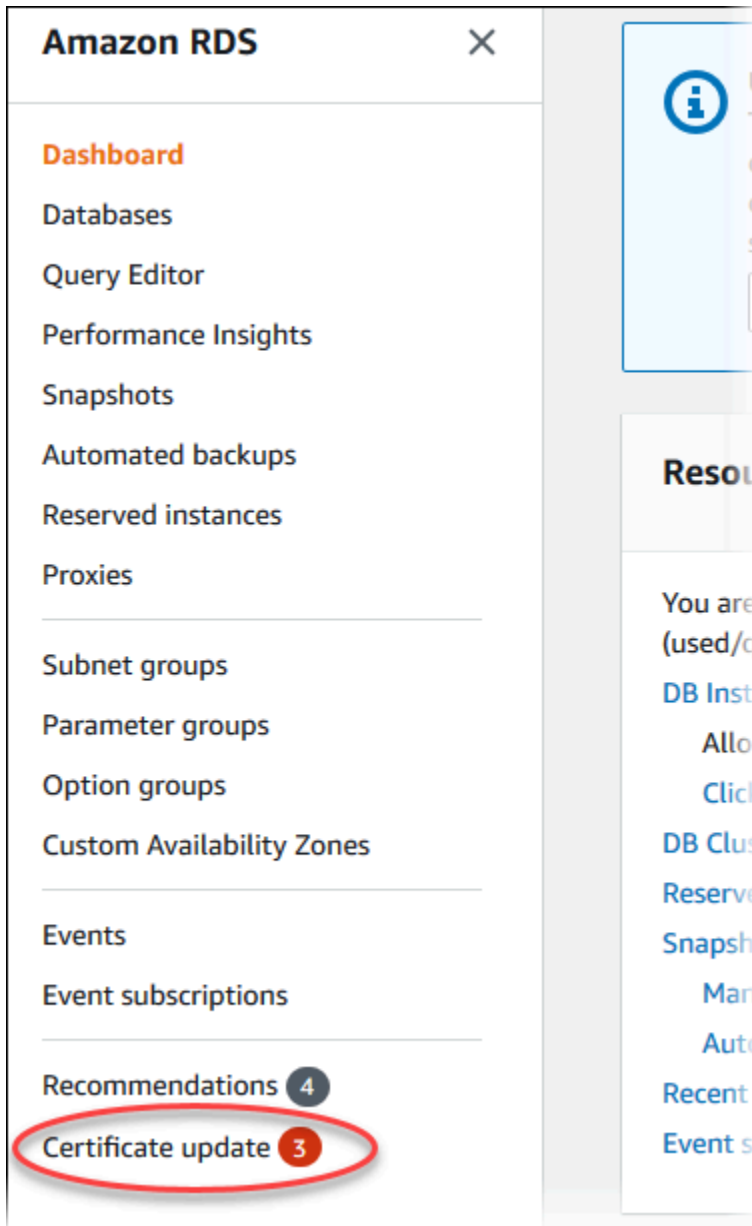
```
aws rds modify-db-cluster ^  
  --db-cluster-identifier mydbcluster ^  
  --ca-certificate-identifier rds-ca-rsa2048-g1
```

透過套用維護來更新憑證授權機構憑證

執行下列步驟，透過套用維護來更新您的 CA 憑證。

套用維護來更新您的 CA 憑證

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在功能窗格中，選擇 [憑證更新]。



需要更新憑證的資料庫頁面隨即顯示。


RDS > Certificate update

Databases requiring certificate update (2) Export list Schedule Apply now

Rotate your CA Certificates before expiry date or risk losing SSL/TLS connectivity to your existing DB instances.

Filter by Databases


	DB identifier ▲	Status ▼	Certificate authority ▼	CA expiration date ▼	Role ▼	Restart Required ▼	Scheduled Changes ▼	Mainten
<input type="radio"/>	database-1	Available	rds-ca-2019	⚠ June 30, 2024, 10:26 (UTC-07:00)	Instance	No	No	March 03
<input type="radio"/>	database-2	Available	rds-ca-2019	⚠ June 30, 2024, 10:26 (UTC-07:00)	Multi-AZ DB cluster	No	No	March 07

 Note

此頁面僅顯示目前的資料庫執行個體和叢集 AWS 區域。如果您有多個資料庫，請查看每個資料庫中的此頁面 AWS 區域，AWS 區域 以查看所有具有舊 SSL/TLS 憑證的資料庫執行個體。

3. 選擇您要更新的資料庫執行個體或異地同步備份資料庫叢集。

您可以選擇排程，以排定下一個維護時段的憑證輪換。您可以選擇立即套用以立即套用輪換。

 Important



如果在憑證到期後發生連線問題，請使用立即套用選項。

4. a. 如果您選擇排程，系統會提示您確認 CA 憑證輪換。此提示也會指出排定的更新時段。

Schedule updating your certificates ✕

Select Certificate Authority (CA)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 ▼
Expiry: May 24, 2061

 **RDS Certificate Authority**
For more information about the certificate, see [RDS Certificate Authority](#) .
Certificate update **does not require restarting your database.**

Click **Schedule** to update your certificate during the next scheduled maintenance window at September 11, 2023 02:17 - 02:47 UTC-7



Cancel **Schedule**

- b. 如果您選擇立即套用，系統會提示您確認 CA 憑證輪換。

Confirm updating your certificates now ✕

Select Certificate Authority (CA)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1
Expiry: May 24, 2061

 **RDS Certificate Authority**
For more information about the certificate, see [RDS Certificate Authority](#) .

Certificate update **does not require restarting your database.**

Click **Confirm** to apply certificate immediately.

Cancel **Confirm**

Important

在資料庫上排定 CA 憑證輪換之前，請先更新使用 SSL/TLS 的任何用戶端應用程式和要連結伺服器憑證。這些更新專屬於您的資料庫引擎。更新這些用戶端應用程式之後，您可以確認 CA 憑證輪換。

若要繼續，請選擇核取方塊，然後選擇 Confirm (確認)。

5. 針對您要更新的每個資料庫執行個體和叢集重複步驟 3 和 4。

自動伺服器憑證輪換

如果您的 CA 支援自動伺服器憑證輪換，RDS 會自動處理資料庫伺服器憑證的輪換。RDS 會使用相同的根 CA 進行此自動輪換，因此您不需要下載新的 CA 套件。請參閱[憑證授權單位](#)。

資料庫伺服器憑證的輪換和有效期取決於資料庫引擎：

- 如果您的資料庫引擎支援在不重新啟動的情況下輪換，則 RDS 會自動輪換資料庫伺服器憑證，而不需要您採取任何動作。RDS 會嘗試在資料庫伺服器憑證有效期過半時，在您偏好的維護時段內輪換您的資料庫伺服器憑證。新的資料庫伺服器憑證有效期為 12 個月。
- 如果您的資料庫引擎不支援在不重新啟動的情況下輪換，則 RDS 會在資料庫伺服器憑證到期之前至少 6 個月，通知您有關維護事件的資訊。新的資料庫伺服器憑證有效期為 36 個月。

使用命 [describe-db-engine-versions](#) 命並檢

查 `SupportsCertificateRotationWithoutRestart` 旗標，以識別 DB 引擎版本是否支援在不重新啟動的情況下旋轉憑證。如需詳細資訊，請參閱 [設定資料庫的 CA](#)。

將憑證匯入信任存放區的範例指令碼

以下是範例 Shell 指令碼，會將憑證套件匯入信任存放區。

每個範例 Shell 指令碼都使用 `keytool`，其是 Java 開發套件 (JDK) 的一部分。如需安裝 JDK 的詳細資訊，請參閱 [《JDK 安裝指南》](#)。

主題

- [在 Linux 上匯入憑證的範例指令碼](#)
- [在 macOS 上匯入憑證的範例指令碼](#)

在 Linux 上匯入憑證的範例指令碼

以下範例 Shell 指令碼將憑證套件匯入 Linux 作業系統上的信任存放區。

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
  ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/ {split_after=1}
{print > "rds-ca-" n+1 ".pem"}' < ${mydir}/global-bundle.pem
```

```

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:;/
s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
-d " " -f3- | while read alias
do
  expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
"${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }`
  echo " Certificate ${alias} expires in '$expiry'"
done

```

在 macOS 上匯入憑證的範例指令碼

以下是範例 Shell 指令碼，會將憑證套件匯入 macOS 上的信任存放區。

```

mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:;/
s/.*(CN=|CN = )//; print')
  echo "Importing $alias"

```



```
keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
${truststore} -noprompt
rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
-d " " -f3- | while read alias
do
    expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
"${alias}" | grep Valid | perl -ne 'if(/until: (.*)\n/) { print "$1\n"; }`
    echo " Certificate ${alias} expires in '$expiry'"
done
```

網際網路流量隱私權

Amazon RDS 與內部部署應用程式之間，以及 Amazon RDS 與相同 AWS 區域內其他 AWS 資源之間的連線都會受到保護。

服務和內部部署用戶端與應用程式之間的流量。

在您的私有網路和 AWS 之間，您有兩個連線選項：

- AWS Site-to-Site VPN 連接。如需更多詳細資訊，請參閱[什麼是 AWS Site-to-Site VPN ?](#)
- AWS Direct Connect 連線。如需更多詳細資訊，請參閱[什麼是 AWS Direct Connect ?](#)

您可以使用 AWS 發佈的 API 操作，透過網路存取 Amazon RDS。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密 (PFS) 的密碼套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取索引鍵來簽署。或者，您可以使用[AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

Amazon RDS 的 Identity and access management

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員可控制哪些人員可進行身分驗證 (登入) 並獲得授權 (具有許可) 以使用 Amazon RDS 資源。IAM 是您可以使用的 AWS 服務，無需額外付費。

主題

- [物件](#)
- [使用身分來驗證](#)
- [使用政策管理存取權](#)
- [Amazon RDS 如何搭配 IAM 運作](#)
- [Amazon RDS 以身分為基礎的政策範例](#)
- [AWS Amazon RDS 的受管政策](#)
- [Amazon RDS 更新受 AWS 管政策](#)
- [防止跨服務混淆代理人問題](#)
- [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)
- [對 Amazon RDS 身分與存取進行故障診斷](#)

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在 Amazon RDS Amazon 中所做的工作。

服務使用者 – 若您使用 Amazon RDS 來執行任務，您的管理員可以提供您需要的登入資料和許可。隨著您為了執行作業而使用的 Amazon RDS 功能數量變多，您可能會需要額外的許可。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 Amazon RDS 中的某項功能，請參閱[對 Amazon RDS 身分與存取進行故障診斷](#)。

服務管理員 – 若您在公司負責管理 Amazon RDS 資源，您應該具備 Amazon RDS 的完整存取權限。您的任務是判斷員工應存取的 Amazon RDS 功能及資源。您接著必須將請求提交給您的管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解您公司可搭配 Amazon RDS 使用 IAM 的方式，請參閱[Amazon RDS 如何搭配 IAM 運作](#)。

管理員 - 如果您是管理員，請了解如何撰寫政策來管理 Amazon RDS 存取權的詳細資訊。若要檢視您可以在 IAM 中使用的範例 Amazon RDS 身分類型政策，請參閱 [Amazon RDS 以身分為基礎的政策範例](#)。

使用身分來驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中 [的如何登入](#) 您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的 [簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [多重要素驗證](#) 和 IAM 使用者指南中的 [在 AWS 中使用多重要素驗證 \(MFA\)](#)。

AWS 帳號根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的 [需要根使用者憑證的任務](#)。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [什麼是 IAM Identity Center ?](#)。

IAM 使用者和群組

[IAM 使用者](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 [IAM 使用者指南](#)中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的 [建立 IAM 使用者 \(而非角色\) 的時機](#)。

您可以使用 IAM 資料庫身分驗證來驗證您的資料庫執行個體。

IAM 資料庫身分驗證使用下列資料庫引擎：

- RDS for MariaDB
- RDS for MySQL
- RDS for PostgreSQL

如需使用 IAM 驗證資料庫執行個體身分的詳細資訊，請參閱 [適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證](#)。

IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它與使用者相似，但是不會與特定人員建立關聯。您可以 [切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的 [使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 暫時使用者許可 - 使用者可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 [IAM 使用者指南](#) 中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [許可集](#)。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 [IAM 角色與資源類型政策的差異](#)。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
 - 轉寄存取工作階段 — 當您使用 IAM 使用者或角色執行中的動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。
 - 服務角色 – 服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務](#)。
 - 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的 [利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

若要了解是否要使用 IAM 角色，請參閱 [《IAM 使用者指南》](#) 中的 [何時建立 IAM 角色 \(而不是使用者\)](#)。

使用政策管理存取權

您可以透過 AWS 過建立政策並將其附加到 IAM 身分或 AWS 資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當實體 (根使用者、使用者或 IAM 角色) 提出要求時，評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 [JSON 政策概觀](#)。

管理員可以使用策略來指定誰可以存取 AWS 資源，以及他們可以對這些資源執行哪些動作。每個 IAM 實體 (許可集或角色) 在開始時都沒有許可。換句話說，根據預設，使用者無法執行任何作業，甚至也無法變更他們自己的密碼。若要授予使用者執行動作的許可，管理員必須將許可政策連接到使用者。或者，管理員可以將使用者新增到具備預定許可的群組。管理員將許可給予群組時，該群組中的所有使用者都會獲得那些許可。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分類型政策是您可以連接到身分 (例如許可集或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的 [建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一許可集或角色。受管理的原則是獨立的原則，您可以附加至 AWS 帳戶中的多個權限集和角色。受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的 [在受管政策和內嵌政策間選擇](#)。

如需 Amazon RDS Amazon 特有的 AWS 受管政策的相關資訊，請參閱 [AWS Amazon RDS 的受管政策](#)。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- **許可界限：**許可界限是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (許可集或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定許可集或角色的資源型政策則不會受到許可界限限制。所有這類政策中的

明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 實體許可範圍](#)。

- 服務控制策略 (SCP) — SCP 是 JSON 策略，用於指定中組織或組織單位 (OU) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 [SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是許可集或角色身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的 [工作階段政策](#)。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的 [政策評估邏輯](#)。

Amazon RDS 如何搭配 IAM 運作

在您使用 IAM 管理 Amazon RDS 的存取權前，建議您先了解可搭配 Amazon RDS 使用的 IAM 功能有哪些。

您可以搭配 Amazon RDS 使用的 IAM 功能

IAM 功能	Amazon RDS 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
以屬性為基礎的存取控制 (ABAC)	是

IAM 功能	Amazon RDS 支援
臨時憑證	是
轉寄存取會話	是
服務角色	是
服務連結角色	是

若要深入瞭解 Amazon RDS Amazon 和其他 AWS 服務如何與 IAM 搭配使用，請參閱 IAM 使用者指南中的可與 IAM 搭配使用的[AWS 服務](#)。

主題

- [Amazon RDS 以身分為基礎的政策](#)
- [Amazon RDS 內以資源為基礎的政策](#)
- [Amazon RDS 的政策動作](#)
- [Amazon RDS 的政策資源](#)
- [Amazon RDS 的政策條件金鑰](#)
- [Amazon RDS 中的存取控制清單 \(ACL\)](#)
- [政策中具有 Amazon RDS 標籤的以屬性為基礎存取控制 Attribute-based access control \(ABAC\)](#)
- [將暫時登入資料與 Amazon RDS 搭配使用](#)
- [Amazon RDS 的轉發存取工作階段](#)
- [Amazon RDS 的服務角色](#)
- [Amazon RDS 的服務連結角色](#)

Amazon RDS 以身分為基礎的政策

支援身分型政策	是
---------	---

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。

Amazon RDS 以身分為基礎的政策範例

若要檢視 Amazon RDS 身分類型政策的範例，請參閱 [Amazon RDS 以身分為基礎的政策範例](#)。

Amazon RDS 內以資源為基礎的政策

支援以資源基礎的政策	否
------------	---

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南 [中的 IAM 中的跨帳戶資源存取](#)。

Amazon RDS 的政策動作

支援政策動作	是
--------	---

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

Amazon RDS 中的政策動作會在動作前使用以下前綴：`rds:`。例如，若要授予某人使用 Amazon RDS `DescribeDBInstances` API 操作描述資料庫執行個體的許可，請在其政策中包含 `rds:DescribeDBInstances` 動作。政策陳述式必須包含 `Action` 或 `NotAction` 元素。Amazon RDS 會定義自己的一組動作，描述您可以使用此服務執行的任務。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示。

```
"Action": [
  "rds:action1",
  "rds:action2"
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，如需指定開頭是 `Describe` 文字的所有動作，請包含以下動作：

```
"Action": "rds:Describe*"
```

若要查看 Amazon RDS 動作的清單，請參閱服務授權參考中的 [Amazon RDS 定義的動作](#)。

Amazon RDS 的政策資源

支援政策資源 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

資料庫執行個體資源具有以下 Amazon Resource Name (ARN)。

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

如需 ARN 格式的詳細資訊，請參閱 [Amazon 資源名稱 \(ARN\) 和 AWS 服務命名空間](#)。

例如，若要在陳述式中指定 dbtest 資料庫執行個體，請使用以下 ARN。

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

若要指定屬於特定帳戶的所有資料庫執行個體，請使用萬用字元 (*)。

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:*"
```

有些 RDS API 操作 (例如用來建立資源的操作)，無法在特定資源上執行。在這些情況下，請使用萬用字元 (*)。

```
"Resource": "*"
```

許多 Amazon RDS API 操作都涉及多個資源。例如，CreateDBInstance 會建立資料庫執行個體。建立資料庫執行個體時，您可指定使用者必須使用特定的安全群組和參數群組。若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN。

```
"Resource": [  
    "resource1",  
    "resource2"
```

若要查看 Amazon RDS 資源類型及其 ARN 的詳細資訊，請參閱服務授權參考中的 [Amazon RDS 定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon RDS 定義的動作](#)。

Amazon RDS 的政策條件金鑰

支援服務特定政策條件金鑰

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用 [條件運算子](#) 的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容金鑰](#)。

Amazon RDS 會定義自己的一組條件金鑰，也支援一些全域條件金鑰的使用。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容金鑰](#)。

所有 RDS API 操作均支援 `aws:RequestedRegion` 條件金鑰。

若要查看 Amazon RDS 條件金鑰清單，請參閱服務授權參考中的 [Amazon RDS 的條件金鑰](#)。若要了解您可以搭配哪些動作和資源使用條件金鑰，請參閱 [Amazon RDS 定義的動作](#)。

Amazon RDS 中的存取控制清單 (ACL)

支援存取控制清單 (ACL)	否
----------------	---

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

政策中具有 Amazon RDS 標籤的以屬性為基礎存取控制 Attribute-based access control (ABAC)

在政策中支援以屬性為基礎的存取控制 (ABAC) 標籤	是
-----------------------------	---

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱 IAM 使用者指南中的[什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的[使用屬性型存取控制 \(ABAC\)](#)。

如需標記 Amazon RDS 資源的詳細資訊，請參閱[指定條件：使用自訂標籤](#)。若要檢視身分型原則範例，以根據該資源上的標籤來限制存取資源，請參閱[利用具有兩個不同值的特定標籤，對資源上的動作授予許可](#)。

將暫時登入資料與 Amazon RDS 搭配使用

支援臨時憑證	是
--------	---

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料[搭配AWS 服務 使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的[切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱[IAM 中的暫時性安全憑證](#)。

Amazon RDS 的轉發存取工作階段

支持轉發訪問會話	是
----------	---

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

Amazon RDS 的服務角色

支援服務角色 是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務服務](#)。

Warning

變更服務角色的許可可能會中斷 Amazon RDS 功能。只有在 Amazon RDS 提供方法指引時，才能編輯服務角色。

Amazon RDS 的服務連結角色

支援服務連結角色 是

服務連結角色是一種連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需使用 Amazon RDS 服務連結角色的詳細資訊，請參閱 [使用 Amazon RDS 的服務連結角色](#)。

Amazon RDS 以身分為基礎的政策範例

根據預設，許可集和角色不具備建立或修改 Amazon RDS 資源的許可。他們也無法使用 AWS Management Console AWS CLI、或 AWS API 執行工作。IAM 管理員必須建立 IAM 政策，授予許可集和角色在指定資源上執行特定 API 操作的所需許可。管理員接著必須將這些政策連接至需要這些許可的許可集或角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的 [在 JSON 標籤上建立政策](#)。

主題

- [政策最佳實務](#)
- [使用 Amazon RDS 主控台](#)

- [允許使用者檢視他們自己的許可](#)
- [允許使用者在 AWS 帳戶中建立資料庫執行個體](#)
- [使用主控台所需的許可](#)
- [允許使用者對任何 RDS 資源執行任何描述動作](#)
- [允許使用者建立可使用指定資料庫參數群組和子網路群組的資料庫執行個體](#)
- [利用具有兩個不同值的特定標籤，對資源上的動作授予許可](#)
- [防止使用者刪除資料庫執行個體](#)
- [拒絕對資源的所有存取](#)
- [範例政策：使用條件金鑰](#)
- [指定條件：使用自訂標籤](#)

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 Amazon RDS 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的 AWS 受管理原則。它們在您的 AWS 帳戶。建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。

使用 Amazon RDS 主控台

若要存取 Amazon RDS 主控台，您必須擁有最低的一組許可。這些許可必須允許您列出和檢視有關 Amazon RDS Amazon 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合您嘗試執行之 API 作業的動作就可以了。

為確保這些實體仍然可以使用 Amazon RDS 主控台，請將下列 AWS 受管政策附加到實體。

```
AmazonRDSReadOnlyAccess
```

如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
```



```

        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

允許使用者在 AWS 帳戶中建立資料庫執行個體

以下是範例政策，允許具有 ID 123456789012 的使用者為您的 AWS 帳戶建立資料庫執行個體。此政策需要新資料庫執行個體的名稱以 test 開頭。新的資料庫執行個體也須使用 MySQL 資料庫引擎和 db.t2.micro 資料庫執行個體類別。此外，新的資料庫執行個體還須使用選項群組，以及以 default 開頭的資料庫參數群組，而且它還須使用 default 子網路群組。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateDBInstanceOnly",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBInstance"
      ],
      "Resource": [
        "arn:aws:rds*:123456789012:db:test*",
        "arn:aws:rds*:123456789012:og:default*",
        "arn:aws:rds*:123456789012:pg:default*",
        "arn:aws:rds*:123456789012:subgrp:default"
      ],
      "Condition": {
        "StringEquals": {
          "rds:DatabaseEngine": "mysql",
          "rds:DatabaseClass": "db.t2.micro"
        }
      }
    }
  ]
}

```

```
}
```

政策包括單一陳述式，指定使用者的下列許可：

- 此原則可讓使用者使用 [CreateDbInstance API 作業來建立資料庫執行個體](#) (這也適用於建立 AWS CLI 的 DB 執行個體命令)。AWS Management Console
- Resource 元素指定使用者可對資源或搭配資源執行動作。您可以使用 Amazon Resource Name (ARN) 來指定資源。此 ARN 包括資源所屬的服務名稱 (rds)、「AWS 區域」(*表示此範例中的任何區域)、AWS 帳號 (在此範例中123456789012是帳號) 以及資源類型。如需建立 ARN 的詳細資訊，請參閱[在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

範例中的 Resource 元素對使用者的資源指定下列政策限制：

- 新資料庫執行個體的資料庫執行個體識別符必須以 test 開頭 (例如，testCustomerData1、test-region2-data)。
- 新資料庫執行個體的選項群組必須以 default 開頭。
- 新資料庫執行個體的資料庫參數群組必須以 default 開頭。
- 新資料庫執行個體的字網路群組必須是 default 子網路群組。
- Condition 元素指定資料庫引擎必須是 MySQL，而且資料庫執行個體類別必須是 db.t2.micro。Condition 元素指定政策應該生效時的條件。您可以使用 Condition 元素來新增其他許可或限制。如需指定條件的詳細資訊，請參閱[Amazon RDS 的政策條件金鑰](#)。此範例指定 rds:DatabaseEngine 和 rds:DatabaseClass 條件。如需 rds:DatabaseEngine 有效條件值的相關資訊，請參閱 [CreateDBInstance](#) 中的 Engine 參數下的清單。如需 rds:DatabaseClass 的有效條件值的相關資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。

此政策不指定 Principal 元素，因為您不會在以身分為基礎的政策中，指定取得許可的主體。當您將政策連接至使用者時，這名使用者即為隱含主體。當您將許可政策連接至 IAM 角色，該角色的信任政策中所識別的主體即取得許可。

若要查看 Amazon RDS 動作的清單，請參閱服務授權參考中的 [Amazon RDS 定義的動作](#)。

使用主控台所需的許可

針對使用主控台的使用者，該使用者必須擁有一組符合最低限制的許可。這些許可允許使用者描述其 AWS 帳戶的 Amazon RDS 資源，並提供其他相關資訊，包括 Amazon EC2 安全性和網路資訊。

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言，主控台就無法如預期運作。為確保這些使用者仍可使用主控台，也請將 AmazonRDSReadOnlyAccess 受管政策連接至使用者，如[使用政策管理存取權](#)所述。

對於僅呼叫 AWS CLI 或 Amazon RDS API 的使用者，您不需要允許其最基本主控台許可。

下列政策授與根 AWS 帳戶的所有 Amazon RDS Amazon 資源的完整存取權：

```
AmazonRDSFullAccess
```

允許使用者對任何 RDS 資源執行任何描述動作

下列許可政策會授予使用者執行開頭為 Describe 之所有動作的許可。這些動作會顯示 RDS 資源 (如資料庫執行個體) 的相關資訊。Resource 元素中的萬用字元 (*) 表示可對帳戶擁有的所有 Amazon RDS 資源執行動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRDSDescribe",
      "Effect": "Allow",
      "Action": "rds:Describe*",
      "Resource": "*"
    }
  ]
}
```

允許使用者建立可使用指定資料庫參數群組和子網路群組的資料庫執行個體

以下許可政策授予許可，以允許使用者只能建立一個必須使用 mydbpg 資料庫參數群組和 mydbsubnetgroup 資料庫子網路群組的資料庫執行個體。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
```

```

    "Action": "rds:CreateDBInstance",
    "Resource": [
      "arn:aws:rds:*:*:pg:mydbpg",
      "arn:aws:rds:*:*:subgrp:mydbsubnetgroup"
    ]
  }
]
}

```

利用具有兩個不同值的特定標籤，對資源上的動作授予許可

您可以在身分類型政策中使用條件，根據標籤來控制存取 Amazon RDS 資源。下列政策允許在 stage 標籤設為 development 或 test 的資料庫執行個體上執行 CreateDBSnapshot API 操作的許可。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAnySnapshotName",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBSnapshot"
      ],
      "Resource": "arn:aws:rds:*:123456789012:snapshot:*"
    },
    {
      "Sid": "AllowDevTestToCreateSnapshot",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBSnapshot"
      ],
      "Resource": "arn:aws:rds:*:123456789012:db:*",
      "Condition": {
        "StringEquals": {
          "rds:db-tag/stage": [
            "development",
            "test"
          ]
        }
      }
    }
  ]
}

```

下列政策允許在 stage 標籤設為 development 或 test 的資料庫執行個體上執行 ModifyDBInstance API 操作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowChangingParameterOptionSecurityGroups",
      "Effect": "Allow",
      "Action": [
        "rds:ModifyDBInstance"
      ],
      "Resource": [
        "arn:aws:rds:*:123456789012:pg:*",
        "arn:aws:rds:*:123456789012:secgrp:*",
        "arn:aws:rds:*:123456789012:og:*"
      ]
    },
    {
      "Sid": "AllowDevTestToModifyInstance",
      "Effect": "Allow",
      "Action": [
        "rds:ModifyDBInstance"
      ],
      "Resource": "arn:aws:rds:*:123456789012:db:*",
      "Condition": {
        "StringEquals": {
          "rds:db-tag/stage": [
            "development",
            "test"
          ]
        }
      }
    }
  ]
}
```

防止使用者刪除資料庫執行個體

以下許可政策授予許可，以防止使用者刪除特定的資料庫執行個體。例如，您可能想拒絕給予任何非管理員使用者刪除生產資料庫執行個體的能力。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyDelete1",
      "Effect": "Deny",
      "Action": "rds:DeleteDBInstance",
      "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
    }
  ]
}
```

拒絕對資源的所有存取

您可以明確拒絕對資源的存取權。拒絕政策優先於允許政策。下列政策明確拒絕使用者管理資源的能力：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "rds:*",
      "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"
    }
  ]
}
```

範例政策：使用條件金鑰

以下是如何在 Amazon RDS IAM 許可政策中使用條件金鑰的範例。

範例 1：授予許可來建立一個使用特定資料庫引擎且不是多個可用區的資料庫執行個體

下列政策會使用 RDS 條件金鑰，並允許使用者只建立使用 MySQL 資料庫引擎，而不使用 MultiAZ 的資料庫執行個體。Condition 元素表示資料庫引擎是 MySQL 的需求。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowMySQLCreate",
```

```

    "Effect": "Allow",
    "Action": "rds:CreateDBInstance",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "rds:DatabaseEngine": "mysql"
      },
      "Bool": {
        "rds:MultiAz": false
      }
    }
  }
]
}

```

範例 2：明確拒絕許可，不得為特定資料庫執行個體類別建立資料庫執行個體，以及不得建立使用佈建 IOPS 的資料庫執行個體

下列政策明確拒絕許可，不得建立使用資料庫執行個體類別 r3.8xlarge 和 m4.10xlarge 的資料庫執行個體，因為它們是最大且最昂貴的資料庫執行個體類別。此政策也會防止使用者建立使用佈建 IOPS 的資料庫執行個體，因為它們會產生額外的成本。

明確拒絕許可會取代任何其他已授予的許可。這確保身分不會意外取得您從未想要授予的許可。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyLargeCreate",
      "Effect": "Deny",
      "Action": "rds:CreateDBInstance",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "rds:DatabaseClass": [
            "db.r3.8xlarge",
            "db.m4.10xlarge"
          ]
        }
      }
    },
    {
      "Sid": "DenyPIOPSCreate",

```

```

    "Effect": "Deny",
    "Action": "rds:CreateDBInstance",
    "Resource": "*",
    "Condition": {
      "NumericNotEquals": {
        "rds:Piops": "0"
      }
    }
  ]
}

```

範例 3：限制可用來標記資源的一組標籤金鑰和值

下列政策使用 RDS 條件索引鍵，並允許將索引鍵為 stage 且值為 test、qa 和 production 的標籤新增給資源。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource": "*",
      "Condition": {
        "streq": {
          "rds:req-tag/stage": [
            "test",
            "qa",
            "production"
          ]
        }
      }
    }
  ]
}

```

指定條件：使用自訂標籤

Amazon RDS 支援在 IAM 政策中使用自訂標籤來指定條件。

例如，假設您將值為 `environment`、`beta`、`staging` 等等的 `production` 標籤新增至資料庫執行個體。如果這樣做，您就可以建立政策，根據 `environment` 標籤值限制某些資料庫執行個體的使用者。

Note

自訂標籤識別符會區分大小寫。

下表列出您可以在 `Condition` 元素中使用的 RDS 標籤識別符。

RDS 標籤識別符	適用對象
<code>db-tag</code>	資料庫執行個體，包括讀取複本
<code>snapshot-tag</code>	資料庫快照
<code>ri-tag</code>	預留資料庫執行個體
<code>og-tag</code>	資料庫選項群組
<code>pg-tag</code>	資料庫參數群組
<code>subgrp-tag</code>	資料庫子網路群組
<code>es-tag</code>	事件訂閱
<code>cluster-tag</code>	資料庫叢集
<code>cluster-pg-tag</code>	資料庫叢集參數群組
<code>cluster-snapshot-tag</code>	資料庫叢集快照

自訂標籤條件的語法如下：

```
"Condition":{"StringEquals":{"rds:rds-tag-identifier/tag-name":["value"]}}
```

例如，下列 `Condition` 元素會套用至標籤名稱為 `environment` 且標籤值為 `production` 的資料庫執行個體。

```
"Condition":{"StringEquals":{"rds:db-tag/environment":["production"]}} }
```

如需建立標籤的相關資訊，請參閱[標記 Amazon RDS 資源](#)。

Important

如果您使用標記功能來管理 RDS 資源的存取，則我們建議您安全存取 RDS 資源的標籤。您可以建立 `AddTagsToResource` 和 `RemoveTagsFromResource` 動作的政策，來管理標籤的存取。例如，下列政策會拒絕使用者可對所有資源新增或移除標籤的能力。然後，您可以建立政策來允許特定使用者新增或刪除標籤。

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"DenyTagUpdates",
      "Effect":"Deny",
      "Action":[
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource":"*"
    }
  ]
}
```

若要查看 Amazon RDS 動作的清單，請參閱服務授權參考中的 [Amazon RDS 定義的動作](#)。

範例政策：使用自訂標籤

以下是如何在 Amazon RDS IAM 許可政策中使用自訂標籤的範例。如需將標籤新增至 Amazon RDS 資源的詳細資訊，請參閱在 [Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

Note

所有範例都使用 us-west-2 區域，且其中的帳戶 ID 皆為虛構。

範例 1：利用具有兩個不同值的特定標籤，對資源上的動作授予許可

下列政策允許在 stage 標籤設為 development 或 test 的資料庫執行個體上執行 CreateDBSnapshot API 操作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAnySnapshotName",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBSnapshot"
      ],
      "Resource": "arn:aws:rds:*:123456789012:snapshot:*"
    },
    {
      "Sid": "AllowDevTestToCreateSnapshot",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBSnapshot"
      ],
      "Resource": "arn:aws:rds:*:123456789012:db:*",
      "Condition": {
        "StringEquals": {
          "rds:db-tag/stage": [
            "development",
            "test"
          ]
        }
      }
    }
  ]
}
```

下列政策允許在 stage 標籤設為 development 或 test 的資料庫執行個體上執行 ModifyDBInstance API 操作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowChangingParameterOptionSecurityGroups",
```

```

    "Effect": "Allow",
    "Action": [
        "rds:ModifyDBInstance"
    ],
    "Resource": [
        "arn:aws:rds:*:123456789012:pg:*",
        "arn:aws:rds:*:123456789012:secgrp:*",
        "arn:aws:rds:*:123456789012:og:*"
    ]
},
{
    "Sid": "AllowDevTestToModifyInstance",
    "Effect": "Allow",
    "Action": [
        "rds:ModifyDBInstance"
    ],
    "Resource": "arn:aws:rds:*:123456789012:db:*",
    "Condition": {
        "StringEquals": {
            "rds:db-tag/stage": [
                "development",
                "test"
            ]
        }
    }
}
]
}
}

```

範例 2：明確拒絕許可，不得建立一個使用所指定之資料庫參數群組的資料庫執行個體

下列政策明確拒絕許可，不得建立一個使用資料庫參數群組與特定標籤值搭配的資料庫執行個體。如果您需要在建立資料庫執行個體時，一律使用特定客戶建立的資料庫參數群組，則可能會套用此政策。使用 Deny 的政策最常用來限制更廣泛政策所授予的存取。

明確拒絕許可會取代任何其他已授予的許可。這確保身分不會意外取得您從未想要授予的許可。

```

{
    "Version": "2012-10-17",
    "Statement": [
        {

```

```

    "Sid": "DenyProductionCreate",
    "Effect": "Deny",
    "Action": "rds:CreateDBInstance",
    "Resource": "arn:aws:rds:*:123456789012:pg:*",
    "Condition": {
      "StringEquals": {
        "rds:pg-tag/usage": "prod"
      }
    }
  ]
}

```

範例 3：對資料庫執行個體上的動作授予許可，而此資料庫執行個體的執行個體名稱字首會加上使用者名稱

下列政策允許在資料庫執行個體上呼叫任何 API (但 `AddTagsToResource` 或 `RemoveTagsFromResource` 除外) 的許可，而此資料庫執行個體具有字首會加上使用者名稱的資料庫執行個體名稱，以及具有稱為 `stage` 等於 `devo` 的標籤，或沒有稱為 `stage` 的標籤。

政策中的 `Resource` 一行會依其 Amazon Resource Name (ARN) 識別資源。如需使用 ARN 與 Amazon RDS 資源的詳細資訊，請參閱在 [Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFullDevAccessNoTags",
      "Effect": "Allow",
      "NotAction": [
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource": "arn:aws:rds:*:123456789012:db:${aws:username}*",
      "Condition": {
        "StringEqualsIfExists": {
          "rds:db-tag/stage": "devo"
        }
      }
    }
  ]
}

```


AWS Amazon RDS 的受管政策

若要新增權限集和角色的權限，使用 AWS 受管理的原則比自行撰寫原則更容易。建立 [IAM 客戶受管政策](#) 需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的 AWS 受管政策。這些政策涵蓋常見的使用案例，並可在您的 AWS 帳戶中使用。如需 AWS 受管政策的詳細資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#)。

AWS 服務 維護和更新 AWS 受管理的策略。您無法變更 AWS 受管理原則中的權限。服務有時會將其他權限新增至受 AWS 管理的策略，以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (許可集和角色)。當新功能啟動或新作業可用時，服務最有可能更新 AWS 受管理的策略。服務不會從 AWS 受管理的政策移除權限，因此政策更新不會破壞您現有的權限。

此外，還 AWS 支援跨多個服務之工作職能的受管理原則。例如，ReadOnlyAccess AWS 受管理的策略提供對所有資源 AWS 服務 和資源的唯讀存取。當服務啟動新功能時，會為新作業和資源新 AWS 增唯讀權限。如需任務職能政策的清單和說明，請參閱 IAM 使用者指南中 [有關任務職能的 AWS 受管政策](#)。

主題

- [AWS 管理策略：亞馬遜 ReadOnlyAccess](#)
- [AWS 管理策略：亞馬遜 FullAccess](#)
- [AWS 管理策略：亞馬遜 DataFullAccess](#)
- [AWS 管理策略：亞馬遜 EnhancedMonitoringRole](#)
- [AWS 管理策略：亞馬遜 PerformanceInsightsReadOnly](#)
- [AWS 管理策略：亞馬遜 PerformanceInsightsFullAccess](#)
- [AWS 管理策略：亞馬遜 DirectoryServiceAccess](#)
- [AWS 管理策略：亞馬遜 ServiceRolePolicy](#)
- [AWS 管理策略：亞馬遜 CustomServiceRolePolicy](#)
- [AWS 管理策略：亞馬遜自定義實例 ProfileRolePolicy](#)

AWS 管理策略：亞馬遜 ReadOnlyAccess

此政策允 Amazon 透過 AWS Management Console.

許可詳細資訊

此政策包含以下許可：

- rds – 允許主體說明 Amazon RDS 資源並列出 Amazon RDS 資源的標籤。
- cloudwatch— 允許校長取得 Amazon CloudWatch 指標統計資料。
- ec2 – 允許主體說明可用區域和聯網資源。
- logs— 可讓主參與者描述 CloudWatch 記錄群組的記錄檔資料流，以及取得 CloudWatch 記錄記錄事件。
- devops-guru— 允許主體描述具有 Amazon DevOps Guru 涵蓋範圍的資源，這是由 CloudFormation 堆疊名稱或資源標籤指定的。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》ReadOnlyAccess 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 FullAccess

此政策提供完整的 Amazon RDS 存取權，透過 AWS Management Console。

許可詳細資訊

此政策包含以下許可：

- rds – 允許主體完整存取 Amazon RDS。
- application-autoscaling – 允許主體說明和管理 應用程式自動擴展擴展目標和政策。
- cloudwatch— 允許主體取得 CloudWatch 度量靜態和管理 CloudWatch 警示。
- ec2 – 允許主體說明可用區域和聯網資源。
- logs— 可讓主參與者描述 CloudWatch 記錄群組的記錄檔資料流，以及取得 CloudWatch 記錄記錄事件。
- outposts— 允許主參與者取得 AWS Outposts 實例類型。
- pi – 允許主體取得績效詳情指標。
- sns – 允許主體訂閱 Amazon Simple Notification Service (Amazon SNS) 和主題，並發佈 Amazon SNS 訊息。
- devops-guru— 允許主體描述具有 Amazon DevOps Guru 涵蓋範圍的資源，這是由 CloudFormation 堆疊名稱或資源標籤指定的。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》FullAccess 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 DataFullAccess

此原則允許在特定 Aurora Serverless 叢集上使用 Data API 和查詢編輯器的完整存取權 AWS 帳戶。此原則允許從中取得密碼的值 AWS Secrets Manager。AWS 帳戶

您可將 AmazonRDSDataFullAccess 政策連接到 IAM 身分。

許可詳細資訊

此政策包含以下許可：

- dbqms – 允許主體存取、建立、刪除、說明和更新查詢。Database Query Metadata Service (dbqms) 是僅內部服務。它為多個查詢編輯器提供您最近和保存的查詢查詢 AWS 服務，包括 Amazon RDS。AWS Management Console
- rds-data – 允許主體在 Aurora Serverless 資料庫執行 SQL 陳述式。
- secretsmanager— 允許主參與者從 AWS Secrets Manager 中取得密碼值。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》DataFullAccess 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 EnhancedMonitoringRole

此政策可讓您存取 Amazon RDS 增強型監控的 Amazon CloudWatch 日誌。

許可詳細資訊

此政策包含以下許可：

- logs— 允許主體建立 CloudWatch 記錄檔記錄群組和保留原則，以及建立和說明 CloudWatch 記錄群組的記錄檔資料流。它還允許主體放置和獲取 CloudWatch 日誌記錄事件。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》EnhancedMonitoringRole 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 PerformanceInsightsReadOnly

此政策提供對 Amazon RDS 資料庫執行個體和 Amazon Aurora 資料庫叢集的 Amazon RDS 績效詳情唯讀存取權限。

此政策現在包含 Sid (陳述式 ID) 作為政策陳述式的識別符。

許可詳細資訊

此政策包含以下許可：

- rds – 允許主體說明 Amazon RDS 資料庫執行個體和 Amazon Aurora 資料庫叢集。
- pi – 允許主體呼叫 Amazon RDS 績效詳情 API 並存取績效詳情指標。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》PerformanceInsightsReadOnly 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 PerformanceInsightsFullAccess

此政策提供對 Amazon RDS 資料庫執行個體和 Amazon Aurora 資料庫叢集的 Amazon RDS Performance Insights 完整存取權限。

此政策現在包含 Sid (陳述式 ID) 作為政策陳述式的識別符。

許可詳細資訊

此政策包含以下許可：

- rds – 允許主體說明 Amazon RDS 資料庫執行個體和 Amazon Aurora 資料庫叢集。
- pi – 允許主體呼叫 Amazon RDS Performance Insights API，以及建立、檢視和刪除績效分析報告。
- cloudwatch— 允許主體列出所有 Amazon CloudWatch 指標，並取得指標資料和統計資料。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》PerformanceInsightsFullAccess 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 DirectoryServiceAccess

此政策允許 Amazon RDS 呼叫 AWS Directory Service。

許可詳細資訊

此政策包含以下許可：

- ds— 允許主參與者描述 AWS Directory Service 目錄並控制對目 AWS Directory Service 錄的授權。

如需有關此原則 (包括 JSON 政策文件) 的詳細資訊，請參閱《AWS 受管原則參考指南》DirectoryServiceAccess 中的 [AmazonRDS](#)。

AWS 管理策略：亞馬遜 ServiceRolePolicy

您無法將 AmazonRDSServiceRolePolicy 政策附加至 IAM 實體。此政策會附加至服務連結角色，而此角色可讓 Amazon RDS 代表您執行動作。如需詳細資訊，請參閱 [Amazon RDS 的服務連結角色許可](#)。

AWS 管理策略：亞馬遜 CustomServiceRolePolicy

您無法將 AmazonRDSCustomServiceRolePolicy 政策附加至 IAM 實體。此政策會附加至服務連結角色，而此角色可讓 Amazon RDS 代表您執行動作。如需詳細資訊，請參閱 [Amazon RDS Custom 的服務連結角色許可](#)。

AWS 管理策略：亞馬遜自定義實例 ProfileRolePolicy

您不應將 AmazonRDSCustomInstanceProfileRolePolicy 連接到 IAM 實體。它應該只附加到執行個體設定檔角色，該角色用於將權限授與 Amazon RDS 自訂資料庫執行個體以執行各種自動化動作和資料庫管理任務。在建立 RDS 自訂執行個體期間，將執行個體設定檔作為 custom-iam-instance-profile 參數傳遞，RDS Custom 會將此執行個體設定檔與您的資料庫執

許可詳細資訊

此政策包含以下許可：

- ssmssmmessages, ec2messages-允許 RDS Custom 透過系統管理員在資料庫執行個體上進行通訊、執行自動化和維護代 Systems Manager 程式。
- ec2, s3-允許 RDS Custom 在提供 point-in-time 還原功能的資料庫執行個體上執行備份作業。
- secretsmanager-允許 RDS 自訂管理 RDS 自訂建立的資料庫執行個體特定密碼。
- cloudwatch, logs-允許 RDS Custom CloudWatch 透過 CloudWatch 代理程式將資料庫執行個體指標和記錄上傳至。
- events, sqs-允許 RDS 自訂傳送和接收有關資料庫執行個體的狀態資訊。
- kms-允許 RDS 自訂使用執行個體特定的 KMS 金鑰，對 RDS 自訂管理的密碼和 S3 物件執行加密。

如需有關此政策的詳細資訊，包括 JSON 政策文件，請參閱《AWS 受管策略參考指南》ProfileRolePolicy 中的 [AmazonrdsCustom 執行個體](#)。

Amazon RDS 更新受 AWS 管政策

自此服務開始追蹤這些變更以來，檢視 Amazon RDS AWS 受管政策更新的詳細資訊。如需有關此頁面變更的自動提醒，請訂閱 Amazon RDS [Document history \(文件歷程記錄\)](#) 頁面上的 RSS 摘要。

變更	描述	日期
Amazon RDS Custom 的服務連結角色許可 – 更新現有政策	Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy。此新權限允許 RDS Custom 將服務角色作為執行個體設定檔與 RDS 自訂執行個體建立關聯。如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可 。	2024年4月19日
AWS Amazon RDS 的受管政策 – 更新現有政策	Amazon RDS 為 AWSServiceRoleForRDSCustom 服務連結角色新增了新 AmazonRDS CustomServiceRolePolicy 的權限，以允許 SQL Server 的 RDS 自訂修改基礎資料庫主機執行個體類型。RDS 也新增了取 ec2:DescribeInstanceTypes 得資料庫主機執行個體類型資訊的權限。如需詳細資訊，請參閱 AWS Amazon RDS 的受管政策 。	2024年4月8日
AWS Amazon RDS 的受管政策 – 新政策	Amazon RDS 新增名為的新受管政策，AmazonRDS Custom InstanceProfileRolePolicy 允許 RDS Custom	2024年2月27日

變更	描述	日期
	<p>透過 EC2 執行個體設定檔執行自動化動作和資料庫管理任務。如需詳細資訊，請參閱 AWS Amazon RDS 的受管政策。</p>	
<p>Amazon RDS 的服務連結角色許可 – 更新現有政策</p>	<p>Amazon RDS 在AWSServiceRoleForRDS 服務連結角色中新增了新AmazonRDSServiceRolePolicy 的陳述式 ID。</p> <p>如需詳細資訊，請參閱 Amazon RDS 的服務連結角色許可。</p>	<p>2024 年 1 月 19 日</p>
<p>AWS Amazon RDS 的受管政策 – 更新現有政策</p>	<p>AmazonRDSPerformanceInsightsReadOnly 和 AmazonRDSPerformanceInsightsFullAccess 受管政策現在包含 Sid (陳述式 ID) 作為政策陳述式中的識別符。</p> <p>如需詳細資訊，請參閱 AWS 管理策略：亞馬遜 PerformanceInsightsReadOnly 和 AWS 管理策略：亞馬遜 PerformanceInsightsFullAccess</p>	<p>2023 年 10 月 23 日</p>

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy。這些新權限允許 Oracle 的 RDS 自訂建立、修改和刪除 EventBridge 受管規則。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 9 月 20 日
AWS Amazon RDS 的受管政策 – 更新現有政策	<p>Amazon RDS 將新的許可新增到 AmazonRDSFullAccess 受管政策。這些許可允許您產生、檢視及刪除一段時間區間內的績效分析報告。</p> <p>如需有關設定 Performance Insights 存取政策的詳細資訊，請參閱 設定績效詳情的存取政策</p>	2023 年 8 月 17 日

變更	描述	日期
AWS Amazon RDS 的受管政策 – 新政策和更新現有政策	<p>Amazon RDS 新增了新的許可至 AmazonRDSPerformanceInsightsReadOnly 受管策略和名為 AmazonRDSPerformanceInsightsFullAccess 的新受管政策。這些許可允許您分析一段時間區間內的 Performance Insights、檢視分析結果及建議，以及刪除報告。</p> <p>如需有關設定 Performance Insights 存取政策的詳細資訊，請參閱 設定績效詳情的存取政策</p>	2023 年 8 月 16 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy。這些新權限允許 RDS Custom for Oracle 使用資料庫快照。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 6 月 23 日

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 。這些新權限允許 RDS Custom for Oracle 使用資料庫快照。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 6 月 23 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 。這些新權限允許 RDS Custom 建立網路介面。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 5 月 30 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 。這些新許可允許 RDS Custom 呼叫 Amazon EBS 來檢查儲存配額。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 4 月 18 日

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS Custom 已將新許可新增至 <code>AWSServiceRoleForRDSCustom</code> 服務連結角色的 <code>AmazonRDSCustomServiceRolePolicy</code>，以與 Amazon SQS 整合。RDS Custom 需要與 Amazon SQS 整合，才能在客戶帳戶中建立和管理 SQS 佇列。SQS 佇列名稱遵循格式 <code>do-not-delete-rds-custom-[identifier]</code>，並標記為 Amazon RDS Custom。也新增了 <code>ec2:CreateSnapshot</code> 的許可，以允許 RDS Custom 為連接至執行個體的磁碟區建立備份。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2023 年 4 月 6 日

變更	描述	日期
AWS Amazon RDS 的受管政策 – 更新現有政策	<p>Amazon RDS ListMetrics 向和添加了一個新的 Amazon CloudWatch 命名空間 AmazonRDSFullAccess 和 AmazonRDSReadOnlyAccess。</p> <p>Amazon RDS 需要此命名空間才能列出特定資源用量指標。</p> <p>如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的管理 CloudWatch 資源存取許可概觀。</p>	2023 年 4 月 4 日
AWS Amazon RDS 的受管政策 – 更新現有政策	<p>Amazon RDS 新增了新的許可 AmazonRDSFullAccess 和 AmazonRDSReadOnlyAccess 受管政策，以允許在 RDS 主控台中顯示 Amazon DevOps Guru 發現項目。</p> <p>需要此權限才能顯示 DevOps Guru 發現項目。</p> <p>如需詳細資訊，請參閱 Amazon RDS 對 AWS 受管政策的更新。</p>	2023 年 3 月 30 日

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 為AWSServiceRoleForRDS 服務連結角色新增了新AmazonRDS ServiceRolePolicy 的許可，以便與 AWS Secrets Manager之整合。RDS 需要與 Secrets Manager 整合，才能在 Secrets Manager 中管理主要使用者密碼。密碼使用舊有的命名慣例並限制客戶更新。</p> <p>如需詳細資訊，請參閱 使用 Amazon RDS Amazon 和密碼管理 AWS Secrets Manager。</p>	2022 年 12 月 22 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 已將新許可新增至 AWSServiceRoleForRDS Custom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 。RDS Custom 支援資料庫叢集。原則中的這些新權限允許 RDS Custom 代表您的資料庫叢集呼叫 AWS 服務。</p> <p>如需詳細資訊，請參閱 Amazon RDS Custom 的服務連結角色許可。</p>	2022 年 11 月 9 日

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 新增了對 <code>AWSServiceRoleForRDS</code> 服務連結角色的新許可，以便與 AWS Secrets Manager 整合。</p> <p>SQL Server Reporting Services (SSRS) 電子郵件必須與 Secrets Manager 整合，才能在 RDS 上運作。SSRS 電子郵件會代表客戶建立一個秘密。密碼使用舊有的命名慣例並限制客戶更新。</p> <p>如需詳細資訊，請參閱 使用 SSRS 電子郵件傳送報告。</p>	2022 年 8 月 26 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 添加了一個新的 Amazon CloudWatch 命名空間 <code>AmazonRDSPreviewServiceRolePolicy</code> 間到 <code>PutMetricData</code>。</p> <p>Amazon RDS 需要此命名空間才能發佈資源用量指標。</p> <p>如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的使用條件金鑰限制對 CloudWatch 命名空間的存取。</p>	2022 年 6 月 7 日

變更	描述	日期
<p>Amazon RDS 的服務連結角色許可 – 更新現有政策</p>	<p>Amazon RDS 添加了一個新的 Amazon CloudWatch 命名空間 AmazonRDSBetaServiceRolePolicy 間到 PutMetricData .</p> <p>Amazon RDS 需要此命名空間才能發佈資源用量指標。</p> <p>如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的使用條件金鑰限制對 CloudWatch 命名空間的存取。</p>	<p>2022 年 6 月 7 日</p>
<p>Amazon RDS 的服務連結角色許可 – 更新現有政策</p>	<p>Amazon RDS 添加了一個新的 Amazon CloudWatch 命名空間 AWSServiceRoleForRDS 間到 PutMetricData .</p> <p>Amazon RDS 需要此命名空間才能發佈資源用量指標。</p> <p>如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的使用條件金鑰限制對 CloudWatch 命名空間的存取。</p>	<p>2022 年 4 月 22 日</p>

變更	描述	日期
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 會將新的許可新增至 <code>AWSServiceRoleForRDS</code> 服務連結角色來管理客戶擁有的 IP 集區和本機閘道路由表 (LGW-RTB) 的許可。</p> <p>Outpost 上的 RDS 需要這些許可權限，才可跨 Outpost 的本機網路執行多可用區域複寫。</p> <p>如需詳細資訊，請參閱 在 AWS Outposts 上使用 Amazon RDS 的異地同步備份部署。</p>	2022 年 4 月 19 日
身分型政策 – 更新現有政策	<p>Amazon RDS 將新的許可權限新增至 <code>AmazonRDSFullAccess</code> 受管政策來說明 LGW-RTB 上的許可權限。</p> <p>Outpost 上的 RDS 需要此許可權限，才可跨 Outpost 的本機網路執行多可用區域複寫。</p> <p>如需詳細資訊，請參閱 在 AWS Outposts 上使用 Amazon RDS 的異地同步備份部署。</p>	2022 年 4 月 19 日

變更	描述	日期
AWS Amazon RDS 的受管政策 – 新政策	<p>Amazon RDS 新增了一個名為的新受管政策，AmazonRDS PerformanceInsightsReadOnly 以允許 Amazon RDS 代表您的資料庫執行個體呼叫 AWS 服務。</p> <p>如需有關設定 Performance Insights 存取政策的詳細資訊，請參閱 設定績效詳情的存取政策</p>	2022 年 3 月 10 日
Amazon RDS 的服務連結角色許可 – 更新現有政策	<p>Amazon RDS 添加了新AWSServiceRoleForRDS 的 PutMetricData Amazon CloudWatch 命名空間。</p> <p>Amazon 文件資料庫 (與 MongoDB 相容性) 和亞馬 Amazon Neptune 需要這些命名空間才能發佈指標。CloudWatch</p> <p>如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的使用條件金鑰限制對 CloudWatch命名空間的存取。</p>	2022 年 3 月 4 日
Amazon RDS Custom 的服務連結角色許可 – 新政策	<p>Amazon RDS 新增名為 AWSServiceRoleForRDS Custom 的新服務連結角色，以允許 RDS Custom 代表您的資料庫執行個體呼叫 AWS 服務。</p>	2021 年 10 月 26 日

變更	描述	日期
Amazon RDS 開始追蹤變更	Amazon RDS 開始追蹤其 AWS 受管政策的變更。	2021 年 10 月 26 日

防止跨服務混淆代理人問題

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多權限的實體執行該動作。在 AWS 中，跨服務模擬可能會導致混淆代理人問題。

在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況，AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。如需詳細資訊，請參閱《IAM 使用者指南》中的[混淆代理問題](#)。

若要限制 Amazon RDS 為特定資源提供另一項服務的許可，建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵。

某些情況下，[aws:SourceArn](#) 值不包含帳戶 ID，例如使用 Amazon S3 儲存貯體的 Amazon 資源名稱 (ARN) 時。在這些情況下，請務必同時使用兩個全域條件內容索引鍵來限制許可。在某些情況下，可以同時使用全域條件內容索引鍵和包含帳號 ID 的 [aws:SourceArn](#) 值。在這些情況下，當在相同政策陳述式中使用 [aws:SourceAccount](#) 值和 [aws:SourceArn](#) 裡的帳戶時，請確認兩者使用同樣的帳戶 ID。如果您想要僅允許一個資源與跨服務存取相關聯，請使用 [aws:SourceArn](#)。如果您想要允許特定 AWS 帳戶中的任何資源與跨服務使用相關聯，請使用 [aws:SourceAccount](#)。

請確定 [aws:SourceArn](#) 的值是 Amazon RDS 資源類型的 ARN。如需詳細資訊，請參閱 [在 Amazon RDS 中使用 Amazon Resource Name \(ARN\)](#)。

防範混淆代理人問題的最有效方法是使用 [aws:SourceArn](#) 全域條件內容索引鍵，以及資源的完整 ARN。在某些情況下，您可能不知道資源的完整 ARN，或者指定了多個資源。在這些情況下，請針對 ARN 的未知部分，使用含有萬用字元 (*) 的全域條件內容索引鍵 ([aws:SourceArn](#))。例如，`arn:aws:rds:*:123456789012:*`。

下列範例示範如何使用 Amazon RDS 中的 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，來預防混淆代理人問題。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
```

```
"Condition": {
  "ArnLike": {
    "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
  },
  "StringEquals": {
    "aws:SourceAccount": "123456789012"
  }
}
}
```

如需更多使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵的政策範例，請參閱以下區段：

- [授予將通知發佈至 Amazon SNS 主題的許可](#)
- [手動建立原生備份與還原的 IAM 角色](#)
- [為 SQL Server 資料庫執行個體設定 Windows 身分驗證](#)
- [整合 RDS for SQL Server 與 S3 的先決條件](#)
- [手動建立適用於 SQL Server 稽核的 IAM 角色](#)
- [設定 IAM 許可權限以進行 RDS for Oracle 與 Amazon S3 的整合](#)
- [設定對 Amazon S3 儲存貯體的存取權 \(PostgreSQL 匯入\)](#)
- [設定對 Amazon S3 儲存貯體的存取權 \(PostgreSQL 匯出\)](#)

適用於 MariaDB、MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證

您可以使用 AWS Identity and Access Management (IAM) 資料庫身份驗證向資料庫執行個體進行驗證。IAM 資料庫身分驗證可搭配 MariaDB、MySQL 和 PostgreSQL 運作。透過此身分驗證方法，您連線至資料庫執行個體時不需要使用密碼。而是改用身分驗證字符。

身分驗證字符是 Amazon RDS 依請求而產生的唯一字元字串。使用 AWS 簽名版本 4 生成身份驗證令牌。每個字符的存留期為 15 分鐘。您不需要將使用者登入資料存放在資料庫，因為身分驗證是利用 IAM 在外部管理。您仍可使用標準資料庫身分驗證。字符僅用於身分驗證，並且在建立後不會影響工作階段。

IAM 資料庫身分驗證提供下列優點：

- 使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來加密往返資料庫的網路流量。如需使用 SSL/TLS 搭配 Amazon RDS 的詳細資訊，請參閱。
- 您可以使用 IAM 來集中管理資料庫資源的存取，而不需要在每個資料庫執行個體上個別地管理存取。
- 對於 Amazon EC2 上執行的應用程式，您可以使用 EC2 執行個體專用的設定檔登入資料 (而非密碼) 來存取資料庫，如此安全性更高。

通常，當應用程式每秒建立的連線少於 200 個，且您不希望直接以應用程式程式碼管理使用者名稱和密碼時，請考慮使用 IAM 資料庫身分驗證。

Amazon Web Services (AWS) JDBC 驅動程序支持 IAM 數據庫身份驗證。如需詳細資訊，請參閱 [Amazon Web Services 中的 AWS IAM 身份驗證外掛程式 \(AWS\) JDBC 驅動程式 GitHub 存放庫](#)。

Amazon Web Services (AWS) Python 驅動程序支持 IAM 數據庫身份驗證。如需詳細資訊，請參閱 [Amazon Web Services 中的 AWS IAM 身份驗證外掛程式 \(AWS\) Python 驅動程式 GitHub 存放庫](#)。

主題

- [區域和版本可用性](#)
- [CLI 和開發套件支援](#)
- [IAM 資料庫身分驗證的限制](#)
- [適用於 IAM 資料庫身分驗證的建議](#)
- [不支援的 AWS 全域條件內容鍵](#)
- [啟用和停用 IAM 資料庫身分驗證](#)

- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)
- [使用 IAM 身分驗證連接至資料庫執行個體](#)

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域而有所不同。如需 Amazon RDS 和 IAM 資料庫身分驗證的版本和區域可用性的詳細資訊，請參閱 [Amazon RDS 中的 IAM 資料庫身份驗證支援的區域和資料庫引擎](#)。

CLI 和開發套件支援

IAM 資料庫身份驗證適用 [AWS CLI](#) 於以下語言特定 AWS SDK：

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

IAM 資料庫身分驗證的限制

使用 IAM 資料庫身分驗證，會套用以下限制：

- IAM 資料庫身份驗證會在下列案例中限制連線：
 - 您每秒使用身份驗證令牌超過 20 個連接，每個連接都由不同的 IAM 身份簽名。
 - 您使用不同的身份驗證令牌每秒超過 200 個連接。

使用相同驗證 Token 的連線不會受到限制。我們建議您盡可能重複使用身份驗證令牌。

- 目前，IAM 資料庫驗證不支援所有全域條件內容金鑰。

如需有關全域條件內容索引鍵的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容索引鍵](#)。

- 對於 PostgreSQL，如果 IAM 角色 (rds_iam) 新增至使用者 (包括 RDS 主要使用者)，則 IAM 身分驗證優先於密碼身分驗證，因此使用者必須以 IAM 使用者身分登入。
- 對於 PostgreSQL，Amazon RDS 不支援同時啟用 IAM 和 Kerberos 身分驗證方法。
- 對於 PostgreSQL，您無法使用 IAM 身分驗證來建立複寫連線。
- 您無法使用自訂 Route 53 DNS 記錄，替代資料庫執行個體端點來產生身分驗證字符。
- CloudWatch 並且 CloudTrail 不要記錄 IAM 身份驗證。這些服務不會追蹤授權 IAM 角色以啟用資料庫連線的 generate-db-auth-token API 呼叫。如需詳細資訊，請參閱使用基於屬性的存取控制 [使用 Amazon RDS IAM 身份驗證實現可稽核性](#)。

適用於 IAM 資料庫身分驗證的建議

使用 IAM 資料庫身分驗證時，建議您採取以下做法：

- 當您的應用程式每秒需要少於 200 個新的 IAM 資料庫驗證連線時，請使用 IAM 資料庫驗證。

搭配 Amazon RDS 使用的資料庫引擎完全不限制每秒的身分驗證嘗試次數。不過，當您使用 IAM 資料庫身分驗證時，應用程式必須產生身分驗證字符。然後，應用程式會使用該字符來連接至資料庫執行個體。如果您超過每秒新連線數上限，IAM 資料庫身分驗證的額外負荷會導致連線調節。

請考慮在應用程式中使用連線集區，以緩解持續建立連線的問題。如此可以減少 IAM 資料庫身分驗證的額外負荷，並允許您的應用程式重複使用現有連線。或者，請考慮針對這些使用案例採用 RDS Proxy。RDS Proxy 需額外收費。請參閱 [RDS Proxy 定價](#)。

- IAM 資料庫身分驗證字符的大小取決於許多因素，包括 IAM 標籤的數量、IAM 服務政策、ARN 長度，以及其他 IAM 和資料庫屬性。此字符的最小大小一般約為 1 KB，但可以更大。因為此字符用作連線字串中的密碼，以使用 IAM 身分驗證連線至資料庫，所以您應該確保資料庫驅動程式 (例如 ODBC) 和/或任何工具不會限制或由於其大小而以其他方式截斷此字符。截斷的字符將導致資料庫和 IAM 執行的身分驗證失敗。
- 如果您是在建立 IAM 資料庫身分驗證字符時使用臨時憑證，則在使用 IAM 資料庫身分驗證字符發出連線請求時，臨時憑證必須仍然有效。

不支援的 AWS 全域條件內容鍵

IAM 資料庫身份驗證不支援下列 AWS 全域條件內容金鑰子集。

- aws:Referer
- aws:SourceIp

- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

如需詳細資訊，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

啟用和停用 IAM 資料庫身分驗證

依預設，資料庫執行個體上會停用 IAM 資料庫身分驗證。您可以使用 AWS Management Console、AWS CLI 或 API 來啟用或停用 IAM 資料庫身分驗證。

您可以在執行下列其中一個動作時啟用 IAM 資料庫身分驗證：

- 若要建立啟用 IAM 資料庫身分驗證的新資料庫執行個體，請參閱[建立 Amazon RDS 資料庫執行個體](#)。
- 若要修改資料庫執行個體以啟用 IAM 資料庫身分驗證，請參閱[修改 Amazon RDS 資料庫執行個體](#)。
- 如要從啟用 IAM 資料庫身分驗證的快照還原資料庫執行個體，請參閱[從資料庫快照還原](#)。
- 若要在啟用 IAM 資料庫身分驗證的情況下，將資料庫執行個體叢集還原至某個時間點，請參閱[將資料庫執行個體還原至指定的時間](#)。

PostgreSQL 資料庫執行個體的 IAM 身分驗證要求 SSL 值為 1。如果 SSL 值為 0，您即無法啟用 PostgreSQL 資料庫執行個體的 IAM 身分驗證。如果對於 PostgreSQL 資料庫執行個體啟用 IAM 身分驗證，則無法將 SSL 值變更為 0。

主控台

每個建立或修改工作流程都有一個 Database authentication (資料庫身分驗證) 區段，您可以在其中啟用或停用 IAM 資料庫身分驗證。在該區段中，選擇 Password and IAM database authentication (密碼和 IAM 資料庫身分驗證) 以啟用 IAM 資料庫身分驗證。

為現有資料庫執行個體啟用或停用 IAM 資料庫身分驗證

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要修改的資料庫執行個體。

Note

確認資料庫執行個體與 IAM 身分驗證相容。查看 [區域和版本可用性](#) 中的相容性要求。

4. 選擇 Modify (修改)。
5. 在 Database authentication (資料庫驗證) 區段中，選擇 Password and IAM database authentication (密碼和 IAM 資料庫身分驗證) 以啟用 IAM 資料庫身分驗證。選擇密碼身分驗證或密碼和 Kerberos 身分驗證以停用 IAM 身分驗證。
6. 選擇 Continue (繼續)。
7. 若要立即套用變更，請在 Scheduling of modifications (修改排程) 區段中選擇 Immediately (立即)。
8. 選擇 Modify DB instance (修改資料庫執行個體)。

AWS CLI

若要使用 AWS CLI 建立支援 IAM 身分驗證的新資料庫執行個體，請使用 [create-db-instance](#) 命令。指定 `--enable-iam-database-authentication` 選項，如以下範例所示。

```
aws rds create-db-instance \  
  --db-instance-identifier mydbinstance \  
  --db-instance-class db.m3.medium \  
  --engine MySQL \  
  --allocated-storage 20 \  
  --master-username masterawsuser \  
  --manage-master-user-password \  
  --enable-iam-database-authentication
```

若要將現有資料庫執行個體更新為進行或不進行 IAM 身分驗證，請使用 AWS CLI 命令 [modify-db-instance](#)。視需要指定 `--enable-iam-database-authentication` 或 `--no-enable-iam-database-authentication` 選項。

Note

確認資料庫執行個體與 IAM 身分驗證相容。查看 [區域和版本可用性](#) 中的相容性要求。

依預設，Amazon RDS 會在下一次維護時段執行修改。如果您不想等待，而希望儘快啟用 IAM 資料庫身分驗證，請使用 `--apply-immediately` 參數。

以下範例示範如何對現有的資料庫執行個體立即啟用 IAM 身分驗證。

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --apply-immediately \  
  --enable-iam-database-authentication
```

如果您要還原資料庫執行個體，請使用下列其中一個 AWS CLI 命令：

- [restore-db-instance-to-point-in-time](#)
- [restore-db-instance-from-db-snapshot](#)

IAM 資料庫身分驗證設定預設為來源快照的設定。若要變更此設定，請視需要設定 `--enable-iam-database-authentication` 或 `--no-enable-iam-database-authentication` 選項。

RDS API

若要使用 API 建立支援 IAM 身分驗證的新資料庫執行個體，請使用 API 操作 [CreateDBInstance](#)。將 `EnableIAMDatabaseAuthentication` 參數設為 `true`。

若要更新現有資料庫執行個體進行或不進行 IAM 身分驗證，請使用 API 操作 [ModifyDBInstance](#)。將 `EnableIAMDatabaseAuthentication` 參數設為 `true` 以啟用 IAM 身分驗證，或設為 `false` 表示停用。

Note

確認資料庫執行個體與 IAM 身分驗證相容。查看 [區域和版本可用性](#) 中的相容性要求。

如果您要還原資料庫執行個體，請使用下列其中一個 API 操作：

- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceToPointInTime](#)

IAM 資料庫身分驗證設定預設為來源快照的設定。若要變更此設定，請將 `EnableIAMDatabaseAuthentication` 參數設為 `true` 以啟用 IAM 身分驗證，或設為 `false` 表示停用。

建立並使用 IAM 政策進行 IAM 資料庫存取

若要允許使用者或角色連接資料庫執行個體，您必須建立 IAM 政策。然後，您可以將政策連接到許可集或角色。

Note

若要進一步了解 IAM 政策，請參閱 [Amazon RDS 的 Identity and access management](#)。

以下範例政策可讓使用者利用 IAM 資料庫身分驗證來連接資料庫執行個體。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds-db:connect"
      ],
      "Resource": [
        "arn:aws:rds-db:us-east-2:1234567890:dbuser:db-ABCDEFGHIJKL01234/db_user"
      ]
    }
  ]
}
```

Important

具有管理員許可的使用者可以存取資料庫執行個體，而不需 IAM 政策中的明確許可。如果您想要將管理員的存取權限制為資料庫執行個體，您可以建立具有適當、特殊權限較低的許可的 IAM 角色，並將其指派給管理員。

Note

請勿混淆 `rds-db:` 前綴和其他以 `rds:` 開頭的 RDS API 操作前綴。您只能對 IAM 資料庫身分驗證使用 `rds-db:` 字首和 `rds-db:connect` 動作。它們不適用於任何其他內容。

範例政策包含單一陳述式與下列元素：

- **Effect** – 指定 `Allow` 以授權存取資料庫執行個體。如果您未明確允許存取，預設將會拒絕存取。
- **Action** – 指定 `rds-db:connect` 以允許連線至資料庫執行個體。
- **Resource** – 指定 Amazon Resource Name (ARN) 以描述一個資料庫執行個體中的一個資料庫帳戶。ARN 格式如下。

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

請在此格式中更換下列項目：

- *region* 是資料庫執行個體的 AWS 區域。在此範例政策中，AWS 區域是 `us-east-2`。
- *account-id* 是資料庫執行個體的 AWS 帳戶號碼。在此範例政策中，帳戶號碼是 `1234567890`。使用者的帳戶必須與資料庫執行個體的帳戶相同。

若要執行跨帳戶存取權，請使用資料庫執行個體帳戶中，上述的政策來建立 IAM 角色，並允許您的其他帳戶擔任該角色。

- *DbiResourceId* 是資料庫執行個體的識別符。此為 AWS 區域的唯一識別符，且永不變更。在此範例政策中，識別符是 `db-ABCDEFGHIJKL01234`。

若要在 AWS Management Console 中尋找 Amazon RDS 的資料庫執行個體資源 ID，請選擇資料庫執行個體以查看其詳細資訊。然後選擇 Configuration (組態) 標籤。Resource ID (資源 ID) 顯示在 Configuration (組態) 區段中。

或是您可以使用 AWS CLI 命令，列出目前 AWS 區域中所有資料庫執行個體的識別符及資源 ID，如下所示。

```
aws rds describe-db-instances --query "DBInstances[*].  
[DBInstanceIdentifier,DbiResourceId]"
```

若您使用的是 Amazon Aurora，請指定 `DbClusterResourceId` 而非 `DbiResourceId`。如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的[建立並使用 IAM 政策進行 IAM 資料庫存取](#)。

Note

如果您是透過 RDS Proxy 連線至資料庫，請指定代理資源 ID，例如 `prx-ABCDEFGHIJKL01234`。如需搭配 RDS Proxy 使用 IAM 資料庫驗證的相關資訊，請參閱[使用 IAM 身分驗證連線到代理](#)。

- `db-user-name` 是要與 IAM 身分驗證產生關聯的資料庫帳戶的名稱。在範例政策中，資料庫帳戶是 `db_user`。

您可以建構其他 ARN 來支援各種存取模式。下列政策允許存取資料庫執行個體中的兩個不同資料庫帳戶：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds-db:connect"
      ],
      "Resource": [
        "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/jane_doe",
        "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/mary_roe"
      ]
    }
  ]
}
```

以下政策使用 "*" 字元，以符合特定 AWS 帳戶和 AWS 區域的所有資料庫執行個體和資料庫帳戶。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds-db:connect"
      ],
      "Resource": [
        "arn:aws:rds-db:us-east-2:1234567890:dbuser:*/*"
      ]
    }
  ]
}
```

以下政策符合特定 AWS 帳戶和 AWS 區域的所有資料庫執行個體。不過，此政策僅授權存取具有 jane_doe 資料庫帳戶的資料庫執行個體。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds-db:connect"
      ],
      "Resource": [
        "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"
      ]
    }
  ]
}
```

使用者或角色只能存取資料庫使用者可存取的資料庫。例如，假設資料庫執行個體有一個名為 dev 的資料庫，還有另一個名為 test 的資料庫。如果資料庫使用者 jane_doe 只能存取 dev，則任何以 jane_doe 使用者存取該資料庫執行個體的使用者或角色，也只能存取 dev。其他資料庫物件同樣受此存取限制，例如資料表和檢視等。

管理員必須建立 IAM 政策，授予實體在其所需的指定資源上執行特定 API 操作的許可。管理員接著必須將這些政策連接至需要這些許可的許可集或角色。如需政策範例，請參閱 [Amazon RDS 以身分為基礎的政策範例](#)。

將 IAM 政策連接到許可集或角色

在您建立 IAM 政策以允許資料庫身分驗證之後，您必須將該政策連接至許可集或角色。如需本主題的教學課程，請參閱《IAM 使用者指南》中的 [建立並連接您的第一個客戶受管原則](#)。

在您進行教學課程時，可使用本節所示的其中一個政策範例做為起點，並依您的需求進行自訂。在教學課程結束時，您會有一個具有連接政策且可使用 `rds-db:connect` 動作的許可集。

Note

您可以將多個許可集或角色映射至相同的資料庫使用者帳戶。例如，假設 IAM 政策指定下列資源 ARN。

```
arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/
jane_doe
```

如果您將此政策連接至 Jane、Bob 和 Diego，則每個使用者都能使用 `jane_doe` 資料庫帳戶連線至指定的資料庫執行個體。

使用 IAM 身分驗證建立資料庫帳戶

採用 IAM 資料庫身分驗證時，不需要指派資料庫密碼給您所建立的使用者帳戶。如果您移除已映射至資料庫帳戶的使用者，則應該使用 `DROP USER` 陳述式來一併移除該資料庫帳戶。

Note

用於 IAM 身分驗證的使用者名稱必須與資料庫中的使用者名稱大小寫相符。

主題

- [搭配 IAM 身分驗證使用 MariaDB and MySQL](#)
- [將 IAM 身分驗證搭配 PostgreSQL 使用](#)

搭配 IAM 身分驗證使用 MariaDB and MySQL

透過 MariaDB and MySQL，會由 `AWSAuthenticationPlugin` 處理身分驗證 – 這是 AWS 提供的外掛程式，可順暢搭配 IAM 來對使用者進行身分驗證。以主要使用者身分或可以建立使用者並授予權限的不同使用者身分連線至資料庫執行個體。連線後，發出 `CREATE USER` 陳述式，如下列範例所示。

```
CREATE USER jane_doe IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

`IDENTIFIED WITH` 子句可讓 MariaDB and MySQL 使用 `AWSAuthenticationPlugin` 來驗證資料庫帳戶 (`jane_doe`)。AS `'RDS'` 子句指的是身分驗證方法。確定指定的資料庫使用者名稱與 IAM 資料庫存取的 IAM 政策中的資源相同。如需更多詳細資訊，請參閱 [建立並使用 IAM 政策進行 IAM 資料庫存取](#)。

Note

如果您看見下列訊息，表示 AWS 提供的外掛程式不適用於目前的資料庫執行個體。

```
ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded
```

若要對此錯誤進行故障診斷，請確認您使用支援的組態，且已在資料庫執行個體上啟用 IAM 資料庫身分驗證。如需更多詳細資訊，請參閱 [區域和版本可用性](#) 及 [啟用和停用 IAM 資料庫身分驗證](#)。

使用 `AWSAuthenticationPlugin` 建立帳戶之後，管理此帳戶的方式就像管理其他資料庫帳戶一樣。例如，您可以使用 `GRANT` 和 `REVOKE` 陳述式來修改帳戶權限，或使用 `ALTER USER` 陳述式來修改各種帳戶屬性。

使用 IAM 時，資料庫網路流量會使用 SSL/TLS 加密。若要允許 SSL 連線，請透過下列命令修改使用者帳戶。

```
ALTER USER 'jane_doe'@'%' REQUIRE SSL;
```

將 IAM 身分驗證搭配 PostgreSQL 使用

若要搭配 PostgreSQL 使用 IAM 身分驗證，請以主要使用者身分或可以建立使用者並授予權限的不同使用者身分連線至資料庫執行個體。連線後，請建立資料庫使用者，然後將 `rds_iam` 角色授予他們，如下列範例所示。

```
CREATE USER db_userx;  
GRANT rds_iam TO db_userx;
```

確定指定的資料庫使用者名稱與 IAM 資料庫存取的 IAM 政策中的資源相同。如需更多詳細資訊，請參閱 [建立並使用 IAM 政策進行 IAM 資料庫存取](#)。

使用 IAM 身分驗證連接至資料庫執行個體

採用 IAM 資料庫身分驗證時，您會使用身分驗證字符來連接資料庫執行個體。身分驗證字符是可用來代替密碼的字元字串。產生身分驗證字符之後，過期之前的有效期限為 15 分鐘。如果您嘗試使用過期的字符來連接，則會拒絕連接請求。

每個身分驗證字符必須附帶有效的簽章，並使用 AWS 簽章第 4 版。如需詳細資訊，[請參閱 AWS 一般參考](#)。) AWS CLI 和 AWS SDK (例如 AWS SDK for Java 或 AWS SDK for Python (Boto3)) 可以自動簽署您建立的每個權杖。

當您從其他 AWS 服務 (例如) 連線到 Amazon RDS Amazon 時，您可以使用身份驗證令牌 AWS Lambda。使用字符就可以避免將密碼寫在程式碼中。或者，您可以使用 AWS SDK 以程式設計方式建立和以程式設計方式簽署驗證 Token。

取得已簽署的 IAM 身分驗證字符之後，您就可以連接至 Amazon RDS 資料庫執行個體。接下來，您可以了解如何使用命令行工具或 AWS SDK 來執行此操作，例如 AWS SDK for Java 或 AWS SDK for Python (Boto3)。

如需詳細資訊，請參閱下列部落格文章：

- [使用 IAM 身分驗證將 SQL Workbench/J 連線至 Aurora MySQL 或 Amazon RDS for MySQL](#)
- [Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS for PostgreSQL](#)

必要條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)

主題

- [使用與 AWS 驅動程式的 IAM 身份驗證連接到資料庫執行個體](#)
- [從命令列使用 IAM 身份驗證連接到您的資料庫執行個體：AWS CLI 和 mysql 用戶端](#)
- [連線至您的資料庫執行個體，方法為從命令列使用 IAM 身分驗證：AWS CLI 和 psql 用戶端](#)
- [使用 IAM 身分驗證和 AWS SDK for .NET 連線至資料庫執行個體](#)
- [使用 IAM 身分驗證和 AWS SDK for Go 連線至資料庫執行個體](#)
- [使用 IAM 身份驗證連接到您的資料庫執行個體AWS SDK for Java](#)
- [使用 IAM 身分驗證和 AWS SDK for Python \(Boto3\) 連線至資料庫執行個體](#)

使用與 AWS 驅動程式的 IAM 身份驗證連接到資料庫執行個體

驅動程式 AWS 套件的設計旨在提供更快的切換和容錯移轉時間，以及使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合身分進行身份驗證的支援。AWS 驅動程式仰賴監控資料庫執行個體狀態，並瞭解執行個體拓撲來判斷新的寫入器。這種方法可將切換和容錯移轉時間縮短為個位數秒，而開放原始碼驅動程式則需要數十秒。

如需有關 AWS 驅動程式的詳細資訊，請參閱適用於您的 RDS、[MySQL 版 RDS MariaDB 適用於 PostgreSQL 資料庫執行個體的 RDS](#) 的對應語言驅動程式。

Note

適用於 MariaDB 的 RDS 唯一支援的功能是使用 AWS Secrets Manager、AWS Identity and Access Management (IAM) 和聯合身分進行身分驗證。

從命令列使用 IAM 身份驗證連接到您的資料庫執行個體：AWS CLI 和 mysql 用戶端

您可以使用 AWS CLI 和命令列工具從命mysql令列連線到 Amazon RDS 資料庫執行個體，如下所述。

必要條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)

Note

有關使用 SQL Workbench/J 與 IAM 身分驗證連接到您的資料庫的資訊，請參閱部落格文章 [使用 IAM 身分驗證將 SQL Workbench/J 連線至 Aurora MySQL 或 Amazon RDS for MySQL](#)。

主題

- [產生 IAM 身分驗證字符](#)
- [連接至資料庫執行個體](#)

產生 IAM 身分驗證字符

以下範例顯示如何使用 AWS CLI 來取得已簽署的身分驗證字符。

```
aws rds generate-db-auth-token \  
  --hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \  
  --port 3306 \  
  --region us-west-2 \  
  --username jane_doe
```

範例中的參數如下：

- `--hostname` – 您想要存取之資料庫執行個體的主機名稱
- `--port` – 用於連線資料庫執行個體的連接埠號碼
- `--region` – 執行資料庫執行個體的 AWS 區域
- `--username` – 您想要存取的資料庫帳戶

字符開頭的前幾個字元看起來如下所示。

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?  
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

連接至資料庫執行個體

連線的一般格式如下所示。

```
mysql --host=hostName --port=portNumber --ssl-ca=full_path_to_ssl_certificate --enable-  
cleartext-plugin --user=userName --password=authToken
```

參數如下：

- `--host` – 您想要存取之資料庫執行個體的主機名稱
- `--port` – 用於連線資料庫執行個體的連接埠號碼
- `--ssl-ca` – 包含公有金鑰之 SSL 憑證檔案的完整路徑

如需 SSL/TLS 支援 MariaDB 的詳細資訊，請參閱 [將 SSL/TLS 與 MariaDB 資料庫執行個體搭配使用](#)。

如需 SSL/TLS 支援 MySQL 的詳細資訊，請參閱 [將 SSL/TLS 搭配 MySQL 資料庫執行個體使用](#)。

若要下載 SSL 憑證，請參閱。

- `--enable-cleartext-plugin` – 指定 `AWSAuthenticationPlugin` 必須用於此連線的值
如果您使用的是 MariaDB 用戶端，該 `--enable-cleartext-plugin` 選項不是必需的。
- `--user` – 您想要存取的資料庫帳戶
- `--password` – 已簽署的 IAM 身分驗證字符

身分驗證字符由數百個字元組成。在命令列上可能會顯得雜亂。此問題的一個解決方法是將字符儲存到環境變數，然後在連接時使用該變數。以下範例顯示此解決方法的一種型態。在範例中，`/sample_dir/` 是包含公有金鑰之 SSL 憑證檔案的完整路徑。

```
RDSHOST="mysqldb.123456789012.us-east-1.rds.amazonaws.com"  
TOKEN="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-  
west-2 --username jane_doe )"  
  
mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/global-bundle.pem --enable-  
cleartext-plugin --user=jane_doe --password=$TOKEN
```

使用 `AWSAuthenticationPlugin` 來連接時會以 SSL 保護連線。若要確認是否如此，請在 `mysql>` 命令提示中輸入下列命令。

```
show status like 'Ssl%';
```

輸出中的以下幾行顯示更多詳細資訊。

```
+-----+-----+
| Variable_name | Value
+-----+-----+
| ...          | ...
| Ssl_cipher   | AES256-SHA
+-----+-----+
| ...          | ...
| Ssl_version  | TLSv1.1
+-----+-----+
| ...          | ...
+-----+-----+
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

連線至您的資料庫執行個體，方法為從命令列使用 IAM 身分驗證：AWS CLI 和 psql 用戶端

您可以使用 AWS CLI 和 psql 命令列工具，從命令列連線至 Amazon RDS for PostgreSQL 資料庫執行個體，如下所述。

先決條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)

Note

如需有關使用 pgAdmin 搭配 IAM 身分驗證連線到資料庫的資訊，請參閱部落格文章 [使用 IAM 身分驗證與 pgAdmin Amazon Aurora PostgreSQL 或 Amazon RDS for PostgreSQL 連線](#)。

主題

- [產生 IAM 身分驗證字符](#)
- [連接至 Amazon RDS PostgreSQL 執行個體](#)

產生 IAM 身分驗證字符

身分驗證字符由數百個字元組成，因此在命令列上可能會顯得雜亂。此問題的一個解決方法是將字符儲存到環境變數，然後在連接時使用該變數。以下程式碼範例顯示如何使用 AWS CLI 以 `generate-db-auth-token` 命令取得簽署的身分驗證字符，然後在 `PGPASSWORD` 環境變數中儲存該字符。

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"  
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --  
region us-west-2 --username jane_doe )"
```

在範例中，`generate-db-auth-token` 命令的參數如下所示：

- `--hostname` – 您想要存取之資料庫執行個體的主機名稱
- `--port` – 用於連線資料庫執行個體的連接埠號碼
- `--region` – 執行資料庫執行個體的 AWS 區域
- `--username` – 您想要存取的資料庫帳戶

所產生字符開頭的前幾個字元看起來如下所示。

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?  
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

連接至 Amazon RDS PostgreSQL 執行個體

使用 `psql` 連線的一般格式如下所示。

```
psql "host=hostName port=portNumber sslmode=verify-full  
sslrootcert=full_path_to_ssl_certificate dbname=DBName user=userName  
password=authToken"
```

參數如下：

- `host` – 您想要存取之資料庫執行個體的主機名稱
- `port` – 用於連線資料庫執行個體的連接埠號碼
- `sslmode` – 要使用的 SSL 模式

當您使用 `sslmode=verify-full` 時，SSL 連線會根據 SSL 憑證中的端點來驗證資料庫執行個體端點。

- `sslrootcert` – 包含公有金鑰之 SSL 憑證檔案的完整路徑

如需更多詳細資訊，請參閱 [將 SSL 與 PostgreSQL 資料庫執行個體搭配使用](#)。

若要下載 SSL 憑證，請參閱。

- `dbname` – 您想要存取的資料庫
- `user` – 您想要存取的資料庫帳戶
- `password` – 已簽署的 IAM 身分驗證字符

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

以下範例示範使用 `psql` 進行連線。在範例中，`psql` 針對主機使用環境變數 `RDSHOST`，產生的字符則使用環境變數 `PGPASSWORD`。此外，`/sample_dir/` 是包含公有金鑰之 SSL 憑證檔案的完整路徑。

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"  
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --  
region us-west-2 --username jane_doe )"  
  
psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/global-  
bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

使用 IAM 身分驗證和 AWS SDK for .NET 連線至資料庫執行個體

您可以如下所述使用 AWS SDK for .NET，連線至 RDS for MariaDB、MySQL 或 PostgreSQL 資料庫執行個體。

先決條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)

範例

以下程式碼範例顯示如何產生身分驗證字符，然後用來連線至資料庫執行個體。

若要執行此程式碼範例，您必須從 AWS SDK for .NET 網站上取得 [AWS](#)。AWSSDK.CORE 和 AWSSDK.RDS 套件是必需的。若要連接到資料庫執行個體，請使用資料庫引擎的 .NET 資料庫連接器，例如 MariaDB 或 MySQL 的 MySqlConnection 或 PostgreSQL 的 Npgsql。

此程式碼連接到 MariaDB 或 MySQL 資料庫執行個體。視需要修改下列變數的值：

- `server` – 您想要存取之資料庫執行個體的端點
- `user` – 您想要存取的資料庫帳戶
- `database` – 您想要存取的資料庫
- `port` – 用於連線資料庫執行個體的連接埠號碼
- `SslMode` – 要使用的 SSL 模式

當您使用 `SslMode=Required` 時，SSL 連線會根據 SSL 憑證中的端點來驗證資料庫執行個體端點。

- `SslCa` – Amazon RDS 之 SSL 憑證的完整路徑

若要下載憑證，請參閱。

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

```
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
Amazon.RDS.Util.RDSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
"mysqldb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
            // for debug only Console.WriteLine("{0}\n", pwd); //this verifies the token is
generated

            MySqlConnection conn = new MySqlConnection($"server=mysqldb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;
conn.Open();

            // Define a query
            MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

            // Execute a query
            MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();

            // Read all rows and output the first column in each row
            while (mysqlDataRdr.Read())
                Console.WriteLine(mysqlDataRdr[0]);

            mysqlDataRdr.Close();
            // Close connection
            conn.Close();
        }
    }
}
```

此程式碼會連線至 PostgreSQL 資料庫執行個體。

視需要修改下列變數的值：

- Server – 您想要存取之資料庫執行個體的端點
- User ID – 您想要存取的資料庫帳戶
- Database – 您想要存取的資料庫
- Port – 用於連線資料庫執行個體的連接埠號碼
- SSL Mode – 要使用的 SSL 模式

當您使用 SSL Mode=Required 時，SSL 連線會根據 SSL 憑證中的端點來驗證資料庫執行個體端點。

- Root Certificate – Amazon RDS 之 SSL 憑證的完整路徑

若要下載憑證，請參閱。

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
                RDSAuthTokenGenerator.GenerateAuthToken("postgresmydb.123456789012.us-
                east-1.rds.amazonaws.com", 5432, "jane_doe");
            // for debug only Console.WriteLine("{0}\n", pwd); //this verifies the token is generated

            NpgsqlConnection conn = new
                NpgsqlConnection($"Server=postgresmydb.123456789012.us-east-1.rds.amazonaws.com;User
                Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Root
                Certificate=full_path_to_ssl_certificate");
            conn.Open();

            // Define a query
```



```
        NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM
pg_user", conn);

        // Execute a query
        NpgsqlDataReader dr = cmd.ExecuteReader();

        // Read all rows and output the first column in each row
        while (dr.Read())
            Console.WriteLine("{0}\n", dr[0]);

        // Close connection
        conn.Close();
    }
}
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

使用 IAM 身分驗證和 AWS SDK for Go 連線至資料庫執行個體

您可以如下所述使用 AWS SDK for Go，連線至 RDS for MariaDB、MySQL 或 PostgreSQL 資料庫執行個體。

先決條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)

範例

若要執行這些程式碼範例，您必須從 AWS SDK for Go 網站上取得 [AWS](#)。

視需要修改下列變數的值：

- dbName – 您想要存取的資料庫
- dbUser – 您想要存取的資料庫帳戶
- dbHost – 您想要存取之資料庫執行個體的端點

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

- `dbPort` – 用於連線資料庫執行個體的連接埠號碼
- `region` – 執行資料庫執行個體的 AWS 區域

此外，請確定範例程式碼中匯入的程式庫存在於您的系統上。

Important

本節中的範例使用下列程式碼來提供從本機環境存取資料庫的登入資料：

```
creds := credentials.NewEnvCredentials()
```

如果您要從 AWS 服務 (例如 Amazon EC2 或 Amazon ECS) 存取資料庫，您可以使用下列程式碼取代程式碼：

```
sess := session.Must(session.NewSession())
```

```
creds := sess.Config.Credentials
```

如果您進行此變更，請確定您新增了下列匯入：

```
"github.com/aws/aws-sdk-go/aws/session"
```

主題

- [使用 IAM 身分驗證和 AWS SDK for Go V2 進行連線](#)
- [使用 IAM 身分驗證和 AWS SDK for Go V1 進行連線。](#)

使用 IAM 身分驗證和 AWS SDK for Go V2 進行連線

使用 IAM 身分驗證和 AWS SDK for Go V2 連線至資料庫執行個體。

以下程式碼範例顯示如何產生身分驗證字符，然後用來連線至資料庫執行個體。

此程式碼連接到 MariaDB 或 MySQL 資料庫執行個體。

```
package main

import (
    "context"
    "database/sql"
```

```
"fmt"

"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/feature/rds/auth"
_ "github.com/go-sql-driver/mysql"
)

func main() {

    var dbName string = "DatabaseName"
    var dbUser string = "DatabaseUser"
    var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    var dbPort int = 3306
    var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
    var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

此程式碼會連線至 PostgreSQL 資料庫執行個體。

```
package main

import (
    "context"
    "database/sql"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
    _ "github.com/lib/pq"
)

func main() {

    var dbName string = "DatabaseName"
    var dbUser string = "DatabaseUser"
    var dbHost string = "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    var dbPort int = 5432
    var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
    var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authenticationToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
```

```
        panic(err)
    }
}
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

使用 IAM 身分驗證和 AWS SDK for Go V1 進行連線。

使用 IAM 身分驗證和 AWS SDK for Go V1 連線至資料庫執行個體

以下程式碼範例顯示如何產生身分驗證字符，然後用來連線至資料庫執行個體。

此程式碼連接到 MariaDB 或 MySQL 資料庫執行個體。

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )
}
```

```
db, err := sql.Open("mysql", dsn)
if err != nil {
    panic(err)
}

err = db.Ping()
if err != nil {
    panic(err)
}
}
```

此程式碼會連線至 PostgreSQL 資料庫執行個體。

```
package main

import (
    "database/sql"
    "fmt"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/lib/pq"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
```

```
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

使用 IAM 身分驗證連接到您的資料庫執行個體AWS SDK for Java

您可以使用以下所述連線到 RDS 適用於 MariaDB 資料庫、MySQL 或 PostgreSQL 資料庫執行。

AWS SDK for Java

必要條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)
- [使用 IAM 身分驗證建立資料庫帳戶](#)
- [設定適用於 Java 的 AWS 開發套件](#)

如需有關如何使用適用於 Java 2.x 的開發套件的範例，請參閱[使用適用於 Java 2.x 的開發套件的 Amazon RDS 範例](#)。


主題

- [產生 IAM 身分驗證字符](#)
- [手動建構 IAM 身分驗證字符](#)
- [連接至資料庫執行個體](#)

產生 IAM 身分驗證字符

如果您正在使用撰寫程式 AWS SDK for Java，您可以使用 `RdsIamAuthTokenGenerator` 類別取得已簽署的驗證權杖。使用此類別需要您提供 AWS 認證。要做到這一點，你創建一個 `DefaultAWSCredentialsProviderChain` 類的實例。

DefaultAWSCredentialsProviderChain 使用在 [預設認證提供者鏈結](#) 中找到的第一個 AWS 存取金鑰和秘密金鑰。如需有關 AWS 存取金鑰的詳細資訊，請參閱 [管理使用者的存取金鑰](#)。

 Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

建立 RdsIamAuthTokenGenerator 的執行個體之後，您可以呼叫 getAuthToken 方法來取得已簽署的字符。提供 AWS 區域、主機名稱、連接埠名稱和使用者名稱。以下程式碼範例示範如何執行此作業。

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

        String region = "us-west-2";
        String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        System.out.println(generateAuthToken(region, hostname, port, username));
    }

    static String generateAuthToken(String region, String hostName, String port, String
username) {

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
            .credentials(new DefaultAWSCredentialsProviderChain())
            .region(region)
            .build();

        String authToken = generator.getAuthToken(
            GetIamAuthTokenRequest.builder()
                .hostname(hostName)
                .port(Integer.parseInt(port))
```



```
        .userName(username)
        .build());

    return authToken;
}

}
```

手動建構 IAM 身分驗證字符

在 Java 中，產生身分驗證字符最簡單的方式就是使用 `RdsIamAuthTokenGenerator`。此類為您創建一個身份驗證令牌，然後使用 AWS 簽名版本 4 對其進行簽名。如需詳細資訊，請參閱 AWS 一般參考中的 [Signature 第 4 版簽署程序](#)。

不過，您也可以手動建構和簽署身分驗證字符，如下列程式碼範例所示。

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;

import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
    public static String canonicalURIPParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
```

```
public static String algorithm = "AWS4-HMAC-SHA256";
public static String serviceName = "rds-db";
public static String requestWithoutSignature;

public static void main(String[] args) throws Exception {

    String region = "us-west-2";
    String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
    String port = "3306";
    String username = "jane_doe";

    Date now = new Date();
    String date = new SimpleDateFormat("yyyyMMdd").format(now);
    String dateTimeStamp = new
SimpleDateFormat("yyyyMMdd'T'HHmmss'Z']").format(now);
    DefaultAWSCredentialsProviderChain creds = new
DefaultAWSCredentialsProviderChain();
    String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
    String awsSecretKey = creds.getCredentials().getAWSSecretKey();
    String expiryMinutes = "900";

    System.out.println("Step 1: Create a canonical request:");
    String canonicalString = createCanonicalString(username, awsAccessKey, date,
dateTimeStamp, region, expiryMinutes, instanceName, port);
    System.out.println(canonicalString);
    System.out.println();

    System.out.println("Step 2: Create a string to sign:");
    String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
awsAccessKey, date, region);
    System.out.println(stringToSign);
    System.out.println();

    System.out.println("Step 3: Calculate the signature:");
    String signature = BinaryUtils.toHex(calculateSignature(stringToSign,
newSigningKey(awsSecretKey, date, region, serviceName)));
    System.out.println(signature);
    System.out.println();

    System.out.println("Step 4: Add the signing info to the request");

    System.out.println(appendSignature(signature));
    System.out.println();
```

```
}

//Step 1: Create a canonical request date should be in format YYYYMMDD and dateTime
should be in format YYYYMMDDTHHMSSZ
public static String createCanonicalString(String user, String accessKey, String
date, String dateTime, String region, String expiryPeriod, String hostName, String
port) throws Exception {
    canonicalQueryParameters.put("Action", action);
    canonicalQueryParameters.put("DBUser", user);
    canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
    canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date +
"%2F" + region + "%2F" + serviceName + "%2Faws4_request");
    canonicalQueryParameters.put("X-Amz-Date", dateTime);
    canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
    canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
    String canonicalQueryString = "";
    while(!canonicalQueryParameters.isEmpty()) {
        String currentQueryParameter = canonicalQueryParameters.firstKey();
        String currentQueryParameterValue =
canonicalQueryParameters.remove(currentQueryParameter);
        canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
currentQueryParameterValue;
        if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
            canonicalQueryString += "&";
        }
    }
    String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
    requestWithoutSignature = hostName + ":" + port + "/" + canonicalQueryString;

    String hashedPayload = BinaryUtils.toHex(hash(payload));
    return httpMethod + '\n' + canonicalURIPParameter + '\n' + canonicalQueryString
+ '\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;
}

//Step 2: Create a string to sign using sig v4
public static String createStringToSign(String dateTime, String canonicalRequest,
String accessKey, String date, String region) throws Exception {
    String credentialScope = date + "/" + region + "/" + serviceName + "/"
aws4_request";
    return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
BinaryUtils.toHex(hash(canonicalRequest));
}
}
```

```
//Step 3: Calculate signature
/**
 * Step 3 of the &AWS; Signature version 4 calculation. It involves deriving
 * the signing key and computing the signature. Refer to
 * http://docs.aws.amazon
 * .com/general/latest/gr/sigv4-calculate-signature.html
 */
public static byte[] calculateSignature(String stringToSign,
                                       byte[] signingKey) {
    return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
               SigningAlgorithm.HmacSHA256);
}

public static byte[] sign(byte[] data, byte[] key,
                          SigningAlgorithm algorithm) throws SdkClientException {
    try {
        Mac mac = algorithm.getMac();
        mac.init(new SecretKeySpec(key, algorithm.toString()));
        return mac.doFinal(data);
    } catch (Exception e) {
        throw new SdkClientException(
            "Unable to calculate a request signature: "
            + e.getMessage(), e);
    }
}

public static byte[] newSigningKey(String secretKey,
                                    String dateStamp, String regionName, String
serviceName) {
    byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
    byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
    byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
    byte[] kService = sign(serviceName, kRegion,
                           SigningAlgorithm.HmacSHA256);
    return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
}

public static byte[] sign(String stringData, byte[] key,
                          SigningAlgorithm algorithm) throws SdkClientException {
    try {
        byte[] data = stringData.getBytes(UTF8);
        return sign(data, key, algorithm);
    } catch (Exception e) {
```

```
        throw new SdkClientException(
            "Unable to calculate a request signature: "
                + e.getMessage(), e);
    }
}

//Step 4: append the signature
public static String appendSignature(String signature) {
    return requestWithoutSignature + "&X-Amz-Signature=" + signature;
}

public static byte[] hash(String s) throws Exception {
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        md.update(s.getBytes(UTF8));
        return md.digest();
    } catch (Exception e) {
        throw new SdkClientException(
            "Unable to compute hash while signing request: "
                + e.getMessage(), e);
    }
}
}
```

連接至資料庫執行個體

以下程式碼範例顯示如何產生身分驗證權杖，然後用來連接至執行 MariaDB 或 MySQL 的執行個體。

若要執行此程式碼範例，您需要在[AWS SDK for Java](#) AWS 網站上找到的。此外，您還需要下列項目：

- MySQL Connector/J。本程式碼範本經過 `mysql-connector-java-5.1.33-bin.jar` 的測試。
- 適用於某個 AWS 區域的 Amazon RDS 的中繼憑證。(如需詳細資訊，請參閱「」。) 在執行時間，類別載入器會在此 Java 程式碼範例所在的同一個目錄中尋找憑證，因此類別載入器可以找到憑證。
- 視需要修改下列變數的值：
 - `RDS_INSTANCE_HOSTNAME` – 您想要存取之資料庫執行個體的主機名稱。
 - `RDS_INSTANCE_PORT` – 用於連線至 PostgreSQL 資料庫執行個體的連接埠號碼。
 - `REGION_NAME`— 執行資料庫執行個體的 AWS 區域。
 - `DB_USER` – 您想要存取的資料庫帳戶。
 - `SSL_CERTIFICATE`— 適用於某個 AWS 區域的 Amazon RDS 的 SSL 證書。

若要下載適用於 AWS 區域的憑證，請參閱。將 SSL 憑證放入此 Java 程式檔案所在的同一個目錄中，讓類別載入器在執行時間可以找到憑證。

此程式碼範例會從[預設認證提供者鏈結](#)取得 AWS 認證。

Note

為 DEFAULT_KEY_STORE_PASSWORD 指定此處所顯示提示以外的密碼，作為安全最佳實務。

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //AWS; Credentials of the IAM user with policy enabling IAM Database Authenticated
    access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
    DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
    creds.getCredentials().getAWSAccessKeyId();
```

```
private static final String AWS_SECRET_KEY =
creds.getCredentials().getAWSSecretKey();

//Configuration parameters for the generation of the IAM Database Authentication
token
private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
private static final int RDS_INSTANCE_PORT = 3306;
private static final String REGION_NAME = "us-west-2";
private static final String DB_USER = "jane_doe";
private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME +
":" + RDS_INSTANCE_PORT;

private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

private static final String KEY_STORE_TYPE = "JKS";
private static final String KEY_STORE_PROVIDER = "SUN";
private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-
cacerts";
private static final String KEY_STORE_FILE_SUFFIX = ".jks";
private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";

public static void main(String[] args) throws Exception {
    //get the connection
    Connection connection = getDBConnectionUsingIam();

    //verify the connection is successful
    Statement stmt= connection.createStatement();
    ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
    while (rs.next()) {
        String id = rs.getString(1);
        System.out.println(id); //Should print "Success!"
    }

    //close the connection
    stmt.close();
    connection.close();

    clearSslProperties();
}

/**
```

```

    * This method returns a connection to the db instance authenticated using IAM
Database Authentication
    * @return
    * @throws Exception
    */
private static Connection getDBConnectionUsingIam() throws Exception {
    setSslProperties();
    return DriverManager.getConnection(JDBC_URL, setMySQLConnectionProperties());
}

/**
 * This method sets the mysql connection properties which includes the IAM Database
Authentication token
 * as the password. It also specifies that SSL verification is required.
 * @return
 */
private static Properties setMySQLConnectionProperties() {
    Properties mysqlConnectionProperties = new Properties();
    mysqlConnectionProperties.setProperty("verifyServerCertificate", "true");
    mysqlConnectionProperties.setProperty("useSSL", "true");
    mysqlConnectionProperties.setProperty("user", DB_USER);
    mysqlConnectionProperties.setProperty("password", generateAuthToken());
    return mysqlConnectionProperties;
}

/**
 * This method generates the IAM Auth Token.
 * An example IAM Auth Token would look like follows:
 * btusi123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?
Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-
Credential=AKIAPFXHGVDI5RNF04AQ%2F20171003%2Fcn-north-1%2Frds-db%2Faws4_request&X-Amz-
Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfd1322eed15483b
    * @return
    */
private static String generateAuthToken() {
    BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
AWS_SECRET_KEY);

    RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
        .credentials(new
AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
    return generator.getAuthToken(GetIamAuthTokenRequest.builder())

```



```
.hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
}

/**
 * This method sets the SSL properties which specify the key store file, its type
and password:
 * @throws Exception
 */
private static void setSslProperties() throws Exception {
    System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
    System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
    System.setProperty("javax.net.ssl.trustStorePassword",
DEFAULT_KEY_STORE_PASSWORD);
}

/**
 * This method returns the path of the Key Store File needed for the SSL
verification during the IAM Database Authentication to
 * the db instance.
 * @return
 * @throws Exception
 */
private static String createKeyStoreFile() throws Exception {
    return createKeyStoreFile(createCertificate()).getPath();
}

/**
 * This method generates the SSL certificate
 * @return
 * @throws Exception
 */
private static X509Certificate createCertificate() throws Exception {
    CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
    URL url = new File(SSL_CERTIFICATE).toURI().toURL();
    if (url == null) {
        throw new Exception();
    }
    try (InputStream certInputStream = url.openStream()) {
        return (X509Certificate) certFactory.generateCertificate(certInputStream);
    }
}

/**
```

```
* This method creates the Key Store File
* @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
* @return
* @throws Exception
*/
private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
Exception {
    File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
KEY_STORE_FILE_SUFFIX);
    try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
        KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
        ks.load(null);
        ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
        ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
    }
    return keyStoreFile;
}

/**
* This method clears the SSL properties.
* @throws Exception
*/
private static void clearSslProperties() throws Exception {
    System.clearProperty("javax.net.ssl.trustStore");
    System.clearProperty("javax.net.ssl.trustStoreType");
    System.clearProperty("javax.net.ssl.trustStorePassword");
}
}
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

使用 IAM 身分驗證和 AWS SDK for Python (Boto3) 連線至資料庫執行個體

您可以如下所述使用 AWS SDK for Python (Boto3)，連線至 RDS for MariaDB、MySQL 或 PostgreSQL 資料庫執行個體。

先決條件

以下是使用 IAM 身分驗證連線至資料庫執行個體的先決條件：

- [啟用和停用 IAM 資料庫身分驗證](#)
- [建立並使用 IAM 政策進行 IAM 資料庫存取](#)

• [使用 IAM 身分驗證建立資料庫帳戶](#)

此外，請確定範例程式碼中匯入的程式庫存在於您的系統上。

範例

程式碼範例使用設定檔進行共用登入資料。如需指定登入資料的相關資訊，請參閱 AWS SDK for Python (Boto3) 文件中的[登入資料](#)。

以下程式碼範例顯示如何產生身分驗證字符，然後用來連線至資料庫執行個體。

若要執行此程式碼範例，您必須從 AWS SDK for Python (Boto3) 網站上取得 [AWS](#)。

視需要修改下列變數的值：

- ENDPOINT – 您想要存取之資料庫執行個體的端點
- PORT – 用於連線資料庫執行個體的連接埠號碼
- USER – 您想要存取的資料庫帳戶
- REGION – 執行資料庫執行個體的 AWS 區域
- DBNAME – 您想要存取的資料庫
- SSLCERTIFICATE – Amazon RDS 之 SSL 憑證的完整路徑

對於 `ssl_ca`，指定 SSL 憑證。若要下載 SSL 憑證，請參閱。

Note

您無法使用自訂 Route 53 DNS 記錄替代資料庫執行個體端點來產生身分驗證字符。

此程式碼連接到 MariaDB 或 MySQL 資料庫執行個體。

在執行此程式碼之前，請遵循 [Python 套件索引](#) 中的指示，來安裝 PyMySQL 驅動程式。

```
import pymysql
import sys
import boto3
import os
```

```
ENDPOINT="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
    Region=REGION)

try:
    conn = pymysql.connect(host=ENDPOINT, user=USER, passwd=token, port=PORT,
        database=DBNAME, ssl_ca='SSLCERTIFICATE')
    cur = conn.cursor()
    cur.execute("SELECT now()")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

此程式碼會連線至 PostgreSQL 資料庫執行個體。

在執行此程式碼之前，請按照 [Psycopg 文件](#) 中的指示來安裝 psycopg2。

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')
```

```
token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
    Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USER,
        password=token, sslrootcert="SSLCERTIFICATE")
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

如果要透過 Proxy 連線到資料庫執行個體，請參閱 [使用 IAM 身分驗證連線到代理](#)。

對 Amazon RDS 身分與存取進行故障診斷

請使用以下資訊來協助您診斷和修復使用 Amazon RDS 和 IAM 時發生的常見問題。

主題

- [我未獲授權在 Amazon RDS 中執行動作](#)
- [我未獲授權執行 iam:PassRole](#)
- [我想允許 AWS 帳戶外的人員存取我的 Amazon RDS 資源](#)

我未獲授權在 Amazon RDS 中執行動作

若 AWS Management Console 告知您並未獲得執行動作的授權，您必須聯絡您的管理員以取得協助。您的管理員是您的登入憑證提供者。

以下範例錯誤會在 mateojackson 使用者嘗試使用主控台檢視 *Widget* 的詳細資訊，但卻沒有 `rds:GetWidget` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
    rds:GetWidget on resource: my-example-widget
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-widget* 動作存取 `rds:GetWidget` 資源。

我未獲授權執行 iam:PassRole

若您收到錯誤，告知您並未獲得執行 iam:PassRole 動作的授權，您必須聯絡您的管理員以取得協助。您的管理員是您的登入憑證提供者。要求該人員更新您的政策，允許您將角色傳遞給 Amazon RDS。

有些 AWS 服務允許您傳遞現有的角色至該服務，而無須建立新的服務角色或服務連結角色。若要執行此作業，您必須擁有將角色傳遞至該服務的許可。

以下範例錯誤會在名為 marymajor 的使用者嘗試使用主控台在 Amazon RDS 中執行動作時發生。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 會請求管理員更新她的政策，允許她執行 iam:PassRole 動作。

我想允許 AWS 帳戶外的人員存取我的 Amazon RDS 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任對象取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您資源的許可。

若要進一步了解，請參閱以下內容：

- 若要了解 Amazon RDS 是否支援這些功能，請參閱 [Amazon RDS 如何搭配 IAM 運作](#)。
- 若要了解如何對您擁有的所有 AWS 帳戶提供資源的存取權，請參閱《IAM 使用者指南》中的 [對您所擁有的另一個 AWS 帳戶中的 IAM 使用者提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的 [將存取權提供給第三方擁有的 AWS 帳戶](#)。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的 [將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 [IAM 角色與資源型政策的差異](#)。

Amazon RDS 中的記錄和監控

監控是維護 Amazon RDS 及您 AWS 解決方案可靠性、可用性和效能的重要部分。您應該從 AWS 解決方案各個部分收集監控資料，以便在發生多點故障時，可更輕鬆地偵錯。AWS 提供多種工具，讓您監控 Amazon RDS 資源及回應潛在的事件：

Amazon CloudWatch 警報

使用 Amazon CloudWatch 警示，您可以在指定的時間段內觀看單一指標。如果指標超過指定臨界值，則會向 Amazon SNS 主題或 AWS Auto Scaling 政策傳送通知。CloudWatch 警示不會叫用動作，因為它們處於特定狀態。必須是狀態已變更並維持了所指定的時間長度，才會呼叫動作。

AWS CloudTrail 日誌

CloudTrail 提供使用者、角色或 AWS 服務在 Amazon RDS Amazon 中所採取的動作記錄。CloudTrail 擷取 Amazon RDS Amazon 的所有 API 呼叫做為事件，包括來自主控台的呼叫，以及從程式碼呼叫到 Amazon RDS API 操作。使用收集的資訊 CloudTrail，您可以判斷向 Amazon RDS Amazon 發出的請求、提出請求的來源 IP 地址、提出請求的人員、提出請求的時間以及其他詳細資訊。如需詳細資訊，請參閱 [在 AWS CloudTrail 中監控 Amazon RDS API 呼叫](#)。

Enhanced Monitoring (增強型監控)

Amazon RDS 可針對資料庫執行個體執行所在的作業系統 (OS) 即時提供指標。您可以使用主控台檢視資料庫執行個體的指標，或在您選擇的監控系統中使用 Amazon CloudWatch Logs 的增強型監控 JSON 輸出。如需詳細資訊，請參閱 [使用增強型監控來監控作業系統指標](#)。

Amazon RDS 績效詳情

績效詳情會延伸現有 Amazon RDS 監控功能的基礎，藉此說明資料庫效能，並幫助您分析可能影響效能的任何問題。利用績效詳情儀表板，您可以將資料庫負載視覺化，並依等候、SQL 陳述式、主機或使用者篩選負載。如需更多詳細資訊，請參閱 [在 Amazon RDS 上使用績效詳情監控資料庫負載](#)。

資料庫日誌

您可以使用 AWS Management Console、AWS CLI、或者 RDS API 來檢視、下載並查看資料庫日誌。如需更多詳細資訊，請參閱 [監控 Amazon RDS 日誌檔案](#)。

Amazon RDS 建議事項

Amazon RDS 提供自動化的資料庫資源建議。這些建議事項會透過分析資料庫執行個體、組態、用量及效能資料以提供最佳實務指南。如需詳細資訊，請參閱 [檢視和回應 Amazon RDS 建議](#)。

Amazon RDS 事件通知

Amazon RDS 使用 Amazon Simple Notification Service (Amazon SNS) 在 Amazon RDS 事件發生時提供通知。這些通知可以採用 Amazon SNS 在 AWS 區域中支援的任何通知形式，例如電子郵件、文字訊息或呼叫 HTTP 端點。如需更多詳細資訊，請參閱 [使用 Amazon RDS 事件通知](#)。

AWS Trusted Advisor

為成千上萬 AWS 客戶提供服務的過程中，學習到的最佳實務，都體現在 Trusted Advisor 中。Trusted Advisor 可檢查您的 AWS 環境，並在有可能節省成本、提升系統可用性與效能或填補安全漏洞時向您提出建議。所有 AWS 客戶都能存取五項 Trusted Advisor 檢查。商業或企業支援方案客戶，可以檢視所有 Trusted Advisor 檢查。

Trusted Advisor 具有下列 Amazon RDS 相關檢查：

- Amazon RDS 閒置資料庫執行個體
- Amazon RDS 安全群組存取風險
- Amazon RDS 備份
- Amazon RDS 多個可用區域

如需這些檢查的詳細資訊，請參閱 [Trusted Advisor 最佳實務 \(檢查\)](#)。

如需 Amazon RDS 監控的詳細資訊，請參閱 [監控 Amazon RDS 執行個體中的指標](#)。

Amazon RDS 的合規驗證

在多個 AWS 合規計劃中，第三方稽核人員會評估 Amazon RDS 的安全與合規。這些計劃包括 SOC、PCI、FedRAMP、HIPAA 等等。

如需特定合規計劃範圍內的 AWS 服務清單，請參閱[合規計劃內的 AWS 服務](#)。如需一般資訊，請參閱[AWS 合規計劃](#)。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱[在 AWS Artifact 中下載報告](#)。

您在使用 Amazon RDS 時的合規責任，取決於資料的敏感性、您組織的合規目標，以及適用的法律和法規。AWS 會提供以下資源協助您處理合規事宜：

- [安全與合規快速入門指南](#) – 這些部署指南討論在 AWS 上部署以安全及合規為重心基準環境的架構考量和步驟。
- [Amazon Web Services 的 HIPAA 安全與合規架構](#) - 本白皮書說明公司可如何運用 AWS 來建立符合 HIPAA 規範的應用程式。
- [AWS 合規資源](#) – 這組手冊和指南可能適用於您的產業和位置。
- [AWS Config](#) – 此 AWS 服務可評定資源組態與內部實務、業界準則和法規的合規狀態。
- [AWS Security Hub](#) – 此 AWS 服務 可供您全面檢視 AWS 中的安全狀態。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的合規是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub 控制參考](#)。

Amazon RDS 的復原功能

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS 區域提供多個分開且隔離的實際可用區域，它們以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域與可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

除了 AWS 全球基礎設施外，Amazon RDS 還提供支援資料復原和備份需求的多項功能。

備份和還原

Amazon RDS 建立並儲存您的資料庫執行個體的自動備份。Amazon RDS 會建立資料庫執行個體的儲存體磁碟區快照，因此會備份整個資料庫執行個體，而不只是個別的資料庫。

Amazon RDS 會在資料庫執行個體的備份時段建立資料庫執行個體的自動備份。Amazon RDS 會根據您指定的備份保留期儲存資料庫執行個體的自動備份。如有需要，您可將資料庫復原至備份保留期間的任何時間點。您也可以手動建立資料庫快照，以手動備份資料庫執行個體。

如果來源資料庫執行個體失敗，您可透過還原由此資料庫快照建立資料庫執行個體，做為災難復原解決方案。

如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

複寫

Amazon RDS 會使用 MariaDB、MySQL、Oracle 與 PostgreSQL 資料庫引擎的內建複寫功能，從來源資料庫執行個體建立一種特殊類型的資料庫執行個體，稱為僅供讀取複本。對來源資料庫執行個體所做的更新，會以非同步方式複製到僅供讀取複本。您可以將讀取查詢從應用程式路由到僅供讀取複本，以減少來源資料庫執行個體的負載。使用僅供讀取複本，您可以彈性地擴展超越單一資料庫執行個體的容量限制，以處理高讀取量的資料庫工作負載。您可以在來源資料庫執行個體失敗時，提升僅供讀取複本為獨立的執行個體，以做為災難復原解決方案。針對某些資料庫引擎，Amazon RDS 也支援其他複寫選項。

如需更多詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。

容錯移轉

使用異地同步備份部署，Amazon RDS 可為資料庫執行個體提供高可用性和容錯移轉支援。Amazon RDS 使用幾個不同技術來提供容錯移轉支援。Oracle、PostgreSQL、MySQL 和 MariaDB 資料庫執行

個體的異地同步備份部署是採用 Amazon 的容錯移轉來實作，而 SQL Server 資料庫執行個體則會使用 SQL Server 資料庫鏡像 (DBM) 技術。

如需更多詳細資訊，請參閱 [設定及管理多可用區部署](#)。

Amazon RDS 中的基礎設施安全

作為受管服務，Amazon Relational Database Service 受到 AWS 全球網路安全的保護。如需有關 AWS 安全服務以及 AWS 如何保護基礎設施的詳細資訊，請參閱 [AWS 雲端安全](#)。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱安全性支柱 AWS 架構良好的框架中的 [基礎設施保護](#)。

您會使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon RDS。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密 (PFS) 的密碼套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取索引鍵來簽署。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

此外，Amazon RDS 提供有助於支援基礎設施安全的功能。

安全群組

安全群組會控制進出資料庫執行個體流量的存取權限。資料庫執行個體的網路存取預設是關閉的。您可以在允許從 IP 地址範圍、連接埠或安全群組存取的安全群組中指定規則。設定傳入規則後，相同規則就會套用到與該安全群組相關聯的所有資料庫執行個體。

如需更多詳細資訊，請參閱 [使用安全群組控制存取](#)。

Public accessibility (公開存取性)

當您根據 Amazon VPC 服務啟動虛擬私有雲端 (VPC) 內的資料庫執行個體時，您可以開啟或關閉該資料庫執行個體的公開存取性。請使用 Public accessibility (公開存取權限) 參數，來指定您建立的資料庫執行個體是否有解析為公有 IP 地址的 DNS。使用此參數，您就可以指定資料庫執行個體是否可供公開存取。您可以修改 Public accessibility (公開存取權限) 參數，藉此修改資料庫執行個體設定，以開啟或關閉公開存取權限。

如需更多詳細資訊，請參閱 [在 VPC 中的網際網路中隱藏資料庫執行個體](#)。

Note

如果您的資料庫執行個體位於 VPC 中，但無法公開存取，您也可以使用 AWS Site-to-Site VPN 連接或 AWS Direct Connect 連線從私有網路存取。如需更多詳細資訊，請參閱 [網際網路流量隱私權](#)。

Amazon RDS API 和界面 VPC 端點 (AWS PrivateLink)

您可以建立界面 VPC 端點，以在您的 VPC 與 Amazon RDS API 端點之間建立私有連線。界面端點是採用 [AWS PrivateLink](#) 技術。

AWS PrivateLink 可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線的情況下私有存取 Amazon RDS API 操作。VPC 中的資料庫執行個體不需要公有 IP 地址，就能與 Amazon RDS API 端點進行通訊，以啟動、修改或終止資料庫執行個體。您的資料庫執行個體也不需要公有 IP 地址，就能使用任何可用的 RDS API 操作。您的 VPC 與 Amazon RDS 之間的流量，都會在 Amazon 網路的範圍內。

每個界面端點都是由您子網路中的一或多個彈性網路界面表示。如需彈性網路界面的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [彈性網路界面](#)。

如需 VPC 端點的詳細資訊，請參閱 Amazon VPC [使用者指南中的介面虛擬私人雲端端點 \(AWS PrivateLink\)](#)。如需 RDS API 操作的資訊，請參閱 [Amazon RDS API 參考](#)。

您連線至資料庫執行個體時不需要界面 VPC 端點。如需詳細資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

VPC 端點的考量事項

在設定 Amazon RDS API 端點的界面 VPC 端點前，請務必檢閱《Amazon VPC 使用者指南》中的 [界面端點屬性和限制](#)。

所有與管理 Amazon RDS 資源相關的 RDS API 操作都從使用 AWS PrivateLink 的 VPC 提供。

RDS API 端點支援 VPC 端點政策。根據預設，允許透過端點對 RDS API 操作進行完整存取。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [使用 VPC 端點控制對服務的存取](#)。

可用性

Amazon RDS API 目前支援下列 AWS 區域的 VPC 人雲端節點：

- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)
- 非洲 (開普敦)

- 亞太區域 (香港)
- 亞太區域 (孟買)
- 亞太區域 (大阪)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太區域 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 加拿大西部 (卡加利)
- 中國 (北京)
- 中國 (寧夏)
- 歐洲 (法蘭克福)
- 歐洲 (蘇黎世)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- 歐洲 (巴黎)
- 歐洲 (斯德哥爾摩)
- 歐洲 (米蘭)
- 以色列 (特拉維夫)
- Middle East (Bahrain)
- 南美洲 (聖保羅)
- AWS GovCloud (美國東部)
- AWS GovCloud (美國西部)

為 Amazon RDS API 建立界面 VPC 端點

您可以使用 Amazon 虛擬私人雲端主控台或 AWS Command Line Interface (AWS CLI) 為 Amazon RDS API 建立 VPC 端點。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [建立介面端點](#)。

使用服務名稱 `com.amazonaws.region.rds`，為 Amazon RDS API 建立 VPC 端點。

不包括中國境內的區 AWS 域，如果您為端點啟用私有 DNS，則可以使用該 AWS 區域的預設 DNS 名稱向 Amazon RDS 發出 API 請求。`rds.us-east-1.amazonaws.com` 對於中國 (北京) 和中國

(寧夏) AWS 區域，您可以分別使用 `rd-api.cn-north-1.amazonaws.com.cn` 和 `rd-api.cn-northwest-1.amazonaws.com.cn` 向 VPC 端點提出 API 要求。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [透過介面端點存取服務](#)。

為 Amazon RDS API 建立 VPC 端點政策

您可以將端點政策連接至控制 Amazon RDS API 存取權限的 VPC 端點。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [使用 VPC 端點控制對服務的存取](#)。

範例：Amazon RDS API 動作的 VPC 端點政策

以下是 Amazon RDS API 端點政策的範例。連接至端點後，此政策會針對所有資源上的所有委託人，授予列出的 Amazon RDS API 動作的存取權限。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBInstance",
        "rds:ModifyDBInstance",
        "rds:CreateDBSnapshot"
      ],
      "Resource": "*"
    }
  ]
}
```

範例：拒絕來自指定帳戶的所有存取的 VPC 端點策略 AWS

下列 VPC 端點策略拒絕 AWS 帳戶 123456789012 所有使用端點存取資源的帳戶。此政策允許來自其他帳戶的所有動作。

```
{
```



```
"Statement": [
  {
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  },
  {
    "Action": "*",
    "Effect": "Deny",
    "Resource": "*",
    "Principal": { "AWS": [ "123456789012" ] }
  }
]
```

Amazon RDS 的安全最佳實務

使用 AWS Identity and Access Management (IAM) 帳戶來控制對 Amazon RDS API 操作的存取，尤其是建立、修改或刪除 Amazon RDS Amazon 資源的操作。這類資源包含資料庫執行個體、安全群組和參數群組。同時也請使用 IAM 控制執行常見管理動作的動作，例如備份和還原資料庫執行個體。

- 為管理 Amazon RDS 資源的每個人 (包括您自己) 建立個別使用者。請勿使用 AWS 根登入資料來管理 Amazon RDS 資源。
- 授予每個使用者執行其職責所需最低程度的許可。
- 使用 IAM 群組來有效管理多個使用者的許可。
- 定期輪替您的 IAM 登入資料。
- 設定 AWS Secrets Manager 為自動輪替 Amazon RDS Amazon 的密碼。如需詳細資訊，請參閱 AWS Secrets Manager 使用指南中的 [旋轉 AWS Secrets Manager 密碼](#)。您也可以 AWS Secrets Manager 透過程式設計方式擷取憑證。如需更多詳細資訊，請參閱 AWS Secrets Manager 使用者指南中的 [擷取密碼值](#)。

如需 Amazon RDS 安全性的詳細資訊，請參閱 [Amazon RDS 中的安全](#)。如需關於 IAM 的詳細資訊，請參閱 [AWS Identity and Access Management](#)。如需 IAM 最佳實務的資訊，請參閱 [IAM 最佳實務](#)。

AWS Security Hub 使用安全控制來評估資源配置和安全標準，以幫助您遵守各種合規性框架。如需有關使用 Security Hub 評估 RDS 資源的詳細資訊，請參閱 [使用 AWS Security Hub 者指南中的 Amazon Relational Database Service 控制](#)。

透過使用 Security Hub 監控您 RDS 的使用狀況，因為它關係到安全最佳實務。如需詳細資訊，請參閱 [什麼是 AWS Security Hub？](#)。

使用 AWS Management Console、AWS CLI、或 RDS API 來變更主要使用者的密碼。如果您使用其他工具 (如 SQL 用戶端) 來變更主要使用者的密碼，可能會導致系統意外撤銷使用者權限。

使用安全群組控制存取

VPC 安全群組會控制進出資料庫執行個體流量的存取權限。根據預設，您資料庫執行個體的網路存取是關閉的。您可以在允許從 IP 地址範圍、連接埠或安全群組存取的安全群組中指定規則。設定傳入規則後，相同規則就會套用到與該安全群組相關聯的所有資料庫執行個體。您最多可在安全群組中指定 20 條規則。

VPC 安全群組概觀

每個 VPC 安全群組都可能可以讓特定來源存取 VPC 中與該 VPC 安全群組相關聯的資料庫執行個體。來源可以是地址的來源 (例如，203.0.113.0/24) 或另一個 VPC 安全群組。藉由指定 VPC 安全群組做為來源，您允許從使用來源 VPC 安全群組的所有執行個體 (通常指的是應用程式伺服器) 傳入的流量。VPC 安全群組可以擁有同時掌管入站和出站流量的規則。但是，傳出流量規則通常不適用於資料庫執行個體。僅在資料庫執行個體充當用戶端時，傳出流量規則才適用。例如，傳出流量規則適用於具有傳出資料庫連結的 Oracle 資料庫執行個體。您必須使用 [Amazon EC2 API](#) 或 VPC 主控台上的 Security Group (安全群組) 選項，才能建立 VPC 安全群組。

為您的 VPC 安全群組建立允許存取 VPC 中執行個體的規則時，您必須針對規則允許存取的每個地址範圍指定一個連接埠。例如，如果您想要對 VPC 中的執行個體開啟安全殼 (SSH) 存取，則可以建立一個規則，允許存取所指定之地址範圍的 TCP 連接埠 22。

您可以設定多個 VPC 安全群組，允許存取 VPC 中不同執行個體的不同連接埠。例如，您可以建立 VPC 安全群組，允許存取 VPC 中 Web 伺服器的 TCP 連接埠 80。然後，您可以建立另一個 VPC 安全群組，允許存取 VPC 中 RDS for MySQL 資料庫執行個體的 TCP 連接埠 3306。

如需 VPC 安全群組的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [安全群組](#)。

Note

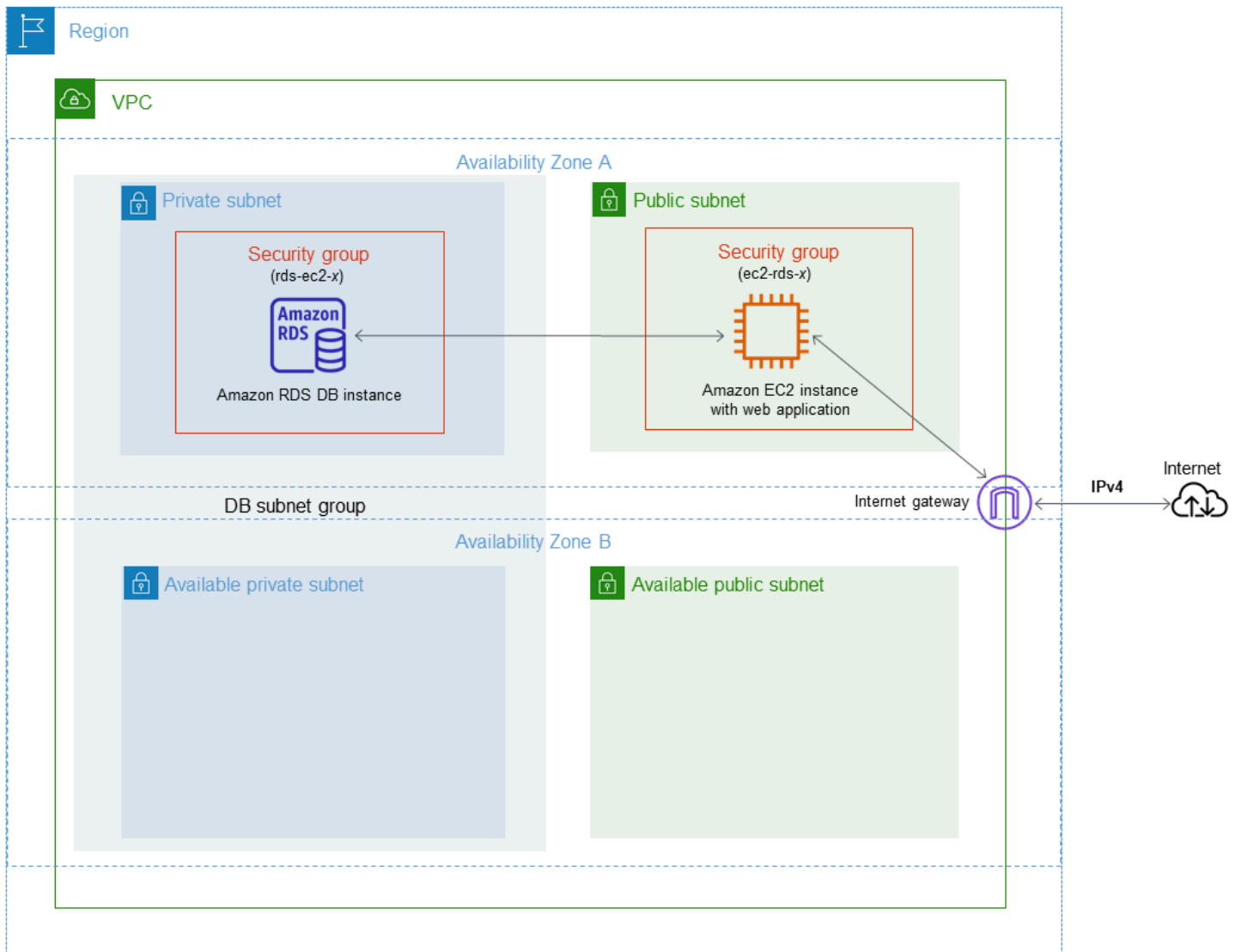
如果您的資料庫執行個體位於 VPC 中，但無法公開存取，您也可以使用 AWS Site-to-Site VPN 連線或連線，從私人網路存取該叢集。AWS Direct Connect 如需詳細資訊，請參閱 [網際網路流量隱私權](#)。

安全群組案例

VPC 中資料庫執行個體的常見使用方式，是與在相同 VPC 的 Amazon EC2 執行個體中執行的應用程式伺服器共享資料，這是由 VPC 外的應用程式用戶端存取的。針對此案例，您可以使用 AWS Management Console 上的 RDS 和 VPC 頁面，或 RDS 和 EC2 API 操作來建立必要的執行個體和安全群組：

1. 建立 VPC 安全群組 (例如，sg-0123ec2example)，並定義使用用戶端應用程式之 IP 地址做為來源的傳入規則。此安全群組可讓您的用戶端應用程式連接至 VPC 中使用此安全群組的 EC2 執行個體。
2. 建立應用程式的 EC2 執行個體，並將 EC2 執行個體新增至您在前一個步驟中建立的 VPC 安全群組 (sg-0123ec2example)。
3. 建立第二個 VPC 安全群組 (例如，sg-6789rdsexample)，並建立新規則，方法為指定您在步驟 1 中建立的 VPC 安全群組 (sg-0123ec2example) 做為來源。
4. 建立新的資料庫執行個體，並將資料庫執行個體新增至您在前一個步驟中建立的 VPC 安全群組 (sg-6789rdsexample)。當建立資料庫執行個體時，請使用您為步驟 3 中所建立 VPC 安全群組 (sg-6789rdsexample) 規則指定的相同連接埠號碼。

此案例可以下列圖表顯示。



如需此案例中設定 VPC 的詳細說明，請參閱 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。如需使用 VPC 的詳細資訊，請參閱 [Amazon VPC](#) 和 [Amazon RDS](#)。

建立 VPC 安全群組

您可以使用 VPC 主控台，建立資料庫執行個體的 VPC 安全群組。如需建立安全群組的相關資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#) 和 [安全群組](#)。

將安全群組與資料庫執行個體建立關聯

您可以使用 RDS 主控台上的修改、ModifyDBInstance Amazon RDS API 或 modify-db-instance AWS CLI 命令，將安全群組與資料庫執行個體建立關聯。

下列 CLI 範例會關聯特定 VPC 安全群組，並從資料庫執行個體移除資料庫安全群組

```
aws rds modify-db-instance --db-instance-identifier dbName --vpc-security-group-ids sg-ID
```

如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。如需您從資料庫快照還原資料庫執行個體時的安全群組考量，請參閱[安全群組考量](#)。

Note

如果連接埠值設定為非預設值，RDS 主控台會為您的資料庫顯示不同的安全群組規則名稱。

對於 RDS for Oracle 資料庫執行個體，您可以填入 Oracle 企業管理員資料庫快速 (OEM)、企業管理員雲端控制的 Oracle 管理代理程式 (OEM 代理程式) 和 Oracle 安全通訊端層選項的安全性群組選項設定來關聯其他安全性群組。在此情況下，與資料庫執行個體相關聯的安全群組和選項設定都會套用到資料庫執行個體。如需這些選項群組的詳細資訊，請參閱[Oracle Enterprise Manager](#)，請參閱[適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent](#)、和[Oracle Secure Sockets Layer](#)。

主要使用者帳戶權限

在您建立新資料庫執行個體時，您使用的預設主要使用者會取得該資料庫執行個體的特定權限。您無法在資料庫執行個體建立之後變更主要使用者名稱。

Important

我們強烈建議您不要直接在您的應用程式中使用主要使用者。而是遵循最佳實務，使用以應用程式所需的最低權限建立的資料庫使用者。

Note

如果您不慎除主要使用者的許可，則可以透過修改資料庫執行個體並設定新的主要使用者密碼來還原它們。如需修改資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

下表顯示主要使用者針對每一個資料庫引擎取得的權限和資料庫角色。

資料庫引擎	系統權限	資料庫角色
適用於 Db2 的 RDS	<p>主要使用者會指派給masterdba 群組，並將master_user_role .</p> <p>SYSMON, DBADM DATAACCESS 與ACCESSCTRL ,,BINDADD,CONNECT,CREATETAB ,CREATE_SECURE_OBJECT ,EXPLAIN,,IMPLICIT_SCHEMA ,LOAD,SQLADM, WLMADM</p>	<p>DBA, DBA_RESTRICTED , DEVELOPER , ROLE_NULL ID_PACKAGES , ROLE_PROCEDURES , ROLE_TABLESPACES</p>
RDS for MariaDB	<p>SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, REPLICATION SLAVE</p>	—
適用於 MySQL 8.0.36 及更高版本的 RDS	<p>SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN , SET_USER_ID , XA_RECOVER_ADMIN</p>	<p>rds_superuser_role</p> <p>如需 rds_superuser_role 的相關資訊，請參閱 角色型權限模型。</p>
適用於 MySQL 的 RDS 版本低於 8.0.36	<p>SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, REPLICATION SLAVE</p>	—

資料庫引擎	系統權限	資料庫角色
RDS for PostgreSQL	CREATE ROLE, CREATE DB, PASSWORD VALID UNTIL INFINITY, CREATE EXTENSION , ALTER EXTENSION , DROP EXTENSION , CREATE TABLESPACE , ALTER <OBJECT> OWNER, CHECKPOINT , PG_CANCEL_BACKEND() , PG_TERMINATE_BACKEND() , SELECT PG_STAT_REPLICATION , EXECUTE PG_STAT_STATEMENTS_RESET() , OWN POSTGRES_FDW_HANDLER() , OWN POSTGRES_FDW_VALIDATOR() , OWN POSTGRES_FDW , EXECUTE PG_BUFFERCACHE_PAGES() , SELECT PG_BUFFERCACHE	RDS_SUPERUSER 如需 RDS_SUPERUSER 的詳細資訊，請參閱 了解 PostgreSQL 角色和許可 。
RDS for Oracle	ADMINISTER DATABASE TRIGGER , ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, AUDIT SYSTEM, CHANGE NOTIFICATION , DROP ANY DIRECTORY , EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, EXEMPT REDACTION POLICY, FLASHBACK ANY TABLE, GRANT ANY OBJECT PRIVILEGE , RESTRICTED SESSION , SELECT ANY TABLE, UNLIMITED TABLESPACE	DBA <div data-bbox="1068 961 1507 1711" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px;"> <p> Note</p> <p>DBA角色可免除下列權限：</p> <p>ALTER DATABASE, ALTER SYSTEM, CREATE ANY DIRECTORY , CREATE EXTERNAL JOB, CREATE PLUGGABLE DATABASE, GRANT ANY PRIVILEGE , GRANT ANY ROLE, READ ANY FILE GROUP</p> </div>

資料庫引擎	系統權限	資料庫角色
Amazon RDS for Microsoft SQL Server	ADMINISTER BULK OPERATIONS , ALTER ANY CONNECTION , ALTER ANY CREDENTIAL , ALTER ANY EVENT SESSION, ALTER ANY LINKED SERVER, ALTER ANY LOGIN, ALTER ANY SERVER AUDIT, ALTER ANY SERVER ROLE, ALTER SERVER STATE, ALTER TRACE, CONNECT SQL, CREATE ANY DATABASE, VIEW ANY DATABASE, VIEW ANY DEFINITION , VIEW SERVER STATE, ALTER ON ROLE SQLAgentOperatorRole	DB_OWNER (資料庫層級角色)、PROCESSADMIN (伺服器層級角色)、SETUPADMIN (伺服器層級角色)、SQLAgentUserRole (資料庫層級角色)

使用 Amazon RDS 的服務連結角色

Amazon RDS 會使用 AWS Identity and Access Management (IAM) [服務連結角色](#)。服務連結角色是直接連結至 Amazon RDS 的一種特殊 IAM 角色類型。服務連結角色由 Amazon RDS 預先定義，且內含該服務代您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆使用 Amazon RDS，因為您不需要手動新增必要許可。Amazon RDS 定義其服務連結角色的許可，除非另有定義，否則僅有 Amazon RDS 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能附加到任何其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除角色。如此可保護您的 Amazon RDS 資源，避免您不小心移除資源的存取許可。

如需關於支援服務連結角色的其他服務資訊，請參閱 [《可搭配 IAM 運作的 AWS 服務》](#)，並尋找在 Service-Linked Role (服務連結角色) 欄中顯示為 Yes (是) 的服務。選擇具有連結的 Yes (是)，以檢視該服務的服務連結角色文件。

Amazon RDS 的服務連結角色許可

Amazon RDS 使用名為 AWSServiceRoleForRDS 的服務連結角色，以允許 Amazon RDS 代表資料庫執行個體來呼叫 AWS 服務。

AWSServiceRoleForRDS 服務連結角色信任下列服務可擔任該角色：

- `rds.amazonaws.com`

此服務連結角色具有名為 AmazonRDSServiceRolePolicy 的許可政策，該政策會授予此角色在帳戶中操作的許可。此角色許可政策允許 Amazon RDS 對指定資源完成下列動作：

如需有關此政策的詳細資訊，包括 JSON 政策文件，請參閱 [《AWS 受管政策參考指南》](#) 中的 [AmazonRDSServiceRolePolicy](#)。

Note

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如果您遇到下列錯誤訊息：

無法建立資源。請確認您擁有建立服務連結角色的許可。否則請等待，然後再試一次。

請確定您已啟用下列許可：

```
{
```

```
"Action": "iam:CreateServiceLinkedRole",
"Effect": "Allow",
"Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
"Condition": {
  "StringLike": {
    "iam:AWSServiceName": "rds.amazonaws.com"
  }
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[服務連結角色許可](#)。

為 Amazon RDS 建立服務連結角色

您不需要手動建立一個服務連結角色。當您建立資料庫執行個體時，Amazon RDS 會為您建立服務連結角色。

Important

Amazon RDS 自 2017 年 12 月 1 日開始支援服務連結角色，若您在這之前就有使用此服務，則 Amazon RDS 會在帳戶中建立 AWSServiceRoleForRDS 角色。若要進一步了解，請參閱[顯示在我的 AWS 帳戶中的新角色](#)。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立資料庫執行個體時，Amazon RDS 會再次為您建立服務連結角色。

為 Amazon RDS 編輯服務連結角色

Amazon RDS 不允許您編輯 AWSServiceRoleForRDS 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的[編輯服務連結角色](#)。

為 Amazon RDS 刪除服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。但是，您必須先刪除所有資料庫執行個體，才能刪除服務連結角色。

清除服務連結角色

您必須先確認服務連結角色沒有作用中的工作階段，並移除該角色使用的資源，之後才能使用 IAM 將其刪除。

檢查服務連結角色是否於 IAM 主控台有作用中的工作階段

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後選擇 AWSServiceRoleForRDS 角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 分頁中，檢閱服務連結角色的近期活動。

Note

如果您不確定 Amazon RDS 是否正在使用 AWSServiceRoleForRDS 角色，可嘗試刪除該角色。如果服務正在使用該角色，則刪除會失敗，而您可以檢視正在使用該角色的 AWS 區域。如果服務正在使用該角色，您必須先等到工作階段結束，才能刪除該角色。您無法撤銷服務連結角色的工作階段。

如果您想要移除 AWSServiceRoleForRDS 角色，首先務必要刪除所有資料庫執行個體。

刪除您的所有執行個體

使用下列其中一個程序來刪除您的每一個執行個體。

刪除執行個體 (主控台)

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇想要刪除的執行個體。
4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 若出現 Create final Snapshot? (是否建立最終快照?) 提示訊息，請選擇 Yes (是) 或 No (否)。
6. 如果您在前一個步驟中選擇 Yes (是)，則對於 Final snapshot name (最終快照名稱)，輸入您的最終快照名稱。
7. 選擇 Delete (刪除)。

刪除執行個體 (CLI)

請參閱 AWS CLI 命令參考中的 [delete-db-instance](#)。

刪除執行個體 (API)

請參閱 Amazon RDS API 參考中的 [DeleteDBInstance](#)。

使用 IAM 主控台、IAM CLI 或 IAM API 刪除 AWSServiceRoleForRDS 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 [刪除服務連結角色](#)。

Amazon RDS Custom 的服務連結角色許可

Amazon RDS Custom 使用名為 AWSServiceRoleForRDSCustom 的服務連結角色，以允許 RDS Custom 代表您的資料庫執行個體叢集來呼叫 AWS 服務。

AWSServiceRoleForRDSCustom 服務連結角色信任下列服務可擔任該角色：

- `custom.rds.amazonaws.com`

此服務連結角色具有名為 AmazonRDSCustomServiceRolePolicy 的許可政策，該政策會授予此角色在帳戶中操作的許可。此角色許可政策允許 RDS Custom 對指定資源完成下列動作：

如需有關此政策的詳細資訊，包括 JSON 政策文件，請參閱《AWS 受管政策參考指南》中的 [AmazonRDSCustomServiceRolePolicy](#)。

為 RDS Custom 建立、編輯或刪除服務連結角色的運作方式與 Amazon RDS 相同。如需詳細資訊，請參閱 [Amazon RDS 的服務連結角色許可](#)。

Note

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如果您遇到下列錯誤訊息：
無法建立資源。請確認您擁有建立服務連結角色的許可。否則請等待，然後再試一次。
請確定您已啟用下列許可：

```
{
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
  "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/AmazonRDSCustomServiceRolePolicy",
```

```
"Condition": {
  "StringLike": {
    "iam:AWSServiceName": "custom.rds.amazonaws.com"
  }
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的[服務連結角色許可](#)。

Amazon VPC 和 Amazon RDS

Amazon Virtual Private Cloud (Amazon VPC) 可讓您在 Virtual Private Cloud (VPC) 中啟動 Amazon RDS 資料庫執行個體叢集等 AWS 資源。

使用 VPC 時，您可以掌控您的虛擬聯網環境。您可以選擇自己的 IP 地址範圍、建立子網路，以及設定路由和存取控制清單。在 VPC 中執行資料庫執行個體無需額外成本。

帳戶有預設 VPC。系統會在預設 VPC 中建立所有新的資料庫執行個體，除非您另外指定。

主題

- [在 VPC 中使用資料庫執行個體](#)
- [更新資料庫執行個體的 VPC](#)
- [在 VPC 中存取資料庫執行個體的案例](#)
- [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)
- [教學課程：建立要與資料庫執行個體搭配使用的 \(VPC\)\(雙堆疊模式\)](#)
- [將不在 VPC 中的資料庫執行個體移入 VPC](#)

在以下內容中，您可以找到 Amazon RDS 資料庫執行個體的 VPC 功能相關的討論。如需 Amazon VPC 的詳細資訊，請參閱 [Amazon VPC 入門指南](#) 和 [Amazon VPC 使用者指南](#)。

在 VPC 中使用資料庫執行個體

您的資料庫執行個體位在虛擬私有雲端 (VPC) 內。VPC 是虛擬網路，在邏輯上，它與 AWS 雲端中的其他虛擬網路相互獨立。Amazon VPC 可讓您在 VPC 中啟動各種 AWS 資源，例如 Amazon RDS 資料庫執行個體或 Amazon EC2 執行個體。VPC 可以是您帳戶隨附的預設 VPC，或是您自行建立的 VPC。所有 VPC 皆會與您的 AWS 帳戶相互關聯。

您的預設 VPC 有三個子網路，可供您在 VPC 內隔離資源。預設的 VPC 也有網際網路閘道，只要利用網際網路閘道，即可允許從 VPC 以外的位置存取 VPC 內的資源。

如需涉及 VPC 內外 Amazon RDS 資料庫執行個體的藍本清單，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

主題

- [在 VPC 中使用資料庫執行個體](#)
- [使用資料庫子網路群組](#)

- [共用子網路](#)
- [Amazon RDS IP 定址](#)
- [在 VPC 中的網際網路中隱藏資料庫執行個體](#)
- [在 VPC 中建立資料庫執行個體](#)

於下列教學課程中，您可學習建立可用於常見 Amazon RDS 案例的 VPC：

- [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)
- [教學課程：建立要與資料庫執行個體搭配使用的 \(VPC\)\(雙堆疊模式\)](#)

在 VPC 中使用資料庫執行個體

在 VPC 中使用資料庫執行個體的要訣如下：

- 您的 VPC 必須至少要有兩個子網路。這些子網路必須位於您要部署資料庫執行個體的 AWS 區域，且位於兩個不同的可用區域中。子網路是指可供您指定之 VPC 的 IP 地址範圍區段，您可根據安全與運作需求將資料庫執行個體分組。

若為異地同步備份部署，為 AWS 區域 區域中至少兩個可用區域定義子網路，可允許 Amazon RDS 依需要於其他可用區域中建立新的待命。即使是單一可用區部署也務必要這麼做，以便未來需要將它們轉換為異地同步備份部署。

Note

本地區域的資料庫子網路群組只能有一個子網路。

- 如果您要開放資料庫執行個體在 VPC 中供公開存取，須啟用 VPC 屬性 DNS hostnames (DNS 主機名稱) 和 DNS resolution (DNS 解析)。
- 您的 VPC 必須具備您所建立的資料庫子網路群組。您可以透過指定您所建立的子網路來建立資料庫子網路群組。Amazon RDS 會選擇該子網路群組內的子網路和 IP 地址，以與您的資料庫執行個體建立關聯。資料庫執行個體使用包含子網路的可用區域。
- 您的 VPC 必須具備能允許存取資料庫執行個體的 VPC 安全群組。

如需詳細資訊，請參閱[在 VPC 中存取資料庫執行個體的案例](#)。

- 您每個子網路中的 CIDR 區塊大小必須足以容納備用 IP 地址，以供 Amazon RDS 在維護活動 (包括容錯移轉與運算擴展) 期間使用。例如，10.0.0.0/24 和 10.0.1.0/24 等範圍通常就夠大了。

- VPC 的執行個體租用屬性可以是 default (預設) 或 dedicated (專用)。所有預設 VPC 已將執行個體租用屬性設為預設，且預設的 VPC 可支援任何資料庫執行個體類別。

若您選擇讓資料庫執行個體位於執行個體租用屬性設為「專用」的專屬 VPC 中，資料庫執行個體的資料庫執行個體類別必須選擇獲核准的 Amazon EC2 專用預留執行個體類型。舉例來說，r5.large EC2 專用預留執行個體對應到 db.r5.large 資料庫執行個體類別。如需 VPC 中執行個體租用的相關資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的[專用執行個體](#)。

如需進一步了解專用執行個體中可能出現的執行個體類型，請參閱 EC2 定價頁面上的[Amazon EC2 專用執行個體](#)。

Note

將執行個體租用屬性設定為專用於資料庫執行個體時，並不保證資料庫執行個體會專用於執行個體上執行。

- 選項群組指派給資料庫執行個體後，就會與資料庫執行個體的 VPC 建立關聯。在這樣的連結設定下，若嘗試將資料庫執行個體還原至不同的 VPC，將無法使用指派給資料庫執行個體的選項群組。
- 若將資料庫執行個體還原至不同的 VPC，您必須將預設選項群組指派給該資料庫執行個體，指派連結至該 VPC 的選項群組，或是建立新的選項群組並將其指派給該資料庫執行個體。若使用 Oracle TDE 等持續或永久選項，將資料庫執行個體還原至不同的 VPC 時，您必須建立包含持續或永久選項的新選項群組。

使用資料庫子網路群組

Subnets (子網路) 是可供您指定之 VPC 的 IP 地址範圍區段，您可根據安全與運作需求將資源分組。資料庫子網路群組是一種子網路集合 (通常為私有)，您必須先在 VPC 中建立這些群組，然後指定群組供您的資料庫執行個體使用。透過使用資料庫子網路群組，您可在使用 AWS CLI 或 RDS API 建立資料庫執行個體時指定特定的 VPC。如果使用主控台，您可以選擇要使用的 VPC 和子網路群組。

各個資料庫子網路群組在指定的 AWS 區域中，皆應在至少兩個可用區域中具有子網路。在 VPC 中建立資料庫執行個體時，您必須為其選擇資料庫子網路群組。從資料庫子網路群組中，Amazon RDS 會選擇子網路和該子網路內的 IP 地址，以便與中的資料庫執行個體建立關聯。資料庫使用包含子網路的可用區域。

若多可用區部署的主要資料庫執行個體發生故障，Amazon RDS 可擢升相對應的待命執行個體，稍後使用任一其他可用區域子網路的 IP 地址建立新的待命執行個體。

資料庫子網路群組中的子網路可設為公開或私有，視您針對子網路的網路存取控制清單 (網路 ACL) 和路由表所設定的組態而定，子網路為公開或私有。若要使資料庫執行個體可公開存取，其資料庫子網路群組中的所有子網路皆必須為公有。若與可公開存取的資料庫執行個體關聯的子網路從公有變更為私有，則可能會影響資料庫執行個體可用性。

如要建立支援雙堆疊模式的資料庫子網路群組，請確保新增至資料庫子網路群組的每個子網路皆具有與其關聯的網際網路通訊協定第 6 版 (IPv6) CIDR 區塊。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [Amazon RDS IP 定址](#) 和 [遷移至 IPv6](#)。

Note

本地區域的資料庫子網路群組只能有一個子網路。

Amazon RDS 在 VPC 中建立資料庫執行個體時，會使用來自您資料庫子網路群組的 IP 地址，將網路介面指派給您的資料庫執行個體。然而，我們強烈建議您使用網域名稱系統 (DNS) 名稱，與您的資料庫執行個體連線。我們如此建議，是因為基本的 IP 地址在容錯移轉期間會有所變動。

Note

在資料庫子網路群組的每個子網路中，請確保為 VPC 中執行的每個資料庫執行個體保留至少一個地址，以供 Amazon RDS 執行復原動作。

共用子網路

您可以在共用 VPC 中建立資料庫執行個體。

使用共用 VPC 時要記住的一些考量：

- 您可以將資料庫執行個體從共用 VPC 子網路移至非共用 VPC 子網路，反之亦然。
- 共用 VPC 的參與者必須在 VPC 中建立安全群組，才能允許他們建立資料庫執行個體。
- 共用 VPC 中的擁有者和參與者可以使用 SQL 查詢存取資料庫。不過，只有資源的建立者才能對資源進行任何 API 呼叫。

Amazon RDS IP 定址

IP 地址可讓您 VPC 中的資源彼此互相通訊，也能和網際網路上的資源通訊。Amazon RDS 同時支援 IPv4 和 IPv6 定址通訊協定。依預設，Amazon RDS 和 Amazon VPC 使用 IPv4 定址通訊協定。您無法關閉此行為。當您建立 VPC 時，請務必指定 IPv4 CIDR 區塊 (私有 IPv4 地址的範圍)。您可選擇將 IPv6 CIDR 區塊指派給 VPC 和子網路，並將 IPv6 地址從該區塊指派給子網路中的資料庫執行個體。

對 IPv6 通訊協定的支援擴展受支援的 IP 地址的數量。使用 IPv6 通訊協定，您可以確保擁有足夠的可用地址，可應對網際網路的未來發展。新的和現有的 RDS 資源可於您的 VPC 中使用 IPv4 和 IPv6 地址。在應用程式的不同部分中使用的兩個通訊協定之間進行設定、保護和轉譯網路流量可能會造成操作額外負荷。您可在 IPv6 通訊協定上對 Amazon RDS 資源進行標準化，以簡化您的網路組態。

主題

- [IPv4 地址](#)
- [IPv6 地址](#)
- [雙堆疊模式](#)

IPv4 地址

在您建立 VPC 時，必須以 CIDR 區塊的形式為 VPC 指定 IPv4 地址的範圍，例如 10.0.0.0/16。資料庫子網路群組定義資料庫執行個體可以使用之此 CIDR 區塊中的 IP 地址範圍。這些 IP 地址可為私有或公有。

私有 IPv4 地址是無法在網際網路存取的 IP 地址。您可使用私有 IPv4 地址，在相同 VPC 中的資料庫執行個體及其他資源 (例如 Amazon EC2 執行個體) 之間進行通訊。每個資料庫執行個體具有用於在 VPC 中進行通訊的私有 IP 地址。

公有 IP 地址是可從網際網路存取的 IPv4 地址。您可使用公有地址，在資料庫執行個體和網際網路上的資源 (例如 SQL 用戶端) 之間進行通訊。您控制您的資料庫執行個體是否接收公有 IP 地址。

如需示範如何建立僅包含私有 IPv4 地址以用於常見 Amazon RDS 案例之 VPC 的教學課程，請參閱 [教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。

IPv6 地址

您可以選擇性的將 IPv6 CIDR 區塊與您的 VPC 和子網路建立關聯，並指派該區塊的 IPv6 地址給您 VPC 中的資源。每個 IPv6 地址都是全域唯一的。

您的 VPC 的 IPv6 CIDR 區塊會自動從 Amazon 的 IPv6 地址集區自動指派。您無法自行選擇範圍。

連線至 IPv6 地址時，請確保符合下列條件：

- 用戶端已設定為允許透過 IPv6 進行用戶端至資料庫的流量。
- 資料庫執行個體使用的 RDS 安全群組已正確設定，允許透過 IPv6 進行用戶端至資料庫的流量。
- 用戶端作業系統堆疊允許 IPv6 地址上的流量，且作業系統驅動程式和程式庫會進行設定以選擇正確的預設資料庫執行個體端點 (IPv4 或 IPv6)。

如需 IPv6 的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [IP 定址](#)。

雙堆疊模式

當資料庫執行個體可以同時透過 IPv4 和 IPv6 定址通訊協定進行通訊時，會在雙堆疊模式下執行。因此，資源可透過 IPv4、IPv6 或兩者與資料庫執行個體進行通訊。RDS 停用私有雙堆疊模式資料庫執行個體之 IPv6 端點的網際網路閘道存取。RDS 這麼做是為了確保您的 IPv6 端點為私有，且僅可從您的 VPC 內存取。

主題

- [雙堆疊模式和資料庫子網路群組](#)
- [使用雙堆疊模式資料庫執行個體](#)
- [將僅限 IPv4 的資料庫執行個體修改為使用雙堆疊模式](#)
- [區域和版本可用性](#)
- [雙堆疊網路資料庫執行個體的限制](#)

如需示範如何建立包含 IPv4 和 IPv6 兩個地址以用於常見 Amazon RDS 案例之 VPC 的教學課程，請參閱 [教學課程：建立要與資料庫執行個體搭配使用的 \(VPC\)\(雙堆疊模式\)](#)。

雙堆疊模式和資料庫子網路群組

如要使用雙堆疊式，請確保與資料庫執行個體關聯之資料庫子網路群組中的每個子網路皆有一個與之關聯的 IPv6 CIDR 區塊。您可建立新的資料庫子網路群組或修改現有資料庫子網路群組，以滿足此要求。於資料庫執行個體處於雙堆疊模式後，用戶端可進行正常連線。確保客戶端安全防火牆和 RDS 資料庫執行個體安全群組已準確設定，允許透過 IPv6 的流量。如要進行連線，用戶端會使用資料庫執行個體的端點。用戶端應用程式可指定連線至資料庫時所偏好的通訊協定。於雙堆疊模式下，資料庫執行個體會偵測用戶端的偏好網路通訊協定 (IPv4 或 IPv6)，並將該通訊協定用於連線。

若資料庫子網路群組因子網路刪除或 CIDR 斷開關聯而停止支援雙堆疊模式，則與資料庫子網路群組相關聯的資料庫執行個體存在網路狀態不相容的風險。此外，建立新的雙堆疊模式資料庫執行個體時，無法使用資料庫子網路群組。

如要使用 AWS Management Console 決定資料庫子網路群組是否支援雙堆疊模式，請在資料庫子網路群組的詳細資料頁面上檢視 Network type (網路類型)。若要使用判斷資料庫子網路群組是否支援雙堆疊模式 AWS CLI，請執行 [describe-db-subnet-groups](#) 命令並在輸出 SupportedNetworkTypes 中檢視。

將僅供讀取複本視為獨立的資料庫執行個體，且可具有不同於主資料庫執行個體的網路類型。若您變更僅供讀取複本之主資料庫執行個體的網路類型，則僅供讀取複本不會受到影響。還原資料庫執行個體時，您可將其還原為支援的任何網路類型。

使用雙堆疊模式資料庫執行個體

當您建立或修改資料庫執行個體時，您可指定雙堆疊模式，以允許您的資源透過 IPv4、IPv6 或兩者與資料庫執行個體進行通訊。

當您使用 AWS Management Console 來建立或修改資料庫執行個體時，您可於 Network type (網路類型) 區段中指定雙堆疊模式。下圖顯示主控台內的 Network type (網路類型) 區段。

Network type [Info](#)
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

當您使用 AWS CLI 建立或修改資料庫執行個體時，請將 `--network-type` 選項設定為 DUAL，以使用雙堆疊模式。當您使用 RDS API 建立或修改資料庫執行個體時，請將 NetworkType 參數設定為 DUAL，以使用雙堆疊模式。當您修改資料庫執行個體的網路類型時，可能會出現停機時間。若指定的資料庫引擎版本或資料庫子網路群組不支援雙堆疊模式，則會傳回 NetworkTypeNotSupported 錯誤。

如需建立資料庫執行個體的詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

若要使用主控台來決定資料庫執行個體是否處於雙堆疊模式，請檢視資料庫執行個體 Connectivity & security (連線能力與安全性) 索引標籤上的 Network type (網路類型)。

將僅限 IPv4 的資料庫執行個體修改為使用雙堆疊模式

您可將僅限 IPv4 的資料庫執行個體修改為使用雙堆疊模式。如此，請變更資料庫執行個體的網路類型。修改可能會造成停機。

建議您在維護時段變更 Amazon RDS 資料庫執行個體的網路類型。目前不支援將新執行個體的網路類型設定為雙堆疊模式。您可以使用 `modify-db-instance` 命令手動設定網路類型。

將資料庫執行個體修改為使用雙堆疊模式之前，請確保其資料庫子網路群組支援雙堆疊模式。若與資料庫執行個體關聯的資料庫子網路群組不支援雙堆疊模式，請於修改資料庫執行個體時指定支援該子網路群組的其他資料庫子網路群組。修改資料庫執行個體的資料庫子網路群組可能會導致停機。

若在將資料庫執行個體變更為使用雙堆疊模式之前修改資料庫執行個體的資料庫子網路群組，請確保該資料庫子網路群組在變更前後對資料庫執行個體有效。

對於適用於 PostgreSQL 的 RDS、適用於 MySQL 的 RDS、Oracle 版 RDS 以及適用於 MariaDB 單一可用區執行個體的 RDS，我們建議您只在將 `--network-type` 參數設為的情況下執行 [modify-db-instance](#) 命令，以將網路變更 DUAL 為雙堆疊模式。在相同的 API 呼叫中新增其他參數和 `--network-type` 參數可能會導致停機時間。若要修改多個參數，請確定已成功完成網路類型修改，然後再傳送另一個具有其他參數的 `modify-db-instance` 請求。

如果您只使用 `--network-type` 參數或在命令中合併參數，適用於 RDS for PostgreSQL、RDS for Oracle 以及適用於 MariaDB 異地同步備份資料庫執行個體的網路類型修改會造成短暫停機時間並觸發容錯移轉。 `modify-db-instance`

如果您只使用 `--network-type` 參數或在 `modify-db-instance` 命令中合併參數，則 RDS for SQL Server 單一可用區或多可用區資料庫執行個體的網路類型修改會導致停機。修改網路類型會造成 SQL Server 多可用區執行個體的容錯移轉。

若於變更後無法連線至資料庫執行個體，請確認用戶端和資料庫安全防火牆和路由表已正確設定，以允許流量傳輸至選定網路上的資料庫 (IPv4 或 IPv6)。您可能還需要修改作業系統參數、程式庫或驅動程式才可使用 IPv6 地址進行連線。

當您將資料庫執行個體修改為使用雙堆疊模式時，不可有從單一可用區部署至多可用區部署，或從多可用區部署至單一可用區部署的待處理變更。

如要將僅限 IPv4 的資料庫執行個體修改為使用雙堆疊模式

1. 修改資料庫子網路群組以支援雙堆疊模式，或建立支援雙堆疊模式的資料庫子網路群組：
 - a. 建立 IPv6 CIDR 區塊與 VPC 的關聯。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[將 IPv6 CIDR 區塊新增至 VPC](#)。

- b. 將 IPv6 CIDR 區塊連接至資料庫子網路群組中的所有子網路。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[將 IPv6 CIDR 區塊新增至子網路](#)。

- c. 確認資料庫子網路群組支援雙堆疊模式。

若您使用 AWS Management Console，請選取資料庫子網路群組，並確保 Supported network types (支援的網路類型) 值為 Dual, IPv4 (雙, IPv4)。

如果您使用的是 AWS CLI，請執行 [describe-db-subnet-groups](#) 命令，並確定資料庫執行個體的 SupportedNetworkType 值為 Dual, IPv4。

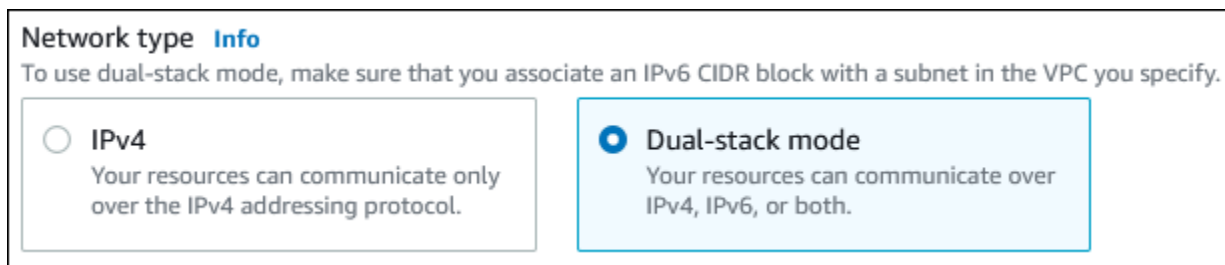
2. 修改與資料庫執行個體關聯的安全群組，以允許 IPv6 連線至資料庫，或建立允許 IPv6 連線的新安全群組。

如需說明，請參閱《Amazon VPC 使用者指南》中的[安全群組規則](#)。

3. 修改資料庫執行個體以支援雙堆疊模式。若要這麼做，請將 Network type (網路類型) 設為 Dual-stack mode (雙堆疊模式)。

若您使用主控台，請確保下列設定正確：

- Network type (網路類型) – Dual-stack mode (雙堆疊模式)



Network type [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

- DB subnet group (資料庫子網路群組) – 您在前一步驟中設定的資料庫子網路群組
- Security group (安全群組) – 您在上一個步驟中設定的安全性

若您使用 AWS CLI，請確保下列設定正確：

- `--network-type` – dual
- `--db-subnet-group-name` – 您在前一步驟中設定的資料庫子網路群組
- `--vpc-security-group-ids` – 您在前一步驟中設定的 VPC 安全群組

例如：

```
aws rds modify-db-instance --db-instance-identifier my-instance --network-type "DUAL"
```

4. 確認資料庫執行個體是否支援雙堆疊模式。

若您使用主控台，請選擇資料庫執行個體的 Connectivity & security (連線能力與安全性) (組態) 索引標籤。在該索引標籤，確保 Network type (網路類型) 值為 Dual-stack mode (雙堆疊模式)。

如果您使用的是AWS CLI，請執行[describe-db-instances](#)命令，並確定資料庫執行個體的NetworkType值為dual。

執行資料庫執行個體端點上的 dig 命令來辨識與其關聯的 IPv6 地址。

```
dig db-instance-endpoint AAAA
```

使用資料庫執行個體端點 (非 IPv6 地址)，連線至資料庫執行個體。

區域和版本可用性

功能可用性和支援會因每個資料庫引擎的特定版本以及 AWS 區域 而有所不同。如需雙堆疊模式的版本和區域可用性的相關資訊，請參閱 [Amazon RDS 中支援雙堆疊模式的區域和資料庫引擎](#)。

雙堆疊網路資料庫執行個體的限制

下列限制適用於雙堆疊網路資料庫執行個體：

- 資料庫執行個體無法僅使用 IPv6 通訊協定。其可專門使用 IPv4，也可使用 IPv4 和 IPv6 通訊協定 (雙堆疊模式)。
- Amazon RDS 不支援原生 IPv6 子網路。
- 使用雙堆疊模式的資料庫執行個體必須為私有。其無法公開存取。
- 雙堆疊模式不支援 db.m3 和 db.r3 資料庫執行個體類別。
- 若為 RDS for SQL Server，使用 Always On AG 可用性群組接聽程式端點的雙堆疊模式資料庫執行個體僅提供 IPv4 地址。
- 您無法將 RDS Proxy 和雙堆疊模式資料庫執行個體搭配使用。
- 您無法在 AWS Outposts 資料庫執行個體上使用雙堆疊模式和 RDS。

- 您無法於本機區域中使用雙堆疊模式和資料庫執行個體。

在 VPC 中的網際網路中隱藏資料庫執行個體

VPC 中，有個 EC2 執行個體上有開放存取的 web 應用程式，同時資料庫執行個體上有不可公開存取的資料庫，這樣的 Amazon RDS 藍本並不罕見。例如，您可以建立具有公有子網路和私有子網路的 VPC。作為 Web 伺服器的 Amazon EC2 執行個體可部署於公有子網路中，而資料庫執行個體則可部署於私有子網路中。在這樣的部署下，只有 Web 伺服器可以存取資料庫執行個體。如需這種案例的圖示，請參閱 [由相同 VPC 中的 EC2 執行個體在 VPC 中存取的資料庫執行個體](#)。

當您在 VPC 內啟動資料庫執行個體時，資料庫執行個體在 VPC 內具有流量的私人的 IP 地址。此私人 IP 地址無法公開存取。您可以使用 Public access (公用存取) 選項來指定除了私有 IP 地址之外，資料庫執行個體是否還具有公有 IP 地址。如果指定資料庫執行個體為可公開存取，則其 DNS 端點會從 VPC 內解析為私人 IP 地址。它會從 VPC 外部解析為公用 IP 地址。資料庫執行個體的存取權限最終是由其使用的安全群組所控制。若指派給該資料庫執行個體的安全群組沒有包含允許的傳入規則，則該資料庫執行個體就無法開放供公開存取。若要開放公開存取資料庫執行個體，其資料庫子網路群組中的子網路必須具備網際網路閘道。如需詳細資訊，請參閱 [無法連線至 Amazon RDS 資料庫執行個體](#)

您可以修改 Public access (公用存取) 選項，藉此修改資料庫執行個體以開啟或關閉公開存取性。下圖顯示 Additional connectivity configuration (其他連線能力組態) 區段中的 Public access (公用存取) 選項。若要設定此選項，請開啟 Connectivity (連線能力) 區段中的 Additional connectivity configuration (其他連線能力組態) 區段。

Connectivity G

Virtual private cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-2aed394c) ▼

Only VPCs with a corresponding DB subnet group are listed.

i After a database is created, you can't change its VPC.

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB cluster can use in the VPC you selected.

default ▼

Public access [Info](#)

Yes
Amazon EC2 instances and devices outside the VPC can connect to your DB cluster. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the DB cluster.

No
Amazon RDS will not assign a public IP address to the DB cluster. Only Amazon EC2 instances and devices inside the VPC can connect to your DB cluster.

VPC security group
Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

default X

► **Additional configuration**

如需有關修改資料庫執行個體以設定 Public access (公用存取) 選項的資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

在 VPC 中建立資料庫執行個體

下列步驟可協助您在 VPC 中建立資料庫執行個體。若要使用預設 VPC，您可以從步驟 2 開始，使用系統已為您建立的 VPC 和資料庫子網路群組。您想要建立額外的 VPC 的話，可以建立新的 VPC。

Note

如果您要開放資料庫執行個體在 VPC 中供公開存取，須啟用 VPC 屬性 DNS hostnames (DNS 主機名稱) 和 DNS resolution (DNS 解析)，更新 VPC 的 DNS 資訊。若要進一步了解如何更新 VPC 執行個體的 DNS 資訊，請參閱[更新 VPC 的 DNS 支援](#)。

請依照以下步驟在 VPC 中建立資料庫執行個體：

- [步驟 1：建立 VPC](#)
- [步驟 2：建立資料庫子網路群組](#)
- [步驟 3：建立 VPC 安全群組](#)
- [步驟 4：在 VPC 中建立資料庫執行個體](#)

步驟 1：建立 VPC

建立在至少兩個可用區域內有子網路的 VPC。建立資料庫子網路群組時，您會需要使用這些子網路。如有預設的 VPC，則系統會自動為您於所在 AWS 區域的每個可用區域中建立子網路。

如需詳細資訊，請參閱[建立含私有和公有子網路的 VPC](#) 或 Amazon VPC 使用者指南中的[建立 VPC](#)。

步驟 2：建立資料庫子網路群組

資料庫子網路群組是一種子網路集合 (通常為私有)，您必須先為 VPC 建立這些群組，然後指定給資料庫執行個體使用。若使用 AWS CLI 或 RDS API 建立資料庫執行個體，資料庫子網路群組可允許您指定特定的 VPC。若使用主控台，只需選擇要使用的 VPC 和子網路即可。各個資料庫子網路群組在 AWS 區域中，須至少於兩個可用區域中具有至少一個子網路。作為最佳實務，每個資料庫子網路群組應於 AWS 區域中，為每個可用區域擁有至少有一個子網路。

若為異地同步備份部署，為 AWS 區域中的所有可用區域定義子網路則會啟用 Amazon RDS，進而在必要時於其他可用區域中建立新的待命複本。即使是單一可用區域部署，您也可以遵循此最佳實務，因為未來可能會將它們轉換為異地同步備份部署。

若要開放資料庫執行個體供公開存取，資料庫子網路群組中的子網路必須具備網際網路閘道。如需子網路網際網路閘道的詳細資訊，請參閱 Amazon VPC 使用者指南中的[使用網際網路閘道連線至網際網路](#)。

Note

本地區域的資料庫子網路群組只能有一個子網路。

在 VPC 中建立資料庫執行個體時，您可以選擇資料庫子網路群組。Amazon RDS 會選擇子網路和該子網路內的 IP 地址，以便與您的資料庫執行個體建立關聯。如果資料庫子網路群組不存在，建立預設資料庫執行個體時，Amazon RDS 會建立預設子網路群組。Amazon RDS 會建立彈性網路界面，並透過該 IP 地址將該界面與資料庫執行個體相關聯。資料庫執行個體使用包含子網路的可用區域。

若為異地同步備份部署，為 AWS 區域 區域中至少兩個可用區域定義子網路可允許 Amazon RDS 在必要情況下於其他可用區域中建立新的待命。即使是單一可用區部署也需要這麼做，以便未來需要將它們轉換為多可用區部署。

在此步驟中，您會建立資料庫子網路群組，並新增您為 VPC 建立的子網路。

建立資料庫子網路群組

1. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。
3. 選擇 Create DB Subnet Group (建立資料庫子網路群組)。
4. 在 Name (名稱) 欄位輸入您資料庫子網路群組的名稱。
5. 在 Description (描述) 欄位輸入您資料庫子網路群組的描述。
6. 對於 VPC，選擇預設 VPC 或先前建立的 VPC。
7. 在 Add subnets (新增子網路) 區段中，選擇包含 Availability Zones (可用區域) 中子網路的可用區域，然後從 Subnets (子網路) 選擇子網路。

RDS > Subnet groups > Create DB subnet group

Create DB Subnet Group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-079bd4b8953aee1dd	10.0.0.0/24
us-east-1c	subnet-057e85b72c46fdd9a	10.0.1.0/24

Note

如果您已啟用「本地區域」，您可以在 Create DB subnet group (建立資料庫子網路群組) 頁面中選擇「可用區域」群組。在此情況下，請選擇 Availability Zone group (可用區域群組)、Availability Zones (可用區域) 和 Subnets (子網路)。

8. 選擇建立。

您的新資料庫子網路群組會顯示在 RDS 主控台的資料庫子網路群組清單中。您可以選擇資料庫子網路群組，在視窗底部的詳細資訊窗格查看詳細資訊，包括與該群組相關聯的所有子網路。

步驟 3：建立 VPC 安全群組

建立資料庫執行個體之前，您必須先建立 VPC 安全群組，才能與您的資料庫執行個體建立關聯。如果您沒有建立 VPC 安全群組，您可以在建立資料庫執行個體時使用預設安全群組。如需如何建立資料庫執行個體安全群組的說明，請參閱 [建立私有資料庫執行個體的 VPC 安全群組](#)，或參閱 Amazon VPC 使用者指南中的 [使用安全群組控制到資源的流量](#)。

步驟 4：在 VPC 中建立資料庫執行個體

在此步驟中，您將建立資料庫執行個體，並使用您在先前步驟中建立的 VPC 名稱、資料庫子網路群組以及 VPC 安全群組。

Note

如果您要開放資料庫執行個體在 VPC 中供公開存取，須啟用 VPC 屬性 DNS hostnames (DNS 主機名稱) 和 DNS resolution (DNS 解析)。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 的 DNS 屬性](#)。

如需如何建立資料庫執行個體的詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#)。

Connectivity (連線) 區段中顯示提示訊息時，請輸入 VPC 名稱、資料庫子網路群組以及 VPC 安全群組。

更新資料庫執行個體的 VPC

您可以使用 AWS Management Console 將您的資料庫執行個體移動至不同的 VPC。

如需修改資料庫執行個體的相關資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。如下所示，在修改頁面的 Connectivity (連線) 區段中，為 DB Subnet group (資料庫子網路群組) 輸入新的資料庫子網路群組。新的子網路群組必須是新 VPC 中的子網路群組。



Connectivity

Subnet group

default-vpc-665e7a1f ▼

Security group

List of DB security groups to associate with this DB instance.

如果滿足以下條件，便無法變更資料庫執行個體的 VPC：

- 資料庫執行個體位於多個可用區域。您可以將資料庫執行個體轉換至單一可用區域、移至新的 VPC，然後再轉換回多可用區域資料庫執行個體。如需詳細資訊，請參閱 [設定及管理多可用區部署](#)。
- 資料庫執行個體具有一或多個僅供讀取複本。您可以移除僅供讀取複本、將資料庫執行個體移到新的 VPC，然後再次新增僅供讀取複本。如需詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。
- 資料庫執行個體為僅供讀取複本。您可以提升僅供讀取複本，然後將獨立資料庫執行個體移至新的 VPC。如需詳細資訊，請參閱 [提升僅供讀取複本為獨立的資料庫執行個體](#)。
- 目標 VPC 中的子網路群組在資料庫執行個體的可用區域中沒有子網路。您可以將資料庫執行個體可用區域中的子網路新增到資料庫子網路群組，然後將資料庫執行個體移至新的 VPC。如需詳細資訊，請參閱 [使用資料庫子網路群組](#)。

在 VPC 中存取資料庫執行個體的案例

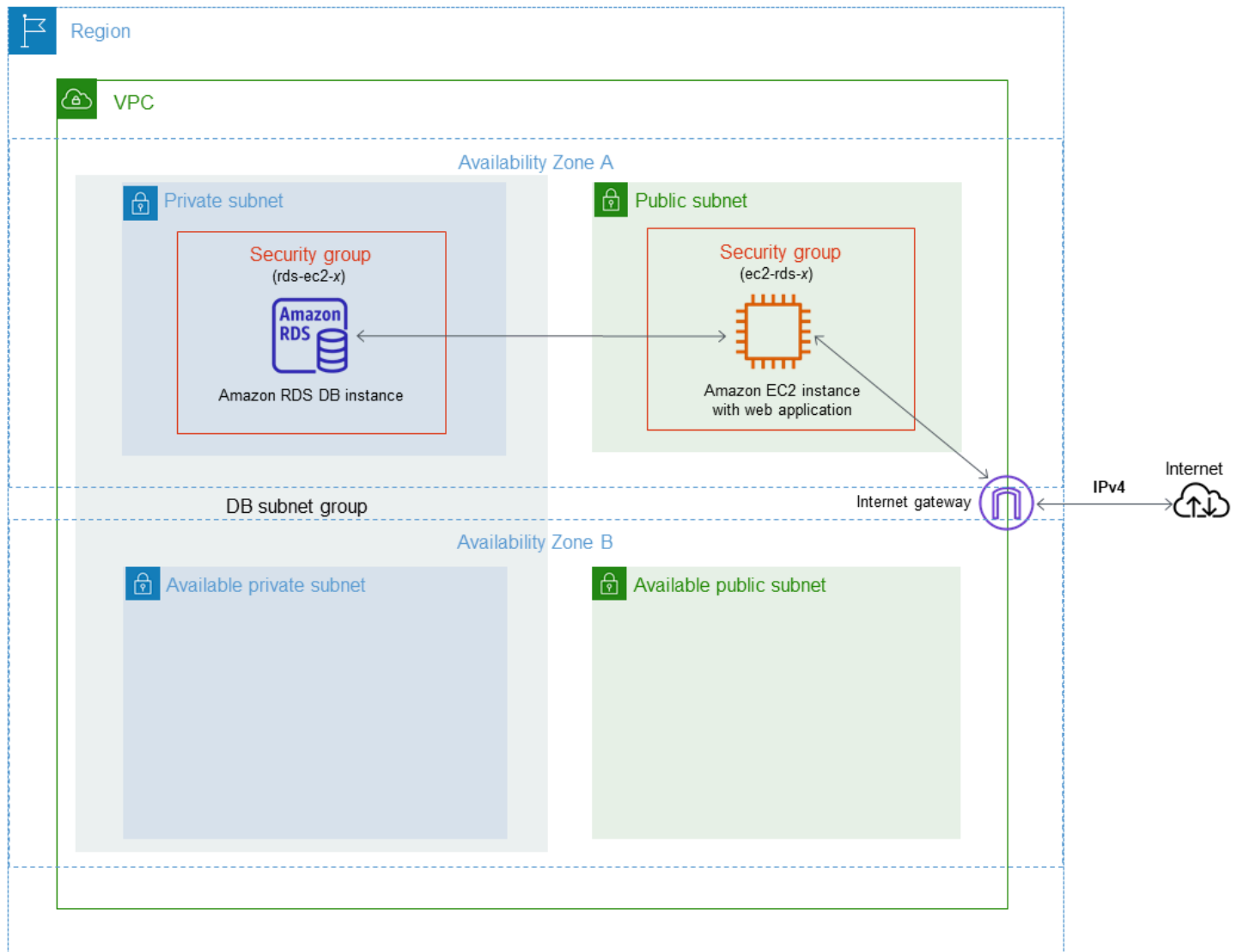
Amazon RDS 支援下列在 VPC 中存取資料庫執行個體的使用案例：

- [相同 VPC 中的 EC2 執行個體](#)
- [EC2 執行個體位於不同 VPC](#)
- [連上網際網路的用戶端應用程式](#)
- [私有網路](#)

由相同 VPC 中的 EC2 執行個體在 VPC 中存取的資料庫執行個體

資料庫執行個體在 VPC 中常見的使用方式，是與在同一 VPC 之 EC2 執行個體中執行的應用程式伺服器共用資料。

此案例可以下列圖表顯示。



若要管理相同 VPC 中 EC2 執行個體與資料庫執行個體之間的存取權限，最簡單的方式如下：

- 建立將包含資料庫執行個體的 VPC 安全群組。此安全群組可用來限制資料庫執行個體的存取權限。例如，您可以為此安全群組建立自訂規則，允許使用您在建立自訂規則時指派給資料庫執行個體的連接埠存取 TCP，並可建立一組存取資料庫執行個體的 IP 地址，做為開發或其他用途使用。

- 建立將包含 EC2 執行個體 (web 伺服器 and 用戶端) 的 VPC 安全群組。若有需要，此安全群組可允許藉由使用 VPC 路由表存取網際網路上的 EC2 執行個體。舉例來說，您可以在此安全群組上設定規則，允許 TCP 透過連接埠 22 存取 EC2 執行個體。
- 在您資料庫執行個體的安全群組中建立自訂規則，允許來自您為 EC2 執行個體所建立之安全群組的連線要求。這些規則可能會允許安全群組的所有成員存取資料庫執行個體。

在單獨的可用區域中，還有一個額外的公有和私有子網路。RDS 資料庫子網路群組需要至少兩個可用區域中的子網路。額外的子網路可讓您在未來輕鬆切換到多可用區域資料庫執行個體部署。

如需相關教學課程，了解如何為此案例建立包含公有和私有子網路的 VPC，請參閱[教學課程：建立要與資料庫執行個體搭配使用的 VPC \(僅限 IPv4\)](#)。

Tip

您可以在建立資料庫執行個體時，自動設定 Amazon EC2 執行個體與資料庫執行個體之間的連線。如需詳細資訊，請參閱[設定與 EC2 執行個體的自動網路連線](#)。

若要在允許其他安全群組連線要求的 VPC 安全群組中建立規則，請按照以下步驟操作：

1. 登入 AWS Management Console 並開啟 Amazon VPC 主控台，網址為 <https://console.aws.amazon.com/vpc>。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選擇或建立要允許其他安全群組成員存取的安全群組。在前一個藍本中，這是您用於資料庫執行個體的安全群組。選擇 Inbound rules (傳入規則) 索引標籤，然後選擇 Edit inbound rules (編輯對內規則)。
4. 在 Edit inbound rules (編輯對內規則) 頁面上，選擇 Add rule (新增規則)。
5. 對於類型，選擇對應您在建立資料庫執行個體時所使用之連接埠的項目，例如 MySQL/Aurora。
6. 在來源方塊中，開始輸入安全群組 ID，這會列出相符的安全群組。選擇安全群組，允許其成員存取由此安全群組所保護的資源。在前一個藍本中，這是您用於 EC2 執行個體的安全群組。
7. 視需要使用所有 TCP 做為類型並在來源方塊中輸入安全群組來建立規則，以重複 TCP 通訊協定的步驟。若您打算使用 UDP 通訊協定，請使用 All UDP (所有 UDP) 作為 Type (類型)，並在 Source (來源) 中輸入安全群組，以建立規則。
8. 選擇儲存規則。

下列畫面顯示安全群組針對其來源的對內規則。

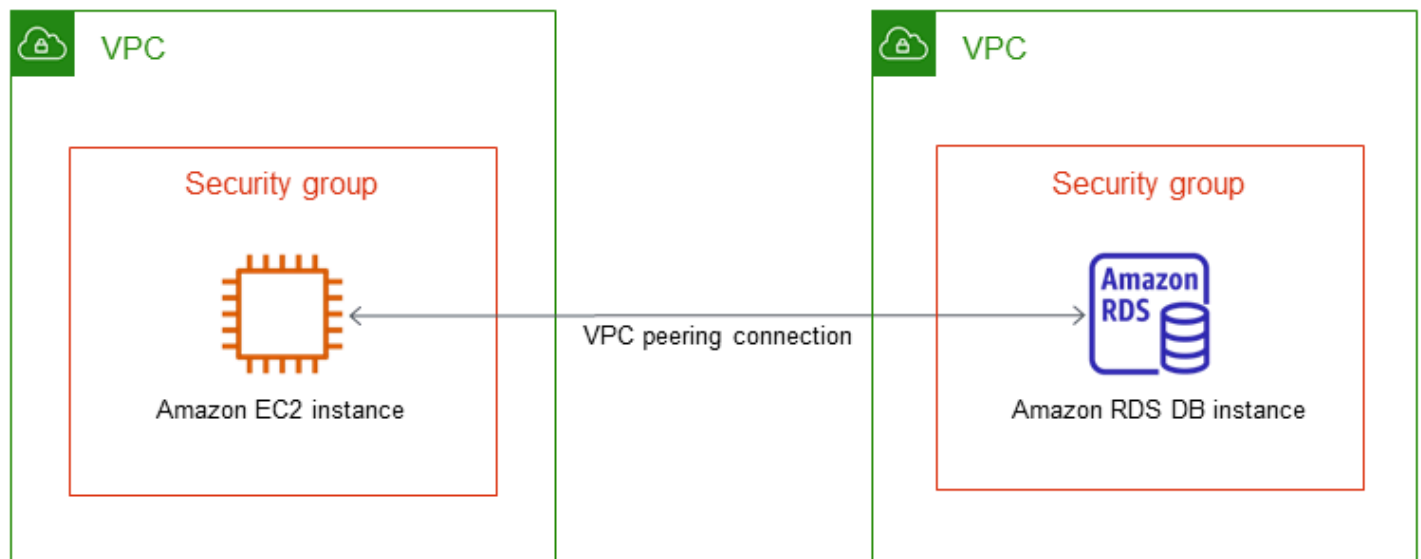
Details	Inbound rules	Outbound rules	Tags
Inbound rules Edit inbound rules			
Type	Protocol	Port range	Source
MYSQL/Aurora	TCP	3306	sg-00bd2328e37926844 (tutorial-securitygroup)

如需從 EC2 執行個體連線至資料庫執行個體的詳細資訊，請參閱[連接至 Amazon RDS 資料庫執行個體](#)。

由不同 VPC 中的 EC2 執行個體存取 VPC 中的資料庫執行個體

當您的資料庫執行個體與您用來存取資料庫執行個體叢集的 EC2 執行個體不在相同 VPC 中時，您就能使用 VPC 對等連線存取該資料庫執行個體。

此案例可以下列圖表顯示。

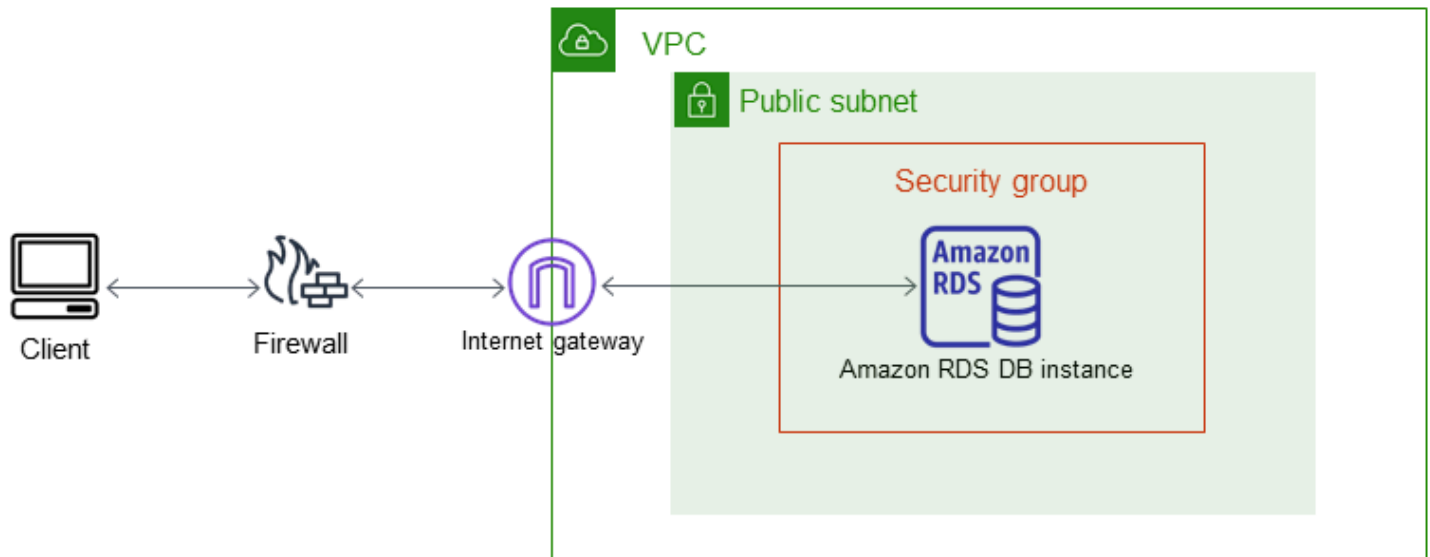


VPC 對等連線是指兩個 VPC 之間的網路連線，透過此機制，您就可以使用私有 IP 地址在兩者之間路由流量。這兩個 VPC 中的資源能彼此通訊，有如位於相同網路中一樣。您可以在自己的 VPC、其他 AWS 帳戶中使用 VPC 或不同的 VPC 建立 VPC 對等連線。AWS 區域若要進一步了解 VPC 互連，請參閱《Amazon Virtual Private Cloud 使用者指南》中的[VPC 互連](#)。

由用戶端應用程式透過網際網路存取 VPC 中的資料庫執行個體

若要由用戶端應用程式透過網際網路存取 VPC 中的資料庫執行個體，您必須設定單一公有子網路的 VPC 以及網際網路閘道，啟用網際網路通訊。

此案例可以下列圖表顯示。



我們建議您使用下列組態：

- 大小為 /16 的 VPC (例如，CIDR: 10.0.0.0/16)。此大小可提供 65,536 個私有 IP 地址。
- 大小為 /24 的子網路 (例如，CIDR: 10.0.0.0/24)。此大小可提供 256 個私有 IP 地址。
- 與 VPC 和子網路相關聯的 Amazon RDS 資料庫執行個體。Amazon RDS 會將子網路中的 IP 地址指派給資料庫執行個體。
- 網際網路閘道會將 VPC 連線至網際網路和其他 AWS 產品。
- 與資料庫執行個體相關聯的安全群組。安全群組的傳入規則允許您的用戶端應用程式存取您的資料庫執行個體。

如需有關在 VPC 中建立資料庫執行個體的資訊，請參閱 [在 VPC 中建立資料庫執行個體](#)。

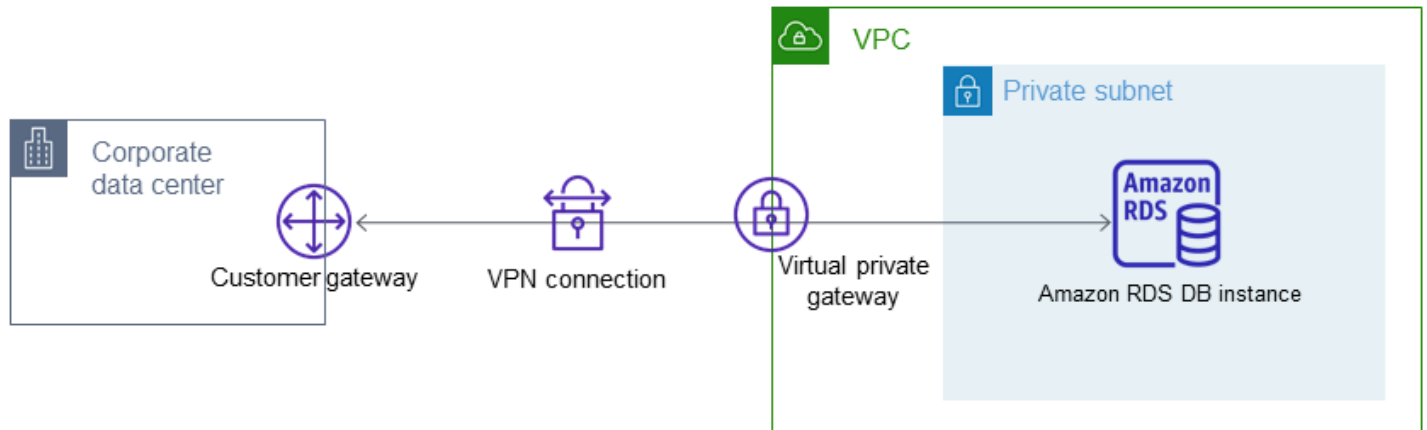
透過私有網路存取 VPC 中的資料庫執行個體

如果您的資料庫執行個體無法公開存取，可以使用下列選項從私有網路存取：

- AWS Site-to-Site VPN 連線。如需詳細資訊，請參閱 [什麼是 AWS Site-to-Site VPN ?](#)
- 一個 AWS Direct Connect 連接。如需詳細資訊，請參閱 [什麼是 AWS Direct Connect ?](#)

- 一個 AWS Client VPN 連接。如需詳細資訊，請參閱[什麼是 AWS Client VPN？](#)

下圖顯示具有站 AWS Site-to-Site VPN 連線的案例。

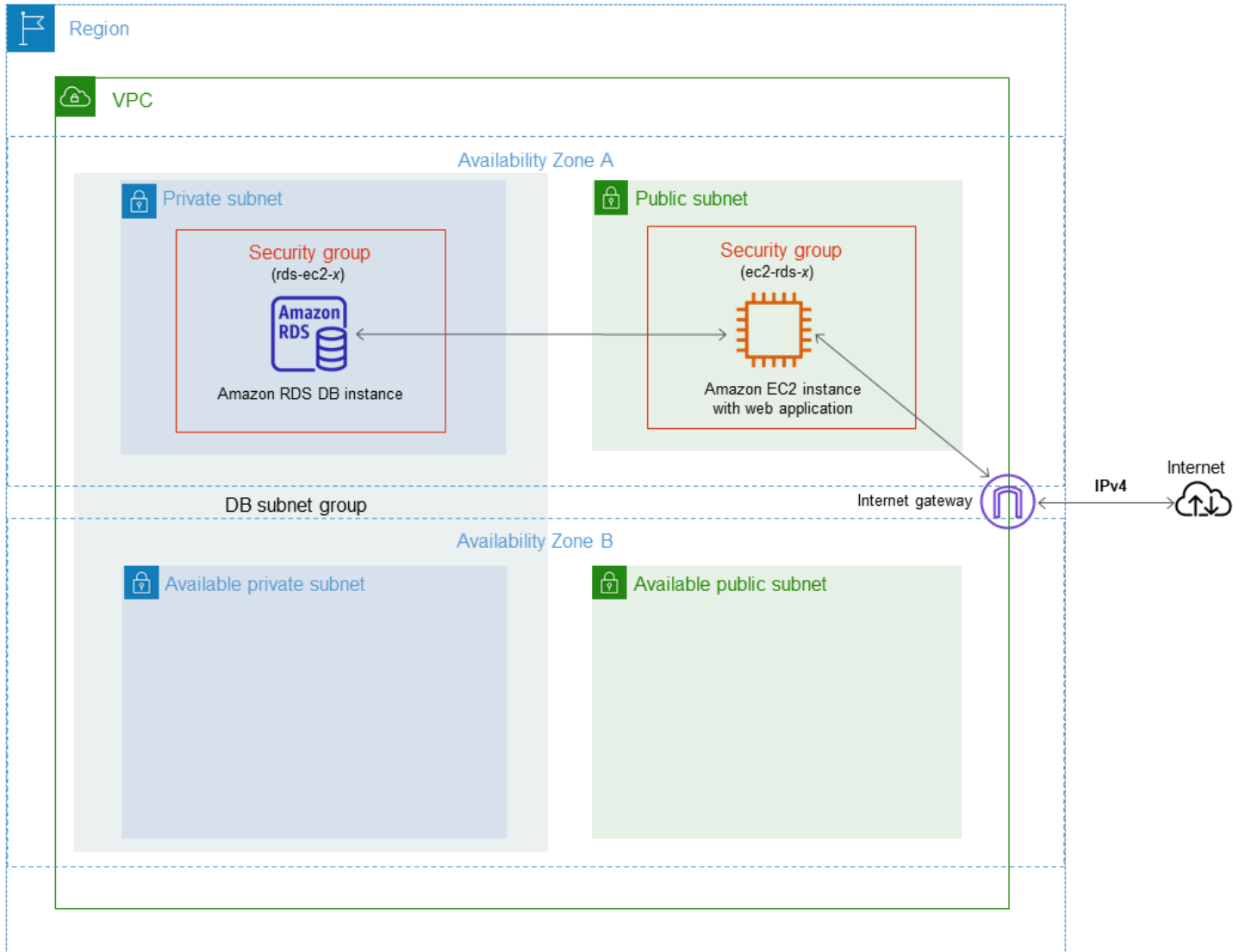


如需更多詳細資訊，請參閱[網際網路流量隱私權](#)。

教學課程：建立要與資料庫執行個體搭配使用的 VPC (僅限 IPv4)

常用案例包括以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中的資料庫執行個體。此 VPC 與在相同 VPC 中執行的 Web 伺服器共用資料。在本教學課程中，您會針對此案例建立 VPC。

此案例可以下列圖表顯示。如需其他案例的相關資訊，請參閱[在 VPC 中存取資料庫執行個體的案例](#)。



您的資料庫執行個體僅需供您的 Web 伺服器使用，無需供公有網際網路使用。因此，您建立同時包含公有和私有子網路的 VPC。Web 伺服器是在公有子網路中託管，以便它可以到達公有網際網路。資料庫執行個體被託管於私有子網路中。Web 伺服器可連線至資料庫執行個體，因為其託管於相同 VPC 中。但是，公有網際網路無法使用資料庫執行個體，提供更高的安全性。

本教學課程會在單獨的可用區域中設定額外的公有子網路和私有子網路。教學課程不會使用這些子網路。RDS 資料庫子網路群組需要至少兩個可用區域中的子網路。額外的子網路可讓您在未來更輕鬆地切換到多可用區域資料庫執行個體部署。

本教學課程說明為 Amazon RDS 資料庫執行個體配置 VPC。如需展示如何為此 VPC 案例建立 Web 伺服器的教學課程，請參閱[教學：建立 Web 伺服器和 Amazon RDS 資料庫執行個體](#)。如需 Amazon VPC 的詳細資訊，請參閱[Amazon VPC 入門指南](#)和[Amazon VPC 使用者指南](#)。

Tip

當您建立資料庫執行個體時，您可以在 Amazon EC2 執行個體和資料庫執行個體之間自動設定網路連線。網路組態與本教學課程中描述的組態類似。如需詳細資訊，請參閱[設定與 EC2 執行個體的自動網路連線](#)。

建立含私有和公有子網路的 VPC

使用下列程序來建立一個同時含公有和私有子網路的 VPC。

建立 VPC 和子網路

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在 AWS Management Console 的右上角，選擇要在其中建立 VPC 的區域。此範例使用 美國西部 (奧勒岡) 區域。
3. 在左上角，選擇 VPC Dashboard (VPC 儀表板)。若要開始建立 VPC，請選擇 Create VPC (建立 VPC)。
4. 在 VPC Settings (VPC 設定) 的 Resources to create (建立資源) 下，選擇 VPC and more (VPC 和更多)。
5. 對於 VPC settings (VPC 設定)，設定這些值：
 - Name tag auto-generation (自動產生名稱標籤) – **tutorial**
 - IPv4 CIDR block (IPv4 CIDR 區塊) – **10.0.0.0/16**
 - IPv6 CIDR block (IPv6 CIDR 區塊) – No IPv6 CIDR Block (無 IPv6 CIDR 區塊)
 - Tenancy (租用) – Default (預設)
 - Number of Availability Zones (AZs) (可用區域 (AZ) 的數量) – 2
 - Customize AZs (自訂可用區域) - 保留預設值。

- Number of public subnet (公有子網路數量) – 2
- Number of private subnets (私有子網路數量) – 2
- Customize subnets CIDR blocks (自訂子網路 CIDR 區塊) – 保留預設值。
- NAT gateways (\$) (NAT 閘道 (\$)) – None (無)
- VPC endpoints (VPC 端點) – None (無)
- DNS options (DNS 選項) – 保留預設值。

Note

Amazon RDS 需要至少兩個不同可用區域中的子網路，以支援多可用區域資料庫執行個體部署。本教學課程會建立單一可用區部署，但這項需求可讓您更輕鬆地在未來轉換為多可用區域資料庫執行個體部署。

6. 選擇 Create VPC (建立 VPC)。

建立公有 Web 伺服器的 VPC 安全群組

接著建立公開存取的 VPC 安全群組。若要連線至 VPC 中的公有 EC2 執行個體，請將傳入規則新增至 VPC 安全群組。這些規則允許流量從網際網路連線。

建立 VPC 安全群組

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 選擇 VPC Dashboard (VPC 儀表板)，再選擇 Security Groups (安全群組)，然後選擇 Create security group (建立安全群組)。
3. 在 Create security group (建立安全群組) 頁面上，設定下列值：
 - 安全群組名稱：**tutorial-securitygroup**
 - 描述：**Tutorial Security Group**
 - VPC: 選擇您先前建立的 VPC，例如：**vpc-### (tutorial-vpc)**
4. 將傳入規則新增至安全群組
 - a. 決定用於使用安全殼層 (SSH) 連接至 VPC 中 EC2 執行個體的 IP 地址。若要判斷公有 IP 地址，您可以在不同的瀏覽器視窗或索引標籤中使用 <https://checkip.amazonaws.com> 中的服務。IP 地址的範例為 203.0.113.25/32。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請找出用戶端電腦所使用的 IP 地址範圍。

⚠ Warning

如果您使用 `0.0.0.0/0` 進行 SSH 存取，則可讓所有 IP 地址使用 SSH 存取您的公有執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，您應只授權特定 IP 地址或特定範圍的地址可使用 SSH 存取您的執行個體。

- b. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
 - c. 針對您的新傳入規則設定下列值，以允許透過 SSH 存取您的 Amazon EC2 執行個體。若是這麼做，則您可以連線至 Amazon EC2 執行個體來安裝 Web 伺服器和其他公用程式。您也會連線至 EC2 執行個體來上傳 Web 伺服器的內容。
 - 類型: **SSH**
 - Source (來源): 來自步驟 a 的 IP 地址或範圍，例如：**203.0.113.25/32**。
 - d. 選擇 Add rule (新增規則)。
 - e. 針對您的新傳入規則設定下列值，以允許透過 HTTP 存取您的 Web 伺服器。
 - Type (類型) : **HTTP**
 - 來源 : **0.0.0.0/0**
5. 請選擇 Create security group (建立安全群組) 以建立安全群組。

請記下安全群組 ID，因為稍後在本教學中會需要它。

建立私有資料庫執行個體的 VPC 安全群組

若要讓您的資料庫執行個體保持私有，請建立第二個安全群組供私有存取。若要連接至 VPC 中的私有資料庫執行個體，請將傳入規則新增至 VPC 安全群組，僅允許來自 Web 伺服器的流量。

建立 VPC 安全群組

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 選擇 VPC Dashboard (VPC 儀表板)，再選擇 Security Groups (安全群組)，然後選擇 Create security group (建立安全群組)。

3. 在 Create security group (建立安全群組) 頁面上，設定下列值：
 - 安全群組名稱：**tutorial-db-securitygroup**
 - 描述：**Tutorial DB Instance Security Group**
 - VPC: 選擇您先前建立的 VPC，例如：**vpc-###** (tutorial-vpc)
4. 將傳入規則新增至安全群組
 - a. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
 - b. 針對您的新傳入規則設定下列值，來允許連接埠 3306 上來自 Amazon EC2 執行個體的 MySQL 流量。如果這樣做，您可以從 Web 伺服器連線至資料庫執行個體。這樣做，您可以將 Web 應用程式中的資料存放並擷取至資料庫。
 - Type (類型)：**MySQL/Aurora**
 - Source (來源)：您先前在本教學課程中建立的 tutorial-securitygroup 安全群組的識別符，例如：**sg-9edd5cfb**。
5. 請選擇 Create security group (建立安全群組) 以建立安全群組。

建立資料庫子網路群組

資料庫子網路群組是您在 VPC 中建立，然後指派給資料庫執行個體的子網路的集合。資料庫子網路群組可讓您在建立資料庫執行個體時指定特定的 VPC。

建立資料庫子網路群組

1. 識別 VPC 中資料庫的私有子網路。
 - a. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 Subnets (子網路)。
 - c. 記下名為 tutorial-subnet-private1-us-west-2a 和 tutorial-subnet-private2-us-west-2b 之子網路的子網路 ID。

建立資料庫子網路群組時，您會需要這些子網路 ID。

2. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。

請確定您連接到 Amazon RDS 主控台，而非 Amazon VPC 主控台。

3. 在導覽窗格中選擇 Subnet groups (子網路群組)。
4. 選擇 Create DB Subnet Group (建立資料庫子網路群組)。

5. 在 Create DB subnet group (建立資料庫子網路群組) 頁面上，於 Subnet group details (子網路群組詳細資訊) 中設定下列值：

- 名稱: **tutorial-db-subnet-group**
- 描述: **Tutorial DB Subnet Group**
- VPC: tutorial-vpc (vpc-###)

6. 在 Add subnets (新增子網路) 區段中，選擇 Availability Zones (可用區域) 和 Subnets (子網路)。

對於此教學課程，為 Availability Zones (可用區域) 選擇 us-west-2a 和 us-west-2b。對於 Subnets (子網路)，選擇您在上一個步驟找到的私有子網路。

7. 選擇 Create (建立)。

您的新資料庫子網路群組會顯示在 RDS 主控台的資料庫子網路群組清單中。您可以選擇資料庫子網路群組，在視窗底部的詳細資訊窗格查看詳細資訊。這些詳細資料包括與群組關聯的所有子網路。

Note

如果您已建立此 VPC 以完成 [教學：建立 Web 伺服器](#) 和 [Amazon RDS 資料庫執行個體](#)，請依照 [建立 Amazon RDS 資料庫執行個體](#) 中的下列指示建立資料庫執行個體。

刪除 VPC

為此教學課程建立 VPC 和其他資源後，如果不再需要這些資源，便可以將它們刪除。

Note

若您在針對此教學課程建立的 VPC 中新增資源，則可能需要先刪除這些資源，才能刪除 VPC。例如，這些資源可能包括 Amazon EC2 執行個體或 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [刪除 VPC](#)。

刪除 VPC 和相關資源

1. 刪除資料庫子網路群組。

- a. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。

- b. 在導覽窗格中選擇 Subnet groups (子網路群組)。
 - c. 選取您要刪除的資料庫子網路群組，例如 tutorial-db-subnet-group。
 - d. 選擇 Delete (刪除)，然後在確認視窗中選擇 Delete (刪除)。
2. 請注意 VPC ID。
 - a. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 VPC。
 - c. 在清單中，找到您建立的 VPC，例如 tutorial-vpc。
 - d. 請記下您所建立的 VPC 的 VPC ID。在稍後的步驟中，您需要 VPC ID。
 3. 刪除安全群組。
 - a. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 Security Groups (安全群組)。
 - c. 選取 Amazon RDS 資料庫執行個體的安全群組，例如 tutorial-db-securitygroup。
 - d. 若為 Actions (動作)，選擇 Delete security groups (刪除安全群組)，然後在確認頁面上選擇 Delete (刪除)。
 - e. 在 Security Groups (安全群組) 頁面上，選取 Amazon EC2 執行個體的安全群組，例如 tutorial-securitygroup。
 - f. 若為 Actions (動作)，選擇 Delete security groups (刪除安全群組)，然後在確認頁面上選擇 Delete (刪除)。
 4. 刪除 VPC。
 - a. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 VPC。
 - c. 選取您要刪除的 VPC，例如 tutorial-vpc。
 - d. 對於 Actions (動作)，請選擇 Delete VPC (刪除 VPC)。

確認頁面會顯示與 VPC 相關聯的其他資源，這些資源也將遭到刪除，包括與其相關聯的子網路。
 - e. 在確認頁面上，輸入 **delete**，然後選擇 Delete (刪除)。

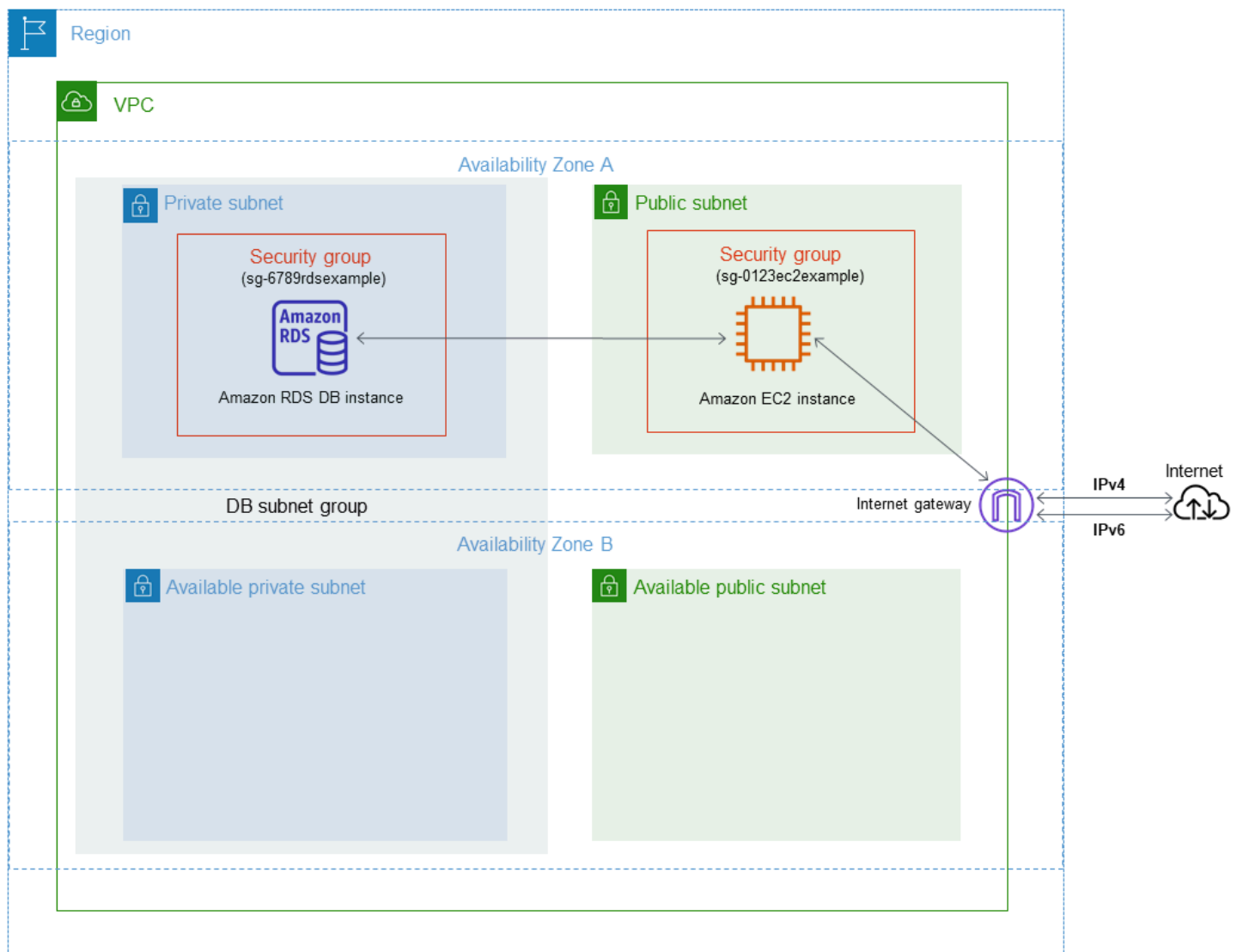
教學課程：建立要與資料庫執行個體搭配使用的 (VPC)(雙堆疊模式)

常用案例包括以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中的資料庫執行個體。此 VPC 與在相同 VPC 中執行的公有 Amazon EC2 執行個體共用資料。

於本教學課程中，您將為此案例建立與以雙堆疊模式執行之資料庫搭配使用的 VPC。雙堆疊模式，可透過 IPv6 定址通訊協定啟用連線。若需 IP 定址的詳細資訊，請參閱 [Amazon RDS IP 定址](#)。

大部分地區都支援雙堆疊網路執行個體。如需更多資訊，請參閱 [區域和版本可用性](#)。若要查看雙堆疊模式的限制，請參閱 [雙堆疊網路資料庫執行個體的限制](#)。

此案例可以下列圖表顯示。



如需其他案例的相關資訊，請參閱 [在 VPC 中存取資料庫執行個體的案例](#)。

您的資料庫執行個體僅需供您的 Amazon EC2 執行個體使用，無需供公有網際網路使用。因此，您建立同時包含公有和私有子網路的 VPC。Amazon EC2 執行個體是在公有子網路中託管，如此其可連接至公有網際網路。資料庫執行個體被託管於私有子網路中。Amazon EC2 執行個體可連線至資料庫執行個體，因為其託管於相同 VPC 中。但是，資料庫執行個體不可用於公有網際網路，以提供更高的安全性。

本教學課程會在單獨的可用區域中設定額外的公有子網路和私有子網路。教學課程不會使用這些子網路。RDS 資料庫子網路群組需要至少兩個可用區域中的子網路。額外的子網路可讓您在未來輕鬆切換到多可用區域資料庫執行個體部署。

若要建立使用雙堆疊模式的資料庫執行個體，請為 Network type (網路類型) 設定指定 Dual-stack mode (雙堆疊模式)。您也可以使用相同設定來修改資料庫執行個體。如需詳細資訊，請參閱 [建立 Amazon RDS 資料庫執行個體](#) 及 [修改 Amazon RDS 資料庫執行個體](#)。

本教學課程說明為 Amazon RDS 資料庫執行個體配置 VPC。如需 Amazon VPC 的詳細資訊，請參閱《Amazon VPC 使用者指南》<https://docs.aws.amazon.com/vpc/latest/userguide/>。

建立含私有和公有子網路的 VPC

使用下列程序來建立一個同時含公有和私有子網路的 VPC。

建立 VPC 和子網路

1. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在的右上角 AWS Management Console，選擇要在其中建立 VPC 的區域。此範例使用 美國東部 (俄亥俄) 區域。
3. 在左上角，選擇 VPC Dashboard (VPC 儀表板)。若要開始建立 VPC，請選擇 Create VPC (建立 VPC)。
4. 在 VPC Settings (VPC 設定) 的 Resources to create (建立資源) 下，選擇 VPC and more (VPC 和更多)。
5. 對於剩下的 VPC 設定，設定這些值：
 - Name tag auto-generation (自動產生名稱標籤) – **tutorial-dual-stack**
 - IPv4 CIDR block (IPv4 CIDR 區塊) – **10.0.0.0/16**
 - IPv6 CIDR block (IPv6 CIDR 區塊) – Amazon-provided IPv6 CIDR block (Amazon 提供的 IPv6 CIDR 區塊)
 - Tenancy (租用) – Default (預設)
 - Number of Availability Zones (AZs) (可用區域 (AZ) 的數量) – 2

- Customize AZs (自訂可用區域) - 保留預設值。
- Number of public subnet (公有子網路數量) – 2
- Number of private subnets (私有子網路數量) – 2
- Customize subnets CIDR blocks (自訂子網路 CIDR 區塊) – 保留預設值。
- NAT gateways (\$) (NAT 閘道 (\$)) – None (無)
- Egress only internet gateway (僅輸出網際網路閘道) – No (否)
- VPC endpoints (VPC 端點) – None (無)
- DNS options (DNS 選項) – 保留預設值。

Note

Amazon RDS 需要至少兩個不同可用區域中的子網路，以支援多可用區域資料庫執行個體部署。本教學課程會建立單一可用區部署，但是，可讓您會來輕鬆轉換為多區域可用資料庫執行個體部署。

6. 選擇建立 VPC。

建立公有 Amazon EC2 執行個體的 VPC 安全群組

接著建立公開存取的 VPC 安全群組。若要連接至 VPC 中的公有 EC2 執行個體，請將傳入規則新增至 VPC 安全群組，以允許流量從網際網路連接。

建立 VPC 安全群組

1. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 選擇 VPC Dashboard (VPC 儀表板)，再選擇 Security Groups (安全群組)，然後選擇 Create security group (建立安全群組)。
3. 在 Create security group (建立安全群組) 頁面上，設定下列值：
 - 安全群組名稱：**tutorial-dual-stack-securitygroup**
 - 描述：**Tutorial Dual-Stack Security Group**
 - VPC: 選擇您先前建立的 VPC，例如 vpc-### (tutorial-dual-stack-vpc)
4. 將傳入規則新增至安全群組
 - a. 決定用於使用安全殼層 (SSH) 連接至 VPC 中 EC2 執行個體的 IP 地址。

網際網路通訊協定第 4 版 (IPv4) 位址的範例為 203.0.113.25/32。網際網路通訊協定第 6 版 (IPv6) 地址範圍的範例為 2001:db8:1234:1a00::/64。

在許多情況下，您可能透過網際網路服務供應商 (ISP) 或是從沒有靜態 IP 地址的防火牆進行連線。若是如此，請找出用戶端電腦所使用的 IP 地址範圍。

⚠ Warning

若您將 0.0.0.0/0 用於 IPv4 或 ::0 用於 IPv6，則可讓所有 IP 地址使用 SSH 存取您的公有執行個體。通常在測試環境中短暫使用此方法是沒有問題的，但在生產環境則不安全。在生產環境中，建議您只授權特定 IP 地址或特定範圍的地址存取您的執行個體。

- b. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
- c. 針對您的新傳入規則設定下列值，來允許透過 Secure Shell (SSH) 存取您的 Amazon EC2 執行個體。若執行此動作，您可連線至 EC2 執行個體來安裝 SQL 用戶端和其他應用程式。指定 IP 地址，讓您可以存取 EC2 執行個體：

- Type (類型) : **SSH**
- Source (來源) : 來自步驟 a 的 IP 地址或範圍。IPv4 IP 地址的範例為 **203.0.113.25/32**。IPv6 IP 地址的範例為 **2001:DB8::/32**。

5. 請選擇 Create security group (建立安全群組) 以建立安全群組。

請記下安全群組 ID，因為稍後在本教學中會需要它。

建立私有資料庫執行個體的 VPC 安全群組

若要讓您的資料庫執行個體保持私有，請建立第二個安全群組供私有存取。若要連線至 VPC 中的私有資料庫執行個體，請將傳入規則新增至 VPC 安全群組。這些規則僅允許來自 Amazon EC2 執行個體的流量。

建立 VPC 安全群組

1. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 選擇 VPC Dashboard (VPC 儀表板)，再選擇 Security Groups (安全群組)，然後選擇 Create security group (建立安全群組)。
3. 在 Create security group (建立安全群組) 頁面上，設定下列值：

- 安全群組名稱：**tutorial-dual-stack-db-securitygroup**
 - 描述：**Tutorial Dual-Stack DB Instance Security Group**
 - VPC: 選擇您先前建立的 VPC，例如 `vpc-###` (tutorial-dual-stack-vpc)
4. 將傳入規則新增至安全群組：
 - a. 在 Inbound rules (傳入規則) 區段中，選擇 Add rule (新增規則)。
 - b. 針對您的新傳入規則設定下列值，來允許連接埠 3306 上來自 Amazon EC2 執行個體的 MySQL 流量。如果這樣做，您可以從 EC2 執行個體連線至資料庫執行個體。這樣做表示您可以從 EC2 執行個體將資料傳送至您的資料庫。
 - Type (類型)：MySQL/Aurora
 - Source (來源)：您先前在本教學課程中建立的 tutorial-dual-stack-securitygroup 安全群組的識別符，例如 sg-9edd5cfb。
 5. 若要建立安全群組，請選擇 Create security group (建立安全群組)。

建立資料庫子網路群組

資料庫子網路群組是您在 VPC 中建立，然後指派給資料庫執行個體的子網路的集合。透過使用資料庫子網路群組，您可在建立資料庫執行個體時指定特定的 VPC。如要建立與 DUAL 相容的資料庫子網路群組，則所有子網路必須與 DUAL 相容。如要成為 DUAL 相容，子網路必須具有與其關聯的 IPv6 CIDR。

建立資料庫子網路群組

1. 識別 VPC 中資料庫的私有子網路。
 - a. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 Subnets (子網路)。
 - c. 記下名為 tutorial-dual-stack-subnet-private1-us-west-2a 和 tutorial-dual-stack-subnet-private2-us-west-2b 之子網路的子網路 ID。

當您建立資料庫子網路群組時，您將需要子網路 ID。

2. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。

請確定您連接到 Amazon RDS 主控台，而非 Amazon VPC 主控台。

3. 在導覽窗格中選擇 Subnet groups (子網路群組)。

4. 選擇 Create DB Subnet Group (建立資料庫子網路群組)。
5. 在 Create DB subnet group (建立資料庫子網路群組) 頁面上，於 Subnet group details (子網路群組詳細資訊) 中設定下列值：
 - 名稱: **tutorial-dual-stack-db-subnet-group**
 - 描述: **Tutorial Dual-Stack DB Subnet Group**
 - VPC: tutorial-dual-stack-vpc (**vpc- ###**)
6. 在 Add subnets (新增子網路) 區段中，選擇 Availability Zones (可用區域) 和 Subnets (子網路) 選項的值。

對於此教學課程，為 Availability Zones (可用區域) 選擇 us-east-2a 和 us-east-2b。對於 Subnets (子網路)，選擇您在上一個步驟找到的私有子網路。
7. 選擇建立。

您的新資料庫子網路群組會顯示在 RDS 主控台的資料庫子網路群組清單中。您可以選擇資料庫子網路群組來查看其詳細資訊。其中包括受支援的定址通訊協定、與該組相關聯的所有子網路，以及資料庫子網路群組支援的網路類型。

在雙堆疊模式下建立 Amazon EC2 執行個體

若要建立 Amazon EC2 執行個體，請遵循 Amazon [EC2 使用者指南中的使用新啟動執行個體精靈](#) 啟動執行個體中的指示。

在 Configure Instance Details (設定執行個體詳細資訊) 頁面上設定這些值，而其他值都維持預設值：

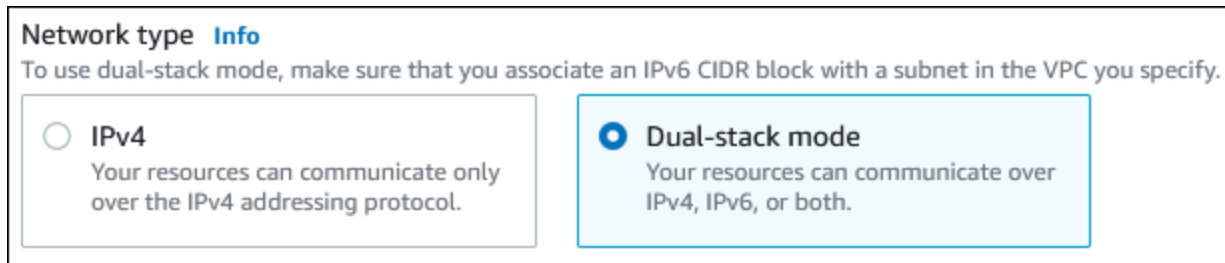
- 網路：選擇具備公有和私有子網路的現有 VPC，例如 [建立含私有和公有子網路的 VPC](#) 中建立的 tutorial-dual-stack-vpc (vpc-*identifier*)。
- 子網路 — 選擇現有的公用子網路，例如子網路 **###** | tutorial-dual-stack-subnet-public1-US-東部 2a | 建立於中。 [建立公有 Amazon EC2 執行個體的 VPC 安全群組](#)
- Auto-assign Public IP (自動指派公有 IP) – 選擇 Enable (啟用)。
- Auto-assign IPv6 IP (自動指派 IPv6 IP) – 選擇 Enable (啟用)。
- Firewall (security groups) 防火牆 (安全群組) – 選擇 Select an existing security group (選取現有的安全群組)。
- Common security groups (一般安全群組) - 選擇現有的安全群組，例如在 [建立公有 Amazon EC2 執行個體的 VPC 安全群組](#) 中建立的 tutorial-securitygroup。請確定您選擇的安全性群組包含安全殼層 (SSH) 和 HTTP 存取的輸入規則。

在雙堆疊模式下建立資料庫執行個體

在此步驟中，您將建立在雙堆疊模式下執行的資料庫執行個體。

建立資料庫執行個體

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在主控台的右上角，選擇您 AWS 區域 要建立資料庫執行個體的位置。此範例使用 美國東部 (俄亥俄) 區域。
3. 在導覽窗格中，選擇 Databases (資料庫)。
4. 選擇 Create database (建立資料庫)。
5. 在 Create database (建立資料庫) 頁面上，確保已選擇 Standard create (標準建立) 選項，接著選擇 MySQL 資料庫引擎類型。
6. 在 Connectivity (連線) 區段中，設定下列值：
 - Network type (網路類型) – 選擇 Dual-stack mode (雙堆疊模式)。



- Virtual private cloud (VPC) 虛擬私有雲端 (VPC) – 選擇具備公有和私有子網路的現有 VPC，例如 [建立含私有和公有子網路的 VPC](#) 中建立的 tutorial-dual-stack-vpc (vpc-###)。

VPC 必須具有位於不同可用區域的子網路。

- DB Subnet group (資料庫子網路群組) – 選擇 VPC 的資料庫子網路群組，例如 [建立資料庫子網路群組](#) 中建立的 tutorial-dual-stack-db-subnet-group。
- Public access (公有存取) – 選擇 No (否)。
- VPC security group (firewall) (VPC 安全群組 (防火牆)) – 選取 Choose existing (選擇現有)。
- Existing VPC security groups (現有 VPC 安全群組) – 選擇設定為私有存取的現有 VPC 安全群組，例如在 [建立私有資料庫執行個體的 VPC 安全群組](#) 中建立的 tutorial-dual-stack-db-securitygroup。

選擇與各項相關聯的 X，以移除其他安全群組，例如預設安全群組。

- Availability Zone (可用區域) – 選擇 us-west-2a。

若要避免跨可用區域流量，請確定資料庫執行個體和 EC2 執行個體位於相同的可用區域。

7. 在其餘區段，指定資料庫執行個體的設定。如需每項設定的相關資訊，請參閱 [資料庫執行個體的設定](#)。

連接至您的 Amazon EC2 執行個體和資料庫執行個體

在雙堆疊模式下建立 Amazon EC2 執行個體和資料庫執行個體後，您可以使用 IPv6 通訊協定連線到每個執行個體。若要使用 IPv6 通訊協定 Connect 到 Amazon EC2 執行個體，請按照 Amazon EC2 使用者指南中[連接到 Linux 執行個體](#)中的指示進行操作。

若要從 Amazon EC2 執行個體連線至您的 RDS for MySQL 資料庫執行個體，請依照[連線至 MySQL 資料庫執行個體](#)中的指示進行。

刪除 VPC

為此教學課程建立 VPC 和其他資源後，如果不再需要這些資源，便可以將它們刪除。

若您在針對此教學課程建立的 VPC 中新增資源，則可能需要先刪除這些資源，才能刪除 VPC。資源範例為 Amazon EC2 執行個體或資料庫執行個體。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[刪除 VPC](#)。

刪除 VPC 和相關資源

1. 刪除資料庫子網路群組：
 - a. 前往 <https://console.aws.amazon.com/rds/>，開啟 Amazon RDS 主控台。
 - b. 在導覽窗格中選擇 Subnet groups (子網路群組)。
 - c. 選取要刪除的資料庫子網路群組，例如 tutorial-db-subnet-group。
 - d. 選擇 Delete (刪除)，然後在確認視窗中選擇 Delete (刪除)。
2. 請記下 VPC ID：
 - a. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 VPC。
 - c. 在清單中，識別您建立的 VPC，例如 tutorial-dual-stack-vpc。
 - d. 請記下您所建立 VPC 的 VPC ID。於後續步驟中，您會需要此 VPC ID。
3. 刪除安全群組：

- a. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 Security Groups (安全群組)。
 - c. 選取 Amazon RDS 資料庫執行個體的安全群組，例如 tutorial-dual-stack-db-securitygroup。
 - d. 若為 Actions (動作)，選擇 Delete security groups (刪除安全群組)，然後在確認頁面上選擇 Delete (刪除)。
 - e. 在 Security Groups (安全群組) 頁面上，選取 Amazon EC2 執行個體的安全群組，例如 tutorial-dual-stack-securitygroup。
 - f. 若為 Actions (動作)，選擇 Delete security groups (刪除安全群組)，然後在確認頁面上選擇 Delete (刪除)。
4. 刪除 NAT 閘道：
- a. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 NAT Gateways (NAT 閘道)。
 - c. 選取您建立之 VPC 的 NAT 閘道。使用 VPC ID 識別正確的 NAT 閘道。
 - d. 若為 Actions (動作)，選擇 Delete NAT gateway (刪除 NAT 閘道)。
 - e. 在確認頁面上，輸入 **delete**，然後選擇 Delete (刪除)。
5. 刪除 VPC
- a. 前往 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
 - b. 選擇 VPC Dashboard (VPC 儀表板)，然後選擇 VPC。
 - c. 選取您要刪除的 VPC，例如 tutorial-dual-stack-vpc。
 - d. 對於 Actions (動作)，請選擇 Delete VPC (刪除 VPC)。
- 確認頁面會顯示與 VPC 相關聯的其他資源，這些資源也將遭到刪除，包括與其相關聯的子網路。
- e. 在確認頁面上，輸入 **delete**，然後選擇 Delete (刪除)。
6. 釋放彈性 IP 地址：
- a. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
 - b. 選擇 EC2 Dashboard (EC2 儀表板)，然後選擇 Elastic IPs (彈性 IP)。
 - c. 選取您要釋放的彈性 IP 地址。
 - d. 若為 Actions (動作)，選擇 Release Elastic IP addresses (釋放彈性 IP 地址)。
 - e. 在確認頁面上，選擇 Release (釋放)。

將不在 VPC 中的資料庫執行個體移入 VPC

EC2-Classic 平台上的部分舊型資料庫執行個體不在 VPC 中。若您的資料庫執行個體不在 VPC 中，您可以使用 AWS Management Console 將資料庫執行個體輕鬆移入 VPC。您必須先建立 VPC，才能將不在 VPC 中的資料庫執行個體移入 VPC。

EC2-Classic 在 2022 年 8 月 15 日淘汰。如果您還沒有從 EC2-Classic 遷移至 VPC，建議您盡快這麼做。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [從 EC2-Classic 遷移至 VPC](#)，以及 [EC2-Classic Networking is Retiring – Here's How to Prepare](#) (EC2-Classic 網路正在淘汰 - 本文介紹如何準備) 部落格文章。

Important

如果您是第一次使用 Amazon RDS 的新客戶，且先前從未建立過資料庫執行個體，或是要在未曾使用的 AWS 區域中建立資料庫執行個體，則在幾乎所有情況下，使用的平台都是 EC2-VPC，並擁有預設的 VPC。如需在 VPC 中使用資料庫執行個體的相關資訊，請參閱 [在 VPC 中使用資料庫執行個體](#)。

請依照以下步驟，為您的資料庫執行個體建立 VPC。

- [步驟 1：建立 VPC](#)
- [步驟 2：建立資料庫子網路群組](#)
- [步驟 3：建立 VPC 安全群組](#)

建立 VPC 後，請依照以下步驟，將您的資料庫執行個體移入 VPC。

- [更新資料庫執行個體的 VPC](#)

強烈建議您在遷移之前，立即建立資料庫執行個體的備份。這麼做可確保在遷移失敗時，可以還原資料。如需更多詳細資訊，請參閱 [備份、還原和匯出資料](#)。

將資料庫執行個體移入 VPC 有幾個限制，說明如下。

- 上一代資料庫執行個體類別 – VPC 平台上可能不支援上一代資料庫執行個體類別。將資料庫執行個體移至 VPC 時，請選擇 db.m3 或 db.r3 資料庫執行個體類別。將資料庫執行個體移至 VPC 後，您

可以擴展資料庫執行個體以使用較新的資料庫執行個體類別。如需 VPC 支援執行個體類別的完整清單，請參閱 [Amazon RDS 執行個體類型](#)。

- 異地同步備份 – 目前系統不支援將不在 VPC 中的異地同步備份資料庫執行個體移入 VPC。若要將資料庫執行個體移至 VPC，請先修改資料庫執行個體，使其成為單一可用區部署。將 Multi-AZ deployment (異地同步備份部署) 設定變更為 No (否)。將資料庫執行個體移至 VPC 後，再次將其修改為異地同步備份部署。如需更多詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。
- 讀取複本 – 目前不支援將具有不在 VPC 中的僅供讀取複本的資料庫執行個體移至 VPC。若要將資料庫執行個體移至 VPC，請先刪除其所有讀取複本。將資料庫執行個體移至 VPC 後，請重新建立讀取複本。如需更多詳細資訊，請參閱 [使用資料庫執行個體僅供讀取複本](#)。
- 選項群組 – 如果您將資料庫執行個體移至 VPC，且資料庫執行個體使用自訂選項群組，請變更與資料庫執行個體關聯的選項群組。選項群組會依平台而異，隨著執行個體移入 VPC，平台也會一併改變。若要在這種情況下使用自訂選項群組，可將預設 VPC 選項群組指派給資料庫執行個體、指派您移動目的地中其他資料庫執行個體所使用的選項群組，或是建立新的選項群組，並將其指派給該資料庫執行個體。如需更多詳細資訊，請參閱 [使用選項群組](#)。

以最短的停機時間，將不在 VPC 中的資料庫執行個體移到 VPC 中的替代方案

使用下列替代方案，您可以將不在 VPC 中的資料庫執行個體移到 VPC 中，而且停機時間最小。這些替代方案會對來源資料庫執行個體造成最小的中斷，並允許它在遷移期間提供使用者流量服務。不過，遷移至 VPC 所需的時間會根據資料庫大小和即時工作負載特性而有所不同。

- AWS Database Migration Service (AWS DMS) – AWS DMS 可啟用資料的即時遷移，同時保持來源資料庫執行個體完全可運作，但它只會複寫一組有限的 DDL 陳述式。AWS DMS 不會傳播項目，例如索引、使用者、權限、預存程序，及其他未與資料表資料直接相關的資料庫變更。此外，AWS DMS 不會自動使用 RDS 快照進行初始資料庫執行個體建立，這會增加遷移時間。如需詳細資訊，請參閱 [AWS Database Migration Service](#)。
- 資料庫快照還原或時間點復原 – 您可以藉由還原資料庫執行個體，或藉由將資料庫執行個體還原至某時間點，以將資料庫執行個體移至 VPC。如需詳細資訊，請參閱 [從資料庫快照還原](#) 及 [將資料庫執行個體還原至指定的時間](#)。

Amazon RDS 的配額和條件限制

您可以在下面找到 Amazon RDS 的資源配額和命名限制的說明。

主題

- [Amazon RDS 中的配額](#)
- [Amazon RDS 中的命名限制](#)
- [資料庫連線數上限](#)
- [Amazon RDS 中的檔案大小限制](#)

Amazon RDS 中的配額

每個 AWS 帳戶對於每個 AWS 區域，都有可建立的 Amazon RDS Amazon 資源數量的配額。在達到資源的配額後，建立該資源的額外呼叫便會失敗並發生例外狀況。

下表列出每個 AWS 區域的資源及其配額。

名稱	預設	可調整	描述
每個資料庫安全群組的授權	每個受支援的區域：20	否	每個資料庫安全群組的安全群組授權數
自訂引擎版本	每個受支援的區域：40	<u>是</u>	在目前區域中，此帳戶所允許的自訂引擎版本數量上限
資料庫叢集參數群組	每個受支援的區域：50	否	資料庫叢集參數群組的數量上限
資料庫叢集	每個受支援的區域：40	<u>是</u>	在目前區域中，此帳戶允許地的 Aurora 叢集數量上限

名稱	預設	可調整	描述
資料庫執行個體	每個受支援的區域：40	是	在目前區域中，此帳戶允許地的資料庫執行個體數量上限
資料庫子網路群組	每個受支援的區域：50	是	資料庫子網路群組的數量上限
資料 API HTTP 要求主體大小	每個受支援的區域：4 MB	否	HTTP 請求內文允許的大小上限。
資料 API 並行叢集密碼配對上限	每個受支援的區域：30	否	目前 AWS 區域中此帳戶的並行資料 API 請求中，Aurora 無伺服器 v1 資料庫叢集和機密的唯一對數目上限。
資料 API 並行請求上限	每個受支援的區域：500	否	使用相同密碼且可以同時處理的 Aurora 無伺服器 v1 資料庫叢集的資料 API 要求數目上限。其他請求會排入佇列，並在同處理序請求完成時進行處理。
資料 API 結果集大小上限	每個受支援的區域：1 MB	否	資料 API 可傳回的資料庫結果集大小上限。
JSON 回應字串的資料 API 大小上限	每個受支援的區域：10 MB	否	RDS 資料 API 傳回的簡化 JSON 回應字串大小上限。

名稱	預設	可調整	描述
每秒的資料 API 請求數	每個受支援的區域：每秒 1,000	否	在目前 AWS 區域中，此帳戶每秒允許的資料 API 要求數目上限。此配額僅適用於 Amazon Aurora 無伺服器 v1 叢集。
事件訂閱	每個受支援的區域：20	是	事件訂閱的數量上限
每個資料庫叢集的 IAM 角色	每個受支援的區域：5	是	與資料庫叢集相關聯之 IAM 角色的數量上限
每個資料庫執行個體的 IAM 角色	每個受支援的區域：5	是	與資料庫執行個體相關聯的 IAM 角色數量上限
手動資料庫叢集快照	每個受支援的區域：100	是	手動資料庫叢集快照的數量上限
資料庫執行個體手動快照	每個受支援的區域：100	是	手動資料庫執行個體快照的數量上限
選項群組	每個受支援的區域：20	是	選項群組的數量上限
參數群組	每個受支援的區域：50	是	參數群組的數量上限
代理	每個受支援的區域：20	是	此帳戶在當 AWS 前區域中允許的最大代理數量
每個主要項目的僅供讀取複本	每個受支援的區域：15	是	每個主資料庫執行個體的僅供讀取複本數量上限 此配額無法針對 Amazon Aurora 調整。

名稱	預設	可調整	描述
預留資料庫執行個體	每個受支援的區域：40	是	目前 AWS 區域中此帳戶允許的預留資料庫執行個體數目上限
每個安全群組的規則數	每個受支援的區域：20	否	每個資料庫安全群組的規則上限
安全群組	每個受支援的區域：25	是	資料庫安全群組數量上限
安全群組 (VPC)	每個受支援的區域：5	否	每個 Amazon VPC 之資料庫安全群組數量上限
每個資料庫子網路群組的子網路	每個受支援的區域：20	否	每個資料庫子網路群組的子網路數目上限
每個資源的標籤	每個受支援的區域：50	否	每個 Amazon RDS 資源的標籤數量上限
所有資料庫執行個體的儲存總量	所有受支援的區域：100,000 GB	是	EBS 磁碟區上所有 Amazon RDS 資料庫執行個體加在一起的儲存體總量上限 (以 GB 為單位) 此配額不適用於 Amazon Aurora，因為每個資料庫叢集的叢集磁碟區上限為 128 TiB。

Note

根據預設，您最多可以擁有 40 個資料庫執行個體。RDS 資料庫執行個體、Aurora 資料庫執行個體、Amazon Neptune 執行個體和 Amazon DocumentDB 執行個體皆適用此配額。

下列限制適用於 Amazon RDS 資料庫執行個體：

- 每個 SQL Server 資料庫版本 (Enterprise、Standard、Web 和 Express) 最多有 10 個「已包含授權」模式的執行個體
- "license-included" 模式下的 10 for Oracle
- 40 代表「bring-your-own-license」授權模式下的 Db2
- 40 個資料庫執行個體可以是 MySQL、MariaDB 或 PostgreSQL
- 40 適用於甲骨文在 "bring-your-own-license" (BYOL) 授權模式下

如果您的應用程式需要更多資料庫執行個體，您可以透過開啟 [Service Quotas 主控台](#) 來請求額外的資料庫執行個體。在導覽窗格中，選擇 AWS services (AWS 服務)。選擇 Amazon Relational Database Service (Amazon RDS)，再選擇配額，然後按照指示請求增加配額。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的 [請求提高配額](#)。

對於 RDS for Oracle 和 RDS for SQL Server，每個區域每個來源資料庫的僅供讀取複本限制為 5。

由管理的備份視 AWS Backup 為手動資料庫快照，但不會計入手動快照配額中。如需相關資訊 AWS Backup，請參閱 [AWS Backup 發人員指南](#)。

如果您使用任何 RDS API 操作，且超過每秒呼叫次數的預設配額，Amazon RDS API 會發出類似下面的錯誤。

ClientError：呼叫 *API_name* 作業時發生錯誤 (ThrottlingException)：超過速率。

在這裡，減少每秒呼叫次數。配額旨在涵蓋大多數使用案例。如果需要更高的配額，您可以使用下列其中一個選項要求增加配額：

- 從主控台開啟「[Service Quotas](#)」主控台。
- 從中 AWS CLI，使用指 [request-service-quota-increase](#) AWS CLI 令。

如需詳細資訊，請參閱 [Service Quotas 使用者指南](#)。

Amazon RDS 中的命名限制

下表說明 Amazon RDS 中的命名限制。

資源或項目	限制
資料庫執行個體識別符	<p>識別符具有這些命名條件限制：</p> <ul style="list-style-type: none">• 必須包含 1–63 個英數字元或連字號。• 第一個字元必須是字母。• 不能以一個連字號結尾或是連續包含兩個連字號。• 每個 AWS 帳戶每個 AWS 區域的所有資料庫執行個體必須是唯一的。
資料庫名稱	<p>資料庫名稱條件限制會因每個資料庫引擎而有所不同。如需更多詳細資訊，請參閱建立每個資料庫執行個體時的可用設定。</p> <div data-bbox="688 806 1507 1073"><p> Note</p><p>這種方法不適用於 SQL Server。針對 SQL Server，您可以在建立資料庫執行個體之後建立資料庫。</p></div>
主要使用者名稱	<p>每個資料庫引擎都有不同的主要使用者名稱限制。如需更多詳細資訊，請參閱建立每個資料庫執行個體時的可用設定。</p>
Master password (主要密碼)	<p>資料庫主要使用者密碼可包含任何可印刷的 ASCII 字元，但 /、'、"、@ 或空格除外。對於 Oracle，& 是額外的字元限制。密碼會有下列數量的可列印 ASCII 字元，取決於資料庫引擎：</p> <ul style="list-style-type: none">• 二級分貝：八至五十五• MariaDB 及 MySQL：8–41• Oracle：8–30• SQL Server 及 PostgreSQL：8–128
資料庫參數群組名稱	<p>這些名稱具有下列條件限制：</p> <ul style="list-style-type: none">• 必須包含 1–255 個英數字元。

資源或項目	限制
	<ul style="list-style-type: none"> 第一個字元必須是字母。 允許連字號，但名稱不得以連字號為結尾或包含兩個連續的連字號。
資料庫子網路群組名稱	<p>這些名稱具有下列條件限制：</p> <ul style="list-style-type: none"> 必須包含 1–255 個字元。 允許英數字母、空格、連字號、底線和句點。

資料庫連線數上限

同時資料庫連線的最大數量取決於資料庫引擎類型和資料庫執行個體類別的記憶體配置。連線數目上限通常會在與資料庫執行個體相關聯的參數群組中設定。例外狀況是 Microsoft SQL Server，它是在 SQL Server Management Studio (SSMS) 中資料庫執行個體的伺服器屬性中設定。

資料庫連線會耗用記憶體。如果其中一個參數的設定值過高，則可能會導致記憶體不足的情況出現，進而導致資料庫執行個體處於 incompatible-parameters (不相容參數) 狀態。如需詳細資訊，請參閱 [診斷和解決記憶體限制的不相容參數狀態](#)。

如果您的應用程式經常開啟和關閉連線，或者保持大量長期連線開啟，我們建議您使用 Amazon RDS Proxy。RDS 代理是個完全受管、高可用性的資料庫代理，其使用連線集區，安全且高效地共用資料庫連線。如要進一步了解 RDS 代理，請參閱 [使用 Amazon RDS Proxy](#)。

Note

對於 Oracle，您可以設定使用者程序，以及使用者與系統工作階段的數量上限。
對於 Db2，您無法設定最大連線數。限制是 64 萬。

資料庫連線數上限

資料庫引擎	參數	允許的值	預設值	描述
MariaDB 及 MySQL	max_connections	1–100000	所有 MariaDB 和 MySQL 版本的預設值，但	允許的同時用戶端連線數目

資料庫引擎	參數	允許的值	預設值	描述
			MariaDB 10.5 和 10.6 版除外： 愉景灣 InstanceClassMemory MariaDB 10.5 和 10.6 版的預設值： 最少 ({{資料庫 InstanceClassMemory /25165760}})	
			<div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>無論是哪一種情況，如果預設值計算結果是大於 16,000 的值，則 Amazon RDS 會將 MariaDB 和 MySQL 資料庫執行個體的限制設定為 16,000。</p> </div>	
Oracle	processes	80–20000	最少 (20000 分貝 InstanceClassMemory /9868951))	使用者程序
	sessions	100–65535	–	使用者和系統工作階段
PostgreSQL	max_connections	6–8388607	最少 ({{分貝 InstanceClassMemory /9531392}})	同時連線的數目上限

資料庫引擎	參數	允許的值	預設值	描述
SQL Server	同時連線的數目上限	0–32767	0 (無限制)	同時連線的數目上限

`DBInstanceClassMemory` 是以位元組為單位。如需如何計算此值的詳細資訊，請參閱 [指定資料庫參數](#)。由於為作業系統和 RDS 管理程序保留了記憶體，此記憶體大小會小於 [的資料庫執行個體類別的硬體規格](#) 中顯示的值 (單位為 gibibyte (GiB))。

例如，某些資料庫執行個體類別具有 8 GiB 的記憶體，也就是 8,589,934,592 個位元組。對於在具有 8 GiB 記憶體 (例如 `db.m7g.large`) 的資料庫執行個體類別上執行的 MySQL 資料庫執行個體，則使用總記憶體的方程式會是 $8589934592 / 12582880 = 683$ 。然而，變數 `DBInstanceClassMemory` 會自動減去為作業系統和管理資料庫執行個體的 RDS 程序所預留的數量。然後將剩餘部分除以 12,582,880。此計算得出的 `max_connections` 值大約為 630，而不是 683。此值取決於資料庫執行個體類別和資料庫引擎。

當 MariaDB 或 MySQL 資料庫執行個體在小型資料庫執行個體類別 (例如 `db.t3.micro` 或 `db.t3.small`) 上執行時，可用的記憶體總數較少。對於這些資料庫執行個體類別，RDS 會預留記憶體的一大部分，而這會影響值 `max_connections`。例如，在 `db.t3.micro` 資料庫執行個體類別上執行的 MySQL 資料庫執行個體，其預設最大連線數約為 60。您可以透過連線至資料庫 MariaDB 或 MySQL 資料庫執行個體並執行下列 SQL 命令來判斷其 `max_connections` 值：

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Amazon RDS 中的檔案大小限制

檔案大小限制適用於特定 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱下列引擎特定限制：

- [Amazon RDS 中的 MariaDB 檔案大小限制](#)
- [Amazon RDS 中的 MySQL 檔案大小限制](#)
- [Amazon RDS 中的 Oracle 檔案大小限制](#)

Amazon RDS 故障診斷

下列各節可幫助您對 Amazon RDS 和 Amazon Aurora 資料庫執行個體的問題進行故障診斷。

主題

- [無法連線至 Amazon RDS 資料庫執行個體](#)
- [Amazon RDS 安全問題](#)
- [對不相容的網路狀態進行故障診斷](#)
- [重新設定資料庫執行個體擁有者密碼](#)
- [Amazon RDS 資料庫執行個體停機或重新開機](#)
- [Amazon RDS 資料庫參數變更未生效](#)
- [Amazon RDS 資料庫執行個體用盡儲存體](#)
- [Amazon RDS 資料庫執行個體容量不足](#)
- [Amazon RDS 中的可用記憶體問題](#)
- [MySQL 和 MariaDB 問題](#)
- [無法將備份保留期間設定為 0](#)

如需使用 Amazon RDS API 偵錯問題的相關資訊，請參閱[對 Amazon RDS 上的應用程式進行故障診斷](#)。

無法連線至 Amazon RDS 資料庫執行個體

無法連線至資料庫執行個體時，下列是常見的原因：

- 傳入規則 – 您的本機防火牆強制執行的存取規則，與您獲授權可存取資料庫執行個體的 IP 地址可能不符。問題很可能在於安全群組中的傳入規則。

預設情況下，資料庫執行個體不允許存取。存取是透過與 VPC 相關聯的安全群組授予，該群組允許可傳入和傳出資料庫執行個體的流量。如有必要，請將特定情況的傳入和傳出規則新增至安全群組。您可以指定 IP 地址、IP 地址範圍，或其他 VPC 安全群組。

Note

新增傳入規則時，您可以針對 Source (來源) 選擇 My IP (我的 IP)，以允許從瀏覽器中偵測到的 IP 位址存取資料庫執行個體。

如需設定安全群組的詳細資訊，請參閱[建立安全群組以存取在您的 VPC 中您的資料庫執行個體](#)。

Note

不允許從 169.254.0.0/16 範圍內的 IP 地址連線的用戶端。這是「自動私有 IP 定址範圍」(APIPA)，它用於本機連結定址。

- 公開存取性– 若要從 VPC 外部連線至資料庫執行個體 (例如，使用用戶端應用程式)，則執行個體必須具有指派給它的公用 IP 位址。

若要讓執行個體可供公開存取，請修改並選擇 Public accessibility (公開存取性) 下方的 Yes (是)。如需更多詳細資訊，請參閱 [在 VPC 中的網際網路中隱藏資料庫執行個體](#)。

- Port (連線埠) – 由於您的本機防火牆限制，建立資料庫執行個體時指定的連線埠無法用來傳送或接收通訊。若要判斷您的網路是否允許將指定的連線埠用於傳入和傳出通訊，請洽詢網路管理員。
- Availability (可用性) – 對於新建立的資料庫執行個體，資料庫執行個體的狀態為 creating，直到資料庫執行個體可供使用為止。狀態變更為 available 時，您便能連線至資料庫執行個體。根據資料庫執行個體的大小，可能需要最多 20 分鐘的時間，執行個體才會可用。
- 網際網路閘道 – 若要開放公開存取資料庫執行個體，其資料庫子網路群組中的子網路必須具備網際網路閘道。

設定子網路的網際網路閘道

1. 登入 AWS Management Console 並開啟 Amazon RDS 主控台，網址為 <https://console.aws.amazon.com/rds/>。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇資料庫執行個體的名稱。
3. 在 Connectivity & security (連線能力和安全性) 標籤中，寫下 VPC 下的 VPC ID，以及 Subnets (子網路) 下的子網路 ID。
4. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
5. 在導覽窗格中，選擇 Internet Gateways (網際網路閘道)。請確認已經有網際網路閘道連結到 VPC。否則，請選擇 Create Internet Gateway (建立網際網路閘道)，來建立網際網路閘道。選

取網際網路閘道，然後選取 Attach to VPC (連結至 VPC)，接著遵循指示，來將此閘道連結至 VPC。

6. 在導覽窗格中，選取 Subnets (子網)，然後選取子網。
7. 在 Route Table (路由表) 索引標籤中，確認有包含 `0.0.0.0/0` 的路由做為目的地，以及有 VPC 的網際網路閘道做為目標。

如果您是使用其 IPv6 地址來連線到執行個體，請確認有針對所有 IPv6 流量 (`::/0`) 的路由，來指向網際網路閘道。若否，請執行下列操作：

- a. 選擇路由表 ID (rtb-xxxxxxx) 以導覽至路由表。
- b. 在 Routes (路由) 標籤中，選擇 Edit routes (編輯路由)。選擇 Add route (新增路由)，使用 `0.0.0.0/0` 做為目的地，並以網際網路閘道為目標。

若是 IPv6，請選擇 Add route (新增路由)，使用 `::/0` 做為目的地，並以網際網路閘道為目標。

- c. 選擇 Save routes (儲存路由)。

此外，若要嘗試連線至 IPv6 端點，請確保用戶端 IPv6 地址範圍有權連線至資料庫執行個體。

如需詳細資訊，請參閱 [在 VPC 中使用資料庫執行個體](#)。

對於引擎特定連線問題，請參閱下列主題：

- [對您的 SQL Server 資料庫執行個體的連線進行故障診斷](#)
- [對您的 Oracle 資料庫執行個體的連線進行故障診斷](#)
- [針對您的 RDS for PostgreSQL 執行個體連線進行疑難排解](#)
- [MySQL 及 MariaDB 連線上限](#)

測試資料庫執行個體的連線

您可以使用通用的 Linux 或 Microsoft Windows 工具來測試您對資料庫執行個體的連線。

您可以從 Linux 或 Unix 終端機測試連線，方法是輸入下列命令。將 `DB-instance-endpoint` 取代為端點並將 `port` 取代為資料庫執行個體的連線埠。

```
nc -zv DB-instance-endpoint port
```

例如，下列示範範例命令和傳回值：

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299

Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/vvvr-data] succeeded!
```

Windows 使用者可以使用 Telnet 來測試對資料庫執行個體的連線。Telnet 動作僅支援用於測試連線。如果連線成功，動作不會傳回訊息。如果連線不成功，您會收到類似如下的錯誤訊息：

```
C:\>telnet sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com 819

Connecting To sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com...Could not
open
connection to the host, on port 819: Connect failed
```

如果 Telnet 動作成功傳回，您的安全群組即已正確設定。

Note

Amazon RDS 不接受網際網路控制訊息通訊協定 (ICMP) 流量，包括 ping。

對連線身分驗證進行故障診斷

在某些情況下，您可以連線至資料庫執行個體，但會收到身分驗證錯誤。在這些情況下，您可能會重設資料庫執行個體的主要使用者密碼。請修改 RDS 執行個體，以便執行此動作。

如需修改 資料庫執行個體的詳細資訊，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

Amazon RDS 安全問題

為避免安全性問題，切勿使用您的主要使用 AWS 者名稱和密碼作為使用者帳戶。最佳做法是使用主版 AWS 帳戶 建立使用者，並將這些使用者指派給 DB 使用者帳戶。如果必要，也可以使用您的主要帳戶來建立其他使用者帳戶。

如需建立使用者的相關資訊，請參閱[在您的 AWS 帳戶中建立 IAM 使用者](#)。如需在中建立使用者的相關資訊 AWS IAM Identity Center，請參閱[在 IAM 身分中心中管理身分](#)。

錯誤訊息「無法擷取帳戶屬性，某些主控台功能可能受損。」

您可以基於數個原因而收到此錯誤。這可能是因為您的帳戶缺少權限，或是您的帳戶並未適當設定。如果您的帳戶是新帳戶，您可能就不需等待帳戶就緒。如果這是現有的帳戶，您的存取政策中可能缺乏許可，因此無法執行某些動作 (例如建立資料庫執行個體)。若要修正問題，您的管理員需要為您的帳戶提供必要角色。如需詳細資訊，請參閱 [IAM 文件](#)。

對不相容的網路狀態進行故障診斷

不相容的網路狀態，表示資料庫可能在資料庫層級仍可存取，但您無法修改或重新啟動它。

原因

資料庫執行個體的不相容網路狀態可能是下列其中一個動作所造成：

- 修改資料庫執行個體類別。
- 修改資料庫執行個體以使用多可用區域資料庫叢集部署。
- 因維護事件而更換主機。
- 啟動更換用資料庫執行個體。
- 從快照備份還原。
- 啟動之前已停止的資料庫執行個體。

解析度

使用start-db-instance 命令

若要修正處於不相容網路狀態的資料庫，請依照下列指示執行：

1. 開啟 <https://console.aws.amazon.com/rds/>，並從導覽窗格選擇資料庫。
2. 選擇處於不相容網路狀態的資料庫執行個體，並記下連線與安全性索引標籤中的資料庫執行個體識別符、VPC ID 和子網路 ID。
3. 使用 AWS CLI 來執行命令start-db-instance令。指定 --db-instance-identifier 值。

Note

若在資料庫處於不相容模式時執行此命令，可能會造成短暫停機時間。

`start-db-instance` 命令無法解決 RDS for SQL Server 資料庫執行個體的此問題。

如果命令成功執行，您的資料庫狀態會變更為可用。

如果您的資料庫重新啟動，資料庫執行個體可能會執行在執行個體進入不相容網路狀態之前，於執行個體上執行的最後一項操作。這樣可能會讓執行個體再次進入不相容的網路狀態。

如果 `start-db-instance` 命令失敗，或執行個體再次進入不相容的網路狀態，請在 RDS 主控台中開啟資料庫，然後選取資料庫。導覽至日誌與事件區段。最近的事件區段會顯示要採取的進一步解決步驟。訊息分類如下：

- INTERNAL RESOURCE CHECK (內部資源檢查)：您的內部資源可能發生問題。
- DNS CHECK (DNS 檢查)：在 VPC 主控台中檢查 VPC 的 DNS 解析和主機名稱。
- ENI CHECK (ENI 檢查)：資料庫的彈性網路介面 (ENI) 可能不存在。
- GATEWAY CHECK (閘道檢查)：公開可用資料庫的網際網路閘道未連線至 VPC。
- IP CHECK (IP 檢查)：子網路中沒有空出的 IP 地址。
- SECURITY GROUP CHECK (安全群組檢查)：您的資料庫沒有相關聯的安全群組，或是安全群組無效。
- SUBNET CHECK (子網路檢查)：您的資料庫子網路群組中沒有有效的子網路，或子網路發生問題。
- VPC CHECK (VPC 檢查)：與您的資料庫相關聯的 VPC 無效。

執行 point-in-time 復原

最佳實務是擁有備份 (快照或邏輯)，以免您的資料庫進入不相容的網路狀態。請參閱[備份簡介](#)。如果您開啟了自動備份，請暫時停止對資料庫的任何寫入並執行 point-in-time 復原。

Note

在執行個體進入不相容的網路狀態後，可能無法存取資料庫執行個體來執行邏輯備份。

如果您未開啟自動備份，請建立新的資料庫執行個體。然後使用 [AWS Database Migration Service \(AWS DMS\)](#)，或使用備份與還原工具來遷移資料。

如果這樣無法解決問題，請聯絡以取得 AWS Support 進一步協助。

重新設定資料庫執行個體擁有者密碼

如果您遭到鎖定資料庫執行個體，則可以用主要使用者身分登入。然後，您可以重設其他系統管理使用者或角色的認證。如果您無法以主要使用者身分登入，AWS 帳戶擁有者可以重設主要使用者密碼。如需您可能需要重設哪些管理帳戶或角色的詳細資訊，請參閱[主要使用者帳戶權限](#)。

您可以使用 Amazon RDS 主控台、AWS CLI 命令 [modify-db-instance](#) 或使用 [修改資料庫執行個體 API 作業來變更資料庫執行個體密碼](#)。如需修改 資料庫執行個體的詳細資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Amazon RDS 資料庫執行個體停機或重新開機

重新啟動資料庫執行個體時，可能會發生資料庫執行個體當機。當資料庫執行個體進入防止受到存取的狀態，以及當資料庫重新啟動時，均可能發生當機。在您手動將資料庫執行個體重新開機時，就會重新開機。當您變更需要重新開機才會生效的資料庫執行個體設定時，也會發生重新開機。

當您變更需要重新開機才會生效的設定，或當您手動進行重新開機時，才會發生資料庫執行個體重新開機。如果您變更設定並要求該變更立即生效，即會立即發生重新開機。或者，它會在資料庫執行個體的維護時段期間發生。

發生下列其中一個動作時，資料庫執行個體會立即重新開機：

- 您將資料庫執行個體的備份保留期間從 0 變更為非零值，或從非零值變更為 0。然後將 Apply Immediately (立即套用) 設為 true。
- 您變更資料庫執行個體類別，並且將 Apply Immediately (立即套用) 設為 true。
- 您將儲存類型從 Magnetic (Standard) (磁帶 (標準)) 變更為 General Purpose (SSD) (一般用途 (SSD)) 或 Provisioned IOPS (SSD) (佈建 IOPS (SSD))，或從 Provisioned IOPS (SSD) (佈建 IOPS (SSD)) 或 General Purpose (SSD) (一般用途 (SSD)) 變更為 Magnetic (Standard) (磁帶 (標準))。

發生下列其中一個動作時，維護時段期間會發生資料庫執行個體重新開機：

- 您將資料庫執行個體的備份保留期間從 0 變更為非零值，或從非零值變更為 0，並且將 Apply Immediately (立即套用) 設為 false。
- 您變更資料庫執行個體類別，並且將 Apply Immediately (立即套用) 設為 false。

變更資料庫參數群組中的靜態參數時，在與參數群組相關聯的資料庫執行個體重新開機之後，變更才會生效。變更需要手動重新開機。在維護時段期間，不會自動重新啟動資料庫執行個體。

如需查看資料表，以了解資料庫執行個體動作與設定 Apply Immediately (立即套用) 值會產生的效果，請參閱[修改 Amazon RDS 資料庫執行個體](#)。

Amazon RDS 資料庫參數變更未生效

在某些情況下，您可能會變更資料庫參數群組中的參數，但不會看到變更生效。若是如此，您可能需要重新啟動與 DB 參數群組關聯的資料庫執行個體。當您變更動態參數時，變更會立即生效。當您變更靜態參數時，在您重新啟動與參數群組關聯的資料庫實體之前，變更將不會生效。

您可以使用 RDS 主控台重新啟動資料庫執行個體。或者，您可以明確呼叫 [RebootDBInstance](#) API 操作。如果資料庫執行個體處於多可用區部署中，即可重新啟動，而不需容錯移轉。若在變更靜態參數之後，要求重新啟動相關聯的資料庫執行個體，將有助於降低參數組態錯誤影響 API 呼叫的風險。其中一個例子是呼叫 [ModifyDBInstance](#) 變更資料庫執行個體類別。如需詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

Amazon RDS 資料庫執行個體用盡儲存體

如果您的資料庫執行個體用盡儲存空間，將不再可用。我們強烈建議您持續監控中發佈的 `FreeStorageSpace` 指標，CloudWatch 以確保資料庫執行個體擁有足夠的可用儲存空間。

如果您的資料庫執行個體用盡儲存體，其狀態將變更為 `storage-full`。例如，對已用完其儲存體之資料庫執行個體 `DescribeDBInstances` API 操作的呼叫有下列輸出：

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa
storage-full mydbinstance.c1la4j4jgyph.us-east-1.rds.amazonaws.com 3306
us-east-1b 3
SECGROUP default active
PARAMGRP default.mysql8.0 in-sync
```

若要從這個案例中復原，請使用 `ModifyDBInstance` API 作業或下列 AWS CLI 命令為執行個體新增更多儲存空間。

對於 Linux/macOS、或 Unix：

```
aws rds modify-db-instance \  
--db-instance-identifier mydbinstance \  
--storage-size 100
```

```
--allocated-storage 60 \  
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --allocated-storage 60 ^  
  --apply-immediately
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa  
storage-full mydbinstance.c1la4j4jgyph.us-east-1.rds.amazonaws.com 3306  
us-east-1b 3 60  
SECGROUP default active  
PARAMGRP default.mysql8.0 in-sync
```

現在，描述資料庫執行個體時，將會看到資料庫執行個體 `modifying` 狀態，這表示正在擴展儲存體。

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa  
modifying mydbinstance.c1la4j4jgyph.us-east-1.rds.amazonaws.com  
3306 us-east-1b 3 60  
SECGROUP default active  
PARAMGRP default.mysql8.0 in-sync
```

一旦儲存體擴展完成，您的資料庫執行個體狀態將變更為 `available`。

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 60 sa  
available mydbinstance.c1la4j4jgyph.us-east-1.rds.amazonaws.com 3306  
us-east-1b 3  
SECGROUP default active  
PARAMGRP default.mysql8.0 in-sync
```

您可以使用 `DescribeEvents` 操作，在您的儲存空間用盡時收到通知。例如，在此案例中，如果您於這些操作之後執行 `DescribeEvents` 呼叫，將看到下列輸出：


```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z mydbinstance Allocated storage has been exhausted db-  
instance  
2009-12-23T00:14:02.737Z mydbinstance Applying modification to allocated storage db-  
instance  
2009-12-23T00:31:54.764Z mydbinstance Finished applying modification to allocated  
storage
```

Amazon RDS 資料庫執行個體容量不足

在嘗試建立、啟動或修改資料庫執行個體時，會傳回 `InsufficientDBInstanceCapacity` 錯誤。在嘗試從資料庫快照還原資料庫執行個體時，也會傳回錯誤。傳回此錯誤時，常見原因是無法在要求的可用區域中使用指定的資料庫執行個體類別。您可以嘗試以下其中一項來解決此問題：

- 使用不同的資料庫執行個體類別來重試此請求。
- 使用不同的可用區域來重試此請求。
- 重試該請求，但不指定明確的可用區域。

如需對 Amazon EC2 執行個體容量問題進行故障診斷的資訊，請參閱《Amazon EC2 使用者指南》中的 [執行個體容量不足](#)。

如需修改資料庫執行個體的相關資訊，請參閱 [修改 Amazon RDS 資料庫執行個體](#)。

Amazon RDS 中的可用記憶體問題

可用記憶體是資料庫執行個體上可供資料庫引擎使用的隨機存取記憶體 (RAM) 總計。它是可用作業系統 (OS) 記憶體與可用緩衝區和頁面快取記憶體的總和。資料庫引擎使用主機上大多數的記憶體，但作業系統處理程序也使用一些 RAM。目前配置給資料庫引擎或作業系統處理程序目前所使用的記憶體不包括在可用記憶體中。當資料庫引擎記憶體不足時，資料庫執行個體會使用一般用於緩衝和快取的暫時空間。如前所述，此暫時空間包括在可用記憶體中。

您可以使用 Amazon 中的 `FreeableMemory` 指標 CloudWatch 來監視可用內存。如需詳細資訊，請參閱 [在 Amazon RDS 中監控指標的概觀](#)。

如果資料庫執行個體在可用記憶體方面持續不足或使用交換空間，請考慮縱向擴展為更大的資料庫執行個體類別。如需詳細資訊，請參閱 [資料庫執行個體類別](#)。

您也可以變更記憶體設定。例如，在 RDS for MySQL，您可以調整 `innodb_buffer_pool_size` 參數的大小。依預設，此參數設定為實體記憶體的 75%。如需更多 MySQL 的疑難排解秘訣，請參閱[如何疑難排解 Amazon RDS for MySQL 資料庫中可用記憶體不足的問題？](#)

MySQL 和 MariaDB 問題

您可以診斷並修正 MySQL 和 MariaDB 資料庫執行個體的問題。

主題

- [MySQL 及 MariaDB 連線上限](#)
- [診斷和解決記憶體限制的不相容參數狀態](#)
- [診斷和解決僅供讀取複本之間的延遲](#)
- [診斷和解決 MySQL 或 MariaDB 僅供讀取複寫失敗](#)
- [建立將二進位日誌啟用的觸發器需要 SUPER 權限](#)
- [診斷和解決point-in-time 還原失敗](#)
- [複寫已停止錯誤](#)
- [僅供讀取複本建立失敗或複寫中斷，出現嚴重錯誤 1236](#)

MySQL 及 MariaDB 連線上限

RDS for MySQL 或 MariaDB 資料庫執行個體允許的連線數上限，取決於資料庫執行個體類別可用的記憶體數量。資料庫執行個體可用的記憶體越多，可用的連線數就越多。如需資料庫執行個體類別的詳細資訊，請參閱[資料庫執行個體類別](#)。

依預設，資料庫執行個體的連線限制會設為資料庫執行個體類別的上限。您可以將並行連線數限制為任何值，最大為允許的連線數上限。使用資料庫執行個體中參數群組的 `max_connections` 參數。如需更多詳細資訊，請參閱[資料庫連線數上限](#) 及 [使用參數群組](#)。

您可以執行下列查詢，以擷取 MySQL 或 MariaDB 資料庫執行個體允許的連線數上限。

```
SELECT @@max_connections;
```

您可以執行下列查詢，以擷取 MySQL 或 MariaDB 資料庫執行個體的作用中連線數目。

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

診斷和解決記憶體限制的不相容參數狀態

當滿足下列條件時，MariaDB 或 MySQL 資料庫執行個體可以置於記憶體限制的不相容參數狀態中：

- 當資料庫執行個體狀態為可用時，執行個體會在一小時內至少重新啟動三次，或在一天內至少重新啟動五次。
- 嘗試重新啟動資料庫執行個體失敗，因為維護動作或監控程序無法重新啟動資料庫執行個體。
- 資料庫執行個體的潛在記憶體用量超過配置給其資料庫執行個體類別的記憶體 1.2 倍。

當資料庫執行個體在一小時內第三次重新啟動時，或在一天內第五次重新啟動時，它會執行記憶體用量檢查。此檢查會計算資料庫執行個體的潛在記憶體用量。計算所傳回的值是下列值的總和：

- 值 1 – 下列參數的總和：
 - `innodb_additional_mem_pool_size`
 - `innodb_buffer_pool_size`

您可以修改的值 `innodb_buffer_pool_size`。但是，該值並不總是與您輸入的內容匹配。發生這種不匹配的原因有幾個。首先，如果資料庫執行個體是微型資料庫執行個體，我們會覆寫預設值並將其設定為 256 MB。如需詳細資訊，請參閱 [覆蓋緩衝池大小](#)。

其次，我們確保在資料庫執行個體上為主機管理員、引擎、作業系統和核心保留了 500 MB 的記憶體。

最後，我們 `innodb_buffer_pool_size` 通過將其劃分為單位進行優化。主持人管理員會捨入到這些單位中最接近的倍數。單位的計算方式為 `innodb_buffer_pool_chunk_size` 乘 `innodb_buffer_pool_instances` 以。如需詳細資訊，請參閱 MySQL 文件中的 [< 設定 InnoDB 緩衝區集區大小 >](#)。

除非小 `innodb_buffer_pool_instances` 於 1 GB，否 `innodb_buffer_pool_size` 則預設值為 8。如果小 `innodb_buffer_pool_size` 於 1 GB，則預設值 `innodb_buffer_pool_instances` 為 1。的預設值 `innodb_buffer_pool_chunk_size` 為 128 MB。

- `innodb_log_buffer_size`
- `key_buffer_size`
- `query_cache_size` (僅限 MySQL 5.7 版)
- `tmp_table_size`

- 值 2 – `max_connections` 參數乘以下參數的總和：
 - `binlog_cache_size`
 - `join_buffer_size`
 - `read_buffer_size`
 - `read_rnd_buffer_size`
 - `sort_buffer_size`
 - `thread_stack`
- 值 3 – 如果已啟用 `performance_schema` 參數，請將 `max_connections` 參數乘以 429498。

如果停用 `performance_schema` 參數，則此值為零。

因此，計算返回的值如下：

Value 1 + Value 2 + Value 3

當此值超過配置給資料庫執行個體使用之資料庫執行個體類別的記憶體 1.2 倍時，資料庫執行個體會置於不相容參數的狀態。如需配置給資料庫執行個體類別之記憶體的相關資訊，請參閱 [的資料庫執行個體類別的硬體規格](#)。

計算將 `max_connections` 參數的值乘以數個參數的總和。如果 `max_connections` 參數設定為較大的值，則可能會導致檢查傳回資料庫執行個體潛在記憶體用量非常高的值。在這種情況下，請考慮降低 `max_connections` 參數的值。

若要解決此問題，請完成下列步驟：

1. 調整與資料庫執行個體相關聯的資料庫參數群組中的記憶體參數。這樣做，可讓潛在的記憶體用量低於配置給其資料庫執行個體類別的記憶體 1.2 倍。

如需有關設定參數的詳細資訊，請參閱 [修改資料庫參數群組中的參數](#)。

2. 重新啟動資料庫執行個體。

如需有關設定參數的詳細資訊，請參閱 [啟動之前已停止的 Amazon RDS 資料庫執行個體](#)。

診斷和解決僅供讀取複本之間的延遲

建立 MySQL 或 MariaDB 僅供讀取複本且僅供讀取複本可供使用之後，Amazon RDS 會先複寫從開始僅供讀取複本建立操作起始起，對來源資料庫執行個體進行的變更。在此階段期間，僅供讀取複本的複

寫延遲時間將大於 0。您可以 CloudWatch 透過檢視 Amazon RDS ReplicaLag 指標，在 Amazon 中監控此延遲時間。

ReplicaLag 指標會回報 MariaDB 的 Seconds_Behind_Master 欄位值，或是 MySQL SHOW REPLICA STATUS 命令。如需詳細資訊，請參閱 MySQL 文件中的 [SHOW REPLICA STATUS 陳述式](#)。

當 ReplicaLag 指標到達 0，複本即已跟上來源資料庫執行個體。如果 ReplicaLag 指標傳回 -1，複寫可能為非作用中。若要對複寫錯誤進行故障診斷，請參閱 [診斷和解決 MySQL 或 MariaDB 僅供讀取複寫失敗](#)。如果 ReplicaLag 的值為 -1，也可能表示系統無法判斷 Seconds_Behind_Master 值，或該值為 NULL。

Note

MariaDB 和 MySQL 以前的版本使用 SHOW SLAVE STATUS 而不是 SHOW REPLICA STATUS。如果您使用的 MariaDB 是 10.5 之前的版本或使用的 MySQL 是 8.0.23 之前的版本，請使用 SHOW SLAVE STATUS。

在網路中斷期間，或在維護時段套用修補程式時，ReplicaLag 指標即會傳回 -1。在這種情況下，請等候系統還原網路連線，或是等待維護時段結束，隨後再重新檢查 ReplicaLag 指標。

MySQL 和 MariaDB 讀取複製技術是非同步的。因此您可以預期來源資料庫執行個體的 BinLogDiskUsage 指標與僅供讀取複本上的 ReplicaLag 指標偶爾會增加。例如，請考慮一個情況，其中來源資料庫執行個體可同時出現大量寫入操作。同時，會使用單一輸入/輸出執行緒，對僅供讀取複本的寫入操作進行序列化。這種情況可能會導致來源執行個體和僅供讀取複本之間發生延遲。

如需僅供讀取複本和 MySQL 的詳細資訊，請參閱 MySQL 文件中的 [複寫實作詳細資訊](#)。如需僅供讀取複本和 MariaDB 的詳細資訊，請參閱 MariaDB 文件中的 [複寫概觀](#)。

您可以利用執行下列動作，減少對來源資料庫執行個體的更新與對僅供讀取複本的後續更新之間的延遲：

- 將僅供讀取複本的資料庫執行個體類別設定為與來源資料庫執行個體具有相等的儲存體大小。
- 確保來源資料庫執行個體和僅供讀取複本所使用的資料庫參數群組中的參數設定相容。如需詳細資訊和範例，請參閱下一節中 max_allowed_packet 參數的討論。
- 停用查詢快取。針對經常修改的資料表，因為快取已遭鎖定並且經常重新整理，使用查詢快取可能會增加複本延遲。若是這種情況，如果停用查詢快取，您可能會發現複本延遲較少。您可以在資料庫執

行個體的資料庫參數群組中將 `query_cache_type` parameter 設定為 0，以停用查詢快取。如需查詢快取的詳細資訊，請參閱[查詢快取組態](#)。

- 為 MySQL 或 MariaDB 的 InnoDB 備妥僅供讀取複本上的緩衝區集區。如果您有經常更新的一組小型資料表，並且使用 InnoDB 或 XtraDB 資料表結構描述。在這種情況下，傾印僅供讀取複本上的那些資料表。這麼做會造成資料庫引擎掃描來自磁碟之那些資料表的資料列，然後將它們快取在緩衝集區中。如此可減少複本延遲。下列顯示一個範例。

對於LinuxmacOS、或Unix：

```
PROMPT> mysqldump \  
-h <endpoint> \  
--port=<port> \  
-u=<username> \  
-p <password> \  
database_name table1 table2 > /dev/null
```

在 Windows 中：

```
PROMPT> mysqldump ^  
-h <endpoint> ^  
--port=<port> ^  
-u=<username> ^  
-p <password> ^  
database_name table1 table2 > /dev/null
```

診斷和解決 MySQL 或 MariaDB 僅供讀取複寫失敗

Amazon RDS 會監控僅供讀取複本的複寫狀態。如果複寫因任何原因停止，RDS 會將僅供讀取複本執行個體的 Replication State (複寫狀態) 欄位更新為 `ERROR`。您可以透過檢視 Replication Error (複寫錯誤) 欄位，檢閱 MySQL 或 MariaDB 引擎擲出之相關聯錯誤的詳細資訊。也會產生指出僅供讀取複本之狀態的事件，包括 [RDS-EVENT-0045](#)、[RDS-EVENT-0046](#) 和 [RDS-EVENT-0057](#)。如需事件和訂閱事件的詳細資訊，請參閱[使用 Amazon RDS 事件通知](#)。如果傳回 MySQL 錯誤訊息，請在 [MySQL 錯誤訊息文件](#) 中檢閱錯誤。如果傳回了 MariaDB 錯誤訊息，請在 [MariaDB 錯誤訊息文件](#) 中檢閱錯誤。

可能造成複寫錯誤的常見情況包括下列：

- 僅供讀取複本之 `max_allowed_packet` 參數的值小於來源資料庫執行個體之 `max_allowed_packet` 參數的值。

`max_allowed_packet` 參數是您可以在資料庫參數群組中設定的自訂參數。`max_allowed_packet` 參數可用來指定可在資料庫上執行的資料處理語言 (DML) 大小上限。在某些情況下，來源資料庫執行個體的 `max_allowed_packet` 值可能大於僅供讀取複本的 `max_allowed_packet` 值。若是如此，複寫程序會擲回錯誤並停止複寫。最常見的錯誤為 `packet bigger than 'max_allowed_packet' bytes`。您可以透過讓來源和僅供讀取複本使用具有相同 `max_allowed_packet` 參數值的資料庫參數群組，藉此修正錯誤。

- 寫入僅供讀取複本上的資料表。如果您在僅供讀取複本上建立索引，您需要將 `read_only` 參數設為 0 才能建立索引。如果您要寫入僅供讀取複本上的資料表，可能中斷複寫。
- 使用非交易儲存引擎 (例如 MyISAM)。僅供讀取複本需要交易儲存引擎。複寫只支持以下儲存引擎：MySQL 或 MariaDB 的 InnoDB。

您可以使用下列命令，將 MyISAM 資料表轉換為 InnoDB：

```
alter table <schema>.<table_name> engine=innodb;
```

- 使用不安全的非確定性查詢 (例如 `SYSDATE()`)。如需詳細資訊，請參閱 MySQL 文件中的 [二進位日誌記錄中安全和不安全陳述式的判定](#)。

下列步驟有助於解決您的複寫錯誤：

- 如果遇到邏輯錯誤，而您可以安全地略過錯誤，請遵循 [略過目前複寫錯誤](#) 中所述的步驟。您的 MySQL 或 MariaDB 資料庫執行個體必須執行包含 `mysql_rds_skip_repl_error` 程序的版本。如需更多詳細資訊，請參閱 [mysql.rds_skip_repl_error](#)。
- 如果遇到二進位記錄檔位置問題，您可以使用 `mysql_rds_next_master_log` 命令來變更從屬重播位置。您的 MySQL 或 MariaDB 資料庫執行個體必須執行支援 `mysql_rds_next_master_log` 命令的版本，才能變更從屬重播位置。如需版本資訊，請參閱 [mysql.rds_next_master_log](#)。
- 由於高 DML 負載，您可能遇到暫時效能問題。若是如此，您可以在僅供讀取複本的資料庫參數群組中將 `innodb_flush_log_at_trx_commit` 參數設定為 2。這麼做有助於僅供讀取複本跟上，不過會暫時減少不可分割性、一致性、隔離和耐用性 (ACID)。
- 您可以刪除僅供讀取複本，並使用相同的資料庫執行個體識別符來建立執行個體。如果您執行這項操作，端點仍會與與舊的僅供讀取複本相同。

如果複寫錯誤已修復，Replication State (複寫狀態) 會變更為 `replicating` (複寫中)。如需更多詳細資訊，請參閱 [對 MySQL 僅供讀取複本問題進行故障診斷](#)。

建立將二進位日誌啟用的觸發器需要 SUPER 權限

嘗試在 RDS for MySQL 或 RDS for MariaDB 資料庫執行個體中建立觸發器時，您可能會收到下列錯誤。

```
"You do not have the SUPER privilege and binary logging is enabled"
```

若要在二進位日誌記錄啟用時使用觸發器，需要 SUPER 權限，該權限受限於 RDS for MySQL 和 RDS for MariaDB 資料庫執行個體。當二進位日誌已啟用但沒有 SUPER 權限時，您可以將 `log_bin_trust_function_creators` 參數設定為 `true` 來建立觸發器。若要將 `log_bin_trust_function_creators` 設定為 `true`，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

您可以建立新的資料庫參數群組，以便您可以在二進位日誌啟用的情況下，在 RDS for MySQL 或 RDS for MariaDB 資料庫執行個體中建立觸發條件。若要這樣做，請使用下列 CLI 命令。若要修改現有的參數群組，請從步驟 2 開始。

使用 CLI 建立新的參數群組以允許啟用二進位日誌的觸發器

1. 建立新的參數群組。

對於LinuxmacOS、或Unix：

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name allow-triggers \  
  --db-parameter-group-family mysql8.0 \  
  --description "parameter group allowing triggers"
```

在 Windows 中：

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name allow-triggers ^  
  --db-parameter-group-family mysql8.0 ^  
  --description "parameter group allowing triggers"
```

2. 修改資料庫參數群組以允許觸發器。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name allow-triggers \  
  --db-parameter-group-family mysql8.0 \  
  --description "parameter group allowing triggers"
```



```
--db-parameter-group-name allow-triggers \  
--parameters "ParameterName=log_bin_trust_function_creators,  
ParameterValue=true, ApplyMethod=pending-reboot"
```

在 Windows 中：

```
aws rds modify-db-parameter-group ^  
--db-parameter-group-name allow-triggers ^  
--parameters "ParameterName=log_bin_trust_function_creators,  
ParameterValue=true, ApplyMethod=pending-reboot"
```

3. 修改資料庫執行個體以使用新資料庫參數群組。

對於LinuxmacOS、或Unix：

```
aws rds modify-db-instance \  
--db-instance-identifier mydbinstance \  
--db-parameter-group-name allow-triggers \  
--apply-immediately
```

在 Windows 中：

```
aws rds modify-db-instance ^  
--db-instance-identifier mydbinstance ^  
--db-parameter-group-name allow-triggers ^  
--apply-immediately
```

4. 若要讓變生效，請手動將資料庫執行個體重新開機。

```
aws rds reboot-db-instance --db-instance-identifier mydbinstance
```

診斷和解決point-in-time 還原失敗

還原包含暫存資料表的資料庫執行個體

嘗試對 MySQL 或 MariaDB 資料庫執行個體進行 point-in-time 還原 (PITR) 時，您可能會遇到以下錯誤。

```
Database instance could not be restored because there has been incompatible database  
activity for restore
```

```
functionality. Common examples of incompatible activity include using temporary tables,
in-memory tables,
or using MyISAM tables. In this case, use of Temporary table was detected.
```

PITR 仰賴於來自 MySQL 或 MariaDB 的備份快照和二進位記錄檔來將資料庫執行個體還原到特定時間。暫存資料表資訊在二進位記錄檔中可能不可靠，並且可能造成 PITR 失敗。如果在 MySQL 或 MariaDB 資料庫執行個體中使用暫存資料表，您可以降低 PITR 失敗的可能性。若要這樣做，請執行更頻繁的備份。暫存資料表的建立和下一個備份快照之間的時間，最可能發生 PITR 失敗。

還原包括記憶體內資料表的資料庫執行個體

還原具有記憶體內資料表的資料庫時，您可能會遇到問題。會在重新啟動期間清除記憶體內資料表。因此，您的記憶體內資料表在重新開機之後可能會是空白的。建議您使用記憶體內資料表時，思考您的解決方案以便在重新啟動時處理空白資料表。如果您使用具有複製資料庫執行個體的記憶體內資料表，則可能需要在重新啟動後，重新建立僅供讀取複本。如果將僅供讀取複本重新開機，而且無法從空白的記憶體內資料表還原資料，則可能必須進行此操作。

如需備份和 PITR 的詳細資訊，請參閱[備份簡介](#)和[將資料庫執行個體還原至指定的時間](#)。

複寫已停止錯誤

當您呼叫 `mysql.rds_skip_repl_error` 命令時，可能會收到錯誤訊息，指出複寫已關閉或停用。

因為複寫已停止且無法重新啟動，因此出現此錯誤訊息。

如果您需要略過大量錯誤，複寫延遲可能增加至超出二進位記錄檔的預設保留期間。在此情況下，由於在清除二進位記錄檔之前已在複本上重播該檔案，您可能會遇到嚴重錯誤。此清除動作會導致複寫停止，而您將無法再呼叫 `mysql.rds_skip_repl_error` 命令來略過複寫錯誤。

透過增加二進位記錄檔在複寫主控端上保留的小時數，即可以減輕此問題。在延長二進位記錄檔保留時間之後，您可以重新啟動複寫，並視需要呼叫 `mysql.rds_skip_repl_error` 命令。

若要設定 binlog 保留時間，請使用 [mysql.rds_set_configuration](#) 程序。並指定 'binlog retention hours' 組態參數，以及資料庫叢集上保留二進位記錄檔的時數 (最多 720 小時 (30 天))。下列範例會將 binlog 檔案的保留期間設定為 48 小時。

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

僅供讀取複本建立失敗或複寫中斷，出現嚴重錯誤 1236

變更 MySQL 或 MariaDB 資料庫執行個體的預設參數值之後，您可能會遇到下列其中一個問題：

- 您無法建立資料庫執行個體的僅供讀取複本。
- 複寫失敗，出現 fatal error 1236。

MySQL 及 MariaDB DB 執行個的一些預設參數，有助於確定資料庫符合 ACID 規定，而且僅供讀取複本不會當機。之所以能實現這個結果，是在認可之前，確定均透過將交易寫入二進位日誌，而讓每項認可獲得完全同步。將這些參數從其預設值進行變更以改善效能，在尚未將交易寫入二進位記錄時，可能造成複寫失敗。

若要解決此問題，請設定下列參數值：

- `sync_binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

無法將備份保留期間設定為 0

需要將備份保留期間設定為 0 有數個原因。例如，您可以透過將保留期間設定為 0 來立即停用自動備份。

在某些情況下，您可能會將值設定為 0，並收到一則訊息，表示保留期間必須介於 1 到 35 之間。在這些情況下，請檢查以確定您尚未設定執行個體的僅供讀取複本。僅供讀取複本需要用於管理僅供讀取複本日誌的備份，因此，您不可以設定 0 的保留期間。

Amazon RDS API 參考

除了 AWS Management Console 和 AWS Command Line Interface (AWS CLI) 之外，Amazon RDS 還提供一個 API。您可以使用 API，將管理資料庫執行個體以及 Amazon RDS 中其他物件的任務自動化。

- 如需依字母排序的 API 操作清單，請參閱[動作](#)。
- 如需依字母排序的資料類型清單，請參閱[資料類型](#)。
- 如需常用查詢參數的清單，請參閱[常用參數](#)。
- 如需錯誤碼的說明，請參閱[常見錯誤](#)。

如需 AWS CLI 的詳細資訊，請參閱 [Amazon RDS 的 AWS Command Line Interface 參考](#)。

主題

- [使用查詢 API](#)
- [對 Amazon RDS 上的應用程式進行故障診斷](#)

使用查詢 API

下節討論與查詢 API 搭配使用的參數和請求身分驗證。

如需查詢 API 運作方式的一般資訊，請參閱《Amazon EC2 API Reference》中的[查詢請求](#)。

查詢參數

HTTP 查詢型請求是使用 HTTP 動詞 GET 或 POST，以及名為 Action 之查詢參數的 HTTP 請求。

每一個查詢請求都須包括一些常用參數，來處理動作的身分驗證和選取。

部分操作有數個參數清單。這些清單是使用 `param.n` 表示法來指定的。`n` 的值是從 1 開始的整數。

如需 Amazon RDS 區域和端點的相關資訊，請參閱 Amazon Web Services 一般參考中「區域與端點」一節的 [Amazon Relational Database Service \(RDS\)](#)。

查詢請求身分驗證

您只能透過 HTTPS 傳送查詢請求，而且必須在每一個查詢請求中包括一個簽章。您必須使用 AWS 簽章第 4 版或簽章第 2 版。如需詳細資訊，請參閱[簽章第 4 版簽署程序](#)和[簽章第 2 版簽署程序](#)。

對 Amazon RDS 上的應用程式進行故障診斷

Amazon RDS 會提供特定和描述性錯誤，以協助您在與 Amazon RDS API 互動時進行故障診斷。

主題

- [擷取錯誤](#)
- [對秘訣進行故障診斷](#)

如需 Amazon RDS 資料庫執行個體疑難排解的資訊，請參閱 [Amazon RDS 故障診斷](#)。

擷取錯誤

通常，您想要應用程式在您花費任何時間處理結果之前，先檢查請求是否已產生錯誤。若要了解系統是否發生錯誤，最簡單的方式即為在 Amazon RDS API 的回應中，尋找 Error 節點。

XPath 語法提供簡單的方式，來搜尋 Error 節點是否存在。它還提供了一種相對簡單的方式，來擷取錯誤碼和訊息。下列程式碼片段使用 Perl 和 XML::XPath 模組，來判斷請求期間是否發生錯誤。如果發生錯誤，程式碼會列印回應中的第一個錯誤碼和訊息。

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
 $xp->findvalue("//Error[1]/Code"), "\n", " ",
 $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

對秘訣進行故障診斷

我們建議以下列程序，來診斷並解決 Amazon RDS API 發生的問題。

- 檢查 <http://status.aws.amazon.com>，驗證 Amazon RDS 在目標 AWS 區域中是否能正常運作。
- 檢查請求的結構。

在 Amazon RDS API 參考中，每項 Amazon RDS 操作都會有一個參考頁面。再次檢查您是否正確使用參數。如需有關可能出錯的概念，請查看範例請求或使用者案例，來查看那些範例是否執行類似操作。

- 檢查 AWS re:Post。

Amazon RDS 具有開發社群，您可在其中搜尋其他人在過程中所遇到問題的解決方案。如要檢視主題，請移至 [AWS re:Post](#)。

文件歷史記錄

目前的 API 版本：2014-10-31

下表會說明 2018 年 5 月後，Amazon RDS 使用者指南每個版本的重要變更。如需有關此文件更新的通知，您可以訂閱 RSS 摘要。

Note

您可以在[資料庫的最新資訊](#)頁面上篩選新的 Amazon RDS 功能。對於 Products (產品)，請選擇 Amazon RDS。然後使用關鍵字搜尋，例如 **RDS Proxy** 或 **Oracle 2023**。

變更	描述	日期
Amazon RDS for Oracle 支援預先設定的 r6i 記憶體優化執行個體類別	db.r6i Oracle 資料庫執行個體類別針對每個 vCPU 需要額外記憶體、儲存和 I/O 的工作負載進行了最佳化。例如，已開啟多執行緒，並且提供的記憶體是資料庫 .r6i.8xlarge 的 4 倍。如需更多詳細資訊，請參閱 RDS for Oracle 執行個體類別 。	2024年6月21日
Amazon RDS 擴展 Support	RDS 延伸 Support 版本現已提供適用於 MySQL 版的 RDS。如需詳細資訊，請參閱 適用於 MySQL 的 Amazon RDS 擴充 Support 版本 。	2024年6月20日
Amazon RDS 支持	您現在可以建立執行 MySQL 8.0.37 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2024年6月18日

[Amazon RDS 支持 MariaDB 10.11.8 , 10.6.18 , 10.5.25 和 10.4.34](#)

您現在可以建立執行 MariaDB 版本 10.11.8、10.5.25 和 10.4.34 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2024年6月14日

[Amazon RDS 即將終止對數據庫 .m4、數據庫 .r4 和數據庫 .t2 數據庫執行個體類別的支持](#)

對於適用於 MariaDB 資料庫的 RDS、RDS 版和適用於 PostgreSQL MySQL 資料庫引擎的 RDS，您無法再建立使用 db.m4、db.r4 和 db.t2 執行個體類別的資料庫執行個體。RDS 會自動將使用這些類別的現有資料庫執行個體升級到新一代。如需更多詳細資訊，請參閱 [資料庫執行個體類型](#)。

2024年6月4日

[異地同步備份資料庫叢集提供額外提供 AWS 區域](#)

您可以建立更 AWS 區域多異地同步備份資料庫叢集。如需顯示所有支援區域的表格，請參閱 [Amazon RDS 中異地同步備份資料庫叢集的支援區域和資料庫引擎](#)。

2024年5月29日

[AWS Python 驅動程序一般可用](#)

Amazon Web Services (AWS) Python 驅動程序被設計為一個先進的 Python 包裝。此包裝器是補充並擴展了開源 Psycopg 驅動程序的功能。如需詳細資訊，請參閱 [使用 AWS 驅動程式連線至資料庫執行個體](#)。

2024年5月23日

[RDS 代理伺服器可在更多地區使用](#)

RDS Proxy 現已在亞太區域 (海德拉巴)、亞太區域 (墨爾本)、中東 (阿拉伯聯合大公國)、以色列 (特拉維夫)、加拿大西部 (卡加利) 和歐洲 (蘇黎世) 區域提供。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2024年5月21 日

[通過 Db2 許可證 AWS Marketplace](#)

透過 Db2 授權 AWS Marketplace，您現在可以按小時費率支付訂閱適用於 Db2 的 Amazon RDS 授權的 Db2 授權。如需詳細資訊，請參閱[Amazon RDS 適用於 Db2 的授權選項](#)。

2024年5月21 日

[Amazon RDS 支援更精細的 Performance Insights 存取](#)

您現在可以在「Performance Insights」中允許或拒絕存取個別維度。此細微存取可用於 GetResourceMetrics、DescribeDimensionKeys、和 GetDimensionKeyDetails 動作。如需詳細資訊，請參閱[授予 Performance Insights 的細微存取權](#)。

2024年5月21 日

[適用於 MySQL 的 Amazon 擴展 Support 版本](#)

您可以檢視適用於 MySQL 版本的 RDS 延伸 Support 的所有版本。如需詳細資訊，請參閱[適用於 MySQL 的 Amazon RDS 擴充 Support 版本](#)。

2024年5月16日

Amazon RDS 在資料庫預覽環境中支援 MySQL 8.3	MySQL 8.3 現在可在美國東部 (俄亥俄) 的「資料庫預覽」環境中使用 AWS 區域。如需詳細資訊，請參閱 資料庫預覽環境中的 MySQL 8.3 版 。	2024 年 4 月 30 日
Amazon 適用於 Db2 的 RDS 支持時區	RDS for Db2 現在支援為 Db2 資料庫執行個體的新 RDS 設定本地時區。如需詳細資訊，請參閱 適用於 Db2 資料庫執行個體的 Amazon RDS 當地時區 。	2024年4月25日
更新至 IAM 服務連結角色許可	此AmazonRDSCustomServiceRolePolicy 原則現在會授與其他權限，以將服務角色做為執行個體設定檔與 RDS Custom 執行個體建立關聯。如需詳細資訊，請參閱 AWS 受管政策的 Amazon RDS 更新 。	2024年4月19日
Amazon RDS for Oracle 文的所有支持 Oracle 數據保護切換 AWS 區域	您現在可以在所有支援的區域中使用「Oracle 資料保全」轉換。如需詳細資訊，請參閱 Oracle 資料保全轉換概要 。	2024年4月16日
RDS 自定義為甲骨文支持甲骨文標準版 2	您現在可以在甲骨文資料庫 12c 版本 1 (12.1)、12 c 版本二 (12.2)、18c 和 19c 上使用標準版 2 建立資料庫執行個體。您可以創建 CDB 和非 CDB。如需詳細資訊，請參閱 適用於 Oracle 的 RDS 自訂版本和授權支援 。	2024年4月11日

[Amazon RDS for Oracle 文支持甲骨文 APEX 版本 23.2.v1](#)

您可以將 APEX 23.2.v1 與甲骨文資料庫 19c 及更高版本搭配使用。如需詳細資訊，請參閱 [Oracle Application Express](#)。

2024年4月11日

[RDS 自訂服務連結角色權限的更新](#)

AmazonRDSCustomServiceRolePolicy 現在授予其他許可，以允許 SQL Server 的 RDS 自訂取得 EC2 執行個體類型資訊和修改資料庫主機執行個體類型。如需詳細資訊，請參閱 [AWS 受管理策略的更新](#)。

2024年4月8日

[Amazon RDS 自訂適用於甲骨文支援 db.x2iezn 資料庫執行個體類別](#)

您現在可以針對 Oracle 資料庫執行個體的 RDS 自訂使用 db.x2iezn 執行個體類別。如需詳細資訊，請參閱 [RDS Custom for Oracle 的資料庫執行個體類別支援](#)。

2024年3月26日

[Amazon RDS 支援異地同步備份資料庫叢集的 db.c6gd 執行個體類別](#)

您現在可以將 db.c6gd 執行個體類別用於異地同步備份資料庫叢集部署。詳情請參閱 [異地同步備份資料庫叢集的執行個體類別可用性](#)。

2024年3月21日

[Amazon RDS 延長支援](#)

現在建立或還原適用於 MySQL 的 RDS 5.7 或 RDS 版資料庫會自動將該資料庫註冊到 Amazon RDS 延伸 Support 中，以便您現有的應用程式繼續正常運作。您可以選擇退出 RDS 延伸 Support，以避免在資料庫引擎的 RDS 標準支援結束日期之後收取費用。如需詳細資訊，請參閱[使用 Amazon RDS 延長支援](#)。

2024年3月21日

[適用於 Db2 的 RDS 與 AWS License Manager](#)

適用於 Db2 的 RDS 現已與 AWS License Manager 整合。如果您使用「自攜授權」模式，則 AWS License Manager 整合有助於監控組織內的 Db2 授權使用情況。如需詳細資訊，請參閱[與整合 AWS License Manager](#)。

2024年3月20日

[異地同步備份資料庫叢集的 CA 憑證輪替](#)

您現在可以輪替異地同步備份資料庫叢集的 CA 憑證。請考慮使用其中一個新的 CA 憑證 rds-ca-rsa 2048-g1、rds-ca-rsa 4096-g1 或 RDS-鈣-ECC 384-g1。如需詳細資訊，請參閱[旋轉您的 SSL/TLS 憑證](#)。

2024年3月6日

[Amazon RDS 支持 io2 塊快速存儲](#)

您現在可以建立使用 io2 區塊快速儲存區類型的 RDS 資料庫執行個體。如需詳細資訊，請參閱[io2 區塊快速儲存裝置](#)。

2024年3月6日

[適用於 SQL 伺服器的 RDS 自訂支援資料庫 .r5b 和資料庫 .x2iedn 資料庫執行個體類別](#)

您現在可以針對 SQL 伺服器資料庫執行個體的 RDS 自訂使用 db.r5b 和 db.x2iedn 執行個體類別。如需詳細資訊，請參閱[適用於 SQL Server 的 RDS 自訂資料庫執行個體類別支援](#)。

2024年3月4日

[中東 \(阿拉伯聯合大公國\) 區域提供適用於 Oracle 的 RDS 自訂服務](#)

您可以在中東 (阿拉伯聯合大公國) 區域為 Oracle 資料庫執行個體建立 RDS 自訂。如需顯示所有支援的表格 AWS 區域，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2024年3月4日

[新的 AWS 受管理策略](#)

Amazon RDS 新增了名為的新受管政策，AmazonRDS Custom InstanceProfileRolePolicy 允許 RDS Custom 透過 EC2 執行個體設定檔執行自動化動作和資料庫管理任務。如需詳細資訊，請參閱[AWS 受管政策的 Amazon RDS 更新](#)。

2024年2月27日

[Amazon RDS 支持 MariaDB 10.11.7、10.6.17、10.5.24 和](#)

您現在可以建立執行 MariaDB 10.11.7 版、10.5.24 和 10.4.33 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2024年2月26日

[Amazon RDS 異地同步備份資料庫叢集支援 Amazon EBS gp3 儲存磁碟區](#)

異地同步備份資料庫叢集現在支援以 SSD 為基礎的 Gp3 EBS 磁碟區。如需詳細資訊，請參閱[gp3 儲存體](#)。

2024年2月26日

[Amazon RDS AWS Secrets Manager 在以色列 \(特拉維夫 \) 區域的支持](#)

Amazon RDS 支持以色列 (特拉維夫) 區域的 Secrets Manager。如需詳細資訊，請參閱[使用 Amazon RDS 和 AWS Secrets Manager 進行密碼管理](#)。

2024年2月21 日

[適用於 Db2 的 Amazon RDS 支援稽核記錄](#)

RDS for Db2 現在支援資料庫層級稽核記錄。當您為 RDS for Db2 資料庫啟用稽核記錄時，Amazon RDS 會記錄資料庫活動並將稽核日誌存放在 Amazon S3 中。如需詳細資訊，請參閱[Db2 稽核記錄](#)。

2024年2月15日

[Amazon RDS 延長支援](#)

現在，當資料庫執行個體和異地同步備份資料 MySQL 叢集中的 RDS 和 RDS 主要引擎版本達到 RDS 標準支援結束日期時，Amazon RDS 現在會自動啟用 Amazon RDS 延伸支援。如需詳細資訊，請參閱[使用 Amazon RDS 延長支援](#)。

2024年2月15日

[Amazon RDS 支持](#)

您現在可以建立執行 MySQL 8.0.36 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2024年2月12日

[Amazon RDS 支援適用於 Db2 RDS 的 EBCDIC 定序](#)

您現在可以建立 Db2 資料庫，這些資料庫使用 EBCDIC 定序順序來排序資料庫中的內容。如需詳細資訊，請參閱 Amazon RDS 上 [Db2 資料庫的 EBCDIC 定序](#)。

2024年1月29日

更新為預設 CA 憑證	預設 CA 憑證設定為 <code>rds-ca-rsa2048-g1</code> 。如需詳細資訊，請參閱 使用 SSL/TLS 加密資料庫執行個體的連線 。	2024年1月26日
Amazon RDS for PostgreSQL 支持兩個新的包裝箱，用於 PL/ 鐵鏽，響應器和大號	您可以在亞馬遜 RDS 中使用兩個新的箱子。如需詳細資訊，請參閱 使用 PL/Rust 的板條箱 。	2024年1月24日
Amazon RDS for PostgreSQL TLS 版本 1.3	您可以在 RDS 中使用傳輸層安全性 (TLS) 1.3 版。如需詳細資訊，請參閱 將 SSL 用於 PostgreSQL 資料庫執行個體 。	2024年1月24日
RDS 自訂 SQL 伺服器支援 Microsoft SQL 伺服器	您現在可以為使用 SQL 伺服器 2022 的 SQL 伺服器資料庫執行個體建立 RDS 自訂。如需詳細資訊，請參閱 使用適用於 SQL 伺服器的 RDS 自訂 。	2024年1月22日
更新為 AWS 受管理策略權限	AWSServiceRoleForRDS 服務連結角色具有新 AmazonRDSServiceRolePolicy 的陳述式 ID。如需詳細資訊，請參閱 AWS 受管理政策的 Amazon RDS 更新 。	2024 年 1 月 19 日
適用於甲骨文的 RDS 自定義支持歐洲 (巴黎) 區域	您可以在歐洲 (巴黎) 區域為 Oracle 資料庫執行個體建立 RDS 自訂。如需詳細資訊，請參閱 適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎 。	2024年1月18日

[適用於 MySQL 的 Amazon RDS for MySQL 支援多來源複寫](#)

您現在可以在適用於 MySQL 資料庫執行個體的 RDS 上使用多來源複寫。如需詳細資訊，請參閱[在適用於 MySQL 的 RDS 上設定多來源複寫](#)。

2024年1月16日

[Amazon RDS 在資料庫預覽環境中支援 MySQL 8.2](#)

MySQL 8.2 現在可在美國東部 (俄亥俄) 的「資料庫預覽」環境中使用 AWS 區域。如需詳細資訊，請參閱[資料庫預覽環境中的 MySQL 8.2 版](#)。

2024年1月11日

[歐洲 \(西班牙\) 區域提供 RDS 代理伺服器](#)

歐洲 (西班牙) 區域現已推出 RDS 代理伺服器。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2024 年 1 月 8 日

[Amazon RDS 在加拿大西部 \(卡加利\) 地區提供](#)

Amazon RDS 現已在加拿大西部 (卡加利) 區域推出。如需更多詳細資訊，請參閱[區域和可用區域](#)。

2023 年 12 月 20 日

[Amazon 適用於 Db2 的 RDS 支援 5,000 個本機使用者](#)

您現在最多可以將 5,000 個本機使用者新增至授權清單。如需詳細資訊，請參閱[使用者](#)。

2023 年 12 月 20 日

[Amazon RDS 支援檢視和回應建議](#)

Amazon RDS 建議現在包含以閾值為基礎的主動式和以機器學習為基礎的反應式建議 (適用於 PostgreSQL)。如需詳細資訊，請參閱[檢視和回應 Amazon RDS 建議](#)。

2023 年 12 月 19 日

[Amazon RDS 支持 MariaDB 10.11.6 , 10.6.16 , 10.5.23 和](#)

您現在可以建立執行 MariaDB 版本 10.11.6、10.5.23 和 10.4.32 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2023 年 12 月 12 日

[Amazon RDS 推出了與 Amazon Redshift 進行零 ETL 集成 \(預覽\)](#)

零 ETL 整合提供全受管的解決方案，可在將交易資料寫入至 MySQL 資料庫執行個體的幾秒鐘內，在 Amazon Redshift 中使用交易資料。如需詳細資訊，請參閱 [使用 Amazon RDS 零 ETL 整合與 Amazon Redshift \(預覽\)](#)。

2023 年 11 月 28 日

[Amazon RDS 支持 IBM Db2 數據庫引擎](#)

您現在可以在 Amazon RDS 中運行 IBM Db2 數據庫引擎。如需詳細資訊，請參閱 [適用於 Db2 的 Amazon RDS](#)。

2023 年 11 月 27 日

[PostgreSQL 援主要版本升級至](#)

有了 RDS 版，您現在可以將資料庫引擎升級至主要版本 16.1，並將次要版本升級至 15.5、14.10、13.13、12.17 和 11.22。如需詳細資訊，請參閱 [升級適用於 Amazon RDS 的 PostgreSQL 資料庫引擎](#)。

2023 年 11 月 17 日

[RDS 自訂適用於甲骨文支援選項群組](#)

您可以建立或修改選項群組，並將其與 Oracle 資料庫執行個體的 RDS 自訂建立關聯。現在支援該 Timezone 選項。如需詳細資訊，請參閱 [在適用於 Oracle 的 RDS 自訂中使用選項群組](#)。

2023 年 11 月 17 日

[適用於 MySQL 的 Amazon RDS for MySQL 支援群組複寫外掛程式](#)

您現在可以使用 MySQL 社群開發和維護的群組複寫外掛程式，使用 RDS for MySQL 8.0.35 版或更高版本的資料庫執行個體來設定主動-主動叢集。如需詳細資訊，請參閱[設定適用於 MySQL 的 RDS 的主動-主動式叢集](#)。

2023 年 11 月 17 日

[Amazon RDS 代 PostgreSQL RDS](#)

您現在可以使用 RDS 代理伺服器建立代理伺服器，適用於 PostgreSQL 16.1 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS Proxy](#)。

2023 年 11 月 17 日

[RDS 自訂 SQL 伺服器支援 Microsoft SQL 伺服器 2019 年開發人員版](#)

您可以為使用 SQL 伺服器 2019 年開發人員版本的 SQL 伺服器資料庫執行個體建立 RDS 自訂。如需詳細資訊，請參閱[透過 RDS Custom for SQL Server 使用自有媒體](#)。

2023 年 11 月 16 日

[異地同步備份資料庫叢集的次要版本升級，停機時間](#)

當您執行異地同步備份資料庫叢集的次要版本升級時，Amazon RDS 現在會在寫入器執行個體之前升級讀取器資料庫執行個體，進而大幅減少停機時間。您可以使用 RDS Proxy 進一步將停機時間減少到一秒鐘或更短的時間。如需詳細資訊，請參閱[升級多可用區域資料庫叢集的引擎版本](#)。

2023 年 11 月 16 日

[SQL 伺服器 RDS 支援 Microsoft SQL 伺服器](#)

您現在可以建立使用 SQL 伺服器 2022 的 RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 Microsoft SQL Server 版本](#)。

2023 年 11 月 15 日

[適用於 MySQL 的 RDS 支援將快照從 5.7 版升級至 8.0](#)

您現在可以將適用於 MySQL 的 RDS 快照集的引擎版本從 5.7 版升級至 8.0 版。您可以使用或 RDS API 或作 ModifyDBSnapshot 業來執行此操作 AWS CLI。AWS Management Console 如需詳細資訊，請參閱 [升級 MySQL 資料庫快照引擎版本](#)。

2023 年 11 月 15 日

[適用於 SQL 伺服器的 RDS 自訂支援 1,000 個資料庫的時間點復原](#)

您現在可以在適用於 SQL Server 的 RDS 自訂資料庫執行個體上建立多達 1,000 個資料庫，進行完整備份和時間點復原資格。如需詳細資訊，請參閱 [將適用於 SQL Server 執行個體的 RDS 自訂還原到某個時間點](#)。

2023 年 11 月 15 日

[適用於 SQL 伺服器的 RDS 自訂支援使用服務主要金鑰](#)

適用於 SQL 伺服器的 RDS 自訂現在支援使用服務主要金鑰 (SMK)。SMK 可讓您加密認證等物件，並使用 TDE 和資料行加密等 SQL Server 功能。如需詳細資訊，請參閱 [< 使用服務主要金鑰搭配使用適用於 SQL Server 的 RDS 自訂 >](#)。

2023 年 11 月 13 日

[Amazon RDS 支援資料庫預覽環境中的 MySQL 8.1](#)

MySQL 8.1 現在可在美國東部 (俄亥俄) 的「資料庫預覽」環境中使用 AWS 區域。如需詳細資訊，請參閱[資料庫預覽環境中的 MySQL 8.1 版](#)。

2023 年 11 月 10 日

[RDS 支援 MySQL 8.0.35 和 MySQL 5.7.44](#)

您現在可以建立執行 MySQL 8.0.35 和 5.7.44 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2023 年 11 月 9 日

[RDS Proxy 支援多可用區資料庫叢集](#)

RDS Proxy 支援連線至多可用區資料庫叢集。如需詳細資訊，請參閱[使用 Amazon RDS Proxy 端點](#)。

2023 年 11 月 9 日

[RDS 自訂適用於甲骨文可在 AWS GovCloud \(US\) Regions](#)

Amazon RDS 現在可在 AWS GovCloud (US) Regions 中使用。如需詳細資訊，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2023 年 11 月 9 日

[Amazon RDS Optimized Writes 支援 db.m5 資料庫執行個體類別](#)

Amazon RDS Optimized Writes 現在支援 db.m5 資料庫執行個體類別。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes for MariaDB 改善寫入效能](#)和[使用 Amazon RDS Optimized Writes for MySQL 改善寫入效能](#)。

2023 年 11 月 9 日

[Amazon RDS for Oracle 支援 CDB 架構的多租戶組態](#)

透過 RDS for Oracle 多租戶功能，RDS 可為您的 Oracle 資料庫提供全受管 Oracle 多租戶架構和體驗。您可以使用 RDS API，在 CDB 中建立多個 PDB (稱為租戶資料庫)。RDS 會提供 CDB 架構的多租戶組態作為舊版單一租戶組態的替代項。如需詳細資訊，請參閱[CDB 架構的多租戶組態](#)。

2023 年 11 月 8 日

[Amazon RDS 將 Performance Insights 指標匯出到 Amazon CloudWatch](#)

Performance Insights 可讓您將預先設定或自訂指標儀表板匯出到 Amazon CloudWatch。匯出的指標儀表板可在 CloudWatch 主控台中檢視。您也可以匯出選取的「Performance Insights」指標 Widget，並在 CloudWatch 主控台中檢視指標資料。如需詳細資訊，請參閱[將 Performance Insights 指標匯出至 CloudWatch](#)。

2023 年 11 月 8 日

[Amazon RDS Custom for Oracle 可讓您升級資料庫執行個體上的作業系統](#)

您現在可以使用 CLI 命令 `modify-db-instance`，升級 RDS Custom for Oracle 資料庫執行個體的資料庫或作業系統 (OS)。如需詳細資訊，請參閱[升級 Amazon RDS Custom for Oracle 的資料庫執行個體](#)。

2023 年 11 月 7 日

[RDS Proxy 支援 RDS for PostgreSQL 的延伸通訊協定](#)

您現在可以在 RDS for PostgreSQL 資料庫執行個體上執行延伸查詢通訊協定。如需詳細資訊，請參閱[Amazon RDS Proxy](#)。

2023 年 11 月 6 日

[RDS 藍/綠部署支援 RDS for PostgreSQL](#)

您現在可以從 RDS for PostgreSQL 資料庫執行個體建立藍/綠部署。如需詳細資訊，請參閱[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

2023 年 10 月 26 日

[更新到 AWS 受管理的策略](#)

AmazonRDSPerformanceInsightsReadOnly 和 AmazonRDSPerformanceInsightsFullAccess 受管政策現在包含 Sid (陳述式 ID) 作為政策陳述式中的識別符。如需詳細資訊，請參閱[AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 10 月 23 日

[RDS Custom for Oracle 支援歐洲 \(米蘭\) 區域](#)

如需詳細資訊，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2023 年 10 月 23 日

[在現有的資料庫上啟用 RDS Optimized Writes](#)

您現在可以在現有的資料庫執行個體上啟用 RDS Optimized Writes，即使它是使用不支援該功能的引擎版本、資料庫執行個體類別或檔案系統組態建立的。如需詳細資訊，請參閱[在現有的資料庫上啟用 RDS Optimized Writes \(適用於 RDS for MySQL\)](#) 和[在現有的資料庫上啟用 RDS Optimized Writes \(適用於 RDS for MariaDB\)](#)。

2023 年 10 月 19 日

[Amazon RDS 支援使用專用日誌磁碟區 \(DLV\)。](#)

您現在可以使用專用日誌磁碟區 (DLV) 搭配 RDS for MariaDB、RDS for MySQL 和 RDS for PostgreSQL。DLV 非常適合具有大型配置儲存體、每秒高 I/O (IOPS) 需求或延遲敏感工作負載的資料庫。如需詳細資訊，請參閱[使用專用日誌磁碟區 \(DLV\)](#)。

2023 年 10 月 17 日

[Amazon RDS for PostgreSQL、MySQL 和 MariaDB 可支援新的資料庫執行個體類別](#)

您可以建立執行 PostgreSQL、MySQL 和 MariaDB 的 Amazon RDS 資料庫執行個體，這些執行個體使用 db.m6.in、db.m6idn、db.r6.in 和 db.r6.idn 資料庫執行個體類別。如需詳細資訊，請參閱[所有可用資料庫執行個體類別的支援資料庫引擎](#)。

2023 年 10 月 12 日

[Amazon RDS for PostgreSQL 支援 pgactive](#)

pgactive 延伸模組已在 Amazon RDS for PostgreSQL 中提供。如需詳細資訊，請參閱[將 PostgreSQL 延伸模組與 Amazon RDS for PostgreSQL 搭配使用](#)。

2023 年 10 月 9 日

[RDS Custom for Oracle 已在亞太區域 \(雅加達\) 提供](#)

您可以在亞太區域 (雅加達) 區域為 Oracle 資料庫執行個體建立 RDS 自訂。如需詳細資訊，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2023 年 10 月 5 日

[RDS Custom for SQL Server 支援新的伺服器層級定序](#)

RDS Custom for SQL Server 現在可針對 SQL_Latin、Japanese、German 和 Arabic 地區設定，支援使用傳統和 UTF-8 編碼的各種不同伺服器定序。如需詳細資訊，請參閱 [RDS Custom for SQL Server 資料庫執行個體的定序和字元支援](#)。

2023 年 9 月 26 日

[更新為 AWS 受管理策略權限](#)

AWSServiceRoleForRDSCustom 服務連結角色具有允許 RDS Custom 建立、修改及刪除 EventBridge 受管規則 AmazonRDSCustomServiceRolePolicy 的新權限。如需詳細資訊，請參閱 [AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 9 月 20 日

[Amazon RDS 向 Amazon 發布 Performance Insights 計數器指標 CloudWatch](#)

CloudWatch 主控台中央的 DB_PERF_INSIGETION 指標數學函數可讓您查詢 Amazon RDS 以 Performance Insights 計數器指標。如需詳細資訊，請參閱 [建立 CloudWatch 警示以監控 Amazon RDS](#)。

2023 年 9 月 20 日

[Performance Insights 可針對 SQL Server 支援摘要層級統計資料](#)

當您使用 Performance Insights 時，可以檢視 Amazon RDS for SQL Server 的陳述式和摘要層級的 SQL 統計資料。如需詳細資訊，請參閱 [分析 SQL Server 中的執行中查詢](#)。

2023 年 9 月 18 日

[Amazon RDS for PostgreSQL、MySQL 和 MariaDB 支援 db.m6.id 和 db.r6.id 資料庫執行個體類別類型](#)

您現在可以建立執行 PostgreSQL、MySQL 和 MariaDB 的 Amazon RDS 資料庫執行個體，這些都會採用記憶體最佳化的 db.m6.id 和 db.r6.id 資料庫執行個體類別類型。這些類型提供了本機 NVMe 型 SSD 儲存體。如需詳細資訊，請參閱[所有可用資料庫執行個體類別的支援資料庫引擎](#)。

2023 年 9 月 11 日

[RDS for PostgreSQL 多可用區域資料庫叢集的主要版本升級支援](#)

您現在可以執行 RDS for PostgreSQL 多可用區域資料庫叢集的主要版本升級。如需詳細資訊，請參閱[升級多可用區域資料庫叢集的引擎版本](#)。

2023 年 9 月 7 日

[Amazon RDS 支援 MariaDB 10.11.5、10.6.15、10.5.22 和 10.4.31](#)

您現在可以建立執行 MariaDB 10.11.5、10.6.15、10.5.22 和 10.4.31 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2023 年 9 月 7 日

[Amazon RDS 延長支援](#)

Amazon RDS 宣布，將在 RDS 標準支援結束日期過後，提供在您的資料庫執行個體中繼續執行 RDS for MySQL 和 RDS for PostgreSQL 主要引擎版本的功能。如需詳細資訊，請參閱[使用 Amazon RDS 延長支援](#)。

2023 年 9 月 1 日

[RDS Custom 支援啟動與停止 RDS Custom for SQL Server 資料庫執行個體](#)

RDS Custom 現在可支援啟動與停止 RDS Custom for SQL Server 資料庫執行個體。如需詳細資訊，請參閱[啟動與停止 RDS Custom for SQL Server 資料庫執行個體](#)。

2023 年 8 月 31 日

[Amazon RDS Optimized Writes 支援 db.r5 資料庫執行個體類別](#)

Amazon RDS Optimized Writes 現在可支援 db.r5 資料庫執行個體類別。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes for MariaDB 改善寫入效能](#)和[使用 Amazon RDS Optimized Writes for MySQL 改善寫入效能](#)。

2023 年 8 月 31 日

[Amazon RDS for Oracle 支援 CDB 的時區檔案自動升級](#)

利用 TIMEZONE_FILE_AUTO UPGRADE 選項，您可在 RDS for Oracle 容器資料庫 (CDB) 上將目前的時區檔案升級為最新版本。如需詳細資訊，請參閱[Oracle 時區檔案自動升級](#)。

2023 年 8 月 29 日

[Amazon RDS Optimized Writes 支援 db.m6g 和 db.m6i 資料庫執行個體類別](#)

Amazon RDS Optimized Writes 現在可支援 db.m6g 和 db.m6i 資料庫執行個體類別。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes for MariaDB 改善寫入效能](#)和[使用 Amazon RDS Optimized Writes for MySQL 改善寫入效能](#)。

2023 年 8 月 28 日

[Amazon RDS 支援 MariaDB 10.11](#)

您現在可以建立執行 MariaDB 10.11 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2023 年 8 月 21 日

[更新為 AWS 受管理策略權限](#)

AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 具有新的權限，可讓 RDS Custom 建立網路介面。如需詳細資訊，請參閱 [AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 8 月 18 日

[更新為 AWS 受管理策略權限](#)

AmazonRDSFullAccess 受管政策具有新的權限，可讓您產生、檢視和刪除一段時間區間內的效能分析報告。如需詳細資訊，請參閱 [AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 8 月 17 日

[更新為 AWS 受管理策略權限](#)

透過向受管政策 AmazonRDSPerformanceInsightsReadOnly 新增新許可權和新增新的受管政策 AmazonRDSPerformanceInsightsFullAccess，您可以產生一段時間區間內的資料庫負載分析報告。如需詳細資訊，請參閱 [AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 8 月 16 日

[Amazon RDS 支援一段時間的
績效分析](#)

Performance Insights 可讓您建立和檢視特定時間區間內的績效分析報告。該報告提供識別出的洞見和解決效能問題的建議。如需詳細資訊，請參閱[分析一段時間區間內的資料庫負載](#)。

2023 年 8 月 16 日

[Amazon RDS Custom
for Oracle 支援 db.r5b 和
db.x2iedn 資料庫執行個體類別](#)

您現在可以將 db.r5b 和 db.x2iedn 執行個體類別用於 RDS Custom for Oracle 資料庫執行個體。如需詳細資訊，請參閱[RDS Custom for Oracle 的資料庫執行個體類別支援](#)。

2023 年 8 月 16 日

[Amazon RDS Custom for
Oracle 支援 db.m6i、db.r6i 和
db.t3 資料庫執行個體類別](#)

您現在可將 db.m6i、db.r6i 和 db.t3 執行個體類別用於 RDS Custom for Oracle 資料庫執行個體。如需詳細資訊，請參閱[RDS Custom for Oracle 的資料庫執行個體類別支援](#)。

2023 年 8 月 15 日

[Amazon RDS for PostgreSQL
現在支援資料庫預覽環境中的
PostgreSQL 16 Beta 3 版](#)

PostgreSQL 版本 16 測試版 3 現在可在美國東部 (俄亥俄州) 的資料庫預覽環境中使用。AWS 區域如需詳細資訊，請參閱[使用資料庫預覽環境](#)。

2023 年 8 月 11 日

[Amazon RDS 支援 MySQL
8.0.34 和 5.7.43](#)

您現在可以建立執行 MySQL 8.0.34 和 5.7.43 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2023 年 8 月 9 日

[RDS for SQL Server 支援待命複本的 OS 指標檢視](#)

您現在可以檢視 RDS for SQL Server 支援待命複本的 OS 指標。如需詳細資訊，請參閱在 [RDS 主控台中檢視 OS 指標](#)。

2023 年 8 月 3 日

[RDS for Oracle 支援 Oracle Data Guard for CDB](#)

Oracle for RDS 支援 Oracle 資料庫 19c 和 21c 容器資料庫 (CDB) 的 Data Guard 僅供讀取複本。您可以使用現有的 RDS API 在 CDB 中建立、管理和升級僅供讀取複本，就像在非 CDB 中一樣。如需詳細資訊，請參閱 [多租戶僅供讀取複本](#)。

2023 年 8 月 1 日

[Amazon RDS 可在以色列 \(特拉維夫\) 區域使用](#)

Amazon RDS 現可於以色列 (特拉維夫) 區域取得。如需更多詳細資訊，請參閱 [區域和可用區域](#)。

2023 年 8 月 1 日

[Amazon RDS 支援 Oracle APEX 23.1.v1 版](#)

您可以將 APEX 23.1.v1 與 Oracle 資料庫 19c 及更新版本一起使用。如需詳細資訊，請參閱 [Oracle Application Express](#)。

2023 年 7 月 26 日

[Amazon RDS Custom for Oracle 支援非預設 Oracle SID](#)

當您使用 Oracle Database 19c 建立 RDS Custom for Oracle 資料庫執行個體時，您可以指定非預設的 Oracle 系統識別符 (Oracle SID)。此值也是 CDB 的名稱。如需詳細資訊，請參閱 [多租用戶架構考量](#)。

2023 年 7 月 21 日

[RDS for SQL Server 支援自我管理 Active Directory](#)

您現在可以使用自我管理 Active Directory，直接將 RDS for SQL Server 資料庫執行個體加入 Microsoft Active Directory (AD) 網域。自我管理 AD 網域可以是內部部署或位於雲端。如需詳細資訊，請參閱[使用自我管理 Active Directory](#)。

2023 年 7 月 7 日

[多可用區域資料庫叢集支援 PostgreSQL 邏輯複寫](#)

您現在可以搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫，來複寫和同步個別資料表，而不是整個資料庫執行個體。如需詳細資訊，請參閱[搭配多可用區域資料庫叢集使用 PostgreSQL 邏輯複寫](#)。

2023 年 7 月 6 日

[Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 第 16 版 Beta 2](#)

PostgreSQL 第 16 版測試版 2 現在可在美國東部 (俄亥俄州) 的資料庫預覽環境中使用。AWS 區域如需詳細資訊，請參閱[使用資料庫預覽環境](#)。

2023 年 7 月 6 日

[更新為 AWS 受管理策略權限](#)

AWSServiceRoleForRDSCustom 服務連結角色的 AmazonRDSCustomServiceRolePolicy 具有新的權限，可讓 RDS Custom for Oracle 使用快照。如需詳細資訊，請參閱[AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 6 月 23 日

RDS 支援 MariaDB 10.6.14、10.5.21 和 10.4.30	您現在可以建立執行 MariaDB 10.6.14、10.5.21 和 10.4.30 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2023 年 6 月 22 日
RDS 支援 MySQL 8.0.33 和 5.7.42	您現在可以建立執行 MySQL 8.0.33 和 5.7.42 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2023 年 6 月 15 日
RDS 支援 MariaDB 10.6.13、10.5.20、10.4.29 和 10.3.39	您現在可以建立執行 MariaDB 10.6.13、10.5.20、10.4.29 和 10.3.39 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2023 年 6 月 15 日
RDS for Oracle 支援可傳輸的資料表空間	您可以使用可傳輸的資料表空間將資料從內部部署 Oracle 資料庫移轉到 RDS for Oracle 資料庫執行個體。這項技術不需要額外的授權，而且是停機時間最少的移轉技術。如需詳細資訊，請參閱 使用 Oracle 可傳輸資料表空間進行移轉 。	2023 年 6 月 15 日
Amazon RDS 支援使用 RDS for MariaDB 第 10.6 版的 RDS Proxy	您現在可以使用 RDS for MariaDB 第 10.6 版資料庫建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱 使用 Amazon RDS Proxy 。	2023 年 6 月 15 日

[RDS Custom for SQL Server 支援使用自有媒體 \(BYOM\)](#)

您現在可以使用自己的 SQL Server 媒體來建立自訂引擎版本 (CEV)。如需詳細資訊，請參閱[透過 RDS Custom for SQL Server 使用自有媒體](#)。

2023 年 6 月 8 日

[RDS for Oracle 可以將 Oracle 資料庫 19c 非 CDB 轉換為 CDB](#)

如果您的資料庫執行個體執行的是具有 2021 年 4 月或更高 RU 的 Oracle 資料庫 19c，則可以將非 CDB 轉換為 CDB (容器資料庫)。轉換架構之後，您可以將 19c CDB 升級為 21c CDB。此步驟是必要的，因為您無法使用單一命令升級資料庫並轉換架構。如需詳細資訊，請參閱[將 RDS for Oracle 非 CDB 轉換為 CDB](#)。

2023 年 5 月 31 日

[多可用區域資料庫叢集已在中國區域供應](#)

異地同步備份資料庫叢集目前可在中 AWS 區域 國 (北京) 和中國 (寧夏) 使用。如需詳細資訊，請參閱[Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎](#)。

2023 年 5 月 30 日

[Amazon RDS Optimized Reads 支援多可用區域資料庫叢集](#)

Amazon RDS Optimized Reads 現在支援多可用區域資料庫叢集 如需詳細資訊，請參閱[使用 Amazon RDS Optimized Reads 改善 RDS for MySQL 的查詢效能](#)和[使用 Amazon RDS Optimized Reads 改善 RDS for PostgreSQL 的查詢效能](#)。

2023 年 5 月 30 日

[Oracle 專用 RDS 自訂支援亞太區域 \(雅加達\) 區域](#)

如需詳細資訊，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2023 年 5 月 29 日

[使用來源 RDS for PostgreSQL 多可用區域資料庫叢集，建立資料庫執行個體僅供讀取複本](#)

您現在可以建立 RDS for PostgreSQL 多可用區域資料庫叢集作為來源的資料庫執行個體僅供讀取複本。之前，只支援 RDS for MySQL。如需詳細資訊，請參閱[從多可用區域資料庫叢集建立資料庫執行個體讀取複本](#)。

2023 年 5 月 24 日

[Amazon RDS 在「Performance Insights」儀表板中提供綜合的 Performance Insights 和 CloudWatch 指標](#)

Amazon RDS 現在可在效能洞見儀表板中提供 Performance Insights 和 CloudWatch 指標的整合檢視。如需詳細資訊，請參閱[在 Amazon RDS 主控台中檢視組合指標](#)。

2023 年 5 月 24 日

[Amazon RDS Optimized Reads 在中國區域中可供使用。](#)

Amazon RDS Optimized Reads 現已在 AWS 區域 中國 (北京) 和中國 (寧夏) 提供。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Reads 改善 RDS for MariaDB 的查詢效能](#)和[使用 Amazon RDS Optimized Reads 改善 RDS for MySQL 的查詢效能](#)。

2023 年 4 月 24 日

[在中國區域提供 AWS Secrets Manager 的 Amazon RDS 支援](#)

Amazon RDS 在中國 (北京) 和中國 (寧夏) 區域支援 Secrets Manager。如需詳細資訊，請參閱[使用 Amazon RDS 和 AWS Secrets Manager 進行密碼管理](#)。

2023 年 4 月 20 日

[RDS Custom for Oracle 支援對新的 CEV 重複使用 AMI ID](#)

當您建立自訂引擎版本 (CEV) 時，RDS Custom for Oracle 會預設為可用的最新 Amazon Machine Image (AMI)。現在，您可以指定在先前 CEV 中使用的 AMI ID。如需詳細資訊，請參閱[建立 CEV](#)。

2023 年 4 月 19 日

[Amazon RDS 支援將事件與標籤發佈至主題訂閱者](#)

傳送至 Amazon 簡易通知服務 (Amazon SNS) 或亞馬遜的 Amazon RDS 事件通知 EventBridge 現在訊息正文中包含事件標籤。這些標籤提供受服務事件影響的資源資料。如需詳細資訊，請參閱[Amazon RDS 事件通知標籤和屬性](#)。

2023 年 4 月 17 日

[購買多可用區域資料庫叢集的預留執行個體](#)

您現在可以為多可用區域資料庫叢集購買預留資料庫執行個體。如需詳細資訊，請參閱[多可用區域資料庫叢集的預留資料庫執行個體](#)。

2023 年 4 月 12 日

[Amazon RDS 支援 db.m7g 和 db.r7g 執行個體類別](#)

您現在可將 db.m7g 和 db.r7g 執行個體類別用於 RDS for MySQL、RDS for MariaDB 和 RDS for PostgreSQL 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體類別的支援資料庫引擎](#)。

2023 年 4 月 12 日

[Amazon RDS Custom 服務連結角色許可的更新](#)

AmazonRDSCustomServiceRolePolicy 現在會授予額外許可，以允許 RDS Custom for SQL Server 使用 Amazon SQS 和建立快照。如需詳細資訊，請參閱 [AWS 受管政策的更新項目](#)。

2023 年 4 月 6 日

[使用僅供讀取複本遷移至 RDS for MySQL 多可用區域資料庫叢集](#)

您現在可以使用僅供讀取複本，將 RDS for MySQL 單一可用區域部署或多可用區域資料庫執行個體部署遷移至 RDS for MySQL 多可用區域資料庫叢集部署，並減少停機時間。如需詳細資訊，請參閱 [使用僅供讀取複本遷移至多可用區域資料庫叢集](#)。

2023 年 4 月 6 日

[從多可用區域資料庫叢集建立資料庫執行個體](#)

您現在可以從多可用區域資料庫叢集建立資料庫執行個體僅供讀取複本，以便擴充超越來源叢集的運算容量。如需詳細資訊，請參閱 [從多可用區域資料庫叢集建立資料庫執行個體讀取複本](#)。

2023 年 4 月 6 日

[Amazon RDS Custom for SQL Server 支援多可用區域](#)

您可以使用 RDS Custom for SQL Server 建立多可用區域部署。如需詳細資訊，請參閱 [管理 RDS Custom for SQL Server 的多可用區部署](#)。

2023 年 4 月 6 日

[更新為 AWS 受管理策略權限](#)

AmazonRDSFullAccess 和 AmazonRDSReadOnlyAccess 政策現在授予其他許可，以允許在 RDS 主控台中顯示 Amazon DevOps Guru 發現項目。如需詳細資訊，請參閱 [AWS 受管政策的 Amazon RDS 更新](#)。

2023 年 3 月 30 日

[Amazon RDS 支援 Oracle APEX 22.2.v1 版](#)

您可以將 APEX 22.2.v1 與 Oracle 資料庫的所有支援版本搭配使用。如需詳細資訊，請參閱 [Oracle Application Express](#)。

2023 年 3 月 30 日

[適用於 RDS 版的 Amazon DevOps 大師](#)

RDS 版 PostgreSQL 會提醒您 Amazon 大師偵測到的最近異常情況。DevOps 主控台的資料庫詳細資料頁面會警示您過去 24 小時內發生的目前和異常狀況。DevOpsGuru 會發佈主動式見解及建議，以協助您在 RDS for PostgreSQL 資料庫中的問題預測發生之前解決問題。如需詳細資訊，請參閱 [RDS DevOps 大師的運作方式](#)。

2023 年 3 月 30 日

[RDS Custom 支援 Amazon EBS gp3 儲存磁碟區](#)

RDS Custom for Oracle 和 RDS Custom for SQL Server 都支援 io1、gp2 和 gp3 SSD 型 EBS 磁碟區。如需詳細資訊，請參閱 [RDS Custom for Oracle 的一般需求](#) 和 [RDS Custom for SQL Server 的一般需求](#)。

2023 年 3 月 29 日

更新為 AWS 受管理策略權限	AmazonRDSFullAccess 和 AmazonRDSReadOnlyAccess 策現在授予 Amazon 的額外許可 CloudWatch。如需詳細資訊，請參閱 AWS 受管政策的 Amazon RDS 更新 。	2023 年 3 月 16 日
RDS Proxy 已在中國區域提供	RDS Proxy 現已在中國 (北京) 和中國 (寧夏) 區域提供。如需 RDS Proxy 的詳細資訊，請參閱 使用 Amazon RDS Proxy 。	2023 年 3 月 15 日
RDS Proxy 已在亞太區域 (雅加達) 區域提供	RDS Proxy 現已在亞太區域 (雅加達) 區域提供。如需 RDS Proxy 的詳細資訊，請參閱 使用 Amazon RDS Proxy 。	2023 年 3 月 8 日
Amazon RDS Optimized Writes 可改善 RDS for MariaDB 的寫入交易效能	您可以使用 Amazon RDS Optimized Writes，改善 RDS for MySQL 資料庫執行個體的寫入交易效能。如需詳細資訊，請參閱 使用 Amazon RDS Optimized Writes 改善 MariaDB 的寫入效能 。	2023 年 3 月 7 日
Amazon RDS for PostgreSQL 15.2 版	Amazon RDS for PostgreSQL 15.2 版的新功能包含適用於條件式 SQL 查詢的 SQL 標準「MERGE」命令、記憶體內和磁碟型排序的效能改善，以及支援邏輯複寫的兩階段遞交和列/欄篩選。	2023 年 2 月 27 日
RDS Custom for Oracle 現已在加拿大 (中部) 區域和南美洲 (聖保羅) 地區提供	如需顯示所有支援的表格 AWS 區域，請參閱 適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎 。	2023 年 2 月 22 日

[Amazon RDS 支援 RDS for MariaDB 和 RDS for MySQL 的跨區域自動備份](#)

您現在可以複製 RDS for MariaDB 和 RDS for MySQL 資料庫執行個體的 AWS 區域間資料庫快照和交易日誌。若需詳細資訊，請參閱[將自動備份複製到另一個 AWS 區域](#)。

2023 年 2 月 22 日

[Amazon RDS for Oracle 支援自動次要版本升級的提前通知](#)

RDS 會提前通知您 RDS for Oracle 引擎的次要版本提供日期。RDS 會在提供日期排程 RDS for Oracle 資料庫執行個體的自動次要版本升級。如需更多詳細資訊，請參閱[排程自動次要版本升級之前](#)。

2023 年 2 月 21 日

[Amazon RDS for SQL Server 支援資料庫活動串流](#)

您現在可以使用「資料庫活動串流」來監督 SQL Server 資料庫執行個體。SQL Server 資料庫執行個體具有由 Amazon RDS 管理的伺服器稽核。您可以定義政策，以便在伺服器稽核規格中記錄伺服器事件。您可以建立資料庫稽核規格，並定義記錄資料庫事件的政策。活動資料串流會收集並傳送至 Amazon Kinesis。您可以透過 Kinesis 監控活動串流以供進一步分析。如需更多詳細資訊，請參閱[使用資料庫活動串流監控 Amazon RDS](#)。

2023 年 2 月 15 日

[RDS 支援 MySQL 8.0.32 和 5.7.41](#)

您現在可以建立執行 MySQL 8.0.32 和 5.7.41 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2023 年 2 月 7 日

[Amazon RDS for Oracle 支援新的 SSL 密碼套件](#)

如果您執行 Oracle Database 19c 或 21c，則可以在 RDS for Oracle 的 SSL 選項中指定六個新密碼套件。這些套件支援 FIPS，並且是 FedRAMP 合規。如需詳細資訊，請參閱 [Oracle Secure Sockets Layer](#)。

2023 年 2 月 3 日

[Amazon RDS for Oracle 支援 Oracle Enterprise Manager 的新 SSL 密碼套件](#)

您可以針對 OEM 選項使用四個新的 FedRAMP 合規加密套件。如需詳細資訊，請參閱 [Oracle Management Agent for Enterprise Manager Cloud Control](#)。

2023 年 2 月 3 日

[RDS for Oracle 支援亞太區域 \(海德拉巴\)、歐洲 \(西班牙\) 和中東 \(阿拉伯聯合大公國\) 區域中的資料庫活動串流](#)

如需詳細資訊，請參閱 [Amazon RDS 中資料庫活動串流的支援區域和資料庫引擎](#)。

2023 年 1 月 27 日

[使用僅供讀取複本遷移至 RDS for PostgreSQL 多可用區域資料庫叢集](#)

透過使用僅供讀取複本，您可以將 RDS for PostgreSQL 單一可用區域部署或多可用區域資料庫執行個體部署遷移至 RDS for PostgreSQL 多可用區域資料庫叢集部署，並減少停機時間。如需詳細資訊，請參閱 [使用僅供讀取複本遷移至多可用區域資料庫叢集](#)。

2023 年 1 月 23 日

[Amazon RDS 可於亞太區域 \(墨爾本\) 區域使用](#)

Amazon RDS 現可於亞太區域 (墨爾本) 區域使用。如需更多詳細資訊，請參閱 [區域和可用區域](#)。

2023 年 1 月 23 日

[RDS for MariaDB 支援強制執行 SSL/TLS 連線](#)

RDS for MariaDB 現在支援強制執行 SSL/TLS 連線，方法是將 `require_secure_transport` 參數設為 ON。如需詳細資訊，請參閱 [MariaDB 資料庫執行個體的所有連線都需要 SSL/TLS](#)。

2023 年 1 月 19 日

[Amazon RDS Optimized Reads 可改善 RDS for MariaDB 的查詢效能](#)

您可以使用 Amazon RDS Optimized Reads，為 RDS for MariaDB 資料庫執行個體實現更快的查詢處理。如需詳細資訊，請參閱 [使用 Amazon RDS Optimized Reads 改善 RDS for MariaDB 的查詢效能](#)。

2023 年 1 月 11 日

[將多可用區域資料庫叢集快照還原至資料庫執行個體](#)

您現在可將多可用區域資料庫叢集快照還原至單一可用區域部署或多可用區資料庫執行個體部署。如需詳細資訊，請參閱 [從多可用區域資料庫叢集快照還原至資料庫執行個體](#)。

2023 年 1 月 10 日

[在建立資料庫執行個體期間指定憑證授權單位 \(CA\)](#)

您現在可以指定哪個 CA 要在建立資料庫執行個體期間用於資料庫執行個體的伺服器憑證。如需詳細資訊，請參閱 [憑證授權單位](#)。

2023 年 1 月 5 日

[RDS Custom for SQL Server 支援自訂引擎版本](#)

RDS Custom for SQL Server 的自訂引擎版本 (AMI) 是已預先安裝 Microsoft SQL Server 的 Amazon Machine Image (AMI)。您可以選擇 Amazon EC2 Windows AMI 用作基礎映像，並且可在作業系統上安裝其他軟體。您可以自訂作業系統和 SQL Server 的組態，以符合您的企業需求。如需詳細資訊，請參閱[使用 RDS Custom for SQL Server 的自訂引擎版本](#)。

2022 年 12 月 28 日

[使用可在其他 AWS 區域使用的 Amazon RDS 藍/綠部署](#)

現可在中國 (北京) 和中國 (寧夏) 區域使用藍/綠部署功能。如需詳細資訊，請參閱[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

2022 年 12 月 22 日

[更新至 IAM 服務連結角色許可](#)

亞馬遜 RDS ServiceRolePolicy 政策現在授予的額外權限。AWS Secrets Manager 如需詳細資訊，請參閱[AWS 受管政策的 Amazon RDS 更新](#)。

2022 年 12 月 22 日

[Amazon RDS 支援重新命名多可用區域資料庫叢集](#)

您現在可以重新命名多可用區域資料庫叢集。如需詳細資訊，請參閱[重新命名多可用區域資料庫叢集](#)。

2022 年 12 月 22 日

[Amazon RDS 與 AWS Secrets Manager 密碼管理集成](#)

Amazon RDS 可在 Secrets Manager 中管理資料庫執行個體或多可用區域資料庫叢集的主要使用者密碼。如需詳細資訊，請參閱[使用 Amazon RDS 和 AWS Secrets Manager 進行密碼管理](#)。

2022 年 12 月 22 日

[Amazon RDS Optimized Writes 支援 db.r6g 和 db.r6gd 資料庫執行個體類別](#)

Amazon RDS Optimized Writes 現在支援 db.r6g 和 db.r6gd 資料庫執行個體類別。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes 改善寫入效能](#)。

2022 年 12 月 22 日

[Amazon RDS 自定義甲骨文支持新 AWS 區域](#)

您可以在亞太區域 (首爾) 和亞太區域 (大阪) 區域中建立 RDS Custom for Oracle 資料庫執行個體。如需詳細資訊，請參閱[適用於 Oracle 的 RDS 自訂支援的區域和資料庫引擎](#)。

2022 年 12 月 21 日

[Amazon RDS 上 AWS Outposts 支援僅供讀取複本](#)

您現在可以從現有的 RDS on Outposts MySQL 或 PostgreSQL DB 資料庫執行個體建立僅供讀取複本。如需詳細資訊，請參閱[建立 Amazon RDS on AWS Outposts 的僅供讀取複本](#)。

2022 年 12 月 19 日

[RDS Custom for Oracle 支援修改資料庫執行個體類別](#)

您現在可以變更 RDS Custom for Oracle 資料庫執行個體的執行個體類別。如需詳細資訊，請參閱[修改 RDS Custom for Oracle 資料庫執行個體](#)。

2022 年 12 月 16 日

[RDS for MySQL 和 RDS for PostgreSQL 支援 db.x2iedn 資料庫執行個體類別](#)

您現在可將 db.x2iedn 資料庫執行個體類別用於 RDS for MySQL 和 RDS for PostgreSQL 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體類別的支援資料庫引擎](#)。

2022 年 12 月 14 日

[Amazon RDS Optimized Writes 支援 db.x2iedn 資料庫執行個體類別](#)

Amazon RDS Optimized Writes 現在支援 db.x2iedn 資料庫執行個體類別。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes 改善寫入效能](#)。

2022 年 12 月 14 日

[Amazon RDS 支援在複製資料庫快照時複製資料庫選項群組](#)

您現在可以複製選項群組，AWS 帳戶 做為 RDS for Oracle 資料庫上的快照複製要求的一部分。如需詳細資訊，請參閱[選項群組考量](#)。

2022 年 12 月 13 日

[Amazon RDS 支援使用 RDS for PostgreSQL 第 14 版的 RDS Proxy](#)

您現在可以使用 RDS for PostgreSQL 第 14 版資料庫建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2022 年 12 月 13 日

[Amazon RDS for Oracle 支援 db.x2idn、db.x2iedn 和 db.x2iezn 執行個體類別](#)

您現在可以將 db.x2idn、db.x2iedn 和 db.x2iezn 執行個體類別用於 Amazon RDS for Oracle 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體類別支援的資料庫引擎](#)和[支援的 RDS for Oracle 執行個體類別](#)。

2022 年 12 月 12 日

[RDS for PostgreSQL 資料庫執行個體支援適用於 PostgreSQL 的受信任語言延伸模組](#)

適用於 PostgreSQL 的受信任語言延伸模組是開放原始碼開發套件，可讓您建置高效能 PostgreSQL 延伸模組，並在 RDS for PostgreSQL 資料庫執行個體上安全地執行這些延伸模組。如需詳細資訊，[使用適用於 PostgreSQL 的受信任語言延伸模組](#)。

2022 年 11 月 30 日

[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)

您可以在預備環境中對資料庫執行個體進行變更，並在不影響生產資料庫執行個體的情況下測試變更。備妥後，您可以將預備環境提升為新的生產環境，停機時間很短。如需詳細資訊，請參閱[使用 Amazon RDS 藍/綠部署進行資料庫更新](#)。

2022 年 11 月 27 日

[Amazon RDS Optimized Writes 可改善 RDS for MySQL 的寫入交易效能](#)

您可以使用 Amazon RDS Optimized Writes，改善 RDS for MySQL 資料庫執行個體的寫入交易效能。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Writes 改善 MySQL 的寫入效能](#)。

2022 年 11 月 27 日

[Amazon RDS Optimized Reads 可改善 RDS for MySQL 的查詢效能](#)

您可以使用 Amazon RDS Optimized Reads，為 RDS for MySQL 資料庫執行個體實現更快的查詢處理。如需詳細資訊，請參閱[使用 Amazon RDS Optimized Reads 改善查詢效能](#)。

2022 年 11 月 27 日

Amazon RDS 可於亞太區域 (海德拉巴) 區域使用	Amazon RDS 現可於亞太區域 (海德拉巴) 區域使用。如需更多詳細資訊，請參閱 區域和可用區域 。	2022 年 11 月 22 日
RDS 支援 MariaDB 10.6.11、10.5.18、10.4.27 和 10.3.37	您現在可以建立執行 MariaDB 10.6.11、10.5.18、10.4.27 和 10.3.37 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2022 年 11 月 18 日
RDS Custom for Oracle 支援在自訂引擎版本 (CEV) 中設定非預設安裝參數	建立 CEV 時，您可以針對 Oracle 基本目錄、Oracle 主目錄、UNIX 使用者名稱與 ID，以及 UNIX 群組名稱與 ID 設定非預設值。如此一來，您可以更好地控制 RDS Custom for Oracle 資料庫執行個體上的資料庫安裝。如需詳細資訊，請參閱 準備 CEV 清單檔案 。	2022 年 11 月 18 日
Amazon RDS 支援 Oracle APEX 22.1.v1 版	您可以搭配所有支援的 Oracle Database 版本使用 APEX 22.1.v1。如需詳細資訊，請參閱 Oracle Application Express 。	2022 年 11 月 18 日
RDS for SQL Server 支援跨區域僅供讀取複本	您現在可以建立跨區域僅供讀取複本，以增強災難復原功能、減少應用程式讀取延遲，以及從主要資料庫執行個體卸載讀取工作負載。如需詳細資訊，請參閱 建立其他僅供讀取複本 AWS 區域 。	2022 年 11 月 16 日

[Amazon RDS 可於歐洲 \(西班牙\) 區域使用](#)

Amazon RDS 現可於歐洲 (西班牙) 區域使用。如需更多詳細資訊，請參閱[區域和可用區域](#)。

2022 年 11 月 16 日

[RDS for SQL Server 支援 Oracle 資料庫的連結伺服器](#)

您現在可以建立連結伺服器來存取外部 Oracle 資料庫，以讀取資料和執行 SQL 命令。如需詳細資訊，請參閱[搭配 Oracle OLEDB 與 RDS for SQL Server 的連結伺服器](#)。

2022 年 11 月 15 日

[RDS Custom for Oracle 支援 Oracle 多租戶](#)

您可以建立 RDS Custom for Oracle 資料庫執行個體，作為容器資料庫 (CDB)。建立後，CDB 包含 CDB 根，PDB 種子和一個 PDB。您可以使用 Oracle SQL 手動建立更多的 PDB。如需詳細資訊，請參閱[Amazon RDS Custom for Oracle 架構的概觀](#)。

2022 年 11 月 15 日

[Amazon RDS for Oracle 支援 Amazon EFS 整合](#)

如果您將 EFS_INTEGRATION 選項新增至選項群組，則可以在 RDS for Oracle 資料庫執行個體與 Amazon EFS 檔案系統之間傳輸檔案。如需詳細資訊，請參閱[Amazon EFS](#)。

2022 年 11 月 15 日

[RDS 支援 MySQL 8.0.31 和 5.7.40](#)

您現在可以建立執行 MySQL 8.0.31 和 5.7.40 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2022 年 11 月 10 日

Amazon RDS 可於歐洲 (蘇黎世) 區域使用	Amazon RDS 現可於歐洲 (蘇黎世) 區域使用。如需更多詳細資訊，請參閱 區域和可用區域 。	2022 年 11 月 9 日
RDS for SQL Server 現在可以存取交易日誌備份	您現在可以檢視資料庫交易日誌備份並將其複製到 Amazon S3 儲存貯體。如需詳細資訊，請參閱 存取交易日誌備份 。	2022 年 11 月 7 日
額外支援異地同步備份資料庫叢集 AWS 區域	異地同步備份資料庫叢集現在提供額 AWS 區域外提供。如需詳細資訊，請參閱 Amazon RDS 中異地同步備份資料庫叢集支援的區域和資料庫引擎 。	2022 年 11 月 4 日
Amazon RDS 支援 gp3 儲存體	您現在可以建立 Amazon RDS 資料庫執行個體，其會使用 Amazon EBS 一般用途 SSD (gp3) 儲存磁碟區，這可讓您獨立於儲存容量自訂儲存體效能。如需詳細資訊，請參閱 一般用途 SSD 儲存體 。	2022 年 11 月 4 日
Amazon RDS 支援作業系統更新的新事件	Amazon RDS 現在支援安全性修補程式事件類別下的新資料庫執行個體事件 (RDS-EVENT-0230)。當作業系統更新可供您的資料庫執行個體使用時，這個新事件會提醒您。如需詳細資訊，請參閱 監控 Amazon RDS 事件 和 使用作業系統更新 。	2022 年 10 月 28 日

[Amazon RDS for Oracle 支援預先設定的 r5b 記憶體優化執行個體類別](#)

db.r5b Oracle 資料庫執行個體類別會針對需要額外記憶體、儲存體和每個 vCPU I/O 的工作負載進行最佳化。例如，db.r5b.4xlarge.tpc2.mem2x 已開啟多執行緒，並提供兩倍於 db.r5b.4xlarge 的記憶體。如需更多詳細資訊，請參閱 [RDS for Oracle 執行個體類別](#)。

2022 年 10 月 27 日

[對於 RDS for MariaDB、MySQL 及 PostgreSQL 資料庫執行個體，Amazon RDS 支援 15 個僅供讀取複本](#)

您現在可以針對 RDS for MariaDB、MySQL 及 PostgreSQL 資料庫執行個體建立至多 15 個僅供讀取複本。如需僅供讀取複本的詳細資訊，請參閱 [使用僅供讀取複本](#)。

2022 年 10 月 20 日

[Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 第 15 版 RC 3](#)

PostgreSQL 版本 15 測試版 3 現在可在美國東部 (俄亥俄州) 的資料庫預覽環境中使用。AWS 區域如需詳細資訊，請參閱 [使用資料庫預覽環境](#)。

2022 年 10 月 18 日

[Amazon RDS 支援自動設定 RDS 資料庫與 EC2 執行個體之間的連線](#)

您可以使用設 AWS Management Console 定現有 RDS 資料庫執行個體或異地同步備份資料庫叢集與 EC2 執行個體之間的連線。如需詳細資訊，請參閱 [自動連線 EC2 執行個體和 RDS 資料庫](#)。

2022 年 10 月 14 日

AWS 適用於 PostgreSQL 的 JDBC 驅動程式正式推出	適用於 PostgreSQL 的 AWS JDBC 驅動程式是專為 RDS 適用於 PostgreSQL 的用戶端驅動程式。AWS JDBC Driver for PostgreSQL 現已全面推出。如需詳細資訊，請參閱 使用適用於 PostgreSQL 的 AWS JDBC 驅動程式連線 。	2022 年 10 月 6 日
Amazon RDS for Oracle 支援 Oracle APEX 21.2.v1 版	APEX 21.2 包含修補程式 33420059。如需詳細資訊，請參閱 APEX 版本需求 。	2022 年 10 月 3 日
RDS 支援 MySQL 5.7.39	您現在可以建立執行 MySQL 5.7.39 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2022 年 9 月 29 日
RDS 支援 MariaDB 10.6.10	您現在可以建立執行 MariaDB 10.6.10 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2022 年 9 月 29 日
RDS Proxy 支援 RDS for SQL Server	您現在可為執行 Microsoft SQL Server 2014 版或更高版本的 RDS 資料庫執行個體建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱 使用 Amazon RDS Proxy 。	2022 年 9 月 19 日

[RDS 支援 MariaDB 10.5.17、10.4.26 和 10.3.36](#)

您現在可以建立執行 MariaDB 10.5.17 版、10.4.26 版和 10.3.36 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2022 年 9 月 15 日

[Amazon RDS for Oracle 支援暫存資料的本機執行個體儲存體](#)

您現在可以在 Amazon EC2 db.r5d 和 db.m5d 執行個體類型上啟動 Amazon RDS for Oracle，並將暫存資料表空間和資料庫智慧型快閃記憶體快取 (快閃記憶體快取) 設定為使用執行個體儲存體。透過在本機存放暫存資料，相較於基於 Amazon EBS 的標準儲存體，您可以達成較低的讀取和寫入延遲。如需詳細資訊，請參閱 [在執行個體儲存體中儲存暫存 Oracle 資料](#)。

2022 年 9 月 14 日

[Performance Insights 顯示前 25 個 SQL 查詢](#)

在 Performance Insights 儀表中，Top SQL (最高 SQL) 索引標籤會顯示在資料庫負載中佔最大比例的 25 個 SQL 查詢。如需詳細資訊，請參閱 [最高 SQL 索引標籤概觀](#)。

2022 年 9 月 13 日

[RDS 支援 MySQL 8.0.30](#)

您現在可以建立執行 MySQL 8.0.30 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2022 年 9 月 9 日

[Amazon RDS 可於中東 \(阿拉伯聯合大公國\) 區域使用](#)

Amazon RDS 現已在中東 (阿拉伯聯合大公國) 區域提供。如需更多詳細資訊，請參閱[區域和可用區域](#)。

2022 年 8 月 30 日

[Amazon RDS for SQL Server 支援 SSRS 電子郵件訂閱](#)

您現在可以使用 SQL Server Reporting Services (SSRS) 電子郵件擴充功能，將報告傳送給使用者，並訂閱報告伺服器上的報告。如需詳細資訊，請參閱[RDS for SQL Server 中的 SQL Server Reporting Services 支援](#)。

2022 年 8 月 26 日

[RDS for Oracle 支援僅供讀取複本備份](#)

您可以為 RDS for Oracle 複本啟用自動備份和手動建立快照。如需詳細資訊，請參閱[使用 RDS for Oracle 複本備份](#)。

2022 年 8 月 23 日

[RDS for Oracle 支援 Oracle Data Guard 轉換](#)

轉換是主要資料庫與掛載或開啟的 Oracle 複本之間的角色反轉。在轉換期間，原始主要資料庫會轉換為待命角色，而原始待命資料庫則會轉換為主要角色。如需詳細資訊，請參閱[執行 Oracle Data Guard 轉換](#)。

2022 年 8 月 23 日

[Amazon RDS 支援自動設定與 EC2 執行個體的連線](#)

建立資料庫執行個體或異地同步備份資料庫叢集時，可以使用設 AWS Management Console 定 Amazon Elastic Compute Cloud 執行個體與新資料庫執行個體或資料庫叢集之間的連線。如需詳細資訊，對於新的資料庫執行個體，請參閱[設定 EC2 執行個體的自動網路連線](#)；對於新的資料庫叢集，請參閱[設定 EC2 執行個體的自動網路連線](#)。

2022 年 8 月 22 日

[RDS Custom for Oracle 支援 Oracle 複本提升](#)

如果您使用 RDS Custom for Oracle，您可以使用 promote-read-replica CLI 命令提升您受管的 Oracle 複本。此外，您也可以刪除主要資料庫執行個體，這會讓 RDS Custom for Oracle 將受管的 Oracle 複本提升為獨立執行個體。如需詳細資訊，請參閱[使用 RDS Custom for Oracle 的 Oracle 複本](#)。

2022 年 8 月 5 日

[RDS for MySQL 支援強制執行 SSL/TLS 連線](#)

RDS for MySQL 現在支援強制執行 SSL/TLS 連線，方法是將 require_secure_transport 參數設為 ON。如需詳細資訊，請參閱[需要以 SSL/TLS 連線至 MySQL 資料庫執行個體](#)。

2022 年 8 月 1 日

Amazon RDS 已淘汰支援 Oracle Database 12c 版本 1 (12.1.0.2)	12.1.0.2 版已淘汰支援 BYOL 和 LI 授權模式。2022 年 8 月 1 日，RDS for Oracle 會開始自動升級 12c 版本 1 (12.1.0.2) 資料庫執行個體，並將 12.1.0.2 快照還原至 Oracle Database 19c。如需詳細資訊，請參閱 AWS re:Post 上的終止支援時間表。	2022 年 8 月 1 日
RDS Proxy 支援 RDS for MariaDB	您現在可以為執行 MariaDB 10.2、10.3、10.4 或 10.5 版的 RDS 資料庫執行個體建立 RDS Proxy。MariaDB 的支援包含在 MySQL 引擎系列之下。如需 RDS Proxy 的詳細資訊，請參閱 使用 Amazon RDS Proxy 。	2022 年 7 月 26 日
RDS for MariaDB 支援 db.r5b 資料庫執行個體類別	您現在可以建立採用 db.r5b 資料庫執行個體類別的 RDS for MariaDB 資料庫執行個體。如需詳細資訊，請參閱 資料庫執行個體類別的支援資料庫引擎 。	2022 年 7 月 25 日
RDS for Oracle 支援修改資料庫活動串流	如果您使用 RDS for Oracle，可以將資料庫活動串流的稽核政策狀態變更為已鎖定 (預設) 或已解鎖。您可以解鎖其政策狀態、自訂稽核政策，然後重新鎖定政策狀態，而不用停止活動串流。如需詳細資訊，請參閱 修改資料庫活動串流 。	2022 年 7 月 22 日

[績效詳情支援亞太區域 \(雅加達\) 區域](#)

之前您無法在亞太區域 (雅加達) 使用績效詳情。此限制已經移除。如需詳細資訊，請參閱 [Amazon RDS 中支援的區域和資料庫引擎以 Performance Insights](#)。

2022 年 7 月 21 日

[Microsoft SQL Server 2012 已經終止在 Amazon RDS 上的支援](#)

Microsoft SQL Server 2012 已經終止支援，此版本相符的 Microsoft 支援計畫延長到 2022 年 7 月 12 日終止。任何現存的 Microsoft SQL Server 2012 執行個體，將於 2022 年 6 月 1 日起自動升級到 Microsoft SQL Server 2014 最新的次要版本。如需詳細資訊，請參閱 [Microsoft SQL Server 2012 支援 Amazon RDS](#)。

2022 年 7 月 12 日

[RDS 支援 MariaDB 10.6.8、10.5.16、10.4.25、10.3.35 和 10.2.44](#)

您現在可以建立執行 MariaDB 10.6.8、10.5.16、10.4.25、10.3.35 和 10.2.44 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上支援的 MariaDB 版本](#)。

2022 年 7 月 8 日

[RDS 績效詳情支援更多保留期間](#)

之前，績效詳情僅提供兩個保留期間：7 天 (預設) 或 2 年 (731 天)。現在，如果您需要保留績效資料超過 7 天，可以指定 1-24 個月。如需詳細資訊，請參閱 [績效詳情的定價和資料保留](#)。

2022 年 7 月 1 日

[RDS Custom 支援亞太區域 \(孟買\) 和歐洲 \(倫敦\) 區域](#)

您可以在兩個新的方式中建立適用於 Oracle 和 RDS 自訂 SQL Server 資料庫執行個體的 RDS 自訂 AWS 區域：亞太區域 (孟買) 和歐洲 (倫敦)。如需詳細資訊，請參閱 [RDS Custom for Oracle 的AWS 區域支援](#) 和 [RDS Custom for SQL Server 的AWS 區域支援](#)。

2022 年 6 月 21 日

[RDS Custom for Oracle 支援 Oracle Database 18c 與 12c 第 2 版 \(12.2\)](#)

現在您可以使用 Oracle Database 18c 和 12c Release 2 (12.2) 的安裝檔案，為 RDS Custom for Oracle 建立 CEV。您可以使用這些 CEV 建立 RDS Custom for Oracle 資料庫執行個體。如需詳細資訊，請參閱 [使用 Amazon RDS Custom for Oracle 的自訂引擎版本](#)。

2022 年 6 月 21 日

[多可用區域資料庫叢集支援 db.m5d 和 db.r5d 資料庫執行個體類別](#)

現在您可以建立使用 db.m5d 與 db.r5d 資料庫執行個體類別的多可用區域資料庫叢集。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署與資料庫執行個體類別類型](#)。

2022 年 6 月 21 日

[另外提供異地同步備份資料庫叢集 AWS 區域](#)

現在，您可以在以下區域建立多可用區域資料庫叢集：歐洲 (法蘭克福) 和歐洲 (斯德哥爾摩)。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

2022 年 6 月 21 日

RDS for Microsoft SQL Server 支援遷移採用透明資料加密 (TDE) 的資料庫	RDS for SQL Server 現在支援在 TDE 啟用的情況下，使用原生備份與還原遷移 Microsoft SQL Server 資料庫。如需詳細資訊，請參閱 支援 SQL Server 的透明資料加密 。	2022 年 6 月 14 日
Amazon RDS 支援將事件發佈至加密的 Amazon SNS 主題	Amazon RDS 現在可以將事件發佈到已啟用伺服器端加密 (SSE) 的 Amazon Simple Notification Service (Amazon SNS) 主題，以便為帶有敏感資料的事件提供額外保護。如需詳細資訊，請參閱 訂閱 Amazon RDS 事件通知 。	2022 年 6 月 1 日
RDS 支援 MySQL 5.7.38	您現在可以建立執行 MySQL 5.7.38 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2022 年 5 月 31 日
RDS for PostgreSQL 支援階層式僅供讀取複本	您現在可將階層式僅供讀取複本與 RDS for PostgreSQL 14.1 版和更高版本搭配使用。如需詳細資訊，請參閱 關於 Amazon RDS 中使用 PostgreSQL 僅供讀取複本 。	2022 年 5 月 4 日
Amazon RDS on AWS Outposts 支援擴展儲存和自動調整規模操作	您現在可以變更 Outpost 上資料庫執行個體的儲存體大小，並使用儲存體自動擴展。如需詳細資訊，請參閱 AWS Outposts 上的 Amazon RDS 支援 Amazon RDS 功能 。	2022 年 5 月 2 日

[另外提供異地同步備份資料庫叢集 AWS 區域](#)

您現在可在下列區域建立多可用區域資料庫叢集：亞太區域 (新加坡) 和亞太區域 (雪梨)。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

2022 年 4 月 29 日

[Amazon RDS 支援雙堆疊模式](#)

資料庫執行個體現在可以雙堆疊模式執行。於雙堆疊模式中，資源可透過 IPv4、IPv6 或兩者與資料庫執行個體進行通訊。如需詳細資訊，請參閱 [Amazon RDS IP 定址](#)。

2022 年 4 月 29 日

[Amazon RDS 向 Amazon 發布用量指標 CloudWatch](#)

Amazon 中的 AWS/Usage 命名空間 CloudWatch 包含 Amazon RDS 服務配額的帳戶層級使用量指標。如需詳細資訊，請參閱 [Amazon RDS 的 Amazon CloudWatch 使用量指標](#)。

2022 年 4 月 28 日

[Amazon RDS for MySQL 支援 db.m6i 和 db.r6i 資料庫執行個體類別](#)

您現在可將 db.m6i 和 db.r6i 資料庫執行個體類別用於執行 MySQL 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。

2022 年 4 月 28 日

[Amazon RDS for PostgreSQL 支援 db.m6i 和 db.r6i 資料庫執行個體類別](#)

您現在可將 db.m6i 和 db.r6i 資料庫執行個體類別用於執行 PostgreSQL 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。

2022 年 4 月 27 日

[Amazon RDS for MariaDB 支援 db.m6i 和 db.r6i 資料庫執行個體類別](#)

您現在可將 db.m6i 和 db.r6i 資料庫執行個體類別用於執行 MariaDB 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體類別的支援資料庫引擎](#)。

2022 年 4 月 26 日

[Amazon RDS 上 AWS Outposts 支持異地同步備份部署](#)

您現在可在其他 Outpost 上建立待命資料庫執行個體。如需詳細資訊，請參閱[針對 Amazon RDS 功能的 Amazon RDS AWS Outposts 支援](#)。

2022 年 4 月 19 日

[Amazon RDS for Oracle 支援 db.m6i 和 db.r6i 執行個體類別](#)

若您執行 Oracle 資料庫 19c，您可使用 db.m6i 和 db.r6i 執行個體類別。db.m6i 類別是非常適合用於廣範圍工作負載的一般用途執行個體類別。如需更多詳細資訊，請參閱[RDS for Oracle 執行個體類別](#)。

2022 年 4 月 8 日

[Amazon RDS for SQL Server 支援 SQL Server Agent 任務複寫](#)

開啟此功能時，在主要主機上建立、修改或刪除的 SQL Server Agent 任務將自動同步至多可用區組態中的次要主機。如需更多資訊，請參閱[使用 SQL Server Agent](#)。

2022 年 4 月 7 日

[Amazon RDS 支援使用 RDS for PostgreSQL 第 13 版的 RDS Proxy](#)

您現在可以使用 RDS for PostgreSQL 第 13 版資料庫建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2022 年 4 月 4 日

[Amazon RDS 計劃取代 Oracle Database 12c](#)

Oracle Database 12c 即將被取代。甲骨文公司將不再為 Oracle 資料庫 12c 版本提供修正程式 end-of-support。Amazon RDS 計劃啟動將 Oracle Database 12c 資料庫執行個體自動升級至 Oracle Database 19c。

2022 年 3 月 22 日

[Amazon RDS for PostgreSQL 版本備註](#)

現在有一個單獨的 Amazon RDS for PostgreSQL 版本備註的指南。如需詳細資訊，請參閱 [Amazon RDS for PostgreSQL 版本備註](#)。

2022 年 3 月 22 日

[Amazon RDS for Oracle 版本備註](#)

現在有一個單獨的 Amazon RDS for Oracle 版本備註的指南。如需詳細資訊，請參閱 [Amazon RDS for Oracle 版本備註](#)。

2022 年 3 月 22 日

[另外提供異地同步備份資料庫叢集 AWS 區域](#)

現在，您可以在以下區域建立多可用區域資料庫叢集：美國東部 (俄亥俄) 和亞太區域 (東京)。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

2022 年 3 月 15 日

[Amazon RDS for PostgreSQL 版本 14.2、13.6、12.10、11.15 及 10.20](#)

RDS for PostgreSQL 現在支援版本 14.2、13.6、12.10、11.15 及 10.20。版本 14.2 和 13.6 新增支援兩個新的外部資料包裝函式。mysql_fdw 擴展功能可讓 PostgreSQL 處理存放在 MySQL、MariaDB 及 Aurora MySQL 資料庫中的資料。tds_fdw 擴展功能可讓 PostgreSQL 處理存放在 SQL Server 資料庫中的資料。如需詳細資訊，請參閱[支援的 PostgreSQL 資料庫版本](#)。

2022 年 3 月 12 日

[RDS 支援 MySQL 5.7.37](#)

您現在可以建立執行 MySQL 5.7.37 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2022 年 3 月 11 日

[Amazon RDS for SQL Server 支援新的資料庫執行個體類別](#)

您現在可以建立使用 db.m6i 和 db.r6i 資料庫執行個體類別來執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Microsoft SQL Server 的資料庫執行個體類別支援](#)。

2022 年 3 月 9 日

[Amazon RDS for Oracle 支援 Oracle Database 21c](#)

您現在可以建立執行 Oracle Database 21c (21.0.0.0) 的 Amazon RDS 資料庫執行個體。這是第一個僅支援多重租用 (CDB) 架構的 Oracle Database 版本。如需詳細資訊，請參閱[Oracle Database 21c 搭配 Amazon RDS](#)。

2022 年 3 月 7 日

[RDS 支援 MariaDB 10.6.7、10.5.15、10.4.24、10.3.34 和 10.2.43](#)

您現在可以建立執行 MariaDB 10.6.7、10.5.15、10.4.24、10.3.34 和 10.2.43 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2022 年 3 月 3 日

[AWS 適用於 MySQL 的 JDBC 驅動程序一般可用](#)

對於 MySQL 的 AWS JDBC 驅動程序是專為 MySQL 的 RDS 的客戶端驅動程序。適用於 MySQL 的 AWS JDBC 驅動程式現已正式推出。如需詳細資訊，請參閱 [與 Amazon Web Services JDBC Driver for MySQL 連線](#)。

2022 年 3 月 2 日

[多可用區域資料庫叢集全面推出](#)

多可用區域資料庫叢集部署是 Amazon RDS 的高可用性部署模式，具有兩個可讀取待命資料庫執行個體。多可用區域資料庫叢集現在全面推出。如需詳細資訊，請參閱 [多可用區域資料庫叢集部署](#)。

2022 年 3 月 1 日

[RDS 支援 MySQL 8.0.28](#)

您現在可以建立執行 MySQL 8.0.28 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2022 年 2 月 28 日

Amazon RDS for Oracle 支援原生網路加密 (NNE) 的新設定	若要控制用戶端是否可以使用非安全加密和檢查總和方法進行連線，請在 NNE 選項中設定 SQLNET.ALLOW_WEAK_CRYPT0_CLIENTS 和 SQLNET.ALLOW_WEAK_CRYPT0。不安全方法的範例包括 DES、3DES、RC4 和 MD5。如需詳細資訊，請參閱 NNE 選項設定 。	2022 年 2 月 25 日
Amazon RDS for SQL Server 支援 Microsoft SQL Server 2017 Standard Edition 的 Always On 可用性群組	當在 SQL Server 2017 Standard Edition 14.00.3401.7 和更高版本上使用多可用區域組態建立資料庫執行個體時，RDS 會自動使用可用性群組。如需詳細資訊，請參閱 Microsoft SQL Server 的異地同步備份部署 。	2022 年 2 月 18 日
RDS for Oracle 支援亞太區域 (雅加達) 區域的資料庫活動串流	如需詳細資訊，請參閱 資料庫活動串流 AWS 區域的 Support 。	2022 年 2 月 16 日
Amazon RDS Custom for Oracle 支援 Oracle 資料庫 12.1	您可以現在為使用 Oracle Database 12.1 Enterprise Edition 的 RDS Custom for Oracle 建立自訂引擎版本。如需詳細資訊，請參閱 使用 Amazon RDS Custom for Oracle 的自訂引擎版本 。	2022 年 2 月 4 日
Amazon RDS for MariaDB 支援新的主要版本	您現在可以建立執行 MariaDB 10.6 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 10.6 支援 。	2022 年 2 月 3 日

[績效詳情支援 Oracle 查詢的計畫擷取](#)

績效詳情主控台支援最高 SQL 的新計畫維度。按計畫配量時，您可以查看最高 Oracle 查詢正在使用哪些計畫。如果查詢使用多個計畫，可以在主控台中並排比較計畫，確定哪個計畫最有效。您還可以向下切入以查看計畫中哪些步驟的成本最高。如需詳細資訊，請參閱[使用績效詳情儀表板來分析 Oracle 執行計畫](#)。

2022 年 1 月 27 日

[績效詳情支援新的 API](#)

績效詳情支援以下 API：GetResourceMetadata、ListAvailableResourceDimensions 及 ListAvailableResourceMetrics。如需詳細資訊，請參閱本手冊中的[使用績效詳情 API 來擷取指標](#)，以及 [Amazon RDS 績效詳情 API 參考](#)。

2022 年 1 月 12 日

[RDS Proxy 支援事件](#)

RDS Proxy 現在會產生事件，您可以在事件中訂閱和檢視這些 CloudWatch 事件，或設定為傳送至 Amazon EventBridge。如需詳細資訊，請參閱[使用 RDS Proxy 事件](#)。

2022 年 1 月 11 日

[Amazon RDS for SQL Server 支援 SSAS 多維度模式](#)

RDS for SQL Server 支援在表格或多維度模式中執行 SQL Server Analysis Services (SSAS)。如需詳細資訊，請參閱[支援 RDS for SQL Server 中的 SQL Server Analysis Services](#)。

2022 年 1 月 7 日

[RDS 代理伺服器提供額外的 AWS 區域](#)

RDS Proxy 現在已在下列區域開放使用：非洲 (開普敦)、亞太區域 (香港)、亞太區域 (大阪)、歐洲 (米蘭)、歐洲 (巴黎)、歐洲 (斯德哥爾摩)、中東 (巴林) 和南美洲 (聖保羅)。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2022 年 1 月 5 日

[RDS 支援 MySQL 8.0.27](#)

您現在可以建立執行 MySQL 8.0.27 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2021 年 12 月 21 日

[Amazon RDS 可於亞太區域 \(雅加達\) 區域使用](#)

Amazon RDS 現可於亞太區域 (雅加達) 區域使用。如需更多詳細資訊，請參閱[區域和可用區域](#)。

2021 年 12 月 13 日

[Amazon RDS 支援 MariaDB 10.5.13、10.4.22、10.3.32 和 10.2.41](#)

您現在可以建立執行 MariaDB 版本 10.5.13 ; 10.4.22、10.3.32 和 10.2.41 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2021 年 12 月 8 日

[Amazon RDS Custom for SQL Server](#)

Amazon RDS Custom 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊版、自訂和封裝應用程式。使用 Amazon RDS Custom，您可以獲得 Amazon RDS 自動化和 Amazon EC2 的彈性。如需詳細資訊，請參閱[使用 Amazon RDS Custom](#)。

2021 年 12 月 1 日

[多可用區域資料庫叢集 \(預覽\)](#)

您現在可以為 RDS for MySQL 和 RDS for PostgreSQL 建立多可用區域資料庫叢集。多可用區域資料庫叢集部署是 Amazon RDS 的高可用性部署模式，具有兩個可讀取待命資料庫執行個體。多可用區域資料庫叢集處於預覽狀態。如需詳細資訊，請參閱[多可用區域資料庫叢集部署 \(預覽\)](#)。

2021 年 11 月 23 日

[Amazon RDS 支援使用 RDS for PostgreSQL 12 版的 RDS Proxy](#)

您現在可以使用 RDS for PostgreSQL 第 12 版資料庫建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱[使用 Amazon RDS Proxy](#)。

2021 年 11 月 22 日

[Amazon RDS 上 AWS Outposts 支持本地備份](#)

您可以將自動備份和手動快照儲存在 Outpost AWS 區域或本機中。如需詳細資訊，請參閱[針對 Amazon RDS 功能的 Amazon RDS AWS Outposts 支援](#)。

2021 年 11 月 22 日

[Amazon RDS 跨帳戶支持
AWS KMS keys](#)

將資料庫快照匯出到 Amazon S3 時，您可以使用不同 AWS 帳戶的 KMS 金鑰進行加密。如需更多詳細資訊，請參閱[將資料庫快照資料匯出至 Amazon S3](#)。

2021 年 11 月 3 日

[AWS Outposts 上的 Amazon RDS 支援將資料庫引擎日誌發佈到 CloudWatch 日誌](#)

Outposts 上的 RDS 現在支援將資料庫引擎記錄發佈至記錄 CloudWatch 檔。如需詳細資訊，請參閱[Amazon RDS 對 AWS Amazon RDS 功能的支援](#)。

2021 年 11 月 2 日

[Amazon RDS Custom for Oracle](#)

Amazon RDS Custom 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊版、自訂和封裝應用程式。使用 Amazon RDS Custom，您可以獲得 Amazon RDS 自動化和 Amazon EC2 的彈性。如需詳細資訊，請參閱[使用 Amazon RDS Custom](#)。

2021 年 10 月 26 日

[支援 RDS for MySQL 8.0 版的延遲複寫](#)

從 RDS for MySQL 8.0.26 版開始，您可以針對 RDS for MySQL 8.0 版資料庫執行個體設定延遲複寫。如需詳細資訊，請參閱[透過 MySQL 設定延遲複寫](#)。

2021 年 10 月 25 日

[支援 MySQL 8.0.26](#)

您現在可以建立執行 MySQL 8.0.26 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2021 年 10 月 25 日

[支援 RDS for MySQL 8.0 版的 GTID 式複寫](#)

從 RDS for MySQL 8.0.26 版開始，您可以針對 RDS for MySQL 8.0 版資料庫執行個體設定 GTID 式複寫。如需詳細資訊，請參閱[使用 RDS for MySQL 的 GTID 式複寫](#)。

2021 年 10 月 25 日

[Amazon RDS 支援使用 RDS for MySQL 8.0 的 RDS Proxy](#)

您現在可以為 RDS for MySQL 8.0 資料庫執行個體建立 RDS Proxy。如需詳細資訊，請參閱[Amazon RDS Proxy](#)。

2021 年 10 月 21 日

[AWS Outposts 上的 Amazon RDS 支持其他適用於 MySQL 的 RDS 版本](#)

RDS on Outposts 現在支援 RDS for MySQL 8.0.23 和 8.0.25 版本。如需詳細資訊，請參閱[Amazon RDS 對 AWS Amazon RDS 功能的支援](#)。

2021 年 10 月 20 日

[Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 14 版 RC 1](#)

PostgreSQL 第 14 版 RC 1 現在可在美國東部 (俄亥俄) 的資料庫預覽環境中使用。AWS 區域如需詳細資訊，請參閱[使用資料庫預覽環境](#)。

2021 年 10 月 19 日

[Amazon RDS 額外支援 Performance Insights AWS 區域](#)

績效詳情已在中東 (巴林)、非洲 (開普敦)、歐洲 (米蘭) 和亞太區域 (大阪) 區域推出。如需詳細資訊，請參閱[Amazon RDS 中支援的區域和資料庫引擎以 Performance Insights](#)。

2021 年 10 月 5 日

[績效詳情支援 Oracle 的摘要層級統計資料](#)

當您使用績效詳情時，您可以檢視 Amazon RDS for Oracle 的陳述式和摘要層級的 SQL 統計資料。如需詳細資訊，請參閱[分析 Oracle 中的執行中查詢](#)。

2021 年 10 月 4 日

[AWS Outposts 地台上的 Amazon RDS 支援其他 RDS PostgreSQL 本](#)

RDS on Outposts 現在支援 RDS for PostgreSQL 12.8 和 13.4 版。如需詳細資訊，請參閱 [Amazon RDS 對 AWS Amazon RDS 功能的支援](#)。

2021 年 10 月 1 日

[Amazon RDS 支援 Oracle APEX 21.1.v1 版](#)

您可以搭配所有支援版本的 Oracle 資料庫使用 APEX 21.1.v1。如需詳細資訊，請參閱 [Oracle Application Express](#)。

2021 年 9 月 24 日

[Amazon RDS for Oracle 支援 NNE 的用戶端加密](#)

設定 NNE 時，您可能想要避免在伺服器端強制加密。例如，您可能不希望因為伺服器要求而強制所有用戶端通訊使用加密。在這種情況下，您可以使用 SQLNET.*CLIENT 選項，強制在用戶端上執行加密。如需詳細資訊，請參閱 [Oracle 原生網路加密](#)。

2021 年 9 月 24 日

[Amazon RDS for MySQL 和 RDS for PostgreSQL 支援新的資料庫執行個體類別](#)

您現在可以使用 db.r5b、db.t4g 和 db.x2g 執行個體類別，建立執行 MySQL 或 PostgreSQL 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [資料庫執行個體類別的支援資料庫引擎](#)。

2021 年 9 月 15 日

Amazon RDS for Microsoft SQL Server 支援 Java Database Connectivity (JDBC) 與 Microsoft Distributed Transaction Coordinator (MSDTC)	MSDTC for SQL Server 2017 14.00.3223.3 版和更新版本以及 SQL Server 2019 現在支援 JDBC XA 交易。如需詳細資訊，請參閱 RDS for SQL Server 中的 Microsoft Distributed Transaction Coordinator 支援 。	2021 年 9 月 7 日
Amazon RDS 支援 MariaDB 10.5.12、10.4.21、10.3.31 和 10.2.40	您現在可以建立執行 MariaDB 版本 10.5.12、10.4.21、10.3.31 和 10.2.40 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2021 年 9 月 2 日
Amazon RDS 已終止對 Oracle Database 18c 的支援	您只能為 Oracle 資料庫 12c 和 Oracle 資料庫 19c 建立資料庫執行個體。若您有 Oracle Database 18c 快照，請將其升級至最新版本。如需詳細資訊，請參閱 升級 Oracle 資料庫快照 。	2021 年 8 月 17 日
Amazon RDS for SQL Server 支援自動次要版本升級	您現在可將 RDS for SQL Server 資料庫執行個體自動升級至最新的次要版本。如需詳細資訊，請參閱 升級 Microsoft SQL Server 資料庫引擎 。	2021 年 8 月 13 日

[Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 14 beta 2](#)

如需 PostgreSQL 版本 14 beta 1 的詳細資訊，請參閱 [PostgreSQL 14 beta 1 版本備註](#)。如需 PostgreSQL 版本 14 beta 2 的詳細資訊，請參閱 [PostgreSQL 14 beta 2 版本備註](#)。如需資料庫預覽環境的相關資訊，請參閱 [使用資料庫預覽環境](#)。

2021 年 8 月 9 日

[Amazon RDS 支援共用 VPC 中的 RDS Proxy](#)

您現在可於共用 VPC 中建立 RDS Proxy。如需 RDS Proxy 的詳細資訊，請參閱 [《Amazon RDS 使用者指南》](#) 或 [《Aurora 使用者指南》](#) 中的「使用 Amazon RDS Proxy 管理連線」。

2021 年 8 月 6 日

[Amazon RDS 支援 MariaDB 10.2.39](#)

您現在可以建立執行 MariaDB 10.2.39 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2021 年 8 月 4 日

[Amazon RDS for Oracle 新增 TIMEZONE_FILE_AUTO UPGRADE 選項](#)

利用此選項，您可在您的 Oracle 資料庫執行個體上將目前的時區檔案升級為最新版本。如需詳細資訊，請參閱 [Oracle 時區檔案自動升級](#)。

2021 年 7 月 30 日

[Amazon RDS 延伸支援跨區域自動備份](#)

您現在可以在更多的 AWS 區域之間複寫資料庫快照和交易日誌。如需詳細資訊，請參閱 [將自動備份複製到其他 AWS 區域](#)。

2021 年 7 月 19 日

[支援 MySQL 5.7.34](#)

您現在可以建立執行 MySQL 5.7.34 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2021 年 7 月 8 日

[AWS Outposts 地台上的 Amazon RDS 支援其他 RDS PostgreSQL 本](#)

RDS on Outposts 現在支援 RDS for PostgreSQL 12.7 和 13.3 版本。如需詳細資訊，請參閱 [Amazon RDS 對 AWS Amazon RDS 功能的支援](#)。

2021 年 7 月 8 日

[Amazon RDS for PostgreSQL 支援 oracle_fdw](#)

您現在可以使用 `oracle_fdw` 延伸來提供外部資料包裝，以存取 Oracle 資料庫。如需更多詳細資訊，請參閱 [使用 oracle_fdw 延伸存取外部資料](#)。

2021 年 7 月 8 日

[Amazon RDS 支援 Oracle Management Agent \(OMA\) 13.5 版](#)

您可以將 Oracle Management Agent (OMA) 13.5 版與 Oracle Enterprise Manager (OEM) Cloud Control 13c 版本 5 及更新版本搭配使用。Amazon RDS for Oracle 會安裝 OMA，接著由 OMA 和您的 Oracle Management Service (OMS) 進行通訊，提供監控資訊。如果您執行 OMS 13.5，可以安裝 OMA 13.5 來管理資料庫。如需詳細資訊，請參閱 [Oracle Management Agent for Enterprise Manager Cloud Control](#)。

2021 年 7 月 7 日

[Amazon RDS for Oracle 支援 從 Amazon S3 下載日誌](#)

如果封存的重做日誌不在您的執行個體上，但受到備份保留期間的保護，請從 Amazon S3 使用 `rdsadmin.rdsadmin_archive_log_download` 下載。RDS for Oracle 會將日誌儲存至資料庫執行個體上的 `/rdsdbdata/log/arch` 目錄。如需更多詳細資訊，請參閱[從 Amazon S3 下載封存的重做日誌](#)。

2021 年 7 月 2 日

[Amazon RDS 支援 MariaDB 版本 10.4.18 和 10.5.9](#)

您現在可以建立執行 MariaDB 10.4.18 和 10.5.9 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2021 年 6 月 30 日

[Amazon RDS for Oracle 支援 資料庫活動串流](#)

您現在可以使用「資料庫活動串流」來監督 Oracle 資料庫執行個體。Oracle 資料庫會將稽核記錄寫入統一稽核權杖。當您在 Oracle 資料庫執行個體上啟動資料庫活動串流時，Amazon Kinesis 會串流符合 Oracle 資料庫稽核政策的所有活動。如需更多詳細資訊，請參閱[使用資料庫活動串流監控 Amazon RDS](#)。

2021 年 6 月 23 日

[Amazon RDS for Oracle 推出 記憶體最佳化執行個體類別](#)

新 Oracle 資料庫執行個體會針對需要額外記憶體、儲存空間和每個 vCPU 輸入/輸出的工作負載進行最佳化。如需更多詳細資訊，請參閱[RDS for Oracle 執行個體類別](#)。

2021 年 6 月 23 日

支援 MySQL 8.0.25	您現在可以建立執行 MySQL 8.0.25 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2021 年 6 月 18 日
AWS Outposts 地台上的 Amazon RDS 支援其他 RDS PostgreSQL 本	RDS on Outposts 現在支援 RDS for PostgreSQL 12.5、12.6、13.1 和 13.2 版本。如需詳細資訊，請參閱 Amazon RDS 對 AWS Amazon RDS 功能的支援 。	2021 年 5 月 28 日
Amazon RDS 支援 MariaDB 版本 10.2.37 和 10.3.28	您現在可以建立執行 MariaDB 10.2.37 版和 10.3.28 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2021 年 5 月 27 日
Amazon RDS for Oracle 支援多租用戶容器資料庫 (CDB)	多租用戶架構可讓 Oracle 資料庫成為 CDB。在 Oracle 資料庫 19c 中，您的 CDB 可以包括一個單一的 PDB。採用 PDB 的使用者體驗與採用非 CDB 的使用者體驗大致相同。如需詳細資訊，請參閱 RDS for Oracle 架構 。	2021 年 5 月 25 日
Amazon RDS 在 AWS Outposts 支持 Amazon RDS for SQL Server	RDS on Outposts 現在支援 Amazon RDS for SQL Server。如需詳細資訊，請參閱 Amazon RDS 對 AWS Amazon RDS 功能的支援 。	2021 年 5 月 11 日

[Amazon RDS 延伸支援跨區域自動備份](#)

您現在可以設定執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體，將資料庫快照和交易日誌複製到不同的 AWS 區域。如需詳細資訊，請參閱將[自動備份複製到其他 AWS 區域](#)。

2021 年 5 月 7 日

[Amazon RDS 支援加密資料庫執行個體的跨區域自動備份](#)

對於執行 Oracle 或 PostgreSQL 的加密 Amazon RDS 資料庫執行個體，現在可將資料庫快照和交易日誌複製到其他 AWS 區域。如需詳細資訊，請參閱將[自動備份複製到其他 AWS 區域](#)。

2021 年 5 月 3 日

[AWS Outposts 上的 Amazon RDS 支持 Amazon 監控 CloudWatch](#)

Outposts 上的 RDS 現在支持 Amazon CloudWatch 監控。如需詳細資訊，請參閱[Amazon RDS 對 AWS Amazon RDS 功能的支援](#)。

2021 年 4 月 21 日

[RDS 支 AWS PostgreSQL da 函數](#)

您現在可以為 RDS 版 PostgreSQL 資料庫執行個體叫用 AWS Lambda 函數。如需詳細資訊，請參閱[從 RDS for PostgreSQL 資料庫執行個體叫用 AWS Lambda 函數](#)。

2021 年 4 月 13 日

[RDS for SQL Server 支援擴充事件](#)

您可以使用 SQL Server 擴充事件來擷取偵錯和疑難排解資訊。如需詳細資訊，請參閱[搭配 Amazon RDS for Microsoft SQL Server 使用擴充事件](#)。

2021 年 4 月 8 日

對於 MySQL 8.0.23、5.7.33 和 5.6.51 的支援	您現在可以建立執行 MySQL 8.0.23、5.7.33 和 5.6.51 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2021 年 3 月 31 日
Amazon RDS for MySQL 升級失敗時自動轉返	如果資料庫執行個體從 MySQL 5.7 版升級至 MySQL 8.0 版失敗，Amazon RDS 會自動轉返升級執行的變更。轉返後，MySQL 資料庫執行個體將執行 MySQL 5.7 版。如需詳細資訊，請參閱 從 MySQL 5.7 升級至 8.0 失敗後轉返 。	2021 年 3 月 18 日
Amazon RDS 支援選擇加入區域中的跨區域僅供讀取複本	您現在可以將資料庫執行個體複寫至選擇加入區域。如需詳細資訊，請參閱 在不同 AWS 區域建立僅供讀取複本 。	2021 年 3 月 18 日
Amazon RDS 計劃取代 Oracle Database 18c	Oracle 資料庫 18c (18.0.0.0) 即將被取代。甲骨文公司將不再為「Oracle 資料庫 18c」提供修補程式。end-of-support2021 年 7 月 1 日，Amazon RDS 計劃啟動將 Oracle 資料庫 18c 執行個體自動升級至 Oracle 資料庫 19c。在自動升級開始之前，我們強烈建議您手動將現有的 Oracle 資料庫 18c 執行個體升級至 Oracle 資料庫 19c。如需詳細資訊，請參閱 準備 Oracle 資料庫 18c 的自動升級 。	2021 年 3 月 11 日

[Amazon RDS 已終止對 Oracle Database 11g 的支援](#)

您只能為 Oracle 資料庫 12c 版本 1 (12.1.0.2) 及更新版本建立資料庫執行個體。如果您有 Oracle Database 11g 快照，請將其升級至更新版本。如需詳細資訊，請參閱[升級 Oracle 資料庫快照](#)。

2021 年 3 月 11 日

[Amazon RDS 支援資料庫執行個體的持續備份 AWS Backup](#)

您現在可以在這些備份中建立自動備份，AWS Backup 並將資料庫執行個體還原到指定的時間。如需詳細資訊，請參閱[使用 AWS Backup 來管理自動備份](#)。

2021 年 3 月 10 日

[Amazon RDS 支援 Oracle Management Agent \(OMA\) 13.4 版](#)

您可以將 Oracle Management Agent (OMA) 13.4 版與 Oracle Enterprise Manager (OEM) Cloud Control 13c 版本 4 更新 9 搭配使用。Amazon RDS for Oracle 會安裝 OMA，接著由 OMA 和您的 Oracle Management Service (OMS) 進行通訊，提供監控資訊。如果您執行 OMS 13.4，可以安裝 OMA 13.4 來管理資料庫。如需詳細資訊，請參閱[Oracle Management Agent for Enterprise Manager Cloud Control](#)。

2021 年 3 月 10 日

[RDS 代理端點增強功能](#)

您可以建立與每個 RDS 代理相關聯的其他端點。在不同的 VPC 中建立端點，可啟用代理的跨 VPC 存取。Aurora MySQL 叢集代理也可以擁有唯讀端點。這些讀取器端點會連線到叢集中的讀取者資料庫執行個體，並可改善查詢密集型應用程式的讀取延展性和可用性。如需 RDS Proxy 的詳細資訊，請參閱 [Amazon RDS 使用者指南](#) 或 [Aurora 使用者指南](#) 中的「使用 Amazon RDS Proxy 管理連線」。

2021 年 3 月 8 日

[Amazon RDS 延伸對跨區域自動備份的支援](#)

您現在可以設定執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，將資料庫快照和交易日誌複製到不同 AWS 的區域。如需詳細資訊，請參閱 [將自動備份複製到其他 AWS 區域](#)。

2021 年 3 月 8 日

[中國 \(北京\) 區域和中國 \(寧夏\) 區域支援 Amazon RDS for MariaDB 和 MySQL 的複寫篩選](#)

中國 (北京) 區域和中國 (寧夏) 區域現可支援複寫篩選。如需詳細資訊，請參閱 [使用 MariaDB 設定複寫篩選條件](#) 和 [使用 MySQL 設定複寫篩選條件](#)。

2021 年 3 月 5 日

[Amazon RDS 支援選擇加入區域中的跨區域資料庫快照複本](#)

您現在可以將資料庫快照複製到選擇加入的 AWS 區域，或從中複製。如需詳細資訊，請參閱 [跨 AWS 區域複製快照](#)。

2021 年 3 月 4 日

Amazon RDS for SQL Server 支援 Standard Edition 的 Always On 可用性群組	當您針對 Standard Edition 資料庫引擎使用 SQL Server 2019 上的異地同步備份組態建立資料庫執行個體時，RDS 會自動使用可用性群組。如需詳細資訊，請參閱 Microsoft SQL Server 的異地同步備份部署 。	2021 年 2 月 23 日
Amazon RDS for Oracle 推出建議程式相關程序	rdsadmin_util 套件包含程序 advisor_task_set_parameter、advisor_task_drop 和 dbms_stats_init。您可以使用這些程序來修改、停止及重新啟用 AUTO_STATS_ADVISOR_TASK 等建議程式任務。如需詳細資訊，請參閱 設定建議程式任務的參數 。	2021 年 2 月 23 日
Amazon RDS 提供異地同步備份資料庫執行個體的容錯移轉原因	當異地同步備份資料庫執行個體容錯移轉至待命複本時，您即刻可以看到更詳細的說明。如需詳細資訊，請參閱的 Amazon RDS 容錯移轉程序 。	2021 年 2 月 18 日
Amazon RDS 將對匯出快照的支援延伸至 Amazon S3	您現在可以將資料庫快照資料匯出至中國中 Amazon S3。如需詳細資訊，請參閱 將資料庫快照資料匯出至 Amazon S3 。	2021 年 2 月 17 日

[Amazon RDS for MariaDB 和 MySQL 的複寫篩選條件](#)

您可以為 MySQL 和 MariaDB 執行個體設定複寫篩選條件。複寫篩選條件指定要在僅供讀取複本中複寫的資料庫和資料表。您可以為每個僅供讀取複本建立要包含或排除的資料庫和資料表清單。如需詳細資訊，請參閱[使用 MariaDB 設定複寫篩選條件](#)和[使用 MySQL 設定複寫篩選條件](#)。

2021 年 2 月 12 日

[RDS for Oracle 支援 APEX 20.2v1](#)

您可以將 APEX 20.2.v1 與 Oracle 資料庫的所有支援版本搭配使用。如需詳細資訊，請參閱[Oracle Application Express](#)。

2021 年 2 月 2 日

[Amazon RDS for SQL Server 支援 tempdb 資料庫的本機執行個體儲存體](#)

您現在可以在 Amazon EC2 db.r5d 和 db.m5d 執行個體類型上啟動 Amazon RDS for SQL Server，並將 tempdb 資料庫設定為使用執行個體存放區。透過將 tempdb 資料檔案和日誌檔案放置於本機，相較於基於 Amazon EBS 的標準儲存體，您可以達成較低的讀取和寫入延遲。如需詳細資訊，請參閱[Amazon RDS for SQL Server 上對於 tempdb 資料庫的執行個體存放區支援](#)。

2021 年 1 月 27 日

Amazon RDS for PostgreSQL 可支援 pg_partman 和 pg_cron	Amazon RDS for PostgreSQL 現可支援 pg_partman 和 pg_cron 擴充功能。如需有關 pg_partman 擴充功能的詳細資訊，請參閱 使用 pg_partman 擴充功能來管理 PostgreSQL 分割區 。如需有關 pg_cron 擴充功能的詳細資訊，請參閱 使用 PostgreSQL pg_cron 擴充功能來安排維護 。	2021 年 1 月 12 日
Amazon RDS 支援將 Oracle 管理代理程式日誌發佈到 Amazon CloudWatch 日誌	Oracle Management Agent 日誌包含 emctl.log、emdctlj.log、gcagent.log、gcagent_errors.log、emagent.nohup 和 secure.log。Amazon RDS 會將這些日誌中的每一個日誌作為單獨的日 CloudWatch 誌串流發佈。如需詳細資訊，請參閱將 Oracle 日誌發佈到 Amazon CloudWatch 日誌 。	2020 年 12 月 28 日
AWS Outposts 上的 Amazon RDS 支援其他資料庫版本	Outposts 上的 RDS 現已支援其他 MySQL 和 PostgreSQL 版本。如需詳細資訊，請參閱 Amazon RDS 對 AWS Amazon RDS 功能的支援 。	2020 年 12 月 23 日
AWS Outposts 上的 Amazon RDS 支持 COIP	Outposts 上的 RDS 現已支援客戶擁有的 IP 地址 (CoIP)。CoIP 可透過內部部署網路，為您 Outpost 子網路中的資源提供本機或外部連線能力。如需詳細資訊，請參閱 適用於 Outposts 上 RDS 的客戶擁有的 IP 地址 。	2020 年 12 月 22 日

[Amazon RDS for Oracle 規劃將 11g BYOL 執行個體升級至 19c](#)

2021 年 1 月 4 日，我們計劃開始自動將所有使用自有授權 (BYOL) 模組上的 Oracle 資料庫 11g 執行個體版本升級至 Oracle 資料庫 19c。所有 Oracle 資料庫 11g 執行個體 (包括預留執行個體) 都將移轉至最新可用的版本更新 (RU)。如需詳細資訊，請參閱[準備 Oracle 資料庫 11g BYOL 的自動升級](#)。

2020 年 12 月 11 日

[Amazon RDS 支援將自動備份複寫到另一個 AWS 區域](#)

您現在可以設定 Amazon RDS 資料庫執行個體，將快照和交易日誌複寫到您選擇的目的地 AWS 區域。如需詳細資訊，請參閱[將自動備份複製到其他 AWS 區域](#)。

2020 年 12 月 4 日

[Amazon RDS for Oracle 和 Microsoft SQL Server 支援新的資料庫執行個體類別](#)

您現在可以使用 db.r5b 執行個體來建立執行 Oracle 或 SQL Server 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體類別的支援資料庫引擎](#)。

2020 年 12 月 4 日

[支援 MariaDB 10.2.32](#)

您現在可以建立執行 MariaDB 10.2.32 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2020 年 11 月 25 日

[Amazon RDS for SQL Server 可在 SQL Server 2019 上支援 Microsoft Business Intelligence Suite](#)

您現在可以使用最新的主要版本，在資料庫執行個體上執行 Server Analysis Services、SQL Server Integration Services 和 SQL Server Reporting Services。如需詳細資訊，請參閱 [Microsoft SQL Server 資料庫引擎的選項](#)。

2020 年 11 月 24 日

[資料庫預覽環境中的 Amazon RDS for PostgreSQL 13 版](#)

Amazon RDS for PostgreSQL 現已支援資料庫預覽環境中的 PostgreSQL 13 版。如需詳細資訊，請參閱 [PostgreSQL 13 版](#)。

2020 年 11 月 24 日

[Amazon RDS 績效詳情引進新的維度](#)

您可以根據資料庫 (PostgreSQL、MySQL 和 MariaDB)、應用程式 (PostgreSQL) 和工作階段類型 (PostgreSQL) 的維度群組對資料庫負載分組。Amazon RDS 還支援維度 db.name (PostgreSQL、MySQL 和 MariaDB)、db.application.name (PostgreSQL) 和 db.session_type.name (PostgreSQL)。如需詳細資訊，請參閱 [最高負載資料表](#)。

2020 年 11 月 24 日

[Amazon RDS for MariaDB 支援新的主要版本](#)

您現在可以建立執行 MariaDB 10.5 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2020 年 11 月 23 日

[支援 MySQL 5.6.49](#)

您現在可以建立執行 MySQL 5.6.49 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2020 年 11 月 20 日

[支援 MySQL 5.5.62](#)

您現在可以建立執行 MySQL 5.5.62 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2020 年 11 月 20 日

[績效詳情支援分析執行 PostgreSQL 查詢的統計資訊](#)

您現在可以使用績效詳情，分析對 PostgreSQL 資料庫執行個體執行查詢的統計資訊。如需詳細資訊，請參閱 [PostgreSQL 的統計資料](#)。

2020 年 11 月 18 日

[Amazon RDS 擴充功能支援儲存體自動調整規模](#)

您現在可以在建立僅供讀取複本、將資料庫執行個體還原至指定的時間，或從 Amazon S3 備份還原 MySQL 資料庫執行個體時，啟用儲存體自動調整規模。如需詳細資訊，請參閱 [使用 Amazon RDS 儲存體自動調整規模自動管理容量](#)。

2020 年 11 月 18 日

[Amazon RDS for SQL Server 支援 Database Mail](#)

使用 Database Mail 讓您可從 Amazon RDS for SQL Server 資料庫執行個體傳送電子郵件訊息。指定電子郵件收件者之後，您可以將檔案或查詢結果新增至您傳送的訊息。如需詳細資訊，請參閱 [在 Amazon RDS for SQL Server 上使用 Database Mail](#)。

2020 年 11 月 4 日

[支援 MySQL 8.0.21](#)

您現在可以建立執行 MySQL 8.0.21 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2020 年 10 月 22 日

[Amazon RDS 將對匯出快照的支援延伸至 Amazon S3](#)

您現在可以將資料庫快照資料匯出到所有商業 AWS 區域的 Amazon S3。如需更多詳細資訊，請參閱 [將資料庫快照資料匯出至 Amazon S3](#)。

2020 年 10 月 22 日

[Amazon RDS for PostgreSQL 支援僅供讀取複本升級](#)

使用 Amazon RDS for PostgreSQL，當您執行主要資料庫執行個體的主要版本升級時，僅供讀取複本也會自動升級。如需詳細資訊，請參閱 [升級 PostgreSQL 資料庫引擎](#)。

2020 年 10 月 15 日

[Amazon RDS for MariaDB、MySQL 和 PostgreSQL 支援 Graviton2 資料庫執行個體類別](#)

您現在可以使用 Graviton2 資料庫執行個體類別 db.m6g.x 和 db.r6g.x 來建立執行 MariaDB、MySQL 或 PostgreSQL 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [所有可用資料庫執行個體類別的支援資料庫引擎](#)。

2020 年 10 月 15 日

[Amazon RDS for SQL Server 支援升級至 SQL Server 2019](#)

您可以將 SQL Server 資料庫執行個體升級至 SQL Server 2019。如需詳細資訊，請參閱 [升級 Microsoft SQL Server 資料庫引擎](#)。

2020 年 10 月 6 日

Amazon RDS for Oracle 支援指定國家字元集	國家字元集 (又稱為 NCHAR 字元集)，用於 NCHAR、NVARCHAR2 和 NCLLOB 資料類型。當您建立資料庫時，可以將 AL16UTF16 (預設) 或 UTF8 指定為 NCHAR 字元集。如需詳細資訊，請參閱 Amazon RDS 中支援的 Oracle 字元集 。	2020 年 10 月 2 日
支援 MySQL 5.7.31	您現在可以建立執行 MySQL 5.7.31 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2020 年 10 月 1 日
Amazon RDS for PostgreSQL 支援將資料導出到 Amazon S3	您可以從 PostgreSQL 資料庫執行個體查詢資料，然後將資料直接匯出至 Amazon S3 儲存貯體中存放的檔案。如需詳細資訊，請參閱 從 RDS for PostgreSQL 資料庫執行個體中匯出資料到 Amazon S3 。	2020 年 9 月 24 日
Amazon RDS for MySQL 為 MySQL 8.0 支持佩爾科納 XtraBackup	您現在可以使用佩爾科納 XtraBackup 將備份還原到 Amazon RDS 版 MySQL 8.0 資料庫執行個體。如需詳細資訊，請參閱 將備份還原至 MySQL 資料庫執行個體 。	2020 年 9 月 17 日
Amazon RDS for SQL Server 支援具有僅供讀取複本的資料庫執行個體上的原生備份和還原	您可以將 SQL Server 原生備份還原至已設定僅供讀取複本的資料庫執行個體。如需詳細資訊，請參閱 匯入和匯出 SQL Server 資料庫 。	2020 年 9 月 16 日

Amazon RDS for SQL Server 支援其他時區	您可以將資料庫執行個體時區與您選擇的時區進行比對。如需詳細資訊，請參閱 Microsoft SQL Server 資料庫執行個體的本機時區 。	2020 年 9 月 11 日
資料庫預覽環境中 Amazon RDS for PostgreSQL 版本 13 Beta 3	Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 13 Beta 3。如需詳細資訊，請參閱 PostgreSQL 13 版 。	2020 年 9 月 9 日
Amazon RDS for SQL Server 支援追蹤旗標 692	您現在可以使用追蹤旗標 692 做為啟動參數使用資料庫參數群組。啟用此追蹤旗標會在大量載入資料至堆積或叢集索引時停用快速插入。如需詳細資訊，請參閱 在大量載入期間停用快速插入 。	2020 年 8 月 27 日
Amazon RDS for SQL Server 支援 Microsoft SQL Server 2019	您現在可以建立使用 SQL Server 2019 的 RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 Microsoft SQL Server 版本 。	2020 年 8 月 26 日
RDS for Oracle 支援掛載複本資料庫	建立或修改 Oracle 複本時，您可以將其置於掛載模式。由於複本資料庫不接受使用者連線，因此無法提供唯讀工作負載。掛載複本會在套用封存重做日誌檔後刪除它們。掛載複本的主要用途是跨區域災難復原。如需詳細資訊，請參閱 Oracle 複本概觀 。	2020 年 8 月 13 日

[RDS for Oracle 規劃升級 11g SE1 LI 執行個體](#)

在 2020 年 11 月 1 日，我們計劃開始將 Oracle 資料庫 11g SE1 License Included (LI) 執行個體自動升級至適用於 Amazon RDS for Oracle 的 Oracle 資料庫 19c。所有 11g 執行個體 (包括預留執行個體) 都將移轉至最新可用的 Oracle 版本更新 (RU)。如需詳細資訊，請參閱[準備 Oracle 資料庫 11g SE1 的自動升級](#)。

2020 年 7 月 31 日

[Amazon RDS 可在 PostgreSQL 和 MySQL 的預覽版本中支援全新 Graviton2 資料庫執行個體類別](#)

您現在可以建立執行 PostgreSQL 或 MySQL 的 Amazon RDS 資料庫執行個體，其會採用 db.m6g.x 和 db.r6g.x 資料庫執行個體類別。如需詳細資訊，請參閱[所有可用資料庫執行個體類別的支援資料庫引擎](#)。

2020 年 7 月 30 日

[RDS for Oracle 支援 APEX 20.1v1](#)

您可以將 APEX 20.1v1 與 Oracle 資料庫的所有支援版本搭配使用。如需詳細資訊，請參閱[Oracle Application Express](#)。

2020 年 7 月 28 日

[支援 MySQL 8.0.20](#)

您現在可以建立執行 MySQL 8.0.20 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2020 年 7 月 23 日

Amazon RDS for MariaDB 和 MySQL 可支援新的資料庫執行個體類別	您現在可以建立執行 MariaDB 和 MySQL 的 Amazon RDS 資料庫執行個體，這些執行個體使用 db.m5.16xlarge、db.m5.8xlarge、db.r5.16xlarge 及 db.r5.8xlarge 執行個體類別。如需詳細資訊，請參閱 所有可用資料庫執行個體類別的支援資料庫引擎 。	2020 年 7 月 23 日
RDS for SQL Server 支援停用舊版 TLS 和密碼	您可以開啟和關閉特定安全性通訊協定和密碼。如需詳細資訊，請參閱 設定安全性通訊協定和密碼 。	2020 年 7 月 21 日
RDS 可在 SE2 上支援 Oracle Spatial	針對 12.2、18c 和 19c 的所有版本，您可在 Standard Edition 2 (SE2) 中使用 Oracle Spatial。如需詳細資訊，請參閱 Oracle Spatial 。	2020 年 7 月 9 日
Amazon RDS 支持 AWS PrivateLink	Amazon RDS 現在支援為 Amazon RDS API 呼叫建立 Amazon VPC 端點，以便在 AWS 網路中保留應用程式和 Amazon RDS 之間的流量。如需更多詳細資訊，請參閱 Amazon RDS 和界面 VPC 端點 (AWS PrivateLink) 。	2020 年 7 月 9 日
Amazon RDS for PostgreSQL 9.4.x 版已經終止支援。	Amazon RDS for PostgreSQL 不再支援 9.4.x 版。如需支援的版本，請參閱 支援的 PostgreSQL 資料庫版本 。	2020 年 7 月 8 日

[支援 MariaDB 10.3.23 和 10.4.13](#)

您現在可以建立執行 MariaDB 10.3.23 和 10.4.13 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MariaDB 版本](#)。

2020 年 7 月 6 日

[Amazon RDS 上 AWS Outposts](#)

您可以在 AWS Outposts 上建立 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 [使用 Amazon RDS on AWS Outposts](#)。

2020 年 7 月 6 日

[Amazon RDS for Oracle 會自動建立庫存檔案](#)

為針對 BYOL 客戶開立服務請求，Oracle Support 會請求 Opatch 產生的庫存檔案。Amazon RDS for Oracle 會每小時自動在 BDUMP 目錄中建立庫存檔案。如需詳細資訊，請參閱 [存取 Opatch 檔案](#)。

2020 年 7 月 6 日

[對於 MySQL 5.7.30 和 5.6.48 的支援](#)

您現在可以建立執行 MySQL 5.7.30 和 5.6.48 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2020 年 6 月 25 日

[Amazon RDS for Oracle 支援 ADRCI](#)

自動診斷儲存器命令解譯器 (ADRCI) 公用程式是 Oracle 命令列工具，您可以用來管理診斷資料。透過使用 Amazon RDS 套件中的函數 `rdsadmin_adrci_util`，您可以列出並封裝問題和事件，也可以顯示追蹤檔案。如需詳細資訊，請參閱 [Oracle 資料庫執行個體的一般 DBA 診斷任務](#)。

2020 年 6 月 17 日

[支援 MySQL 8.0.19](#)

您現在可以建立執行 MySQL 8.0.19 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [Amazon RDS 上的 MySQL 版本](#)。

2020 年 6 月 2 日

[MySQL 8.0 支援小寫資料表名稱](#)

您現在可以針對執行 MySQL 8.0.19 版及更高版本 8.0 版的 Amazon RDS 資料庫執行個體，將 `lower_case_table_names` 參數設定為 1。如需詳細資訊，請參閱 [Amazon RDS 資料庫執行個體的 MySQL 參數例外狀況](#)。

2020 年 6 月 2 日

[Amazon RDS for Microsoft SQL Server 支援 SQL Server Integration Services \(SSIS\)](#)

SSIS 是資料整合和工作流程應用程式的平台。您可以在現有或新的資料庫執行個體上啟用 SSIS。它安裝在與資料庫引擎相同的資料庫執行個體上。如需詳細資訊，請參閱 [SQL Server 中的 SQL Server Integration Services 支援](#)。

2020 年 5 月 19 日

[Amazon RDS for Microsoft SQL Server 支援 SQL Server Reporting Services \(SSRS\)](#)

SSRS 是以伺服器為基礎的應用程式，用於產生和分發報告。您可以在現有或新的資料庫執行個體上啟用 SSRS。它安裝在與資料庫引擎相同的資料庫執行個體上。如需詳細資訊，請參閱 [SQL Server 中的 SQL Server Reporting Services 支援](#)。

2020 年 5 月 15 日

[Amazon RDS for Microsoft SQL Server 支援多個可用區執行個體上的 S3 整合](#)

您現在可以使用 Amazon S3 搭配 SQL Server 功能，例如在多個可用區資料庫執行個體上進行大量插入。如需詳細資訊，請參閱 [整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

2020 年 5 月 15 日

[Amazon RDS for Oracle 支援清除資源回收筒](#)

`rdsadmin.rdsadmin_util.purge_dba_recyclebin` 程序會清除資源回收筒。如需詳細資訊，請參閱 [清除資源回收筒](#)。

2020 年 5 月 13 日

[Amazon RDS for Oracle 改善了自動工作負載儲存庫 \(AWR\) 的可管理性](#)

這些 `rdsadmin.rdsadmin_diagnostic_util` 程序會產生 AWR 報告，並將 AWR 資料擷取至傾印檔案。如需詳細資訊，請參閱 [使用自動工作負載儲存庫 \(AWR\) 產生效能報告](#)。

2020 年 5 月 13 日

Amazon RDS for Microsoft SQL Server 支援 Microsoft Distributed Transaction Coordinator (MSDTC)	Amazon RDS for SQL Server 支援主機之間的分散式交易。如需詳細資訊，請參閱 SQL Server 中的 Microsoft Distributed Transaction Coordinator 支援 。	2020 年 5 月 4 日
Amazon RDS for Microsoft SQL Server 支援新版本	您現在可以針對所有版本，建立執行 SQL Server 2017 CU19 14.00.328 1.6、2016 SP2 CU11 13.00.5598.27、2014 SP3 CU4 12.00.6329.1，以及 2012 SP4 GDR 11.0.7493.4 版本的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 Microsoft SQL Server 版本 。	2020 年 4 月 28 日
Amazon RDS 可於歐洲 (米蘭) 區域使用	Amazon RDS 現可於歐洲 (米蘭) 區域使用。如需詳細資訊，請參閱 區域和可用區域 。	2020 年 4 月 28 日
Local Zones 支援 Amazon RDS	您現在可以對本地區域子網路啟動資料庫執行個體。如需詳細資訊，請參閱 區域、可用區域和 Local Zones 。	2020 年 4 月 23 日
Amazon RDS 可於非洲 (開普敦) 區域使用	Amazon RDS 現可於非洲 (開普敦) 區域使用。如需詳細資訊，請參閱 區域和可用區域 。	2020 年 4 月 22 日

[Amazon RDS for Microsoft SQL Server 支援 SQL Server Analysis Services \(SSAS\)](#)

SSAS 是安裝在 SQL Server 內的線上分析處理 (OLAP) 和資料採礦工具。您可以在現有或新的資料庫執行個體上啟用 SSAS。它安裝在與資料庫引擎相同的資料庫執行個體上。如需詳細資訊，請參閱[支援 SQL Server 中的 SQL Server Analysis Services](#)。

2020 年 4 月 17 日

[適用於 PostgreSQL 的 Amazon RDS Proxy](#)

Amazon RDS Proxy 現在可用於 PostgreSQL。您可以使用 RDS Proxy 來減少資料庫執行個體上的連線管理額外負荷，也可以減少發生「連線過多」錯誤的機會。RDS Proxy 目前位於 PostgreSQL 的公開預覽中。如需更多詳細資訊，請參閱[使用 Amazon RDS Proxy 管理連線 \(預覽版\)](#)。

2020 年 4 月 8 日

[Amazon RDS for Oracle 支援 Oracle APEX 19.2.v1 版](#)

Amazon RDS for Oracle 現支援 Oracle Application Express (APEX) 19.2.v1 版。如需詳細資訊，請參閱[Oracle Application Express](#)。

2020 年 4 月 8 日

[Amazon RDS for MariaDB 支援新的主要版本](#)

您現在可以建立執行 MariaDB 10.4 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2020 年 4 月 6 日

Amazon RDS 績效詳情可供 Amazon RDS for MariaDB 10.4 版使用	Amazon RDS 績效詳情現可供 Amazon RDS for MariaDB 10.4 版使用。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2020 年 4 月 6 日
Amazon RDS for PostgreSQL 9.3.x 版已經終止支援	Amazon RDS for PostgreSQL 不再支援 9.3.x 版。如需支援的版本，請參閱 支援的 PostgreSQL 資料庫版本 。	2020 年 4 月 3 日
Amazon RDS for Microsoft SQL Server 支援僅供讀取複本	您現在可以建立適用於 SQL Server 資料庫執行個體的僅供讀取複本。如需詳細資訊，請參閱 使用僅供讀取複本 。	2020 年 4 月 3 日
Amazon RDS for Microsoft SQL Server 支援多文件備份	您現在可以使用 SQL Server 原生備份和還原，將資料庫備份到多個檔案。如需詳細資訊，請參閱 備份資料庫 。	2020 年 4 月 2 日
Amazon RDS for Oracle 集成 AWS License Manager	亞馬遜 RDS 適用於甲骨文現已與 AWS License Manager。如果您使用「自攜授權」模式，AWS License Manager 整合可讓您更輕鬆地監控組織內的 Oracle 授權使用情況。如需詳細資訊，請參閱 與整合 AWS License Manager 。	2020 年 3 月 23 日

[支援 Amazon RDS for MariaDB 和 MySQL 中的 64 TiB 的 db.r5 執行個體](#)

現在您可以建立適用於 MariaDB 與 MySQL 的 Amazon RDS 資料庫執行個體，該執行個體使用高達 64 TiB 儲存空間的 db.r5 資料庫執行個體類別。如需詳細資訊，請參閱[影響儲存體效能的因素](#)。

2020 年 3 月 18 日

[支援 MySQL 8.0.17](#)

您現在可以建立執行 MySQL 版本 8.0.17 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2020 年 3 月 10 日

[Amazon RDS 績效詳情可供 Amazon RDS for MySQL 8.0 版使用](#)

Amazon RDS 績效詳情現已可供 Amazon RDS for MySQL 8.0.17 版和更高的 8.0 版使用。如需詳細資訊，請參閱[使用 Amazon RDS 績效詳情](#)。

2020 年 3 月 10 日

[支援 MySQL 5.6.46](#)

您現在可以建立執行 MySQL 版本 5.6.46 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2020 年 2 月 28 日

[Amazon RDS 績效詳情可供 Amazon RDS for MariaDB 10.3 版使用](#)

Amazon RDS 績效詳情現在可供 Amazon RDS for MariaDB 版本 10.3.13 和更高版本的 10.3 使用。如需詳細資訊，請參閱[使用 Amazon RDS 績效詳情](#)。

2020 年 2 月 26 日

支援 MySQL 5.7.28	您現在可以建立執行 MySQL 版本 5.7.28 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2020 年 2 月 20 日
支援 MariaDB 10.3.20	您現在可以建立執行 MariaDB 10.3.20 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2020 年 2 月 20 日
Amazon RDS for Microsoft SQL Server 支援新的資料庫執行個體類別	您現在可以建立執行 SQL Server 的 Amazon RDS 資料庫執行個體，其會採用 db.z1d 資料庫執行個體類別。如需詳細資訊，請參閱 Microsoft SQL Server 的資料庫執行個體類別支援 。	2020 年 2 月 19 日
支援 Amazon RDS for SQL Server 中的跨帳戶、跨 VPC Active Directory 網域	Amazon RDS for Microsoft SQL Server 現在支援將資料庫執行個體與不同帳戶和 VPC 擁有的 Active Directory 網域相關聯。如需詳細資訊，請參閱 使用 Windows 身分驗證搭配 SQL Server 資料庫執行個體 。	2020 年 2 月 13 日
Oracle OLAP 選項	Amazon RDS for Oracle 現在支援 Oracle 資料庫執行個體的線上分析處理 (OLAP) 選項。您可以使用 Oracle OLAP，依據 OLAP 標準建立維度物件和立方體來分析大量的資料。如需詳細資訊，請參閱 Oracle OLAP 。	2020 年 2 月 13 日

[適用於 Oracle 的 FIPS 140-2 支援](#)

Amazon RDS for Oracle 支援 SSL/TLS 連線的聯邦資訊處理標準發行版 140-2 (FIPS 140-2)。如需詳細資訊，請參閱 [FIPS 支援](#)。

2020 年 2 月 11 日

[Amazon RDS for PostgreSQL 支援新的資料庫執行個體類別](#)

您現在可以建立執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，這些執行個體使用 db.m5.16xlarge、db.m5.8xlarge、db.r5.16xlarge 及 db.r5.8xlarge 執行個體類別。如需詳細資訊，請參閱 [所有可用資料庫執行個體類別的支援資料庫引擎](#)。

2020 年 2 月 11 日

[績效詳情支援分析執行 MariaDB 與 MySQL 查詢的統計資料](#)

您現在可以使用績效詳情，分析對 MariaDB 與 MySQL 資料庫執行個體執行查詢的統計資料。如需詳細資訊，請參閱 [分析執行中查詢的統計資料](#)。

2020 年 2 月 4 日

[支援將資料庫快照資料匯出至適用於 MariaDB、MySQL 和 PostgreSQL 的 Amazon S3](#)

Amazon RDS 支援將資料庫快照資料匯出至適用於 MariaDB、MySQL 和 PostgreSQL 的 Amazon S3。如需詳細資訊，請參閱 [將資料庫快照資料匯出至 Amazon S3](#)。

2020 年 1 月 23 日

[Amazon RDS for MySQL 支援 Kerberos 身分驗證](#)

您現在可以使用 Kerberos 身分驗證來在使用者連線到您的 Amazon RDS for MySQL 資料庫執行個體時對其進行身分驗證。如需詳細資訊，請參閱 [針對 MySQL 使用 Kerberos 身分驗證](#)。

2020 年 1 月 21 日

[Amazon RDS 績效詳情支援
檢視更多 Amazon RDS for
Microsoft SQL Server 的 SQL
文字](#)

Amazon RDS 績效詳情現支援在績效詳情儀表中，檢視更多 Amazon RDS for Microsoft SQL Server 資料庫執行個體的 SQL 文字。如需詳細資訊，請參閱[在績效詳情儀表板內檢視更多 SQL 文字](#)。

2019 年 12 月 17 日

[Amazon RDS proxy](#)

您可以使用 Amazon RDS Proxy，減少叢集上連線管理的負擔，並減少發生「連線過多」錯誤的機會。您可以將每個代理與 RDS 資料庫執行個體或 Aurora 資料庫叢集建立關聯。然後，您可以在應用程式的連線字串中使用代理端點。Amazon RDS Proxy 目前處於公開預覽狀態。它支援 RDS for MySQL 資料庫引擎。如需詳細資訊，請參閱[使用 Amazon RDS Proxy 管理連線 \(預覽\)](#)。

2019 年 12 月 3 日

[Amazon RDS 上 AWS
Outposts \(預覽版\)](#)

使用 Amazon RDS on AWS Outposts，您可以在現場部署資料中心建立 AWS 受管的關聯式資料庫。RDS on Outposts 可讓您在 AWS Outposts 上執行 RDS 資料庫。如需更多詳細資訊，請參閱[Amazon RDS on AWS Outposts \(預覽\)](#)。

2019 年 12 月 3 日

[Amazon RDS for Oracle 支援
跨區域僅供讀取複本](#)

Amazon RDS for Oracle 現在支援含 Active Data Guard 的跨區域僅供讀取複本。如需詳細資訊，請參閱[使用僅供讀取複本](#)及[使用 Oracle 僅供讀取複本](#)。

2019 年 11 月 26 日

[「績效詳情」支援分析執行中 Oracle 查詢的統計資訊](#)

您現在可以使用績效詳情，對 Oracle 資料庫執行個體分析執行中查詢的統計資訊。如需詳細資訊，請參閱[分析執行中查詢的統計資料](#)。

2019 年 11 月 25 日

[Amazon RDS for Microsoft SQL Server 支持將日誌發布到 CloudWatch 日誌](#)

您可以將 Amazon RDS for SQL Server 資料庫執行個體設定為將日誌事件直接發佈到 Amazon CloudWatch 日誌。如需詳細資訊，請參閱將[SQL Server 日誌發佈到 Amazon CloudWatch 日誌](#)。

2019 年 11 月 25 日

[Amazon RDS for Microsoft SQL Server 支援新的資料庫執行個體類別](#)

您現在可以建立執行 SQL Server 的 Amazon RDS 資料庫執行個體，而且支援使用 db.x1e 和 db.x1 資料庫執行個體類別。如需詳細資訊，請參閱[Microsoft SQL Server 的資料庫執行個體類別支援](#)。

2019 年 11 月 25 日

[Amazon RDS for Microsoft SQL Server 支援差異和日誌還原](#)

您可以使用 SQL Server 原生備份與還原來還原差異備份和日誌。如需詳細資訊，請參閱[使用原生備份與還原](#)。

2019 年 11 月 25 日

[新的區域支援 Amazon RDS for Microsoft SQL Server 上的異地同步備份](#)

現在已開放在中國、中東 (巴林) 和歐洲 (斯德哥爾摩) 使用 SQL Server 上的異地同步備份。如需詳細資訊，請參閱[Microsoft SQL Server 的異地同步備份部署](#)。

2019 年 11 月 22 日

[Amazon RDS for Microsoft SQL Server 現在支援大量插入和 S3 整合](#)

您可以在 SQL Server 資料庫執行個體與 Amazon S3 儲存貯體之間傳輸檔案。此外，您可以搭配 SQL Server 功能來使用 Amazon S3，例如大量插入。如需詳細資訊，請參閱[整合 Amazon RDS for SQL Server 資料庫執行個體和 Amazon S3](#)。

2019 年 11 月 21 日

[Amazon RDS for Microsoft SQL Server 的績效詳情計數器](#)

您現在可以將效能計數器新增至 Microsoft SQL Server 資料庫執行個體的績效詳情圖表。如需詳細資訊，請參閱[Amazon RDS for Microsoft SQL Server 的績效詳情計數器](#)。

2019 年 11 月 12 日

[Amazon RDS for Microsoft SQL Server 支援新的資料庫執行個體類別大小](#)

您現在可以建立執行 SQL Server 的 Amazon RDS 資料庫執行個體，而且支援 db.m5 和 db.r5 資料庫執行個體類別使用 8xlarge 和 16xlarge 執行個體大小。db.t3 執行個體類別現在可以使用從 small 到 2xlarge 的執行個體大小。如需詳細資訊，請參閱[Microsoft SQL Server 的資料庫執行個體類別支援](#)。

2019 年 11 月 11 日

[支援 PostgreSQL 快照升級](#)

如果您的 Amazon RDS PostgreSQL 資料庫執行個體已有手動的資料庫快照，您現在可以將這些快照升級至更新版本的 PostgreSQL 資料庫引擎。如需詳細資訊，請參閱[升級 PostgreSQL 資料庫快照](#)。

2019 年 11 月 7 日

Amazon RDS for Oracle 支援新的主要版本	您現在可以建立執行 Oracle 資料庫 19c (19.0) 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Oracle 資料庫 19c 搭配 Amazon RDS 。	2019 年 11 月 7 日
資料庫預覽環境中的 Amazon RDS for PostgreSQL 版本 12.0	Amazon RDS for PostgreSQL 現在於資料庫預覽環境中支援 PostgreSQL 版本 12.0。如需詳細資訊，請參閱 資料庫預覽環境中的 PostgreSQL 版本 12.0 。	2019 年 11 月 1 日
Amazon RDS for PostgreSQL 支援 Kerberos 身分驗證	您現在可以使用 Kerberos 身分驗證，在使用者連線至執行 PostgreSQL 的 Amazon RDS 資料庫執行個體時加以驗證。如需詳細資訊，請參閱 搭配 Amazon RDS for PostgreSQL 使用 Kerberos 身分驗證 。	2019 年 10 月 28 日
Oracle 資料庫執行個體的 OEM Management Agent 資料庫任務	Amazon RDS for Oracle 資料庫執行個體現在支援程序在 Management Agent 上呼叫特定 EMCTL 指令。如需詳細資訊，請參閱 OEM Agent 資料庫任務 。	2019 年 10 月 24 日
Amazon RDS for PostgreSQL 支援 PostgreSQL Transportable Databases	PostgreSQL Transportable Databases 提供非常快速的方式，合併 RDS PostgreSQL 資料庫和兩個資料庫執行個體之間。如需詳細資訊，請參閱 在資料庫執行個體之間轉換 PostgreSQL 資料庫 。	2019 年 10 月 8 日

Amazon RDS for Oracle 支援 Kerberos 身分驗證	您現在可以使用 Kerberos 身分驗證，在使用者連線至執行 Oracle 的 Amazon RDS 資料庫執行個體時加以驗證。如需詳細資訊，請參閱 搭配 Amazon RDS for Oracle 使用 Kerberos 身分驗證 。	2019 年 9 月 30 日
資料庫預覽環境中 Amazon RDS for PostgreSQL 版本 12 Beta 3	Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 12 Beta 3。如需詳細資訊，請參閱 在資料庫預覽環境中 Amazon RDS 上的 PostgreSQL 版本 12 Beta 3 。	2019 年 8 月 28 日
支援 MySQL 8.0.16	您現在可以建立執行 MySQL 版本 8.0.16 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2019 年 8 月 19 日
Amazon RDS for Oracle 支援新的主要版本	您現在可以建立執行 Oracle 資料庫 18c (18.0) 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Oracle 資料庫 18c 搭配 Amazon RDS 。	2019 年 8 月 15 日
OEM 13c 第 3 版適用的 Management Agent	Amazon RDS for Oracle 資料庫執行個體現在支援 Oracle Enterprise Manager (OEM) Cloud Control 13c 第 3 版適用的 Management Agent。如需詳細資訊，請參閱 適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent 。	2019 年 8 月 7 日

[資料庫預覽環境中 Amazon RDS for PostgreSQL 版本 12 Beta 2](#)

Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 12 Beta 2。如需詳細資訊，請參閱[在資料庫預覽環境中 Amazon RDS 上的 PostgreSQL 版本 12 Beta 2](#)。

2019 年 8 月 6 日

[Amazon RDS 支援 SQL Server 的伺服器定序](#)

Amazon RDS for SQL Server 支援新資料庫執行個體的定序選項。如需詳細資訊，請參閱[Microsoft SQL Server 的定序和字元集](#)。

2019 年 7 月 29 日

[Amazon RDS for Oracle 支援 Oracle APEX 19.1.v1 版](#)

Amazon RDS for Oracle 現支援 Oracle Application Express (APEX) 19.1.v1 版。如需詳細資訊，請參閱[Oracle Application Express](#)。

2019 年 6 月 28 日

[資料庫預覽環境中 Amazon RDS for PostgreSQL 版本 13 Beta 1](#)

Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 13 Beta 1。如需詳細資訊，請參閱[PostgreSQL 13 版](#)。

2019 年 6 月 22 日

[Amazon RDS 儲存體自動調整規模](#)

Amazon RDS 資料庫執行個體的儲存自動調度資源可讓 Amazon RDS 自動擴充與資料庫執行個體相關聯的儲存，以減少發生 out-of-space 情況的機會。如需儲存體自動調整規模的資訊，請參閱[使用 Amazon RDS 資料庫執行個體的儲存體](#)。

2019 年 6 月 20 日

Amazon RDS for Oracle 支援 db.z1d 資料庫執行個體類別	您現在可以建立執行 Oracle 的 Amazon RDS 資料庫執行個體，其會採用 db.z1d 資料庫執行個體類別。如需詳細資訊，請參閱 資料庫執行個體類別 。	2019 年 6 月 13 日
Amazon RDS 績效詳情支援 Amazon RDS for Oracle 檢視更多的 SQL 文字	Amazon RDS 績效詳情現支援在績效詳情儀表中，Amazon RDS for Oracle 資料庫執行個體檢視更多的 SQL 文字。如需詳細資訊，請參閱 在績效詳情儀表板內檢視更多 SQL 文字 。	2019 年 6 月 10 日
Amazon RDS 新增對高達 16 TB SQL Server 資料庫原生還原功能的支援	您現在可以將高達 16 TB 的資料從 SQL Server 原生還原至 Amazon RDS。如需詳細資訊，請參閱 Amazon RDS for SQL Server：限制與建議 。	2019 年 6 月 4 日
Amazon RDS 新增對 Microsoft SQL Server 稽核機制的支援	使用 Amazon RDS for Microsoft SQL Server，您可以使用 SQL Server Audit 稽核伺服器與資料庫事件等級，並且在資料庫執行個體中檢視結果或直接送出稽核日誌檔案到 Amazon S3。如需詳細資訊，請參閱 SQL Server Audit 。	2019 年 5 月 23 日
改善 Amazon RDS 建議	Amazon RDS 已改善自動化的資料庫資源建議。例如，Amazon RDS 現提供資料庫參數建議。如需詳細資訊，請參閱 使用 Amazon RDS 建議 。	2019 年 5 月 22 日

[Amazon RDS for SQL Server 每個資料庫執行個體支援更多資料庫](#)

您可以在執行 Microsoft SQL Server 的每一個資料庫執行個體上最多建立 30 個資料庫。如需詳細資訊，請參閱 [Microsoft SQL Server 的資料庫執行個體限制](#)。

2019 年 5 月 21 日

[對 Amazon RDS for MariaDB、MySQL 和 PostgreSQL 支援 64 TiB 和 80k IOPS 儲存體](#)

您現在可以為 MariaDB、MySQL 和 PostgreSQL 建立具有最高 64 TiB 儲存體和佈建最高可到 80,000 IOPS 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [資料庫執行個體儲存體](#)。

2019 年 5 月 20 日

[Amazon RDS for MySQL 支援升級前檢查](#)

當您資料庫執行個體從 MySQL 5.7 升級到 MySQL 8.0 時，Amazon RDS 執行不相容性前置檢查。如需詳細資訊，請參閱 [從 MySQL 5.7 升級到 8.0 的前置檢查](#)。

2019 年 5 月 17 日

[支援 MySQL 密碼驗證外掛程式](#)

您現在可使用 MySQL validate_password 外掛程式加強 Amazon RDS for MySQL 資料庫執行個體的安全性。如需詳細資訊，請參閱 [使用密碼驗證外掛程式](#)。

2019 年 5 月 16 日

[Amazon RDS for Oracle 的績效詳情計數器](#)

您現在可以將效能計數器新增至您 Oracle 資料庫執行個體的績效詳情圖表。如需詳細資訊，請參閱 [Amazon RDS for Oracle 的績效詳情計數器](#)。

2019 年 5 月 8 日

[支援以秒計費](#)

Amazon RDS 現在在所有 AWS 區域 AWS GovCloud (美國) 的隨需執行個體以 1 秒增量計費。如需詳細資訊，請參閱 [Amazon RDS 資料庫執行個體計費](#)。

2019 年 4 月 25 日

[Amazon RDS for PostgreSQL 支援從 Amazon S3 匯入資料](#)

您現在可以從 Amazon S3 檔案匯入資料到 RDS PostgreSQL 資料庫執行個體的表格。如需詳細資訊，請參閱 [將 Amazon S3 資料匯入 RDS for PostgreSQL 資料庫執行個體](#)。

2019 年 4 月 24 日

[支援從 Amazon S3 還原 5.7 備份](#)

現在，您能夠建立現場部署資料庫的備份，並將其存放在 Amazon S3，接著將備份檔案還原至執行 MySQL 版本 5.7 的新 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 [將備份還原至 MySQL 資料庫執行個體](#)。

2019 年 4 月 17 日

[支援 Amazon RDS for PostgreSQL 的主要版本升級](#)

透過 Amazon RDS for PostgreSQL，當您升級資料庫引擎時，您現在可以從多個主要版本選擇。當您升級選取的 PostgreSQL 引擎版本時，此功能可以讓您快速前進到較新的主要版本。如需詳細資訊，請參閱 [升級 PostgreSQL 資料庫引擎](#)。

2019 年 4 月 16 日

[Amazon RDS for Oracle 支援 64 TiB 儲存體](#)

您現在可以為 Oracle 建立具有最高 64 TiB 儲存體和佈建最高可到 80,000 IOPS 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[資料庫執行個體儲存體](#)。

2019 年 4 月 4 日

[支援 MySQL 8.0.15](#)

您現在可以建立執行 MySQL 版本 8.0.15 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2019 年 4 月 3 日

[支援 MariaDB 10.3.13](#)

您現在可以建立執行 MariaDB 版本 10.3.13 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MariaDB 版本](#)。

2019 年 4 月 3 日

[Microsoft SQL Server 2008 R2 已經終止在 Amazon RDS 上的 支援](#)

Microsoft SQL Server 2008 R2 已經終止支援，此版本相符的 Microsoft 支援計畫延長到 2019 年 7 月 9 日終止。任何現存於 Microsoft SQL Server 2008 R2 上的快照，將於 2019 年 6 月 1 日自動升級到 Microsoft SQL Server 2012 最新的次要版本。如需詳細資訊，請參閱[Microsoft SQL Server 2008 R2 支援 Amazon RDS](#)。

2019 年 4 月 2 日

Microsoft SQL Server 2017 支援的 Always on 可用性群組	您現在可以在 Server 2017 Enterprise Edition 14.00.304 9.1 或之後的版本中使用 Always on 可用性群組。如需詳細資訊，請參閱 Microsoft SQL Server 的異地同步備份部署 。	2019 年 3 月 29 日
檢視磁碟區指標	您現在可以檢視 Amazon Elastic Block Store (Amazon EBS) 的指標，這是用於資料庫及日誌儲存的實體裝置。如需詳細資訊，請參閱 檢視增強型監控 。	2019 年 3 月 20 日
支援 MySQL 5.7.25	您現在可以建立執行 MySQL 5.7.25 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2019 年 3 月 19 日
Amazon RDS for Oracle 支援 RMAN DBA 任務	Amazon RDS for Oracle 現在支援 Oracle 修復管理 (RMAN) DBA 任務，包括 RMAN 備份。如需詳細資訊，請參閱 Oracle 資料庫執行個體常見 DBA 修復管理任務 。	2019 年 3 月 14 日
Amazon RDS for PostgreSQL 支援 11.1 版本	您現在可以建立執行 PostgreSQL 11.1 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 PostgreSQL 11.1 版 。	2019 年 3 月 12 日

多個檔案還原適用於 Amazon RDS for SQL Server	您現在可以使用 Amazon RDS for SQL Server 從多個檔案中還原。如需詳細資訊，請參閱 還原資料庫 。	2019 年 3 月 11 日
MariaDB 10.2.21	您現在可以建立執行 MariaDB 10.2.21 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2019 年 3 月 11 日
Amazon RDS for Oracle 支援僅供讀取複本	Amazon RDS for Oracle 現在支援含 Active Data Guard 的僅供讀取複本。如需詳細資訊，請參閱 使用僅供讀取複本及使用 Oracle 僅供讀取複本 。	2019 年 3 月 11 日
Amazon RDS 績效詳情可供 Amazon RDS for MariaDB 使用	Amazon RDS 績效詳情現可供 Amazon RDS for MariaDB 使用。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2019 年 3 月 11 日
MySQL 8.0.13 和 5.7.24	您現在可以建立執行 MySQL 8.0.13 版、和 5.7.24 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2019 年 3 月 8 日
Amazon RDS 績效詳情可供 Amazon RDS for SQL Server 使用	Amazon RDS 績效詳情現在可供 Amazon RDS for SQL Server 使用。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2019 年 3 月 4 日

[Amazon RDS for Oracle 支援 Amazon S3 整合](#)

您現在可以在 Amazon RDS for Oracle 資料庫執行個體與 Amazon S3 儲存貯體之間傳輸檔案。如需詳細資訊，請參閱[整合 Amazon RDS for Oracle 與 Amazon S3](#)。

2019 年 2 月 26 日

[Amazon RDS for MySQL 和 Amazon RDS for MariaDB 可支援 db.t3 資料庫執行個體類別](#)

您現在可以建立執行 MySQL 或 MariaDB 的 Amazon RDS 資料庫執行個體，其會採用 db.t3 資料庫執行個體類別。如需詳細資訊，請參閱[資料庫執行個體類別](#)。

2019 年 2 月 20 日

[Amazon RDS for MySQL 和 Amazon RDS for MariaDB 可支援 db.r5 資料庫執行個體類別](#)

您現在可以建立執行 MySQL 或 MariaDB 的 Amazon RDS 資料庫執行個體，其會採用 db.r5 資料庫執行個體類別。如需詳細資訊，請參閱[資料庫執行個體類別](#)。

2019 年 2 月 20 日

[RDS for MySQL 和 PostgreSQL 的績效詳情計數器](#)

您現在可以將效能計數器新增至您的 MySQL 和 PostgreSQL 資料庫執行個體的績效詳情圖表。如需詳細資訊，請參閱[績效詳情儀表板元件](#)。

2019 年 2 月 19 日

[Amazon RDS for PostgreSQL 現在支援彈性自動資料清理參數調整](#)

使用 Amazon RDS for PostgreSQL 彈性自動資料清理參數調整，透過自動調整自動資料清理參數的值來協助防止概括交易 ID。如需詳細資訊，請參閱[降低交易 ID 包圍的可能性](#)。

2019 年 2 月 12 日

[Amazon RDS for Oracle 支援 Oracle APEX 18.1.v1 及 18.2.v1 版](#)

Amazon RDS for Oracle 現在支援 Oracle Application Express (APEX) 18.1.v1 及 18.2.v1 版。如需詳細資訊，請參閱 [Oracle Application Express](#)。

2019 年 2 月 11 日

[Amazon RDS 績效詳情支援 RDS for MySQL 檢視更多的 SQL 文字](#)

Amazon RDS 績效詳情現在支援績效詳情儀表中，MySQL 資料庫執行個體檢視更多的 SQL 文字。如需詳細資訊，請參閱 [在績效詳情儀表板內檢視更多 SQL 文字](#)。

2019 年 2 月 6 日

[Amazon RDS for PostgreSQL 支援 db.t3 資料庫執行個體類別](#)

您現在可以建立執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，其會採用 db.t3 資料庫執行個體類別。如需詳細資訊，請參閱 [資料庫執行個體類別](#)。

2019 年 1 月 25 日

[Amazon RDS for Oracle 支援 db.t3 資料庫執行個體類別](#)

您現在可以建立執行 Oracle 的 Amazon RDS 資料庫執行個體，其會採用 db.t3 資料庫執行個體類別。如需詳細資訊，請參閱 [資料庫執行個體類別](#)。

2019 年 1 月 25 日

[Amazon RDS 績效詳情支援 Amazon RDS PostgreSQL 檢視更多的 SQL 文字](#)

Amazon RDS 績效詳情現在支援績效詳情儀表中，Amazon RDS PostgreSQL 資料庫執行個體檢視更多的 SQL 文字。如需詳細資訊，請參閱 [在績效詳情儀表板內檢視更多 SQL 文字](#)。

2019 年 1 月 24 日

Amazon RDS for Oracle 支援新版 SQLT	Amazon RDS for Oracle 現在支援 SQLT 12.2.180725 版。如需詳細資訊，請參閱 Oracle SQLT 。	2019 年 1 月 22 日
Amazon RDS for PostgreSQL 支援 db.r5 資料庫執行個體類別	您現在可以建立執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，其會採用 db.r5 資料庫執行個體類別。如需詳細資訊，請參閱 資料庫執行個體類別 。	2018 年 12 月 19 日
Amazon RDS for PostgreSQL 現在支援限制的密碼管理	Amazon RDS for PostgreSQL 可讓您使用參數 <code>rds_restrict_password_commands</code> 和角色 <code>rds_password</code> 來限制誰可以管理使用者密碼和密碼到期的變更。如需詳細資訊，請參閱 限制密碼管理 。	2018 年 12 月 19 日
Amazon RDS for PostgreSQL 支援將資料庫日誌上傳到 Amazon 日誌 CloudWatch	亞馬遜 RDS 支援將資料庫日誌上傳到 CloudWatch 日誌。如需詳細資訊，請參閱將 PostgreSQL 記錄檔發佈至 CloudWatch 記錄檔 。	2018 年 12 月 10 日
Amazon RDS for Oracle 支援 db.r5 資料庫執行個體類別	您現在可以建立執行 Oracle 的 Amazon RDS 資料庫執行個體，其會採用 db.r5 資料庫執行個體類別。如需詳細資訊，請參閱 資料庫執行個體類別 。	2018 年 11 月 20 日
刪除資料庫執行個體時保留備份	Amazon RDS 支援在刪除資料庫執行個體時保留自動備份。如需詳細資訊，請參閱 使用備份 。	2018 年 11 月 15 日

Amazon RDS for PostgreSQL 支援 db.m5 資料庫執行個體類別	您現在可以建立執行 PostgreSQL 的 Amazon RDS 資料庫執行個體，其會採用 db.m5 資料庫執行個體類別。如需詳細資訊，請參閱 資料庫執行個體類別 。	2018 年 11 月 15 日
Amazon RDS for Oracle 支援新的主要版本	您現在可以建立執行 Oracle 12.2 版的 Amazon RDS 資料庫執行個體。	2018 年 11 月 13 日
Amazon RDS for SQL Server 支援 Always On	Amazon RDS for SQL Server 支援 Always On 可用性群組。如需詳細資訊，請參閱 Microsoft SQL Server 的異地同步備份部署 。	2018 年 11 月 8 日
Amazon RDS for PostgreSQL 支援使用自訂 DNS 伺服器進行傳出網路存取	Amazon RDS for PostgreSQL 支援使用自訂 DNS 伺服器進行傳出網路存取。如需詳細資訊，請參閱 對傳出網路存取使用自訂 DNS 伺服器 。	2018 年 11 月 8 日
Amazon RDS for MariaDB、MySQL 和 PostgreSQL 支援 32 TiB 的儲存體	您現在可以為 MySQL、MariaDB 和 PostgreSQL 建立具有最高 32 TiB 儲存體的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 資料庫執行個體儲存體 。	2018 年 11 月 7 日

[Amazon RDS for Oracle 支援擴充資料類型](#)

您現在可以對於執行 Oracle 的 Amazon RDS 資料庫執行個體上啟用擴充資料類型。使用擴充資料類型後，VARCHAR2、NVARCHAR2 和 RAW 的大小上限是 32,767 位元組。如需詳細資訊，請參閱[使用擴充資料類型](#)。

2018 年 11 月 6 日

[Amazon RDS for Oracle 支援 db.m5 資料庫執行個體類別](#)

您現在可以建立執行 Oracle 的 Amazon RDS 資料庫執行個體，其會採用 db.m5 資料庫執行個體類別。如需詳細資訊，請參閱[資料庫執行個體類別](#)。

2018 年 11 月 2 日

[從 SE、SE1 或 SE2 到 EE 的 Amazon RDS for Oracle 遷移](#)

您現在可以從任何 Oracle Database Standard Edition (SE、SE1 或 SE2) 遷移到 Oracle Database Enterprise Edition (EE)。如需詳細資訊，請參閱[在 Oracle 版本之間遷移](#)。

2018 年 10 月 31 日

[Amazon RDS 現在可停止多個可用區執行個體](#)

Amazon RDS 現在可停止屬於異地同步備份部署中的資料庫執行個體。過去，停止執行個體功能有多個可用區執行個體的限制。如需詳細資訊，請參閱[暫時停止 Amazon RDS 資料庫執行個體](#)。

2018 年 10 月 29 日

[Amazon RDS 績效詳情可供 Amazon RDS for Oracle 使用](#)

Amazon RDS 績效詳情現在可供 Amazon RDS for Oracle 使用。如需詳細資訊，請參閱[使用 Amazon RDS 績效詳情](#)。

2018 年 10 月 29 日

Amazon RDS for PostgreSQL 支援資料庫預覽環境中的 PostgreSQL 版本 11	Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 11。如需詳細資訊，請參閱 在資料庫預覽環境中 Amazon RDS 上的 PostgreSQL 版本 11 。	2018 年 10 月 25 日
MySQL 支援新的主要版本	您現在可以建立執行 MySQL 版本 8.0 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2018 年 10 月 23 日
MariaDB 支援新的主要版本	您現在可以建立執行 MariaDB 版本 10.3 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2018 年 10 月 23 日
Amazon RDS for Oracle 支援 Oracle JVM	Amazon RDS for Oracle 現已支援 Oracle Java Virtual Machine (JVM) 選項。如需詳細資訊，請參閱 Oracle Java 虛擬機器 。	2018 年 10 月 16 日
用於還原和時間點復原的自訂參數群組	當還原快照或執行時間點復原操作時，您現在可以指定自訂參數群組。如需詳細資訊，請參閱 從資料庫快照還原 和 將資料庫執行個體還原至指定的時間 。	2018 年 10 月 15 日
Amazon RDS for Oracle 支援 32 TiB 儲存體	您現在可以建立具有最高 32 TiB 儲存體的 Oracle RDS 資料庫執行個體。如需詳細資訊，請參閱 資料庫執行個體儲存體 。	2018 年 10 月 15 日

Amazon RDS for MySQL 支援 GTID	Amazon RDS for MySQL 現在支援全域交易識別符 (GTID)，這是對於所有資料庫執行個體和複寫組態中的唯一識別符。如需詳細資訊，請參閱 使用 RDS for MySQL 的 GTID 式複寫 。	2018 年 10 月 10 日
MySQL 5.7.23、5.6.41 和 5.5.61	您現在可以建立執行 MySQL 版本 5.7.23、5.6.41 和 5.5.61 的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MySQL 版本 。	2018 年 10 月 8 日
Amazon RDS for Oracle 支援新版 SQLT	Amazon RDS for Oracle 現在支援 SQLT 12.2.180331 版。如需詳細資訊，請參閱 Oracle SQLT 。	2018 年 10 月 4 日
Amazon RDS for PostgreSQL 現已支援 IAM 身分驗證	Amazon RDS for PostgreSQL 現已支援 IAM 身分驗證。如需詳細資訊，請參閱 MySQL 和 PostgreSQL 的 IAM 資料庫身分驗證 。	2018 年 9 月 27 日
您可以針對 Amazon RDS 資料庫執行個體啟用刪除保護	當您針對資料庫執行個體啟用刪除保護時，任何使用者皆無法刪除資料庫。如需詳細資訊，請參閱 刪除資料庫執行個體 。	2018 年 9 月 26 日

[Amazon RDS for MySQL 和 Amazon RDS for MariaDB 可支援 db.m5 資料庫執行個體類別](#)

您現在可以建立執行 MySQL 或 MariaDB 的 Amazon RDS 資料庫執行個體，其會採用 db.m5 資料庫執行個體類別。如需詳細資訊，請參閱[資料庫執行個體類別](#)。

2018 年 9 月 18 日

[Amazon RDS 現可支援升級至 SQL Server 2017](#)

除了 SQL Server 2008 之外，您可以將任意版本的現有資料庫執行個體升級至 SQL Server 2017。若要升級 SQL Server 2008 的資料庫執行個體，請先將其升級為其他版本。如需詳細資訊，請參閱[升級 Microsoft SQL Server 資料庫引擎](#)。

2018 年 9 月 11 日

[Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 11 Beta 3](#)

在此版本中，預寫記錄檔 (WAL) 的區段大小 (wal_segment_size) 現已設定為 64 MB。如需 PostgreSQL 版本 11 Beta 3 的詳細資訊，請參閱 [PostgreSQL 11 Beta 3 Released \(PostgreSQL 版本 11 Beta 3 已發行\)](#)。如需資料庫預覽環境的相關資訊，請參閱[使用資料庫預覽環境](#)。

2018 年 9 月 7 日

[Amazon Aurora 使用者指南](#)

《[Amazon Aurora 使用者指南](#)》說明 Amazon Aurora 的所有概念，並提供透過主控台與命令列界面使用各種功能的指示。Amazon RDS 使用者指南現已涵蓋非 Aurora 資料庫引擎的使用方式。

2018 年 8 月 31 日

[Amazon RDS 績效詳情可供 RDS for MySQL 使用](#)

Amazon RDS 績效詳情現可供 RDS for MySQL 使用。如需詳細資訊，請參閱[使用 Amazon RDS 績效詳情](#)。

2018 年 8 月 28 日

[Aurora PostgreSQL 相容版本現可支援 Aurora Auto Scaling](#)

Aurora PostgreSQL 相容版本現能使用 Aurora 複本的 Auto Scaling 功能。如需詳細資訊，請參閱[使用 Amazon Aurora Auto Scaling 搭配 Aurora 複本](#)。

2018 年 8 月 16 日

[適用於 Aurora MySQL 的 Aurora Serverless](#)

Aurora Serverless 是 Amazon Aurora 的隨需、自動擴展組態。如需詳細資訊，請參閱[使用 Amazon Aurora Serverless](#)。

2018 年 8 月 9 日

[MySQL 5.7.22 和 5.6.40](#)

您現在可以建立執行 MySQL 5.7.22 版和 5.6.40 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱[Amazon RDS 上的 MySQL 版本](#)。

2018 年 8 月 6 日

[中國 \(寧夏\) 區域現在可以使用 Aurora](#)

中國 (寧夏) 區域現能使用 Aurora MySQL 與 Aurora PostgreSQL。如需詳細資訊，請參閱[Amazon Aurora MySQL 的可用性](#)以及[Amazon Aurora PostgreSQL 的可用性](#)。

2018 年 8 月 6 日

Amazon RDS for MySQL 可支援延遲複寫	Amazon RDS for MySQL 現可支援延遲複寫，做為災難復原的策略。如需詳細資訊，請參閱 透過 MySQL 設定延遲複寫 。	2018 年 8 月 6 日
Amazon RDS Performance Insights 可供 Aurora MySQL 使用	Aurora MySQL 現能使用 Amazon RDS Performance Insights。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2018 年 8 月 6 日
Amazon RDS Performance Insights 與 Amazon 的整合 CloudWatch	Amazon RDS Performance Insights 會自動將指標發佈到 Amazon CloudWatch。如需詳細資訊，請參閱 發佈至 CloudWatch 。	2018 年 8 月 6 日
Amazon RDS 建議	Amazon RDS 現提供自動化的資料庫資源建議。如需詳細資訊，請參閱 使用 Amazon RDS 建議 。	2018 年 7 月 25 日
跨 AWS 區域的增量快照複製	Amazon RDS 支援未加密和加密執行個體跨 AWS 區域的增量快照複製。如需詳細資訊，請參閱 跨 AWS 區域複製快照 。	2018 年 7 月 24 日
Amazon RDS Performance Insights 可供 Amazon RDS for PostgreSQL 使用	Amazon RDS for PostgreSQL 現能使用 Amazon RDS Performance Insights。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2018 年 7 月 18 日

Amazon RDS for Oracle 支援 Oracle APEX 5.1.4.v1 版	Amazon RDS for Oracle 現支援 Oracle Application Express (APEX) 5.1.4.v1 版。如需詳細資訊，請參閱 Oracle Application Express 。	2018 年 7 月 10 日
Amazon RDS for Oracle 支持將日誌發佈到 Amazon CloudWatch 日誌	Amazon RDS for Oracle 現在支援將警示、稽核、追蹤和接聽程式日誌資料發佈到 CloudWatch 日誌中的日誌群組。如需詳細資訊，請參閱將 Oracle 日誌發佈到 Amazon CloudWatch 日誌 。	2018 年 7 月 9 日
MariaDB 10.2.15、10.1.34 和 10.0.35	您現在可以建立執行 MariaDB 10.2.15 版、10.1.34 版和 10.0.35 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2018 年 7 月 5 日
Aurora PostgreSQL 1.2 現已提供並相容於 PostgreSQL 9.6.8	Aurora PostgreSQL 1.2 現可供使用並相容於 PostgreSQL 9.6.8 如需詳細資訊，請參閱 1.2 版 。	2018 年 27 月 6 日
Amazon RDS PostgreSQL 僅供讀取複本支援異地同步備份部署	Amazon RDS PostgreSQL 中的 RDS 僅供讀取複本現在支援多個可用區域。如需詳細資訊，請參閱 使用 PostgreSQL 僅供讀取複本 。	2018 年 6 月 25 日
Performance Insights 可供 Aurora PostgreSQL 使用	Performance Insights 已普遍供 Aurora PostgreSQL 使用，並支援效能資料的延長保留。如需詳細資訊，請參閱 使用 Amazon RDS 績效詳情 。	2018 年 6 月 21 日

[美國西部 \(加州北部\) 區域可以使用 Aurora PostgreSQL](#)

Aurora PostgreSQL 現可供美國西部 (加州北部) 區域使用。如需詳細資訊，請參閱 [Amazon Aurora PostgreSQL 的可用性](#)。

2018 年 6 月 11 日

[Amazon RDS for Oracle 現在支援 CPU 組態](#)

Amazon RDS for Oracle 現支援 CPU 核心數量的設定，以及每個核心供資料庫執行個體類別處理器使用的執行緒數量設定。如需詳細資訊，請參閱 [設定資料庫執行個體類別的處理器](#)。

2018 年 6 月 5 日

舊版更新

下表會說明 2018 年 6 月前，Amazon RDS 使用者指南每個版本的重要變更。

變更	描述	變更日期
Amazon RDS for PostgreSQL 現在支援資料庫預覽環境中的 PostgreSQL 版本 11 Beta 1	PostgreSQL 版本 11 Beta 1 包含 PostgreSQL 11 Beta 1 已發行 中所述的數個改善項目。 如需資料庫預覽環境的資訊，請參閱 使用資料庫預覽環境 。	2018 年 5 月 31 日
Amazon RDS for Oracle 現在支援 TLS 版本 1.0 和 1.2	Amazon RDS for Oracle 現已支援 Transport Layer Security (TLS) 版本 1.0 和 1.2。如需更多詳細資訊，請參閱 Oracle SSL 選項的 TLS 版本 。	2018 年 5 月 30 日
Aurora MySQL 支援將日誌發佈到 Amazon CloudWatch 日誌	Aurora MySQL 現在支援將一般、緩慢、稽核和錯誤記錄資料發佈到 CloudWatch 記錄檔中的記錄群組。如需詳細資訊，請參閱將 Aurora MySQL 發佈至 CloudWatch 記錄檔 。	2018 年 5 月 23 日

變更	描述	變更日期
Amazon RDS PostgreSQL 的資料庫預覽環境	您現在可以在預覽模式中啟動 Amazon RDS PostgreSQL 的新執行個體。如需資料庫預覽環境的詳細資訊，請參閱 使用資料庫預覽環境 。	2018 年 5 月 22 日
Amazon RDS for Oracle 資料庫執行個體支援新的資料庫執行個體類別	Oracle 資料庫執行個體現在支援 db.x1e 和 db.x1 資料庫執行個體類別。如需更多詳細資訊，請參閱 資料庫執行個體類別 及 RDS for Oracle 執行個體類別 。	2018 年 5 月 22 日
Amazon RDS PostgreSQL 現在於僅供讀取複本上支援 postgres_fdw。	您現在可以使用 postgres_fdw 來從僅供讀取複本連線至遠端伺服器。如需更多詳細資訊，請參閱 使用 postgres_fdw 擴充功能存取外部資料 。	2018 年 5 月 17 日
Amazon RDS for Oracle 現在支援設定 sqlnet.ora 參數	您現在可以使用 Amazon RDS for Oracle 來設定 sqlnet.ora 參數。如需更多詳細資訊，請參閱 使用 sqlnet.ora 參數修改連線屬性 。	2018 年 5 月 10 日
Aurora PostgreSQL 已在亞太 (首爾) 區域提供。	Aurora PostgreSQL 現已在亞太 (首爾) 區域提供。如需詳細資訊，請參閱 Amazon Aurora PostgreSQL 的可用性 。	2018 年 5 月 9 日
Aurora MySQL 支援恢復	Aurora MySQL 現在支援將資料庫叢集「倒轉」至特定時間，而不需從備份還原資料。如需詳細資訊，請參閱 恢復 Aurora 資料庫叢集 。	2018 年 5 月 9 日
Aurora MySQL 支援來自外部 MySQL 的加密遷移和複寫	Aurora MySQL 現在支援來自外部 MySQL 資料庫的加密遷移和複寫。如需詳細資訊，請參閱 從外部 MySQL 資料庫將資料遷移至 Amazon Aurora MySQL 資料庫叢集 ，以及在 Aurora 和 MySQL 間或 Aurora 和另一個 Aurora 資料庫叢集間進行複寫 。	2018 年 4 月 25 日

變更	描述	變更日期
Aurora PostgreSQL 相容版本 支援寫入時複製通訊協定。	您現在可以在 Aurora PostgreSQL 資料庫叢集中複製資料庫。如需詳細資訊，請參閱 複製 Aurora 資料庫叢集中的資料庫 。	2018 年 4 月 10 日
MariaDB 10.2.12、10.1.31 和 10.0.34	您現在可以建立執行 MariaDB 10.2.12 版、10.1.31 版和 10.0.34 版的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2018 年 3 月 21 日
Aurora PostgreSQL 支援新的區域	Aurora PostgreSQL 現已在歐洲 (倫敦) 和亞太 (新加坡) 區域提供。如需詳細資訊，請參閱 Amazon Aurora PostgreSQL 的可用性 。	2018 年 3 月 13 日
MySQL 5.7.21、5.6.39 和 5.5.59	您現在可以建立執行 MySQL 5.7.21 版、5.6.39 版和 5.5.59 版的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 MySQL on Amazon RDS 版本 。	2018 年 3 月 9 日
Amazon RDS for Oracle 現在支援 Oracle REST Data Services	Amazon RDS for Oracle 隨著 APEX 選項支援 Oracle REST Data Services。如需更多詳細資訊，請參閱 Oracle Application Express (APEX) 。	2018 年 3 月 9 日
Amazon Aurora 與 MySQL 相容的版本已在新區域推出 AWS	Aurora MySQL 現已在亞太 (新加坡) 區域提供。如需 Aurora MySQL AWS 區域的完整清單，請參閱 Amazon Aurora MySQL 的可用性 。	2018 年 3 月 6 日
執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體可支援變更資料擷取 (CDC)	執行 Amazon RDS for Microsoft SQL Server 的資料庫執行個體現在可支援變更資料擷取 (CDC)。如需更多詳細資訊，請參閱 Microsoft SQL Server 資料庫執行個體的變更資料擷取支援 。	2018 年 2 月 6 日

變更	描述	變更日期
Aurora MySQL 支援新的主要版本	您現在可以建立執行 MySQL 5.7 版的 Aurora MySQL 資料庫叢集。如需詳細資訊，請參閱 Amazon Aurora MySQL 資料庫引擎更新 2018-02-06 。	2018 年 2 月 6 日
將 MySQL 和 MariaDB 日誌發佈到 Amazon 日誌 CloudWatch	現在，您可以將 MySQL 和 MariaDB 的日誌數據發佈到 CloudWatch 日誌中。如需更多詳細資訊，請參閱 將 MySQL 日誌發佈到 Amazon CloudWatch 日誌 及 將 MariaDB 日誌發佈到 Amazon 日誌 CloudWatch 。	2018 年 1 月 17 日
僅供讀取複本的異地同步備份支援	您現在可以建立僅供讀取複本並用作多可用區域資料庫執行個體。Amazon RDS 會在另一個可用區域中建立您的複本的待命複本，以獲得該複本的容錯移轉支援。建立您的僅供讀取複本做為異地同步備份部署資料庫執行個體，與來源資料庫是否為異地同步備份部署資料庫執行個體無關。如需更多詳細資訊，請參閱 使用資料庫執行個體僅供讀取複本 。	2018 年 1 月 11 日
Amazon RDS for MariaDB 支援新的主要版本	您現在可以建立執行 MariaDB 10.2 版的 Amazon RDS 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 上的 MariaDB 10.2 支援。	2018 年 1 月 3 日
Amazon Aurora PostgreSQL 相容版本 已在新的 AWS 區域提供	Aurora PostgreSQL 現已在歐洲 (巴黎) 區域提供。如需 Aurora 的完整 AWS 區域清單，請參閱 Amazon Aurora PostgreSQL 的可用性 。	2017 年 12 月 22 日
Aurora PostgreSQL 可支援新的執行個體類型	Aurora PostgreSQL 現在支援新的執行個體類型。如需執行個體類型的完整清單，請參閱 選擇資料庫執行個體類別 。	2017 年 12 月 20 日
Amazon Aurora 與 MySQL 相容的版本 已在新區域推出 AWS	Aurora MySQL 現已在歐洲 (巴黎) 區域提供。如需 Aurora MySQL AWS 區域的完整清單，請參閱 Amazon Aurora MySQL 的可用性 。	2017 年 12 月 18 日

變更	描述	變更日期
Aurora MySQL 支援雜湊聯結	當您需要透過使用對等聯結來聯結大量資料時，此功能可改善查詢效能。如需詳細資訊，請參閱 在 Aurora MySQL 中使用雜湊聯結 。	2017 年 12 月 11 日
Aurora MySQL 可支援原生函數以呼叫 AWS Lambda 函數	使用 Aurora MySQL 時，您可以呼叫原生函數 <code>lambda_sync</code> 與 <code>lambda_async</code> 。如需詳細資訊，請參閱 從 Amazon Aurora MySQL 資料庫叢集呼叫 Lambda 函數 。	2017 年 12 月 11 日
新增 Aurora PostgreSQL HIPAA 資格	Aurora PostgreSQL 現在支援建置符合 HIPAA 資格的應用程式。如需詳細資訊，請參閱 使用 Amazon Aurora PostgreSQL 。	2017 年 12 月 6 日
具有 PostgreSQL 相容性的 Amazon Aurora 可用的其他 AWS 區域	具備 PostgreSQL 相容性的 Amazon Aurora 現已在四個新 AWS 區域推出。如需詳細資訊，請參閱 Amazon Aurora PostgreSQL 的可用性 。	2017 年 11 月 22 日
修改執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體的儲存體	您現在可以修改您執行 SQL Server 的 Amazon RDS 資料庫執行個體的儲存體。如需更多詳細資訊，請參閱 修改 Amazon RDS 資料庫執行個體 。	2017 年 11 月 21 日
Amazon RDS 對 Linux 式引擎支援 16 TiB 儲存體	您現在可以建立具有最高 16 TiB 儲存體的 MySQL、MariaDB、PostgreSQL 和 Oracle RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。	2017 年 11 月 21 日
Amazon RDS 支援快速擴展儲存體	您現在可以在幾分鐘內新增儲存體至 MySQL、MariaDB、PostgreSQL 和 Oracle RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。	2017 年 11 月 21 日

變更	描述	變更日期
Amazon RDS 支援 MariaDB 版本 10.1.26 和 10.0.32	您現在可以建立執行 MariaDB 10.1.26 版和 10.0.32 版的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2017 年 11 月 20 日
Amazon RDS for Microsoft SQL Server 現在支援新的資料庫執行個體類別	您現在可以建立使用 db.r4 和 db.m4.16xlarge 資料庫執行個體類別、執行 SQL Server 的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Microsoft SQL Server 的資料庫執行個體類別支援 。	2017 年 11 月 20 日
Amazon RDS for MySQL 和 MariaDB 現在支援新的資料庫執行個體類別	您現在可以建立使用 db.r4, db.m4.16xlarge、db.t2.xlarge 和 db.t2.2xlarge 資料庫執行個體類別、執行 MySQL 和 MariaDB 的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 資料庫執行個體類別 。	2017 年 11 月 20 日
SQL Server 2017	您現在可以建立執行 Microsoft SQL Server 2017 的 Amazon RDS 資料庫執行個體。您也可以建立執行 SQL Server 2016 SP1 CU5 的資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 。	2017 年 11 月 17 日
從 Amazon S3 還原 MySQL 備份	現在，您能夠建立現場部署資料庫的備份，並將其存放在 Amazon S3，接著將備份檔案還原至執行 MySQL 的新 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 將備份還原至 MySQL 資料庫執行個體 。	2017 年 11 月 17 日
Aurora 複本的 Auto Scaling	Amazon Aurora MySQL 現可支援 Aurora Auto Scaling。Aurora Auto Scaling 會根據連線或工作負載的增減，動態調整 Aurora 複本的數量。如需詳細資訊，請參閱 使用 Amazon Aurora Auto Scaling 搭配 Aurora 複本 。	2017 年 11 月 17 日
Oracle 預設版本支援	Amazon RDS for Oracle 資料庫執行個體現在支援設定資料庫執行個體的預設版本。如需更多詳細資訊，請參閱 設定資料庫執行個體的預設版本 。	2017 年 11 月 3 日

變更	描述	變更日期
Oracle 資料庫執行個體檔案驗證	Amazon RDS for Oracle 資料庫執行個體現在支援使用 Oracle Recovery Manager (RMAN) 邏輯驗證公用程式來驗證資料庫執行個體檔案。如需更多詳細資訊，請參閱 在 RDS 中為甲骨文驗證數據庫文件 。	2017 年 11 月 3 日
OEM 13c 適用的 Management Agent	Amazon RDS for Oracle 資料庫執行個體現在支援 Oracle Enterprise Manager (OEM) Cloud Control 13c 適用的 Management Agent。如需更多詳細資訊，請參閱 適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent 。	2017 年 11 月 1 日
Microsoft SQL Server 快照的儲存體重新組態	您現在可以在將快照還原到執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體時重新設定儲存體。如需更多詳細資訊，請參閱 從資料庫快照還原 。	2017 年 10 月 26 日
Aurora MySQL 相容版本的非同步索引鍵預先提取	非同步索引鍵預先提取 (AKP) 透過在需要時及早預先提取記憶體中的索引鍵，可改善非快取索引聯結的效能。如需詳細資訊，請參閱 在 Amazon Aurora 中使用非同步索引鍵預先提取 。	2017 年 10 月 26 日
MySQL 5.7.19、5.6.37 和 5.5.57	您現在可以建立執行 MySQL 5.7.19 版、5.6.37 版和 5.5.57 版的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 MySQL on Amazon RDS 版本 。	2017 年 10 月 25 日
正式推出具有 PostgreSQL 相容性的 Amazon Aurora	具有 PostgreSQL 相容性的 Amazon Aurora 以簡單且經濟實惠的方式設定、操作及擴展新的及現有的 PostgreSQL 部署，讓您得以心無旁騖，專注於業務和應用程式。如需詳細資訊，請參閱 使用 Amazon Aurora PostgreSQL 。	2017 年 10 月 24 日
Amazon RDS for Oracle 資料庫執行個體支援新的資料庫執行個體類別	Amazon RDS for Oracle 資料庫執行個體現在支援記憶體最佳化最新一代 (db.r4) 執行個體類別。Amazon RDS for Oracle 資料庫執行個體現在也支援下列新的最新一代執行個體類別：db.m4.16xlarge、db.t2.xlarge 和 db.t2.2xlarge。如需更多詳細資訊，請參閱 資料庫執行個體類別 及 RDS for Oracle 執行個體類別 。	2017 年 10 月 23 日

變更	描述	變更日期
新功能	您的新的和現有保留執行個體現在可在相同資料庫執行個體類別中涵蓋多個大小。大小靈活的預留執行個體適用於具有相同 AWS 區域、資料庫引擎和執行個體系列的資料庫執行個體，以及跨 AZ 組態。彈性大小的保留執行個體可供下列資料庫引擎使用：Amazon Aurora、MariaDB、MySQL、Oracle (使用自有授權)、PostgreSQL。如需更多詳細資訊，請參閱 彈性大小的預留資料庫執行個體 。	2017 年 10 月 11 日
新功能	您現在可以使用 Oracle SQLT 選項來調整 SQL 陳述式以獲得最佳效能。如需更多詳細資訊，請參閱 Oracle SQLT 。	2017 年 9 月 22 日
新功能	如果有 Amazon RDS for Oracle 資料庫執行個體的現有手動資料庫快照，則現在可以將它們升級至更新版本的 Oracle 資料庫引擎。如需更多詳細資訊，請參閱 升級 Oracle 資料庫快照 。	2017 年 9 月 20 日
新功能	您現在可以使用 Oracle Spatial 來儲存、擷取、更新和查詢您執行 Oracle 的 Amazon RDS 資料庫執行個體中的空間資料。如需更多詳細資訊，請參閱 Oracle Spatial 。	2017 年 9 月 15 日
新功能	您現在可以使用 Oracle Locator 透過您執行 Oracle 的 Amazon RDS 資料庫執行個體，來支援網際網路和無線服務型應用程式與合作夥伴型 GIS 解決方案。如需更多詳細資訊，請參閱 Oracle Locator 。	2017 年 9 月 15 日
新功能	您現在可以使用 Oracle Multimedia 來存放、管理和擷取您執行 Oracle 的 Amazon RDS 資料庫執行個體中的影像、音訊、視訊和其他異質媒體資料。	2017 年 9 月 15 日
新功能	您現在可以將稽核日誌從 Amazon Aurora MySQL 資料庫叢集匯出到 Amazon CloudWatch 日誌。如需詳細資訊，請參閱將 Aurora MySQL 日誌發佈到 Amazon CloudWatch 日誌 。	2017 年 9 月 14 日

變更	描述	變更日期
新功能	Amazon RDS 現在對您執行 Oracle 的資料庫執行個體支援多個版本的 Oracle Application Express (APEX)。如需更多詳細資訊，請參閱 Oracle Application Express (APEX) 。	2017 年 9 月 13 日
新功能	現在，您能夠利用 Amazon Aurora，將未加密或加密的資料庫快照或 MySQL 資料庫執行個體遷移至加密的 Aurora MySQL 資料庫叢集。如需詳細資訊，請參閱 將 RDS for MySQL 快照遷移至 Aurora ，以及 使用 Aurora 僅供讀取複本將資料從 MySQL 資料庫執行個體遷移至 Amazon Aurora MySQL 資料庫叢集 。	2017 年 9 月 5 日
新功能	您可以使用 Amazon RDS for Microsoft SQL Server 資料庫來建置符合 HIPAA 規範的應用程式。如需更多詳細資訊，請參閱 Microsoft SQL Server 資料庫執行個體的合規計劃支援 。	2017 年 8 月 31 日
新功能	您現在可以使用 Amazon RDS for MariaDB 資料庫來建置符合 HIPAA 規範的應用程式。如需更多詳細資訊，請參閱 Amazon RDS for MariaDB 。	2017 年 8 月 31 日
新功能	您現在可以建立執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體，其中所配置的儲存空間高達 16 TiB，且佈建 IOPS 與儲存空間範圍的比率為 1:1–50:1。如需更多詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體 。	2017 年 8 月 22 日
新功能	您現在可以對在歐洲 (法蘭克福) 區域中執行 Microsoft SQL Server 的資料庫執行個體使用多個可用區部署。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 的異地同步備份部署 。	2017 年 8 月 3 日
新功能	您現在可以建立執行 MariaDB 10.1.23 版和 10.0.31 版的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 Amazon RDS 上的 MariaDB 版本 。	2017 年 7 月 17 日

變更	描述	變更日期
新功能	Amazon RDS 現在支持 Microsoft SQL 伺服器企業版，並在所有 AWS 區域使用包含許可證模式。如需詳細資訊，請參閱 Amazon RDS 上的 Microsoft SQL Server 授權 。	2017 年 7 月 13 日
新功能	Amazon RDS for Oracle 現在支援 Linux 核心巨型分頁以增加資料庫可擴展性。使用巨型分頁會造成較小型的分頁資料表，以及耗費在記憶體管理方面較少的 CPU，增加大型資料庫執行個體的效能。您能夠使用巨型分頁，進而搭配使用執行 Oracle 12.1.0.2 版和 11.2.0.4 版所有版次的 Amazon RDS 資料庫執行個體。如需更多詳細資訊，請參閱 對 RDS for Oracle 執行個體開啟巨型分頁 。	2017 年 7 月 7 日
新功能	更新以針對所有非 Aurora 資料庫引擎的 db.t2.small 和 db.t2.medium 資料庫執行個體類別支援靜態加密 (EAR)。如需更多詳細資訊，請參閱 Amazon RDS 加密的可用性 。	2017 年 6 月 27 日
新功能	更新以在歐洲 (法蘭克福) 區域中支援 Amazon Aurora。如需詳細資訊，請參閱 Amazon Aurora MySQL 的可用性 。	2017 年 6 月 16 日
新功能	您現在可以在跨 AWS 區域複製資料庫快照時指定選項群組。如需詳細資訊，請參閱 選項群組考量 。	2017 年 6 月 12 日
新功能	您現在可以跨 AWS 區域複製從專用資料庫執行個體建立的資料庫快照。您可以從使用 Oracle TDE、Microsoft SQL Server TDE 和 Microsoft SQL Server 具有鏡像的多個可用區的資料庫執行個體複製快照。如需更多詳細資訊，請參閱 複製資料庫快照 。	2017 年 6 月 12 日
新功能	Amazon Aurora 現在可讓您在 Amazon Aurora 資料庫叢集中快速且具成本效益地複製您的所有資料庫。如需詳細資訊，請參閱 複製 Aurora 資料庫叢集中的資料庫 。	2017 年 6 月 12 日

變更	描述	變更日期
新功能	Amazon RDS 現在支援 Microsoft SQL Server 2016 SP1 CU2。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 。	2017 年 6 月 7 日
預覽版	公開預覽具有 PostgreSQL 相容性的 Amazon Aurora。如需詳細資訊，請參閱 使用 Amazon Aurora PostgreSQL 。	2017 年 4 月 19 日
新功能	Amazon Aurora 現在能讓您近乎即時地執行 ALTER TABLE tbl_name ADD COLUMN col_name column_definition 操作。此操作不需要複製資料表即可完成，且對其他 DML 陳述式沒有實質影響。如需詳細資訊，請參閱 使用快速 DDL 更改 Amazon Aurora 中的資料表 。	2017 年 4 月 5 日
新功能	我們新增新的監控命令 SHOW VOLUME STATUS 來顯示磁碟區中節點和磁碟的數量。如需詳細資訊，請參閱 顯示 Aurora 資料庫叢集的磁碟區狀態 。	2017 年 4 月 5 日
新功能	您現在可以針對 Amazon RDS 上的 Oracle 在您的自訂密碼驗證函數中使用您自己的自訂邏輯。如需更多詳細資訊，請參閱 建立自訂函數以驗證密碼 。	2017 年 3 月 21 日
新功能	您現在可以在 Amazon RDS 上您的 Oracle 資料庫執行個體中存取您的線上和封存重做日誌檔案。如需更多詳細資訊，請參閱 存取線上和封存的重新日誌 。	2017 年 3 月 21 日
新功能	您現在可以在相同區域的帳戶之間複製加密和未加密的資料庫叢集快照。如需詳細資訊，請參閱 跨帳戶複製資料庫叢集快照 。	2017 年 3 月 7 日
新功能	您現在可以在相同區域的帳戶之間共用加密的資料庫叢集快照。如需詳細資訊，請參閱 共用資料庫叢集快照 。	2017 年 3 月 7 日
新功能	您現在可以複製加密的 Amazon Aurora MySQL 資料庫叢集，以建立跨區域 Aurora 複本。如需詳細資訊，請參閱 跨 AWS 區域複製 Aurora MySQL 資料庫叢集 。	2017 年 3 月 7 日

變更	描述	變更日期
新功能	您現在可以要求對您執行 Microsoft SQL Server 的資料庫執行個體的所有連線使用 Secure Sockets Layer (SSL)。如需更多詳細資訊，請參閱 對 Microsoft SQL Server 資料庫執行個體使用 SSL 。	2017 年 2 月 27 日
新功能	您現在可以將您的本機時區設定為 15 個額外時區中的一個。如需更多詳細資訊，請參閱 支援的時區 。	2017 年 2 月 27 日
新功能	您現在能夠使用 Amazon RDS 程序 msdb.dbo.rds_shrink_tempdbfile，藉此在執行 Microsoft SQL Server 的資料庫執行個體上縮減 tempdb 資料庫。如需更多詳細資訊，請參閱 縮減 tempdb 資料庫 。	2017 年 2 月 17 日
新功能	現在，當您從 Amazon RDS 資料庫執行個體將 Enterprise 與 Standard Edition Microsoft SQL Server 資料庫匯出至 Amazon S3 時，能夠壓縮備份檔案。如需更多詳細資訊，請參閱 壓縮備份檔案 。	2017 年 2 月 17 日
新功能	Amazon RDS 現在支援自訂 DNS 伺服器，以解析在您執行 Oracle 的資料庫執行個體上傳出網路存取中使用的 DNS 名稱。如需更多詳細資訊，請參閱 設定自訂 DNS 伺服器 。	2017 年 1 月 26 日
新功能	Amazon RDS 現在支援在另一個區域中建立加密的僅供讀取複本。如需詳細資訊，請參閱 以不同的方式建立僅供讀取複本 AWS 區域 和 CreateDB 複 Instance Read 本。	2017 年 1 月 23 日
新功能	Amazon RDS 現可支援將 MySQL 資料庫快照從 MySQL 5.1 升級至 MySQL 5.5。	2017 年 1 月 20 日
新功能	Amazon RDS 現在支援將加密的資料庫快照複製到 MariaDB、MySQL、Oracle、PostgreSQL 和 Microsoft SQL Server 資料庫引擎的另一個區域。如需詳細資訊，請參閱 複製資料庫快照 和 CopyDBSnapshot 。	2016 年 12 月 20 日

變更	描述	變更日期
新功能	<p>Amazon Aurora MySQL 現在支援空間索引。</p> <p>空間索引可改善使用空間資料對大型資料集進行查詢的查詢效能。如需詳細資訊，請參閱 Amazon Aurora MySQL 和空間資料。</p>	2016 年 12 月 14 日
新功能	<p>Amazon RDS 現在於您執行 Oracle 的資料庫執行個體上支援傳出網路存取。您可以使用 utl_http、utl_tcp 和 utl_smtp 從您的資料庫執行個體連線至網路。如需更多詳細資訊，請參閱 使用憑證和 Oracle 錢包設定 UTL_HTTP 存取。</p>	2016 年 12 月 5 日
新功能	<p>Amazon RDS 不再支援 MySQL 5.1 版。不過，您仍可將現有的 MySQL 5.1 快照還原至 MySQL 5.5 執行個體。如需更多詳細資訊，請參閱 支援的 RDS for MySQL 儲存引擎。</p>	2016 年 11 月 15 日
新功能	<p>Amazon RDS 現在支援 Microsoft SQL Server 2016 RTM CU2。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server。</p>	2016 年 11 月 4 日
新功能	<p>Amazon RDS 現在支援執行 Oracle 的資料庫執行個體的主要版本升級。您現在可以將您的 Oracle 資料庫執行個體從 11g 升級至 12c。如需更多詳細資訊，請參閱 升級 RDS for Oracle 資料庫引擎。</p>	2016 年 11 月 2 日
新功能	<p>您現在可以建立執行 Microsoft SQL Server 2014 Enterprise Edition 的資料庫執行個體。Amazon RDS 現可支援 SQL Server 2014 SP2，所有版本及所有地區均適用。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server。</p>	2016 年 10 月 25 日

變更	描述	變更日期
新功能	Amazon Aurora MySQL 現在與其他 AWS 服務整合：您可以從 Amazon S3 儲存貯體將文字或 XML 資料載入資料表中，或從資料庫程式碼叫用 AWS Lambda 函數。如需詳細資訊，請參閱 整合 Aurora MySQL 與其他 AWS 服務 。	2016 年 10 月 18 日
新功能	您現在可以在您執行 Microsoft SQL Server 的 Amazon RDS 資料庫執行個體上存取 tempdb 資料庫。您可以使用 Transact-SQL 透過 Microsoft SQL Server Management Studio (SSMS)，或任何其他標準 SQL 用戶端應用程式來存取 tempdb 資料庫。如需更多詳細資訊，請參閱 在 Amazon RDS 上存取 Microsoft SQL Server 資料庫執行個體上的 tempdb 資料庫 。	2016 年 9 月 29 日
新功能	您現在可以利用您執行 Oracle 的 Amazon RDS 資料庫執行個體來使用 UTL_MAIL 套件。如需更多詳細資訊，請參閱 Oracle UTL_MAIL 。	2016 年 9 月 20 日
新功能	您現在可以將新 Microsoft SQL Server 資料庫執行個體的時區設定為本機時區，以符合應用程式的時區。如需更多詳細資訊，請參閱 Microsoft SQL Server 資料庫執行個體的本機時區 。	2016 年 9 月 19 日
新功能	您現在可以使用 Oracle Label Security 選項來控制對您執行 Oracle 資料庫 12c 的 Amazon RDS 資料庫執行個體中個別資料表資料列的存取。有了 Oracle Label Security，您可以利用政策式管理模式來強制執行法規合規，並確保將對機密資料的存取限制在僅具有適當許可層級的使用者才能存取。如需更多詳細資訊，請參閱 Oracle Label Security 。	2016 年 9 月 8 日

變更	描述	變更日期
新功能	您現在可以使用讀取器端點來連線至 Amazon Aurora 資料庫叢集，該端點可將資料庫叢集中可用的 Aurora 複本上連線的負載平衡。由於用戶端會向讀取器端點要求新連線，Aurora 會在資料庫叢集中的 Aurora 複本間分配連線請求。此功能有助於在您的資料庫叢集中的多個 Aurora 複本間平衡您的讀取工作負載。如需詳細資訊，請參閱 Amazon Aurora 端點 。	2016 年 9 月 8 日
新功能	您現在可以在您執行 Oracle 的 Amazon RDS 資料庫執行個體上支援 Oracle Enterprise Manager Cloud Control。您可以在您的資料庫執行個體上啟用 Management Agent，並與您的 Oracle Management Service (OMS) 共用資料。如需更多詳細資訊，請參閱 適用於 Enterprise Manager Cloud Control 的 Oracle Management Agent 。	2016 年 9 月 1 日
新功能	此版本新增了支援以取得資源的 ARN。如需更多詳細資訊，請參閱 取得現有的 ARN 。	2016 年 8 月 23 日
新功能	您現在可以指派最多 50 個標籤給每個 Amazon RDS 資源，用於管理資源和追蹤成本。如需更多詳細資訊，請參閱 標記 Amazon RDS 資源 。	2016 年 8 月 19 日
新功能	<p>Amazon RDS 現在支援 Oracle Standard Edition Two 的「已包含授權」模式。如需更多詳細資訊，請參閱 建立 Amazon RDS 資料庫執行個體。</p> <p>您現在可以變更您執行 Microsoft SQL Server 和 Oracle 的 Amazon RDS 資料庫執行個體的授權模式。如需更多詳細資訊，請參閱 Amazon RDS 上的 Microsoft SQL Server 授權 及 RDS for Oracle 授權選項。</p>	2016 年 8 月 5 日

變更	描述	變更日期
新功能	Amazon RDS 現在支援使用完整備份檔案 (.bak 檔案) 進行 Microsoft SQL Server 資料庫的原生備份與還原。您現在可以輕鬆地將 SQL Server 資料庫遷移到 Amazon RDS，並在單一、易於攜帶的檔案中匯入和匯出資料庫，並使用 Amazon S3 進行儲存和 AWS KMS 加密。如需詳細資訊，請參閱 使用原生備份和還原，匯入和匯出 SQL Server 資料庫 。	2016 年 7 月 27 日
新功能	您現在可以將 MySQL 資料庫中的來源檔案複製到 Amazon Simple Storage Service (Amazon S3) 儲存貯體，然後從這些檔案還原 Amazon Aurora 資料庫叢集。比起使用 <code>mysqldump</code> ，此選項遷移資料的速度更快。如需詳細資訊，請參閱 從外部 MySQL 資料庫將資料遷移至 Aurora MySQL 資料庫叢集 。	2016 年 7 月 20 日
新功能	您現在可以透過在還原作業期間包含 AWS Key Management Service (AWS KMS) 加密金鑰來還原未加密的 Amazon Aurora 資料庫叢集快照，以建立加密的 Amazon Aurora 資料庫叢集。如需詳細資訊，請參閱 加密 Amazon RDS 資源 。	2016 年 6 月 30 日
新功能	您可以使用 Oracle Repository Creation Utility (RCU) 在 Amazon RDS for Oracle 上建立儲存庫。如需更多詳細資訊，請參閱 在 RDS for Oracle 上使用 Oracle Repository Creation Utility 。	2016 年 6 月 17 日
新功能	新增 PostgreSQL 跨區域僅供讀取複本的支援。如需更多詳細資訊，請參閱 以不同的方式建立僅供讀取複本 AWS 區域 。	2016 年 6 月 16 日
新功能	您現在可以使用將具有鏡像的異地同步備份輕鬆新增至 Microsoft SQL Server 資料庫執行個體。AWS Management Console 如需詳細資訊，請參閱 將多可用區域新增至 Microsoft SQL Server 資料庫執行個體 。	2016 年 6 月 9 日

變更	描述	變更日期
新功能	您現在可以在下列額外的區域中使用具有 SQL Server 鏡像的異地同步備份部署：亞太區域 (雪梨)、亞太區域 (東京) 和南美洲 (聖保羅)。如需詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 的異地同步備份部署 。	2016 年 6 月 9 日
新功能	更新以支援 MariaDB 10.1 版本。如需更多詳細資訊，請參閱 Amazon RDS for MariaDB 。	2016 年 6 月 1 日
新功能	已更新以支援屬於僅供讀取複本的 Amazon Aurora 跨區域資料庫叢集。如需更多詳細資訊，請參閱 跨 AWS 區域複寫 Aurora MySQL 資料庫叢集 。	2016 年 6 月 1 日
新功能	Enhanced Monitoring 現在可供 Oracle 資料庫執行個體使用。如需更多詳細資訊，請參閱 使用增強型監控來監控作業系統指標 及 修改 Amazon RDS 資料庫執行個體 。	2016 年 5 月 27 日
新功能	更新以支援 Amazon Aurora 資料庫叢集快照的手動快照共用。如需詳細資訊，請參閱 共用資料庫叢集快照 。	2016 年 5 月 18 日
新功能	您現在可以使用 MariaDB 稽核外掛程式來記錄 MariaDB 和 MySQL 資料庫執行個體上的資料庫活動。如需更多詳細資訊，請參閱 MariaDB 資料庫引擎的選項 及 MySQL 資料庫執行個體的選項 。	2016 年 4 月 27 日
新功能	從 MySQL 5.6 版升級至 5.7 版時，現在可以使用就地升級的主要版本。如需更多詳細資訊，請參閱 升級 MySQL 資料庫引擎 。	2016 年 4 月 26 日
新功能	Enhanced Monitoring 現在可供 Microsoft SQL Server 執行個體使用。如需更多詳細資訊，請參閱 使用增強型監控來監控作業系統指標 。	2016 年 4 月 22 日
新功能	更新以在 Amazon RDS 主控台中提供 Amazon Aurora Clusters (叢集) 檢視。如需詳細資訊，請參閱 檢視 Aurora 資料庫叢集 。	2016 年 4 月 1 日

變更	描述	變更日期
新功能	更新以在亞太區域 (首爾) 區域中支援搭配鏡像的 SQL Server 多個可用區。如需更多詳細資訊，請參閱 Amazon RDS for Microsoft SQL Server 的異地同步備份部署 。	2016 年 3 月 31 日
新功能	更新以在亞太區域 (首爾) 區域中支援搭配鏡像的 Amazon Aurora 多個可用區。如需詳細資訊，請參閱 Amazon Aurora MySQL 的可用性 。	2016 年 3 月 31 日
新功能	PostgreSQL 資料庫執行個體有能力可要求連線使用 SSL。如需更多詳細資訊，請參閱 將 SSL 與 PostgreSQL 資料庫執行個體搭配使用 。	2016 年 3 月 25 日
新功能	Enhanced Monitoring 現在可供 PostgreSQL 資料庫執行個體使用。如需更多詳細資訊，請參閱 使用增強型監控來監控作業系統指標 。	2016 年 3 月 25 日
新功能	Microsoft SQL Server 資料庫執行個體現在可對使用者身分驗證使用 Windows 身分驗證。如需更多詳細資訊，請參閱 使用適用於 SQL 伺服器的 RDS AWS 受管理的使用中目錄 。	2016 年 3 月 23 日
新功能	Enhanced Monitoring 現已在亞太 (首爾) 區域提供。如需更多詳細資訊，請參閱 使用增強型監控來監控作業系統指標 。	2016 年 3 月 16 日
新功能	您現在可以自訂在容錯移轉期間將 Aurora 複本提升至主要執行個體的順序。如需詳細資訊，請參閱 Aurora 資料庫叢集的容錯能力 。	2016 年 3 月 14 日
新功能	更新以在遷移至 Aurora 資料庫叢集時支援加密。如需詳細資訊，請參閱 將資料遷移至 Aurora 資料庫叢集 。	2016 年 3 月 2 日
新功能	更新以支援 Aurora 資料庫叢集的本機時區。如需詳細資訊，請參閱 Aurora 資料庫叢集的本機時區 。	2016 年 3 月 1 日

變更	描述	變更日期
新功能	更新以新增最新一代 Amazon RDS 資料庫執行個體類別 MySQL 版本 5.7 的支援。	2016 年 2 月 22 日
新功能	已更新，可支援 AWS GovCloud (美國西部) 區域中的 db.r3 和 db.t2 資料庫執行個體類別。	2016 年 2 月 11 日
新功能	更新以支援加密資料庫快照的複本和共用加密的資料庫快照。如需更多詳細資訊，請參閱 複製資料庫快照 及 共享資料庫快照 。	2016 年 2 月 11 日
新功能	更新以在亞太區域 (雪梨) 區域中支援 Amazon Aurora。如需詳細資訊，請參閱 Amazon Aurora MySQL 的可用性 。	2016 年 2 月 11 日
新功能	更新以針對 Oracle 資料庫執行個體支援 SSL。如需更多詳細資訊，請參閱 使用 SSL 搭配 RDS for Oracle 資料庫執行個體 。	2016 年 2 月 9 日
新功能	更新以支援 MySQL 和 MariaDB 資料庫執行個體的本機時區。如需更多詳細資訊，請參閱 MySQL 資料庫執行個體的本機時區 及 MariaDB 資料庫執行個體的本機時區 。	2015 年 12 月 21 日
新功能	更新以支援 MySQL 和 MariaDB 執行個體和 Aurora 資料庫叢集作業系統指標的 Enhanced Monitoring。如需更多詳細資訊，請參閱 在 Amazon RDS 主控台中檢視指標 。	2015 年 12 月 18 日
新功能	更新以支援 MySQL 5.5 版的 db.t2、db.r3 和 db.m4 資料庫執行個體類別。如需更多詳細資訊，請參閱 資料庫執行個體類別 。	2015 年 12 月 4 日
新功能	更新以支援修改現有資料庫執行個體的資料庫連接埠。	2015 年 12 月 3 日
新功能	更新以支援 PostgreSQL 執行個體資料庫引擎主要版本的升級。如需更多詳細資訊，請參閱 升級 Amazon RDS 的 PostgreSQL 資料庫引擎 。	2015 年 11 月 19 日

變更	描述	變更日期
新功能	更新以支援修改現有資料庫執行個體的公開存取性。更新以支援 db.m4 標準資料庫執行個體類別。	2015 年 11 月 11 日
新功能	更新以支援手動資料庫快照共用。如需更多詳細資訊，請參閱 共享資料庫 快照 。	2015 年 10 月 28 日
新功能	更新以支援 Microsoft SQL Server 2014 的 Web、Express 和 Standard 版本。	2015 年 10 月 26 日
新功能	更新以支援 MySQL 型 MariaDB 資料庫引擎。如需更多詳細資訊，請參閱 Amazon RDS for MariaDB 。	2015 年 10 月 7 日
新功能	更新以在亞太區域 (東京) 區域中支援 Amazon Aurora。如需詳細資訊，請參閱 Amazon Aurora MySQL 的可用性 。	2015 年 10 月 7 日
新功能	更新以支援所有資料庫引擎的 db.t2 具備高載的資料庫執行個體類別，以及新加入的 db.t2.large 資料庫執行個體類別。如需更多詳細資訊，請參閱 資料庫執行個體類別 。	2015 年 9 月 25 日
新功能	更新以在 R3 和 T2 資料庫執行個體類別上支援 Oracle 資料庫執行個體。如需更多詳細資訊，請參閱 資料庫執行個體類別 。	2015 年 8 月 5 日
新功能	Microsoft SQL Server Enterprise Edition 現在以「已包含授權」服務模式提供。如需更多詳細資訊，請參閱 Amazon RDS 上的 Microsoft SQL Server 授權 。	2015 年 7 月 29 日
新功能	Amazon Aurora 已正式推出。Amazon Aurora 資料庫引擎支援資料庫叢集中的多個資料庫執行個體。如需詳細資訊，請參閱 什麼是 Amazon Aurora ? 。	2015 年 7 月 27 日
新功能	更新以支援複製標籤至資料庫快照。	2015 年 7 月 20 日
新功能	更新以支援所有資料庫引擎儲存體大小的增加和 SQL Server 佈建 IOPS 中的增加。	2015 年 6 月 18 日

變更	描述	變更日期
新功能	更新保留資料庫執行個體的選項。	2015 年 6 月 15 日
新功能	更新以支援對使用 TDE 的 Oracle 資料庫執行個體使用 Amazon CloudHSM。	2015 年 1 月 8 日
新功能	更新以支援靜態加密資料和新的 API 版本 2014-10-31。	2015 年 1 月 6 日
新功能	更新以包含新的 Amazon 資料庫引擎：Aurora。Amazon Aurora 資料庫引擎支援資料庫叢集中的多個資料庫執行個體。Amazon Aurora 目前為預覽版本，並可能有所變更。如需詳細資訊，請參閱 什麼是 Amazon Aurora ? 。	2014 年 11 月 12 日
新功能	已更新以支援 PostgreSQL 僅供讀取複本。	2014 年 11 月 10 日
新 API 和功能	更新以支援 GP2 儲存體類型和新的 API 版本 2014-09-01。更新以支援複製現有選項或參數群組以建立新選項或參數群組的功能。	2014 年 10 月 7 日
新功能	更新以支援執行 MySQL 版本 5.6.19 和更新版本資料庫執行個體的 InnoDB 快取預備。	2014 年 9 月 3 日
新功能	更新以在連線至 MySQL 版本 5.6、SQL Server 和 PostgreSQL 資料庫引擎時支援 SSL 憑證驗證。	2014 年 8 月 5 日
新功能	更新以支援 db.t2 具備爆量資料庫執行個體類別。	2014 年 8 月 4 日
新功能	更新以支援 db.r3 記憶體最佳化資料庫執行個體類別，以搭配使用 MySQL (版本 5.6)、SQL Server 和 PostgreSQL 資料庫引擎。	2014 年 5 月 28 日
新功能	更新以支援使用 SQL Server 鏡像的 SQL Server 多個可用區部署。	2014 年 5 月 19 日
新功能	更新以支援從 MySQL 版本 5.5 升級至版本 5.6。	2014 年 4 月 23 日

變更	描述	變更日期
新功能	已更新以支持甲骨文 GoldenGate。	2014 年 4 月 3 日
新功能	更新以支援 M3 資料庫執行個體類別。	2014 年 2 月 20 日
新功能	更新以支援 Oracle 時區選項。	2014 年 1 月 13 日
新功能	更新以支援不同區域的 MySQL 資料庫執行個體之間的複寫。	2013 年 11 月 26 日
新功能	更新以支援 PostgreSQL 資料庫引擎。	2013 年 11 月 14 日
新功能	更新以支援 SQL Server 透明資料加密支援 (TDE)。	2013 年 11 月 7 日
新 API 和新功能	更新以支援跨區域資料庫快照複本；新的 API 版本 2013-09-09。	2013 年 10 月 31 日
新功能	更新以支援 Oracle Statspack。	2013 年 9 月 26 日
新功能	更新以支援使用複寫操作，藉此針對 Amazon RDS 中執行的 MySQL 執行個體，以及現場部署或 Amazon EC2 上執行的 MySQL 執行個體，在兩者之間匯入或匯出資料。	2013 年 9 月 5 日
新功能	更新以支援 MySQL 5.6 的 db.cr1.8xlarge 資料庫執行個體類別。	2013 年 9 月 4 日
新功能	已更新以支援僅供讀取複本的複寫。	2013 年 8 月 28 日
新功能	已更新以支援平行僅供讀取複本的建立。	2013 年 7 月 22 日
新功能	更新以支援所有 Amazon RDS 資源的精細定義許可和標記。	2013 年 7 月 8 日
新功能	更新以支援 MySQL 5.6 的新執行個體，包括支援 MySQL 5.6 memcached 界面和二進位日誌存取。	2013 年 7 月 1 日

變更	描述	變更日期
新功能	更新以支援從 MySQL 5.1 至 MySQL 5.5 的主要版本升級。	2013 年 6 月 20 日
新功能	已更新資料庫參數群組以允許參數值的表達式。	2013 年 6 月 20 日
新 API 和新功能	已更新以支援僅供讀取複本狀態；新 API 版本 2013-05-15。	2013 年 5 月 23 日
新功能	更新以支援 Oracle Advanced Security 功能的原生網路加密和 Oracle 透明資料加密。	2013 年 4 月 18 日
新功能	更新以支援 SQL Server 的主要版本升級和佈建 IOPS 的其他功能。	2013 年 3 月 13 日
新功能	更新以預設支援 RDS 的 VPC。	2013 年 3 月 11 日
新 API 和功能	更新以支援日誌存取；新 API 版本 2013-02-12	2013 年 3 月 4 日
新功能	更新以支援 RDS 事件通知訂閱。	2013 年 2 月 4 日
新 API 和功能	更新以支援資料庫執行個體重新命名和將 VPC 中的資料庫安全群組成員遷移至 VPC 安全群組。	2013 年 1 月 14 日
新功能	更新了 AWS GovCloud (美國西部) 支持。	2012 年 12 月 17 日
新功能	更新以支援 m1.medium 和 m1.xlarge 資料庫執行個體類別。	2012 年 11 月 6 日
新功能	已更新以支援僅供讀取複本提升。	2012 年 10 月 11 日
新功能	更新以在 Microsoft SQL Server 資料庫執行個體中支援 SSL。	2012 年 10 月 10 日
新功能	更新以支援 Oracle 微型資料庫執行個體。	2012 年 9 月 27 日
新功能	更新以支援 SQL Server 2012。	2012 年 9 月 26 日

變更	描述	變更日期
新 API 和功能	更新以支援佈建 IOPS。API 版本 2012-09-17。	2012 年 9 月 25 日
新功能	更新對 VPC 中資料庫執行個體的 SQL Server 的支援和 Data Pump 的 Oracle 支援。	2012 年 9 月 13 日
新功能	更新對 SQL Server 代理程式的支援。	2012 年 8 月 22 日
新功能	更新以支援資料庫執行個體的標記。	2012 年 8 月 21 日
新功能	更新以支援 Oracle APEX 和 XML DB、Oracle 時區和 VPC 中的 Oracle 資料庫執行個體。	2012 年 8 月 16 日
新功能	更新以支援 SQL Server Database Engine Tuning Advisor 和 VPC 中的 Oracle 資料庫執行個體。	2012 年 7 月 18 日
新功能	更新以支援選項群組和第一個選項 Oracle Enterprise Manager 資料庫控制。	2012 年 5 月 29 日
新功能	已更新以在 Amazon Virtual Private Cloud 中支援僅供讀取複本。	2012 年 5 月 17 日
新功能	更新對 Microsoft SQL Server 的支援。	2012 年 5 月 8 日
新功能	更新以支援強制容錯移轉、Oracle 資料庫執行個體的多可用區部署，以及 Oracle 資料庫執行個體非預設的字元集。	2012 年 5 月 2 日
新功能	更新對 Amazon Virtual Private Cloud (VPC) 的支援。	2012 年 2 月 13 日
已更新內容	對新保留執行個體類型進行更新。	2011 年 12 月 19 日
新功能	更新對 Oracle 引擎的支援。	2011 年 5 月 23 日
已更新內容	主控台更新。	2011 年 5 月 13 日
已更新內容	編輯縮短備份和維護時段的內容。	2011 年 2 月 28 日
新功能	新增對 MySQL 5.5 的支援。	2011 年 1 月 31 日

變更	描述	變更日期
新功能	已新增對僅供讀取複本的支援。	2010 年 10 月 4 日
新功能	已新增對 AWS Identity and Access Management (IAM) 的支援。	2010 年 9 月 2 日
新功能	新增資料庫引擎版本管理。	2010 年 8 月 16 日
新功能	新增保留的資料庫執行個體。	2010 年 8 月 16 日
新功能	Amazon RDS 現在支援對您的資料庫執行個體進行 SSL 連線。	2010 年 6 月 28 日
新的指南	此為 Amazon RDS 使用者指南 的第一版。	2010 年 6 月 7 日

AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的[AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。