



使用者指南

Amazon ElastiCache



API 版本 2015-02-02

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

Amazon ElastiCache: 使用者指南

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

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

Table of Contents

什麼是 ElastiCache ?	1
無伺服器快取	1
自行設計的叢集	1
相關服務	2
運作方式	3
快取和快取引擎	3
選擇部署選項	7
ElastiCache 資源	12
AWS 區域和可用區域	13
使用案例	15
入門 ElastiCache	22
設定 ElastiCache	22
註冊 AWS 帳戶	22
建立具有管理存取權的使用者	23
授與程式設計存取權	24
設定許可	25
設定 EC2	26
授予網路存取權	26
設定命令列存取	27
建立 Valkey 無伺服器快取	28
讀取和寫入資料	29
清除	31
後續步驟	32
建立 Valkey 或 Redis 無OSS伺服器快取	32
讀取和寫入資料	33
清除	35
後續步驟	36
建立 Memcached 無伺服器快取	36
讀取和寫入資料	37
清除	42
後續步驟	43
教學課程：開始使用 Python 和 ElastiCache	43
Python 和 ElastiCache	43
教學課程：設定 Lambda 在 ElastiCache 中存取 VPC	61

步驟 1：建立無 ElastiCache 伺服器快取。	61
步驟 2：為 建立 Lambda 函數 ElastiCache	64
步驟 3：使用 測試 Lambda 函數 ElastiCache	68
步驟 4：清除（選用）	68
設計您自己的 ElastiCache 叢集	70
元件和功能	70
節點	71
ElastiCache 碎片	71
ElastiCache 叢集	72
ElastiCache 複寫	74
ElastiCache 端點	75
參數群組	76
ElastiCache 安全性	76
子網路群組	77
ElastiCache 備份	77
事件	77
ElastiCache 術語	79
教學課程：如何設計自己的叢集	81
設計您自己的 ElastiCache（Valkey）叢集	81
設計您自己的 ElastiCache（RedisOSS）叢集	99
刪除叢集	118
其他教學課程和影片	119
影片	121
在 中管理節點 ElastiCache	125
檢視 ElastiCache 節點狀態	126
Valkey 或 Redis OSS 節點和碎片	130
連線至節點	132
支援的節點類型	137
重新啟動節點	151
更換節點（Valkey 和 RedisOSS）	156
更換節點（Memcached）	162
預留節點	163
遷移上一代節點	177
在 中管理叢集 ElastiCache	180
在 中選擇網路類型 ElastiCache	182
自動探索（Memcached）	187

中的資料分層 ElastiCache	223
在中準備叢集 ElastiCache	229
為 Valkey 或 Redis 建立叢集 OSS	238
為 Memcached 建立叢集	246
檢視 ElastiCache 叢集的詳細資訊	249
修改 ElastiCache 叢集	262
將節點新增至 ElastiCache 叢集	268
從 ElastiCache 叢集移除節點	278
取消中的待定新增或刪除節點操作 ElastiCache	286
在中刪除叢集 ElastiCache	287
存取 ElastiCache 叢集或複寫群組	290
在中尋找連線端點 ElastiCache	298
中的碎片 ElastiCache	312
比較 ValkeyOSS、Redis 和 Memcached 自行設計的快取	317
Valkey 或 Redis 的線上遷移 OSS	321
概觀	322
遷移步驟	322
準備您的來源和目標以進行遷移	322
測試資料遷移	324
開始遷移	324
驗證資料遷移程序	326
完成資料遷移	326
使用主控台執行線上資料遷移	327
選擇的區域和可用區域 ElastiCache	328
使用 Memcached 的可用區域考量	329
安置您的節點	331
支援的區域和端點	331
搭配使用本機區域 ElastiCache	336
搭配使用 Outposts ElastiCache	337
使用 ElastiCache	341
快照和還原	341
限制	342
自行設計叢集之備份的效能影響	342
排程自動備份	344
取得手動備份	345
建立最終備份	351

描述備份	354
複製備份	356
匯出備份	358
從備份進行還原	366
刪除備份	368
標記備份	370
教學課程：使用備份查看自我設計的叢集	371
在 中的引擎版本和升級 ElastiCache	379
的版本管理 ElastiCache	379
如何升級引擎版本	383
如何觸發從 Redis OSS 到 Valkey 的跨引擎升級	383
支援的版本	385
與 Valkey 的主要版本行為和相容性差異	404
與 Redis 的主要版本行為和相容性差異 OSS	404
使用 Valkey 或 Redis 解決封鎖的引擎升級 OSS	407
最佳實務和快取策略	408
整體最佳實務	408
支援和限制的 Valkey、Redis OSS和 Memcached 命令	409
Valkey 和 Redis OSS組態和限制	444
IPv6 Valkey、Redis OSS和 Memcached 的用戶端範例	447
用戶端的最佳實務（Valkey 和 RedisOSS）	448
用戶端的最佳實務（Memcached）	468
TLS 已啟用雙堆疊 ElastiCache 叢集	471
管理 Valkey 和 Redis 的預留記憶體 OSS	475
使用 Valkey 和 Redis OSS自行設計的叢集時的最佳實務	480
Memcached 的快取策略	485
在 中管理自行設計的叢集 ElastiCache	490
Auto Scaling Valkey 和 Redis OSS叢集	491
修改叢集模式	531
使用全域資料存放區跨 AWS 區域複寫	534
使用複寫群組的高可用性	558
管理 ElastiCache 叢集維護	638
使用參數群組設定引擎 ElastiCache 參數	640
擴展 ElastiCache	737
Scaling ElastiCache Serverless	737
設定擴展限制以管理成本	737

使用 ElastiCache Serverless 預先擴展	737
使用主控台 和 設定擴展限制 AWS CLI	738
擴展自我設計的叢集	739
適用於 Valkey 和 Redis JSON 的 入門 OSS	808
JSON 資料類型概觀	809
JSON 命令	820
標記您的 ElastiCache 資源	861
使用標籤監控成本	871
使用 管理標籤 AWS CLI	873
使用 管理標籤 ElastiCache API	877
Amazon ElastiCache Well-Architected 鏡頭	880
卓越運作支柱	881
安全支柱	888
可靠性支柱	893
效能達成效率支柱	897
成本最佳化支柱	906
在 中進行故障診斷 ElastiCache	911
連線問題	911
Valkey 或 Redis OSS用戶端錯誤	912
對 ElastiCache Serverless 中的高延遲進行故障診斷	912
對 ElastiCache Serverless 中的限流問題進行故障診斷	914
持續性連線問題	914
相關主題	931
安全	932
資料保護	932
Amazon 中的資料安全 ElastiCache	933
網際網路流量隱私權	1003
Amazon VPCs和安全性 ElastiCache	1003
ElastiCache API 和介面VPC端點 (AWS PrivateLink)	1025
子網路和子網路群組	1028
身分和存取權管理	1034
物件	1035
使用身分驗證	1035
使用政策管理存取權	1038
Amazon 如何使用 ElastiCache IAM	1040
身分型政策範例	1046

故障診斷	1048
存取控制	1050
管理存取概觀	1051
法規遵循驗證	1094
其他資訊	1095
恢復能力	1095
減少故障	1096
基礎架構安全	1100
服務更新	1100
管理服務更新	1101
已解決安全漏洞	1105
日誌記錄和監控	1107
Valkey 和 Redis 的無伺服器指標和事件 OSS	1107
無伺服器指標	1107
無伺服器事件	1114
自行設計的叢集指標和事件	1123
Memcached 的無伺服器指標和事件	1131
無伺服器指標	1131
無伺服器事件	1134
使用 記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail	1141
中的 Amazon ElastiCache 資訊 CloudTrail	1141
了解 Amazon ElastiCache 日誌檔案項目	1142
Amazon SNS事件監控	1145
管理 ElastiCache Amazon SNS通知	1146
檢視 ElastiCache 事件	1150
事件通知和 Amazon SNS	1154
日誌傳送	1160
慢速日誌項目的內容	1161
引擎日誌項目的內容	1161
設定記錄功能的許可	1161
日誌類型和日誌格式規格	1162
ElastiCache 記錄目的地	1163
使用主控台指定日誌傳送	1165
使用 指定日誌交付 AWS CLI	1166
監控使用情況	1172
主機層級指標	1172

Valkey 和 Redis 的指標 OSS	1175
Memcached 的指標	1188
應監控哪些指標？	1192
選擇指標統計資料與期間	1196
監控 CloudWatch 叢集和節點指標	1196
配額	1200
參考資料	1202
使用 ElastiCache API	1202
使用查詢 API	1202
可用程式庫	1205
對應用程式進行疑難排解	1206
設定 AWS CLI 的 ElastiCache	1207
必要條件	1207
取得命令列工具	1209
設定工具	1209
為工具提供憑證	1210
環境變數	1211
錯誤訊息	1212
通知	1213
一般 ElastiCache 通知	1213
ElastiCache (Memcached) 通知	1213
ElastiCache (RedisOSS) 特定通知	1214
ElastiCache 文件歷史記錄	1215
AWS 詞彙表	1241
.....	mccxliv

什麼是 Amazon ElastiCache ？

歡迎使用 Amazon ElastiCache 使用者指南。Amazon ElastiCache 是一種 Web 服務，可讓您輕鬆地在雲端中設定、管理和擴展分散式記憶體內資料存放區或快取環境。它提供高效能、可擴展且符合成本效益的快取解決方案。同時也有助於消除部署和管理分散式快取環境的相關複雜性。

您可以 ElastiCache 兩種格式操作 Amazon。您可以開始使用無伺服器快取，或選擇自行設計快取叢集。

Note

Amazon ElastiCache 適用於 ValkeyOSS、Redis 和 Memcached 引擎。如果您不確定要使用哪個引擎，請參閱本指南中的[比較 ValkeyOSS、Redis 和 Memcached 自行設計的快取](#)。

無伺服器快取

ElastiCache 提供無伺服器快取，可簡化新增和操作應用程式的快取。ElastiCache Serverless 可讓您在一分鐘內建立高可用性快取，無需佈建執行個體或設定節點或叢集。開發人員可以使用 ElastiCache 主控台 SDK 或指定快取名稱來建立無伺服器快取 CLI。

ElastiCache Serverless 也不需要規劃和管理快取容量。ElastiCache 會持續監控快取的記憶體，運算、應用程式使用的和網路頻寬，和擴展以滿足您應用程式的需求。為開發人員 ElastiCache 提供簡單的端點體驗，透過抽象基礎快取基礎設施和叢集 design. ElastiCache manages 硬體佈建，監控、節點替換、和軟體修補，以便您可以專注於應用程式開發而不是操作快取。

ElastiCache Serverless 與 Valkey 7.2、Redis OSS 7.1 及更高版本，以及 Memcached 1.6.21 及更高版本相容。

設計您自己的 ElastiCache 叢集

如果您需要對 ElastiCache 叢集進行精細控制，您可以選擇使用來設計自己的 Valkey、Redis OSS 或 Memcached 叢集 ElastiCache。ElastiCache 可讓您為叢集選擇節點類型、節點數量，以及跨 AWS 可用區域放置節點，藉此設計叢集。因為 ElastiCache 是完全受管的服務，它會自動管理叢集的硬體佈建、監控、節點替換和軟體修補。

設計您自己的 ElastiCache 叢集可為您的叢集提供更大的彈性和控制。例如，您可以根據需要選擇操作具有單一 AZ 可用性或多 AZ 可用性的叢集。您也可以選擇在叢集模式下執行 Valkey、Redis OSS 或

Memcached，以啟用水平擴展，或選擇不使用叢集模式，只進行垂直擴展。自行設計叢集時，您必須負責正確選擇節點類型和數量，以確保快取擁有足以因應您應用程式所需的容量。您也可以選擇何時將新的軟體修補程式套用至您的 Valkey 或 Redis OSS 叢集。

設計您自己的 ElastiCache 叢集時，您可以選擇執行 Valkey 7.2、Redis OSS 4.0 至 7.1 或 Memcached 1.4 及更高版本。

相關服務

[MemoryDB](#)

決定是否使用 ElastiCache 或 MemoryDB 時，請考慮下列比較：

- ElastiCache 是一項服務，通常用於使用 Valkey、Redis OSS 或 Memcached 快取來自其他資料庫和資料存放區的資料。您應該考慮快取要透過現有主資料庫或資料存放區加速資料存取的工作負載 ElastiCache（微秒讀取和寫入效能）。您也應考慮使用您想要使用 Valkey 或 Redis OSS 資料結構 ElastiCache 的使用案例 APIs，並存取存放在主要資料庫或資料存放區中的資料。
- ElastiCache 也可以透過將經常存取的資料儲存在快取中，協助您節省資料庫成本。如果您的應用程式具有高讀取輸送量需求，您可以使用來實現大規模、快速效能並降低資料儲存成本 ElastiCache，而不是擴展基礎資料庫。
- MemoryDB 是耐用的記憶體內資料庫，適用於需要超快速主要資料庫的工作負載。它與 Valkey 和 Redis 相容 OSS。若您的工作負載需要可提供超快效能（微秒讀取和 1 毫秒寫入延遲）的耐用資料庫，您應該考慮使用 MemoryDB。如果您想要使用 Valkey 或 Redis OSS 資料結構和 APIs 主要耐用資料庫建置應用程式，MemoryDB 可能也適合您使用案例。最後，您應考慮使用 MemoryDB 來簡化您的應用程式架構，並以快取取代使用資料庫，以提高耐久性和效能，進而降低成本。

[Amazon Relational Database Service](#)

ElastiCache 可以透過將經常存取的資料儲存在快取中，協助您節省資料庫成本。如果您的應用程式具有高讀取輸送量需求，您可以使用來實現大規模、快速效能並降低資料儲存成本 ElastiCache，而不是擴展基礎資料庫。

如需相關 Amazon Relational Database Service 服務的進一步背景資訊，請參閱 [Amazon RDS](#)

ElastiCache 可以透過將經常存取的資料儲存在快取中，協助您節省資料庫成本。如果您的應用程式具有高讀取輸送量需求，您可以使用來實現大規模、快速效能並降低資料儲存成本 ElastiCache，而不是擴展基礎資料庫。

ElastiCache 運作方式

您可以在這裡找到 ElastiCache 部署主要元件的概觀。

快取和快取引擎

快取是記憶體內資料存放區，可用來存放快取的資料。一般而言，您的應用程式會在快取中快取經常存取的資料，以最佳化回應時間。ElastiCache 提供兩種部署選項：無伺服器 and 自行設計的叢集。請參閱 [選擇部署選項](#)。

Note

Amazon ElastiCache 適用於 Valkey、Redis OSS 和 Memcached 引擎。如果您不確定要使用哪個引擎，請參閱本指南中的 [比較 ValkeyOSS、Redis 和 Memcached 自行設計的快取](#)。

主題

- [ElastiCache 運作方式](#)
- [定價維度](#)
- [ElastiCache 備份](#)

ElastiCache 運作方式

ElastiCache 無伺服器

ElastiCache Serverless 可讓您建立快取，而不必擔心容量規劃、硬體管理或叢集設計。您只需為您的快取提供名稱，即可收到單一端點，您可以在 Valkey、Redis OSS 或 Memcached 用戶端中設定，以開始存取快取。

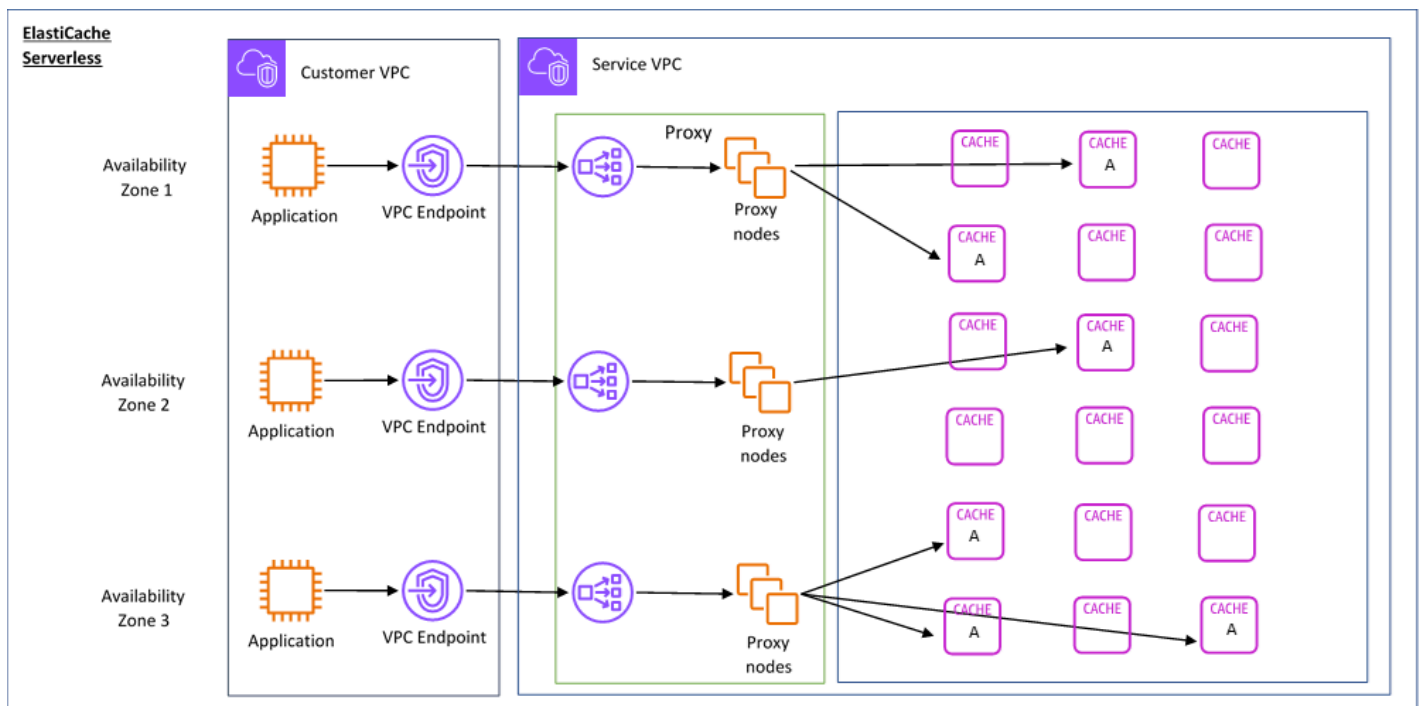
Note

- ElastiCache Serverless 會以叢集模式執行 Valkey、Redis OSS 或 Memcached，且僅與支援的用戶端相容 TLS。

主要優點

- 無容量規劃：無ElastiCache 伺服器可消除規劃容量的需求。無 ElastiCache 伺服器會持續監控快取的記憶體、運算和網路頻寬使用率，並垂直和水平擴展。它可一邊讓快取節點的大小增加，同時一邊啟動橫向擴展操作，以確保快取隨時可因應您應用程式的需求擴展。
- Pay-per-use：使用 ElastiCache Serverless 時，您需為快取上工作負載所儲存和運算的資料付費。請參閱 [定價維度](#)。
- 高可用性：無ElastiCache 伺服器會自動跨多個可用區域（AZ）複寫您的資料，以取得高可用性。它會自動監控底層快取節點，並且在發生故障時進行替換。它為每個快取提供 99.99% SLA 的可用性。
- 自動軟體升級：無ElastiCache 伺服器會自動將快取升級至最新的次要和修補程式軟體版本，而不會對您的應用程式造成任何可用性影響。當新的主要版本可用時，ElastiCache 會傳送通知給您。
- 安全性：Serverless 一律會加密傳輸中的資料和靜態資料。您可以使用服務受管金鑰或使用您自己的客戶受管金鑰來加密靜態資料。

下圖說明 ElastiCache Serverless 的運作方式。



當您建立新的無伺服器快取時，會在您選擇的子網路中 ElastiCache 建立 Virtual Private Cloud (VPC) 端點VPC。您的應用程式可以透過這些VPC端點連線至快取。

使用 ElastiCache Serverless，您會收到應用程式連線的單一DNS端點。當您請求端點的新連線時，ElastiCache Serverless 會透過代理層處理所有快取連線。代理層有助於減少複雜的用戶端組態，因為

在基礎叢集發生變更時，用戶端不需要重新探索叢集拓撲。代理層是一組代理節點，可使用 Network Load Balancer 處理連線。

當您的應用程式建立新的快取連線時，Network Load Balancer 會將請求傳送至代理節點。當您的應用程式執行快取命令時，連線至應用程式的代理節點會在快取中的快取節點上執行請求。代理層會從用戶端抽象化快取叢集拓撲和節點。這 ElastiCache 可讓以智慧方式載入平衡、橫向擴展和新增快取節點、在快取節點失敗時取代快取節點，以及更新快取節點上的軟體，所有這些都不會影響您的應用程式或必須重設連線。

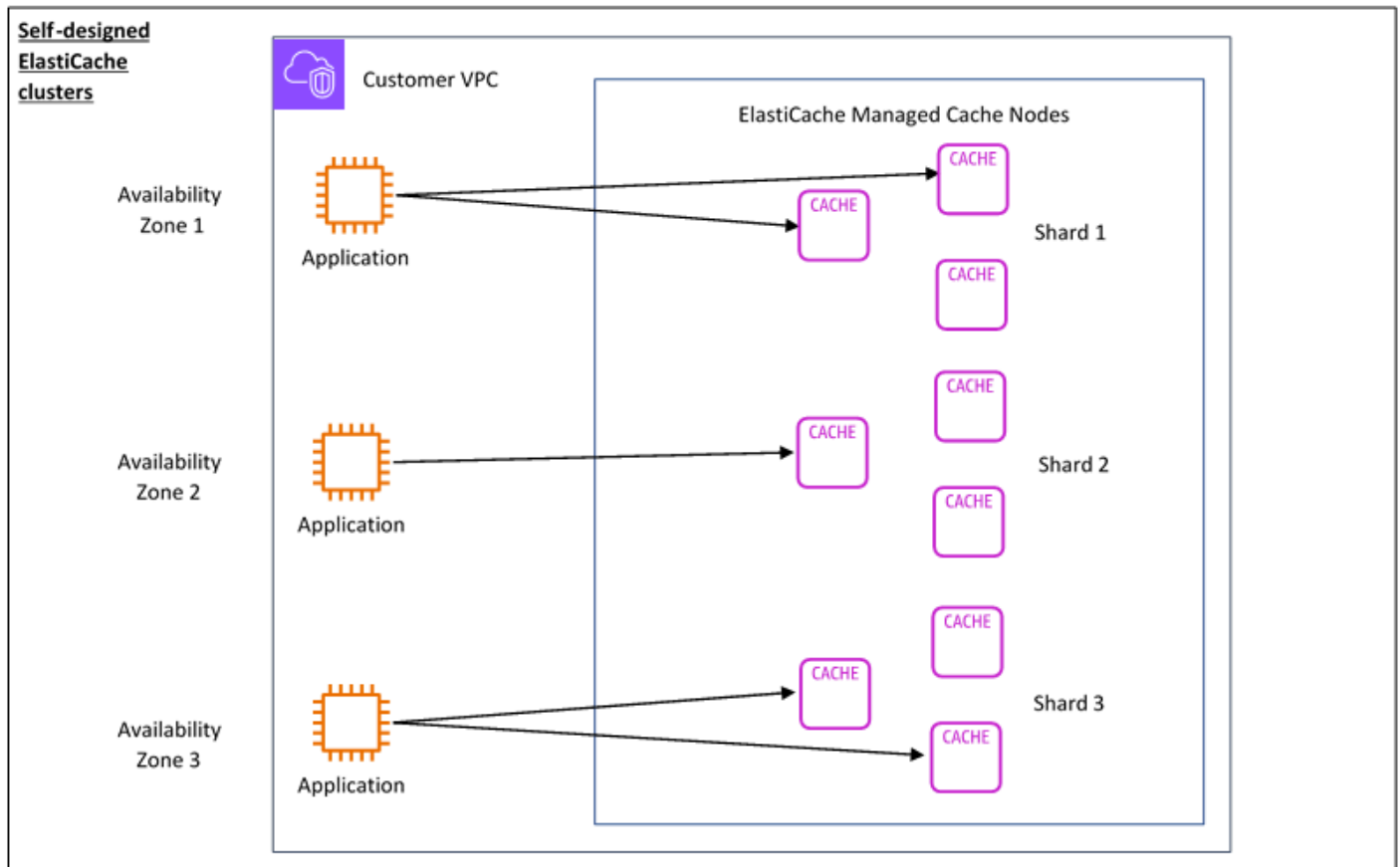
自行設計的 ElastiCache 叢集

您可以選擇快取節點系列、大小和 ElastiCache 叢集的節點數量，來設計自己的叢集。自行設計叢集能讓您更精準控制，並選擇快取中的碎片數目，以及每個碎片中的節點數目 (主要和複本)。您可以選擇 OSS 在叢集模式下操作 Valkey 或 Redis，方法是建立具有多個碎片的叢集，或使用單一碎片在非叢集模式下操作。

主要優點

- **設計您自己的叢集：**使用 ElastiCache，您可以設計自己的叢集，然後選擇要放置快取節點的位置。例如，如果您希望犧牲應用程式的高可用性來換取低延遲，可以選擇在單一 AZ 中部署快取節點。或者，您可以使用多個節點來設計叢集 AZs，以達到高可用性。
- **精細控制：**在自行設計叢集時，您可以透過微調快取的設定來進行更精準的控制。例如，您可以使用 [Valkey 和 Redis OSS 參數](#) 或 [Memcached 專用參數](#) 來設定快取引擎。
- **垂直和水平擴展：**您可以選擇在需要時增加或減少快取節點大小來手動擴展叢集。您也可以透過新增碎片或新增更多複本至碎片來進行水平擴展。您也可以使用 Auto-Scaling 功能，根據排程或快取上的 CPU 和記憶體用量等指標來設定擴展。

下圖說明 ElastiCache 自行設計的叢集如何運作。



定價維度

您可以在兩個部署選項 ElastiCache 中部署。部署 ElastiCache Serverless 時，您需要為以 GB 小時為單位儲存的資料支付用量，並以 ElastiCache 處理器（ECPU）為單位計算。選擇設計自己的 ElastiCache 叢集時，您每小時需支付快取節點用量的費用。請參閱[此處](#)的定價詳細資訊。

資料儲存

您需為儲存在 ElastiCache Serverless 中的資料支付 gigabyte-hours（GB-hrs）的帳單費用。ElastiCache 無伺服器會持續監控儲存在快取中的資料，每分鐘取樣多次，並計算每小時平均值，以判斷快取的資料儲存用量，以 GB 為單位。每個無 ElastiCache 伺服器快取都會計量至少儲存 1 GB 的資料。

ElastiCache 處理單位（ECPUs）

您需為應用程式在 ElastiCache Serverless in ElastiCache Processing Units（ECPUs）上執行的請求付費，該單位包含 vCPU 時間和傳輸的資料。

- 簡單讀取和寫入需要 1 ECPU 個傳輸的資料千位元組 (KB)。例如，傳輸最多 1 KB 資料的 GET 命令會耗用 1 ECPU。傳輸 3.2 KB 資料的 SET 請求將耗用 3.2 ECPUs。
- 使用 Valkey 和 Redis OSS 時，ECPUs 根據兩個維度中的較高者，會耗用更多 vCPU 時間並傳輸更多資料的命令。例如，如果您的應用程式使用 HMGET 命令，會耗用 3 倍的 vCPU 時間作為簡單 SET/GET 命令，並傳輸 3.2 KB 的資料，則會耗用 3.2 ECPU。或者，如果只傳輸 2 KB 的資料，則會耗用 3 個 ECPUs。
- 使用 Valkey 和 Redis 時 OSS，需要額外 vCPU 時間的命令會按比例消耗更多 ECPUs。例如，如果您的應用程式使用 Valkey 或 Redis OSS [HMGET 命令](#)，並且將 vCPU 時間的 3 倍作為簡單 SET/GET 命令，則會消耗 3 倍 ECPUs。
- 使用 Memcached 時，在多個項目上操作的命令會按比例消耗更多 ECPUs。例如，如果您的應用程式對 3 個項目執行多點擷取，則會耗用 3 個 ECPUs。
- 使用 Memcached，ECPUs 根據兩個維度中的較高者，在更多項目上操作並傳輸更多資料消耗的命令。例如，如果您的應用程式使用 GET 命令，會擷取 3 個項目，並傳輸 3.2 KB 的資料，則會耗用 3.2 ECPU。或者，如果只傳輸 2 KB 的資料，則會耗用 3 個 ECPUs。

ElastiCache Serverless 會發出名為 `ElastiCacheProcessingUnits` 的新指標，協助您了解工作負載 ECPUs 耗用的。

節點小時數

您可以選擇 EC2 節點系列、大小、節點數量和跨可用區域的位置，以選擇設計自己的快取叢集。自行設計叢集時，您需以每小時為單位支付每個快取節點的費用。

ElastiCache 備份

備份是 point-in-time 無伺服器快取的副本，或是 Valkey 或 Redis OSS 自行設計的叢集。ElastiCache 可讓您隨時備份資料或設定自動備份。備份可用來還原現有的快取或植入新快取。備份包含快取中的所有資料加上一些中繼資料。如需詳細資訊，請參閱 [快照和還原](#)。

選擇部署選項

Amazon ElastiCache 有兩個部署選項：

- 無伺服器快取
- 自行設計的叢集

如需兩者支援的命令清單，請參閱 [支援和限制的 Valkey、Redis OSS 和 Memcached 命令](#)。

無伺服器快取

Amazon ElastiCache Serverless 可簡化快取建立，並立即擴展以支援客戶最嚴苛的應用程式。透過 ElastiCache Serverless，您可以在一分鐘內建立高可用性和可擴展的快取，無需佈建、規劃和管理快取叢集容量。ElastiCache Serverless 會自動跨三個可用區域以備援方式存放資料，並提供 99.99% 的可用服務層級協議 (SLA)。從自行設計的 Valkey 或 Redis OSS 叢集備份可以還原為無伺服器組態。

自行設計的叢集

如果您需要對 Valkey、Redis OSS 或 Memcached 叢集進行精細控制，您可以選擇使用 來設計自己的叢集 ElastiCache。ElastiCache 可讓您在叢集的 AWS 可用區域之間選擇節點類型、節點數量和節點放置，以操作節點型叢集。因為 ElastiCache 是完全受管的服務，它有助於管理叢集的硬體佈建、監控、節點替換和軟體修補。自行設計的叢集可以提供高達 99.99% 的可用性 SLA。從無伺服器 Valkey 或 Redis OSS 快取的備份可以還原到自行設計的叢集。

選擇部署選項

下列情況請選擇無伺服器快取：

- 您正在為新工作負載或難以預測的工作負載建立快取。
- 您有無法預期的應用程式流量。
- 您希望以最簡單的方式開始使用快取。

選擇在下列情況下設計您自己的 ElastiCache 叢集：

- 您已經在執行 ElastiCache Serverless，並想要更精細地控制執行 Valkey、Redis OSS 或 Memcached 的節點類型、節點數量，以及這些節點的位置。
- 您預期應用程式流量相對可預測，而且您希望精細控制效能、可用性和成本。
- 您可以預測容量需求以控制成本。

比較無伺服器快取和自行設計的叢集

功能	無伺服器快取	自行設計的叢集
快取設定	在一分鐘內僅建立名稱的快取	對快取叢集設計提供精細的控制。使用者可以選擇節點類型、節點數量，以及跨 AWS 可用區域的置放

功能	無伺服器快取	自行設計的叢集
支援的 ElastiCache 版本	Valkey 7.2 和更新版本、Redis 7.1 版和更新OSS版本、Memcached 1.6.21 和更新版本	Valkey 7.2 和更新版本、Redis 4.0 和更新OSS版本、Memcached 1.4 和更新版本
叢集模式 (Valkey 和 RedisOSS)	cluster mode enabled 僅在中操作引擎。用戶端必須支援 cluster mode enabled 才能連線至 ElastiCache Serverless。	可設定為在叢集模式中操作，或停用叢集模式。
擴展	無需任何容量管理，即可以垂直和水平方式自動擴展引擎。	<p>提供對擴展的控制，同時還需要監控，以確保目前的容量足以滿足需求。</p> <p>對於 Valkey 和 Redis OSS，您可以選擇視需要增加或減少快取節點大小，以垂直擴展。您也可以透過新增碎片或將更多複本新增至碎片來水平擴展。此功能不適用於 Memcached。</p> <p>使用 Auto-Scaling 功能，您也可以根據排程來設定擴展，或根據快取上的 CPU 和 記憶體用量等指標來擴展。</p>
用戶端連線	用戶端連接至單一端點。這可讓基礎快取節點拓撲 (擴展、取代和升級) 變更，而不會中斷用戶端連線。	用戶端會連線至每個個別快取節點。如果已取代節點，用戶端會重新探索叢集拓撲，並重新建立連線。

功能	無伺服器快取	自行設計的叢集
可設定性	沒有可用的精細組態。客戶可以設定基本設定，包括可存取快取的子網路、自動備份是開啟還是關閉，以及快取用量限制上限。	自行設計的叢集提供精細的組態選項。客戶可以使用參數群組進行精細控制。如需根據節點類型的參數值表，請參閱 引擎特定參數 。
Multi-AZ	資料會在多個可用區域以非同步方式複寫，以提供更高的可用性並改善讀取延遲。	提供在單一可用區域中或跨多個可用區域 () 設計叢集的選項AZs。使用 Valkey 或 Redis 時OSS，為多可用區域叢集提供跨多個可用區域非同步複寫的資料，以提高可用性並改善讀取延遲。
靜態加密	一律啟用。客戶可以在 中使用 AWS 受管金鑰 或客戶受管金鑰 AWS KMS。	啟用或停用靜態加密的選項。啟用時，客戶可以在 中使用 AWS 受管金鑰 或客戶受管金鑰 AWS KMS。
傳輸中加密 (TLS)	一律啟用。用戶端必須支援 TLS連線。	啟用或停用的選項。
備份	支援快取的自動和手動備份，不會影響效能。 Valkey 和 Redis OSS 備份是跨相容性的，可以還原至無 ElastiCache 伺服器快取或自我設計的叢集。	支援 Valkey 和 Redis 的自動和手動備份OSS。叢集可能會看到一些效能影響，具體取決於可用的預留記憶體。如需詳細資訊，請參閱 管理 Valkey 和 Redis 的預留記憶體 OSS 。 Valkey 和 Redis OSS 備份是跨相容性的，可以還原至無 ElastiCache 伺服器快取或自我設計的叢集。

功能	無伺服器快取	自行設計的叢集
監控	<p>支援快取層級指標，包括快取命中率、快取遺失率、資料大小和ECPUs消耗量。</p> <p>ElastiCache 當快取上發生重大事件 EventBridge 時，Serverless 會使用 傳送事件。您可以選擇使用 Amazon 來監控、擷取、轉換和對 ElastiCache 事件採取行動 EventBridge。如需詳細資訊，請參閱無伺服器快取事件。</p>	<p>ElastiCache 自我設計的叢集在每個節點層級發出指標，包括主機層級指標和快取指標。</p> <p>自行設計的叢集會發出重大事件的SNS通知。請參閱Memcached 的指標 和 Valkey 和 Redis 的指標 OSS。</p>
可用性	<p>99.99% 可用性服務等級協議 (SLA)</p>	<p>自行設計的叢集可以根據組態，達到高達 99.99% 的可用性服務等級協議 (SLA)。</p>
軟體升級和修補	<p>自動將快取軟體升級至最新的次要和修補程式版本，而不會影響應用程式。客戶會收到主要版本升級的通知，而且客戶可以視需要升級至最新的主要版本。</p>	<p>自行設計的叢集提供客戶啟用的自助服務，適用於次要和修補版本升級，以及主要版本升級。受管更新會在客戶定義的維護時段期間自動套用。客戶也可以選擇隨需套用次要或修補程式版本升級。</p>
全球資料存放區	<p>不支援</p>	<p>支援全域資料存放區，可透過單一區域寫入和多區域讀取進行跨區域複寫</p>

功能	無伺服器快取	自行設計的叢集
資料分層	不支援	使用 r6gd 系列節點設計的叢集，其資料會在記憶體和本機 SSD（固態硬碟）儲存之間分層。除了將資料儲存在記憶體之外，資料分層還在每個叢集節點中使用低成本的固態硬碟（SSDs），為 Valkey 和 Redis OSS 工作負載提供價格效能選項。
定價方式	Pay-per-use，根據以 GB 小時為單位儲存的資料和 ElastiCache 以處理器（ECPU）為單位的請求。請參閱 此處 的定價詳細資訊。	Pay-per-hour，根據快取節點用量。請參閱 此處 的定價詳細資訊。

相關主題:

- [設計和管理您自己的 ElastiCache 叢集](#)

適用於首次使用者的 Amazon ElastiCache 資源

我們建議第一次使用者先閱讀下列各節，並視需要參考。

- 服務重點和定價 – [產品詳細資訊頁面](#)提供 ElastiCache、服務重點和定價的一般產品概觀。
- ElastiCache 影片 – [ElastiCache 影片](#)本節提供向您介紹 Amazon 的影片 ElastiCache。這些影片涵蓋 ElastiCache 和 示範如何使用 ElastiCache 來降低延遲並改善應用程式的輸送量的常見使用案例。
- 入門 - 「[Amazon 入門 ElastiCache](#)」一節包含建立快取叢集的相關資訊。其中包含如何授權存取快取叢集、連線至快取節點以及刪除快取叢集的程序。
- 大規模效能 – [Amazon 白皮書的大規模效能 ElastiCache](#)可解決快取策略，協助您的應用程式大規模運作。

完成前面的章節之後，請閱讀這些章節：

- [選擇您的節點大小](#)

建議具有夠大的節點，以包含所有要快取的資料。同時，您不想要支付超出所需快取數量的費用。此主題可協助您選取最佳節點大小。

- [ElastiCache 最佳實務和快取策略](#)

識別並解決可能影響叢集效率的問題。

如果您想要使用 AWS Command Line Interface (AWS CLI)，您可以使用這些文件來協助您開始使用：

- [AWS Command Line Interface 文件](#)

本節提供有關下載 AWS CLI、在系統上 AWS CLI 進行工作，以及提供 AWS 憑證的資訊。

- [AWS CLI 文件 ElastiCache](#)

此個別文件涵蓋 AWS CLI ElastiCache 命令的所有，包括語法和範例。

您可以編寫應用程式，以搭配各種常用程式設計語言使用 ElastiCache API。以下是一些可用資源：

- [適用於 Amazon Web Services 的工具](#)

Amazon Web Services 提供多種支援的軟體開發套件 (SDKs) ElastiCache。您可以使用 ElastiCache Java、.NET、PHP、Ruby 和其他語言編寫程式碼。這些 SDKs 可以大幅簡化應用程式開發，方法是將請求格式化為 ElastiCache、剖析回應，以及提供重試邏輯和錯誤處理。

- [使用 ElastiCache API](#)

如果您不想使用 AWS SDKs，則可以 ElastiCache 直接使用查詢與互動 API。您可以在本節中找到建立與驗證請求與處理回應的故障診斷秘訣和資訊。

- [Amazon ElastiCache API 參考](#)

此個別文件涵蓋所有 ElastiCache API 操作，包括語法和範例。

AWS 區域和可用區域

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

每個 AWS 區域包含多個不同的位置，稱為可用區域，或 AZs。每個可用區域旨在隔離其他可用區域的故障。每個都旨在為相同 AWS 區域中的其他可用區域提供便宜且低延遲的網路連線。藉由在個別的可用區域中啟動執行個體，您就可以保護應用程式免於發生單點故障。如需詳細資訊，請參閱[選擇區域和可用區域](#)。

您可以在多個可用區域中執行叢集，這種方法稱為異地同步備份部署。選擇這個選項後，Amazon 會自動佈建並維護不同可用區域中的次要備用節點執行個體。您的主節點執行個體會同步跨可用區域同步複寫至次要執行個體。此方法協助提供資料備援與容錯移轉支援、消除輸入/輸出凍結，以及在系統備分期間將延遲峰值降至最低。如需詳細資訊，請參閱[使用多可用區將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)。

常見 ElastiCache 使用案例及 ElastiCache 如何提供協助

不論是提供最新的新聞、前 10 名排行榜、產品目錄或是出售活動門票，速度都是遊戲的重點。傳遞內容的速度，會顯著影響您網站和商業的成功與否。

在「[對於耐心不足的 Web 使用者來說，一眨眼的等待時間也嫌太長](#)」報導中，New York Times 指出，使用者可以注意到競爭網站之間 250 毫秒 (1/4 秒) 的差異。使用者傾向於放棄速度較慢的網站，而選擇速度較快的網站。Amazon 完成的測試，引自[網頁載入時間與訪客之間的關聯](#)，指出每載入 100 毫秒 (1/10 秒) 的負載時間，銷售會下降 1%。

當某人想要資料，若這些資料已經過快取，您就可以更迅速提供資料。無論是網頁還是影響商業決策的報告，都適用同樣的道理。您的企業是否能快取您的網頁、以最短的延遲將其傳遞？

很顯然，需要最大的項目，就是您最想要快取的。但為何不快取較不常用的項目？即使是最最佳化的資料庫查詢或遠端API呼叫，也明顯比從記憶體內快取擷取平面金鑰慢。「明顯較慢」這件事，就會造成客戶流失。

下列範例說明使用 ElastiCache 可改善應用程式整體效能的一些方法。

主題

- [記憶體內資料存放區](#)
- [遊戲排行榜](#)
- [訊息 \(Pub/Sub \)](#)
- [建議資料 \(雜湊 \)](#)
- [ElastiCache 客戶見證](#)

記憶體內資料存放區

記憶體內鍵/值存放區的主要目的，是提供超快速 (亞毫秒級延遲) 和經濟實惠的資料複本存取。大多數資料存放區，都具有經常存取但不常更新的資料區域。此外，對資料庫進行查詢，總是比在鍵/值對快取中找出鍵要更慢，且費用更高。執行某些資料庫查詢的費用特別昂貴。其中一個例子是涉及跨多個資料表之聯結的查詢，或具有密集型計算的查詢。透過快取這類查詢結果，您只需支付查詢的一次性費用。然後就可以快速擷取資料多次，而無需重新執行查詢。

我應該快取什麼？

在決定要快取哪些資料時，請考慮下列因素：

速度和費用 - 從資料庫取得資料，總是比從快取要來得慢，且費用更高。某些資料庫查詢在本質上比其他查詢更慢且更昂貴。例如，在多個資料表上執行聯結查詢，與簡單的單一資料表查詢相比，前者的速度明顯較慢且費用更高。如果需要以較慢且費用較高的查詢方式來取得所需資料，則適合改為快取。如果需要以相對快速和簡單的查詢來取得資料，仍可能適合使用快取，具體取決於其他因素。

資料和存取模式 - 判斷要快取的內容也涉及了解資料本身及其存取模式。例如，快取快速變動或很少存取的資料並沒有意義。若要讓快取提供真正的益處，資料應為相對靜態且經常存取。例如社群媒體網站上的個人資料。相反地，如果快取資料不能提供速度或成本優勢，則不建議快取資料。例如，快取會傳回搜尋結果的網頁並沒有意義，因為查詢和結果通常都是獨一無二的。

過時 - 根據定義，快取的資料是過時的資料。即使某些情況下並非過時，仍應該一律視為過時。若要判斷您的資料是否適合快取，您需要判斷應用程式對過時資料的容錯能力。

您的應用程式或許能承受某個內容中的過時資料，但不能承受另一個內容中的過時資料。例如，假設您的網站提供公開交易股票價格。在附有免責聲明，表示可能有 n 分鐘延遲的情況下，您的客戶可能會接受一定程度的過時性。但是，如果是向銷售或購買的經紀人提供股票價格，您需要即時的資料。

如果下列描述成立，便可考慮快取您的資料：

- 與快取擷取相比，取得您資料的速度緩慢或費用高昂。
- 使用者經常存取您的資料。
- 您的資料相對保持無變動，或者資料會快速變動但過時性不夠成大問題。

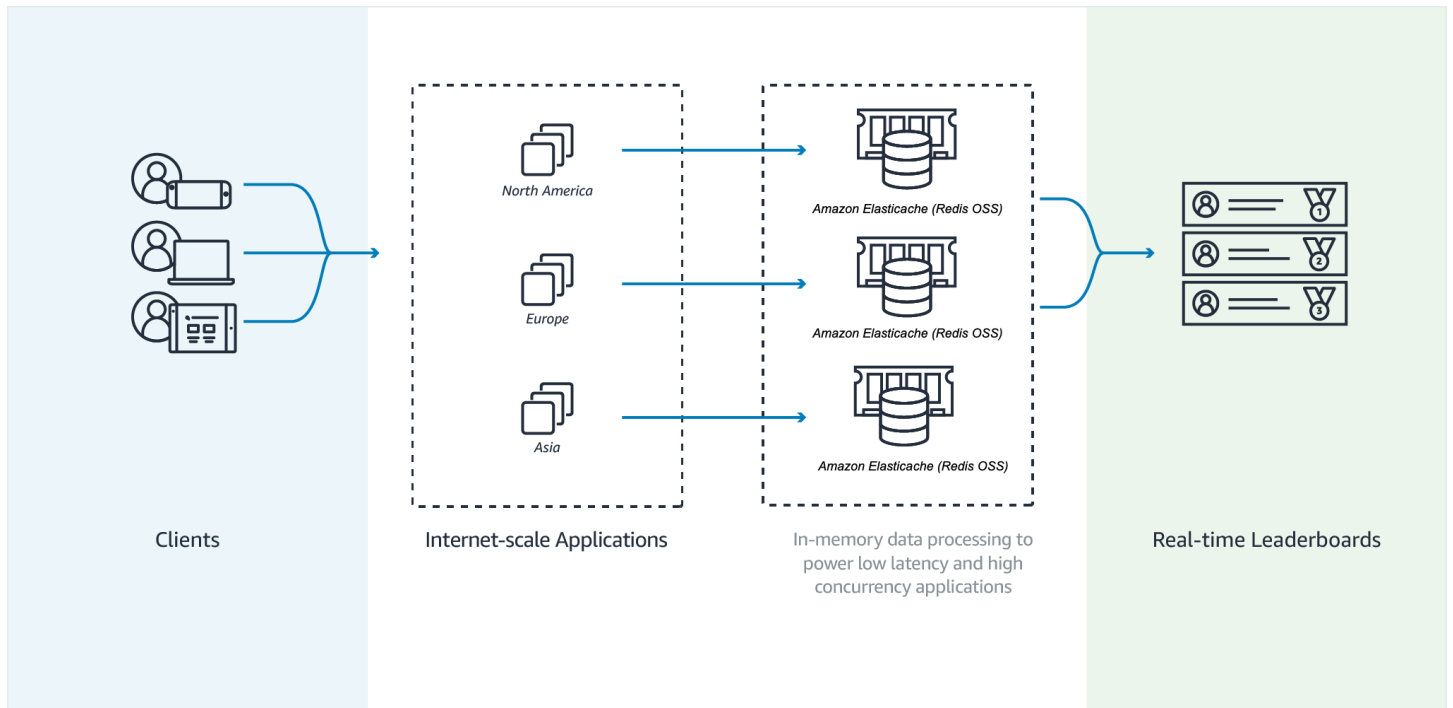
如需詳細資訊，請參閱 [Memcached 的快取策略](#)

遊戲排行榜

使用 Valkey 或 Redis OSS 排序集，您可以將排行榜的運算複雜性從應用程式移至叢集。

排行榜 (例如遊戲得分的前 10 名) 的運算方式很複雜。當有大量玩家同時遊戲且分數不斷變動時，運算尤其複雜。Valkey 和 Redis OSS 排序集保證唯一性和元素排序。使用排序集，每次將新元素新增至排序集時，都會即時重新排序。然後會以正確的數字順序新增至集合。

在下圖中，您可以看到 ElastiCache 遊戲排行榜的運作方式。



Example Valkey 或 Redis OSS 排行榜

在此範例中，四名玩家及其分數會透過 ZADD 輸入排序清單中。ZREVRANGEBYSCORE 命令會依照分數列出玩家，由高至低。接下來，ZADD 會用於更新 June 的分數 (透過覆寫現有項目)。最後，ZREVRANGEBYSCORE 會依照分數由高至低列出玩家。清單顯示 June 的排名已上升。

```
ZADD leaderboard 132 Robert
ZADD leaderboard 231 Sandra
ZADD leaderboard 32 June
ZADD leaderboard 381 Adam

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) Sandra
3) Robert
4) June

ZADD leaderboard 232 June

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) June
3) Sandra
4) Robert
```

下列命令可讓 June 得知她在所有玩家中的排名。由於排名為零，因此 6 月會 ZREVRANK 傳回 1，其處於第二個位置。

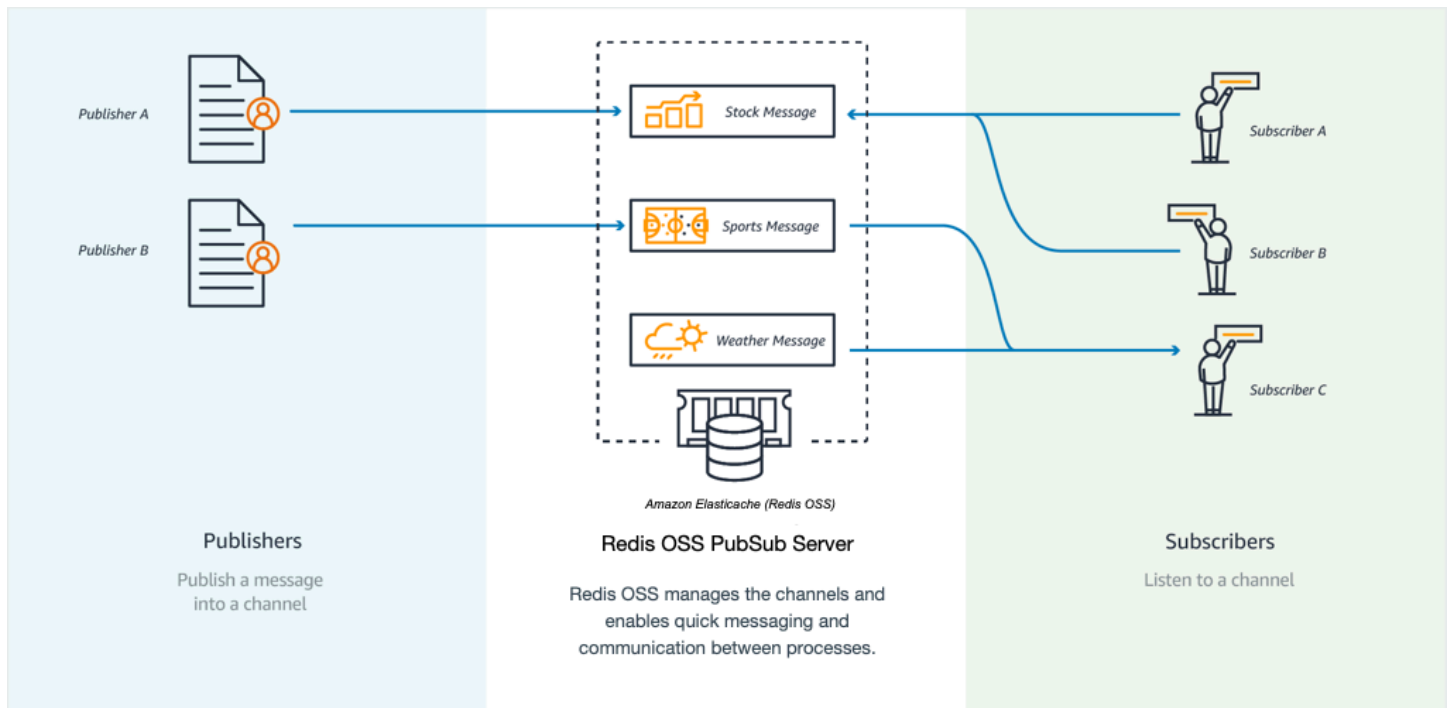
```
ZREVRANK leaderboard June
1
```

如需詳細資訊，請參閱有關排序集的 [Valkey 文件](#)。

訊息 (Pub/Sub)

當您傳送電子郵件時，您會將郵件傳送給一或多個指定的收件人。在 Valkey 和 Redis OSS pub/sub 範式中，您會將訊息傳送至特定頻道，不知道誰會收到訊息。訂閱該頻道的人員會收到訊息。例如，假設您訂閱了 news.sports.golf 頻道。您和 news.sports.golf 頻道的其他訂閱者，都會收到發佈至 news.sports.golf 頻道的所有訊息。

Pub/sub 功能與任何金鑰空間無關。因此，在任何層級上都不會造成干擾。在下圖中，您可以找到使用 Valkey 和 Redis 傳送訊息的 ElastiCache 圖例 OSS。



訂閱

若要接收某頻道的訊息，您需訂閱該頻道。您可以訂閱單一頻道、多個頻道或所有匹配某種模式的頻道。若要取消訂閱，請從您訂閱的指定頻道取消訂閱。或者，如果您使用模式比對功能進行訂閱，取消訂閱時也會使用與之前相同的模式。

Example - 訂閱單一頻道

若要訂閱單一頻道，請使用 SUBSCRIBE 命令來指定您要訂閱的頻道。在下列範例中，用戶端訂閱了 news.sports.golf 頻道。

```
SUBSCRIBE news.sports.golf
```

一段時間後，用戶端會使用指定取消訂閱之頻道的 UNSUBSCRIBE 命令來取消對頻道的訂閱。

```
UNSUBSCRIBE news.sports.golf
```

Example - 訂閱多個特定頻道

若要訂閱多個特定頻道，請使用 SUBSCRIBE 命令列出頻道。在下列範例中，用戶端同時訂閱了 news.sports.golf、news.sports.soccer 和 news.sports.skiing 頻道。

```
SUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

若要取消特定頻道的訂閱，請使用 UNSUBSCRIBE 命令並指定要取消訂閱的頻道。

```
UNSUBSCRIBE news.sports.golf
```

若要取消多個頻道的訂閱，請使用 UNSUBSCRIBE 命令並指定要取消訂閱的頻道。

```
UNSUBSCRIBE news.sports.golf news.sports.soccer
```

若要取消所有訂閱，請使用 UNSUBSCRIBE 並指定每個頻道。或使用 UNSUBSCRIBE 且不指定頻道。

```
UNSUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

或

```
UNSUBSCRIBE
```

Example - 使用模式匹配的訂閱

用戶端可以使用 PSUBSCRIBE 命令訂閱符合模式的所有頻道。

在下列範例中，用戶端訂閱了所有體育頻道。您不需要分別列出所有體育頻道，如同使用 SUBSCRIBE。而是可透過 PSUBSCRIBE 命令使用模式比對功能。

```
PSUBSCRIBE news.sports.*
```

Example 取消訂閱

若要取消訂閱這些頻道，請使用 PUNSUBSCRIBE 命令。

```
PUNSUBSCRIBE news.sports.*
```

Important

傳送至 **【P】** SUBSCRIBE 命令和傳送至 **【P】** UNSUBSCRIBE 命令的頻道字串必須相符。您不能對 news.* 使用 PSUBSCRIBE、對 news.sports.* 使用 PUNSUBSCRIBE，也不能對 news.sports.golf 使用 UNSUBSCRIBE。

發布

若要向某頻道中的所有訂閱者傳送訊息，請使用 PUBLISH 命令，指定頻道和訊息。下列範例將「這是個晴朗的星期六，我要前往連結了。」訊息發佈到 news.sports.golf 頻道。

```
PUBLISH news.sports.golf "It's Saturday and sunny. I'm headed to the links."
```

用戶端無法發佈至訂閱的頻道。

如需詳細資訊，請參閱 Valkey 文件中的 [Pub/Sub](#)。

建議資料 (雜湊)

在 Valkey INCR 或 Redis DECR 中使用或 OSS 可讓編譯建議變得簡單。當每次有使用者「喜歡」某項產品時，會遞增 item:productID:like 計數器。當每次有使用者「不喜歡」某項產品時，會遞增 item:productID:dislike 計數器。使用雜湊，您也可以維護喜歡或不喜歡產品的每個人清單。

Example - 喜歡和不喜歡

```
INCR item:38923:likes  
HSET item:38923:ratings Susan 1
```

```
INCR item:38923:dislikes  
HSET item:38923:ratings Tommy -1
```

ElastiCache 客戶見證

若要了解 Airbnb、PBSEsri 等企業如何使用 Amazon ElastiCache 來擴展業務，並改善客戶體驗，請參閱 [Others 如何使用 Amazon。ElastiCache](#)

您也可以觀看 [教學影片](#)，了解其他 ElastiCache 客戶使用案例。

Amazon 入門 ElastiCache

使用本節中的實作教學課程，協助您開始使用 並進一步了解如何使用 ElastiCache。

主題

- [設定 ElastiCache](#)
- [建立 Valkey 無伺服器快取](#)
- [建立 Valkey 或 Redis 無OSS伺服器快取](#)
- [建立 Memcached 無伺服器快取](#)
- [教學課程：開始使用 Python 和 ElastiCache](#)
- [教學課程：設定 Lambda 在 ElastiCache 中存取 VPC](#)

設定 ElastiCache

若要使用 ElastiCache Web 服務，請遵循下列步驟。

主題

- [註冊 AWS 帳戶](#)
- [建立具有管理存取權的使用者](#)
- [授與程式設計存取權](#)
- [設定您的許可（僅限新 ElastiCache 使用者）](#)
- [設定 EC2](#)
- [將網路存取權從 Amazon VPC安全群組授予快取](#)
- [下載並設定命令列存取](#)

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟以建立。

若要註冊 AWS 帳戶

1. 開啟<https://portal.aws.amazon.com/billing/註冊>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，AWS 帳戶根使用者會建立。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

AWS 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時前往 <https://aws.amazon.com/> 並選擇我的帳戶 來檢視目前的帳戶活動和管理帳戶。

建立具有管理存取權的使用者

註冊後 AWS 帳戶，請保護您的 AWS 帳戶根使用者、啟用 AWS IAM Identity Center 並建立管理使用者，這樣您就不會將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者[AWS Management Console](#)身分登入。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 為您的根使用者開啟多重要素驗證 (MFA)。

如需指示，請參閱 IAM 使用者指南 中的[為 AWS 帳戶根使用者 \(主控台\) 啟用虛擬MFA裝置](#)。

建立具有管理存取權的使用者

1. 啟用IAM身分中心。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，將管理存取權授予使用者。

如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱 AWS IAM Identity Center 使用者指南 中的[使用 設定使用者存取權 IAM Identity Center 目錄](#)。

以具有管理存取權的使用者身分登入

- 若要使用 IAM Identity Center 使用者登入，請使用您建立 IAM Identity Center 使用者時URL傳送到您電子郵件地址的登入。

如需使用 IAM Identity Center 使用者登入的協助，請參閱 AWS 登入 使用者指南 中的[登入 AWS 存取入口網站](#)。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立遵循套用最低權限許可最佳實務的許可集。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[建立許可集](#)。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[新增群組](#)。

授與程式設計存取權

如果使用者想要與 AWS 外部互動，則需要程式設計存取權 AWS Management Console。授予程式設計存取權的方式取決於存取的使用者類型 AWS。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 (在 IAM Identity Center 中管理的使用者)	使用暫時憑證簽署對 AWS CLI、AWS SDKs、或的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> 對於 AWS CLI，請參閱 使用者指南 中的設定 AWS CLI 要使用 AWS IAM Identity Center的。AWS Command Line Interface 如需 AWS SDKs、工具和 AWS APIs，請參閱 AWS SDKs和 工具參考指南 中的IAM身分中心身分驗證。

哪個使用者需要程式設計存取權？	到	By
IAM	使用暫時憑證簽署對 AWS CLI、AWS SDKs、或的程式設計請求 AWS APIs。	請遵循 IAM 使用者指南 中的 將臨時憑證與 AWS 資源搭配使用 的指示。
IAM	(不建議使用) 使用長期憑證簽署對 AWS CLI、AWS SDKs、或的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> 對於 AWS CLI，請參閱 AWS Command Line Interface 使用者指南 中的 使用 IAM 使用者憑證進行驗證。 如需 AWS SDKs 和工具，請參閱 AWS SDKs 和工具參考指南 中的 使用長期憑證進行身分驗證。 對於 AWS APIs，請參閱 IAM 使用者指南 中的 管理 IAM 使用者的存取金鑰。

相關主題：

- IAM 使用者指南 中的 [內容 IAM](#)。
- AWS 一般參考 中的 [AWS 安全憑證](#)。

設定您的許可（僅限新 ElastiCache 使用者）

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 [建立權限合集](#) 說明進行操作。

- IAM 透過身分提供者在 中管理的使用者：

建立聯合身分的角色。請遵循 IAM 使用者指南 中 [為第三方身分提供者（聯合）建立角色](#) 的指示。

- IAM 使用者：
 - 建立您的使用者可擔任的角色。請遵循 IAM 使用者指南 中 [為IAM使用者建立角色](#) 的指示。
 - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南 中的 [將許可新增至使用者（主控台）](#) 中的指示。

Amazon ElastiCache 會建立並使用服務連結角色來佈建資源，並代表您存取其他 AWS 資源和服務。若要 ElastiCache 為您建立服務連結角色，請使用名為 `AmazonElastiCacheFullAccess` 的 AWS 受管政策。此角色隨附了預先佈建、服務代表您建立服務連結角色所需的許可。

您可能決定不使用預設的政策，而是改為使用自訂的受管政策。在此情況下，請確定您有呼叫的許可，`iam:createServiceLinkedRole` 或已建立 ElastiCache 服務連結角色。

如需詳細資訊，請參閱下列內容：

- [建立新政策（IAM）](#)
- [AWS Amazon 的受管政策 ElastiCache](#)
- [使用 Amazon 的服務連結角色 ElastiCache](#)

設定 EC2

您需要設定執行個體，EC2 從中連線至快取。

- 如果您還沒有 EC2 執行個體，請在此處了解如何設定 EC2 執行個體：[開始使用 EC2](#)。
- 您的 EC2 執行個體必須位於相同的 VPC 中，且具有與快取相同的安全群組設定。根據預設，Amazon 會在您的預設中 ElastiCache 建立快取 VPC，並使用預設安全群組。若要遵循本教學課程，請確保您的 EC2 執行個體處於預設狀態，VPC 且具有預設安全群組。

將網路存取權從 Amazon VPC 安全群組授予快取

ElastiCache 自我設計的叢集使用連接埠 6379 作為 Valkey 和 Redis OSS 命令，而無 ElastiCache 伺服器則同時使用連接埠 6379 和連接埠 6380。為了成功從 EC2 執行個體連線和執行 Valkey 或 Redis OSS 命令，您的安全群組必須允許視需要存取這些連接埠。

ElastiCache (Memcached) 使用 11211 和 11212 連接埠來接受 Memcached 命令。為了成功連接和執行來自 EC2 執行個體的 Memcached 命令，您的安全群組必須允許存取這些連接埠。

1. 登入 AWS Command Line Interface 並開啟 [Amazon EC2 主控台](#)。
2. 在導覽窗格中，在 Network & Security (網路與安全) 下，選擇 Security Groups (安全群組)。
3. 從安全群組清單中，選擇 Amazon 的安全群組 VPC。除非您建立安全群組以供 ElastiCache 使用，否則此安全群組將命名為預設。
4. 選擇「傳入」索引標籤，然後：
 - a. 選擇編輯。
 - b. 選擇新增規則。
 - c. 在類型欄中，選擇自訂 TCP 規則。
 - d. 如果使用 Valkey 或 Redis OSS，請在連接埠範圍方塊中，輸入 6379。

如果使用 Memcached，則在連接埠範圍方塊中，鍵入 11211。

- e. 在來源方塊中，選擇具有連接埠範圍 (0.0.0.0/0) 的任何地方，以便在 Amazon 中啟動的任何 Amazon EC2 執行個體 VPC 都可以連線至快取。
- f. 如果您使用的是無 ElastiCache 伺服器，請選擇新增規則 來新增另一個規則。
- g. 在類型欄中，選擇自訂 TCP 規則。
- h. 如果使用 ElastiCache (Redis OSS)，則在連接埠範圍方塊中，輸入 6380。

如果使用 ElastiCache (Memcached)，則在連接埠範圍方塊中，輸入 11212。

- i. 在來源方塊中，選擇具有連接埠範圍 (0.0.0.0/0) 的任何地方，以便在 Amazon 中啟動的任何 Amazon EC2 執行個體 VPC 都可以連線至快取。
- j. 選擇儲存

下載並設定命令列存取

下載並安裝 valkey-cli 公用程式。

如果您 ElastiCache 搭配 Valkey 使用，您可能會發現 valkey-cli 公用程式很有用。如果您使用 ElastiCache (Redis OSS) 搭配 redis-cli，請考慮切換到 valkey-cli，因為它 OSS 也適用於 Redis。

1. 使用您選擇的連線公用程式連線至您的 Amazon EC2 執行個體。如需如何連線至 Amazon EC2 執行個體的指示，請參閱 [Amazon EC2 入門指南](#)。
2. 執行適用於設定的命令，以下載並安裝 valkey-cli 公用程式。

Amazon Linux 2023

```
sudo yum install redis6 -y
```

Amazon Linux 2

```
sudo amazon-linux-extras install epel -y
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make BUILD_TLS=yes
```

Note

- 當您安裝 redis6 套件時，它會安裝 redis6-cli 並提供預設加密支援。
- 安裝 valkey-cli 或 redis-cli TLS時，請務必取得 的建置支援。只有在 TLS 啟用 時才能存取 ElastiCache Serverless。
- 如果您要連線至未加密的叢集，則不需要 Build_TLS=yes 選項。

建立 Valkey 無伺服器快取

在此步驟中，您可以在 Amazon 中建立新的快取 ElastiCache。

AWS Management Console

若要使用 ElastiCache 主控台建立新的快取：

1. 登入 AWS Management Console 並開啟 <https://console.aws.amazon.com/connect/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 快取。
3. 在主控台右側，選擇建立 Valkey 快取
4. 在快取設定中，輸入名稱。您可以選擇性地輸入快取的說明。
5. 讓預設設定保持選取狀態。
6. 按一下建立以建立快取。

7. 一旦快取處於「ACTIVE」狀態，您就可以開始將資料寫入快取並讀取。

AWS CLI

下列 AWS CLI 範例會使用 建立新的快取 create-serverless-cache。

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine valkey
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine valkey
```

請注意，「狀態」欄位的值會設定為 CREATING。

若要驗證 ElastiCache 是否已完成建立快取，請使用 describe-serverless-caches 命令。

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

建立新快取後，繼續進行 [讀取和寫入資料至快取](#)。

讀取和寫入資料至快取

本節假設您已建立 Amazon EC2 執行個體，且可與其連線。如需如何執行此操作的指示，請參閱 [Amazon EC2 入門指南](#)。

本節也假設您具有要從中連線至快取之 EC2 執行個體的設定 VPC 存取和安全群組設定，以及在 EC2 執行個體上設定 valkey-cli。如需該步驟的詳細資訊，請參閱 [設定 ElastiCache](#)。

尋找您的快取端點

AWS Management Console

若要使用 ElastiCache 主控台尋找快取的端點：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 快取。
3. 在控制台的右側，按一下您剛建立的快取名稱。
4. 在快取詳細資訊中，尋找並複製快取端點。

AWS CLI

下列 AWS CLI 範例顯示 使用 describe-serverless-caches 命令尋找新快取的端點。執行命令後，查詢「端點」欄位。

Linux

```
aws elasticache describe-serverless-caches \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
  --serverless-cache-name CacheName
```

連線至您的 Valkey 快取 (Linux)

現在您已擁有所需的端點，您可以登入 EC2 執行個體並連線至快取。在下列範例中，您可以使用 valkey-cli 公用程式連線到叢集。下列命令會連線至快取 (注意：將 cache-endpoint 取代為您在上一步驟中擷取的端點)。

```
src/valkey-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

連線至您的 Valkey 快取 (Windows)

現在您已擁有所需的端點，您可以登入EC2執行個體並連線至快取。在下列範例中，您可以使用 valkey-cli 公用程式連線到叢集。下列命令會連線至快取。開啟命令提示字元並變更為 Valkey 或 Redis OSS目錄，然後執行命令（注意：將 Cache_Endpoint 取代為您在上一步中擷取的端點）。

```
c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint --tls -p 6379
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
```

現在您可以繼續進行 [\(選用\) 清除](#)。

(選用) 清除

如果您不再需要您建立的 Amazon ElastiCache 快取，則可以將其刪除。此步驟有助於確保您不會為了未使用的資源而付費。您可以使用 ElastiCache 主控台、AWS CLI或 ElastiCache API來刪除快取。

AWS Management Console

使用主控台來刪除快取：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 快取。
3. 選擇您要刪除之快取旁的選項按鈕。
4. 選取右上角的動作，然後選取刪除。
5. 您可以選擇在刪除快取之前，先拍攝最終快照。
6. 在刪除確認畫面中，再次輸入快取名稱並選擇刪除以刪除叢集，或選擇取消以保留叢集。

一旦快取移至 DELETING 狀態，您就會停止產生該快取的費用。

AWS CLI

下列 AWS CLI 範例會使用 delete-serverless-cache 命令刪除快取。

Linux

```
aws elasticache delete-serverless-cache \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^  
  --serverless-cache-name CacheName
```

請注意，狀態欄位的值設定為 DELETING。

現在您可以繼續進行 [後續步驟](#)。

後續步驟

如需詳細資訊，ElastiCache 請參閱下列頁面：

- [使用 ElastiCache](#)
- [擴展 ElastiCache](#)
- [在 Amazon 中記錄和監控 ElastiCache](#)
- [ElastiCache 最佳實務和快取策略](#)
- [快照和還原](#)
- [Amazon ElastiCache 事件SNS監控](#)

建立 Valkey 或 Redis 無OSS伺服器快取

在此步驟中，您可以在 Amazon 中建立新的快取 ElastiCache。

AWS Management Console

若要使用 ElastiCache 主控台建立新的快取：

1. 登入 AWS Management Console 並開啟 <https://console.aws.amazon.com/connect/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 快取或 Redis OSS快取。
3. 在主控台右側，選擇建立 Valkey 快取或建立 Redis OSS快取
4. 在快取設定中，輸入名稱。您可以選擇性地輸入快取的說明。
5. 讓預設設定保持選取狀態。

- 按一下建立以建立快取。
- 一旦快取處於「ACTIVE」狀態，您就可以開始將資料寫入快取並讀取。

AWS CLI

下列 AWS CLI 範例會使用 建立新的快取 `create-serverless-cache`。

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine redis
```

請注意，「狀態」欄位的值會設定為 `CREATING`。

若要驗證 ElastiCache 是否已完成建立快取，請使用 `describe-serverless-caches` 命令。

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

建立新快取後，繼續進行 [讀取和寫入資料至快取](#)。

讀取和寫入資料至快取

本節假設您已建立 Amazon EC2 執行個體，且可與其連線。如需如何執行此操作的指示，請參閱 [Amazon EC2 入門指南](#)。

本節也假設您具有要從中連線至快取之 EC2 執行個體的設定 VPC 存取和安全群組設定，以及在 EC2 執行個體上設定 `valkey-cli`。如需該步驟的詳細資訊，請參閱 [設定 ElastiCache](#)。

尋找您的快取端點

AWS Management Console

若要使用 ElastiCache 主控台尋找快取的端點：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 快取 Redis OSS快取。
3. 在控制台的右側，按一下您剛建立的快取名稱。
4. 在快取詳細資訊中，尋找並複製快取端點。

AWS CLI

下列 AWS CLI 範例顯示 使用 describe-serverless-caches 命令尋找新快取的端點。執行命令後，查詢「端點」欄位。

Linux

```
aws elasticache describe-serverless-caches \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
  --serverless-cache-name CacheName
```

連線至您的 Valkey 或 Redis OSS 快取 (Linux)

現在您已擁有所需的端點，您可以登入EC2執行個體並連線至快取。在下列範例中，您可以使用 valkey-cli 公用程式連線到叢集。下列命令會連線至快取 (注意：將 cache-endpoint 取代為您在上一步驟中擷取的端點)。

```
src/valkey-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

連線至您的 Valkey 或 Redis OSS 快取 (Windows)

現在您已擁有所需的端點，您可以登入 EC2 執行個體並連線至快取。在下列範例中，您可以使用 `valkey-cli` 公用程式連線到叢集。下列命令會連線至快取。開啟命令提示字元並變更為 Valkey 目錄，然後執行命令（注意：將 `Cache_Endpoint` 取代為您在上一個步驟中擷取的端點）。

```
c:\Redis>valkey-cli -h Redis_Cluster_Endpoint --tls -p 6379
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
```

現在您可以繼續進行 [\(選用\) 清除](#)。

(選用) 清除

如果您不再需要您建立的 Amazon ElastiCache 快取，則可以將其刪除。此步驟有助於確保您不會為了未使用的資源而付費。您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 來刪除快取。

AWS Management Console

使用主控台來刪除快取：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Valkey 或 Redis OSS 快取。
3. 選擇您要刪除之快取旁的選項按鈕。
4. 選取右上角的動作，然後選取刪除。
5. 您可以選擇在刪除快取之前，先拍攝最終快照。
6. 在刪除確認畫面中，再次輸入快取名稱並選擇刪除以刪除叢集，或選擇取消以保留叢集。

一旦快取移至 DELETING 狀態，您就會停止產生該快取的費用。

AWS CLI

下列 AWS CLI 範例會使用 `delete-serverless-cache` 命令刪除快取。

Linux

```
aws elasticache delete-serverless-cache \
```



```
--serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^  
--serverless-cache-name CacheName
```

請注意，狀態欄位的值設定為 DELETING。

現在您可以繼續進行 [後續步驟](#)。

後續步驟

如需詳細資訊，ElastiCache 請參閱下列頁面：

- [使用 ElastiCache](#)
- [擴展 ElastiCache](#)
- [在 Amazon 中記錄和監控 ElastiCache](#)
- [ElastiCache 最佳實務和快取策略](#)
- [快照和還原](#)
- [Amazon ElastiCache 事件SNS監控](#)

建立 Memcached 無伺服器快取

AWS Management Console

若要使用 ElastiCache 主控台建立新的 Memcached 無伺服器快取：

1. 登入 AWS Management Console 並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Memcached 快取。
3. 在主控台的右側，選擇建立 Memcached 快取。
4. 在快取設定中，輸入名稱。您可以選擇性地輸入快取的說明。
5. 讓預設設定保持選取狀態。
6. 按一下建立以建立快取。
7. 一旦快取處於「ACTIVE」狀態，您就可以開始將資料寫入快取並讀取。

若要使用 建立新的快取 AWS CLI

下列 AWS CLI 範例會使用 建立新的快取 `create-serverless-cache`。

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine memcached
```

請注意，「狀態」欄位的值會設定為 `CREATING`。

若要驗證 ElastiCache 是否已完成建立快取，請使用 `describe-serverless-caches` 命令。

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

建立新快取後，繼續進行 [讀取和寫入資料至快取](#)。

讀取和寫入資料至快取

本節假設您已建立 Amazon EC2 執行個體，且可與其連線。如需如何執行此操作的指示，請參閱 [Amazon EC2 入門指南](#)。

根據預設，會在您的預設中 ElastiCache 建立快取 VPC。請確定您的 EC2 執行個體也在預設中建立 VPC，以便能夠連線至快取。

尋找您的快取端點

AWS Management Console

若要使用 ElastiCache 主控台尋找快取的端點：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Memcached 快取。
3. 在控制台的右側，按一下您剛建立的快取名稱。
4. 在快取詳細資訊中，尋找並複製快取端點。

AWS CLI

下列 AWS CLI 範例顯示 使用 describe-serverless-caches 命令尋找新快取的端點。執行命令後，查詢「端點」欄位。

Linux

```
aws elasticache describe-serverless-caches \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
  --serverless-cache-name CacheName
```

使用 Open 連線SSL

如需如何使用 Open 連線的資訊SSL，請參閱 [ElastiCache 傳輸中加密 \(TLS \)](#)

使用 Memcached Java 用戶端進行連線

如需如何使用 Memcached Java 用戶端進行連線的詳細資訊，請參閱 [ElastiCache 傳輸中加密 \(TLS \)](#)

使用 Memcached PHP用戶端連線

```
<?php  
$cluster_endpoint = "mycluster.serverless.use1.cache.amazonaws.com";  
$server_port = 11211;  
  
/* Initialize a persistent Memcached client in TLS mode */  
$tls_client = new Memcached('persistent-id');
```

```
$tls_client->addServer($cluster_endpoint, $server_port);
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.serverless.us-east-1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
$tls_client->createAndSetTLSContext((array)$tls_config);

/* store the data for 60 seconds in the cluster */
$tls_client->set('key', 'value', 60);
?>
```

使用 Memcached Python 用戶端 (Pymemcache) 進行連線

請參閱 https://pymemcache.readthedocs.io/en/latest/getting_started.html

```
import ssl
from pymemcache.client.base import Client

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
target_port = 11211
memcached_client = Client("{cluster_endpoint}", target_port, tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

使用 Memcached NodeJS/TS 用戶端 (Electrode-IO memcache) 進行連線

請參閱 <https://github.com/electrode-io/memcache> 和 <https://www.npmjs.com/package/memcache-client>

透過 npm i memcache-client 進行安裝

在應用程式中，建立 Memcached TLS用戶端，如下所示：

```
var memcache = require("memcache-client");
const client = new memcache.MemcacheClient({server: "{cluster_endpoint}:11211", tls:
  {}});
client.set("key", "value");
```

使用 Memcached Rust 用戶端 (rust-memcache) 進行連線

請參閱 <https://crates.io/crates/memcache> 和 <https://github.com/aisk/rust-memcache>。

```
// create connection with to memcached server node:
let client = memcache::connect("memcache+tls://<cluster_endpoint>:11211?
verify_mode=none").unwrap();

// set a string value
client.set("foo", "bar", 0).unwrap();
```

使用 Memcached Go 用戶端 (Gomemcache) 進行連線

請參閱 <https://github.com/bradfitz/gomemcache>

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

使用 Memcached Ruby 用戶端 (Dalli) 進行連線

請參閱 <https://github.com/petergoldstein/dalli>

```
require 'dalli'
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_context.ssl_version = :SSLv23
ssl_context.verify_hostname = true
ssl_context.verify_mode = OpenSSL::SSL::VERIFY_PEER
client = Dalli::Client.new("<cluster_endpoint>:11211", :ssl_context => ssl_context);
client.get("abc")
```

使用 Memcached .NET client (EnyimMemcachedCore) 連線

請參閱 <https://github.com/cnblogs/EnyimMemcachedCore>

```
"MemcachedClient": {  
  "Servers": [  
    {  
      "Address": "{cluster_endpoint}",  
      "Port": 11211  
    }  
  ],  
  "UseSslStream": true  
}
```

現在您可以繼續進行 [\(選用\) 清除](#)。

(選用) 清除

使用 AWS Management Console

以下程序會從您的部署中刪除單一快取。若要刪除多個快取，請針對每個要刪除的快取重複此程序。您不需要等待某個快取完成刪除，即可開始刪除其他快取。

刪除快取

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在 ElastiCache 主控台儀表板中，選擇您要刪除的快取正在執行的引擎。接著會出現執行該引擎的所有快取清單。
3. 若要選擇要刪除的快取，請從快取清單中選擇快取的名稱。

Important

您一次只能從 ElastiCache 主控台刪除一個快取。選擇多個快取會停用刪除操作。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 在刪除快取確認畫面中，選擇刪除以刪除快取，或選擇取消以保留叢集。
6. 如果您選擇刪除，快取的狀態就會變更為正在刪除。

一旦快取移至 DELETING 狀態，您就會停止產生該快取的費用。

使用 AWS CLI

下列程式碼會刪除 my-cache 快取。

```
aws elasticache delete-serverless-cache --serverless-cache-name my-cache
```

此 delete-serverless-cache CLI 動作只會刪除一個無伺服器快取。若要刪除多個快取，請呼叫 delete-serverless-cache 您要刪除的每個無伺服器快取。您不需要等待某個無伺服器快取完成刪除，即可刪除其他快取。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name my-cache
```

用於 Windows :

```
aws elasticache delete-serverless-cache ^  
  --serverless-cache-name my-cache
```

如需詳細資訊，請參閱 ElastiCache 主題 AWS CLI 的 delete-serverless-cache。

現在您可以繼續進行 [後續步驟](#)。

後續步驟

如需詳細資訊，ElastiCache 請參閱：

- [使用 ElastiCache](#)
- [擴展 ElastiCache](#)
- [的配額 ElastiCache](#)
- [ElastiCache 最佳實務和快取策略](#)
- [檢視 ElastiCache 事件](#)

教學課程：開始使用 Python 和 ElastiCache

本節包含實作教學課程，協助您了解 ElastiCache Valkey 和 Redis OSS。我們鼓勵您遵循其中一個特定語言的教學。

Note

AWS SDKs 適用於各種語言。如需取得完整清單，請參閱 [Amazon Web Services 工具](#)。

主題

- [Python 和 ElastiCache](#)

Python 和 ElastiCache

在本教學課程中，您可以使用 for Python AWS SDK (Boto3) 編寫簡單的程式來執行下列 ElastiCache (RedisOSS) 操作：

- 建立 ElastiCache (Redis OSS) 叢集 (啟用叢集模式和停用叢集模式)

- 檢查使用者或使用者群組是否存在，否則請建立它們。（此功能適用於 Valkey 7.2 及更新版本，以及 Redis OSS 6.0 及更新版本。）
- 連線至 ElastiCache
- 執行各種作業，例如設定和取得字串、讀取和寫入串流，以及從發佈/訂閱頻道發佈和訂閱。

完成本教學課程時，您可以參考 for Python AWS SDK (Boto) 文件。以下章節是 特有的 ElastiCache：[ElastiCache 低階用戶端](#)

教學課程事前準備

- 設定 AWS 存取金鑰以使用 AWS SDKs。如需詳細資訊，請參閱[設定 ElastiCache](#)。
- 安裝 Python 3.0 或更新版本。如需詳細資訊，請參閱 <https://www.python.org/downloads>。如需說明，請參閱 Boto 3 文件中的[快速入門](#)。

主題

- [教學課程：建立 ElastiCache 叢集和使用者](#)
- [教學課程：連線至 ElastiCache](#)
- [使用範例](#)

教學課程：建立 ElastiCache 叢集和使用者

下列範例使用 boto3 SDK 進行 ElastiCache (Redis OSS) 管理操作 (叢集或使用者建立) 和重新解散/redis-py-cluster 進行資料處理。

主題

- [建立停用叢集模式的叢集](#)
- [使用 TLS 和 建立停用叢集模式的叢集 RBAC](#)
- [建立啟用叢集模式的叢集](#)
- [使用 TLS 和 建立啟用叢集模式的叢集 RBAC](#)
- [檢查使用者/使用者群組是否存在，若不存在則加以建立](#)

建立停用叢集模式的叢集

複製下列程式，並將其貼到名為 CreateClusterModeDisabledCluster.py 的檔案中。

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
create_cluster_mode_disabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheClusters=1,
cache_cluster',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
node) and maximum 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'

    response = client.create_replication_group(
        AutomaticFailoverEnabled=True,
        CacheNodeType=CacheNodeType,
        Engine='valkey',
        EngineVersion=EngineVersion,
        NumCacheClusters=NumCacheClusters,
        ReplicationGroupDescription=ReplicationGroupDescription,
        ReplicationGroupId=ReplicationGroupId,
        SnapshotRetentionLimit=30,
    )
    return response
```

```
if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
    nodes, Valkey 7.2, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104053'
    )

    logging.info(elasticacheResponse)
```

若要執行程式，請輸入下列命令：

```
python CreateClusterModeDisabledCluster.py
```

如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

使用 TLS和 建立停用叢集模式的叢集 RBAC

為了確保安全，您可以在建立停用叢集模式的叢集時，使用 Transport Layer Security (TLS) 和角色型存取控制 (RBAC)。與 Valkey 或 Redis OSS 不同AUTH，在驗證所有已驗證的用戶端的權杖時，其具有完整複寫群組存取權，RBAC可讓您透過使用者群組控制叢集存取權。這些使用者群組的設計目的是整理複寫群組的存取權。如需詳細資訊，請參閱[角色型存取控制 \(RBAC \)](#)。

複製下列程式，並將其貼到名為 ClusterModeDisabledWithRBAC.py 的檔案中。

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheC
    cache cluster', ReplicationGroupId=None, UserGroupIds=None,
    SecurityGroupIds=None, CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response
```

```
:param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/CacheNodes.SupportedTypes.html for supported node types
:param EngineVersion: Engine version to be used. If not specified, latest will be
used.
:param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
node) and maximum 6 (1 primary and 5 replicas).
If not specified, cluster will be created with 1 primary and 1 replica.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Mandatory name for the cluster.
:param UserGroupIds: The ID of the user group to be assigned to the cluster.
:param SecurityGroupIds: List of security groups to be assigned. If not defined,
default will be used
:param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
defined, default will be used.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return {'Error': 'ReplicationGroupId parameter is required'}
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'valkey',
        'EngineVersion': EngineVersion,
        'NumCacheClusters': NumCacheClusters,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds
        }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response
```

```
if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
    nodes, Valkey 7.2, one primary and two replicas.
    # Assigns the existent user group "mygroup" for RBAC authentication

    response=create_cluster_mode_disabled_rbac(
        CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104',
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'
    )

    logging.info(response)
```

若要執行程式，請輸入下列命令：

```
python ClusterModeDisabledWithRBAC.py
```

如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

建立啟用叢集模式的叢集

複製下列程式，並將其貼到名為 ClusterModeEnabled.py 的檔案中。

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
    ReplicationGroupDescription='Sample cache with cluster mode
    enabled', ReplicationGroupId=None):
```

```

"""Creates an ElastiCache Cluster with cluster mode enabled

Returns a dictionary with the API response

:param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/CacheNodes.SupportedTypes.html for supported node types
:param EngineVersion: Engine version to be used. If not specified, latest will be
used.
:param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'

response = client.create_replication_group(
    AutomaticFailoverEnabled=True,
    CacheNodeType=CacheNodeType,
    Engine='valkey',
    EngineVersion=EngineVersion,
    ReplicationGroupDescription=ReplicationGroupDescription,
    ReplicationGroupId=ReplicationGroupId,
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
node (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=NumNodeGroups,
    ReplicasPerNodeGroup=ReplicasPerNodeGroup,
    CacheParameterGroupName='default.valkey7.2.cluster.on'
)

return response

# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Valkey cluster mode enabled with replicas',

```

```
    ReplicationGroupId='valkey20210',
#   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
  (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=2,
    ReplicasPerNodeGroup=1,
  )

logging.info(response)
```

若要執行程式，請輸入下列命令：

```
python ClusterModeEnabled.py
```

如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

使用 TLS和 建立啟用叢集模式的叢集 RBAC

為了確保安全，您可以在建立啟用叢集模式的叢集時，使用 Transport Layer Security (TLS) 和角色型存取控制 (RBAC)。與 Valkey 或 Redis OSS 不同 AUTH，在驗證所有已驗證的用戶端的權杖時，其具有完整複寫群組存取權，RBAC可讓您透過使用者群組控制叢集存取權。這些使用者群組的設計目的是整理複寫群組的存取權。如需詳細資訊，請參閱[角色型存取控制 \(RBAC \)](#)。

複製下列程式，並將其貼到名為 ClusterModeEnabledWithRBAC.py 的檔案中。

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticsearch')

def
  create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
  ReplicationGroupDescription='Sample cache with cluster
  mode enabled', ReplicationGroupId=None, UserGroupIds=None,
  SecurityGroupIds=None, CacheSubnetGroupName=None, CacheParameterGroupName='default.valkey7.2.clu
  """"Creates an ElastiCache Cluster with cluster mode enabled and RBAC

  Returns a dictionary with the API response

  :param CacheNodeType: Node type used on the cluster. If not specified,
  cache.t3.small will be used
  Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
  CacheNodes.SupportedTypes.html for supported node types
```

```
:param EngineVersion: Engine version to be used. If not specified, latest will be
used.
:param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster.
:param CacheParameterGroupName: Parameter group to be used. Must be compatible with
the engine version and cluster mode enabled.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'valkey',
        'EngineVersion': EngineVersion,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds,
        'NumNodeGroups': NumNodeGroups,
        'ReplicasPerNodeGroup': ReplicasPerNodeGroup,
        'CacheParameterGroupName': CacheParameterGroupName
    }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
```



```
CacheNodeType='cache.m6g.large',
EngineVersion='7.2',
ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
ReplicationGroupId='valkey2021',
# Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
(implicit) and 2 replicas (replicasPerNodeGroup)
NumNodeGroups=2,
ReplicasPerNodeGroup=1,
UserGroupIds=[
    'mygroup'
],
SecurityGroupIds=[
    'sg-7cc73803'
],
CacheSubnetGroupName='default'

)

logging.info(response)
```

若要執行程式，請輸入下列命令：

```
python ClusterModeEnabledWithRBAC.py
```

如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

檢查使用者/使用者群組是否存在，若不存在則加以建立

透過 RBAC，您可以使用存取字串建立使用者並為其指派特定許可。您可以將使用者指派給與特定角色（管理員、人力資源）一致的使用者群組，然後部署到一或多個 ElastiCache（RedisOSS）複寫群組。透過這樣做，您可以使用相同的 Valkey 或 Redis OSS 複寫群組在用戶端之間建立安全界限，並防止用戶端存取彼此的資料。如需詳細資訊，請參閱[角色型存取控制（RBAC）](#)。

複製下列程式，並將其貼到名為 UserAndUserGroups.py 的檔案中。更新提供憑證的機制。此範例中的憑證顯示為可取代，並且有指派的未宣告項目。避免對憑證進行硬式編碼。

此範例使用具有使用者許可的存取字串。如需存取字串的詳細資訊，請參閱[使用存取字串指定許可](#)。

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticsearch')
```

```
def check_user_exists(UserId):
    """Checks if UserId exists

    Returns True if UserId exists, otherwise False
    :param UserId: ElastiCache User ID
    :return: True|False
    """
    try:
        response = client.describe_users(
            UserId=UserId,
        )
        if response['Users'][0]['UserId'].lower() == UserId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def check_group_exists(UserGroupId):
    """Checks if UserGroupId exists

    Returns True if Group ID exists, otherwise False
    :param UserGroupId: ElastiCache User ID
    :return: True|False
    """

    try:
        response = client.describe_user_groups(
            UserGroupId=UserGroupId
        )
        if response['UserGroups'][0]['UserGroupId'].lower() == UserGroupId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserGroupNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def create_user(UserId=None, Username=None, Password=None, AccessString=None):
    """Creates a new user
```

```
Returns the ARN for the newly created user or the error message
:param UserId: ElastiCache user ID. User IDs must be unique
:param UserName: ElastiCache user name. ElastiCache allows multiple users with the
same name as long as the associated user ID is unique.
:param Password: Password for user. Must have at least 16 chars.
:param AccessString: Access string with the permissions for the user.
:return: user ARN
"""
try:
    response = client.create_user(
        UserId=UserId,
        UserName=UserName,
        Engine='Redis',
        Passwords=[Password],
        AccessString=AccessString,
        NoPasswordRequired=False
    )
    return response['ARN']
except Exception as e:
    logging.info(e.response['Error'])
    return e.response['Error']

def create_group(UserGroupId=None, UserIds=None):
    """Creates a new group.
    A default user is required (mandatory) and should be specified in the UserIds list

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
            Engine='Redis',
            UserIds=UserIds
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])

if __name__ == '__main__':
```

```
groupName='mygroup2'
userName = 'myuser2'
userId=groupName+'-'+userName

# Creates a new user if the user ID does not exist.
for tmpUserId,tmpUserName in [ (userId,userName), (groupName+'-
default','default')]:
    if not check_user_exists(tmpUserId):
        response=create_user(UserId=tmpUserId,
UserName=EXAMPLE,Password=EXAMPLE,AccessString='on ~* +@all')
        logging.info(response)
    # assigns the new user ID to the user group
if not check_group_exists(groupName):
    UserIds = [ userId , groupName+'-default']
    response=create_group(UserGroupId=groupName,UserIds=UserIds)
    logging.info(response)
```

若要執行程式，請輸入下列命令：

```
python UserAndUserGroups.py
```

教學課程：[連線至 ElastiCache](#)

下列範例使用 Valkey 或 Redis OSS 用戶端連線至 ElastiCache。

主題

- [連線至停用叢集模式的叢集](#)
- [連線至啟用叢集模式的叢集](#)

連線至停用叢集模式的叢集

複製下列程式，並將其貼到名為 ConnectClusterModeDisabled.py 的檔案中。更新提供憑證的機制。此範例中的憑證顯示為可取代，並且有指派的未宣告項目。避免對憑證進行硬式編碼。

```
from redis import Redis
import logging

logging.basicConfig(level=logging.INFO)
redis = Redis(host='primary.xxx.yyyyyy.zzz1.cache.amazonaws.com', port=6379,
decode_responses=True, ssl=True, username=example, password=EXAMPLE)
```

```
if redis.ping():
    logging.info("Connected to Redis")
```

若要執行程式，請輸入下列命令：

```
python ConnectClusterModeDisabled.py
```

連線至啟用叢集模式的叢集

複製下列程式，並將其貼到名為 ConnectClusterModeEnabled.py 的檔案中。

```
from rediscluster import RedisCluster
import logging

logging.basicConfig(level=logging.INFO)
redis = RedisCluster(startup_nodes=[{"host":
    "xxx.yyy.clustercfg.zzz1.cache.amazonaws.com", "port": "6379"}],
    decode_responses=True, skip_full_coverage_check=True)

if redis.ping():
    logging.info("Connected to Redis")
```

若要執行程式，請輸入下列命令：

```
python ConnectClusterModeEnabled.py
```

使用範例

下列範例使用 boto3 SDK ElastiCache 讓使用 ElastiCache (Redis OSS)。

主題

- [設定和取得字串](#)
- [設定和取得具有多個項目的雜湊](#)
- [從發佈/訂閱頻道發佈 \(寫入\) 和訂閱 \(讀取\)](#)
- [從串流寫入和讀取](#)

設定和取得字串

複製下列程式，並將其貼到名為 SetAndGetStrings.py 的檔案中。

```
import time
import logging
logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
currTime=time.ctime(time.time())

# Set the key 'mykey' with the current date and time as value.
# The Key will expire and removed from cache in 60 seconds.
redis.set(keyName, currTime, ex=60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieve the key value and current TTL
keyValue=redis.get(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
keyValue, keyTTL))
```

若要執行程式，請輸入下列命令：

```
python SetAndGetStrings.py
```

設定和取得具有多個項目的雜湊

複製下列程式，並將其貼到名為 SetAndGetHash.py 的檔案中。

```
import logging
import time

logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
keyValues={'datetime': time.ctime(time.time()), 'epochtime': time.time()}

# Set the hash 'mykey' with the current date and time in human readable format
# (datetime field) and epoch number (epochtime field).
redis.hset(keyName, mapping=keyValues)

# Set the key to expire and removed from cache in 60 seconds.
redis.expire(keyName, 60)
```

```
# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieves all the fields and current TTL
keyValues=redis.hgetall(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
keyValues, keyTTL))
```

若要執行程式，請輸入下列命令：

```
python SetAndGetHash.py
```

從發佈/訂閱頻道發佈 (寫入) 和訂閱 (讀取)

複製下列程式，並將其貼到名為 PubAndSub.py 的檔案中。

```
import logging
import time

def handlerFunction(message):
    """Prints message got from PubSub channel to the log output

    Return None
    :param message: message to log
    """
    logging.info(message)

logging.basicConfig(level=logging.INFO)
redis = Redis(host="redis202104053.tihewd.ng.0001.use1.cache.amazonaws.com", port=6379,
decode_responses=True)

# Creates the subscriber connection on "mychannel"
subscriber = redis.psubsub()
subscriber.subscribe(**{'mychannel': handlerFunction})

# Creates a new thread to watch for messages while the main process continues with its
routines
thread = subscriber.run_in_thread(sleep_time=0.01)

# Creates publisher connection on "mychannel"
```

```
redis.publish('mychannel', 'My message')

# Publishes several messages. Subscriber thread will read and print on log.
while True:
    redis.publish('mychannel',time.ctime(time.time()))
    time.sleep(1)
```

若要執行程式，請輸入下列命令：

```
python PubAndSub.py
```

從串流寫入和讀取

複製下列程式，並將其貼到名為 ReadWriteStream.py 的檔案中。

```
from redis import Redis
import redis.exceptions as exceptions
import logging
import time
import threading

logging.basicConfig(level=logging.INFO)

def writeMessage(streamName):
    """Starts a loop writting the current time and thread name to 'streamName'

    :param streamName: Stream (key) name to write messages.
    """
    fieldsDict={'writerId':threading.currentThread().getName(),'myvalue':None}
    while True:
        fieldsDict['myvalue'] = time.ctime(time.time())
        redis.xadd(streamName,fieldsDict)
        time.sleep(1)

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
    used to coordinate data distribution.
    Once a thread acknowleges the message, it won't be provided again. If message
    wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group were multiple threads will read.
    :param streamName: Stream (key) name where messages will be read.
    """
```



```
readerID=threading.currentThread().getName()
while True:
    try:
        # Check if the stream has any message
        if redis.xlen(streamName)>0:
            # Check if if the messages are new (not acknowledged) or not (already
processed)
            streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
            if len(streamData) > 0:
                msgId,message = streamData[0][1][0]
                logging.info("{}: Got {} from ID
{}".format(readerID,message,msgId))
                #Do some processing here. If the message has been processed
sucessfully, acknowledge it and (optional) delete the message.
                redis.xack(streamName,groupName,msgId)
                logging.info("Stream message ID {} read and processed successfully
by {}".format(msgId,readerID))
                redis.xdel(streamName,msgId)
            else:
                pass
        except:
            raise

        time.sleep(0.5)

# Creates the stream 'mystream' and consumer group 'myworkergroup' where multiple
threads will write/read.
try:
    redis.xgroup_create('mystream','myworkergroup',mkstream=True)
except exceptions.ResponseError as e:
    logging.info("Consumer group already exists. Will continue despite the error:
{}".format(e))
except:
    raise

# Starts 5 writer threads.
for writer_no in range(5):
    writerThread = threading.Thread(target=writeMessage, name='writer-'+str(writer_no),
args=('mystream',),daemon=True)
    writerThread.start()

# Starts 10 reader threads
```

```
for reader_no in range(10):
    readerThread = threading.Thread(target=readMessage, name='reader-'+str(reader_no),
    args=('myworkergroup', 'mystream', ), daemon=True)
    readerThread.daemon = True
    readerThread.start()

# Keep the code running for 30 seconds
time.sleep(30)
```

若要執行程式，請輸入下列命令：

```
python ReadWriteStream.py
```

教學課程：設定 Lambda 在 ElastiCache 中存取 VPC

在本教學課程中，您可以了解如何建立無 ElastiCache 伺服器快取、建立 Lambda 函數，然後測試 Lambda 函數，然後選擇性地清除。

主題

- [步驟 1：建立無 ElastiCache 伺服器快取。](#)
- [步驟 2：為 建立 Lambda 函數 ElastiCache](#)
- [步驟 3：使用 測試 Lambda 函數 ElastiCache](#)
- [步驟 4：清除（選用）](#)

步驟 1：建立無 ElastiCache 伺服器快取。

若要建立無伺服器快取，請遵循下列步驟。

步驟 1.1：建立無伺服器快取

在此步驟中，您可以使用 AWS Command Line Interface () 在帳戶中 us-east-1 區域中的預設 Amazon VPC 中建立無伺服器快取 CLI。如需使用 ElastiCache 主控台或 建立無伺服器快取的資訊 API，請參閱 [建立 Valkey 無伺服器快取](#)。

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --description "ElastiCache IAM auth application" \  
  --vpc-subnet-ids subnet-12345678
```

```
--engine valkey
```

請注意，「狀態」欄位的值會設定為 CREATING。可能需要一分鐘 ElastiCache 的時間才能完成建立快取。

步驟 1.2：複製無伺服器快取端點

確認 ElastiCache (Redis OSS) 已完成使用 `describe-serverless-caches` 命令建立快取。

```
aws elasticache describe-serverless-caches \  
--serverless-cache-name cache-01
```

複製輸出中所顯示的端點地址。您為 Lambda 函數建立部署套件時，會需要這個地址。

步驟 1.3：建立 IAM 角色

1. 為您的角色建立 IAM 信任政策文件，允許您的帳戶擔任新角色，如下所示。將政策儲存到名為 `trust-policy.json` 的檔案。

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
    "Action": "sts:AssumeRole"  
  },  
  {  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "lambda.amazonaws.com"  
    },  
    "Action": "sts:AssumeRole"  
  }  
}]  
}
```

2. 建立 IAM 政策文件，如下所示。將政策儲存到名為 `policy.json` 的檔案。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": "lambda:InvokeFunction",  
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:my-function",  
      "Effect": "Allow"  
    }  
  ]  
}
```

```
{
  "Effect" : "Allow",
  "Action" : [
    "elasticache:Connect"
  ],
  "Resource" : [
    "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",
    "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
  ]
}
```

3. 建立 IAM 角色。

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. 建立IAM政策。

```
aws iam create-policy \
--policy-name "elasticache-allow-all" \
--policy-document file://policy.json
```

5. 將IAM政策連接至角色。

```
aws iam attach-role-policy \
--role-name "elasticache-iam-auth-app" \
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

步驟 1.4 : 建立無伺服器快取

1. 建立新的預設使用者。

```
aws elasticache create-user \
--user-name default \
--user-id default-user-disabled \
--engine redis \
--authentication-mode Type=no-password-required \
--access-string "off +get ~keys*"
```

2. 建立新的IAM啟用的使用者。

```
aws elasticache create-user \  
  --user-name iam-user-01 \  
  --user-id iam-user-01 \  
  --authentication-mode Type=iam \  
  --engine redis \  
  --access-string "on ~* +@all"
```

3. 建立使用者群組並連接使用者。

```
aws elasticache create-user-group \  
  --user-group-id iam-user-group-01 \  
  --engine redis \  
  --user-ids default-user-disabled iam-user-01  
  
aws elasticache modify-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --user-group-id iam-user-group-01
```

步驟 2：為 建立 Lambda 函數 ElastiCache

若要建立 Lambda 函數以存取 ElastiCache 快取，請執行下列步驟。

步驟 2.1：建立 Lambda 函數

在本教學課程中，我們會在 Python 中提供 Lambda 函數的範例程式碼。

Python

下列 Python 程式碼範例會讀取項目並將其寫入 ElastiCache 快取。複製程式碼並將它儲存到名為 app.py 的檔案中。請務必將程式碼中的elasticache_endpoint值取代為您在上一步驟中複製的端點地址。

```
from typing import Tuple, Union  
from urllib.parse import ParseResult, urlencode, urlunparse  
  
import boto3.session  
import redis  
from boto3.model import ServiceId  
from boto3.signers import RequestSigner  
from cachetools import TTLCache, cached
```

```
import uuid

class ElastiCacheIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cache_name, is_serverless=False, region="us-east-1"):
        self.user = user
        self.cache_name = cache_name
        self.is_serverless = is_serverless
        self.region = region

        session = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )

    # Generated IAM tokens are valid for 15 minutes
    @cached(cache=TTLCache(maxsize=128, ttl=900))
    def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
        query_params = {"Action": "connect", "User": self.user}
        if self.is_serverless:
            query_params["ResourceType"] = "ServerlessCache"
        url = urlunparse(
            ParseResult(
                scheme="https",
                netloc=self.cache_name,
                path="/",
                query=urlencode(query_params),
                params="",
                fragment="",
            )
        )
        signed_url = self.request_signer.generate_presigned_url(
            {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
            operation_name="connect",
            expires_in=900,
            region_name=self.region,
        )
        # RequestSigner only seems to work if the URL has a protocol, but
        # Elasticache only accepts the URL without a protocol
        # So strip it off the signed URL before returning
```

```
        return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.us-east-1.cache.amazonaws.com" #
    replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
    is_serverless=True)
    redis_client = redis.Redis(host=elasticache_endpoint, port=6379,
    credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Valkey.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
        {decoded_result}")

    return "Fetched value from Valkey"
```

此程式碼使用 Python redis-py 程式庫，將項目放入快取並擷取它們。此程式碼使用 cachetools IAM 將產生的身分驗證權杖快取 15 分鐘。若要建立包含重新解散和 cachetool 的部署套件，請執行下列步驟。

在包含 app.py 原始程式碼檔案的專案目錄中，建立資料夾套件以安裝 redis-py 和 cachetools 程式庫。

```
mkdir package
```

使用 pip 安裝 redis-py、Cachetools。

```
pip install --target ./package redis
pip install --target ./package cachetools
```

建立包含 redis-py 和 cachetools 程式庫的 .zip 檔案。在 Linux 和 macOS 中，執行下列命令。在 Windows 中，使用您偏好的 zip 公用程式，在根目錄建立具有 redis-py 和 cachetools 程式庫的 .zip 檔案。

```
cd package
zip -r ../my_deployment_package.zip .
```

將您的函數程式碼新增至 .zip 檔案。在 Linux 和 macOS 中，執行下列命令。在 Windows 中，使用您偏好的 zip 公用程式，將 app.py 新增至 .zip 檔案的根目錄。

```
cd ..
zip my_deployment_package.zip app.py
```

步驟 2.2：建立 IAM 角色（執行角色）

將名為 `AWSLambdaVPCLambdaAccessExecutionRole` 的 AWS 受管政策 `AWSLambdaVPCLambdaAccessExecutionRole` 連接至 角色。

```
aws iam attach-role-policy \
  --role-name "elasticache-iam-auth-app" \
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCLambdaAccessExecutionRole"
```

步驟 2.3：上傳部署套件 (建立 Lambda 函數)

在此步驟中，您可以使用 `create-function` AWS CLI 命令建立 Lambda 函數（`AccessValkey`）。

從包含部署套件 .zip 檔案的專案目錄中，執行下列 `create-function` Lambda CLI 命令。

對於角色選項，請使用您在上一個步驟中建立 ARN 的執行角色的。針對 `vpc-config`，輸入預設 VPC 子網路的逗號分隔清單，以及預設 VPC 的安全群組 ID。您可以在 Amazon VPC 主控台中找到這些值。若要尋找預設 VPC 的子網路，請選擇您的 VPCs，然後選擇您 AWS 帳戶的預設 VPC。若要尋找此的安全群組 VPC，請前往安全並選擇安全群組。確認已選取 `us-east-1` 區域。

```
aws lambda create-function \
  --function-name AccessValkey \
  --region us-east-1 \
  --zip-file fileb://my_deployment_package.zip \
  --role arn:aws:iam::123456789012:role/elasticache-iam-auth-app \
  --handler app.lambda_handler \
  --runtime python3.12 \
  --timeout 30 \
```



```
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-  
security-group-id
```

步驟 3：使用 測試 Lambda 函數 ElastiCache

在此步驟中，您可以使用叫用命令手動叫用 Lambda 函數。當 Lambda 函數執行時，它會產生，UUID 並將其寫入您在 Lambda 程式碼中指定的 ElastiCache 快取。然後 Lambda 函數從快取中取回項目。

1. 使用叫用命令叫用 Lambda 函數 AWS Lambda (AccessValkey)。

```
aws lambda invoke \  
--function-name AccessValkey \  
--region us-east-1 \  
output.txt
```

2. 確認 Lambda 函數是否成功執行，如下：

- 檢視 output.txt 檔案。
- 開啟 CloudWatch 主控台並選擇函數的日誌群組 (/aws/lambda/)，以驗證 CloudWatch 日誌中的結果 AccessValkey。日誌串流應包含類似以下的輸出內容：

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched  
826e70c5f4d2478c8c18027125a3e01e from Valkey.
```

- 在 AWS Lambda 主控台中檢閱結果。

步驟 4：清除 (選用)

若要清除，請執行下列步驟。

步驟 4.1：刪除 Lambda 函數

```
aws lambda delete-function \  
--function-name AccessValkey
```

步驟 4.2：刪除無伺服器快取

刪除快取。

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name cache-01
```

移除使用者和使用者群組。

```
aws elasticache delete-user \  
--user-id default-user-disabled  
  
aws elasticache delete-user \  
--user-id iam-user-01  
  
aws elasticache delete-user-group \  
--user-group-id iam-user-group-01
```

步驟 4.3 : 移除IAM角色和政策

```
aws iam detach-role-policy \  
--role-name "elasticache-iam-auth-app" \  
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"  
  
aws iam detach-role-policy \  
--role-name "elasticache-iam-auth-app" \  
--policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"  
  
aws iam delete-role \  
--role-name "elasticache-iam-auth-app"  
  
aws iam delete-policy \  
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

設計和管理您自己的 ElastiCache 叢集

如果您需要精細控制 ElastiCache 叢集，您可以選擇設計自己的叢集。ElastiCache 可讓您在叢集的 AWS 可用區域之間選擇節點類型、節點數量和節點放置，以操作節點型叢集。因為 ElastiCache 是完全受管的服務，它會自動管理叢集的硬體佈建、監控、節點替換和軟體修補。

如需設定的詳細資訊，請參閱 [設定 ElastiCache](#)。如需管理、更新或刪除節點或叢集的詳細資訊，請參閱 [在中管理節點 ElastiCache](#)。如需在設計自己的 ElastiCache 叢集時 Amazon ElastiCache 部署的主要元件概觀，請參閱這些[關鍵概念](#)。

主題

- [ElastiCache 元件和功能](#)
- [ElastiCache 術語](#)
- [教學課程：如何設計自己的叢集](#)
- [刪除叢集](#)
- [其他 ElastiCache 教學課程和影片](#)
- [在中管理節點 ElastiCache](#)
- [在中管理叢集 ElastiCache](#)
- [比較 ValkeyOSS、Redis 和 Memcached 自行設計的快取](#)
- [Valkey 或 Redis 的線上遷移 OSS](#)
- [選擇的區域和可用區域 ElastiCache](#)

ElastiCache 元件和功能

接下來，您可以找到 Amazon ElastiCache 部署主要元件的概觀。

主題

- [ElastiCache 節點](#)
- [ElastiCache 碎片](#)
- [ElastiCache 叢集](#)
- [ElastiCache 複寫](#)
- [ElastiCache 端點](#)

- [ElastiCache 參數群組](#)
- [ElastiCache 安全性](#)
- [ElastiCache 子網路群組](#)
- [ElastiCache 備份](#)
- [ElastiCache 事件](#)

ElastiCache 節點

節點是 ElastiCache 部署的最小建置區塊。節點可以與其他節點隔離而獨立存在，或與其他節點以某種關係存在。

節點是固定大小的安全網路連接 區塊RAM。每個節點都會執行當您建立叢集時所選擇的引擎和版本執行個體。您可視需要將叢集中的節點垂直擴展或縮減為不同的執行個體類型。如需詳細資訊，請參閱[擴展 ElastiCache](#)。

叢集中每個節點的執行個體類型都相同，並且執行相同的快取引擎。每個快取節點都有自己的網域名稱服務（DNS）名稱和連接埠。支援多種類型的快取節點，每個節點都有各種數量的相關記憶體。如需支援的節點執行個體類型清單，請參閱[支援的節點類型](#)。

您可以購買節點 pay-as-you-go，其中您只需為節點的使用付費。或以大幅降低的每小時費率購買預留節點。如果您的用量很高，購買預留節點可以為您節省成本。假設您的叢集幾乎總是在使用中，且您會偶爾新增節點來處理使用尖峰。在此情況下，您可以購買多個預留節點來在大多數時間執行。然後，您可以在偶爾需要新增節點的時間購買 pay-as-you-go 節點。如需預留節點的詳細資訊，請參閱[預留節點](#)。

如需節點的詳細資訊，請參閱[在中管理節點 ElastiCache](#)。

ElastiCache 碎片

Valkey 或 Redis OSS 碎片（在 API 和 中稱為節點群組 CLI）是一到六個相關節點的群組。啟用叢集模式的 Valkey 或 Redis OSS 叢集一律至少有一個碎片。

碎片化是一種資料庫分割方法，可將大型資料庫分割為更小、更快速且更易於管理的區塊，稱為資料碎片。這可以透過在多個不同的區段中分佈操作來提高資料庫效率。使用碎片可以提供許多好處，包括改善效能、可擴展性和成本效益。

啟用叢集模式的 Valkey 和 Redis OSS 叢集最多可以有 500 個碎片，您的資料會分割到碎片。如果 Valkey 或 Redis OSS 引擎版本為 5.0.6 或更新版本，節點或碎片限制可以增加到每個叢集最多 500

個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個（每個碎片一個主要版本和 5 個複本）到 500 個（單一主要版本並且沒有複本）之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路CIDR太小，範圍太小，或者子網路被其他叢集共用和大量使用。如需詳細資訊，請參閱[建立子網路群組](#)。對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱[AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type（每執行個體類型每叢集的節點數）。

多個節點碎片實作複寫的方式，是使用一個讀/寫主節點及 1 至 5 個複本節點。如需詳細資訊，請參閱[使用複寫群組的高可用性](#)。

如需碎片的詳細資訊，請參閱[在中使用碎片 ElastiCache](#)。

ElastiCache 叢集

叢集是一或多個[節點](#)的邏輯分組。資料會跨 Memcached 叢集中的節點分割，並跨啟用叢集模式的 Valkey 或 Redis OSS叢集中的碎片分割。

許多 ElastiCache 操作以叢集為目標：

- 建立叢集
- 修改叢集
- 擷取叢集的快照 (所有 Redis 版本)
- 刪除叢集
- 在叢集中檢視元素
- 從叢集中新增或移除成本配置標籤

如需詳細資訊，請參閱下列相關主題：

- [在中管理叢集 ElastiCache](#) 和 [在中管理節點 ElastiCache](#)

叢集、節點和相關操作的詳細資訊。

- [AWS 服務限制：Amazon ElastiCache](#)

有關 ElastiCache 限制的資訊，例如節點或叢集的數量上限。若要超過某些限制，您可以使用[Amazon ElastiCache 快取節點請求表單 提出請求](#)。

- [減少故障](#)

有關改善叢集和 Valkey 或 Redis OSS 複寫群組容錯能力的資訊。

一般叢集組態

以下是一般叢集組態。

Valkey 或 Redis OSS 叢集

停用叢集模式的 Valkey 或 Redis OSS 叢集一律只包含一個碎片（在 API 和中 CLI，一個節點群組）。Valkey 或 Redis OSS 碎片包含一到六個節點。如果一個碎片中有多個節點，該碎片便可支援複寫。在這種情況下，一個節點是讀/寫主節點，其他節點則是唯讀複本節點。

為了改善容錯能力，我們建議在 Valkey 或 Redis OSS 叢集中至少有兩個節點，並啟用多可用區域。如需詳細資訊，請參閱[減少故障](#)。

隨著 Valkey 或 Redis OSS 叢集的需求變更，您可以進行擴展或縮減。若要這麼做，請將叢集移至不同的節點執行個體類型。如果您的應用程式需要密集讀取，建議您將僅供讀取複本新增至叢集。如此一來，您就可以將讀取作業分散在更適當數量的節點間。

您也可以使用資料分層。將較常存取的資料存放在記憶體中，而較不常存取的資料存放在硬碟上。使用資料分層的優點在於可降低記憶體需求。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

ElastiCache 支援動態將 Valkey 或 Redis OSS 叢集的節點類型變更為較大的節點類型。如需擴展或縮減的資訊，請參閱[擴展 Valkey 或 Redis 的單節點叢集 OSS \(停用叢集模式\)](#) 或 [擴展 Valkey 或 Redis 的複本節點 OSS \(停用叢集模式\)](#)。

Memcached 的一般叢集組態

Memcached 支援每個 AWS 區域每個客戶最多 300 個節點，每個叢集都有 1–60 個節點。您可將您的資料分割到 Memcached 叢集的各節點中。

當您執行 Memcached 引擎時，叢集可以由 1–60 個節點組成。您可將您的資料庫分割至各節點中。您的應用程式會讀取和寫入每個節點的端點。如需詳細資訊，請參閱[自動探索](#)。

為了改善容錯能力，請在叢集的 AWS 區域內的各種可用區域（AZs）中尋找 Memcached 節點。如此一來，單一 AZ 中的故障，對整個叢集和應用程式的影響會降到最低。如需詳細資訊，請參閱[減少故障](#)。

隨著 Memcached 叢集的需求變更，您可以透過新增或移除節點來擴展或縮減節點，在節點的新數量之間重新分割資料。當您分割資料時，建議您使用一致性雜湊。如需一致性雜湊的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以進行有效的負載平衡 \(Memcached\)](#)。

ElastiCache 複寫

對於 Valkey 和 Redis OSS，複寫是透過碎片中從兩個到六個節點（在 API 和 CLI 中，稱為節點群組）進行分組來實作。其中一個節點是讀取/寫入主要節點。所有其他節點則是唯讀複本節點。複寫僅適用於 ElastiCache 搭配 Valkey 和 Redis 的 OSS，不適用於 ElastiCache（Memcached）。

每個複本節點都會保有從主要節點取得的資料複本。複本節點使用非同步複寫機制，與主要節點保持同步。應用程式可以從叢集中的任何節點讀取，但只能寫入到主要節點。僅供讀取複本可藉由將讀取分佈到多個端點來增強擴展性。僅供讀取複本也可藉由保留多個資料複本來提升容錯能力。將僅供讀取複本放置在多個可用區域可進一步提升容錯能力。如需容錯能力的詳細資訊，請參閱[減少故障](#)。

Valkey 或 Redis OSS 叢集支援一個碎片（在 API 和 CLI 中，稱為節點群組）。

從 API 和 CLI 角度複寫使用不同的術語來維持與先前版本的相容性，但結果相同。下表顯示實作複寫的 API 和 CLI 術語。

比較複寫：Valkey 或 Redis OSS（停用叢集模式）和 Valkey 或 Redis OSS（啟用叢集模式）--> 啟用叢集模式的 Valkey 或 Redis OSS 叢集與停用叢集模式的 Valkey 或 Redis OSS 叢集

在下表中，您可以找到 Valkey 或 Redis OSS（停用叢集模式）和 Valkey 或 Redis OSS（啟用叢集模式）複寫群組的功能比較。

	停用OSS叢集模式的 Valkey 或 Redis 叢集	啟用OSS叢集模式的 Valkey 或 Redis 叢集
碎片 (節點群組)	1	1-500
每個碎片的複本 (節點群組)	0 - 5	0 - 5
資料分割	否	是
新增/刪除複本	是	是
新增/刪除節點群組	否	是
支援擴展	是	是
支援引擎升級	是	是
提升複本為主要	是	自動

	停用OSS叢集模式的 Valkey 或 Redis 叢集	啟用OSS叢集模式的 Valkey 或 Redis 叢集
Multi-AZ	選用	必要
備份/還原	是	是

備註：

如果主節點沒有複本而主節點失敗，您會遺失該主節點的所有資料。

您可以使用備份和還原來遷移至 Valkey 或 Redis OSS (啟用叢集模式) 。

您可以使用備份和還原來調整 Valkey 或 Redis OSS (啟用叢集模式) 叢集的大小。

所有碎片 (在 API 和 CLI 中，節點群組) 和節點都必須位於相同的 AWS 區域中。不過，您可以在該 AWS 區域內的多個可用區域中佈建個別節點。

僅供讀取複本可防止資料遺失，因為您的資料是透過兩個或多個節點 (主節點和一或多個僅供讀取複本) 進行複寫。為了取得更佳的可靠性和更快的復原速度，建議您在不同可用區域內建立一或多個僅供讀取複本。

您也可以利用全域資料儲存。透過使用 Global Datastore for Redis OSS 功能，您可以跨 AWS 區域使用完全受管、快速、可靠和安全的複寫。使用此功能，您可以為 建立跨區域僅供讀取複本叢集 ElastiCache，以跨 AWS 區域啟用低延遲讀取和災難復原。如需詳細資訊，請參閱 [使用全域資料存放區跨 AWS 區域複寫](#)。

複寫：限額和排除

- 節點類型 T1 不支援多個可用區。

ElastiCache 端點

端點是應用程式用來連線至 ElastiCache 節點或叢集的唯一地址。

停用叢集模式的 Valkey 或 Redis OSS 的單一節點端點

單一節點 Valkey 或 Redis OSS 叢集的端點用於連線至叢集進行讀取和寫入。

停用OSS叢集模式的 Valkey 或 Redis 的多節點端點

停用叢集模式的多節點 Valkey 或 Redis OSS叢集有兩種端點類型。主要端點一律會連線到叢集中的主要節點，即使主要角色中的特定節點變更也一樣。使用主要端點對叢集進行所有寫入。

使用「讀取器端點」來將傳入連線平均分散在所有讀取複本間的端點中。使用個別節點端點進行讀取操作（在 API/CLI 中，這些端點稱為讀取端點）。

Valkey 或 Redis OSS（啟用叢集模式）端點

啟用OSS叢集模式的 Valkey 或 Redis 叢集具有單一組態端點。只要連線到組態端點，您的應用程式就可以探索叢集中每個碎片的主要和讀取端點。

如需詳細資訊，請參閱[在中尋找連線端點 ElastiCache](#)。

ElastiCache（Memcached）端點

Memcached 叢集中的每個節點都有自己的端點。叢集也具有稱為「組態端點」的端點。如果您啟用自動探索並連線至組態端點，您的應用程式會自動「知道」每個節點端點，即使是在叢集新增或移除節點後。如需詳細資訊，請參閱[自動探索](#)。

如需詳細資訊，請參閱 [端點](#)。

ElastiCache 參數群組

快取參數群組是用於管理受支援引擎軟體之執行時間設定的便利方式。參數用於控制記憶體用量、移出政策、項目大小等等。ElastiCache 參數群組是引擎特定參數的命名集合，您可以套用至叢集。藉由執行此操作，您可以確保該叢集中的所有節點都以完全相同的方式設定。

如需支援的參數清單、其預設值以及可修改的參數，請參閱 [DescribeEngineDefaultParameters \(CLI: describe-engine-default-parameters\)](#)。

如需 ElastiCache 參數群組的詳細資訊，請參閱 [使用參數群組設定引擎 ElastiCache 參數](#)。

ElastiCache 安全性

為了增強安全性，ElastiCache 節點存取僅限於您允許之 Amazon EC2執行個體上執行的應用程式。您可以使用安全群組控制可存取叢集的 Amazon EC2執行個體。

根據預設，所有新 ElastiCache 叢集都會在 Amazon Virtual Private Cloud（AmazonVPC）環境中啟動。您可以使用子網路群組，授予在特定子網路上執行之 Amazon EC2執行個體的叢集存取權。

除了限制節點存取之外，還支援執行指定版本的節點 ElastiCache TLS和就地加密 ElastiCache。如需詳細資訊，請參閱下列內容：

- [Amazon 中的資料安全 ElastiCache](#)
- [使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)

ElastiCache 子網路群組

子網路群組是一組子網路（通常是私有的），您可以為在 Amazon VPC環境中執行的叢集指定。

如果您在 Amazon 中建立叢集VPC，則必須指定快取子網路群組。ElastiCache 使用該快取子網路群組來選擇該子網路中的子網路和 IP 地址，以與您的快取節點建立關聯。

如需 Amazon VPC環境中快取子網路群組用量的詳細資訊，請參閱下列內容：

- [Amazon VPCs和安全性 ElastiCache](#)
- [步驟 3。授權對叢集的存取](#)
- [子網路和子網路群組](#)

ElastiCache 備份

備份是 point-in-time Valkey 或 Redis OSS叢集或無伺服器快取，或 Memcached 無伺服器快取的副本。備份可用來還原現有的叢集或植入新叢集。備份包含叢集中的所有資料加上一些中繼資料。

根據叢集OSS上執行的 Valkey 或 Redis 版本，備份程序需要不同數量的預留記憶體才能成功。如需詳細資訊，請參閱下列內容：

- [快照和還原](#)
- [同步與備份的實作方式](#)
- [自行設計叢集之備份的效能影響](#)
- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)

ElastiCache 事件

當快取叢集上發生重要事件時，ElastiCache 會將通知傳送至特定 Amazon SNS主題。這些事件可能包含新增節點失敗或故障、安全群組修改和其他事件等。藉由監控重要的事件，您可以了解叢集目前的狀態，許多情況下還能採取修正動作。

如需 ElastiCache 事件的詳細資訊，請參閱 [Amazon ElastiCache 事件SNS監控](#)。

ElastiCache 術語

2016 年 10 月，Amazon ElastiCache 推出對 Redis 3.2 OSS 的支援。此時，我們新增了將資料分割到最多 500 個碎片的支援（在 ElastiCache API 和 中稱為節點群組 AWS CLI）。為了保持與先前版本的相容性，我們擴展了 2015-02-02 API 版操作，以包含新的 Redis OSS 功能。

與此同時，我們開始使用 ElastiCache 主控台內的術語，該術語用於此新功能，並在整個產業中通用。這些變更表示在某些時間點，API 和 中使用的術語 CLI 可能與主控台中使用的術語不同。下列清單識別 API 和 CLI 和 主控台之間可能不同的術語。

快取叢集 (節點) 與節點

當 one-to-one 沒有複本節點時，節點與快取叢集之間存在關係。因此，ElastiCache 主控台經常可互換地使用術語。主控台現在統一使用節點一詞。唯一的例外是 Create Cluster (建立叢集) 按鈕，這會啟動程序，以建立具有複本節點或完全沒有複本節點的叢集。

和 ElastiCache API AWS CLI 會繼續使用過去的術語。

叢集與 Valkey 或 Redis OSS 複寫群組

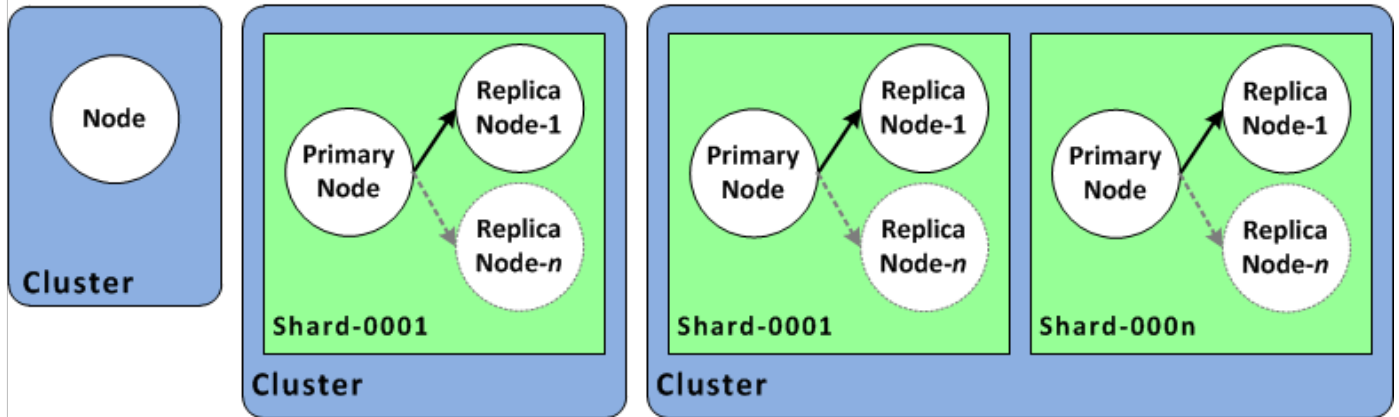
主控台現在對所有 ElastiCache (Redis OSS) 叢集使用術語叢集。主控台會在下列所有情況中使用叢集一詞：

- 當叢集是單一節點 Valkey 或 Redis OSS 叢集時。
- 當叢集是支援單一碎片內複寫的 Valkey 或 Redis OSS (停用叢集模式) 叢集時 (在 API 和 中 CLI，稱為節點群組)。
- 當叢集是支援在 1-90 個碎片內複寫的 Valkey 或 Redis OSS (啟用叢集模式) 叢集時，或最多 500 個具有限制增加請求的叢集。若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

如需 Valkey 或 Redis OSS 複寫群組的詳細資訊，請參閱 [使用複寫群組的高可用性](#)。

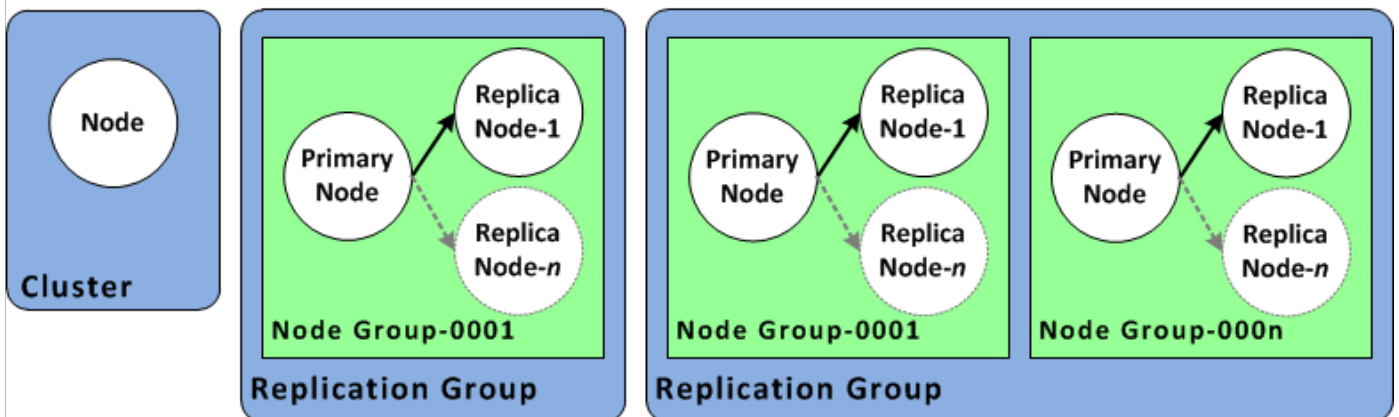
下圖從主控台的角度說明 ElastiCache (Redis OSS) 叢集各種拓撲。

ElastiCache (Redis OSS): Console View



和 ElastiCache API AWS CLI 操作仍然區分單一節點 ElastiCache (RedisOSS) 叢集與多節點 Valkey 或 Redis OSS 複寫群組。下圖從 ElastiCache API 和 AWS CLI 角度說明各種 ElastiCache (Redis OSS) 拓撲。

ElastiCache (Redis OSS): API/CLI View



Valkey 或 Redis OSS 複寫群組與全域資料存放區

全域資料存放區是一或多個叢集的集合，這些叢集會在區域之間彼此複寫，而 Valkey 或 Redis OSS 複寫群組會使用多個碎片，在已啟用叢集模式的叢集中複寫資料。全域資料存放區由下列項目組成：

- 主要 (主動) 叢集 - 主要叢集接受複寫到全域資料存放區內所有叢集的寫入。主要叢集也接受讀取要求。
- 次要 (被動) 叢集 - 次要叢集只接受讀取要求，並從主要叢集複寫資料更新。次要叢集需要位於與主要叢集不同的 AWS 區域。

如需全域資料存放區的相關資訊，請參閱「[使用全域資料存放區跨 AWS 區域複寫](#)」。

教學課程：如何設計自己的叢集

以下是如何為 Valkey 和 Redis 設計自己的叢集OSS。

主題

- [設計您自己的 ElastiCache \(Valkey \) 叢集](#)
- [設計您自己的 ElastiCache \(RedisOSS \) 叢集](#)

設計您自己的 ElastiCache (Valkey) 叢集

以下是您開始設計 ElastiCache (Valkey) 叢集時必須採取的一次性動作。

步驟 1：建立子網路群組

建立 ElastiCache (Valkey) 叢集之前，請先建立子網路群組。快取子網路群組是您可能想要為 中的快取叢集指定的子網路集合VPC。在 中啟動快取叢集時VPC，您需要選取快取子網路群組。然後使用 ElastiCache 該快取子網路群組，將該子網路內的 IP 地址指派給叢集中的每個快取節點。

建立新子網路群組時，請記下可用 IP 地址的數量。如果子網路有很少可用的 IP 地址，對於您還可以新增至叢集的節點數量，您可能受到限制。若要解決此問題，您可以對子網路群組指定一或多個子網路，使得您在叢集的可用區域中有足夠數量的 IP 地址。在那之後，您便可以將更多節點新增至您的叢集。

如需設定的詳細資訊，ElastiCache 請參閱 [設定 ElastiCache](#)。

下列程序顯示如何建立名為 mysubnetgroup (控制台) 的子網路群組和 AWS CLI。

建立子網路群組 (主控台)

下列程序顯示如何建立子網路群組 (主控台)。

建立子網路群組 (主控台)

1. 登入 AWS 管理主控台，然後在 開啟 ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽清單中，選擇 Subnet Groups (子網路群組)。
3. 選擇 Create Subnet Group (建立子網路群組)。
4. 在 Create Subnet Group (建立子網路群組) 精靈中，執行下列動作。當您滿意所有設定後，請選擇 Yes, Create (是，建立)。
 - a. 在 Name (名稱) 方塊中，輸入子網路群組的名稱。

- b. 在 Description (描述) 方塊中，輸入子網路群組的描述。
- c. 在 VPC ID 方塊中，選擇您建立VPC的 Amazon。
- d. 在可用區域和子網路 ID 清單中，選擇私有子網路的可用區域 或 [搭配 使用本機區域 ElastiCache](#)和 ID，然後選擇新增。

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID

The identifier for the VPC environment where your cluster is to run.

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

5. 在出現的確認訊息中，選擇 Close (關閉)。

您的新子網路群組會出現在 ElastiCache 主控台的子網路群組清單中。您可以在視窗底部選擇要查看詳細資訊的子網路群組，例如與此群組相關聯的所有子網路。

建立子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `create-cache-subnet-group` 來建立子網路群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup \  
  --cache-subnet-group-description "Testing" \  
  --subnet-ids subnet-53df9c3a
```

針對 Windows :

```
aws elasticache create-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

此命令應該產生類似下列的輸出 :

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ],  
    "CacheSubnetGroupName": "mysubnetgroup"  
  }  
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [create-cache-subnet-group](#).

步驟 2：建立叢集

在建立供生產使用的叢集之前，您明顯需要考慮如何設定叢集以符合您的業務需求。這些問題在「[在中準備叢集 ElastiCache](#)」一節中說明。此入門練習的用意是讓您建立停用叢集模式的叢集，且可以接受它們適用的預設組態值。

您建立的叢集將會實際上線，而非在沙盒中執行。在您刪除執行個體之前，會產生執行個體的標準 ElastiCache 使用費。如果您一口氣地完成這裡所述的練習，並在完成時刪除您的叢集，則總計費用會很少 (通常不到 1 美元)。如需 ElastiCache 使用率的詳細資訊，請參閱 [Amazon ElastiCache](#)。

您的叢集會根據 Amazon VPC 服務在虛擬私有雲端 (VPC) 中啟動。

建立 Valkey (停用叢集模式) 叢集 (主控台)

使用 ElastiCache 主控台建立 Valkey (停用叢集模式) 叢集

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇要啟動此叢集 AWS 的區域。
3. 從導覽窗格選擇 Get started (開始使用)。
4. 選擇建立VPC並遵循[建立虛擬私有雲端 \(VPC \)](#) 中概述的步驟。
5. 在 ElastiCache 儀表板頁面上，選擇 Valkey 快取或 Redis OSS快取，然後選擇建立 Valkey 快取或 Redis OSS快取。
6. 在 Cluster settings (叢集設定) 中，執行下列動作：
 - a. 選擇 Configure and create a new cluster (設定和建立新叢集)。
 - b. 針對 Cluster mode (叢集模式)，選擇 Disabled (已停用)。
 - c. 針對 Cluster info (叢集資訊)，輸入 Name (名稱) 值。
 - d. (選用) 輸入 Description (描述) 值。
7. 在 Location (位置) 中：

AWS Cloud

1. 針對 AWS 雲端，建議您接受多可用區域和自動容錯移轉的預設設定。如需詳細資訊，請參閱[使用多可用區 將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)。
2. 在 Cluster settings (叢集設定) 中
 - a. 針對 Engine version (引擎版本)，選擇可用的版本。
 - b. 針對 Port (連接埠)，使用預設連接埠 6379。如果您基於任何原因需要使用不同的連接埠，請輸入連接埠號碼。
 - c. 對於 Parameter group (參數群組)，請選擇參數群組或新建參數群組。參數群組可控制叢集的執行時間參數。如需參數群組的詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#)和[建立 ElastiCache 參數群組](#)。

Note

當您選取參數群組來設定引擎組態值時，該參數群組會套用至全域資料存放區中的所有叢集。在 Parameter Groups (參數群組) 頁面上，yes/no (是/否) Global (全域) 屬性指出參數群組是否為全域資料存放區的一部分。

- d. 在 Node type (節點類型) 中，選擇向下箭頭 (▼)。
在 Change node type (變更節點類型) 對話方塊中，為您想要的節點類型選擇 Instance family (執行個體系列) 的值。然後選擇您要用於此叢集的節點類型，然後選擇 Save (儲存)。

如需詳細資訊，請參閱[選擇您的節點大小](#)。

如果選擇 r6gd 節點類型，將自動啟用資料分層。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

- e. 針對 Number of replicas (複本數)，選擇您想要的僅供讀取複本數目。如果您已啟用 Multi-AZ (多可用區域)，數字必須介於 1-5。

3. 在 Connectivity (連線) 下

- a. 針對 Network type (網路類型)，請選擇此叢集支援的 IP 版本。
- b. 對於子網路群組，請選擇您要套用至此叢集的子網路。ElastiCache 使用該子網路群組來選擇該子網路內的子網路和 IP 地址，以與節點建立關聯。ElastiCache 叢集需要同時指派給它們的雙堆疊子網路 IPv4 和 IPv6 地址，才能在雙堆疊模式下操作，而 IPv6 僅子網路只能作為 IPv6 操作。

建立新的子網路群組時，請輸入其所屬的 VPC ID。

如需詳細資訊，請參閱：

- [在中選擇網路類型 ElastiCache](#)。
- [在中建立子網路 VPC](#)。

如果您是 [搭配使用本機區域 ElastiCache](#)，必須建立或選擇位於本地區域中的子網路。

如需詳細資訊，請參閱[子網路和子網路群組](#)。

4. 針對 Availability zone placements (可用區域置放)，您有兩種選項：

- 沒有偏好設定 – ElastiCache 選擇可用區域。
- Specify availability zones (指定可用區域) - 您可為每個叢集指定可用區域。


如果您選擇指定可用區域，請針對每個碎片中的每個叢集，從清單中選擇可用區域。

如需詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。

5. 選擇 Next (下一步)

6. 在進階 Valkey 或 Redis OSS設定下

- 針對 Security (安全)：
 - i. 若要加密資料，您有下列選項：
 - Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

 Note

您可以選擇 Customer Managed AWS KMS 金鑰並選擇 金鑰，以提供不同的加密金鑰。如需詳細資訊，請參閱[從 AWS 使用客戶受管金鑰 KMS](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。針對 Valkey 和 Redis OSS引擎 6.0 版及更新版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：
 - No Access Control (無存取控制) - 這是預設的設定。這表示使用者存取叢集時不受任何限制。
 - User Group Access Control List (使用者群組存取控制清單) - 選取的使用者群組需具有可存取叢集的已定義使用者集。如需詳細資訊，請參閱[使用主控台和管理使用者群組 CLI](#)。
 - AUTH 預設使用者 – Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱 [AUTH](#)。
 - AUTH – Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱 [AUTH](#)。

Note

對於 3.2.6 版及更新OSS版本之間的 Valkey 和 Redis，除 3.2.10 版外，Redis OSSAUTH是唯一的選項。

- ii. 在 Security groups (安全群組) 中，選擇要用於此叢集的安全群組。安全群組可做為防火牆來控制叢集的網路存取。您可以使用的預設安全群組，VPC或建立新的安全群組。

如需安全群組的詳細資訊，請參閱 Amazon 使用者指南中的 [的安全群組VPC](#)。
VPC

7. 針對定期排程自動備份，請選取 Enable automatic backups (啟用自動備份)，然後輸入每份自動備份要保存的天數，過了此天數後即會自動將其刪除。如果您不想定期排程自動備份，請清除 Enable automatic backups (啟用自動備份) 核取方塊。在任何一種情況下，您都可以選擇建立手動備份。

如需 Redis OSS 備份和還原的詳細資訊，請參閱 [快照和還原](#)。

8. (選用) 指定維護時段。維護時段是每週 ElastiCache排程叢集系統維護的時間，通常為一小時。您可以允許 ElastiCache 為您的維護時段選擇日期和時間（無偏好設定），也可以自行選擇天數、時間和持續時間（指定維護時段）。如果您從清單中選擇 Specify maintenance window (指定維護時段)，請為您的維護時段選擇 Start day (開始日)、Start time (開始時間) 和 Duration (持續時間)。所有時間都是 UCT 次。

如需詳細資訊，請參閱[管理 ElastiCache 叢集維護](#)。

9. (選用) 針對 Logs (日誌)：
 - 在日誌格式下，選擇文字或 JSON。
 - 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
 - 在日誌目的地下，選擇建立新 CloudWatch 日誌群組名稱或 Firehose 串流名稱，或選擇選取現有日誌群組名稱或 Firehose 串流名稱，然後選擇您的 CloudWatch 日誌群組名稱或您的 Firehose 串流名稱。
10. 對於標籤，為協助您管理叢集和其他 ElastiCache 資源，您可以標籤形式將自己的中繼資料指派給每個資源。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。
11. 選擇 Next (下一步)。
12. 檢閱所有項目和選項，然後進行任何所需的更正。當您準備好時，請選擇 Create (建立)。

On premises

1. 針對 On premises (內部部署)，建議您將 Auto-failover (自動容錯移轉) 保持啟用。如需詳細資訊，請參閱[使用多可用區將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)
2. 若要完成建立叢集的程序，請按照[使用 Outposts](#) 的步驟執行。

一旦叢集的狀態可用，您就可以授予 Amazon EC2存取權、連線到叢集，並開始使用叢集。如需詳細資訊，請參閱 [步驟 3. 授權對叢集的存取](#) 和 [步驟 4. 連線至叢集的節點](#)。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立 Valkey (停用叢集模式) 叢集 (AWS CLI)

Example

下列CLI程式碼會建立沒有複本的 Valkey (停用叢集模式) 快取叢集。

若為 Linux、macOS 或 Unix :

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine valkey \  
--num-cache-nodes 1 \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

針對 Windows :

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine valkey ^  
--num-cache-nodes 1 ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

若要在叢集已啟用的情況下作業，請參閱下列主題：

- 若要使用主控台，請參閱[建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(主控台 \)](#)。
- 若要使用 AWS CLI，請參閱[建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(AWS CLI \)](#)。

步驟 3。授權對叢集的存取

本節假設您熟悉啟動和連線至 Amazon EC2執行個體。如需詳細資訊，請參閱 [Amazon EC2 入門指南](#)。

所有 ElastiCache 叢集的設計都是從 Amazon EC2執行個體存取。最常見的案例是從相同 Amazon Virtual Private Cloud (AmazonVPC) 中的 Amazon EC2執行個體存取 ElastiCache 叢集，這將是本練習的情況。

根據預設，僅有用來建立叢集的帳戶可以透過網路存取您的叢集。您必須先授權EC2執行個體存取叢集，才能從EC2執行個體連線至叢集。

最常見的使用案例是部署在EC2執行個體上的應用程式需要連線到相同 中的叢集VPC。管理相同EC2執行個體和叢集之間存取的最簡單方法是VPC執行下列動作：

1. 為您的叢集建立VPC安全群組。此安全群組可用來限制叢集執行個體的存取權限。例如，您可以為此安全群組建立自訂規則，以允許TCP使用您在建立叢集時指派給叢集的連接埠和用於存取叢集的 IP 地址進行存取。

Valkey 或 Redis OSS叢集和複寫群組的預設連接埠為 6379。

Important

Amazon ElastiCache 安全群組僅適用於未在 Amazon Virtual Private Cloud 環境 () 中執行的叢集VPC。如果您在 Amazon Virtual Private Cloud 中執行，主控台的導覽窗格中的 Security Groups (安全群組) 就無法使用。

如果您在 Amazon 中執行 ElastiCache 節點VPC，您可以使用與 ElastiCache 安全群組不同的 Amazon VPC安全群組控制對叢集的存取。如需在 Amazon ElastiCache 中使用的詳細資訊VPC，請參閱 [Amazon VPCs和安全性 ElastiCache](#)

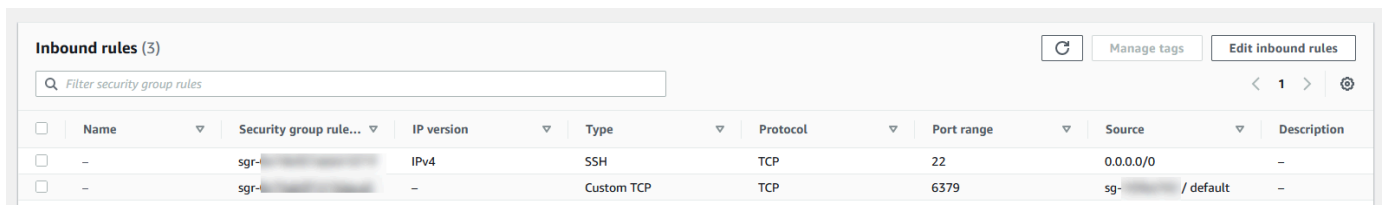
2. 為您的EC2執行個體 (網路和應用程式伺服器) 建立VPC安全群組。如有需要，此安全群組可以透過的VPC路由表允許從網際網路存取EC2執行個體。例如，您可以在此安全群組上設定規則，以允許透過連接埠 22 TCP存取EC2執行個體。
3. 在叢集的安全群組中建立自訂規則，以允許來自您為EC2執行個體建立的安全群組的連線。這樣做會允許安全群組的所有成員存取叢集。

Note

如果您打算使用**本地區域**，請確認本地區域已啟用。當您在該本機區域建立子網路群組時，您的 VPC 會延伸到該本機區域，而您的 VPC 會將子網路視為任何其他可用區域中的任何子網路。所有相關閘道和路由表都將自動調整。

在允許從其他 VPC 安全群組連線的安全群組中建立規則

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/vpc> 開啟 Amazon VPC 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選取或建立您將用於叢集執行個體的安全群組。在 Inbound Rules (傳入規則) 下方，選取 Edit Inbound Rules (編輯傳入規則)，然後選取 Add Rule (新增規則)。此安全群組將允許其他安全群組成員存取。
4. 從類型選擇自訂 TCP 規則。
 - a. 針對 Port Range (連接埠範圍)，指定您在建立叢集時所使用的連接埠。
Valkey 或 Redis OSS 叢集和複寫群組的預設連接埠為 6379。
 - b. 在 Source (來源) 方塊中輸入安全群組的 ID。從清單中選取您要用於 Amazon EC2 執行個體的安全群組。
5. 完成後，請選擇 Save (儲存)。



Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-

啟用存取權之後，您便可連線到節點，如下節所述。

如需從不同 Amazon VPC、不同 AWS 區域或甚至公司網路存取 ElastiCache 叢集的資訊，請參閱下列內容：

- [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)
- [從外部存取 ElastiCache 資源 AWS](#)

步驟 4. 連線至叢集的節點

在繼續之前，請先完成[步驟 3. 授權對叢集的存取](#)。

本節假設您已建立 Amazon EC2 執行個體，且可與其連線。如需如何執行此操作的指示，請參閱[Amazon EC2 入門指南](#)。

只有在您授權 Amazon EC2 執行個體執行此操作時，才能連線至叢集節點。

尋找您的節點端點

當您的叢集處於可用狀態且您已授權存取該叢集時，您可以登入 Amazon EC2 執行個體並連線至叢集。若要執行此作業，您必須先判斷端點。

尋找 Valkey（停用叢集模式）叢集的端點（主控台）

如果 Valkey（停用叢集模式）叢集只有一個節點，則節點的端點會同時用於讀取和寫入。如果叢集有多個節點，則端點類型有三種：主要端點、讀取者端點和節點端點。

主要端點是一律解析為叢集中主要節點 DNS 的名稱。主要端點不會受到您叢集變更的影響（例如將僅供讀取複本提升至主要角色）。針對寫入活動，我們建議您的應用程式連線到主要端點。

讀取器端點將在 ElastiCache 叢集中的所有僅供讀取複本之間平均分割端點的傳入連線。其他因素（例如應用程式建立連線或應用程式如何（重新）使用連線）將決定流量分佈。隨著複本的新增或移除，讀取器端點會跟著叢集的變更即時保持在最新狀態。您可以將 ElastiCache 叢集的多個僅供讀取複本放置在不同的 AWS 可用區域（AZ），以確保讀取器端點的高可用性。

Note

讀取者端點並非負載平衡器。它是以循環方式解析為其中一個複本節點的 IP 地址 DNS 的記錄。

針對讀取活動，應用程式也可連線到叢集中的任何節點。與主要端點不同，節點端點會解析至特定端點。若您在叢集中進行變更（例如新增或刪除複本），您必須更新您應用程式中的節點端點。

尋找 Valkey（停用叢集模式）叢集的端點

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

2. 從導覽窗格中，選擇 Valkey 快取或 Redis OSS快取。

叢集畫面將會出現清單，其中包含任何現有的 Valkey 或 Redis 無OSS伺服器快取、Valkey（停用叢集模式）和 Valkey（啟用叢集模式）叢集。選擇您在 [建立 Valkey（停用叢集模式）叢集（主控台）](#) 一節中建立的叢集。

3. 若要尋找叢集的主要和/或讀取器端點，請選擇叢集名稱（而非選項按鈕）。

▼ Cluster details			
Cluster name	Description	Node type	Status
		cache.r6g.large	Available
Engine	Engine version	Global datastore	Global datastore role
Redis OSS	6.0.5	-	-
Update status	Cluster mode	Shards	Number of nodes
Update available	Off	1	3
Data tiering	Multi-AZ	Auto-failover	Encryption in transit
Disabled	Enabled	Enabled	Disabled
Encryption at rest	Parameter group	Outpost ARN	Configuration endpoint
Disabled	default.redis6.x	-	-
Primary endpoint	Reader endpoint	ARN	
[redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	[redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	[redacted]	

Valkey（停用叢集模式）叢集的主要和讀取器端點

如果叢集中只有一個節點，就不會有主要端點，因此您必須繼續下一個步驟。

4. 如果 Valkey（停用叢集模式）叢集有複本節點，您可以選擇叢集的名稱，然後選擇節點索引標籤，以尋找叢集的複本節點端點。

隨即顯示節點畫面，其顯示叢集中的每個節點、主要複本和複本，並與其端點一起列出。

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-002	available	replica	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-003	available	replica	6379	[redacted].amazonaws.com

Valkey（停用叢集模式）叢集的節點端點

5. 將端點複製到剪貼簿：
 - a. 逐一尋找您要複製的端點。

b. 選擇端點前面的複製圖示。

現在，端點即已複製到您的剪貼簿。如需使用端點連線至節點的相關資訊，請參閱 [連線至節點](#)。

Valkey（停用叢集模式）主要端點看起來如下所示。其中的差異取決於是否啟用傳輸中加密。

未啟用傳輸中加密

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

已啟用傳輸中加密

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

若要進一步探索您的端點，請參閱您所執行引擎和叢集類型的相關主題。

- [在中尋找連線端點 ElastiCache](#)
- [尋找 Valkey 或 Redis OSS（啟用叢集模式）叢集（主控台）的端點](#) - 您需要叢集的組態端點。
- [尋找端點 \(AWS CLI\)](#)
- [尋找端點（ElastiCache API）](#)

連線至 Valkey 或 Redis OSS 叢集或複寫群組（Linux）

現在您已擁有所需的端點，您可以登入 EC2 執行個體並連線至叢集或複寫群組。在下列範例中，您可以使用 valkey-cli 公用程式連線到叢集。最新版本的 valkey-cli 也支援 SSL/TLS 連接啟用加密/身分驗證的叢集。

下列範例使用執行 Amazon Linux 和 Amazon Linux 2 的 Amazon EC2 執行個體。如需搭配其他 Linux 發行版本安裝和編譯 valkey-cli 的詳細資訊，請參閱特定作業系統的文件。

Note

此程序涵蓋使用 `valkey-cli` 公用程式測試連線，僅供計劃外使用。如需支援的 Valkey 和 Redis OSS 用戶端清單，請參閱 [Valkey 文件](#)。如需搭配 AWS SDKs 使用的範例 ElastiCache，請參閱 [教學課程：開始使用 Python 和 ElastiCache](#)。

連線至停用叢集模式的未加密叢集

1. 執行下列命令以連線至叢集並取代 `primary-endpoint` 以及 `port number` 叢集的端點和連接埠號碼。（Valkey 或 Redis 的預設連接埠 OSS 為 6379。）

```
src/valkey-cli -h primary-endpoint -p port number
```

產生 Valkey 或 Redis OSS 命令提示字元的結果如下所示：

```
primary-endpoint:port number
```

2. 您現在可以執行 Valkey 或 Redis OSS 命令。

```
set x Hello
OK

get x
"Hello"
```

連線至啟用叢集模式的未加密叢集

1. 執行下列命令以連線至叢集並取代 `configuration-endpoint` 以及 `port number` 叢集的端點和連接埠號碼。（Valkey 或 Redis 的預設連接埠 OSS 為 6379。）

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

Note

在上述命令中，選項 `-c` 會在 [重新導向](#) 和 [ASK MOVED](#) 之後啟用叢集模式。

產生 Valkey 或 Redis OSS 命令提示字元的結果如下所示：

```
configuration-endpoint:port number
```

2. 您現在可以執行 Valkey 或 Redis OSS 命令。請注意，發生重新引導是因為您使用 `-c` 選項啟用了此功能。如果未啟用重新導向，命令會傳回 MOVED 錯誤。如需 MOVED 錯誤的詳細資訊，請參閱 [Redis OSS 叢集規格](#)。

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

連線到啟用加密/身分驗證的叢集

依預設，`valkey-cli` 在連線至 Valkey 或 Redis 時使用未加密的 TCP 連線。選項會在編譯 `valkey-cli` 時 `BUILD_TLS=yes` 啟用 SSL/TLS，如 [下載並設定命令列存取](#) 一節所示。啟用 AUTH 是選用的。不過，您必須啟用傳輸中加密，才能啟用 AUTH。如需加密和身分驗證的詳細資訊 ElastiCache，請參閱 [ElastiCache 傳輸中加密 \(TLS \)](#)。

Note

您可以 `--tls` 搭配 `valkey-cli` 使用 `-c` 選項來連線至啟用和停用的叢集模式加密叢集。如果叢集具有 AUTH 權杖集，您可以使用 `-a` 選項提供 AUTH 密碼。

在下列範例中，請務必取代 *cluster-endpoint* 以及 *port number* 叢集的端點和連接埠號碼。
(Valkey 或 Redis 的預設連接埠OSS為 6379。)

連線到停用叢集模式的加密叢集

下列範例會連線到啟用加密和身分驗證的叢集：

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

下列範例會連線到只啟用加密的叢集：

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

連線到啟用叢集模式的加密叢集

下列範例會連線到啟用加密和身分驗證的叢集：

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

下列範例會連線到只啟用加密的叢集：

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

連線至叢集後，您可以執行 Valkey 或 Redis OSS命令，如上述未加密叢集的範例所示。

valkey-cli 替代方案

如果叢集未啟用叢集模式，而且您需要建立叢集的連線以進行簡短測試，但不經過 valkey-cli 編譯，您可以使用 telnet 或 openssl。在下列範例命令中，請務必取代 *cluster-endpoint* 以及 *port number* 叢集的端點和連接埠號碼。(Valkey 或 Redis 的預設連接埠OSS為 6379。)

下列範例會連線到啟用加密和/或身分驗證且停用叢集模式的叢集：

```
openssl s_client -connect cluster-endpoint:port number
```

如果叢集已設定密碼，請先連線至叢集。連線後，請使用下列命令驗證叢集，然後按 Enter 鍵。在下列範例中，取代 *your-password* 使用叢集的密碼。

```
Auth your-password
```

下列範例會連線到沒有啟用加密和/或身分驗證且停用叢集模式的叢集：

```
telnet cluster-endpoint port number
```

連線至 Valkey 或 Redis OSS 叢集或複寫群組 (Windows)

若要使用 Valkey 或 Redis 從 EC2 Windows 執行個體連線至 Valkey CLI 或 Redis OSS 叢集 CLI，您必須下載 valkey-cli 套件，並使用 valkey-cli.exe 從 EC2 Windows 執行個體連線至 Valkey 或 Redis OSS 叢集。

在下列範例中，您可以使用 valkey-cli 公用程式連線到未啟用加密且執行 Valkey 或 Redis 的叢集 OSS。如需 Valkey 或 Redis OSS 和可用命令的詳細資訊，請參閱 [Valkey 網站上的 Valkey 和 Redis OSS 命令](#)。

使用 valkey-cli 連線至未啟用加密的 Valkey 或 Redis OSS 叢集

1. 使用您選擇的連線公用程式連線至您的 Amazon EC2 執行個體。如需如何連線至 Amazon EC2 執行個體的指示，請參閱 [Amazon EC2 入門指南](#)。
2. 在網際網路瀏覽器 <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> 中複製並貼上連結，以從的可用版本下載 Valkey 用戶端的 zip 檔案 <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

將 zip 檔案解壓縮到所需的資料夾/路徑。

開啟命令提示字元並變更為 Valkey 目錄，然後執行命令 `c:\Valkey>valkey-cli -h Redis_Cluster_Endpoint -p 6379`。

例如：

```
c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. 執行 Valkey 或 Redis OSS 命令。

您現在已連線至叢集，可以執行 Valkey 或 Redis OSS 命令，如下所示。

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
```

```
get b // Get value for key "b"
"Good-bye"
// wait >= 5 seconds

get b
(nil) // key has expired, nothing returned
quit // Exit from valkey-cli
```

接下來做些什麼？

現在您已嘗試了入門練習，您可以探索下列章節，進一步了解 ElastiCache 和 可用的工具：

- [入門 AWS](#)
- [適用於 Amazon Web Services 的工具](#)
- [AWS 命令列介面](#)
- [Amazon ElastiCache API 參考](#)

完成入門練習後，您可以閱讀這些章節以進一步了解 ElastiCache 管理：

- [選擇您的節點大小](#)

建議您具備夠大的快取，以包含所有要快取的資料。同時，您不想要支付超出所需快取數量的費用。此主題可協助您選擇最佳節點大小。

- [ElastiCache 最佳實務和快取策略](#)

識別並解決可能影響叢集效率的問題。

設計您自己的 ElastiCache (RedisOSS) 叢集

以下是為了設計自己的 ElastiCache (Redis OSS) 叢集而必須採取的一次性動作。

如需設定的詳細資訊，ElastiCache 請參閱 [設定 ElastiCache](#)。

主題

- [步驟 1：建立子網路群組](#)
- [步驟 2：建立叢集](#)
- [步驟 3：授予叢集的存取權](#)
- [步驟 4：連線至叢集的節點](#)

步驟 1：建立子網路群組

建立叢集前，您需先建立子網路群組。快取子網路群組是您可能想要為叢集中的快取叢集指定的子網路集合VPC。在叢集中啟動快取叢集時VPC，您需要選取快取子網路群組。然後使用 ElastiCache 該快取子網路群組，將該子網路內的 IP 地址指派給叢集中的每個快取節點。

建立新子網路群組時，請記下可用 IP 地址的數量。如果子網路有很少可用的 IP 地址，對於您還可以新增至叢集的節點數量，您可能受到限制。若要解決此問題，您可以對子網路群組指定一或多個子網路，使得您在叢集的可用區域中有足夠數量的 IP 地址。在那之後，您便可以將更多節點新增至您的叢集。

下列程序顯示如何建立名為 `mysubnetgroup` (控制台) 的子網路群組和 AWS CLI。

建立子網路群組 (主控台)

下列程序顯示如何建立子網路群組 (主控台)。

建立子網路群組 (主控台)

1. 登入 AWS 管理主控台，然後在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽清單中，選擇 Subnet Groups (子網路群組)。
3. 選擇 Create Subnet Group (建立子網路群組)。
4. 在 Create Subnet Group (建立子網路群組) 精靈中，執行下列動作。當您滿意所有設定後，請選擇 Yes, Create (是，建立)。
 - a. 在 Name (名稱) 方塊中，輸入子網路群組的名稱。
 - b. 在 Description (描述) 方塊中，輸入子網路群組的描述。
 - c. 在 VPC ID 方塊中，選擇您建立VPC的 Amazon。
 - d. 在 Availability Zone (可用區域) 和 Subnet ID (子網路 ID) 清單中，選擇可用區域或 [本地區域](#)，以及私有子網路的 ID，然後選擇 Add (新增)。

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

 The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID
 The identifier for the VPC environment where your cluster is to run.
 [Create VPC](#)

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet-██████████		172.31.16.0/20
us-east-1b	subnet-██████████		172.31.32.0/20
us-east-1c	subnet-██████████		172.31.0.0/20
us-east-1d	subnet-██████████		172.31.80.0/20

5. 在出現的確認訊息中，選擇 Close (關閉)。

您的新子網路群組會出現在 ElastiCache 主控台的子網路群組清單中。您可以在視窗底部選擇要查看詳細資訊的子網路群組，例如與此群組相關聯的所有子網路。

建立子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `create-cache-subnet-group` 來建立子網路群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "Testing" \
  --subnet-ids subnet-53df9c3a
```

針對 Windows :

```
aws elasticache create-cache-subnet-group ^
  --cache-subnet-group-name mysubnetgroup ^
  --cache-subnet-group-description "Testing" ^
  --subnet-ids subnet-53df9c3a
```

此命令應該產生類似下列的輸出 :

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-37c3cd17",
    "CacheSubnetGroupDescription": "Testing",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-53df9c3a",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [create-cache-subnet-group](#)。

步驟 2：建立叢集

在建立供生產使用的叢集之前，您明顯需要考慮如何設定叢集以符合您的業務需求。這些問題在「[在中準備叢集 ElastiCache](#)」一節中說明。此入門練習的用意是讓您建立停用叢集模式的叢集，且可以接受它們適用的預設組態值。

您建立的叢集將會實際上線，而非在沙盒中執行。在您刪除執行個體之前，會產生執行個體的標準 ElastiCache 使用費。如果您一口氣地完成這裡所述的練習，並在完成時刪除您的叢集，則總計費用會很少 (通常不到 1 美元)。如需 ElastiCache 使用率的詳細資訊，請參閱 [Amazon ElastiCache](#)。

您的叢集會根據 Amazon VPC 服務在虛擬私有雲端 (VPC) 中啟動。

建立 Redis OSS (停用叢集模式) 叢集 (主控台)

使用 ElastiCache 主控台建立 Redis OSS (停用叢集模式) 叢集

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇要啟動此叢集 AWS 的區域。
3. 從導覽窗格選擇 Get started (開始使用)。
4. 選擇建立VPC並遵循[建立虛擬私有雲端 \(VPC \)](#) 中概述的步驟。
5. 在 ElastiCache 儀表板頁面上，選擇 Valkey 快取或 Redis OSS快取 。在此練習中，我們將選擇 Redis OSS快取 ，然後選擇建立 Redis OSS快取 。
6. 在 Cluster settings (叢集設定) 中，執行下列動作：
 - a. 選擇 Configure and create a new cluster (設定和建立新叢集)。
 - b. 針對 Cluster mode (叢集模式)，選擇 Disabled (已停用)。
 - c. 針對 Cluster info (叢集資訊)，輸入 Name (名稱) 值。
 - d. (選用) 輸入 Description (描述) 值。
7. 在 Location (位置) 中：

AWS Cloud

1. 針對 AWS 雲端，建議您接受多可用區域和自動容錯移轉的預設設定。如需詳細資訊，請參閱[使用多可用區 將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)。
2. 在 Cluster settings (叢集設定) 中
 - a. 針對 Engine version (引擎版本)，選擇可用的版本。
 - b. 針對 Port (連接埠)，使用預設連接埠 6379。如果您基於任何原因需要使用不同的連接埠，請輸入連接埠號碼。
 - c. 對於 Parameter group (參數群組)，請選擇參數群組或新建參數群組。參數群組可控制叢集的執行時間參數。如需參數群組的詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#)和[建立 ElastiCache 參數群組](#)。

Note

當您選取參數群組來設定引擎組態值時，該參數群組會套用至全域資料存放區中的所有叢集。在 Parameter Groups (參數群組) 頁面上，yes/no (是/否) Global (全域) 屬性指出參數群組是否為全域資料存放區的一部分。

- d. 在 Node type (節點類型) 中，選擇向下箭頭 (▼)。
在 Change node type (變更節點類型) 對話方塊中，為您想要的節點類型選擇 Instance family (執行個體系列) 的值。然後選擇您要用於此叢集的節點類型，然後選擇 Save (儲存)。

如需詳細資訊，請參閱[選擇您的節點大小](#)。

如果選擇 r6gd 節點類型，將自動啟用資料分層。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

- e. 針對 Number of replicas (複本數)，選擇您想要的僅供讀取複本數目。如果您已啟用 Multi-AZ (多可用區域)，數字必須介於 1-5。

3. 在 Connectivity (連線) 下

- a. 針對 Network type (網路類型)，請選擇此叢集支援的 IP 版本。
- b. 對於子網路群組，請選擇您要套用至此叢集的子網路。ElastiCache 使用該子網路群組來選擇該子網路內的子網路和 IP 地址，以與節點建立關聯。ElastiCache 叢集需要同時具有的雙堆疊子網路 IPv4 和指派給它們的 IPv6 位址，才能在雙堆疊模式下操作，而 IPv6 只有 的子網路只能作為 IPv6 操作。

建立新的子網路群組時，請輸入其所屬的 VPC ID。

如需詳細資訊，請參閱：

- [在中選擇網路類型 ElastiCache](#)。
- [在中建立子網路 VPC](#)。

如果您是 [搭配使用本機區域 ElastiCache](#)，必須建立或選擇位於本地區域中的子網路。


如需詳細資訊，請參閱[子網路和子網路群組](#)。

4. 針對 Availability zone placements (可用區域置放)，您有兩種選項：
 - 無偏好設定 – ElastiCache 選擇可用區域。
 - Specify availability zones (指定可用區域) - 您可為每個叢集指定可用區域。

如果您選擇指定可用區域，請針對每個碎片中的每個叢集，從清單中選擇可用區域。

如需詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。

5. 選擇 Next (下一步)
6. 在進階 Redis OSS設定下
 - 針對 Security (安全)：
 - i. 若要加密資料，您有下列選項：
 - Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

 Note

您可以選擇 Customer Managed AWS KMS 金鑰並選擇金鑰，以提供不同的加密金鑰。如需詳細資訊，請參閱[從 AWS 使用客戶受管金鑰 KMS](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。對於 Redis OSS引擎 6.0 版及更新版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：
 - No Access Control (無存取控制) - 這是預設的設定。這表示使用者存取叢集時不受任何限制。
 - User Group Access Control List (使用者群組存取控制清單) - 選取的使用者群組需具有可存取叢集的已定義使用者集。如需詳細資訊，請參閱[使用主控台和管理使用者群組 CLI](#)。
 - AUTH 預設使用者 – Valkey 和 Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱 [AUTH](#)。
 - AUTH – Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱 [AUTH](#)。

Note

對於 3.2.6 版以後的 Redis OSS 版本，除 3.2.10 版外，Redis OSSAUTH 是唯一的選項。

- ii. 在 Security groups (安全群組) 中，選擇要用於此叢集的安全群組。安全群組可做為防火牆來控制叢集的網路存取。您可以使用的預設安全群組，VPC 或建立新的安全群組。

如需安全群組的詳細資訊，請參閱 Amazon 使用者指南中的 [的安全群組VPC](#)。
VPC

7. 針對定期排程自動備份，請選取 Enable automatic backups (啟用自動備份)，然後輸入每份自動備份要保存的天數，過了此天數後即會自動將其刪除。如果您不想定期排程自動備份，請清除 Enable automatic backups (啟用自動備份) 核取方塊。在任何一種情況下，您都可以選擇建立手動備份。

如需備份和還原的詳細資訊，請參閱 [快照和還原](#)。

8. (選用) 指定維護時段。維護時段是每週 ElastiCache 排程叢集系統維護的時間，通常為一小時。您可以允許 ElastiCache 為您的維護時段選擇日期和時間（無偏好設定），也可以自行選擇天數、時間和持續時間（指定維護時段）。如果您從清單中選擇 Specify maintenance window (指定維護時段)，請為您的維護時段選擇 Start day (開始日)、Start time (開始時間) 和 Duration (持續時間)。所有時間都是 UCT 次。

如需詳細資訊，請參閱 [管理 ElastiCache 叢集維護](#)。

9. (選用) 針對 Logs (日誌)：
 - 在日誌格式下，選擇文字或 JSON。
 - 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
 - 在日誌目的地下，選擇建立新 CloudWatch 日誌群組名稱或 Firehose 串流名稱，或選擇選取現有日誌群組名稱或 Firehose 串流名稱，然後選擇您的 CloudWatch 日誌群組名稱或您的 Firehose 串流名稱。
10. 對於標籤，為協助您管理叢集和其他 ElastiCache 資源，您可以標籤形式將自己的中繼資料指派給每個資源。如需詳細資訊，請參閱 [標記您的 ElastiCache 資源](#)。
11. 選擇 Next (下一步)。
12. 檢閱所有項目和選項，然後進行任何所需的更正。當您準備好時，請選擇 Create (建立)。

On premises

1. 針對 On premises (內部部署)，建議您將 Auto-failover (自動容錯移轉) 保持啟用。如需詳細資訊，請參閱[使用多可用區將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)
2. 若要完成建立叢集的程序，請按照[使用 Outposts](#) 的步驟執行。

一旦叢集的狀態可用，您就可以授予 Amazon EC2存取權、連線到叢集，並開始使用叢集。如需詳細資訊，請參閱 [步驟 3. 授權對叢集的存取](#) 和 [步驟 4. 連線至叢集的節點](#)。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立 Redis OSS (停用叢集模式) 叢集 (AWS CLI)

Example

下列CLI程式碼會建立沒有複本的 Redis OSS (停用叢集模式) 快取叢集。

若為 Linux、macOS 或 Unix :

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

針對 Windows :

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```


若要在叢集已啟用的情況下作業，請參閱下列主題：

- 若要使用主控台，請參閱[建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(主控台 \)](#)。
- 若要使用 AWS CLI，請參閱[建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(AWS CLI \)](#)。

步驟3：授予叢集的存取權

本節假設您熟悉啟動和連線至 Amazon EC2執行個體。如需詳細資訊，請參閱 [Amazon EC2 入門指南](#)。

所有 ElastiCache 叢集的設計都是從 Amazon EC2執行個體存取。最常見的案例是從相同 Amazon Virtual Private Cloud (AmazonVPC) 中的 Amazon EC2執行個體存取 ElastiCache 叢集，這將是本練習的情況。

根據預設，僅有用來建立叢集的帳戶可以透過網路存取您的叢集。您必須先授權EC2執行個體存取叢集，才能從EC2執行個體連線至叢集。所需的步驟取決於您是否將叢集啟動至 EC2-VPC 或 EC2-Classical。

最常見的使用案例是部署在EC2執行個體上的應用程式需要連線到相同 中的叢集VPC。管理相同EC2執行個體和叢集之間存取的最簡單方法是VPC執行下列動作：

1. 為您的叢集建立VPC安全群組。此安全群組可用來限制叢集執行個體的存取權限。例如，您可以為此安全群組建立自訂規則，以允許TCP使用您在建立叢集時指派給叢集的連接埠和用於存取叢集的 IP 地址進行存取。

Redis OSS叢集和複寫群組的預設連接埠為 6379。

Important

Amazon ElastiCache 安全群組僅適用於未在 Amazon Virtual Private Cloud 環境 () 中執行的叢集VPC。如果您在 Amazon Virtual Private Cloud 中執行，主控台的導覽窗格中的 Security Groups (安全群組) 就無法使用。

如果您在 Amazon 中執行 ElastiCache 節點VPC，您可以使用與 ElastiCache 安全群組不同的 Amazon VPC安全群組控制對叢集的存取。如需在 Amazon ElastiCache 中使用的詳細資訊VPC，請參閱 [Amazon VPCs和安全性 ElastiCache](#)

2. 為您的EC2執行個體 (網路和應用程式伺服器) 建立VPC安全群組。如有需要，此安全群組可以透過的VPC路由表允許從網際網路存取EC2執行個體。例如，您可以在此安全群組上設定規則，以允許透過連接埠 22 TCP存取EC2執行個體。
3. 在叢集的安全群組中建立自訂規則，以允許來自您為EC2執行個體建立的安全群組的連線。這樣做會允許安全群組的所有成員存取叢集。

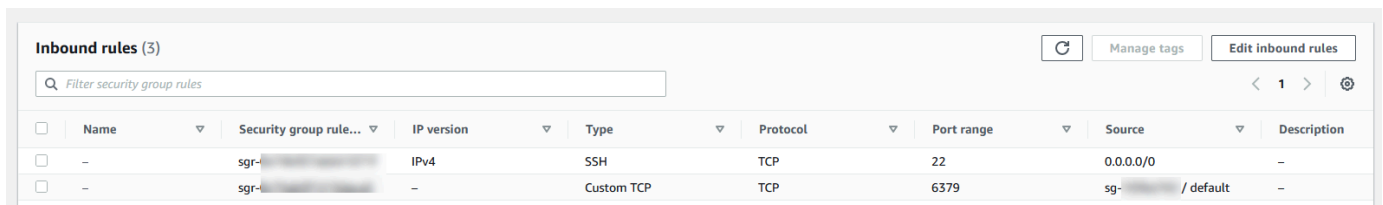
Note

如果您打算使用 [搭配使用本機區域 ElastiCache](#)，請確保您已啟用它們。當您在該本機區域建立子網路群組時，您的 VPC 會延伸到該本機區域，而您的 VPC 會將子網路視為任何其他可用區域中的任何子網路。所有相關閘道和路由表都將自動調整。

在允許從其他 VPC 安全群組連線的安全群組中建立規則

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/vpc> 開啟 Amazon VPC 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選取或建立您將用於叢集執行個體的安全群組。在 Inbound Rules (傳入規則) 下方，選取 Edit Inbound Rules (編輯傳入規則)，然後選取 Add Rule (新增規則)。此安全群組將允許其他安全群組成員存取。
4. 從類型選擇自訂 TCP 規則。
 - a. 針對 Port Range (連接埠範圍)，指定您在建立叢集時所使用的連接埠。

Redis OSS 叢集和複寫群組的預設連接埠為 6379。
 - b. 在 Source (來源) 方塊中輸入安全群組的 ID。從清單中選取您要用於 Amazon EC2 執行個體的安全群組。
5. 完成後，請選擇 Save (儲存)。



	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-
<input type="checkbox"/>	-	sg-...	-	-	-	-	-	-

啟用存取權之後，您便可連線到節點，如下節所述。

如需從不同 Amazon VPC、不同 AWS 區域或甚至公司網路存取 ElastiCache 叢集的資訊，請參閱下列內容：

- [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)
- [從外部存取 ElastiCache 資源 AWS](#)

步驟 4：連線至叢集的節點

在繼續之前，請先完成[步驟3：授予叢集的存取權](#)。

本節假設您已建立 Amazon EC2 執行個體，且可與其連線。如需如何執行此操作的指示，請參閱[Amazon EC2 入門指南](#)。

只有在您授權 Amazon EC2 執行個體執行此操作時，才能連線至叢集節點。

尋找您的節點端點

當您的叢集處於可用狀態且您已授權存取該叢集時，您可以登入 Amazon EC2 執行個體並連線至叢集。若要執行此作業，您必須先判斷端點。

尋找 Valkey 或 Redis OSS (停用叢集模式) 叢集的端點 (主控台)

如果 Redis OSS (停用叢集模式) 叢集只有一個節點，則節點的端點會同時用於讀取和寫入。如果叢集有多個節點，則端點類型有三種：主要端點、讀取者端點和節點端點。

主要端點是一律解析為叢集中主要節點 DNS 的名稱。主要端點不會受到您叢集變更的影響 (例如將僅供讀取複本提升至主要角色)。針對寫入活動，我們建議您的應用程式連線到主要端點。

讀取器端點會在 ElastiCache (Redis OSS) 叢集中的所有僅供讀取複本之間平均分割端點的傳入連線。其他因素 (例如應用程式建立連線或應用程式如何 (重新) 使用連線) 將決定流量分佈。隨著複本的新增或移除，讀取器端點會跟著叢集的變更即時保持在最新狀態。您可以將 ElastiCache (Redis OSS) 叢集的多個僅供讀取複本放在在不同的 AWS 可用區域 (AZ)，以確保讀取器端點的高可用性。

Note

讀取者端點並非負載平衡器。它是以循環方式解析為其中一個複本節點的 IP 地址 DNS 的記錄。

針對讀取活動，應用程式也可連線到叢集中的任何節點。與主要端點不同，節點端點會解析至特定端點。若您在叢集中進行變更 (例如新增或刪除複本)，您必須更新您應用程式中的節點端點。

尋找 Redis OSS (停用叢集模式) 叢集的端點

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

2. 從導覽窗格中，選擇 Redis OSS快取。

叢集畫面會顯示清單，其中包含任何現有的 Valkey 或 Redis 無OSS伺服器快取、Redis OSS (停用叢集模式) 叢集和 Redis OSS (啟用叢集模式) 叢集。選擇您在 [建立 Redis OSS \(停用叢集模式 \) 叢集 \(主控台 \)](#) 一節中建立的叢集。

3. 若要尋找叢集的主要和/或讀取器端點，請選擇叢集名稱 (而非選項按鈕)。

▼ Cluster details			
Cluster name	Description	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Redis OSS (停用叢集模式) 叢集的主要和 Reader 端點

如果叢集中只有一個節點，就不會有主要端點，因此您必須繼續下一個步驟。

4. 如果 Redis OSS (停用叢集模式) 叢集有複本節點，您可以選擇叢集的名稱，然後選擇節點索引標籤，以尋找叢集的複本節點端點。

隨即顯示節點畫面，其顯示叢集中的每個節點、主要複本和複本，並與其端點一起列出。

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-002	available	replica	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-003	available	replica	6379	[redacted].amazonaws.com

Redis OSS (停用叢集模式) 叢集的節點端點

5. 將端點複製到剪貼簿：

a. 逐一尋找您要複製的端點。

- b. 選擇端點前面的複製圖示。

現在，端點即已複製到您的剪貼簿。如需使用端點連線至節點的相關資訊，請參閱 [連線至節點](#)。

Redis OSS (停用叢集模式) 主要端點看起來如下所示。其中的差異取決於是否啟用傳輸中加密。

未啟用傳輸中加密

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

已啟用傳輸中加密

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

若要進一步探索您的端點，請參閱您所執行引擎和叢集類型的相關主題。

- [在中尋找連線端點 ElastiCache](#)
- [尋找 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(主控台 \) 的端點](#) - 您需要叢集的組態端點。
- [尋找端點 \(AWS CLI\)](#)
- [尋找端點 \(ElastiCache API\)](#)

連線至 Valkey 或 Redis OSS叢集或複寫群組 (Linux)

現在您已擁有所需的端點，您可以登入EC2執行個體並連線至叢集或複寫群組。在下列範例中，您可以使用 valkey-cli 公用程式連線到叢集。最新版本的 valkey-cli 也支援 SSL/TLS 連接啟用加密/驗證的叢集。

下列範例使用執行 Amazon Linux 和 Amazon Linux 2 的 Amazon EC2執行個體。如需搭配其他 Linux 發行版本安裝和編譯 valkey-cli 的詳細資訊，請參閱特定作業系統的文件。

Note

此程序涵蓋使用 `valkey-cli` 公用程式測試連線，僅供計劃外使用。如需支援的用戶端清單，請參閱 [Valkey 文件](#)。如需搭配 AWS SDKs 使用的範例 ElastiCache，請參閱 [教學課程：開始使用 Python 和 ElastiCache](#)。

連線至停用叢集模式的未加密叢集

1. 執行下列命令以連線至叢集並取代 `primary-endpoint` 以及 `port number` 叢集的端點和連接埠號碼。（Valkey 和 Redis 的預設連接埠OSS為 6379。）

```
src/valkey-cli -h primary-endpoint -p port number
```

命令提示字元的結果看起來類似於以下內容：

```
primary-endpoint:port number
```

2. 您現在可以執行 Valkey 和 Redis OSS 命令。

```
set x Hello
OK

get x
"Hello"
```

連線至啟用叢集模式的未加密叢集

1. 執行下列命令以連線至叢集並取代 `configuration-endpoint` 以及 `port number` 叢集的端點和連接埠號碼。（Valkey 和 Redis 的預設連接埠OSS為 6379。）

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

Note

在上述命令中，選項 `-c` 會在重新導向和ASK MOVED之後啟用叢集模式。

命令提示字元的結果看起來類似於以下內容：

```
configuration-endpoint:port number
```

2. 您現在可以執行 Valkey 和 Redis OSS命令。請注意，發生重新引導是因為您使用 `-c` 選項啟用了此功能。如果未啟用重新導向，命令會傳回MOVED錯誤。如需MOVED錯誤的詳細資訊，請參閱[叢集規格](#)。

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

連線到啟用加密/身分驗證的叢集

根據預設，`valkey-cli` 在連線至 Valkey 和 Redis 時使用未加密的TCP連線OSS。選項會在編譯 `valkey-cli` 時 `BUILD_TLS=yes` 啟用 SSL/TLS，如上[下載並設定命令列存取](#)一節所示。啟用AUTH是選用的。不過，您必須啟用傳輸中加密，才能啟用 AUTH。如需加密和身分驗證的詳細資訊 ElastiCache，請參閱 [ElastiCache 傳輸中加密 \(TLS \)](#)。

Note

您可以 `--tls` 搭配 `valkey-cli` 使用 選項來連線至啟用和停用的叢集模式加密叢集。如果叢集具有AUTH權杖集，您可以使用 選項 `-a` 提供AUTH密碼。

在下列範例中，請務必取代 *cluster-endpoint* 以及 *port number* 叢集的端點和連接埠號碼。
(Redis 的預設連接埠OSS為 6379。)

連線到停用叢集模式的加密叢集

下列範例會連線到啟用加密和身分驗證的叢集：

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

下列範例會連線到只啟用加密的叢集：

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

連線到啟用叢集模式的加密叢集

下列範例會連線到啟用加密和身分驗證的叢集：

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

下列範例會連線到只啟用加密的叢集：

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

連線至叢集後，您可以執行 Valkey 或 Redis OSS命令，如上述未加密叢集的範例所示。

valkey-cli 或 Redis-cli 的替代方案

如果叢集未啟用叢集模式，且您需要連線至叢集進行簡短測試，但不需要經過 valkey-cli 或 redis-cli 編譯，您可以使用 telnet 或 openssl。在下列範例命令中，請務必取代 *cluster-endpoint* 以及 *port number* 叢集的端點和連接埠號碼。(Redis 的預設連接埠OSS為 6379。)

下列範例會連線到啟用加密和/或身分驗證且停用叢集模式的叢集：

```
openssl s_client -connect cluster-endpoint:port number
```

如果叢集已設定密碼，請先連線至叢集。連線後，請使用下列命令驗證叢集，然後按 Enter 鍵。在下列範例中，取代 *your-password* 使用叢集的密碼。

```
Auth your-password
```

下列範例會連線到沒有啟用加密和/或身分驗證且停用叢集模式的叢集：

```
telnet cluster-endpoint port number
```

連線至 Valkey 或 Redis OSS 叢集或複寫群組 (Windows)

若要使用 Valkey 或 Redis 從 EC2 Windows OSS 執行個體連線至叢集 CLI，您必須下載 valkey-cli 套件，並使用 valkey-cli.exe 從 EC2 Windows 執行個體連線至 Valkey 或 Redis OSS 叢集。

在下列範例中，您可以使用 valkey-cli 公用程式連線到未啟用加密且執行 Valkey 或 Redis 的叢集 OSS。如需 Valkey 和可用命令的詳細資訊，請參閱 [Valkey 網站上的 Valkey 命令](#)。

使用 valkey-cli 連線至未啟用加密的 Valkey 或 Redis OSS 叢集

1. 使用您選擇的連線公用程式連線至您的 Amazon EC2 執行個體。如需如何連線至 Amazon EC2 執行個體的指示，請參閱 [Amazon EC2 入門指南](#)。
2. 在網際網路瀏覽器 <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> 中複製並貼上連結，以從的可用版本下載 Redis OSS 用戶端的 zip 檔案 GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

將 zip 檔案解壓縮到所需的資料夾/路徑。

開啟命令提示字元並變更為 Valkey 目錄，然後執行命令 `c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379`。

例如：

```
c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. 執行 Valkey 或 Redis OSS 命令。

您現在已連線至叢集，可以執行 Valkey 或 Redis OSS 命令，如下所示。

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"
```

```
get b // wait >= 5 seconds
(nil) // key has expired, nothing returned
quit // Exit from valkey-cli
```

刪除叢集

一旦叢集處於「可用」狀態，就會開始向您收費，不論您是否主動使用亦同。若要停止產生費用，請刪除叢集。

Warning

- 刪除 ElastiCache 叢集時，會保留手動快照。您也可以刪除叢集之前建立最終快照。自動快取快照則不會保留。如需詳細資訊，請參閱[快照和還原](#)。
- CreateSnapshot 建立最終快照需要許可。如果沒有此許可，API 呼叫將會失敗，但有 Access Denied 例外狀況。

使用 AWS Management Console

以下程序會從您的部署中刪除單一叢集。若要刪除多個叢集，請針對每個要刪除的叢集重複此程序。您不需要等待某個叢集完成刪除，即可開始刪除其他叢集。

刪除叢集

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在 ElastiCache 引擎儀表板中，選擇 Valkey 或 Redis OSS。

在該引擎上執行的所有快取清單隨即顯示。

3. 要選擇欲刪除的叢集，請從叢集清單選擇叢集的名稱。此案例中為您「[步驟 2：建立叢集](#)」建立的叢集之名稱。

Important

您一次只能從 ElastiCache 主控台刪除一個叢集。選擇多個叢集會停用刪除操作。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。

5. 在刪除叢集確認畫面中，輸入叢集的名稱並選擇最終備份。然後選擇刪除以刪除叢集，或選擇取消以保留叢集。

如果您選擇 Delete (刪除)，叢集的狀態就會變更為 deleting (正在刪除)。

一旦您的叢集不再列於叢集清單，您就不會再因此產生費用。

使用 AWS CLI

下列程式碼會刪除 my-cluster 快取叢集。此案例中請將 my-cluster 取代為您在「[步驟 2：建立叢集](#)」建立的叢集之名稱。

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

此 delete-cache-cluster CLI 動作只會刪除一個快取叢集。若要刪除多個快取叢集，請針對每個要刪除的快取叢集呼叫 delete-cache-cluster。您不需要等待某個快取叢集完成刪除，即可刪除其他叢集。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

針對 Windows：

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

如需詳細資訊，請參閱 ElastiCache 主題 AWS CLI 的 [delete-cache-cluster](#)。

其他 ElastiCache 教學課程和影片

下列教學課程將解決 Amazon ElastiCache 使用者感興趣的任務。

- [ElastiCache 影片](#)
- [教學課程：設定 Lambda 函數以在 Amazon ElastiCache 中存取 Amazon VPC](#)

ElastiCache 影片

以下是協助您學習基本和進階 Amazon ElastiCache 概念的影片。如需 AWS 訓練的相關資訊，請參閱[AWS 訓練與認證](#)。

主題

- [介紹影片](#)
- [進階影片](#)

介紹影片

下列影片向您介紹 Amazon ElastiCache。

主題

- [AWS re : Invent 2020 : Amazon 新功能 ElastiCache](#)
- [AWS re : Invent 2019 : Amazon 新功能 ElastiCache](#)
- [AWS re : Invent 2017 : Amazon 新功能 ElastiCache](#)
- [DAT204—在 AWS 無SQL服務上建置可擴展應用程式 \(re : Invent 2015 \)](#)
- [DAT207—使用 Amazon 加速應用程式效能 ElastiCache \(AWS re : Invent 2013 \)](#)

[AWS re : Invent 2020 : Amazon 新功能 ElastiCache](#)

[AWS re : Invent 2020 : Amazon 新功能 ElastiCache](#)

[AWS re : Invent 2019 : Amazon 新功能 ElastiCache](#)

[AWS re : Invent 2019 : Amazon 新功能 ElastiCache](#)

[AWS re : Invent 2017 : Amazon 新功能 ElastiCache](#)

[AWS re : Invent 2017 : Amazon 新功能 ElastiCache](#)

[DAT204—在 AWS 無SQL服務上建置可擴展應用程式 \(re : Invent 2015 \)](#)

在本課程中，我們會討論無SQL資料庫的優點，並瀏覽 AWS Amazon DynamoDB 和 Amazon 提供的主要無SQL服務 ElastiCache。然後，我們從兩個主要客戶 Expedia 和 Mapbox 那裡聽到有關其使用案例和架構挑戰的資訊，以及他們如何使用 AWS 無SQL服務來解決這些問題，包括設計模式和最佳實務。您應該從本課程中進一步了解否SQL及其強大的功能，準備好自信地應對您的資料庫挑戰。

[DAT204—在 AWS 無SQL服務上建置可擴展應用程式 \(re : Invent 2015 \)](#)

[DAT207—使用 Amazon 加速應用程式效能 ElastiCache \(AWS re : Invent 2013 \)](#)

在本影片中，了解如何使用 Amazon ElastiCache 輕鬆部署記憶體內快取系統，以加速應用程式效能。我們向您展示如何使用 Amazon ElastiCache 來改善應用程式延遲並減少資料庫伺服器的負載。我們也會向您示範如何建置可輕鬆管理，並且可隨著您應用程式成長而擴展的快取 layer。在本課程中，我們會討論各種案例和使用案例，這些案例可透過啟用快取而受益，並討論 Amazon 提供的功能 ElastiCache。

[DAT207 - 使用 Amazon 加速應用程式效能 ElastiCache \(re : Invent 2013 \)](#)

進階影片

下列影片涵蓋更進階的 Amazon ElastiCache 主題。

主題

- [Amazon ElastiCache 最佳實務的成功設計 \(re : Invent 2020 \)](#)
- [使用 Amazon 提升您的即時應用程式 ElastiCache \(re : Invent 2019 \)](#)
- [最佳實務：將 Redis OSS叢集從 Amazon 遷移EC2至 ElastiCache \(re : Invent 2019 \)](#)
- [使用 Amazon ElastiCache 和 Amazon Aurora 擴展 Fantasy Sports Platform STP11 \(re : Invent 2018 \)](#)
- [Amazon 雲端OSS中可靠且可擴展的 Redis ElastiCache \(re : Invent 2018 \)](#)
- [ElastiCache Deep Dive：記憶體內資料存放區的設計模式 \(re : Invent 2018 \)](#)
- [DAT305—Amazon ElastiCache Deep Dive \(re : Invent 2017 \)](#)
- [DAT306—Amazon ElastiCache Deep Dive \(re : Invent 2016 \)](#)
- [DAT317—如何使用 IFTTT ElastiCache \(Redis OSS \) 預測事件 \(re : Invent 2016 \)](#)
- [DAT407—Amazon ElastiCache Deep Dive \(re : Invent 2015 \)](#)
- [SDD402—Amazon ElastiCache Deep Dive \(re : Invent 2014 \)](#)
- [DAT307—深入探討 Amazon ElastiCache 架構和設計模式 \(re : Invent 2013 \)](#)

[Amazon ElastiCache 最佳實務的成功設計 \(re : Invent 2020 \)](#)

隨著以 Redis 建置的業務關鍵即時應用程式爆炸性成長OSS，可用性、可擴展性和安全性已成為首要考量。透過線上擴展、跨多可用區域部署的高可用性和安全性組態，了解設定 Amazon ElastiCache 以取得成功的最佳實務。

[Amazon ElastiCache 最佳實務的成功設計 \(re : Invent 2020 \)](#)

使用 Amazon 提升您的即時應用程式 ElastiCache (re : Invent 2019)

隨著雲端採用的快速成長以及這股趨勢帶來的新使用情境，應用程式需要微秒延遲和高輸送量，才能支援每秒數百萬個請求。開發人員傳統上依賴專門的硬體和因應措施 (例如磁碟資料庫結合資料縮減技術)，藉此管理即時應用程式的資料。這些做法可能很所費不貲，而且不可擴展。了解如何使用完全受管的記憶體內 Amazon 來提升即時應用程式的效能，ElastiCache 以獲得極致效能、高可擴展性、可用性和安全性。

[使用 Amazon 大幅增強您的即時應用程式 ElastiCache \(re : Invent 2019 : \)](#)

最佳實務：將 Redis OSS叢集從 Amazon 遷移EC2至 ElastiCache (re : Invent 2019)

自行管理 Redis OSS叢集可能很困難。您必須時常佈建硬體、修補軟體、備份資料以及監控工作負載。使用 Amazon 新發行的線上遷移功能 ElastiCache，您現在可以輕鬆地將資料從 Amazon OSS 上的自我託管 Redis 移至EC2完全受管的 Amazon ElastiCache，並停用叢集模式。在本課程中，您將了解新的線上遷移工具、參閱示範，更重要的是，您將了解順利遷移至 Amazon 的實作最佳實務 ElastiCache。

[最佳實務：將 Redis OSS叢集從 Amazon 遷移EC2至 ElastiCache \(re : Invent 2019 \)](#)

使用 Amazon ElastiCache 和 Amazon Aurora 擴展 Fantasy Sports Platform STP11 (re : Invent 2018)

Dream11 是印度領先業界的體育科技新創公司。他們超過 4 千萬的成長中使用者群涉及各種運動，包括 Fantasy 板球、足球和籃球，目前可同時為 100 萬名使用者提供服務，且能在 50 毫秒回應時間內每分鐘產生 300 萬個請求。在本演講中，Dream11 CTO Amit Sharma 會說明公司如何使用 Amazon Aurora 和 Amazon ElastiCache 來處理快閃記憶體流量，這在 30 秒的回應時段內可能會增加三倍。Sharma 也會討論如何在不進行鎖定的情況下擴展交易量，並分享處理瞬間流量的步驟，進而為 500 萬使日活躍使用者提供服務。Complete Title : AWS re : Invent 2018 : 使用 Amazon ElastiCache 和 Amazon Aurora 擴展 Fantasy Sports Platform (STP11)

[使用 Amazon ElastiCache 和 Amazon Aurora 擴展 Fantasy Sports Platform STP11 \(re : Invent 2018 \)](#)

Amazon 雲端OSS中可靠且可擴展的 Redis ElastiCache (re : Invent 2018)

本課程涵蓋 Redis OSS相容服務 Amazon ElastiCache (Redis) 的功能和增強功能OSS。我們涵蓋主要功能，例如 Redis OSS 5、可擴展性和效能改善、安全性和合規性等。我們也會討論即將推出的功能和客戶使用案例。

[Amazon 雲端OSS中可靠且可擴展的 Redis ElastiCache \(re : Invent 2018 \)](#)

ElastiCache Deep Dive : 記憶體內資料存放區的設計模式 (re : Invent 2018)

在本課程中，我們提供幕後介紹，以了解 Amazon 的設計和架構 ElastiCache。請參閱我們的 Redis OSS和 Memcached 產品常見的設計模式，以及客戶如何使用它們進行記憶體內資料處理，以減少延遲並改善應用程式輸送量。我們會檢閱 ElastiCache 最佳實務、設計模式和反模式。

[ElastiCache Deep Dive : 記憶體內資料存放區的設計模式 \(re : Invent 2018 \)](#)

DAT305—Amazon ElastiCache Deep Dive (re : Invent 2017)

看看幕後，了解 Amazon ElastiCache的設計和架構。請參閱我們的 Memcached 和 Redis OSS產品常見的設計模式，以及客戶如何使用它們進行記憶體內操作，以減少延遲並改善應用程式輸送量。在此影片中，我們會檢閱 ElastiCache 最佳實務、設計模式和反模式。

影片會介紹以下內容：

- ElastiCache (Redis OSS) 線上轉印
- ElastiCache 安全性和加密
- ElastiCache (Redis OSS) 3.2.10 版

[DAT305—Amazon ElastiCache Deep Dive \(re : Invent 2017 \)](#)

DAT306—Amazon ElastiCache Deep Dive (re : Invent 2016)

看看幕後，了解 Amazon ElastiCache的設計和架構。請參閱我們的 Memcached 和 Redis OSS產品常見的設計模式，以及客戶如何使用它們進行記憶體內操作，以減少延遲並改善應用程式輸送量。在本課程中，我們會檢閱 ElastiCache 最佳實務、設計模式和反模式。

[DAT306—Amazon ElastiCache Deep Dive \(re : Invent 2016 \)](#)

DAT317—如何使用 IFTTT ElastiCache (Redis OSS) 預測事件 (re : Invent 2016)

IFTTT 是一項免費服務，可讓人們使用他們喜愛的服務完成更多工作，從自動化簡單任務到轉換某人與住家互動和控制的方式。IFTTT 使用 ElastiCache (Redis OSS) 在 Amazon S3 上儲存交易執行歷史記錄和排程預測，以及日誌文件的索引。檢視此工作階段，以了解 Lua 的指令碼編寫能力和 Redis 的資料類型如何OSS讓人們完成在其他地方無法完成的任務。

[DAT317—如何使用 IFTTT ElastiCache \(Redis OSS \) 預測事件 \(re : Invent 2016 \)](#)

[DAT407—Amazon ElastiCache Deep Dive \(re : Invent 2015 \)](#)

深入了解幕後，了解 Amazon ElastiCache 的設計和架構。查看 Memcached 和 Redis OSS 產品的常見設計模式，以及客戶如何使用它們進行記憶體內操作，並提高應用程式的延遲和輸送量。在本課程中，我們會檢閱與 Amazon 相關的最佳實務、設計模式和反模式 ElastiCache。

[DAT407—Amazon ElastiCache Deep Dive \(re : Invent 2015 \)](#)

[SDD402—Amazon ElastiCache Deep Dive \(re : Invent 2014 \)](#)

在本影片中，我們會檢查常見的快取使用案例、Memcached 和 Redis OSS 引擎、可協助您判斷哪個引擎更符合您需求的模式、一致的雜湊等，以做為建置快速、可擴展應用程式的方法。Adobe 首席科學家 Frank Wiebe 詳細說明 Adobe 如何使用 Amazon ElastiCache 來改善客戶體驗並擴展業務。

[DAT402—Amazon ElastiCache Deep Dive \(re : Invent 2014 \)](#)

[DAT307—深入探討 Amazon ElastiCache 架構和設計模式 \(re : Invent 2013 \)](#)

在此影片中，我們會檢視快取、策略、向外擴展與監控。我們也比較了 Memcached 和 Redis OSS 引擎。在本課程中，我們也會檢閱與 Amazon 相關的最佳實務和設計模式 ElastiCache。

[DAT307 - 深入探討 Amazon ElastiCache 架構和設計模式 \(AWS re : Invent 2013 \)](#)。

在中管理節點 ElastiCache

節點是 Amazon ElastiCache 部署的最小建置區塊。這是固定大小的安全網路連接 區塊 RAM。每個節點都會執行當您建立或最後一次修改叢集時所選擇的引擎。每個節點都有自己的網域名稱服務 (DNS) 名稱和連接埠。支援多種類型的 ElastiCache 節點，每個節點都有不同數量的關聯記憶體和運算能力。

如需要使用何種節點大小的詳細討論，請參閱「[選擇您的節點大小](#)」。

一般而言，由於支援分割，Valkey 或 Redis OSS (啟用叢集模式) 部署具有許多較小的節點。相反地，Valkey 或 Redis OSS (停用叢集模式) 部署在叢集中的節點較少、較大。如需要使用何種節點大小的詳細討論，請參閱「[選擇您的節點大小](#)」。

主題

- [檢視 ElastiCache 節點狀態](#)
- [Valkey 或 Redis OSS 節點和碎片](#)
- [連線至節點](#)
- [支援的節點類型](#)

- [重新啟動節點](#)
- [更換節點 \(Valkey 和 RedisOSS \)](#)
- [更換節點 \(Memcached \)](#)
- [預留節點](#)
- [遷移上一代節點](#)

與節點有關的一些重要操作如下：

- [將節點新增至 ElastiCache 叢集](#)
- [從 ElastiCache 叢集移除節點](#)
- [擴展 ElastiCache](#)
- [在 中尋找連線端點 ElastiCache](#)
- [自動識別叢集中的節點 \(Memcached \)](#)

檢視 ElastiCache 節點狀態

使用 [ElastiCache 主控台](#)，您可以快速存取 ElastiCache 節點的狀態。ElastiCache 節點的狀態表示節點的運作狀態。您可以使用下列程序，在 Amazon ElastiCache 主控台、AWS CLI 命令或 API 操作中檢視 ElastiCache 節點狀態。

節點的可能狀態值 ElastiCache 如下表所示。此資料表也會顯示是否將向您收取 ElastiCache 節點的費用。

類型	計費	Description
available	計費	ElastiCache 節點運作良好且可用。
creating	不計費	正在建立 ElastiCache 節點。節點正在建立時無法進行存取。
deleting	不計費	正在刪除 ElastiCache 節點。
modifying	計費	由於客戶請求修改 ElastiCache 節點，因此正在修改節點。

類型	計費	Description
updating	計費	<p>更新狀態表示 Amazon ElastiCache 節點符合下列一項或多項條件：</p> <ul style="list-style-type: none"> • ElastiCache 節點正在修補，作為服務 update 的一部分。如需服務更新的詳細資訊，請參閱 Amazon ElastiCache Managed Maintenance and Service Updates 說明頁面。 • VPC 安全群組正在更新 ElastiCache 叢集。 • ElastiCache 叢集正在擴展或縮減。 • 正在修改 ElastiCache 叢集的日誌傳遞組態。 • ElastiCache 節點的刪除操作處於擱置狀態。 • 使用更新/輪換 ElastiCache 具有 Valkey 或 Redis OSS密碼的 AWS Secrets Manager。
rebooting cache cluster nodes	計費	<p>由於客戶請求或 Amazon ElastiCache 程序需要重新啟動節點，因此節點 ElastiCache 正在重新啟動。</p>

類型	計費	Description
<code>incompatible_parameters</code>	不計費	Amazon ElastiCache 無法啟動節點，因為節點參數群組中指定的參數與節點不相容。請回復參數變更，或讓參數與節點相容，以恢復節點的存取。如需不相容參數的詳細資訊，請檢查 ElastiCache 節點的事件清單 。
<code>incompatible_network</code>	不計費	不相容的網路狀態表示 Amazon ElastiCache 節點符合下列一或多個條件： <ul style="list-style-type: none">• ElastiCache 節點啟動的子網路中沒有可用的 IP 地址。• 子網路群組中提到的 ElastiCache 子網路不再存在於 Amazon Virtual Private Cloud (Amazon) 中 VPC。

類型	計費	Description
restore_failed	不計費	<p>還原失敗狀態表示 Amazon ElastiCache 節點符合下列其中一項條件：</p> <ul style="list-style-type: none"> 由於重複發生 執行個體容量不足 的情況，因此更換節點失敗。這通常發生在執行上一代節點時 end-of-life。不過，如果 AWS 沒有足夠的隨需容量來滿足您在指定可用區域中的請求，則替換目前世代節點也會發生這種情況。如需修正或移除這些節點的詳細資訊，請參閱 遷移上一代節點。 指定的RDB快照無法還原。 ElastiCache 叢集 AWS 的帳戶已暫停。 節點失敗，無法復原。
snapshotting	計費	ElastiCache 正在建立 Valkey 或 Redis OSS節點的快照。

使用主控台檢視 ElastiCache 節點狀態

若要使用主控台檢視 ElastiCache 節點的狀態：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Redis OSS Cluster 或 Memcached Clusters。快取頁面會顯示 ElastiCache 節點清單。每個節點的狀態值都會顯示。
3. 然後，您可以導覽至快取的服務更新索引標籤，以顯示適用於快取的服務更新清單。

使用 檢視 ElastiCache 節點狀態 AWS CLI

若要使用 檢視 ElastiCache 節點及其狀態資訊 AWS CLI，請使用 `describe-cache-cluster` 命令。例如，下列 AWS CLI 命令會顯示每個 ElastiCache 節點。

```
aws elasticache describe-cache-clusters
```

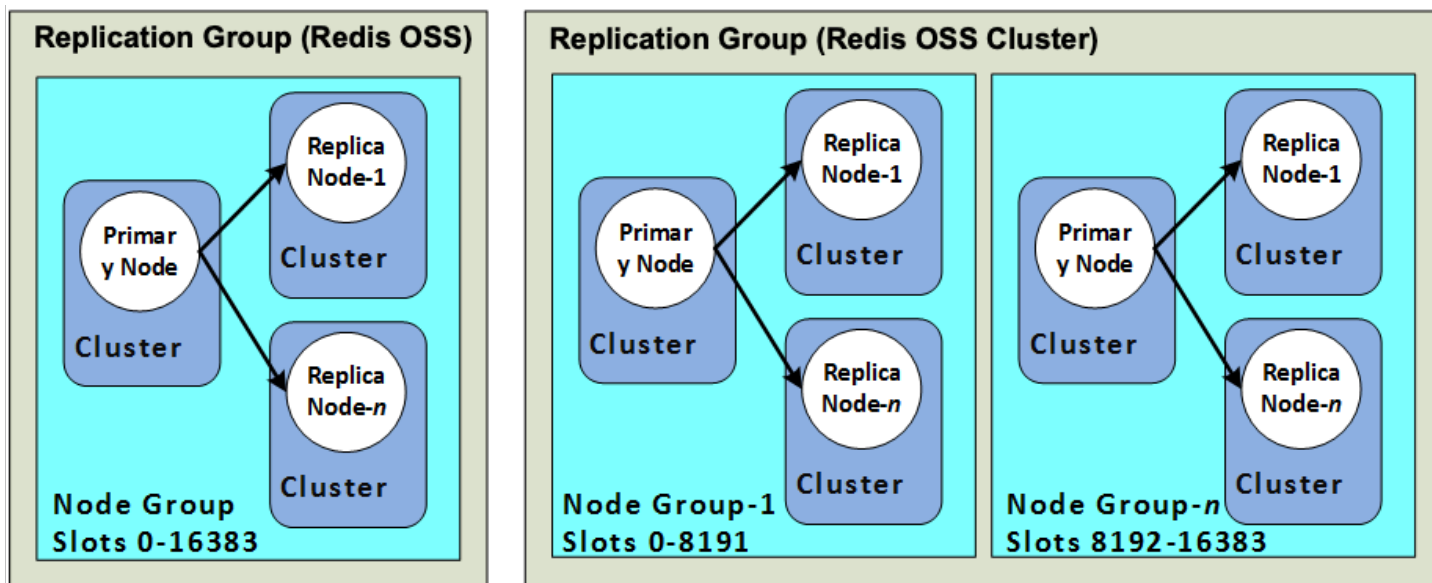
透過 檢視 ElastiCache 節點狀態 API

若要使用 Amazon 來檢視 ElastiCache 節點的狀態 ElastiCache API，請呼叫 `DescribeCacheClusteroperation` 帶有 `ShowCacheNodeInfo` 旗標的，以擷取個別快取節點的相關資訊。

Valkey 或 Redis OSS節點和碎片

碎片（在 API 和 CLI 中，節點群組）是節點的階層排列，每個節點都包裝在叢集中。碎片支援複寫。在一個碎片中，其中一個節點會做為讀取/寫入主要節點。碎片中的所有其他節點都會做為主要節點的唯一讀複本。Valkey 或 Redis 3.2 版及更新 OSS 版本支援叢集內的多個碎片（在 API 和 CLI 中，複寫群組）。此支援可在 Valkey 或 Redis OSS（啟用叢集模式）叢集中分割您的資料。

下圖說明 Valkey 或 Redis OSS（停用叢集模式）叢集與 Valkey 或 Redis OSS（啟用叢集模式）叢集之間的差異。



Valkey 或 Redis OSS（啟用叢集模式）叢集支援透過碎片複寫。API 操作 [DescribeReplicationGroups](#)（CLI：[describe-replication-groups](#)）會列出包含成員節點的節點群組、節點在節點群組中的角色，以及其他資訊。

當您建立 Valkey 或 Redis OSS 叢集時，您可以指定要建立已啟用叢集的叢集。Valkey 或 Redis OSS (停用叢集模式) 叢集永遠不會有多個碎片，可以新增 (最多共五個) 或刪除僅供讀取複本節點，以水平擴展。如需詳細資訊，請參閱「[使用複寫群組的高可用性](#)」、「[新增 Valkey 或 Redis 的僅供讀取複本 OSS \(停用叢集模式 \)](#)」或「[刪除 Valkey 或 Redis 的僅供讀取複本 OSS \(停用叢集模式 \)](#)」。Valkey 或 Redis OSS (停用叢集模式) 叢集也可以透過變更節點類型垂直擴展。如需詳細資訊，請參閱[擴展 Valkey 或 Redis 的複本節點 OSS \(停用叢集模式 \)](#)。

如果引擎是 Valkey 或 Redis 5.0.6 OSS 版或更新版本，則節點或碎片限制可以增加到每個叢集最多 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 (每個碎片一個主要版本和 5 個複本) 到 500 個 (單一主要版本並且沒有複本) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路 CIDR 太小，範圍太小，或者子網路被其他叢集共用和大量使用。如需詳細資訊，請參閱[建立子網路群組](#)。

對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱[AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集後，可以對其進行變更 (放大或縮小)。如需詳細資訊，請參閱[擴展 ElastiCache](#) 和 [更換節點 \(Valkey 和 RedisOSS \)](#)。

建立新叢集時，您可以將舊叢集的資料傳送到新叢集，使其不會在一開始呈現空白狀態。只有在叢集群組擁有的碎片數量與舊叢集相同時，此方法才有效。這在您需要變更節點類型或引擎版本時很有用。如需詳細資訊，請參閱[取得手動備份](#) 和 [從備份還原到新的快取](#)。

連線至節點

連線至 Valkey 或 Redis OSS節點

在嘗試連線到叢集中的 Valkey 或 Redis OSS節點之前，您必須擁有節點的端點。若要尋找端點，請參閱下列主題：

- [尋找 Valkey 或 Redis OSS \(停用叢集模式 \) 叢集的端點 \(主控台 \)](#)
- [尋找 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(主控台 \) 的端點](#)
- [尋找端點 \(AWS CLI\)](#)
- [尋找端點 \(ElastiCache API\)](#)

在下列範例中，您可以使用 valkey-cli 公用程式連線到執行 Valkey 或 Redis 的叢集OSS。

Note

如需可用命令的詳細資訊，請參閱[命令網頁](#)。

使用 valkey-cli 連線至 Valkey 或 Redis OSS叢集

1. 使用您選擇的連線公用程式連線至您的 Amazon EC2執行個體。

Note

如需如何連線至 Amazon EC2執行個體的指示，請參閱 [Amazon EC2 入門指南](#)。

2. 若要建置 valkey-cli，請下載並安裝GNU編譯器集合 (gcc)。在EC2執行個體的命令提示中，輸入下列命令，然後在確認提示y中輸入。

```
sudo yum install gcc
```

隨即出現類似如下的輸出。

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check
```

```
...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB    00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm             | 4.8 MB    00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm               | 2.8 kB    00:00

...(output omitted)...

Complete!
```

3. 下載並編譯 valkey-cli 公用程式。此公用程式包含在 Valkey 軟體分發中。在 EC2 執行個體的命令提示中，輸入下列命令：

Note

針對 Ubuntu 系統，請在執行 make 之前，執行 make distclean。

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean      # ubuntu systems only
make
```

4. 在 EC2 執行個體的命令提示中，輸入下列命令。

```
src/valkey-cli -c -h mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

會出現類似下列的 Valkey 或 Redis OSS 命令提示。

```
redis mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. 透過執行 Valkey 或 Redis OSS 命令來測試連線。

您現在已連線至叢集，可以執行 Valkey 或 Redis OSS 命令。以下是一些包含其 Valkey 或 Redis OSS 回應的範例命令。

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
get b
"Good-bye"

                        // wait 5 seconds
get b
(nil)                   // key has expired, nothing returned
quit                    // Exit from valkey-cli
```

如需連線至具有 Secure Sockets Layer (SSL) 加密 (啟用傳輸中) 的節點或叢集，請參閱 [ElastiCache 傳輸中加密 \(TLS \)](#)。

連線至 Memcached 節點

在嘗試連線到您的 Memcached 叢集前，您必須擁有節點的端點。若要尋找端點，請參閱下列主題：

- [尋找叢集的端點 \(主控台 \) \(Memcached \)](#)
- [尋找端點 \(AWS CLI\)](#)
- [尋找端點 \(ElastiCache API \)](#)

在下列範例中，您會使用 telnet 公用程式連線到執行 Memcached 的節點。

Note

如需 Memcached 和可用 Memcached 命令的詳細資訊，請參閱 [Memcached](#) 網站。

使用 telnet 連線到節點

1. 使用您選擇的連線公用程式連線至您的 Amazon EC2 執行個體。

Note

如需如何連線至 Amazon EC2 執行個體的指示，請參閱 [Amazon EC2 入門指南](#)。

2. 在 Amazon EC2 執行個體上下載並安裝 telnet 公用程式。在 Amazon EC2 執行個體的命令提示中，輸入下列命令，然後在命令提示中輸入 y。

```
sudo yum install telnet
```

隨即出現類似如下的輸出。

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB    00:00

...(output omitted)...

Complete!
```

3. 在 Amazon EC2 執行個體的命令提示中，輸入下列命令，將節點的端點取代為此範例中所示的端點。

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

隨即出現類似如下的輸出。

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^['.
```

```
>
```

4. 透過執行 Memcached 命令來測試連線。

您現在已連線至節點，可以執行 Memcached 命令。以下是範例。

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello           // Set value as "hello"
STORED
get a           // Get value for key "a"
VALUE a 0 5
hello
END
get b           // Get value for key "b" results in miss
END
>
```

支援的節點類型

ElastiCache 支援下列節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需每個節點類型效能詳細資訊的詳細資訊，請參閱 [Amazon EC2 執行個體類型](#)。

如需要使用何種節點大小的詳細資訊，請參閱「[選擇您的節點大小](#)」。

主題

- [目前世代 \(Memcached \)](#)
- [目前世代 \(Valkey 或 RedisOSS \)](#)
- [支援的節點類型 \(依 AWS 區域\)](#)
- [爆量效能執行個體](#)
- [相關資訊](#)

目前世代 (Memcached)

下列表單顯示使用網路 I/O 抵用金機制來超越其基準頻寬之執行個體類型的基準頻寬和高載頻寬。

Note

具有爆量網路效能的執行個體類型使用網路 I/O 點數機制，以竭盡全力爆量到其基準頻寬之外。

一般

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m7g.large		0.937	12.5
cache.m7g.xlarge		1.876	12.5
cache.m7g.2xlarge		3.75	15
cache.m7g.4xlarge		7.5	15

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22.5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1.5.16	0.75	10.0
cache.m6g.xlarge	1.5.16	1.25	10.0
cache.m6g.2xlarge	1.5.16	2.5	10.0
cache.m6g.4xlarge	1.5.16	5.0	10.0
cache.m6g.8xlarge	1.5.16	12	N/A
cache.m6g.12xlarge	1.5.16	20	N/A
cache.m6g.16xlarge	1.5.16	25	N/A
cache.m5.large	1.5.16	0.75	10.0
cache.m5.xlarge	1.5.16	1.25	10.0
cache.m5.2xlarge	1.5.16	2.5	10.0
cache.m5.4xlarge	1.5.16	5.0	10.0
cache.m5.12xlarge	1.5.16	N/A	N/A
cache.m5.24xlarge	1.5.16	N/A	N/A
cache.m4.large	1.5.16	0.45	1.2
cache.m4.xlarge	1.5.16	0.75	2.8
cache.m4.2xlarge	1.5.16	1.0	10.0
cache.m4.4xlarge	1.5.16	2.0	10.0

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m4.10xlarge	1.5.16	5.0	10.0
cache.t4g.micro	1.5.16	0.064	5.0
cache.t4g.small	1.5.16	0.128	5.0
cache.t4g.medium	1.5.16	0.256	5.0
cache.t3.micro	1.5.16	0.064	5.0
cache.t3.small	1.5.16	0.128	5.0
cache.t3.medium	1.5.16	0.256	5.0
cache.t2.micro	1.5.16	0.064	1.024
cache.t2.small	1.5.16	0.128	1.024
cache.t2.medium	1.5.16	0.256	1.024

針對 Memcached 最佳化的記憶體

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r7g.large		0.937	12.5
cache.r7g.xlarge		1.876	12.5
cache.r7g.2xlarge		3.75	15
cache.r7g.4xlarge		7.5	15
cache.r7g.8xlarge		15	N/A
cache.r7g.12xlarge		22.5	N/A
cache.r7g.16xlarge		30	N/A

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r6g.large	1.5.16	0.75	10.0
cache.r6g.xlarge	1.5.16	1.25	10.0
cache.r6g.2xlarge	1.5.16	2.5	10.0
cache.r6g.4xlarge	1.5.16	5.0	10.0
cache.r6g.8xlarge	1.5.16	12	N/A
cache.r6g.12xlarge	1.5.16	20	N/A
cache.r6g.16xlarge	1.5.16	25	N/A
cache.r5.large	1.5.16	0.75	10.0
cache.r5.xlarge	1.5.16	1.25	10.0
cache.r5.2xlarge	1.5.16	2.5	10.0
cache.r5.4xlarge	1.5.16	5.0	10.0
cache.r5.12xlarge	1.5.16	20	N/A
cache.r5.24xlarge	1.5.16	25	N/A
cache.r4.large	1.5.16	0.75	10.0
cache.r4.xlarge	1.5.16	1.25	10.0
cache.r4.2xlarge	1.5.16	2.5	10.0
cache.r4.4xlarge	1.5.16	5.0	10.0
cache.r4.8xlarge	1.5.16	12	N/A
cache.r4.16xlarge	1.5.16	25	N/A

針對 Memcached 最佳化的網路

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.c7gn.large	1.6.6	6.25	30
cache.c7gn.xlarge	1.6.6	12.5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

目前世代 (Valkey 或 RedisOSS)

如需上一代節點的詳細資訊，請參閱[上一代節點](#)。

Note

具有爆量網路效能的執行個體類型使用網路 I/O 點數機制，以竭盡全力爆量到其基準頻寬之外。

一般

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m7g.large	6.2	N	N	N	0.937	12.5
cache.m7g.xlarge	6.2	Y	Y	Y	1.876	12.5
cache.m7g.2xlarge	6.2	Y	Y	Y	3.75	15
cache.m7g.4xlarge	6.2	Y	Y	Y	7.5	15
cache.m7g.8xlarge	6.2	Y	Y	Y	15	N/A
cache.m7g.12xlarge	6.2	Y	Y	Y	22.5	N/A
cache.m7g.16xlarge	6.2	Y	Y	Y	30	N/A
cache.m6g.large	5.0.6	N	N	N	0.75	10.0
cache.m6g.xlarge	5.0.6	Y	Y	Y	1.25	10.0
cache.m6g.2xlarge	5.0.6	Y	Y	Y	2.5	10.0

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m6g .4xlarge	5.0.6	Y	Y	Y	5.0	10.0
cache.m6g .8xlarge	5.0.6	Y	Y	Y	12	N/A
cache.m6g .12xlarge	5.0.6	Y	Y	Y	20	N/A
cache.m6g .16xlarge	5.0.6	Y	Y	Y	25	N/A
cache.m5.large	3.2.4	N	N	N	0.75	10.0
cache.m5.xlarge	3.2.4	Y	N	N	1.25	10.0
cache.m5. 2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.m5. 4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.m5. 12xlarge	3.2.4	Y	Y	Y	12	N/A
cache.m5. 24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.m4.large	3.2.4	N	N	N	0.45	1.2

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m4.xlarge	3.2.4	Y	N	N	0.75	2.8
cache.m4.2xlarge	3.2.4	Y	Y	Y	1.0	10.0
cache.m4.4xlarge	3.2.4	Y	Y	Y	2.0	10.0
cache.m4.10xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.t4g.micro	3.2.4	N	N	N	0.064	5.0
cache.t4g.small	5.0.6	N	N	N	0.128	5.0
cache.t4g.medium	5.0.6	N	N	N	0.256	5.0
cache.t3.micro	3.2.4	N	N	N	0.064	5.0
cache.t3.small	3.2.4	N	N	N	0.128	5.0
cache.t3.medium	3.2.4	N	N	N	0.256	5.0
cache.t2.micro	3.2.4	N	N	N	0.064	1.024
cache.t2.small	3.2.4	N	N	N	0.128	1.024
cache.t2.medium	3.2.4	N	N	N	0.256	1.024

記憶體最佳化

執行個體類型	支援的 Redis OSS版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r7g.large	6.2	N	N	N	0.937	12.5
cache.r7g.xlarge	6.2	Y	Y	Y	1.876	12.5
cache.r7g.2xlarge	6.2	Y	Y	Y	3.75	15
cache.r7g.4xlarge	6.2	Y	Y	Y	7.5	15
cache.r7g.8xlarge	6.2	Y	Y	Y	15	N/A
cache.r7g.12xlarge	6.2	Y	Y	Y	22.5	N/A
cache.r7g.16xlarge	6.2	Y	Y	Y	30	N/A
cache.r6g.large	5.0.6	N	N	N	0.75	10.0
cache.r6g.xlarge	5.0.6	Y	Y	Y	1.25	10.0
cache.r6g.2xlarge	5.0.6	Y	Y	Y	2.5	10.0

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r6g .4xlarge	5.0.6	Y	Y	Y	5.0	10.0
cache.r6g .8xlarge	5.0.6	Y	Y	Y	12	N/A
cache.r6g .12xlarge	5.0.6	Y	Y	Y	20	N/A
cache.r6g .16xlarge	5.0.6	Y	Y	Y	25	N/A
cache.r5.large	3.2.4	N	N	N	0.75	10.0
cache.r5.xlarge	3.2.4	Y	N	N	1.25	10.0
cache.r5.2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.r5.4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.r5.12xlarge	3.2.4	Y	Y	Y	12	N/A
cache.r5.24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	Y	N	N	1.25	10.0

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r4.2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	N/A
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	N/A

利用資料分層最佳化的記憶體

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r6gd.xlarge	6.2.0	Y	N	N	1.25	10
cache.r6gd.2xlarge	6.2.0	Y	Y	Y	2.5	10

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r6g .4xlarge	6.2.0	Y	Y	Y	5.0	10
cache.r6g .8xlarge	6.2.0	Y	Y	Y	12	N/A
cache.r6gd.12xlarge	6.2.0	Y	Y	Y	20	N/A
cache.r6gd.16xlarge	6.2.0	Y	Y	Y	25	N/A

網路最佳化

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.c7gn.large	6.2	N	N	N	6.25	30

執行個體類型	支援的 Redis OSS 版本下限	增強型 I/O (Redis OSS 5.0.6+)	TLS 卸載 (Redis OSS 6.2.5+)	增強型 I/O 多工 (Redis OSS 7.0.4+)	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.c7g n.xlarge	6.2	Y	Y	Y	12.5	40
cache.c7g n.2xlarge	6.2	Y	Y	Y	25	50
cache.c7g n.4xlarge	6.2	Y	Y	Y	50	N/A
cache.c7g n.8xlarge	6.2	Y	Y	Y	100	N/A
cache.c7g n.12xlarge	6.2	Y	Y	Y	150	N/A
cache.c7g n.16xlarge	6.2	Y	Y	Y	200	N/A

支援的節點類型 (依 AWS 區域)

支援的節點類型可能因 AWS 區域而異。如需詳細資訊，請參閱 [Amazon ElastiCache 定價](#)。

爆量效能執行個體

您可以在 Amazon 中啟動一般用途爆量 T4g, T3-Standard 和 T2-Standard 快取節點 ElastiCache。這些節點提供基準 CPU 效能層級，並可隨時爆量 CPU 用量，直到累積的額度用盡為止。CPU 額度提供一分鐘內完整 CPU 核心的效能。

Amazon ElastiCache的 T4g, T3 和 T2 節點設定為標準節點，並適用於平均CPU使用率持續低於執行個體基準效能的工作負載。為了使 超過基準，節點會花費其在點數餘額中累積的CPU點數。若節點的累積點數不足，效能會逐漸降低至基準效能等級。此逐漸降低可確保節點在累積CPU的額度餘額耗盡時不會經歷劇烈的效能下降。如需詳細資訊，請參閱 Amazon 使用者指南中的 [CPU 點數和 Burstable Performance Instances 的基準效能](#)。 EC2

下表列出爆量效能節點類型，即每小時獲得CPU額度的速率。它也會顯示節點可累積的CPU點數上限，以及 vCPUs 每個節點的點數。此外，它以完整核心效能的百分比提供基準效能等級（使用單一 v CPU）。

CPU 每小時獲得的點數	可累積的 額度上限*	vCPUs	每個 v 的 基準效 能CPU	記憶 體 (GiB)	網路效能
12	288	2	10%	0.5	最多 5 GB
24	576	2	20%	1.37	最多 5 GB
24	576	2	20%	3.09	最多 5 GB
12	288	2	10%	0.5	最多 5 GB
24	576	2	20%	1.37	最多 5 GB
24	576	2	20%	3.09	最多 5 GB
6	144	1	10%	0.5	低至中度
12	288	1	20%	1.55	低至中度
24	576	2	20%	3.22	低至中度

* 可累計的額度等於在 24 小時期間可獲得的額度。

** 資料表中的基準效能是每個 v CPU。有些節點大小具有多個 v CPU。對於這些，請將 vCPU 百分比乘以的數量，以計算節點的基準CPU使用率vCPUs。

下列CPU點數指標適用於 T3 和 T4g 爆量效能執行個體：

Note

這些指標不適用於 T2 高載效能執行個體。

- CPUCreditUsage
- CPUCreditBalance

如需這些指標的詳細資訊，請參閱 [CPU 額度指標](#)。

此外，也請注意這些細節：

- 根據VPC預設，所有目前世代節點類型都是在以 Amazon 為基礎的虛擬私有雲端（VPC）中建立。
- T2 執行個體不支援 Redis 僅OSS附加檔案（AOF）。Redis 2.8.22 版appendonly及更新OSS版本appendfsync不支援 Redis OSS組態變數和。

相關資訊

- [Amazon ElastiCache 產品功能和詳細資訊](#)
- [Memcached 的 Memcached Node-Type 特定參數](#)
- [Valkey 和 Redis OSS 參數](#)
- [傳輸中加密（TLS）](#)

重新啟動節點

有些變更需要重新啟動 Redis OSS或 Memcached 叢集，才能套用變更。例如，針對某些參數，當您變更參數群組中的參數值時，只會在重新啟動時套用。

主題

- [重新啟動 Redis OSS節點（僅停用叢集模式）](#)
- [為 Memcached 重新啟動叢集](#)

重新啟動 Redis OSS節點（僅停用叢集模式）

對於 Valkey 或 Redis OSS（停用叢集模式）叢集，只有在重新啟動後才套用的參數群組中的參數為：

- activerehashing
- databases

Redis 節點只能透過 ElastiCache 主控台更新。您一次只能重新啟動單一節點。若要重新啟動多個節點，您必須為每個節點重複此程序。

Valkey 或 Redis OSS (已啟用叢集模式) 參數變更

如果您在 Valkey 或 Redis OSS (啟用叢集模式) 叢集上對下列參數進行變更，請遵循後續步驟。

- activerehashing
- databases

1. 建立叢集的手動備份。請參閱 [取得手動備份](#)。
2. 刪除 Valkey 或 Redis OSS (啟用叢集模式) 叢集。請參閱 [在中刪除叢集 ElastiCache](#)。
3. 您可以使用更改的參數群組和備份來植入新的叢集，以還原叢集。請參閱 [從備份還原到新的快取](#)。

對其他參數進行變更不需要這樣做。

使用 AWS Management Console

您可以使用 ElastiCache 主控台重新啟動節點。

重新啟動節點 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇適用的 AWS 區域。
3. 在左側導覽窗格中，選擇 Redis OSS。

隨即 OSS 出現執行 Redis 的叢集清單。

4. 在 Cluster Name (叢集名稱) 底下選擇叢集。
5. 在 Node name (節點名稱) 下，選擇您要重新啟動的節點旁邊的選項按鈕。

6. 選擇 Actions (動作), 然後選擇 Reboot node (重新啟動節點)。

若要重新啟動多個節點, 請對每個要重新啟動的節點重複步驟 2 到 5。您不需要等待某個節點完成重新啟動, 即可重新啟動其他節點。

為 Memcached 重新啟動叢集

當您重新啟動 Memcached 叢集時, 叢集會清除所有資料並重新啟動其引擎。程序進行期間, 您無法存取叢集。由於叢集會排清其所有資料, 因此當叢集再次可用時, 就是一個空的叢集供您使用。

您可以使用 ElastiCache 主控台、AWS CLI 或重新啟動叢集 ElastiCache API。無論您是使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API, 您都只能啟動重新啟動單一叢集。若要重新啟動多個叢集, 您必須反覆執程序和操作。

使用 AWS Management Console

您可以使用 ElastiCache 主控台重新啟動叢集。

重新啟動叢集 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中, 選擇您感興趣的 AWS 區域。
3. 在導覽窗格中, 選擇您要重新啟動之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

4. 選擇叢集名稱左側的方塊, 以選擇要重新啟動的叢集。

Reboot (重新啟動) 按鈕會變為作用中。

如果您選擇一個以上的叢集, Reboot (重新啟動) 按鈕則不會啟用。

5. 選擇 Reboot (重新啟動)。

重新啟動叢集確認畫面隨即出現。

6. 若要重新啟動叢集, 請選擇 Reboot (重新啟動)。叢集狀態會變更為 rebooting cluster nodes (重新啟動叢集節點中)。

若不要重新啟動叢集, 請選擇 Cancel (取消)。

若要重新啟動多個叢集，請針對每個要重新啟動的叢集重複步驟 2 到 5。您不需要等待某個叢集完成重新啟動，即可重新啟動其他叢集。

若要重新啟動特定節點，請選擇節點，然後選擇 Reboot (重新啟動)。

使用 AWS CLI

若要重新啟動叢集 (AWS CLI)，請使用 `reboot-cache-cluster` CLI 操作。

若要重新啟動叢集中的特定節點，請使用 `--cache-node-ids-to-reboot` 列出要重新啟動的特定叢集。下列命令會重新啟動 `my-cluster` 的 0001、0002 和 0004 節點。

若為 Linux、macOS 或 Unix：

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --cache-node-ids-to-reboot 0001 0002 0004
```

針對 Windows：

```
aws elasticache reboot-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --cache-node-ids-to-reboot 0001 0002 0004
```

若要重新啟動叢集中的所有節點，請使用 `--cache-node-ids-to-reboot` 參數並列出所有叢集的節點 ID。如需詳細資訊，請參閱 [reboot-cache-cluster](#)。

使用 ElastiCache API

若要使用重新啟動叢集 ElastiCache API，請使用 `RebootCacheCluster` 動作。

若要重新啟動叢集中的特定節點，請使用 `CacheNodeIdsToReboot` 列出要重新啟動的特定叢集。下列命令會重新啟動 `my-cluster` 的 0001、0002 和 0004 節點。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RebootCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeIdsToReboot.member.1=0001  
&CacheNodeIdsToReboot.member.2=0002  
&CacheNodeIdsToReboot.member.3=0004  
&Version=2015-02-02  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

若要重新啟動叢集中的所有節點，請使用 `CacheNodeIdsToReboot` 參數並列出所有叢集的節點 ID。如需詳細資訊，請參閱 [RebootCacheCluster](#)。

更換節點 (Valkey 和 RedisOSS)

Amazon ElastiCache 經常透過無縫套用至執行個體的修補程式和升級來升級機群。不過，我們需要不時重新啟動 ElastiCache 節點，才能將強制性作業系統更新套用至基礎主機。必須進行這些替換才能套用升級，以強化安全、可靠性和操作效能。

您可以選擇在排程的節點替換時間之前，隨時自行管理這些替換。當您自行管理替換時，執行個體會重新啟動節點時收到 OS 更新，而排程的節點替換將會取消。您可能會繼續收到提醒，指出節點即將進行替換。若您已手動減少維護的需求，您可以忽略這些提醒。

Note

Amazon 自動產生的替換快取節點 ElastiCache 可能有不同的 IP 地址。您必須負責檢閱應用程式組態，以確保快取節點與適當的 IP 位址相關聯。

下列清單會識別排程 ElastiCache 其中一個 Valkey 或 Redis OSS 節點以進行取代時，您可以採取的動作。若要加快尋找針對您的狀況所需的資訊，請從以下選單選擇。

- [Do nothing](#) – 讓 Amazon 依排程 ElastiCache 取代節點。
- [Change your maintenance window](#) - 將您的維護時段變更為更合適的時間。
- Valkey 或 Redis OSS (啟用叢集模式) 組態
 - [Replace the only node in any Valkey or Redis OSS cluster](#) – 使用備份和還原取代 Valkey 或 Redis OSS 叢集中節點的程序。
 - [Replace a replica node in any Valkey or Redis OSS cluster](#) – 透過增加和減少複本計數而無須叢集停機時間，取代任何 Valkey 或 Redis OSS 叢集中僅供讀取複本的程序。
 - [Replace any node in a Valkey or Redis OSS \(cluster mode enabled\) shard](#) – 動態程序，沒有叢集停機時間，可透過擴展和擴展來取代 Valkey 或 Redis OSS (啟用叢集模式) 叢集中的節點。
- Valkey 或 Redis OSS (停用叢集模式) 組態
 - [Replace the only node in any Valkey or Redis OSS cluster](#) – 使用備份和還原取代 Valkey 或 Redis OSS 叢集中任何節點的程序。
 - [Replace a replica node in any Valkey or Redis OSS cluster](#) – 透過增加和減少複本計數而無須叢集停機時間，取代任何 Valkey 或 Redis OSS 叢集中僅供讀取複本的程序。
 - [Replace a node in a Valkey or Redis OSS \(cluster mode disabled\) cluster](#) – 使用複寫取代 Valkey 或 Redis OSS (停用叢集模式) 叢集中的節點的程序。

- [Replace a Valkey or Redis OSS \(cluster mode disabled\) read-replica](#) – 在 Valkey 或 Redis OSS (停用叢集模式) 複寫群組中手動取代僅供讀取複本的程序。
- [Replace a Valkey or Redis OSS \(cluster mode disabled\) primary node](#) – 手動取代 Valkey 或 Redis OSS (停用叢集模式) 複寫群組中主要節點的程序。

Valkey 和 Redis OSS節點替代選項

- 什麼都不做 – 如果您不做，會如期 ElastiCache 取代節點。

對於已啟用自動容錯移轉的非叢集組態，Valkey 7.2 及更高版本上的叢集和 Redis OSS 5.0.6 及更高版本上的叢集會完成取代，同時叢集會繼續保持連線並提供傳入寫入請求。對於 Redis OSS 4.0.10 或更舊版本的自動容錯移轉啟用叢集，您可能會注意到與DNS更新相關聯的短暫寫入中斷長達幾秒。

如果節點是啟用自動容錯移轉叢集的成員，則 ElastiCache 使用 Valkey 或 Redis OSS可在修補、更新和其他維護相關節點替換期間提供更好的可用性。

對於 ElastiCache，若將 Valkey 或 Redis OSS叢集組態設定為ElastiCache 與 Valkey 或 Redis OSS叢集用戶端搭配使用，替換現在會完成，而叢集會提供傳入寫入請求。

對於已啟用自動容錯移轉的非叢集組態，Valkey 7.2 及更高版本上的叢集和 Redis OSS 5.0.6 及更高版本上的叢集會完成取代，同時叢集會繼續保持連線並提供傳入寫入請求。對於 Redis OSS 4.0.10 或更舊版本的自動容錯移轉啟用叢集，您可能會注意到與DNS更新相關聯的短暫寫入中斷長達幾秒。

如果節點是獨立的，Amazon 會 ElastiCache 先啟動替換節點，然後從現有節點同步。現有節點此時將無法處理服務請求。同步完成後，現有節點會終止，新節點會取代。ElastiCache 會盡最大努力在此操作期間保留您的資料。

- 變更您的維護時段 – 針對排定的維護事件，您會收到來自的電子郵件或通知事件 ElastiCache。在這種情況下，如果在排定的替換時間之前變更維護時段，則現在將在新的時間替換您的節點。如需詳細資訊，請參閱下列內容：
 - [修改 ElastiCache 叢集](#)
 - [修改複寫群組](#)

Note

只有在 ElastiCache 通知包含維護時段時，才能透過移動維護時段來變更取代時段。若通知並未包含維護時段，您便無法變更替換時間。

例如，假設現在是 11 月 9 日星期四下午 3:00，下一個維護時段是 11 月 10 日星期五下午 5:00。以下是三種情況及其結果：

- 您將維護時段變更為星期五下午 4:00，在目前的日期時間之後、下一個排定的維護時段之前。節點將於 11 月 10 日星期五下午 4:00 進行替換。
- 您將維護時段變更為星期六下午 4:00，在目前的日期時間之後，以及下一個排定的維護時段之後。節點將於 11 月 11 日星期六下午 4:00 進行替換。
- 您將維護時段變更為星期三下午 4:00，早於這星期中目前的日期時間。節點將於 11 月 15 日星期三下午 4:00 進行替換。

如需說明，請參閱 [管理 ElastiCache 叢集維護](#)。

- 取代任何 Valkey 或 Redis OSS 叢集中唯一的節點 – 如果叢集沒有任何僅供讀取複本，您可以使用下列程序來取代節點。

使用備份與還原替換唯一節點

1. 建立節點叢集的快照。如需說明，請參閱「[取得手動備份](#)」。
2. 從快照建立新叢集。如需說明，請參閱 [從備份還原到新的快取](#)。
3. 刪除包含已排程替換節點的叢集。如需說明，請參閱 [在中刪除叢集 ElastiCache](#)。
4. 在您的應用程式中，使用新節點的端點替換舊節點的端點。

- 取代任何 Valkey 或 Redis OSS 叢集中的複本節點 – 若要取代複本叢集，請增加複本計數。若要達成此目的，請先新增複本，然後移除您要替換的複本來減少複本計數。此程序為動態程序，因此不會有任何叢集停機時間。

Note

若您的碎片或複寫群組已有 5 個複本，請對調步驟 1 和 2 的順序。

若要取代任何 Valkey 或 Redis OSS 叢集中的複本

1. 透過將複本新增到碎片或複寫群組，來增加複本數。如需詳細資訊，請參閱[增加碎片中的複本數量](#)。
 2. 刪除您欲替換的複本。如需詳細資訊，請參閱[減少碎片中的複本數量](#)。
 3. 更新您應用程式中的端點。
- 取代 Valkey 或 Redis OSS (啟用叢集模式) 碎片中的任何節點 – 若要在沒有停機時間的情況下取代叢集中的節點，請使用線上重新編譯。首先透過向外擴展來新增碎片，然後透過向內擴展來刪除具有要替換之節點的碎片。

若要取代 Valkey 或 Redis OSS (啟用叢集模式) 叢集中的任何節點

1. 向外擴展：使用與包含要替換節點現有碎片相同的組態新增額外的碎片。如需詳細資訊，請參閱[使用線上重新分片功能新增碎片](#)。
 2. 向內擴展：刪除包含要替換節點的碎片。如需詳細資訊，請參閱[使用線上重新分片移除碎片](#)。
 3. 更新您應用程式中的端點。
- 取代 Valkey 或 Redis OSS (停用叢集模式) 叢集中的節點 – 如果叢集是沒有任何僅供讀取複本的 Valkey 或 Redis OSS (停用叢集模式) 叢集，請使用下列程序來取代節點。

使用複寫替換節點 (僅限停用叢集模式)

1. 將複寫新增到包含已排程替換節點的叢集，做為主要節點。請不要在此叢集上啟用異地同步備份。如需說明，請參閱 [若要將複寫新增至沒有碎片的 Valkey 或 Redis OSS 叢集](#)。

2. 將僅供讀取複本新增到叢集。如需說明，請參閱 [將節點新增至 ElastiCache 叢集 \(主控台\)](#)。
 3. 將新建立的僅供讀取複本提升為主要節點。如需說明，請參閱 [針對 Valkey 或 Redis OSS \(停用叢集模式\) 複寫群組，將僅供讀取複本提升為主要複本](#)。
 4. 刪除已排程替換的節點。如需說明，請參閱 [從 ElastiCache 叢集移除節點](#)。
 5. 在您的應用程式中，使用新節點的端點替換舊節點的端點。
- 取代 Valkey 或 Redis OSS (停用叢集模式) 僅供讀取複本 – 如果節點是僅供讀取複本，請取代節點。

若您的叢集只有一個複本節點，並且已啟用異地同步備份，您必須先停用異地同步備份才能刪除複本。如需說明，請參閱 [修改複寫群組](#)。

若要取代 Valkey 或 Redis OSS (停用叢集模式) 僅供讀取複本

1. 刪除已排程進行替換的複本。如需詳細說明，請參閱下列主題：
 - [減少碎片中的複本數量](#)
 - [從 ElastiCache 叢集移除節點](#)
 2. 新增新複本來替換已排程進行替換的複本。若您使用與剛刪除複本相同的名稱，您可以跳過步驟 3。如需詳細說明，請參閱下列主題：
 - [增加碎片中的複本數量](#)
 - [新增 Valkey 或 Redis 的僅供讀取複本 OSS \(停用叢集模式\)](#)
 3. 在您的應用程式中，使用新複本的端點替換舊複本的端點。
 4. 若您在一開始停用異地同步備份，現在請重新啟用。如需說明，請參閱 [啟用多個可用區](#)。
- 取代 Valkey 或 Redis OSS (停用叢集模式) 主要節點 – 如果節點是主要節點，請先將僅供讀取複本提升為主要節點。然後刪除先前做為主要節點的複本。

若您的叢集只有一個複本，並且已啟用異地同步備份，您必須先停用異地同步備份才能在步驟 2 刪除複本。如需說明，請參閱 [修改複寫群組](#)。

若要取代 Valkey 或 Redis OSS (停用叢集模式) 主節點

1. 將僅供讀取複本提升為主要節點。如需說明，請參閱 [針對 Valkey 或 Redis OSS \(停用叢集模式 \) 複寫群組，將僅供讀取複本提升為主要複本](#)。
2. 刪除已排程進行替換的節點 (舊的主要節點)。如需說明，請參閱 [從 ElastiCache 叢集移除節點](#)。
3. 新增新複本來替換已排程進行替換的複本。若您使用與剛刪除節點相同的名稱，您可以跳過變更您應用程式中的端點。

如需說明，請參閱 [新增 Valkey 或 Redis 的僅供讀取複本 OSS \(停用叢集模式 \)](#)。

4. 在您的應用程式中，使用新節點的端點替換舊節點的端點。
5. 若您在一開始停用異地同步備份，現在請重新啟用。如需說明，請參閱「[啟用多個可用區](#)」。

更換節點 (Memcached)

Amazon ElastiCache (Memcached) 經常會升級機群，並無縫地將修補程式和升級套用至執行個體。不過，我們需要不時重新啟動您的 ElastiCache (Memcached) 節點，才能將強制性作業系統更新套用至基礎主機。必須進行這些替換才能套用升級，以強化安全、可靠性和操作效能。

您可以選擇在排程的節點替換時間之前，隨時自行管理這些替換。當您自行管理替換時，執行個體會重新啟動節點時收到 OS 更新，而排程的節點替換將會取消。您可能會繼續收到提醒，指出節點即將進行替換。若您已手動減少維護的需求，您可以忽略這些提醒。

Note

Amazon 自動產生的替換快取節點 ElastiCache 可能有不同的 IP 地址。您必須負責檢閱應用程式組態，以確保快取節點與適當的 IP 位址相關聯。

下列清單會識別排程 ElastiCache 其中一個 Memcached 節點以進行取代時，您可以採取的動作。

- 什麼都不做 – 如果您不做，ElastiCache 會依排程取代節點。當 ElastiCache 自動將節點取代為新節點時，新節點一開始會是空的。
- 變更您的維護時段 – 針對排定的維護事件，您會收到來自的電子郵件或通知事件 ElastiCache。在這種情況下，如果在排定的替換時間之前變更維護時段，則現在將在新的時間替換您的節點。如需詳細資訊，請參閱 [修改 ElastiCache 叢集](#)。

Note

只有在 ElastiCache 通知包含維護時段時，才能透過移動維護時段來變更替換時段。若通知並未包含維護時段，您便無法變更替換時間。

例如，假設現在是 11 月 9 日星期四下午 3:00，下一個維護時段是 11 月 10 日星期五下午 5:00。以下是三種情況及其結果：

- 您將維護時段變更為星期五下午 4:00，在目前的日期時間之後、下一個排定的維護時段之前。節點將於 11 月 10 日星期五下午 4:00 進行替換。
- 您將維護時段變更為星期六下午 4:00，在目前的日期時間之後，以及下一個排定的維護時段之後。節點將於 11 月 11 日星期六下午 4:00 進行替換。
- 您將維護時段變更為星期三下午 4:00，早於這星期中目前的日期時間。節點將於 11 月 15 日星期三下午 4:00 進行替換。

如需說明，請參閱 [管理 ElastiCache 叢集維護](#)。

- 手動替換節點 - 若您需要在下一個維護時段之前替換節點，請手動替換節點。

如果手動替換節點，則會重新配送金鑰。此重新配送會導致快取遺漏。

手動替換 Memcached 節點

1. 刪除已排程替換的節點。如需說明，請參閱 [從 ElastiCache 叢集移除節點](#)。
2. 將新的節點新增到叢集。如需說明，請參閱 [將節點新增至 ElastiCache 叢集](#)。
3. 若您並未在此叢集上使用自動探索，請查看您的應用程式並使用新節點的端點替換舊節點端點的每個執行個體。

預留節點

保留一或多個 ElastiCache 節點可能是降低成本的一種方式。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度 (一或三年)。

若要了解預留節點在您的使用案例中是否可以節省成本，請先判斷您需要的節點大小和節點數量。然後估計節點的用量，並比較使用隨需節點和預留節點時的總成本。您可以在您的叢集中混合搭配使用預留節點與隨需節點。如需定價資訊，請參閱 [Amazon ElastiCache Pricing](#)。

主題

- [管理使用預留節點的成本](#)
- [標準預留節點方案](#)
- [大小彈性預留節點](#)
- [刪除預留節點](#)
- [舊式預留節點方案](#)
- [取得關於預留節點方案的資訊](#)
- [購買預留節點](#)
- [取得預留節點的相關資訊](#)

管理使用預留節點的成本

預留一或多個節點可能可以減少您的成本。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度 (一或三年)。此費用遠低於您使用隨需節點時需支付的每小時使用費。

若要了解預留節點在您的使用案例中是否可以節省成本，請先判斷您需要的節點大小和節點數量。然後估計節點的用量，並比較使用隨需節點和預留節點時的總成本。您可以在您的叢集中混合搭配使用預留節點與隨需節點。如需定價資訊，請參閱 [Amazon ElastiCache Pricing](#)。

AWS 區域、節點類型和期限長度必須在購買時選擇，之後無法變更。

您可以使用 AWS Management Console、AWS CLI 或 ElastiCache API 來列出和購買可用的預留節點方案。

如需預留節點的詳細資訊，請參閱 [Amazon ElastiCache Reserved Nodes](#)。

標準預留節點方案

當您在 Amazon 中購買預留節點執行個體（RI）時 ElastiCache，您可以購買承諾，在特定節點執行個體類型和 AWS 區域取得預留節點執行個體持續時間的折扣費率。若要使用 Amazon ElastiCache 預留節點執行個體，您可以建立新的 ElastiCache 節點執行個體，就像隨需執行個體一樣。

如果新預留節點執行個體的規格與帳戶的現有預留節點執行個體相符，則會以預留節點執行個體提供的折扣費率向您收費。否則，節點執行個體將按隨需費率計費。這些標準可從 R5 和 M5 執行個體系列開始RIs取得。

Note

接下來討論的所有方案類型都提供一年期和三年期。

方案類型

無前期 RI 提供對預留 ElastiCache 執行個體的存取，而不需要預先付款。您的無預付預留 ElastiCache 執行個體會針對期間內每小時收取折扣的每小時費率，無論用量為何。

部分預付 RI 需要預先支付預留 ElastiCache 執行個體的一部分。期間內其餘的時數會以折扣後的每小時費率計費，無論是否有使用。此選項為傳統重度使用選項的替代方案，下一節將說明。

全部預付 RI 要求在 RI 有效期間開始時繳付全額款項。有效期間開始時便支付全額款項，且有效期間的剩餘時間內，無論使用多少小時，都不會產生其他成本或額外的每小時費用。

大小彈性預留節點

所有預留節點都具有大小彈性。當您購買預留節點時，您指定的一件事是節點類型，例如 `cache.r6g.xlarge`。如需節點類型的詳細資訊，請參閱 [Amazon ElastiCache Pricing](#)。

如果您有節點，而且需要將其擴展到更大的容量，則保留的節點會自動套用至擴展的節點。也就是說，您預留的節點會自動套用至相同節點系列中任何大小的使用。大小彈性預留節點適用於具有相同 AWS 區域的節點。大小彈性預留節點只能在其節點系列中擴展。例如，Cache.r6g.xlarge 的預留節點可以套用至 cache.r6g.2xlarge，但不適用於 cache.r6gd.large，因為 cache.r6g 和 cache.r6gd 是不同的節點系列。

大小彈性意味著您可以在相同節點系列中的組態之間自由移動。例如，您可以從 r6g.xlarge 預留節點（8 個標準化單位）移至相同 AWS 區域中的兩個 r6g.large 預留節點（8 個標準化單位）（ $2 \times 4 = 8$ 個標準化單位），無需額外費用。

將預留節點從 Redis 升級至 OSS Valkey

在中啟動 Valkey 後 ElastiCache，您現在可以將 Redis OSS 預留節點折扣套用至 Valkey 快取引擎。您可以從 Redis OSS 升級至 Valkey，同時仍可受益於現有的合約和保留。除了能夠在快取節點系列和引擎中套用您的利益之外，您甚至可以接收更多增量值。Valkey 價格比 Redis 高出 20% 的折扣 OSS，而且有預留節點彈性，您可以使用 Redis OSS 預留節點來涵蓋執行中的 Valkey 節點多出 20%。

若要計算折扣率，每個 ElastiCache 節點和引擎組合都有以單位測量的標準化係數。預留節點單位可以套用至預留節點執行個體系列中特定引擎的任何執行中節點。Redis OSS 保留節點可以另外套用於引擎，以涵蓋執行中的 Valkey 節點。由於 Valkey 的價格比 Redis OSS 和 Memcached 低，因此其特定執行個體類型的單位較低，這允許 Redis OSS 預留節點涵蓋更多 Valkey 節點。

例如，假設您已為 Redis OSS 引擎（32 個單位）購買 cache.r7g.4xlarge 預留節點，並正在執行一個 cache.r7g.4xlarge Redis OSS 節點（32 個單位）。如果您將節點升級至 Valkey，則執行中節點的標準化係數會下降至 25.6 個單位，而您現有的預留節點會為您提供額外的 6.4 個單位，以用於區域內 cache.r7g 系列中任何其他執行中的 Valkey 或 Redis OSS 節點。您可以使用此功能來涵蓋帳戶中另一個 cache.r7g.4xlarge Valkey 節點（25.6 個單位）的 25%，或 cache.r7g.xlarge Valkey 節點（6.4 個單位）的 100%。

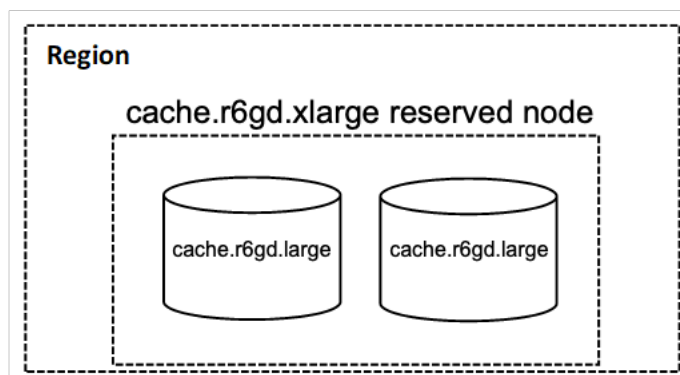
使用標準化單位比較用量

您可以使用標準化單位來比較不同預留節點大小的用量。例如，兩個 cache.r6g.4xlarge 節點上的一小時用量相當於一個 cache.r6g.large 上的 16 小時用量。下表顯示每個節點大小的標準化單位數量：

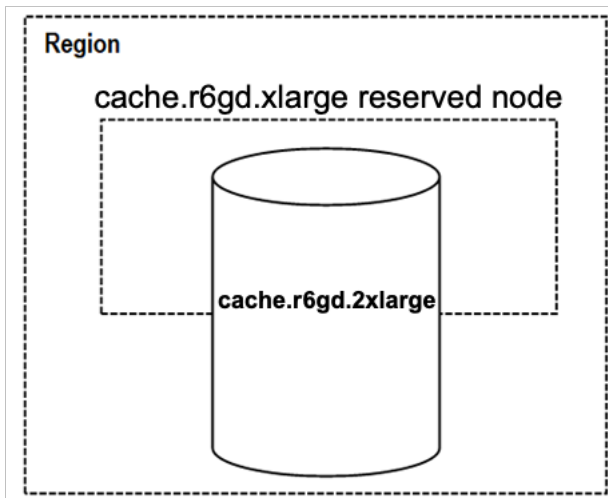
節點大小	標準化單位 (Redis OSS、Memcached)	標準化單位 (Valkey)
微型	0.5	0.4

節點大小	標準化單位 (Redis OSS、Memcached)	標準化單位 (Valkey)
小型	1	8.
中型	2	1.6
大型	4	3.2
xlarge	8	6.4
2xlarge	16	12.8
4xlarge	32	25.6
6xlarge	48	38.4
8xlarge	64	51.2
10xlarge	80	64
12xlarge	96	76.8
16xlarge	128	102.4
24xlarge	192	153.6

例如，您購買 `cache.r6gd.xlarge` 預留節點，且在同一 AWS 區域中的帳戶中有兩個執行中 `cache.r6gd.large` 預留節點。在此情況下，帳單利益會完全套用至兩個節點。



或者，如果您在同一 AWS 區域中的帳戶中有一個 `cache.r6gd.2xlarge` 執行個體正在執行，則帳單利益會套用至預留節點的 50% 用量。



刪除預留節點

預留節點的條款涉及一年或三年的承諾。您無法取消預留節點。不過，您可以刪除保留節點折扣涵蓋的節點。刪除預留節點折扣涵蓋的節點的程序與任何其他節點的程序相同。

如果您刪除預留節點折扣涵蓋的節點，則可以啟動另一個具有相容規格的節點。在此情況下，您仍可以在保留時間（一或三年）內繼續享有折扣費率。

舊式預留節點方案

有三個層級的舊式節點保留：重度使用率、中度使用率和輕度使用率。節點可以在任何使用率層級保留 1 或 3 年。節點類型、使用率層級和保留有效期間會影響您的總成本。購買預留節點之前，請比較各種模型，確認預留節點可為您的業務省下的費用。

在某個使用率層級或期間購買的節點不能轉換為不同的使用率層級或期間。

使用率層級

重度使用率預留節點允許具有一致容量基線的工作負載或執行穩定狀態的工作負載。重度使用率預留節點需要高度的前期承諾，但如果您計劃執行多於 79% 的預留節點期間，您可以獲得最高的節省金額（節省金額最高達隨需價格的 70%）。若採用高度使用率預留節點，您需支付一筆一次性費用。之後無論您的節點是否有執行，再有效期間內都需支付較低的每小時費用。

如果您計劃運用預留節點很長一段時間，且您需要較低的一次性費用，或希望可在節點關閉時停止支付其費用，則中度使用率預留節點為最佳選項。當您計劃執行超過 40% 的預留節點期間，則中度使用率預留節點為更符合成本效益的選項。此選項可以為您節省高達隨需價格的 64%。利用中度使用率預留

節點，您需支付較輕度使用率預留節點稍高的一次性費用，並且會在執行節點時獲得較低的每小時使用費率。

輕度使用率預留節點為一天只會執行幾個小時，或一週執行幾天的定期性工作負載的理想選擇。使用輕度使用率預留節點，您會支付一次性費用，接著是您的節點執行時每小時折扣的使用費。當您的節點執行時間超過預留節點有效期間的 17%，您就可以開始節省成本。預留節點的整個有效期間內，您可以節高達隨需費率的 56%。

舊式預留節點方案

產品	預付成本	使用費	優點
高度使用率	最高	每小時費用最低。無論您是否使用預留節點，會套用至整個期間。	如果您計劃執行預留節點超過三年期間的 79%，則整體成本最低。
中度使用率	中	對執行節點的每個小時收取每小時使用費。節點未執行時不收取每小時費用。	適合彈性工作負載或當您預期的是中度使用量 (超過三年期間的 40%)。
低度使用率	最低	對執行節點的每個小時收取每小時使用費。節點未執行時不收取每小時費用。所有方案類型的每小時費用最高，但費用僅套用在預留節點執行時。	如果您規劃不間斷執行，則整體成本會最高。不過，如果您規劃的預留節點使用頻率較低 (三年有效期間大約超過 15%)，則整體成本最低。
隨需使用 (無預留節點)	無	每小時費用最高。每當節點執行時便套用。	每小時成本最高。

如需詳細資訊，請參閱 [Amazon ElastiCache Pricing](#)。

取得關於預留節點方案的資訊

購買預留節點之前，您可以取得關於可用的預留節點方案的資訊。

下列範例示範如何使用 AWS Management Console AWS CLI、和 取得可用預留節點方案的定價和資訊 ElastiCache API。

主題

- [取得關於預留節點方案的資訊 \(主控台\)](#)
- [取得關於預留節點方案的資訊 \(AWS CLI\)](#)
- [取得預留節點方案的相關資訊 \(ElastiCache API\)](#)

取得關於預留節點方案的資訊 (主控台)

若要使用 取得可用預留叢集方案的定價和其他資訊 AWS Management Console，請使用下列程序。

取得可用預留節點方案的相關資訊

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留節點)。
3. 選擇 Purchase Reserved Nodes (購買預留節點)。
4. 針對引擎，選擇 Valkey、Redis OSS 或 Memcached。
5. 若要判斷可用的方案，請為接下來的選項進行選擇：
 - 節點類型
 - 期間
 - 方案類型

進行這些選擇之後，Reservation details (預留詳細資訊) 中會顯示每一節點的成本和您選擇的總成本。

6. 選擇 Cancel (取消) 來避免購買這些節點和衍生費用。

取得關於預留節點方案的資訊 (AWS CLI)

若要取得 Valkey 或 Redis 可用預留節點方案的定價和其他資訊 OSS，請在命令提示中輸入下列命令：

```
aws elasticache describe-reserved-cache-nodes-offerings
```

此操作會產生類似下列（JSON 格式）的輸出：

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 31536000,
  "FixedPrice": X.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "No Upfront",
  "RecurringCharges": [
```

```

        {
            "RecurringChargeAmount": X.XXX,
            "RecurringChargeFrequency": "Hourly"
        }
    ]
}

```

若要取得 Memcached 可用預留節點方案的定價和其他資訊，請在命令提示中輸入下列命令：

```

{
    "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
    "CacheNodeType": "cache.xxx.large",
    "Duration": 94608000,
    "FixedPrice": XXXX.X,
    "UsagePrice": X.X,
    "ProductDescription": "memcached",
    "OfferingType": "All Upfront",
    "RecurringCharges": [
        {
            "RecurringChargeAmount": X.X,
            "RecurringChargeFrequency": "Hourly"
        }
    ]
},
{
    "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
    "CacheNodeType": "cache.xxx.xlarge",
    "Duration": 94608000,
    "FixedPrice": XXXX.X,
    "UsagePrice": X.X,
    "ProductDescription": "memcached",
    "OfferingType": "Partial Upfront",
    "RecurringCharges": [
        {
            "RecurringChargeAmount": X.XXXX,
            "RecurringChargeFrequency": "Hourly"
        }
    ]
},
{
    "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
    "CacheNodeType": "cache.xx.12xlarge",
    "Duration": 31536000,

```



```
    "FixedPrice": X.X,  
    "UsagePrice": X.X,  
    "ProductDescription": "memcached",  
    "OfferingType": "No Upfront",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": X.XXXX,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ]  
  }  
}
```

如需詳細資訊，請參閱 參考中的 AWS CLI [describe-reserved-cache-nodes-offerings](#)。

取得預留節點方案的相關資訊 (ElastiCache API)

若要取得關於可用的預留節點方案的定價和資訊，請呼叫 DescribeReservedCacheNodesOfferings 動作。

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodesOfferings  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 參考 [DescribeReservedCacheNodesOfferings](#) 中的 ElastiCache API。

購買預留節點

下列範例示範如何使用 AWS Management Console、AWS CLI 和 購買預留節點方案 ElastiCache API。

Important

遵循本節的範例，您的帳戶會產生 AWS 您無法撤銷的費用。

主題

- [購買預留節點 \(主控台\)](#)
- [購買預留節點 \(AWS CLI\)](#)
- [購買預留節點 \(ElastiCache API\)](#)

購買預留節點 (主控台)

此範例顯示購買特定的預留節點方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留節點 ID 為 myreservationID。

下列程序使用 AWS Management Console 來透過提供 ID 購買預留節點方案。

購買預留節點

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽清單中，選擇 Reserved Nodes (預留節點) 連結。
3. 選擇 Purchase Reserved Node (購買預留節點) 按鈕。
4. 針對引擎，選擇 Valkey、Redis OSS 或 Memcached。
5. 若要判斷可用的方案，請為接下來的選項進行選擇：
 - 節點類型
 - 期間
 - 方案類型
 - 選擇性的 Reserved node ID (預留節點識別碼)

進行這些選擇之後，Reservation details (預留詳細資訊) 中會顯示每一節點的成本和您選擇的總成本。

6. 選擇 Purchase (購買)。

購買預留節點 (AWS CLI)

下列範例顯示購買特定的預留叢集方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留節點 ID 為 myreservationID。

在命令提示字元中，輸入下列命令：

若為 Linux、macOS 或 Unix：

```
aws elasticache purchase-reserved-cache-nodes-offering \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \  
  --reserved-cache-node-id myreservationID
```

針對 Windows：

```
aws elasticache purchase-reserved-cache-nodes-offering ^  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^  
  --reserved-cache-node-id myreservationID
```

此命令會傳回類似以下的輸出：

RESERVATION	ReservationId	Class	Start Time	Duration	
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	myreservationid	cache.xx.small	2013-12-19T00:30:23.247Z	1y	
XXX.XX USD	X.XXX USD	1	payment-pending	memcached	Medium Utilization

如需詳細資訊，請參閱 參考中的 AWS CLI [purchase-reserved-cache-nodes-提供](#)。

購買預留節點 (ElastiCache API)

下列範例顯示購買特定的預留節點方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留叢集 ID 為 myreservationID。

搭配下列參數呼叫 PurchaseReservedCacheNodesOffering 操作：

- ReservedCacheNodesOfferingId = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- ReservedCacheNodeID = myreservationID
- CacheNodeCount = 1

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 參考 [PurchaseReservedCacheNodesOffering](#) 中的 ElastiCache API 。

取得預留節點的相關資訊

您可以使用 AWS Management Console、AWS CLI 和 取得所購買預留節點的相關資訊 ElastiCache API。

主題

- [取得預留節點的相關資訊 \(主控台\)](#)
- [取得預留節點的相關資訊 \(AWS CLI\)](#)
- [取得預留節點的相關資訊 \(ElastiCache API\)](#)

取得預留節點的相關資訊 (主控台)

下列程序說明如何使用 AWS Management Console 來取得您購買的預留節點的相關資訊。

取得您購買之預留節點的相關資訊

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽清單中，選擇 Reserved Nodes (預留節點) 連結。

帳戶的預留節點會出現在預留節點清單中。選擇清單中的任何預留節點，便能從主控台底部的詳細資料窗格中查看該預留節點的詳細資訊。

取得預留節點的相關資訊 (AWS CLI)

若要取得您 AWS 帳戶預留節點的相關資訊，請在命令提示中輸入下列命令：

```
aws elasticache describe-reserved-cache-nodes
```

此操作會產生類似下列 (JSON 格式) 的輸出：

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
  "CacheNodeType": "cache.xx.small",
  "DataTiering": "disabled",
  "Duration": "31536000",
  "ProductDescription": "memcached",
  "OfferingType": "Medium Utilization",
```

```
"MaxRecords": 0
}
```

如需詳細資訊，請參閱 AWS CLI 參考中的 [describe--reserved-cache-nodes](#)。

取得預留節點的相關資訊 (ElastiCache API)

若要取得您 AWS 帳戶預留節點的相關資訊，請呼叫 DescribeReservedCacheNodes 操作。

Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReservedCacheNodes
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 參考 [DescribeReservedCacheNodes](#) 中的 ElastiCache API。

遷移上一代節點

上一代節點是逐步遭淘汰的節點類型。如果您沒有使用上一代節點類型的現有叢集，ElastiCache 不支援使用該節點類型建立新叢集。

由於上一代節點類型的數量有限，我們無法保證您叢集中的節點運作狀態不良時可成功替換。在這種情況下，您的叢集可用性可能會受到負面影響。

建議您將叢集遷移至新的節點類型，以獲得更佳的可用性和效能。如需建議的遷移節點類型，請參閱 [升級途徑](#)。如需 中支援節點類型和上一代節點類型的完整清單 ElastiCache，請參閱 [支援的節點類型](#)。

遷移 Valkey 或 Redis OSS 叢集上的節點

下列程序說明如何使用 ElastiCache 主控台遷移您的 Valkey 或 Redis OSS 叢集節點類型。在此過程中，您的 Valkey 或 Redis OSS 叢集將繼續以最短的停機時間提供請求。根據您的叢集組態，您可能看到下列停機時間。以下是估計值，可能會因您的具體組態而有所不同：

- 停用叢集模式（單一節點）DNS 可能會看到大約 60 秒，主要是由於傳播。
- 對於執行 Valkey 7.2 及更高版本或 Redis 5.0.6 及更高版本的叢集，停用叢集模式（具有複本節點）可能會看到大約 OSS 1 秒。所有較低版本都可能經歷約 10 秒。
- 啟用叢集模式可能會經歷大約 1 秒。

若要使用主控台修改 Valkey 或 Redis OSS叢集節點類型：

1. 登入主控台，並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。
3. 從叢集清單中選擇您要遷移的目標叢集。
4. 選擇 Actions (動作)，然後選擇 Modify (修改)。
5. 從節點類型清單中選擇新的節點類型。
6. 如果您要立即執行遷移程序，請選擇 Apply immediately (立即套用)。如果未選擇 Apply immediately (立即套用)，便會在此叢集的下一個維護時段期間執行遷移。
7. 選擇 Modify (修改)。如果您在上一個步驟中選擇 Apply immediately (立即套用)，叢集的狀態變更為 modifying (修改中)。當狀態變更為 available (可用)，修改即已完成，並且您可以開始使用新叢集。

若要使用 修改 Valkey 或 Redis OSS叢集節點類型 AWS CLI：

使用 [modify-replication-group](#) API，如下所示：

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group /
  --replication-group-id my-replication-group /
  --cache-node-type new-node-type /
  --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
  --replication-group-id my-replication-group ^
  --cache-node-type new-node-type ^
  --apply-immediately
```

在此案例中，的值為 *new-node-type* 是您要遷移至的節點類型。若傳遞 `--apply-immediately` 參數，當複寫群組從修改中變為可用狀態時，就會立即套用更新。如果未選擇 Apply immediately (立即套用)，便會在此叢集的下一個維護時段期間執行遷移。

Note

如果無法修改具有 `InvalidCacheClusterState` 錯誤的叢集，您需要先移除還原失敗的節點。

修正或移除 `restore-failed-node` (s)

下列程序說明如何從 Valkey 或 Redis OSS 叢集中修正或移除還原失敗的節點 (`restore-failed node`)。若要進一步了解 ElastiCache 節點如何進入還原失敗狀態 (請參閱)，請參閱 [檢視 ElastiCache 節點狀態](#)。我們建議先移除任何處於還原失敗狀態的節點，然後將 ElastiCache 叢集中剩餘的上一代節點遷移至較新一代節點類型，最後新增所需的節點數量。

移除還原失敗的節點 (主控台)：

1. 登入主控台，並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集。
3. 在叢集清單中，選擇您要從中移除節點的叢集。
4. 在節點清單中，選擇您要從中移除節點的碎片。如果叢集已停用叢集模式，請略過此步驟。
5. 在節點清單中，選擇狀態為 `restore-failed` 的節點。
6. 選擇 Actions (動作)，然後選擇 Delete node (刪除節點)。

從 ElastiCache 叢集移除還原失敗的節點後，您現在可以遷移至較新一代的 (含) 類型。如需詳細資訊，請參閱上方的 [遷移 Valkey 或 Redis OSS 叢集上的節點](#)。

若要將返回節點新增至 ElastiCache 叢集，請參閱 [將節點新增至 ElastiCache 叢集](#)。

遷移 Memcached 叢集上的節點

若要將 ElastiCache (Memcached) 遷移至不同的節點類型，您必須建立新的叢集，該叢集一律會開始為空，您的應用程式可以填入。

若要使用主控台遷移 ElastiCache (Memcached) ElastiCache叢集節點類型：

- 使用新的節點類型來建立新叢集。如需詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。
- 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱「[尋找叢集的端點 \(主控台\) \(Memcached \)](#)」
- 刪除舊叢集。如需詳細資訊，請參閱 [在中刪除叢集 ElastiCache](#)

在中管理叢集 ElastiCache

叢集是一或多個快取節點的集合，所有這些節點都會執行 Valkey、Redis OSS或 Memcached 引擎軟體的執行個體。在您建立叢集時，可以指定要使用的所有節點之引擎和版本。

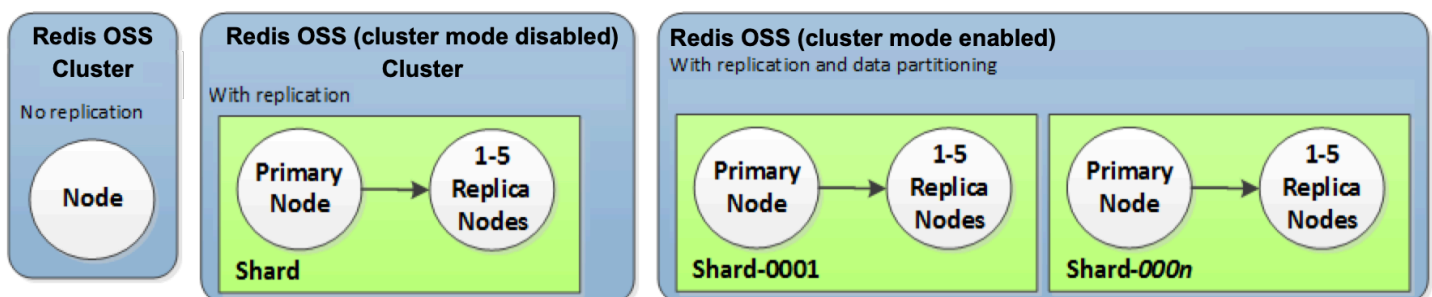
Valkey 和 Redis OSS叢集

下圖說明典型的 Valkey 或 Redis OSS叢集。這些叢集可以在碎片 (API/CLI : 節點群組) 內包含單一節點或最多六個節點，單一節點 Valkey 或 Redis OSS (停用叢集模式) 叢集沒有碎片，而多節點 Valkey 或 Redis OSS (停用叢集模式) 叢集具有單一碎片。Valkey 或 Redis OSS (啟用叢集模式) 叢集最多可以有 500 個碎片，您的資料會分割在碎片上。如果引擎版本為 Valkey 7.2 及更高版本，或 Redis 5.0.6 及更高版本，則節點或碎片限制可以增加到每個叢集最多 OSS 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 (每個碎片一個主要版本和 5 個複本) 到 500 個 (單一主要版本並且沒有複本) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路CIDR太小，範圍太小，或者子網路被其他叢集共用和大量使用。如需詳細資訊，請參閱[建立子網路群組](#)。對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

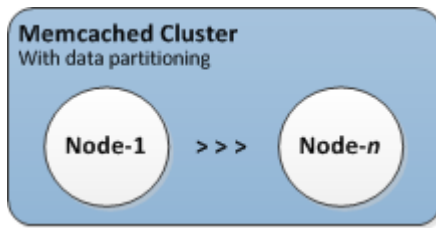
當您在 Valkey 或 Redis OSS 碎片中有多個節點時，其中一個節點是讀取/寫入主要節點。碎片中的所有其他節點則是唯讀複本。

典型的 Valkey 或 Redis OSS叢集如下所示。



Memcached 叢集

典型的 Memcached 叢集如下所示。Memcached 叢集包含 1 到 60 個節點，您可以在其中水平分割資料。



Valkey、Redis OSS和 Memcached 的 ElastiCache 操作

大多數 ElastiCache 操作都會在叢集層級執行。您可以將叢集設定為含特定數量的節點和一個參數群組，以控制每個節點的屬性。叢集內的所有節點都設計為相同節點類型，並具備相同的參數和安全群組設定。

每個叢集都必須有一個叢集識別符。叢集識別符是客戶針對叢集提供的名稱。此識別符會指定與 ElastiCache API和 AWS CLI 命令互動時的特定叢集。叢集識別符對於 AWS 區域中的客戶必須是唯一的。

ElastiCache 支援多個引擎版本。除非您有特定原因，否則我們建議您使用最新版本。

ElastiCache 叢集的設計是使用 Amazon EC2執行個體存取。如果您根據 Amazon VPC服務在虛擬私有雲端（VPC）中啟動叢集，您可以從外部存取叢集 AWS。如需詳細資訊，請參閱[從外部存取 ElastiCache 資源 AWS](#)。

如需支援的版本清單，請參閱 [支援的引擎和版本](#)、[支援的 Redis OSS版本](#)和 [支援的 ElastiCache \(Memcached \) 版本](#)。

在中選擇網路類型 ElastiCache

ElastiCache 支援網際網路通訊協定第 4 版和第 6 版 (IPv4 和 IPv6) ，可讓您將叢集設定為接受：

- 僅IPv4連線
- 僅IPv6連線
- IPv4 和 IPv6 連線 (雙堆疊)

IPv6 支援使用 Valkey 7.2 和更新版本或 Redis OSS引擎 6.2 版和更新版本在 [Nitro 系統](#) 上建置的所有執行個體上的工作負載。透過 ElastiCache 存取 無需額外費用IPv6。

Note

不支援在 / IPV6 雙堆疊可用之前建立的叢集遷移。也不支援在新建立的叢集上切換網路類型。

IPv6 支援在 [Nitro 系統](#) 上建置的所有執行個體上使用 Memcached 引擎 1.6.6 版更新的工作負載。透過 ElastiCache 存取 無需額外費用IPv6。

設定網路類型的子網路

如果您在 Amazon 中建立叢集VPC，則必須指定子網路群組。ElastiCache 使用該子網路群組來選擇該子網路內的子網路和 IP 地址，以與節點建立關聯。ElastiCache 叢集需要同時具有的雙堆疊子網路 IPv4和指派給它們的IPv6位址，才能在雙堆疊模式下操作，而IPv6只有 的子網路只能作為 IPv6操作。

使用雙堆疊

在啟用叢集模式下使用 ElastiCache (Redis OSS) 時，從應用程式的角度來看，透過組態端點連線至所有叢集節點與直接連線至個別快取節點並無不同。若要達成此目標，叢集感知用戶端必須參與叢集探索程序，並要求所有節點的組態資訊。Rediss 的探索通訊協定在每個節點僅支援一個 IP。

當您使用 ElastiCache (Memcached) 建立快取叢集並選擇雙堆疊作為網路類型時，您需要指定 IP 探索類型 – IPv4或 IPv6。ElastiCache 將預設網路類型和 IP 探索為 IPv6，但可以變更。如果您使用自動探索，只有您所選 IP 類型的 IP 地址會傳回至 Memcached 用戶端。如需詳細資訊，請參閱[自動識別叢集中的節點 \(Memcached \)](#)。

為了維持與所有現有用戶端的向後相容性，會引入 IP 探索，可讓您選取要在探索通訊協定中公告的 IP 類型 (即 IPv4或 IPv6)。雖然這將自動探索限制為僅一種 IP 類型，但雙堆疊仍然有利於啟用叢集模

式的工作負載，因為它可以讓在沒有停機時間的情況下從遷移（或復原）IPv4到IPv6 Discovery IP 類型。

TLS 已啟用雙堆疊 ElastiCache 叢集

ElastiCache 叢集TLS啟用時，叢集探索函數，例如 `cluster slots`、`cluster shards`和 `cluster nodes` 搭配 Valkey 或 Redis OSS以及 `config get cluster`搭配 Memcached 傳回主機名稱，而非 IPs。然後，會使用主機名稱來取代 IPs 連接至 ElastiCache 叢集，並執行交TLS握。這表示用戶端不會受到 IP 探索參數的影響。對於TLS已啟用的叢集，IP Discovery 參數不會影響偏好的 IP 通訊協定。相反地，使用的 IP 通訊協定將取決於用戶端在解析DNS主機名稱時偏好的 IP 通訊協定。

如需解析DNS主機名稱時如何設定 IP 通訊協定偏好設定的範例，請參閱 [TLS 已啟用雙堆疊 ElastiCache 叢集](#)。

使用 AWS Management Console（Valkey 和 RedisOSS）

使用 建立叢集時 AWS Management Console，請在連線下選擇網路類型 IPv4IPv6或雙堆疊。如果您要建立 Valkey 或 Redis OSS（啟用叢集模式）叢集並選擇雙堆疊，則必須選取探索 IP 類型 IPv6或 IPv4。

如需詳細資訊，請參閱 [建立 Valkey 或 Redis OSS（啟用叢集模式）叢集（主控台）](#) 或 [建立 Valkey 或 Redis OSS（停用叢集模式）（主控台）](#)。

使用 建立複寫群組時 AWS Management Console，請選擇網路類型 IPv4IPv6或雙堆疊。如果您選擇雙堆疊，則必須選擇 Discovery IP 類型 或 IPv6 IPv4。

如需詳細資訊，請參閱 [從頭開始建立 Valkey 或 Redis OSS（停用叢集模式）複寫群組](#) 或 [從頭開始在 Valkey 或 Redis OSS（已啟用叢集模式）中建立複寫群組](#)。

使用 AWS Management Console（Memcached）

使用 建立快取叢集時 AWS Management Console，請在連線能力下選擇網路類型 IPv4IPv6或雙堆疊。如果您選擇雙堆疊，則您必須選擇 Discovery IP 類型 或 IPv6 IPv4。

如需詳細資訊，請參閱[建立 Memcached 叢集\(主控台\)](#)。

CLI 搭配 Valkey、Redis OSS或 Memcached 使用

Redis OSS

OSS 使用 使用 Valkey 或 Redis 建立快取叢集時CLI，您可以使用 [create-cache-cluster](#)命令並指定 `NetworkType`和 `IPDiscovery` 參數：

若為 Linux、macOS 或 Unix :

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine redis \  
  --cache-node-type cache.m5.large \  
  --num-cache-nodes 1 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

針對 Windows :

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id "cluster-test" ^  
  --engine redis ^  
  --cache-node-type cache.m5.large ^  
  --num-cache-nodes 1 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

使用 建立停用叢集模式的複寫群組時CLI，您可以使用 [create-replication-group](#) 命令並指定 `NetworkType` 和 `IPDiscovery` 參數：

若為 Linux、macOS 或 Unix :

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

針對 Windows :

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^
```

```
--num-cache-clusters 3 ^
--primary-cluster-id redis01 ^
--network-type dual_stack ^
--ip-discovery ipv4
```

在建立啟用叢集模式並使用 IPv4 進行 IP 探索的複寫群組時CLI，您可以使用 [create-replication-group](#) 命令並指定 NetworkType 和 IPDiscovery 參數：

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv4 \
  --region us-east-1
```

針對 Windows：

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

在建立啟用叢集模式並使用 IPv6 進行 IP 探索的複寫群組時CLI，您可以使用 [create-replication-group](#) 命令並指定 NetworkType 和 IPDiscovery 參數：

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
```

```
--cache-node-type cache.m5.large \  
--num-node-groups 2 \  
--engine redis \  
--cache-subnet-group-name xyz \  
--network-type dual_stack \  
--ip-discovery ipv6 \  
--region us-east-1
```

針對 Windows :

```
aws elasticache create-replication-group ^  
  --replication-group-id demo-cluster ^  
  --replication-group-description "demo cluster" ^  
  --cache-node-type cache.m5.large ^  
  --num-node-groups 2 ^  
  --engine redis ^  
  --cache-subnet-group-name xyz ^  
  --network-type dual_stack ^  
  --ip-discovery ipv6 ^  
  --region us-east-1
```

Memcached

使用 使用 Memcached 建立快取叢集時CLI，您可以使用 [create-cache-cluster](#) 命令並指定 NetworkType 和 IPDiscovery 參數：

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine memcached \  
  --cache-node-type cache.m5.large \  
  --num-cache-nodes 1 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

針對 Windows：

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id "cluster-test" ^  
  --engine memcached ^
```

```
--cache-node-type cache.m5.large ^
--num-cache-nodes 1 ^
--network-type dual_stack ^
--ip-discovery ipv4
```

自動識別叢集中的節點 (Memcached)

對於執行 Memcached 引擎的叢集，ElastiCache 支援 Auto Discovery：可讓用戶端程式自動識別快取叢集中的所有節點，以及啟動和維護所有這些節點的連線。

Note

Amazon ElastiCache Memcached 上執行的快取叢集會新增 Auto Discovery。Auto Discovery 不適用於 Valkey 或 Redis OSS 引擎。

透過自動探索，您的應用程式便不需要手動連線到個別快取節點；您的應用程式可改為連線到單一 Memcached 節點，並擷取節點清單。藉由該清單，您的應用程式便會感知到叢集中的其餘節點，並可以連線到其中任何一個節點。您不需要在應用程式中硬式編碼個別快取節點端點。

如果您在叢集上使用雙堆疊網路類型，Auto Discovery 只會傳回 IPv4 或 IPv6 地址，具體取決於您選擇的地址。如需詳細資訊，請參閱 [在中選擇網路類型 ElastiCache](#)。

叢集中所有快取節點都會維持一份其他所有節點的中繼資料清單。此中繼資料會在將節點新增到叢集或從叢集中移除節點時更新。

主題

- [使用 Memcached 自動探索的優點](#)
- [自動探索的運作方式](#)
- [使用自動探索](#)
- [手動連線至 Memached 快取節點](#)
- [將自動探索新增至 Memcached 用戶端程式庫](#)
- [ElastiCache 具有自動探索的用戶端](#)

使用 Memcached 自動探索的優點

使用 Memcached 時，Auto Discovery 提供下列優點：

- 當您增加快取叢集中的節點數時，新的節點會自行向組態端點及所有其他節點註冊。當您從快取叢集移除節點時，離開的節點會自行取消註冊。在這兩種情況下，叢集中所有其他的節點都會使用最新的快取節點中繼資料更新。
- 自動偵測快取節點故障，並會自動取代失敗的節點。

Note

在完成更換節點之前，節點會繼續失敗。

- 用戶端程式只需要連線到組態端點。之後，自動探索程式庫便會連線到叢集內所有其他的節點。
- 用戶端程式會每分鐘輪詢叢集一次 (此間隔可視需要進行調整)。若叢集組態發生任何變更 (例如新增節點或刪除節點)，用戶端便會接收到更新後的中繼資料清單。然後，用戶端便會視需要連線 (或中斷連線) 到這些節點。

所有 ElastiCache Memcached 快取叢集上都會啟用 Auto Discovery。您不需要重新啟動任何快取節點，便可使用這項功能。

自動探索的運作方式

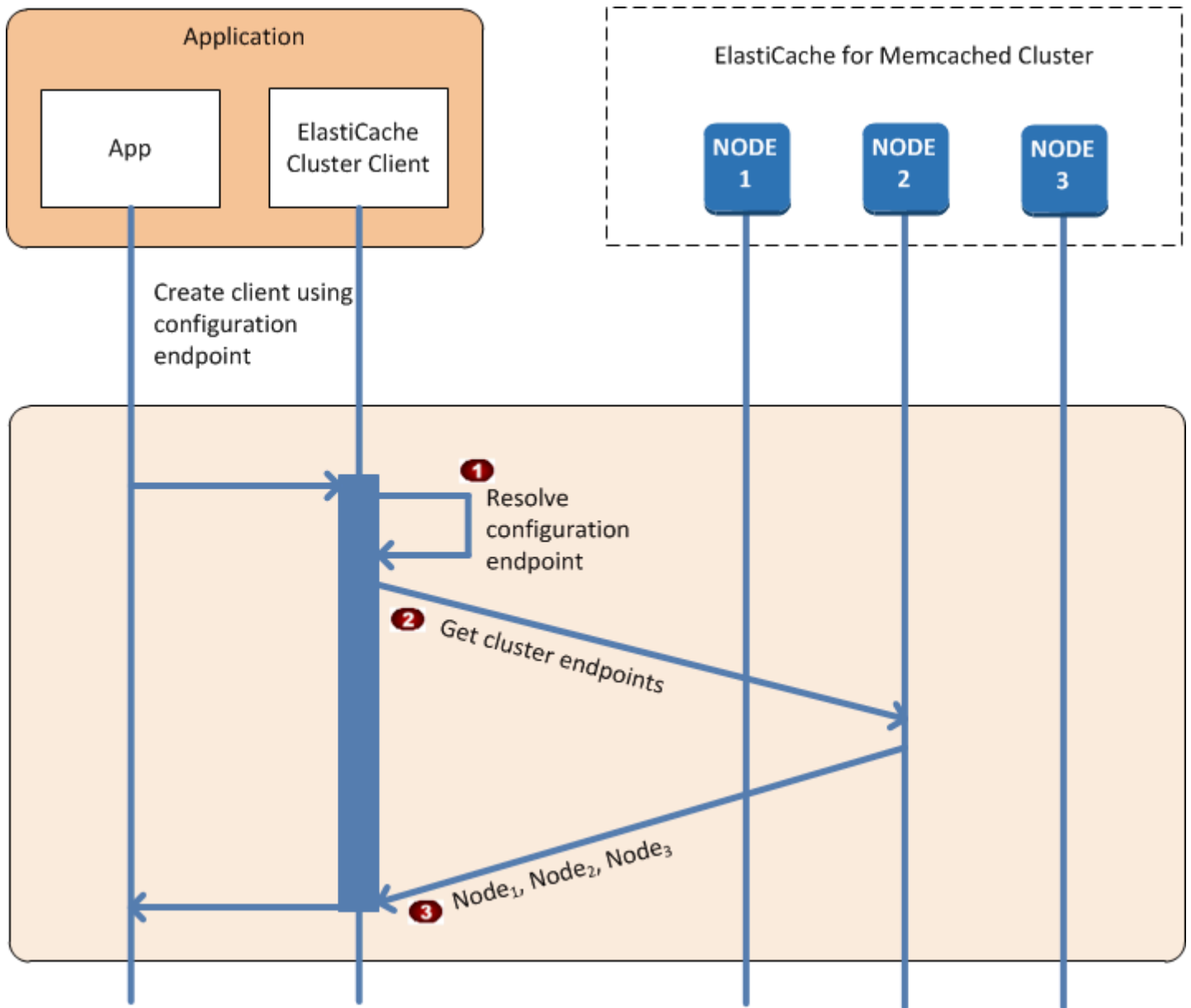
主題

- [連線至快取節點](#)
- [一般叢集操作](#)
- [其他操作](#)

本節說明用戶端應用程式如何使用 ElastiCache 叢集用戶端管理快取節點連線，以及與快取中的資料項目互動。

連線至快取節點

從應用程式的觀點而言，連線到叢集組態端點與直接連線到個別快取節點沒什麼不同。以下順序圖表顯示連線到快取節點的程序。



連線至快取節點的程序

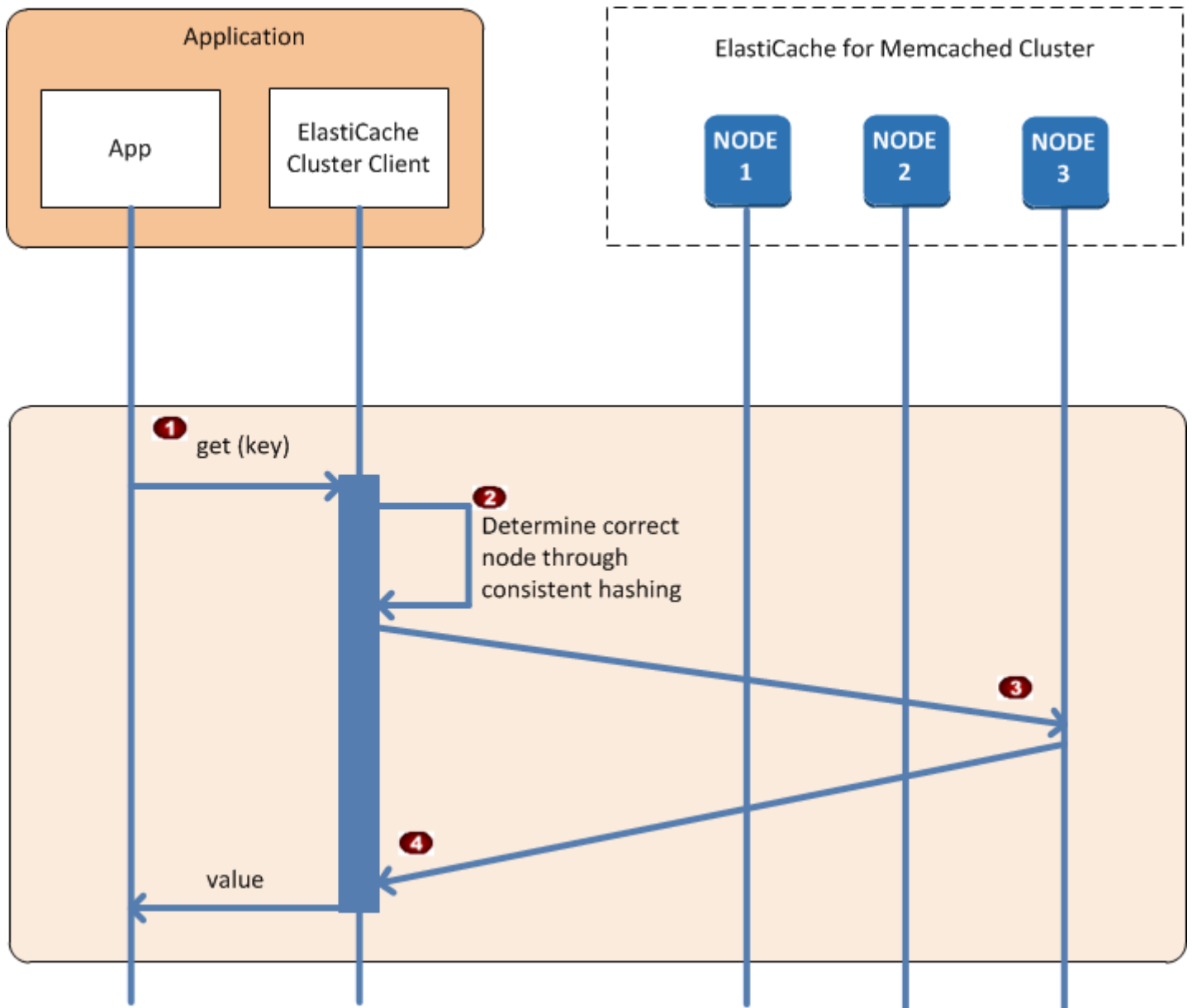
- 應用程式會解析組態端點DNS的名稱。由於組態端點會維護所有快取節點CNAME的項目，因此DNS名稱會解析為其中一個節點；然後用戶端可以連線至該節點。
- 用戶端請求所有其他節點的組態資訊。因為每個節點都會維持叢集中所有節點的組態資訊，任何節點都能在請求時將組態資訊傳遞給用戶端。
- 用戶端接收到目前快取節點主機名稱及IP地址的清單。然後，它便能連線到叢集中所有其他的節點。

Note

用戶端程式每分鐘更新快取節點主機名稱及 IP 地址的清單一次。此輪詢間隔可視需要進行調整。

一般叢集操作

當應用程式連線至所有快取節點時，ElastiCache Cluster Client 會決定哪些節點應存放個別資料項目，以及哪些節點稍後應針對這些資料項目查詢。以下順序圖表顯示一般叢集操作的程序。



一般叢集操作的程序

- 應用程式針對個別資料項目發出 get 請求，以資料項目的鍵進行識別。
- 用戶端針對鍵使用雜湊演算法，判斷哪個快取節點包含該資料項目。
- 隨即從適當節點請求資料項目。
- 資料項目傳回應用程式。

其他操作

在某些情況下，您可能會對叢集的節點進行變更。例如，您可以新增額外節點以滿足其他需求，或刪除節點以在需求減少期間節省成本。或者，您可能會因為任何節點故障而替換掉節點。

當叢集內發生變更，使中繼資料需要更新叢集的端點時，便會同時變更所有節點。因此，任何指定節點中的中繼資料都會與叢集內其他所有節點中的中繼資料保持一致。

在其中任何一種情況下，由於叢集內所有節點的中繼資料都會同時更新，因此所有節點內的中繼資料都會隨時保持一致。建議您一律使用組態端點來取得叢集內各種節點的端點。透過使用組態端點，您可以確保不會從在您面前「消失」的節點取得端點資料。

新增節點

節點啟動期間，中繼資料內不會包含其端點。當節點可供使用時，它便會新增到每個叢集節點的中繼資料。在此案例中，所有節點內的中繼資料都會保持一致，並且您只能在新節點可供使用之後與它互動。在節點可供使用之前，您將無法得知它，並且您會以相同方式與叢集內的節點互動，就好像新節點不存在。

刪除節點

移除節點時，會先從中繼資料移除其端點，然後再從叢集移除節點。在此案例中，所有節點內的中繼資料都會保持一致，並且不會在節點無法使用的同時包含要移除節點的端點。在移除節點期間，中繼資料內不會報告該節點，因此您的應用程式屆時只會與 $n-1$ 個剩餘節點進行互動，宛如節點不存在。

取代節點

如果節點失敗，ElastiCache 會記下該節點並啟動替換。取代程序可能需要幾分鐘。在此期間，所有節點內的中繼資料仍會顯示失敗節點的端點，但任何嘗試與該節點進行的互動都會失敗。因此，建議您的邏輯一律包含重試邏輯。

使用自動探索

若要開始使用 Auto Discovery 搭配 ElastiCache (Memcached) , 請依照下列步驟進行 :

- [取得組態端點](#)
- [下載 ElastiCache 叢集用戶端](#)
- [修改您的應用程式](#)

取得組態端點

為了連線到叢集 , 用戶端程式必須知道叢集的組態端點。請參閱主題 : [尋找叢集的端點 \(主控台 \) \(Memcached \)](#)

您也可以使用 `aws elasticache describe-cache-clusters` 命令搭配 `--show-cache-node-info` 參數 :

無論您用來尋找叢集端點的方法為何 , 組態端點的地址內都會一律包含 `.cfg`。

Example 使用 AWS CLI 的 尋找端點 ElastiCache

若為 Linux、macOS 或 Unix :

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

針對 Windows :

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

此操作會產生類似下列 (JSON 格式) 的輸出 :

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",
```

```
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1e"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1a"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "Multiple",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 2,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
"PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
"CacheNodeType": "cache.r3.large"
```

```
    }  
  ]  
}
```

下載 ElastiCache 叢集用戶端

若要利用 Auto Discovery，用戶端程式必須使用 ElastiCache Cluster Client。ElastiCache Cluster Client 適用於 Java、PHP 和 .NET。NET 包含探索和連線至所有快取節點的所有必要邏輯。

若要下載 ElastiCache 叢集用戶端

1. 登入 AWS 管理主控台，並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 從 ElastiCache 主控台中選擇 ElastiCache Cluster Client，然後選擇下載。

適用於 Java 的 ElastiCache Cluster Client 的原始碼可在 <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java> 取得。此程式庫是以熱門的 Spymemcached 用戶端為基礎。ElastiCache Cluster Client 會根據 Amazon Software License <https://aws.amazon.com/asl> 發行。您可以視需要自由修改來源碼。您甚至可以將程式碼併入其他開放原始碼 Memcached 程式庫，或是併入您自己的用戶端程式碼。

Note

若要將 ElastiCache Cluster Client 用於 PHP，您必須先在 Amazon EC2 執行個體上安裝它。如需詳細資訊，請參閱 [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)。

對於 TLS 支援的用戶端，請下載 7PHP.4 版或更新版本的二進位檔案。

若要將 ElastiCache Cluster Client 用於 .NET，您必須先在 Amazon EC2 執行個體上安裝它。如需詳細資訊，請參閱 [安裝的 ElastiCache 叢集用戶端.NET](#)。

修改您的應用程式

修改您的應用程式以使用自動探索。下列各節說明如何將 ElastiCache Cluster Client 用於 Java、PHP 和 .NET。

Important

指定叢集的組態端點時，請務必確認端點的地址內包含 ".cfg"，如以下所示。請勿使用 CNAME 或中沒有 ".cfg" 的端點。


```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

明確指定叢集組態端點失敗，可能會導致設定至特定節點。

使用適用於 Java 的 ElastiCache 叢集用戶端

下列程式示範如何使用 ElastiCache 叢集用戶端連線至叢集組態端點，並將資料項目新增至快取。使用自動探索，程式便會連線到叢集內的所有節點，而不會進行其他介入。

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the AWS-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
            new InetSocketAddress(configEndpoint,
                clusterPort));

        // The client will connect to the other cache nodes automatically.

        // Store a data item for an hour.
        // The client will decide which cache host will store this item.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

使用 ElastiCache Cluster Client 進行 PHP

下列程式示範如何使用 ElastiCache 叢集用戶端連線至叢集組態端點，並將資料項目新增至快取。使用自動探索，程式將會連線到叢集內的所有節點，而不會進行其他介入。

若要將 ElastiCache Cluster Client 用於 PHP，您必須先在 Amazon EC2 執行個體上安裝它。如需詳細資訊，請參閱 [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)

```
<?php

/**
 * Sample PHP code to show how to integrate with the Amazon ElastiCache
 * Auto Discovery feature.
 */

/* Configuration endpoint to use to initialize memcached client.
 * This is only an example. */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the ElastiCache cluster.
 * This is only an example */
$server_port = 11211;

/**
 * The following will initialize a Memcached client to utilize the Auto Discovery
 * feature.
 *
 * By configuring the client with the Dynamic client mode with single endpoint, the
 * client will periodically use the configuration endpoint to retrieve the current
 * cache
 * cluster configuration. This allows scaling the cache cluster up or down in number
 * of nodes
 * without requiring any changes to the PHP application.
 *
 * By default the Memcached instances are destroyed at the end of the request.
 * To create an instance that persists between requests,
 * use persistent_id to specify a unique ID for the instance.
 * All instances created with the same persistent_id will share the same connection.
 * See http://php.net/manual/en/memcached.construct.php for more information.
 */
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
```

```
*/
$dynamic_client->set('key', 'value', 60);

/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
$static_client = new Memcached('persistent-id');
$static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
$static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
$static_client->set('key', 'value');
?>
```

如需如何在 TLS 啟用的情況下使用 ElastiCache 叢集用戶端的範例，請參閱[使用傳輸中加密搭配 PHP 和 Memcached](#)。

針對使用 ElastiCache 叢集用戶端。NET

Note

ElastiCache .NET 叢集用戶端已於 2022 年 5 月被取代。

。NET 的用戶端 ElastiCache 是位於的開放原始碼<https://github.com/aws-labs/elasticache-cluster-config-net>。

。NET 應用程式通常會從其組態檔案取得其組態。以下是範例應用程式組態檔。

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section
      name="clusterclient"
      type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
Amazon.ElastiCacheCluster" />
```

```
</configSections>

<clusterclient>
  <!-- the hostname and port values are from step 1 above -->
  <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
port="11211" />
</clusterclient>
</configuration>
```

下面的 C# 程式示範如何使用 ElastiCache 叢集用戶端連線到叢集組態端點，並將資料項目新增至快取。使用自動探索，程式將會連線到叢集內的所有節點，而不會進行其他介入。

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCache Auto Discovery
// feature.

using System;

using Amazon.ElastiCacheCluster;

using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    } // end Main

} // end class DotNetAutoDiscoverDemo
```

手動連線至 Memcached 快取節點

如果您的用戶端程式不使用 Auto Discovery，則可以手動連線到每個 Memcached 快取節點。這是 Memcached 用戶端的預設行為。

您可以從 [AWS 管理主控台](#) 取得快取節點主機名稱及連接埠號碼的清單。您也可以將 AWS CLI `aws elasticache describe-cache-clusters` 命令與 `--show-cache-node-info` 參數搭配使用。

Example

以下 Java 程式碼片段示範如何連線到四節點快取叢集內的所有節點：

```
...  
  
ArrayList<String> cacheNodes = new ArrayList<String>(  
    Arrays.asList(  
        "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));  
  
MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));  
  
...
```

Important

若您透過新增或移除節點向上或向下擴展快取叢集，您將需要在用戶端程式碼中更新節點清單。

將自動探索新增至 Memcached 用戶端程式庫

Auto Discovery 的組態資訊會以備援方式存放在每個 Memcached 快取叢集節點中。用戶端應用程式可查詢任何快取節點，並取得叢集內所有節點的組態資訊。

應用程式執行此作業的方式取決於快取引擎版本：

- 若快取引擎版本為 1.4.14 或更高版本，請使用 `config` 命令。
- 若快取引擎版本低於 1.4.14，請使用 `get AmazonElastiCache:cluster` 命令。

這兩個命令的輸出完全相同，並且會在以下的[輸出格式](#)一節中說明。

1.4.14 或更高的快取引擎版本

針對 1.4.14 或更高的快取引擎版本，請使用 `config` 命令。此命令已由新增至 Memcached ASCII 和二進位通訊協定 ElastiCache，並在 ElastiCache 叢集用戶端中實作。若您希望搭配其他用戶端程式庫使用自動探索，該程式庫必須進行擴充以支援 `config` 命令。

Note

下列文件與 ASCII 通訊協定相關；但 `config` 命令同時支援 ASCII 和二進位檔案。如果您想要使用二進位通訊協定新增 Auto Discovery 支援，請參閱 [ElastiCache Cluster Client 的原始程式碼](#)。

語法

```
config [sub-command] [key]
```

選項

名稱	描述	必要
sub-command	用來與快取節點互動的子命令。針對自動探索，此子命令為 <code>get</code> 。	是
key	鍵名稱，叢集組態會存放在該鍵之下。針對自動探索，此鍵名為 <code>cluster</code> 。	是

若要取得叢集組態資訊，請使用以下命令：

```
config get cluster
```

1.4.14 以下的快取引擎版本

若要取得叢集組態資訊，請使用以下命令：

```
get AmazonElastiCache:cluster
```

Note

請勿竄改「AmazonElastiCache:cluster」金鑰，因為這是叢集組態資訊所在的位置。如果您覆寫此金鑰，則在 ElastiCache 自動且正確更新組態資訊之前，用戶端可能會在短時間內（不超過 15 秒）設定不正確。

輸出格式

無論您使用 `config get cluster` 或 `get AmazonElastiCache:cluster`，回應都由兩行文字組成：

- 組態資訊的版本編號。每當新增節點至快取叢集，或從快取叢集移除節點時，版本編號都會加一。
- 快取節點的清單。清單中的每個節點都會以 `hostname|ip-address|port` 群組表示，並且每個節點都會以一個空格分隔。

歸位字元和換行字元 (CR + LF) 會出現在每一行的結尾。資料行的結尾包含一個換行字元 (LF)，CR + LF 會新增到該處。組態版本行會以不帶 CR 的 LF 終止。

包含三個節點的快取叢集會以以下方式表示：

```
configversion\nhostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

每個節點都會顯示 CNAME 和私有 IP 地址。CNAME 永遠存在；如果無法使用私有 IP 地址，則不會顯示該地址；不過，仍會列印管道字元 "|"。

Example

以下是當您查詢組態資訊時，所傳回承載的範例：

```
CONFIG cluster 0 136\r\n
12\r\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\r\n\r\n
END\r\n
```

Note

- 第二行指出組態資訊至今已經過修改十二次。
- 在第三行中，節點清單是以主機名稱的字母順序排序。此順序的排列方式可能會跟您目前於用戶端應用程式中使用的順序不同。

ElastiCache 具有自動探索的用戶端

叢集用戶端程式可以自動識別並連線至執行 Memcached 引擎的所有快取叢集節點。

本節討論如何安裝和設定 ElastiCache PHP 和 .NET 用戶端，以搭配自動探索使用。

主題

- [安裝及編譯叢集用戶端](#)
- [設定 ElastiCache 用戶端](#)

安裝及編譯叢集用戶端

本節涵蓋安裝、設定和編譯 PHP 和 .NET Amazon ElastiCache Auto Discovery 叢集用戶端。

主題

- [安裝的 ElastiCache 叢集用戶端。NET](#)
- [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)
- [編譯 ElastiCache 叢集用戶端的原始程式碼 PHP](#)

安裝的 ElastiCache 叢集用戶端。NET

您可以找到 ElastiCache .NET 叢集用戶端程式碼作為的開放原始碼<https://github.com/awslabs/elasticache-cluster-config-net>。

本節說明如何在 Amazon EC2 執行個體上安裝、更新和移除叢集用戶端的 ElastiCache 。NET 元件。如需自動探索的詳細資訊，請參閱 [自動識別叢集中的節點 \(Memcached \)](#)。如需使用用戶端的範例 .NET 程式碼，請參閱 [針對使用 ElastiCache 叢集用戶端。NET](#)。

主題

- [安裝 。NET](#)
- [下載的 ElastiCache .NET 叢集用戶端 ElastiCache](#)
- [使用安裝 AWS 組件 NuGet](#)

安裝 。NET

您必須具有 。NET 已安裝 3.5 或更新版本，以使用 AWS .NET SDK for ElastiCache。如果您沒有 。NET 3.5 或更新版本，您可以從 <http://www.microsoft.com/net> 下載並安裝最新版本。

下載的 ElastiCache .NET 叢集用戶端 ElastiCache

若要下載 ElastiCache .NET 叢集用戶端

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，按一下 ElastiCache 叢集用戶端。
3. 在下載 ElastiCache Memcached Cluster Client 清單中，選取 .NET，然後按一下下載。

使用安裝 AWS 組件 NuGet

NuGet 是平台 NET 的套件管理系統。NuGet 知道組件相依性，並自動安裝所有必要的檔案。NuGet 已安裝的組件會與您的解決方案一起存放，而不是存放在等中央位置 Program Files，因此您可以安裝應用程式特有的版本，而不會產生相容性問題。

安裝 NuGet

NuGet 可以從 [上的安裝圖庫安裝 MSDN](https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c)；請參閱 <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>。如果您使用 Visual Studio 2010 或更新版本，NuGet 則會自動安裝。

您可以從 Solution Explorer NuGet 或 Package Manager 主控台 使用。

NuGet 從 Solution Explorer 使用

從 Visual Studio 2010 中的 NuGet Solution Explorer 使用

1. 從 Tools (工具) 選單，選取 Library Package Manager (程式庫套件管理員)。
2. 按一下 Package Manager Console (套件管理員主控台)。

從 Visual Studio 2012 或 Visual Studio 2013 中的 NuGet Solution Explorer 使用

1. 從工具功能表中，選取 NuGet Package Manager。
2. 按一下 Package Manager Console (套件管理員主控台)。

從命令列，您可以使用 Install-Package 安裝組件，如下所示。

```
Install-Package Amazon.ElastiCacheCluster
```

若要查看透過提供的每個套件的頁面 NuGet，例如 AWS SDK 和 AWS。擴充元件，請參閱 NuGet 網站 <https://http://www.nuget.org>。每個套件的頁面包含範例命令列，用於使用主控台安裝套件，以及可透過取得的舊版套件清單 NuGet。

如需 Package Manager Console (套件管理員主控台) 命令的詳細資訊，請參閱 <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>。

安裝適用於 PHP 的 ElastiCache 叢集用戶端

本節說明如何在 Amazon EC2 執行個體上安裝、更新及移除適用於 ElastiCache 叢集用戶端的 PHP 元件。如需自動探索的詳細資訊，請參閱[自動識別叢集中的節點 \(Memcached \)](#)。如需要使用用戶端的範例 PHP 程式碼，請參閱[使用 ElastiCache Cluster Client 進行 PHP](#)。

主題

- [下載安裝套件](#)
- [針對已安裝 php-memcached 延伸的使用者](#)
- [新使用者的安裝步驟](#)
- [移除PHP叢集用戶端](#)

下載安裝套件

若要確保使用適用於的正確 ElastiCache 叢集用戶端版本PHP，您需要知道 PHP Amazon EC2執行個體上安裝了哪個版本的。您還需要知道 Amazon EC2執行個體是否正在執行 64 位元或 32 位元版本的 Linux。

判斷安裝在 Amazon EC2執行個體上的PHP版本

- 在命令提示中，執行下列命令：

```
php -v
```

PHP 版本會顯示在輸出中，如本範例所示：

```
PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
```

Note

如果您的 PHP 和 Memcached 版本不相容，您會收到錯誤訊息，如下所示：

```
PHP Warning: PHP Startup: memcached: Unable to initialize module
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
```

```
in Unknown on line 0
```

如果發生這種情況，您需要從來源碼編譯模組。如需詳細資訊，請參閱[編譯 ElastiCache 叢集用戶端的原始程式碼 PHP](#)。

判斷您的 Amazon EC2AMI 架構（64 位元或 32 位元）

1. 登入 AWS Management Console 並在 開啟 Amazon EC2 主控台 <https://console.aws.amazon.com/ec2/>。
2. 在執行個體清單中，按一下您的 Amazon EC2 執行個體。
3. 在描述索引標籤中，尋找 AMI：欄位。64 位元執行個體應該使用 x86_64 做為描述的一部分；若為 32 位元執行個體，請在此欄位中尋找 i386 或 i686。

您現在可以下載 ElastiCache 叢集用戶端。

下載的 ElastiCache 叢集用戶端 PHP

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從 ElastiCache 主控台中選擇 ElastiCache Cluster Client。
3. 從下載 ElastiCache Memcached 叢集用戶端清單中，選擇符合您 PHP 版本和 AMI 架構的 ElastiCache 叢集用戶端，然後選擇下載按鈕。

針對已安裝 php-memcached 延伸的使用者

更新 **php-memcached** 安裝

1. 移除先前安裝之適用於 PHP 的 Memcached 延伸，如[移除 PHP 叢集用戶端](#)主題所述。
2. 安裝新的 ElastiCache php-memcached 延伸，如先前「[新使用者的安裝步驟](#)」中所述。

新使用者的安裝步驟

主題

- [為新使用者安裝 PHP 7.x](#)
- [為新使用者安裝 PHP 5.x](#)

為新使用者安裝 PHP 7.x

主題

- [在 Ubuntu 伺服器 PHP 14.04 上安裝 7 LTSAMI \(64 位元和 32 位元 \)](#)
- [在 PHP Amazon Linux 201609 上安裝 7 AMI](#)
- [在 SUSE Linux 上安裝 PHP 7 AMI](#)

在 Ubuntu 伺服器 PHP 14.04 上安裝 7 LTSAMI (64 位元和 32 位元)

1. 從 啟動新執行個體AMI。
2. 執行下列命令：

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. 安裝 PHP 7。

```
sudo yum install php70
```

4. 下載 Amazon ElastiCache Cluster Client。

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. 將 latest-64bit 解壓縮。

```
tar -zxvf latest-64bit
```

6. 使用 root 許可，將解壓縮的成品檔案 `amazon-elasticache-cluster-client.so` 複製到 `/usr/lib/php/20151012`。

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. 將 `extension=amazon-elasticache-cluster-client.so` 這一行插入到 `/etc/php/7.0/cli/php.ini` 檔案。

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php/7.0/cli/php.ini
```

8. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

在 PHP Amazon Linux 201609 上安裝 7 AMI

1. 從 啟動新執行個體AMI。
2. 執行以下命令：

```
sudo yum install gcc-c++
```

3. 安裝 PHP 7。

```
sudo yum install php70
```

4. 下載 Amazon ElastiCache Cluster Client。

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/  
latest-64bit
```

5. 將 latest-64bit 解壓縮。

```
tar -zxvf latest-64bit
```

6. 使用 root 許可，將解壓縮的成品檔案 `amazon-elasticache-cluster-client.so` 複製到 `/usr/lib64/php/7.0/modules/`。

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. 建立 `50-memcached.ini` 檔案。

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php-7.0.d/50-memcached.ini
```

8. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

在 SUSE Linux 上安裝 PHP 7 AMI

1. 從 啟動新執行個體AMI。
2. 執行以下命令：

```
sudo zypper install gcc
```

3. 安裝 PHP 7。

```
sudo yum install php70
```

4. 下載 Amazon ElastiCache Cluster Client。

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/  
latest-64bit
```

5. 將 latest-64bit 解壓縮。

```
tar -zxvf latest-64bit
```

6. 使用 root 許可，將解壓縮的成品檔案 amazon-elasticache-cluster-client.so 複製到 /usr/lib64/php7/extensions/。

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. 將 extension=amazon-elasticache-cluster-client.so 這一行插入到 /etc/php7/cli/php.ini 檔案。

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php7/cli/php.ini
```

8. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

為新使用者安裝 PHP 5.x

主題

- [在 PHP Amazon Linux 2014.03 上安裝 AMI 5 \(64 位元和 32 位元 \)](#)
- [在 Red Hat Enterprise Linux PHP 7.0 AMI \(64 位元和 32 位元 \) 上安裝 5](#)
- [在 Ubuntu 伺服器 PHP 14.04 上安裝 5 LTSAMI \(64 位元和 32 位元 \)](#)
- [安裝 PHP 5 for SUSE Linux 企業伺服器 11 AMI \(64 位元或 32 位元 \)](#)
- [其他 Linux 發行版本](#)

在 PHP Amazon Linux 2014.03 上安裝 AMI 5 (64 位元和 32 位元)

1. 啟動 Amazon Linux 執行個體 (64 位元或 32 位元) , 並登入其中。
2. 安裝 PHP 相依性 :

```
sudo yum install gcc-c++ php php-pear
```

3. 為您的 Amazon EC2 執行個體和 PHP 版本下載正確的 php-memcached 套件。如需詳細資訊 , 請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑 :

```
sudo pecl install <package download path>
```

以下是 PHP 5.4、64 位元 Linux 的安裝命令範例。在此範例中 , 取代 *X.Y.Z* 實際版本編號 :

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

Note

務必使用安裝成品的最新版本。

5. 使用 root/sudo 許可 , memcached.ini 在 /etc/php.d 目錄中新增名為 的新檔案 , 並在檔案中插入 「extension=amazon-elasticache-cluster-client.so」 :

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php.d/memcached.ini
```

6. 啟動或重新啟動您的 Apache 伺服器。


```
sudo /etc/init.d/httpd start
```

在 Red Hat Enterprise Linux PHP 7.0 AMI (64 位元和 32 位元) 上安裝 5

1. 啟動 Red Hat Enterprise Linux 執行個體 (64 位元或 32 位元) , 並登入其中。
2. 安裝PHP相依性 :

```
sudo yum install gcc-c++ php php-pear
```

3. 為您的 Amazon EC2執行個體和PHP版本下載正確的php-memcached套件。如需詳細資訊 , 請參閱[下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑 :

```
sudo pecl install <package download path>
```

5. 使用 root/sudo 許可 , 在 /etc/php.d 目錄中新增名為 memcached.ini 的新檔案 , 並在該檔案中插入 extension=amazon-elasticache-cluster-client.so。

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

6. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```


在 Ubuntu 伺服器 PHP 14.04 上安裝 5 LTSAMI (64 位元和 32 位元)

1. 啟動 Ubuntu Linux 執行個體 (64 位元或 32 位元) , 並登入其中。
2. 安裝PHP相依性 :

```
sudo apt-get update
sudo apt-get install gcc g++ php5 php-pear
```

3. 為您的 Amazon EC2 執行個體和 PHP 版本下載正確的 php-memcached 套件。如需詳細資訊，請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑。

```
sudo pecl install <package download path>
```

 Note

此安裝步驟會將建置成品 `amazon-elasticache-cluster-client.so` 安裝到 `/usr/lib/php5/20121212*` 目錄。驗證組建成品的絕對路徑，因為您在下一個步驟中需要它。

如果上一個命令無效，您需要 `amazon-elasticache-cluster-client.so` 從下載 `*.tgz` 的檔案手動擷取 PHP 用戶端成品，並將其複製到 `/usr/lib/php5/20121212*` 目錄。

```
tar -xvf <package download path>
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. 使用 `root/sudo` 許可，`memcached.ini` 在 `/etc/php5/cli/conf.d` 目錄中新增名為 `memcached.ini` 的新檔案，並在檔案中插入「`extension=<absolute path to amazon-elasticache-cluster-client.so>`」。

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee
--append /etc/php5/cli/conf.d/memcached.ini
```

6. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

安裝 PHP 5 for SUSE Linux 企業伺服器 11 AMI (64 位元或 32 位元)

1. 啟動 SUSE Linux 執行個體 (64 位元或 32 位元) 並登入。
2. 安裝 PHP 相依性：

```
sudo zypper install gcc php53-devel
```

3. 為您的 Amazon EC2 執行個體和 PHP 版本下載正確的 php-memcached 套件。如需詳細資訊，請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑。

```
sudo pecl install <package download path>
```

5. 使用 root/sudo 許可，在 /etc/php5/conf.d 目錄中新增名為 memcached.ini 的新檔案，並在該檔案中插入 **extension=amazon-elasticache-cluster-client.so**。

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

Note

如果步驟 5 對先前的任何平台沒有作用，請驗證 amazon-elasticache-cluster-client.so 的安裝路徑。同時，請指定延伸中二進位檔的完整路徑。此外，請確認使用 PHP 中的是支援的版本。我們支援 5.3 版到 5.5 版。

其他 Linux 發行版本

在某些系統上，特別是 CentOS7 和 Red Hat Enterprise Linux (RHEL) 7.1 libsas12.so.3 已取代。libsas12.so.2 在這些系統上，當您載入 ElastiCache 叢集用戶端時，它會嘗試，但找不到並載入 libsas12.so.2。若要解決此問題，請建立指向 libsas12.so.3 的符號連結，以便在用戶端嘗試載入 libsas12.so.2 時，將其重新導向至 libsas12.so.3。下列程式碼會建立此符號連結。

```
cd /usr/lib64
sudo ln libsas12.so.3 libsas12.so.2
```

移除 PHP 叢集用戶端

主題

- [移除舊版 PHP 7](#)
- [移除舊版 PHP 5](#)

移除舊版 PHP 7

若要移除舊版 PHP 7

1. 從適當的 PHP lib 目錄中移除amazon-elasticache-cluster-client.so檔案，如安裝說明中先前所示。請參閱[針對已安裝 php-memcached 延伸的使用者](#)中您的安裝所適用章節。
2. 從 php.ini 檔案中移除 extension=amazon-elasticache-cluster-client.so 這一行。
3. 啟動或重新啟動您的 Apache 伺服器。

```
sudo /etc/init.d/httpd start
```

移除舊版 PHP 5

若要移除 5 PHP 的較早版本

1. 移除 php-memcached 延伸：

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. 移除適當目錄中所新增的 memcached.ini 檔案，如先前安裝步驟中所述。

編譯 ElastiCache 叢集用戶端的原始程式碼 PHP

本節說明如何取得和編譯的 ElastiCache Cluster Client 的原始程式碼 PHP。

您需要從 GitHub 和編譯兩個套件：[aws-elasticache-cluster-client-libmemcached](https://github.com/aws/aws-elasticache-cluster-client-libmemcached) 和 [aws-elasticache-cluster-client-memcached-for-php](https://github.com/aws/aws-elasticache-cluster-client-memcached-for-php)。

主題

- [編譯 libmemcached 程式庫](#)
- [編譯的 ElastiCache Memcached 自動探索用戶端 PHP](#)

編譯 libmemcached 程式庫

若要編譯 aws-elasticache-cluster-client-libmemcached 程式庫

1. 啟動 Amazon EC2 執行個體。
2. 安裝程式庫的依存項目。
 - 在 Amazon Linux 201509 上 AMI

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- 在 Ubuntu 14.04 上 AMI

```
sudo apt-get update
sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
```

3. 提取儲存庫並編譯程式碼。

```
Download and install https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz
```

編譯的 ElastiCache Memcached 自動探索用戶端 PHP

下列各節說明如何編譯 ElastiCache Memcached Auto Discovery Client

主題

- [編譯 7 的 ElastiCache Memcached PHP 用戶端](#)
- [編譯 5 的 ElastiCache Memcached PHP 用戶端](#)

編譯 7 的 ElastiCache Memcached PHP 用戶端

在程式碼目錄下執行以下命令組。

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
make
make install
```

Note

您可以靜態地將 libmemcached PHP 程式庫連結至二進位，以便將其移植到各種 Linux 平台。若要執行此作業，請在 make 前執行以下命令：

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

編譯 5 的 ElastiCache Memcached PHP 用戶端

透過在 aws-elasticache-cluster-client-memcached-for-php/ 資料夾下執行以下命令，來編譯 aws-elasticache-cluster-client-memcached-for-php。

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
make install
```

設定 ElastiCache 用戶端

ElastiCache 叢集符合 Valkey、Redis OSS 和 Memcached 的通訊協定。您現在用於現有環境的程式碼、應用程式和最熱門工具，將能與服務無縫搭配使用。

本節討論在 [中](#) 連線至快取節點的特定考量 ElastiCache。

主題

- [受限制的命令](#)
- [尋找節點端點及連接埠號碼](#)
- [進行連線以使用自動探索](#)
- [連線至 Valkey 或 Redis OSS 叢集中的節點](#)
- [DNS 名稱和基礎 IP](#)

受限制的命令

若要提供受管服務體驗，ElastiCache 會限制存取某些需要進階權限的快取引擎特定命令。針對執行 Redis 的快取叢集，以下命令無法使用：

- bgrewriteaof
- bgsave
- config
- debug
- migrate
- replicaof
- save
- slaveof
- shutdown
- sync

尋找節點端點及連接埠號碼

若要連線到快取節點，您的應用程式需要知道該節點的端點及連接埠號碼。

尋找節點端點及連接埠號碼 (主控台)

判斷節點端點和連接埠號碼

1. 登入 [Amazon ElastiCache 管理主控台](#)，然後選擇叢集上執行的引擎。

接著會出現執行所選擇引擎的所有叢集清單。

2. 在您執行的引擎和組態下繼續。
3. 選擇您有興趣的叢集名稱。
4. 尋找您有興趣節點的 Port (連接埠) 及 Endpoint (端點) 欄。

尋找快取節點端點及連接埠號碼 (AWS CLI)

若要判斷快取節點端點和連接埠號碼，請搭配 `--show-cache-node-info` 參數使用 `describe-cache-clusters` 命令。

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

完整DNS名稱和連接埠號碼位於輸出的端點區段中。

尋找快取節點端點和連接埠號碼 (ElastiCache API)

若要判斷快取節點端點和連接埠號碼，請搭配 `ShowCacheNodeInfo=true` 參數使用 `DescribeCacheClusters` 動作。

Example

```
https://elasticache.us-west-2.amazonaws.com /
?Action=DescribeCacheClusters
&ShowCacheNodeInfo=true
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140421T220302Z
&Version=2014-09-30
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20140421T220302Z
&X-Amz-Expires=20140421T220302Z
&X-Amz-Signature=<signature>
```



```
&X-Amz-SignedHeaders=Host
```

進行連線以使用自動探索

若您的應用程式使用自動探索，您只需要知道叢集的組態端點，而無須知道每個快取節點的個別端點。如需詳細資訊，請參閱[自動識別叢集中的節點 \(Memcached \)](#)。

Note

目前，自動探索僅供執行 Memcached 的快取叢集使用。

連線至 Valkey 或 Redis OSS叢集中的節點

Note

目前，支援複寫和僅供讀取複本的叢集 (API/CLI：複寫群組) 僅支援執行 Valkey 或 Redis 的叢集OSS。

對於叢集，ElastiCache 提供主控台、CLI和 API 介面，以取得個別節點的連線資訊。

針對唯讀活動，應用程式可連線到叢集中的任何節點。不過，對於寫入活動，我們建議您的應用程式連線至叢集的主要端點 (Valkey 或 Redis OSS (停用叢集模式)) 或組態端點 (Valkey 或 Redis OSS (啟用叢集模式))，而不是直接連線至節點。這可確保您的應用程式總能找到正確的節點，即使您決定透過將僅供讀取複本提升到主要角色來重新設定叢集也一樣。

連線到複寫群組中的叢集 (主控台)

判斷端點和連接埠號碼

- 請參閱主題：[尋找 Valkey 或 Redis OSS \(停用叢集模式 \) 叢集的端點 \(主控台 \)](#)。

連線到複寫群組中的叢集 (AWS CLI)

判斷快取節點端點和連接埠號碼

搭配您複寫群組的名稱，使用 describe-replication-groups 命令：

```
aws elasticache describe-replication-groups redis2x2
```

此命令應該產生類似下列的輸出：

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "2 shards, 2 nodes (1 + 1 replica)",
      "NodeGroups": [
        {
          "Status": "available",
          "Slots": "0-8191",
          "NodeGroupId": "0001",
          "NodeGroupMembers": [
            {
              "PreferredAvailabilityZone": "us-west-2c",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-001"
            },
            {
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-002"
            }
          ]
        }
      ],
    },
    {
      "Status": "available",
      "Slots": "8192-16383",
      "NodeGroupId": "0002",
      "NodeGroupMembers": [
        {
          "PreferredAvailabilityZone": "us-west-2b",
          "CacheNodeId": "0001",
          "CacheClusterId": "redis2x2-0002-001"
        },
        {
          "PreferredAvailabilityZone": "us-west-2a",
          "CacheNodeId": "0001",
          "CacheClusterId": "redis2x2-0002-002"
        }
      ]
    }
  ],
  "ConfigurationEndpoint": {
```

```

        "Port": 6379,
        "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "redis2x2",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
        "redis2x2-0001-001",
        "redis2x2-0001-002",
        "redis2x2-0002-001",
        "redis2x2-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "PendingModifiedValues": {}
}
]
}

```

連線至複寫群組中的叢集 (ElastiCache API)

判斷快取節點端點和連接埠號碼

使用下列參數呼叫 DescribeReplicationGroups :

ReplicationGroupId = 您複寫群組的名稱。

Example

```

https://elasticache.us-west-2.amazonaws.com /
?Action=DescribeCacheClusters
&ReplicationGroupId=repgroup01
&Version=2014-09-30
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140421T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20140421T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20140421T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>

```

DNS 名稱和基礎 IP

用戶端會維持一份伺服器清單，其中包含保有快取資料的伺服器地址及連接埠。使用時 ElastiCache，DescribeCacheClusters API（或 describe-cache-clusters 命令列公用程式）會傳回可用於伺服器清單的完整 DNS 項目和連接埠號碼。

Important

用戶端應用程式必須設定為在嘗試連線到快取節點端點時，經常解析快取節點 DNS 的名稱。

VPC 安裝

ElastiCache 確保快取節點 DNS 的名稱和 IP 地址在故障時復原快取節點時保持不變。

非VPC安裝

ElastiCache 確保在快取節點復原失敗時，快取節點 DNS 的名稱保持不變；不過，快取節點的基礎 IP 地址可能會變更。

根據預設，大多數的用戶端程式庫都支援持久性快取節點連線。建議您在使用時，使用持久性快取節點連線 ElastiCache。用戶端 DNS 快取可能發生在多個位置，包括用戶端程式庫、語言執行期或用戶端作業系統。建議您檢閱每一個 layer 的應用程式組態，確保為您的快取節點頻繁解析 IP 地址。

中的資料分層 ElastiCache

ElastiCache 包含複寫群組並使用 r6gd 系列中節點類型的 Valkey 或 Redis OSS 叢集，其資料會在記憶體和本機 SSD（固態硬碟）儲存之間分層。資料分層除了將資料儲存在記憶體之外，還在每個叢集節點中使用低成本的固態硬碟（SSDs），為 Valkey 或 Redis OSS 工作負載提供新的價格效能選項。它非常適合定期存取其整體資料集高達 20% 的工作負載，以及可在存取上的資料時容忍額外延遲的應用程式 SSD。

在具有資料分層的 ElastiCache 叢集上，ElastiCache 監控其儲存的每個項目的上次存取時間。當可用的記憶體（DRAM）耗盡時，ElastiCache 會使用最近最少使用的（LRU）演算法，自動將不常存取的項目從記憶體移至 SSD。SSD 隨後存取上的資料時，ElastiCache 會自動和非同步地將其移回記憶體，然後再處理請求。如果您的工作負載只會定期存取其資料的子集，則資料分層是以符合成本效益的方式擴展容量的最佳方式。

請注意，使用資料分層時，金鑰本身一律會保留在記憶體中，而則 LRU 管理記憶體與磁碟上值的位置。一般而言，建議在使用資料分層時，金鑰大小小於您值的大小。

資料分層專為盡量降低對應用程式工作負載的效能影響所設計。例如，假設 500 位元組字串值，與記憶體SSD中的資料相比，對儲存在上的資料提出請求，平均可預期額外 300 微秒的延遲。

使用最大的資料分層節點大小 (cache.r6gd.16xlarge)，可以在單一 500 個節點的叢集中存放高達 1 PB (使用 1 個僅供讀取複本時為 500 TB)。資料分層與中支援的所有 Valkey 或 Redis OSS命令和資料結構相容 ElastiCache。不需要任何用戶端變更就能使用此功能。

主題

- [最佳實務](#)
- [限制](#)
- [定價](#)
- [監控](#)
- [使用資料分層](#)
- [在資料分層啟用的情況下，將資料從備份還原到叢集](#)

最佳實務

建議遵循下列最佳實務：

- 資料分層非常適合定期存取其整體資料集高達 20% 的工作負載，以及可在存取上的資料時容忍額外延遲的應用程式SSD。
- 在資料分層節點上使用可用SSD容量時，我們建議值大小大於金鑰大小。當項目在 DRAM和之間移動時SSD，金鑰一律會保留在記憶體中，而且只有值會移至 SSD 層。

限制

資料分層具有下列限制：

- 只能在屬於複寫群組一部分的叢集上使用資料分層。
- 使用的節點類型必須來自 r6gd 系列，該系列在下列區域可用：us-east-2、us-east-1、us-west-2、us-west-1、eu-west-1、eu-central-1、eu-north-1、eu-west-3、ap-northeast-1、ap-southeast-1、ap-southeast-2、ap-south-1、ca-central-1 和 sa-east-1。
- 您必須使用 Valkey 7.2 或更新版本的引擎，或 Redis 6.2 OSS 或更新版本的引擎。
- 無法將 r6gd 叢集的備份還原到另一個叢集，除非該叢集也使用 r6gd。

- 無法將備份匯出到 Amazon S3 以用於資料分層叢集。
- 在 r6gd 節點類型上執行的叢集不支援線上遷移。
- 不支援從資料分層叢集 (例如，使用 r6gd 節點類型的叢集) 擴展到未使用資料分層的叢集 (例如，使用 r6g 節點類型的叢集)。如需詳細資訊，請參閱[擴展 ElastiCache](#)。
- 針對 Valkey 7.2 版及更新版本，以及 Redis 7.0.7 版及更新OSS版本，使用資料分層的叢集支援自動擴展。如需詳細資訊，請參閱[Auto Scaling Valkey 和 Redis OSS叢集](#)
- 資料分層僅支援 volatile-lru、allkeys-lru、volatile-lfu、allkeys-lfu 和 noeviction 最大記憶體政策。
- Valkey 7.2 版和更新版本以及 Redis 7.0.7 版和更新OSS版本支援無痕儲存。如需詳細資訊，請參閱[同步與備份的實作方式](#)。
- 大於 128 MiB 的項目不會移至 SSD。

定價

R6gd 節點的總容量是 R6g 節點 (記憶體 + SSD) 的 4.8 倍，相較於 R6g 節點 (僅限記憶體) ，可協助您在以最大使用率執行時節省超過 60%。如需詳細資訊，請參閱[ElastiCache 定價](#)。

監控

ElastiCache 提供專為監控使用資料分層之效能叢集而設計的指標。若要監控 中項目DRAM相較於 的比率SSD，您可以使用 Valkey 和 Redis CurrItems 的指標。[OSS](#)您可以計算百分比為：
(CurrItems 使用維度：Tier = Memory * 100) / (CurrItems 沒有維度篩選條件)。

如果設定的清除政策允許，則當記憶體中的項目百分比低於 5% 時，ElastiCache 將開始清除項目。在設定了省略政策的節點上，寫入操作將收到記憶體不足錯誤。

當記憶體中的項目百分比低於 5% 時，仍建議您考慮擴展叢集模式啟用叢集或擴展叢集模式停用叢集。如需擴展的詳細資訊，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \(已啟用叢集模式 \)](#)。如需使用資料分層之 Valkey 或 Redis OSS叢集指標的詳細資訊，請參閱 [Valkey 和 Redis 的指標 OSS](#)。

使用資料分層

使用 使用資料分層 AWS Management Console

在建立叢集做為複寫群組的一部分時，可以從 r6gd 系列中選取節點類型來使用資料分層，例如cache.r6gd.xlarge。選取該節點類型會自動啟用資料分層。

如需有關建立叢集的詳細資訊，請參閱 [為 Valkey 或 Redis 建立叢集 OSS](#)。

使用 啟用資料分層 AWS CLI

使用 建立複寫群組時 AWS CLI，您可以從 r6gd 系列中選取節點類型，例如 cache.r6gd.xlarge 和設定 `--data-tiering-enabled` 參數，以使用資料分層。

從 r6gd 系列中選取節點類型時，無法選擇退出資料分層。如果您設定 `--no-data-tiering-enabled` 參數，操作將會失敗。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled
```

針對 Windows：

```
aws elasticache create-replication-group ^\  
  --replication-group-id redis-dt-cluster ^\  
  --replication-group-description "Redis OSS cluster with data tiering" ^\  
  --num-node-groups 1 ^\  
  --replicas-per-node-group 1 ^\  
  --cache-node-type cache.r6gd.xlarge ^\  
  --engine redis ^\  
  --cache-subnet-group-name default ^\  
  --automatic-failover-enabled ^\  
  --data-tiering-enabled
```

執行此操作之後，將會看到類似如下的回應：

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "redis-dt-cluster",
```

```
    "Description": "Redis OSS cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
        "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
}
}
```

在資料分層啟用的情況下，將資料從備份還原到叢集

您可以使用（主控台）、（AWS CLI）或（）將備份還原至啟用資料分層的新叢集ElastiCache API。當使用 r6gd 系列中的節點類型建立叢集時，會啟用資料分層。

在資料分層啟用的情況下，將資料從備份還原到叢集 (主控台)

在資料分層啟用的情況下，將備份還原到新叢集 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格，選擇 Backups (備份)。
3. 在備份清單中，選擇您要從中還原之備份名稱左側的方塊。
4. 選擇 Restore (還原)。
5. 完成 Restore Cluster (還原叢集) 對話方塊。務必完成所有必填欄位，以及您想要變更預設值的任何其他欄位。

1. Cluster ID (叢集 ID - 必要。新叢集的名稱)。
2. 啟用叢集模式（橫向擴展） – 為 Valkey 或 Redis OSS（啟用叢集模式）叢集選擇此選項。
3. 節點類型 – 指定 cache.r6gd.xlarge 或任何其他 r6gd 系列中的節點類型。
4. 碎片數量 – 選擇您希望在新叢集（API/CLI：節點群組）中的碎片數量。
5. Replicas per Shard (每個碎片的複本數) - 選擇您要讓每個碎片具備的僅供讀取複本節點數。

6. Slots and keyspaces (位置和 Keyspace) - 選擇您要如何將索引鍵分佈到不同的碎片中。如果您選擇指定金鑰分佈，請指定每個碎片的金鑰範圍，以完成資料表。
 7. Availability zone(s) (可用區域) - 指定您要如何選取叢集的可用區域。
 8. Port (連接埠) - 只有在您希望新叢集使用不同的連接埠時，才變更此選項。
 9. 選擇 VPC- 選擇要在VPC其中建立此叢集的。
 10. 參數群組 – 選擇參數群組，為您選取的節點類型保留足夠的記憶體，以用於 Valkey 或 Redis OSS額外負荷。
6. 當您滿意設定後，請選擇 Create (建立)。

如需有關建立叢集的詳細資訊，請參閱 [為 Valkey 或 Redis 建立叢集 OSS](#)。

在資料分層啟用的情況下，將資料從備份還原到叢集 (AWS CLI)

使用 建立複寫群組時 AWS CLI，預設會使用資料分層，方法是從 r6gd 系列中選取節點類型，例如 cache.r6gd.xlarge 和設定 --data-tiering-enabled 參數。

從 r6gd 系列中選取節點類型時，無法選擇退出資料分層。如果您設定 --no-data-tiering-enabled 參數，操作將會失敗。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled \  
  --snapshot-name my-snapshot
```

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group ^ \  
  --replication-group-id redis-dt-cluster ^ \  
  --replication-group-description "Redis OSS cluster with data tiering" ^
```

```
--num-node-groups 1 ^
--replicas-per-node-group 1 ^
--cache-node-type cache.r6gd.xlarge ^
--engine redis ^
--cache-subnet-group-name default ^
--automatic-failover-enabled ^
--data-tiering-enabled ^
--snapshot-name my-snapshot
```

執行此操作之後，將會看到類似如下的回應：

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "redis-dt-cluster",
    "Description": "Redis OSS cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

在中準備叢集 ElastiCache

接下來，您可以找到使用 ElastiCache 主控台、AWS CLI 或 建立叢集的指示 ElastiCache API。

您也可以使用 建立 ElastiCache 叢集 [AWS CloudFormation](#)。如需詳細資訊，請參閱 AWS Cloud Formation 使用者指南 中的 [AWS : : ElastiCache : : CacheCluster](#) : ，其中包含如何實作該方法的指引。

每當您建立叢集或複寫群組，建議您進行一些準備工作，將來就不需要升級或當下進行變更。

主題

- [判斷 ElastiCache 叢集需求](#)
- [選擇您的節點大小](#)

判斷 ElastiCache 叢集需求

準備

了解下列問題的答案有助於讓建立 ElastiCache 叢集更順暢：

- 您需要哪種類型的節點執行個體？

如需選擇執行個體節點類型的指導方針，請參閱[選擇您的節點大小](#)。

- 您是否會在以 Amazon 為基礎的虛擬私有雲端（VPC）中啟動叢集VPC？

Important

如果您要在 VPC 中啟動叢集VPC，請務必在開始建立叢集VPC之前，在相同的 VPC 中建立子網路群組。如需詳細資訊，請參閱[子網路和子網路群組](#)。

ElastiCache 旨在 AWS 使用 Amazon 由 VPC 內部存取 EC2。不過，如果您在 VPC 以 Amazon 為基礎的 VPC 中啟動叢集，且叢集位於 VPC 中，則您可以從外部提供存取權 AWS。如需詳細資訊，請參閱[從外部存取 ElastiCache 資源 AWS](#)。

- 您是否需要自訂任何參數值？

如果需要，請建立自訂參數群組。如需詳細資訊，請參閱[建立 ElastiCache 參數群組](#)。

如果您正在執行 Valkey 或 Redis OSS，請考慮設定 reserved-memory 或 reserved-memory-percent。如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

- 您需要建立自己的VPC安全群組嗎？

如需詳細資訊，請參閱[VPC 中的安全VPC](#)。

- 您要如何實作容錯能力？

如需詳細資訊，請參閱[減少故障](#)。

主題

- [ElastiCache 記憶體和處理器需求](#)
- [Memcached 叢集組態](#)

- [Valkey 和 Redis OSS 叢集組態](#)
- [ElastiCache 擴展需求](#)
- [ElastiCache 存取需求](#)
- [的區域、可用區域和本機區域需求 ElastiCache](#)

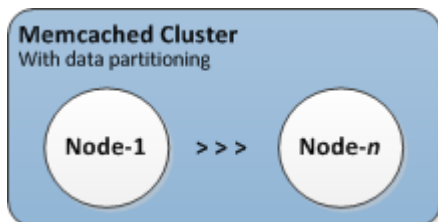
ElastiCache 記憶體和處理器需求

Amazon 的基本建置區塊 ElastiCache 是節點。您可以逐一設定節點或以群組形式設定節點，以形成叢集。當您判斷要為叢集使用何種節點類型時，請一併考量叢集的節點組態和您要存放的資料量。

Memcached 是多執行緒的引擎，因此節點的核心數量會影響叢集可用的運算能力。

Memcached 叢集組態

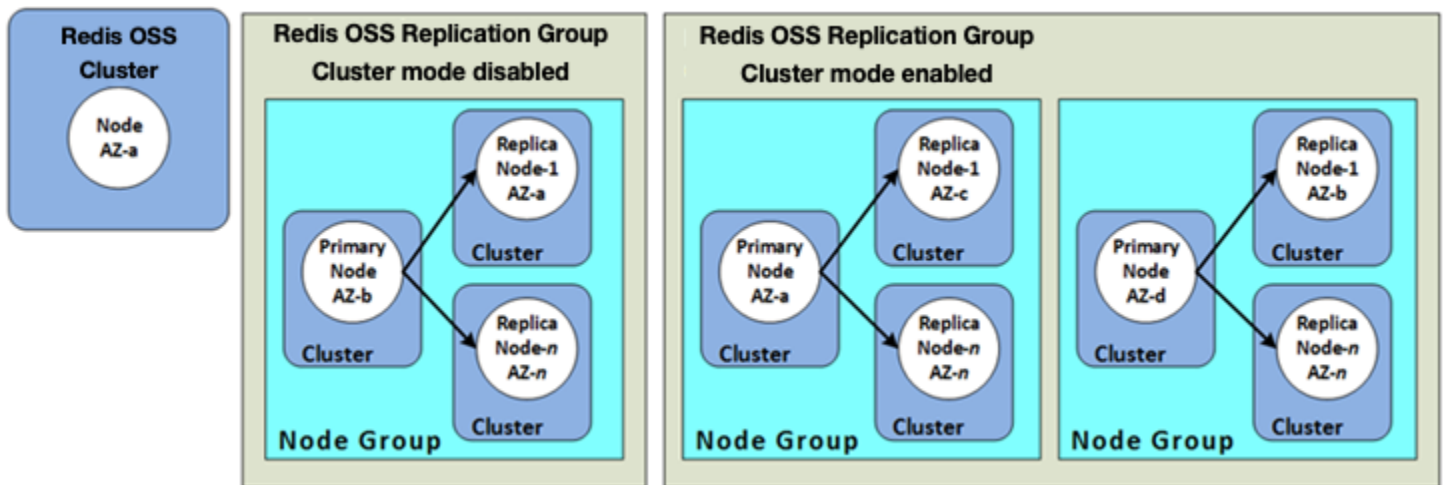
ElastiCache (Memcached) 叢集由 1 到 60 個節點組成。Memcached 叢集中的資料會分割到叢集的各節點中。您的應用程式會使用網路位址 (稱為「端點」) 來與 Memcached 叢集連線。Memcached 叢集中的每個節點都有自己的端點，您的應用程式會使用這些端點來讀取或寫入特定節點。除了節點端點之外，Memcached 叢集本身還有一個端點名為組態端點。您的應用程式可使用此端點來讀取或寫入叢集，並由自動探索判斷要讀取或寫入哪些節點。



如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

Valkey 和 Redis OSS 叢集組態

ElastiCache 使用 Valkey 和 Redis OSS 叢集由 0 到 500 個碎片 (也稱為節點群組) 組成。Valkey 或 Redis OSS 叢集中的資料會分割為叢集中的碎片。您的應用程式會使用稱為端點的網路地址，與 Valkey 或 Redis OSS 叢集連線。Valkey 或 Redis OSS 碎片中的節點會滿足兩個角色之一：一個讀取/寫入主要節點和所有其他節點唯讀次要 (也稱為僅供讀取複本)。除了節點端點之外，Valkey 或 Redis OSS 叢集本身還有一個名為組態端點的端點。您的應用程式可以使用此端點從叢集讀取或寫入，從而決定要從哪個節點讀取或寫入最多 ElastiCache (Redis OSS)。



如需詳細資訊，請參閱[在中管理叢集 ElastiCache](#)。

ElastiCache 擴展需求

您可以使用更大的新節點類型來建立新叢集，以擴展所有叢集的規模。當您擴展 Memcached 叢集時，新的叢集會開始清空。當您擴展 Valkey 或 Redis OSS 叢集時，您可以從備份中植入它，並避免讓新叢集開始為空。

Amazon ElastiCache for Memcached 叢集可以向外擴展或向內擴展。若要擴展或縮減 Memcached 叢集的規模，您只要新增或移除叢集的節點即可。如果您已啟用 Automatic Discovery 且應用程式已連線至叢集的組態端點，則當您新增或移除節點時不需要對應用程式進行任何變更。

如需詳細資訊，請參閱本指南中的[擴展 ElastiCache](#)。

ElastiCache 存取需求

Amazon ElastiCache 叢集的設計是從 Amazon EC2 執行個體存取。ElastiCache 叢集的網路存取權僅限於建立叢集的帳戶。因此，您必須先授權 Amazon EC2 執行個體存取叢集，才能從 Amazon EC2 執行個體存取叢集。執行此操作的步驟會有所不同，具體取決於您是在 EC2-VPC 或 EC2-Classic 中啟動。

如果您將叢集啟動至 EC2-VPC 您需要將網路輸入授予叢集。如果您將叢集啟動至 EC2-Classic，您需要將與執行個體存取相關聯的 Amazon Elastic Compute Cloud 安全群組授予 ElastiCache 安全群組。如需詳細說明，請參閱本指南的[步驟 3。授權對叢集的存取](#)。

的區域、可用區域和本機區域需求 ElastiCache

Amazon ElastiCache 支援所有 AWS 區域。透過將 ElastiCache 叢集放置在靠近應用程式的 AWS 區域中，您可以減少延遲。如果您的叢集有多個節點，將節點安置在不同可用區域或 Local Zones 中可降低故障對叢集的影響。

如需詳細資訊，請參閱下列內容：

- [選擇的區域和可用區域 ElastiCache](#)
- [搭配使用本機區域 ElastiCache](#)
- [減少故障](#)

選擇您的節點大小

您為 ElastiCache 叢集選取的節點大小會影響成本、效能和容錯能力。

節點大小（Valkey 和 RedisOSS）

如需有關 Graviton 處理器優點的詳細資訊，請參閱 [AWS Graviton 處理器](#)。

回答下列問題可協助您判斷OSS您實作 Valkey 或 Redis 所需的最低節點類型：

- 您是否預期會有具有多個用戶端連線的輸送量繫結工作負載？

如果是這種情況，而且您執行的是 Redis 5.0OSS.6 版或更新版本，您可以使用我們增強的 I/O 功能來獲得更好的輸送量和延遲，其中可用 CPUs 代表 Redis OSS引擎卸載用戶端連線。如果您執行 Redis 7OSS.0.4 版或更新版本，除了增強型 I/O 之外，您還可以透過增強型 I/O 多工處理獲得額外的加速，其中每個專用網路 IO 執行緒管道都會從多個用戶端命令進入 Redis OSS引擎，利用 Redis OSS的 可有效率地分批處理命令的能力。在 ElastiCache（Redis OSS）v7.1 和更新版本中，我們擴展了增強的 I/O 執行緒功能，以處理呈現層邏輯。依呈現層而言，我們的意思是增強型 I/O 執行緒現在不僅正在讀取用戶端輸入，而且還會將輸入剖析為 Redis OSS 二進位命令格式，然後轉送到主執行緒以執行，從而提高效能。如需其他詳細資訊，請參閱[部落格文章](#)和[支援的版本](#)頁面。

- 您是否有經常存取少量資料的工作負載？

如果是這種情況，而且您在 Redis OSS引擎 6.2 版或更新版本上執行，您可以選擇 r6gd 節點類型來利用資料分層。使用資料分層時，最近最少使用的資料會儲存在中SSD。當檢索資料時，會有少量的延遲成本，可藉由成本節省加以抵銷。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

如需詳細資訊，請參閱[支援的節點類型](#)。

- 您的資料需要多少總記憶體？

若要取得一般估計值，請使用要快取的項目大小。將此大小乘以您希望同時保持在快取中的項目數。若要取得合理的項目大小估計，收先請序列化您的快取項目，接著計算字元數。然後再除以您叢集中碎片數。

如需詳細資訊，請參閱[支援的節點類型](#)。

- OSS 您正在執行什麼版本的 Redis ？

2.8.22 之前的 Redis OSS 版本需要您為容錯移轉、快照、同步和將複本提升為主要操作預留更多記憶體。此需求的原因是您必須針對處理期間所有發生的寫入擁有足夠的記憶體。

Redis 2.8.22 版及更新 OSS 版本使用無叉儲存程序，其所需的記憶體比先前程序更少。

如需詳細資訊，請參閱下列內容：

- [同步與備份的實作方式](#)
- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)
- 您應用程式所需要的大量寫入程度為何？

需要大量寫入的應用程式可能需要在取得快照或容錯移轉時擁有大上許多的可用記憶體 (尚未由資料使用的記憶體)。每次執行 BGSAVE 處理序時，都必須有資料未使用的足夠記憶體，以容納執行 BGSAVE 處理序時產生的所有寫入。範例包括拍攝快照、使用叢集中的複本同步主要叢集，以及啟用僅附加檔案 (AOF) 功能。另一個範例是將複本提升為主要節點 (若您已啟用多個可用區) 時。最糟糕的情況是執行處理序過程中所有資料都被重寫。在這種情況下，節點執行個體的大小需為單純資料所需記憶體的兩倍。

如需詳細資訊，請參閱[確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)。

- 您的實作是獨立的 Valkey 或 Redis OSS (停用叢集模式) 叢集，還是具有多個碎片的 Valkey 或 Redis OSS (啟用叢集模式) 叢集？

Valkey 或 Redis OSS (停用叢集模式) 叢集

如果您要實作 Valkey 或 Redis OSS (停用叢集模式) 叢集，您的節點類型必須能夠容納所有資料，以及必要的額外負荷，如上一則所述。

例如，假設您估計所有項目的總大小為 12 GB。在這種情況下，您可以使用具有 13.3 GB 記憶體的 `cache.m3.xlarge` 節點或記憶體為 13.5 GB 的 `cache.r3.large` 節點。但是，您可能需要更多記憶體以執行 BGSAVE 作業。如果您的應用程式需執行大量寫入，請將記憶體需求加倍為至少 24 GB。因此，可使用 `cache.m3.2xlarge` (含 27.9 GB 記憶體) 或 `cache.r3.xlarge` (含 30.5 GB 記憶體)。

具有多個碎片的 Valkey 或 Redis OSS (啟用叢集模式)

如果您要使用多個碎片實作 Valkey 或 Redis OSS (啟用叢集模式) 叢集，則節點類型必須能夠容納資料 $\text{bytes-for-data-and-overhead} / \text{number-of-shards}$ 位元組。

例如，假設您估計所有項目的總大小為 12 GB 且有兩個碎片。在這種情況下，您可以使用具有 6.05 GB 記憶體 (12 GB/2) 的 `cache.m3.large` 節點。但是，您可能需要更多記憶體以執行 BGSAVE 作業。如果您的應用程式需執行大量寫入，請將記憶體需求加倍為每個碎片至少 12 GB。因此，可使用 `cache.m3.xlarge` (含 13.3 GB 記憶體) 或 `cache.r3.large` (含 13.5 GB 記憶體)。

- 您是否有使用 Local Zones ?

[本機區域](#)可讓您將 ElastiCache 叢集等資源放置在使用者附近的多個位置。但是選擇節點大小時請注意，無論容量要求為何，目前可用節點都會限制為下列大小：

- 最新一代：

M5 節點類

型：`cache.m5.large`、`cache.m5.xlarge`、`cache.m5.2xlarge`、`cache.m5.4xlarge`、`cache.`

R5 節點類

型：`cache.r5.large`、`cache.r5.xlarge`、`cache.r5.2xlarge`、`cache.r5.4xlarge`、`cache.`

T3 節點類型：`cache.t3.micro`、`cache.t3.small`、`cache.t3.medium`

當您的叢集執行時，您可以監控記憶體用量、處理器使用率、快取命中和快取遺漏發佈至的指標 CloudWatch。您可能會注意到叢集沒有您想要的命中率，或是索引鍵被移出的頻率太頻繁。在這些情況下，您可以選擇具有較大 CPU 和記憶體規格的不同節點大小。

監控 CPU 用量時，請記住 Valkey 和 Redis OSS 是單執行緒。因此，將報告的 CPU 用量乘以 CPU 核心數量，以取得實際用量。例如，CPU 報告 20% 使用率的四核心實際上是 Redis 以 80% 使用率 OSS 執行的一個核心。

節點大小 (Memcached)

Memcached 叢集包含一或多個節點，其中叢集的資料會分割在各個節點中。因此，叢集的記憶體需求與節點的記憶體相關，但不相同。您可以透過擁有幾個大型節點或數個較小的節點，來達到您需要的叢集記憶體容量。除此之外，隨著您的需求變更，您可以新增節點或從叢集移除節點，來只為您需要的項目支付費用。

叢集的總記憶體容量的計算方式是將叢集中的節點數量乘以扣除系統額外負荷後每個節點的 RAM 容量。每個節點的容量是根據節點類型而定。


```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

叢集中的節點數量是您執行 Memcached 叢集可用性的關鍵因素。單一節點發生故障，可能會影響您應用程式的可用性及後端資料庫的負載。在這種情況下，ElastiCache 會為失敗的節點佈建替換，然後重新填入節點。您可以將記憶體和運算容量分散到更多容量較小的節點間，而非使用少數具備高容量的節點，藉此降低此可用性影響。

在希望擁有 35 GB 快取記憶體的案例中，可以透過以下任一組態進行設定：

- 11 個 `cache.t2.medium` 節點，其具備 3.22 GB 的記憶體及 2 個執行緒，合計 35.42 GB 和 22 個執行緒。
- 6 個 `cache.m4.large` 節點，其具備 6.42 GB 的記憶體及 2 個執行緒，合計 38.52 GB 和 12 個執行緒。
- 3 個 `cache.r4.large` 節點，其具備 12.3 GB 的記憶體及 2 個執行緒，合計 36.90 GB 和 6 個執行緒。
- 3 個 `cache.m4.xlarge` 節點，其具備 14.28 GB 的記憶體及 4 個執行緒，合計 42.84 GB 和 12 個執行緒。

比較節點選項

節點類型	記憶體 (以 GiB 為單位)	核心	每小時成 本 *	所需節點 數	總記憶體 (以 GiB 為單位)	總核心數	每月成 本 /
cache.t2. medium	3.22	2	0.068 USD	11	35.42	22	538.56 USD
cache.m4. large	6.42	2	0.156 USD	6	38.52	12	673.92 USD
cache.m4. xlarge	14.28	4	0.311 USD	3	42.84	12	671.76 USD
cache.m5. xlarge	12.93	4	0.311 USD	3	38.81	12	671.76 USD
cache.m6g .large	6.85	2	\$ 0.147	6	41.1	12	\$ 635

節點類型	記憶體 (以 GiB 為單位)	核心	每小時成 本 *	所需節點 數	總記憶體 (以 GiB 為單位)	總核心數	每月成 本 /
cache.r4. large	12.3	2	0.228 USD	3	36.9	6	492.48 USD
cache.r5. large	13.07	2	0.216 USD	3	39.22	6	466.56 USD
cache.r6g .large	13.07	2	\$ 0.205	3	42.12	6	\$ 442

* 截至 2020 年 10 月 8 日的每個節點小時成本。

用量 100%，持續 30 天 (720 小時) 的每月費用。

這些選項每個都能提供相似的記憶體容量，但不同的運算容量及成本。若要比較特定選項的成本，請參閱 [Amazon ElastiCache Pricing](#)。

針對執行 Memcached 的叢集，每個節點上一部分的可用記憶體會用於連線額外負荷。如需詳細資訊，請參閱 [Memcached 連線額外負荷](#)

使用多個節點需要將索引鍵分散到這些節點當中。每個節點都有自己的端點。為了輕鬆管理端點，您可以使用 Auto Discovery ElastiCache 功能，讓用戶端程式自動識別叢集中的所有節點。如需詳細資訊，請參閱 [自動識別叢集中的節點 \(Memcached \)](#)。

在某些案例中，您可能不確定需要多少容量。這個情況下，建議從一個 cache.m5.large 節點開始測試。然後使用發佈至 Amazon 的 ElastiCache 指標來監控記憶體用量、CPU 使用率和快取命中率 CloudWatch。如需 CloudWatch 指標的詳細資訊 ElastiCache，請參閱 [監控與 CloudWatch 指標搭配使用](#)。對於生產和較大的工作負載，R5 節點提供最佳效能和 RAM 成本值。

若您的叢集沒有達到所需的命中率，您可輕易新增更多節點來增加叢集中可用的總記憶體。

如果您的叢集受約束，CPU 但具有足夠的命中率，請使用節點類型來設定新的叢集，以提供更多運算能力。

為 Valkey 或 Redis 建立叢集 OSS

下列範例示範如何使用 AWS Management Console、AWS CLI 和 建立 Valkey 或 Redis OSS叢集 ElastiCache API。

建立 Valkey 或 Redis OSS (停用叢集模式) (主控台)

ElastiCache 當您使用 Valkey 或 Redis OSS引擎時，支援複寫。若要監控將資料寫入 Valkey 或 Redis OSS讀取/寫入主要叢集，以及將資料傳播至唯讀次要叢集之間的延遲，會將特殊金鑰 ElastiCache 新增至叢集ElastiCacheMasterReplicationTimestamp。此金鑰是目前的通用時間 (UTC)。由於 Valkey 或 Redis OSS叢集稍後可能會新增至複寫群組，因此此金鑰會包含在所有 Valkey 或 Redis OSS叢集中，即使最初不是複寫群組的成員。如需複寫群組的詳細資訊，請參閱[使用複寫群組的高可用性](#)。

若要建立 Valkey 或 Redis OSS (停用叢集模式) 叢集，請依照的步驟進行[建立 Valkey \(停用叢集模式 \) 叢集 \(主控台 \)](#)。

一旦叢集的狀態可用，您就可以授予 Amazon EC2存取權、連線到叢集，並開始使用叢集。如需詳細資訊，請參閱 [步驟 3. 授權對叢集的存取](#) 和 [步驟 4. 連線至叢集的節點](#)。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (主控台)

如果您執行 Redis OSS 3.2.4 或更新版本，您可以建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集。Valkey 或 Redis OSS (啟用叢集模式) 叢集支援將資料分割至 1 到 500 個碎片 (API/CLI : 節點群組)，但有一些限制。如需 Valkey 或 Redis OSS (停用叢集模式) 和 Valkey 或 Redis OSS (啟用叢集模式) 的比較，請參閱 [支援的引擎和版本](#)。

使用 ElastiCache 主控台建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇要啟動此叢集 AWS 的區域。
3. 從導覽窗格選擇 Get started (開始使用)。

4. 選擇建立VPC並遵循[建立虛擬私有雲端 \(VPC \)](#) 中概述的步驟。
5. 在 ElastiCache 儀表板頁面上，選擇建立叢集，然後選擇建立 Valkey 叢集或建立 Redis OSS叢集。
6. 在 Cluster settings (叢集設定) 中，執行下列動作：
 - a. 選擇 Configure and create a new cluster (設定和建立新叢集)。
 - b. 針對 Cluster mode (叢集模式)，選擇 Enabled (已啟用)。
 - c. 針對 Cluster info (叢集資訊)，輸入 Name (名稱) 值。
 - d. (選用) 輸入 Description (描述) 值。
7. 在 Location (位置) 中：

AWS Cloud

1. 針對 AWS 雲端，建議您接受多可用區域和自動容錯移轉的預設設定。如需詳細資訊，請參閱[使用多可用區 將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)。
2. 在 Cluster settings (叢集設定) 中
 - a. 針對 Engine version (引擎版本)，選擇可用的版本。
 - b. 針對 Port (連接埠)，使用預設連接埠 6379。如果您基於任何原因需要使用不同的連接埠，請輸入連接埠號碼。
 - c. 對於 Parameter group (參數群組)，請選擇參數群組或新建參數群組。參數群組可控制叢集的執行時間參數。如需參數群組的詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#)和[建立 ElastiCache 參數群組](#)。

Note

當您選取參數群組來設定引擎組態值時，該參數群組會套用至全域資料存放區中的所有叢集。在 Parameter Groups (參數群組) 頁面上，yes/no (是/否) Global (全域) 屬性指出參數群組是否為全域資料存放區的一部分。

- d. 在 Node type (節點類型) 中，選擇向下箭頭 (▼)。
在 Change node type (變更節點類型) 對話方塊中，為您想要的節點類型選擇 Instance family (執行個體系列) 的值。然後選擇您要用於此叢集的節點類型，然後選擇 Save (儲存)。

如需詳細資訊，請參閱[選擇您的節點大小](#)。

如果選擇 r6gd 節點類型，將自動啟用資料分層。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

- e. 針對碎片數量，選擇您想要在此 Valkey 或 Redis（啟用叢集模式）叢集中取得的碎片數量 OSS（分割區/節點群組）。

對於某些版本的 Valkey 或 Redis OSS（啟用叢集模式），您可以動態變更叢集中的碎片數量：

- Redis OSS 3.2.10 及更新版本 – 如果您的叢集執行 Redis OSS 3.2.10 或更新版本，您可以動態變更叢集中的碎片數量。如需詳細資訊，請參閱[在 Valkey 或 Redis 中擴展叢集 OSS（已啟用叢集模式）](#)。
 - 其他 Redis OSS 版本 – 如果您的叢集在 3.2.10 版本 OSS 之前執行 Redis 版本，則有其他方法。在這種情況下，若要變更叢集中的碎片數量，請使用新的碎片數量建立新的叢集。如需詳細資訊，請參閱[從備份還原到新的快取](#)。
- f. 針對 Replicas per shard（每個碎片的複本），選擇您要讓每個碎片具備的僅供讀取複本節點數目。

Valkey 或 Redis 存在下列限制 OSS（啟用叢集模式）。

- 如果您已啟用多個可用區，請確保每個碎片至少有一個複本。
- 使用主控台建立叢集時，每個碎片的複本數都相同。
- 每個碎片的僅供讀取複本數均固定而無法變更。如果您發現每個碎片需要更多或更少的複本（API/CLI：節點群組），則必須使用新的複本數目建立新的叢集。如需詳細資訊，請參閱[教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。

3. 在 Connectivity（連線）下

- a. 針對 Network type（網路類型），請選擇此叢集支援的 IP 版本。
- b. 對於子網路群組，請選擇您要套用至此叢集的子網路。ElastiCache 使用該子網路群組來選擇該子網路中的子網路和 IP 地址，以與節點建立關聯。ElastiCache 叢集需要同時具有 IPv4 和 IPv6 的雙堆疊子網路，以雙堆疊模式操作，而 IPv6 僅的子網路則僅以作為 IPv6 操作。

建立新的子網路群組時，請輸入其所屬的 VPC ID。

選取 Discovery IP type（探索 IP 類型）。僅回傳您選擇的通訊協定 IP 地址。

如需詳細資訊，請參閱：

- [在中選擇網路類型 ElastiCache](#)。
- [在中建立子網路VPC](#)。

如果您是 [搭配使用本機區域 ElastiCache](#)，必須建立或選擇位於本地區域中的子網路。

如需詳細資訊，請參閱[子網路和子網路群組](#)。

4. 針對 Availability zone placements (可用區域置放)，您有兩種選項：

- 無偏好設定 – ElastiCache 選擇可用區域。
- Specify availability zones (指定可用區域) - 您可為每個叢集指定可用區域。


如果您選擇指定可用區域，請針對每個碎片中的每個叢集，從清單中選擇可用區域。

如需詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。

5. 選擇 Next (下一步)

6. 在進階 Valkey 設定或進階 Redis OSS設定下，或

- 針對 Security (安全)：
 - i. 若要加密資料，您有下列選項：
 - Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

 Note

您可以選擇 Customer Managed AWS KMS 金鑰並選擇 [金鑰](#)，以提供不同的加密金鑰。如需詳細資訊，請參閱[從 AWS 使用客戶受管金鑰 KMS](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。對於 Valkey 7.2 及更高版本或 Redis OSS引擎 6.0 及更高版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：

- No Access Control (無存取控制) - 這是預設的設定。這表示使用者存取叢集時不受任何限制。
- User Group Access Control List (使用者群組存取控制清單) - 選取的使用者群組需具有可存取叢集的已定義使用者集。如需詳細資訊，請參閱[使用主控台和管理使用者群組 CLI](#)。
- AUTH 預設使用者 – Valkey 或 Redis OSS 伺服器的身份驗證機制。如需詳細資訊，請參閱 [AUTH](#)。
- AUTH – Valkey 或 Redis OSS 伺服器的身份驗證機制。如需詳細資訊，請參閱 [AUTH](#)。

Note

對於 3.2.6 版以後的 Redis OSS 版本，除了 3.2.10 版之外，AUTH 是唯一的選項。

- ii. 在 Security groups (安全群組) 中，選擇要用於此叢集的安全群組。安全群組可做為防火牆來控制叢集的網路存取。您可以使用的預設安全群組，VPC 或建立新的安全群組。

如需安全群組的詳細資訊，請參閱 Amazon 使用者指南中的 [的安全群組 VPC](#)。

7. 針對定期排程自動備份，請選取 Enable automatic backups (啟用自動備份)，然後輸入每份自動備份要保存的天數，過了此天數後即會自動將其刪除。如果您不想定期排程自動備份，請清除 Enable automatic backups (啟用自動備份) 核取方塊。在任何一種情況下，您都可以選擇建立手動備份。

如需備份和還原的詳細資訊，請參閱 [快照和還原](#)。

8. (選用) 指定維護時段。維護時段是每週 ElastiCache 排程叢集系統維護的時間，通常為一小時。您可以允許 ElastiCache 為您的維護時段選擇日期和時間（無偏好設定），也可以自行選擇天數、時間和持續時間（指定維護時段）。如果您從清單中選擇 Specify maintenance window (指定維護時段)，請為您的維護時段選擇 Start day (開始日)、Start time (開始時間) 和 Duration (持續時間)。所有時間都是 UCT 次。

如需詳細資訊，請參閱 [管理 ElastiCache 叢集維護](#)。

9. (選用) 針對 Logs (日誌)：

- 在日誌格式下，選擇文字或 JSON。
 - 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
 - 在日誌目的地下，選擇建立新 CloudWatch 日誌群組名稱或 Firehose 串流名稱，然後選擇選取現有日誌群組名稱或 Firehose 串流名稱，然後選擇您的 CloudWatch 日誌群組名稱或您的 Firehose 串流名稱。
10. 對於標籤，為協助您管理叢集和其他 ElastiCache 資源，您可以標籤形式將自己的中繼資料指派給每個資源。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。
 11. 選擇 Next (下一步)。
 12. 檢閱所有項目和選項，然後進行任何所需的更正。當您準備好時，請選擇 Create (建立)。

On premises

1. 針對 On premises (內部部署)，建議您將 Auto-failover (自動容錯移轉) 保持啟用。如需詳細資訊，請參閱[使用多可用區將 ElastiCache \(Redis OSS \) 中的停機時間降到最低](#)
2. 按照[使用 Outposts](#) 的步驟執行。

若要使用 ElastiCache API 或 AWS CLI 而非 ElastiCache 主控台建立對等項目，請參閱以下內容：

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

一旦叢集的狀態可用，您就可以授予其 EC2 存取權、連線到它，並開始使用它。如需詳細資訊，請參閱[步驟 3. 授權對叢集的存取](#) 和 [步驟 4. 連線至叢集的節點](#)。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立叢集 (AWS CLI)

若要使用 建立叢集 AWS CLI，請使用 `create-cache-cluster` 命令。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立 Valkey 或 Redis OSS (停用叢集模式) 叢集 (CLI)

Example – 沒有僅供讀取複本的 Valkey 或 Redis OSS (停用叢集模式) 叢集

下列CLI程式碼會建立沒有複本的 Valkey 或 Redis OSS (停用叢集模式) 快取叢集。

Note

使用 `r6gd` 系列中的節點類型建立叢集時，必須傳遞 `data-tiering-enabled` 參數。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--cache-parameter-group default.redis6.x \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

針對 Windows：

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--cache-parameter-group default.redis6.x ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (AWS CLI)

無法使用 `create-cache-cluster` 操作建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (API/CLI : 複寫群組)。若要建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (API/CLI : 複寫群組)，請參閱 [從頭開始建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 複寫群組 \(AWS CLI \)](#)。

如需詳細資訊，請參閱 AWS CLI 以取得 ElastiCache 參考主題 [create-replication-group](#)。

為 Valkey 或 Redis 建立叢集 OSS (ElastiCache API)

若要使用 建立叢集 ElastiCache API，請使用 `CreateCacheCluster` 動作。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

主題

- [建立 Valkey 或 Redis OSS \(停用叢集模式 \) 快取叢集 \(ElastiCache API \)](#)
- [在 Valkey 或 Redis 中建立快取叢集 OSS \(啟用叢集模式 \) \(ElastiCache API \)](#)

建立 Valkey 或 Redis OSS (停用叢集模式) 快取叢集 (ElastiCache API)

下列程式碼會建立 Valkey 或 Redis OSS (停用叢集模式) 快取叢集 () ElastiCache API。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &CacheParameterGroup=default.redis3.2  
  &Engine=redis  
  &EngineVersion=3.2.4  
  &NumCacheNodes=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb  
  &Timestamp=20150508T220302Z
```

```
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20150508T220302Z
&X-Amz-Expires=20150508T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Signature=<signature>
```

在 Valkey 或 Redis 中建立快取叢集 OSS (啟用叢集模式) (ElastiCache API)

無法使用 `CreateCacheCluster` 操作建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (API/CLI : 複寫群組)。若要建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (API/CLI : 複寫群組)，請參閱 [從頭開始在 Valkey 或 Redis OSS \(啟用叢集模式 \) 中建立複寫群組 \(ElastiCache API \)](#)。

如需詳細資訊，請參閱 ElastiCache API 參考主題 [CreateReplicationGroup](#)。

為 Memcached 建立叢集

下列範例示範如何使用 AWS Management Console、AWS CLI 和 建立叢集 ElastiCache API。

建立 Memcached 叢集 (主控台)

當您使用 Memcached 引擎時，Amazon ElastiCache 支援在多個節點上水平分割資料。Memcached 可啟用自動探索功能，因此您不需要追蹤每個節點的端點。Memcached 可追蹤每個節點的端點，並隨著節點新增與移除節點而更新端點清單。您的應用程式只需要組態端點即可與叢集互動。

若要建立 Memcached 叢集，請按照[建立叢集](#)的步驟執行

一旦叢集的狀態可用，您就可以授予 Amazon EC2 存取權、連線到叢集，並開始使用叢集。如需詳細資訊，請參閱 [步驟 3. 授權對叢集的存取](#) 和 [步驟 4. 連線至叢集的節點](#)。

Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立叢集 (AWS CLI)

若要使用 建立叢集 AWS CLI，請使用 `create-cache-cluster` 命令。

⚠ Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

建立 Memcached 快取叢集 (AWS CLI)

下列CLI程式碼會建立具有 3 個節點的 Memcached 快取叢集。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

針對 Windows：

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine memcached ^  
--engine-version 1.4.24 ^  
--cache-parameter-group default.memcached1.4 ^  
--num-cache-nodes 3
```

建立 Memcached 的叢集 (ElastiCache API)

若要使用 建立叢集 ElastiCache API，請使用 CreateCacheCluster 動作。

⚠ Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [在中刪除叢集 ElastiCache](#)。

主題

- [建立 Memcached 快取叢集 \(ElastiCache API\)](#)

建立 Memcached 快取叢集 (ElastiCache API)

下列程式碼會建立具有 3 個節點的 Memcached 叢集 (ElastiCache API)。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &Engine=memcached  
  &NumCacheNodes=3  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150508T220302Z  
  &Version=2015-02-02  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=<credential>  
  &X-Amz-Date=20150508T220302Z  
  &X-Amz-Expires=20150508T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Signature=<signature>
```

檢視 ElastiCache 叢集的詳細資訊

您可以使用 ElastiCache 主控台或 檢視一或多個叢集的詳細資訊 AWS CLI ElastiCache API。

檢視 Memcached 叢集的詳細資訊 (主控台)

您可以使用 ElastiCache 主控台、 AWS CLI for ElastiCache或 來檢視 Memcached 叢集的詳細資訊 ElastiCache API。

下列程序詳細說明如何使用 ElastiCache 主控台檢視 Memcached 叢集的詳細資訊。

檢視 Memcached 叢集的詳細資訊

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇您感興趣的 AWS 區域。
3. 在 ElastiCache 引擎儀表板中，選擇 Memcached 。這樣做會顯示 Memcached 引擎上執行的所有叢集清單。
4. 若要查看叢集的詳細資訊，請選擇叢集名稱左側的方塊。
5. 若要檢視節點資訊，請選擇 Nodes (節點) 標籤，此標籤會顯示節點狀態和端點的相關資訊。
6. 若要檢視指標，請選擇 Metrics (指標) 標籤，此標籤會顯示叢集中所有節點的相關指標。如需詳細資訊，請參閱 [監控與 CloudWatch 指標搭配使用](#)
7. 選擇網路和安全索引標籤，以檢視叢集的網路連線、子網路群組組態和VPC安全群組的詳細資訊。如需詳細資訊，請參閱[子網路和子網路群組](#)。
8. 選擇 Maintenance (維護) 標籤，以檢視叢集維護設定的詳細資訊。如需詳細資訊，請參閱[管理 ElastiCache 叢集維護](#)。
9. 選擇 Tags (標籤) 標籤，以檢視套用至叢集資源的任何標籤詳細資訊。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。

檢視 Valkey 或 Redis OSS (停用叢集模式) 詳細資訊 (主控台)

您可以使用 ElastiCache 主控台、 AWS CLI for 或 ElastiCache來檢視 Valkey 或 Redis OSS (停用叢集模式) 叢集的詳細資訊 ElastiCache API。

下列程序詳細說明如何使用 ElastiCache 主控台檢視 Valkey 或 Redis OSS (停用叢集模式) 叢集的詳細資訊。

檢視 Valkey 或 Redis OSS (停用叢集模式) 叢集的詳細資訊

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在 ElastiCache 引擎儀表板中，選擇 Valkey 或 Redis OSS 以顯示在該引擎上執行的所有叢集清單。
3. 要查看叢集的詳細資訊，請選擇叢集名稱左側的核取方塊。請務必選取執行 Valkey 或 Redis OSS 引擎的叢集，而非叢集 Valkey 或叢集 RedisOSS。這會顯示叢集的詳細資訊，包括叢集的主端點。
4. 檢視節點資訊：
 - a. 選擇叢集名稱。
 - b. 選擇 Shards and nodes (碎片和節點) 標籤。這會顯示各節點的詳細資訊，包括您必須用來讀取叢集的節點之端點。
5. 若要檢視指標，請選擇 Metrics (指標) 標籤，此標籤會顯示叢集中所有節點的相關指標。如需詳細資訊，請參閱 [監控與 CloudWatch 指標搭配使用](#)
6. 若要檢視日誌，請選擇 Logs (日誌) 標籤，此標籤會指出叢集是否使用慢速日誌或引擎日誌，並提供相關詳細資訊。如需詳細資訊，請參閱[日誌傳送](#)。
7. 選擇 Network and security (網路和安全性) 標籤，以檢視叢集網路連線和子網路群組組態的詳細資訊。如需詳細資訊，請參閱[子網路和子網路群組](#)。
8. 選擇 Maintenance (維護) 標籤，以檢視叢集維護設定的詳細資訊。如需詳細資訊，請參閱[管理 ElastiCache 叢集維護](#)。
9. 選擇 Service updates (服務更新) 標籤，以檢視任何可用的服務更新詳細資訊，以及其建議套用日期。如需詳細資訊，請參閱[中的服務更新 ElastiCache](#)。
10. 選擇 Tags (標籤) 標籤，以檢視套用至叢集資源的任何標籤詳細資訊。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。

檢視 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (主控台) 的詳細資訊

您可以使用 ElastiCache 主控台、AWS CLI 的 或 來檢視 Valkey 或 Redis ElastiCacheOSS (啟用叢集模式) 叢集的詳細資訊 ElastiCache API。

下列程序詳細說明如何使用 ElastiCache 主控台檢視 Valkey 或 Redis OSS (啟用叢集模式) 叢集的詳細資訊。

檢視 Valkey 或 Redis OSS (啟用叢集模式) 叢集的詳細資訊

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇您感興趣的 AWS 區域。
3. 在 ElastiCache 引擎儀表板中，選擇 Valkey 或 Redis OSS 以顯示在該引擎上執行的所有叢集清單。
4. 若要查看 Valkey 或 Redis OSS (啟用叢集模式) 叢集的詳細資訊，請選擇叢集名稱左側的方塊。請務必選擇執行 Valkey 或叢集 Redis OSS引擎的叢集。

叢集下方會展開一個畫面，並顯示叢集的詳細資訊，包括叢集的組態端點。

5. 若要查看叢集碎片的清單，以及每個碎片的節點數，請選擇 Shards and nodes (碎片與節點) 標籤。
6. 若要檢視節點的特定資訊：
 - 選擇碎片的 ID。

這麼做會顯示每個節點的資訊，包括您要用來讀取叢集資料之每個節點的端點。

7. 若要檢視指標，請選擇 Metrics (指標) 標籤，此標籤會顯示叢集中所有節點的相關指標。如需詳細資訊，請參閱 [監控與 CloudWatch 指標搭配使用](#)
8. 若要檢視日誌，請選擇 Logs (日誌) 標籤，此標籤會指出叢集是否使用慢速日誌或引擎日誌，並提供相關詳細資訊。如需詳細資訊，請參閱 [日誌傳送](#)。
9. 選擇網路和安全索引標籤，以檢視叢集的網路連線和子網路群組組態、VPC安全群組的詳細資訊，以及叢集上如果有的話，啟用哪些加密方法。如需詳細資訊，請參閱 [子網路和子網路群組](#) 和 [Amazon 中的資料安全 ElastiCache](#)。
10. 選擇 Maintenance (維護) 標籤，以檢視叢集維護設定的詳細資訊。如需詳細資訊，請參閱 [管理 ElastiCache 叢集維護](#)。
11. 選擇 Service updates (服務更新) 標籤，以檢視任何可用的服務更新詳細資訊，以及其建議套用日期。如需詳細資訊，請參閱 [中的服務更新 ElastiCache](#)。
12. 選擇 Tags (標籤) 標籤，以檢視套用至叢集資源的任何標籤詳細資訊。如需詳細資訊，請參閱 [標記您的 ElastiCache 資源](#)。

檢視 ElastiCache 叢集的詳細資訊 (AWS CLI)

下列程式碼會列出 的詳細資訊 *my-cluster*:


```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Replace (取代) *my-cluster* 使用 `create-cache-cluster` 命令建立叢集時，叢集的名稱為 1 個快取節點和 0 個碎片。

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "08:30-09:30",
      "TransitEncryptionEnabled": false,
      "AtRestEncryptionEnabled": false,
      "CacheClusterId": "my-cluster1",
      "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
      "PreferredAvailabilityZone": "us-west-2c",
      "AuthTokenEnabled": false,
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
      },
      "SnapshotRetentionLimit": 0,
      "AutoMinorVersionUpgrade": true,
      "EngineVersion": "3.2.10",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    }
  ]
}
```

```
{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": false,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": false,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": false,
      "ReplicationGroupId": "my-cluster2",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
      "CacheClusterId": "my-cluster2-001",
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x"
      },
      "SnapshotRetentionLimit": 0,
      "EngineVersion": "6.0",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    },
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ]
    }
  ]
}
```

```

    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",

```

```

    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": true,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": true,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": true,
      "ReplicationGroupId": "my-cluster3",
      "Engine": "redis",

```

```

    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2.cluster.on"
    }
  }
}

```

```
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
```

```
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
    }
],
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
"AuthTokenEnabled": true,
"CacheSubnetGroupName": "default",
"SnapshotWindow": "12:30-13:30",
"AutoMinorVersionUpgrade": true,
"CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
"CacheClusterStatus": "available",
"AtRestEncryptionEnabled": true,
"PreferredAvailabilityZone": "us-west-2b",
"TransitEncryptionEnabled": true,
"ReplicationGroupId": "my-cluster3",
"Engine": "redis",
"PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
"CacheClusterId": "my-cluster3-0002-001",
"PendingModifiedValues": {},
"CacheNodeType": "cache.r4.large",
"DataTiering": "disabled",
"CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "ParameterApplyStatus": "in-sync",
    "CacheParameterGroupName": "default.redis6.x.cluster.on"
},
"SnapshotRetentionLimit": 0,
"EngineVersion": "6.0",
"CacheSecurityGroups": [],
"NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
```

```

    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",

```



```
    "CacheClusterId": "my-cluster3-0002-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }
]
```

如果使用 建立叢集 AWS Management Console (啟用或停用具有 1 個或更多碎片的叢集節點)，請使用下列命令來描述叢集的詳細資訊 (取代 *my-cluster* 具有複寫群組名稱 (叢集名稱))：

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

如需詳細資訊，請參閱 ElastiCache 主題 AWS CLI 的 [describe-cache-clusters](#)。

檢視 ElastiCache 叢集的詳細資訊 (ElastiCache API)

您可以使用 DescribeCacheClusters 動作檢視叢集 ElastiCache API 的詳細資訊。如果包含 CacheClusterId 參數，則會傳回指定叢集的詳細資訊。如果省略 CacheClusterId 參數，則會傳回最多 MaxRecords 個叢集 (預設值為 100) 的詳細資訊。MaxRecords 的值不可小於 20 或大於 100。

以下程式碼會列出 my-cluster 的詳細資訊。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterId=my-cluster
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

以下程式碼清單會列出最多 25 個叢集的詳細資訊。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 ElastiCache API 參考主題 [DescribeCacheClusters](#)。

修改 ElastiCache 叢集

除了從 ElastiCache 叢集新增或移除節點之外，有時您可能需要進行其他變更，例如新增安全群組、變更維護時段或參數群組。

建議您將維護時段落在使用量最低的時段。您可能需要不時進行調整。

當變更叢集的參數時，變更即會立即套用至叢集，或在叢集重新啟動之後套用。無論是變更叢集的參數群組本身或是叢集的參數群組內的參數值，均適用此情況。若要判斷何時套用特定參數變更，請參閱 [Memcached 專用參數](#) 和 資料表中詳細資訊欄的變更生效一節 [Valkey 和 Redis OSS 參數](#)。如需重新啟動叢集節點的資訊，請參閱 [重新啟動節點](#)。

使用 ElastiCache AWS Management Console

修改叢集

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇您要修改的叢集所在的 AWS 區域。
3. 在導覽窗格中，選擇您要修改之叢集所執行的引擎。

接著會出現所選引擎的叢集清單。

4. 在叢集清單中，針對您要修改的叢集，選擇其名稱。
5. 選擇 Actions (動作)，然後選擇 Modify (修改)。

Modify Cluster (修改叢集) 視窗隨即出現。

6. 在 Modify Cluster (修改叢集) 視窗中，進行所需的修改。選項包括：

- 描述
- 叢集模式 – 若要將叢集模式從停用修改為啟用，您必須先將叢集模式設定為相容。

相容模式可讓您的 Valkey 或 Redis OSS 用戶端使用啟用叢集模式和停用叢集模式進行連線。遷移所有 Valkey 或 Redis OSS 用戶端以使用已啟用的叢集模式後，您就可以完成叢集模式組態，並將叢集模式設定為已啟用。

- 引擎版本相容性

⚠ Important

您可以升級到較新的引擎版本。如果您升級主要引擎版本 (例如從 5.0.6 升級到 6.0)，則需要選取與新引擎版本相容的參數群組系列。如需這項作業的詳細資訊，請參閱[的版本管理 ElastiCache](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

- VPC 安全群組 (s)
- 參數群組
- 節點類型

ℹ Note

如果叢集使用 r6gd 系列中的節點類型，則只能從該系列中選擇不同的節點大小。如果從 r6gd 系列中選擇節點類型，則將自動啟用資料分層。如需詳細資訊，請參閱[資料分層](#)。

- Multi-AZ
- 自動容錯移轉 (僅適用於停用叢集模式)
- 啟用自動備份
- 備份節點 ID
- 備份保留期間
- 備份時段
- SNS 通知的主題

- Memcached Engine 版本相容性
- Network type (網路類型)

ℹ Note

如果您要從 IPv4 切換至 IPv6，則必須選取或建立與 相容的子網路群組 IPv6。如需詳細資訊，請參閱[在中選擇網路類型 ElastiCache](#)。

- VPC 安全群組 (s)
- 參數群組

- 維護時段
- SNS 通知的主題

Apply Immediately (立即套用) 方塊僅適用於引擎版本的修改。若要立即套用變更，請選擇 Apply Immediately (立即套用) 核取方塊。如果未選擇此方塊，引擎版本的修改將於下次維護時段套用。其他修改 (例如變更維護時段) 則會立即套用。

啟用/停用 Redis 的日誌交付

1. 從叢集清單中選擇您要修改的叢集。請選擇叢集名稱而不是名稱旁邊的核取方塊。
2. 在叢集詳細資訊頁面上，選擇日誌索引標籤。
3. 若要啟用或停用慢速日誌，請選擇啟用或停用。

如果您選擇 Enable (啟用)：

- a. 在日誌格式下，選擇 JSON 或文字。
- b. 在日誌目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
- c. 在日誌目的地下，您可以選擇建立新，並輸入您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。您也可以選擇選取現有，然後選擇您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。
- d. 選擇 啟用。

若要變更 Redis 的組態：

1. 選擇 Modify (修改)。
2. 在日誌格式下，選擇 JSON 或文字。
3. 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
4. 在日誌目的地下，選擇建立新 並輸入您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。或選擇選取現有，然後選擇您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。

AWS CLI 搭配 使用 ElastiCache

您可以使用 AWS CLI `modify-cache-cluster` 操作修改現有叢集。若要修改叢集的組態值，請指定叢集 ID、要變更的參數以及參數的新值。下方範例會變更名稱為 `my-cluster` 之叢集的維護時段，並立即套用變更。

Important

您可以升級至較新的 Memcached 引擎版本。如需這項作業的詳細資訊，請參閱[的版本管理 ElastiCache](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

Important

您可以升級至較新的 Valkey 或 Redis OSS 引擎版本。如果您將主要引擎版本升級，例如從 Redis OSS 5.0.6 升級至 Redis OSS 6.0，則需要選取與新引擎版本相容的參數群組系列。如需這項作業的詳細資訊，請參閱[的版本管理 ElastiCache](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

`--apply-immediately` 參數僅適用於節點類型、引擎版本和叢集節點數的修改。如果您要立即套用其中任何變更，請使用 `--apply-immediately` 參數。如果您想將這些變更延期至下一個維護時段，請使用 `--no-apply-immediately` 參數。其他修改 (例如變更維護時段) 則會立即套用。

如需詳細資訊，請參閱適用於 ElastiCache 主題 AWS CLI 的 [modify-cache-cluster](#)。

使用 ElastiCache API

您可以使用 ElastiCache API `ModifyCacheCluster` 操作修改現有叢集。若要修改叢集的組態值，請指定叢集 ID、要變更的參數以及參數的新值。下方範例會變更名稱為 `my-cluster` 之叢集的維護時段，並立即套用變更。

Important

您可以升級至較新的 Memcached 引擎版本。如需這項作業的詳細資訊，請參閱[版本管理 ElastiCache](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

Important

您可以升級至較新的 Valkey 或 Redis OSS 引擎版本。如果您將主要引擎版本升級，例如從 Redis OSS 5.0.6 升級至 Redis OSS 6.0，則需要選取與新引擎版本相容的參數群組系列。如需這項作業的詳細資訊，請參閱[版本管理 ElastiCache](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&PreferredMaintenanceWindow=sun:23:00-mon:02:00  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150901T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150901T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

`ApplyImmediately` 參數僅適用於節點類型、引擎版本和叢集節點數的修改。如果您要立即套用其中任何變更，請將 `ApplyImmediately` 參數設為 `true`。如果您想將這些變更延期至下一個維護時段，請將 `ApplyImmediately` 參數設為 `false`。其他修改 (例如變更維護時段) 則會立即套用。

如需詳細資訊，請參閱 ElastiCache API 參考主題 [ModifyCacheCluster](#)。

將節點新增至 ElastiCache 叢集

將節點新增至 Memcached 叢集時，會增加叢集的分割區數。當您變更叢集中的分割區數量時，您必須重新對應一些鍵空間，以讓它們對應到正確的節點。重新對應鍵空間時，會暫時增加叢集的快取未命中數。如需詳細資訊，請參閱[設定您的 ElastiCache 用戶端以進行有效的負載平衡 \(Memcached \)](#)。

若要重新設定您的 Valkey 或 Redis OSS (啟用叢集模式) 叢集，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \(已啟用叢集模式 \)](#)

您可以使用 ElastiCache 管理主控台、AWS CLI 或 ElastiCache API 將節點新增至叢集。

使用 ElastiCache AWS Management Console

如果您想要將節點新增至單一節點 Valkey 或 Redis OSS (停用叢集模式) 叢集 (未啟用複寫)，則這是一個兩步驟程序：首先新增複寫，然後新增複本節點。

主題

- [若要將複寫新增至沒有碎片的 Valkey 或 Redis OSS叢集](#)
- [將節點新增至 ElastiCache 叢集 \(主控台 \)](#)

下列程序會將複寫新增至未啟用複寫OSS的單節點 Valkey 或 Redis。當您新增複寫時，現有節點會變成已啟用複寫之叢集中的主要節點。新增複寫之後，您即可新增最多 5 個複本節點至該叢集。

若要將複寫新增至沒有碎片的 Valkey 或 Redis OSS叢集

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。

隨即顯示執行該引擎的叢集清單。

3. 選擇您要為其新增節點的叢集名稱，而非叢集名稱左側的方塊。

對於未啟用複寫的 Redis OSS叢集，下列為真：

- 它正在執行 Redis OSS，而非叢集 Redis OSS。
- 它具有零個碎片。

如果叢集有任何碎片，該叢集就已經啟用複寫，因此您可以繼續 [將節點新增至 ElastiCache 叢集 \(主控台 \)](#)。

4. 選擇 Add replication (新增複寫)。
5. 在 Add Replication (新增複寫) 中，為這個已啟用複寫的叢集輸入說明。
6. 選擇新增。

只要叢集的狀態回到「可用」，您就可以繼續下一個程序並新增複本至叢集。

將節點新增至 ElastiCache 叢集 (主控台)

下列程序可用來將節點新增至叢集。

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇您要為其新增節點之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

3. 從叢集清單中，針對您要新增節點的叢集，選擇其名稱。

如果您的叢集是 Valkey 或 Redis OSS (啟用叢集模式) 叢集，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \(已啟用叢集模式\)](#)。

如果您的叢集是具有零碎片的 Valkey 或 Redis OSS (停用叢集模式) 叢集，請先完成的步驟 [若](#) [要將複寫新增至沒有碎片的 Valkey 或 Redis OSS 叢集](#)。

4. 選擇 Add node (新增節點)。
5. 完成 Add Node (新增節點) 對話方塊中請求的資訊。
6. 選擇 Apply Immediately - Yes (立即套用 - 是) 按鈕可立即新增此節點；或選擇 No (否)，在此叢集的下一個維護時段期間，新增此節點。

新增和移除要求對特定請求的影響

案例	待定作業	新請求	結果
案例 1	Delete	Delete	新的刪除請求 (待定或立即) 會取代待定的刪除請求。 例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出刪除節點 0002 和 0004 的新請求，則只會刪除節點 0002 和 0004。系統不會刪除節點 0001、0003 和 0007。

案例	待定作業	新請求	結果
案例 2	Delete	建立	<p>新的建立請求 (待定或立即) 會取代待定的刪除請求。</p> <p>例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出建立節點的新請求，則會建立新的節點，而不會刪除節點 0001、0003 和 0007。</p>
案例 3	建立	Delete	<p>新的刪除請求 (待定或立即) 會取代待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出刪除節點 0003 的新請求，則不會建立新的節點，且會刪除節點 0003。</p>
案例 4	建立	建立	<p>新的建立請求會新增至待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出建立三個節點的新請求，則新請求會新增至待定請求，並建立五個節點。</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>如果新的建立請求設定為 Apply Immediately - Yes (立即套用 - 是)，則會立即執行所有建立請求。如果新的建立請求設定為 Apply Immediately - No (立即套用 - 否)，則所有建立請求都處於待定狀態。</p> </div>

若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。

7. 選擇 Add (新增) 按鈕。

在一段時間之後，新的節點應會顯示在節點清單中，且狀態為 creating (建立中)。如果並未顯示，請重新整理瀏覽器頁面。如果節點狀態變更為「可用」，新的節點即可供使用。

AWS CLI 搭配 使用 ElastiCache

若要使用 將節點新增至叢集 AWS CLI，請使用 AWS CLI 操作 `modify-cache-cluster` 搭配下列參數：

- `--cache-cluster-id` 您想為其新增節點的快取叢集 ID。
- `--num-cache-nodes` `--num-cache-nodes` 參數可指定套用修改之後您要在此叢集中使用的節點數。若要新增節點至此叢集，`--num-cache-nodes` 必須大於此叢集中的目前節點數。如果此值小於目前的節點數量，ElastiCache 會預期要從叢集中移除的參數 `cache-node-ids-to-remove` 和節點清單。如需詳細資訊，請參閱 [AWS CLI 搭配 使用 ElastiCache](#)。
- `--apply-immediately` 或 `--no-apply-immediately` 指定是否要立即新增這些節點，或等到下個維護時段再新增。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

此操作會產生類似下列（JSON 格式）的輸出：

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-west-2b",
```

```
    "ConfigurationEndpoint": {
      "Port": 11211,
      "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
      "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
  }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [modify-cache-cluster](#)。

AWS CLI 搭配 使用 ElastiCache

如果您想要將節點新增至未啟用複寫的現有 Valkey 或 Redis OSS (停用叢集模式) 叢集，您必須先建立將現有叢集指定為主要叢集的複寫群組。如需詳細資訊，請參閱[使用可用的 Valkey 或 Redis OSS 快取叢集建立複寫群組 \(AWS CLI \)](#)。複寫群組為「可用」之後，您即可繼續下列程序。

若要使用 將節點新增至叢集 AWS CLI，請使用 AWS CLI 操作 `increase-replica-count` 搭配下列參數：

- `--replication-group-id` 您想為其新增節點的複寫群組 ID。
- `--new-replica-count` 可指定套用修改之後您要在此複寫群組中使用的節點數。若要新增節點至此叢集，`--new-replica-count` 必須大於此叢集中的目前節點數。

- `--apply-immediately` 或 `--no-apply-immediately` 指定是否要立即新增這些節點，或等到下個維護時段再新增。

若為 Linux、macOS 或 Unix：

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

針對 Windows：

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

此操作會產生類似下列（JSON 格式）的輸出：

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "node-test-001",  
      "node-test-002",  
      "node-test-003",  
      "node-test-004",  
      "node-test-005"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
```

```
        "Port": 6379
    },
    "NodeGroupMembers": [
        {
            "CacheClusterId": "node-test-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "primary"
        },
        {
            "CacheClusterId": "node-test-002",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2c",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "node-test-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2b",
            "CurrentRole": "replica"
        }
    ]
},
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
```

```
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r5.large",
    "DataTiering": "disabled",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
  }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [increase-replica-count](#)。

使用 ElastiCache API

如果您想要將節點新增至未啟用複寫的現有 Valkey 或 Redis OSS (停用叢集模式) 叢集，您必須先建立將現有叢集指定為主要叢集的複寫群組。如需詳細資訊，請參閱[將複本新增至獨立 Valkey 或 Redis OSS \(停用叢集模式 \) 叢集 \(ElastiCache API \)](#)。複寫群組為「可用」之後，您即可繼續下列程序。

將節點新增至叢集 (ElastiCache API)

- 使用下列參數呼叫 IncreaseReplicaCountAPI 操作：
 - ReplicationGroupId 您想為其新增節點的叢集 ID。
 - NewReplicaCount NewReplicaCount 參數可指定套用修改之後您要在此叢集中使用的節點數。若要新增節點至此叢集，NewReplicaCount 必須大於此叢集中的目前節點數。如果此值小於目前的節點數量，請使用 DecreaseReplicaCountAPI 搭配要從叢集中移除的節點數量。
 - ApplyImmediately 指定是否要立即新增這些節點，或等到下個維護時段再新增。
 - Region 指定您要新增節點的叢集 AWS 區域。

下列範例說明可將節點新增至叢集的呼叫。

Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=IncreaseReplicaCount
&ApplyImmediately=true
&NumCacheNodes=4
&ReplicationGroupId=my-replication-group
&Region=us-east-2
&Version=2014-12-01
```



```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API 主題 [IncreaseReplicaCount](#)。

使用 ElastiCache API

將節點新增至叢集 (ElastiCache API)

- 使用下列參數呼叫 ModifyCacheCluster API 操作：
 - CacheClusterId 您想為其新增節點的叢集 ID。
 - NumCacheNodes NumCachNodes 參數可指定套用修改之後您要在此叢集中使用的節點數。若要新增節點至此叢集，NumCacheNodes 必須大於此叢集中的目前節點數。如果此值小於目前的節點數量，ElastiCache 會預期參數 CacheNodeIdsToRemove 與要從叢集中移除的節點清單（請參閱 [搭配 Memcached ElastiCache API 使用](#)）。
 - ApplyImmediately 指定是否要立即新增這些節點，或等到下個維護時段再新增。
 - Region 指定您要新增節點的叢集 AWS 區域。

下列範例說明可將節點新增至叢集的呼叫。

Example

```
https://elasticache.us-west-2.amazonaws.com/
  ?Action=ModifyCacheCluster
  &ApplyImmediately=true
  &NumCacheNodes=5
  &CacheClusterId=my-cluster
  &Region=us-east-2
  &Version=2014-12-01
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20141201T220302Z
```

```
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API 主題 [ModifyCacheCluster](#)。

從 ElastiCache 叢集移除節點

您可以使用 OSS、或 [Valkey](#)、Redis AWS Management Console AWS CLI或 Memcached 叢集中刪除節點 ElastiCache API。

Note

每當您變更 Memcached 叢集中的節點數量時，必須至少重新對應一些索引鍵空間，使其對應到正確的節點。如需有關負載平衡 Memcached 叢集的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以進行有效的負載平衡 \(Memcached \)](#)。

使用 ElastiCache AWS Management Console

從叢集中移除節點 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇要從中移除節點的叢集 AWS 區域。
3. 在導覽窗格中，選擇您要從中移除節點之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

4. 在叢集清單中，選擇您要從中移除節點的叢集名稱。

接著會出現叢集的節點清單。

5. 選擇您要移除之節點的節點 ID 旁的方塊。使用 ElastiCache 主控台一次只能刪除一個節點，因此選擇多個節點表示無法使用刪除節點按鈕。

Delete Node (刪除節點) 頁面隨即出現。

6. 若要刪除節點，請完成 Delete Node (刪除節點) 頁面，然後選擇 Delete Node (刪除節點)。若要保留節點，請選擇 Cancel (取消)。

Important

使用 Valkey 或 Redis OSS，如果您刪除叢集中不再符合多可用區域標準的節點結果，請務必先清除多可用區域核取方塊，然後刪除節點。如果清除 Multi-AZ (多個可用區) 核取方塊，您可以選擇啟用 Auto failover (自動容錯移轉)。

新增和移除請求對待定請求的影響

案例	待定作業	新請求	結果
案例 1	Delete	Delete	<p>新的刪除請求 (待定或立即) 會取代待定的刪除請求。</p> <p>例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出刪除節點 0002 和 0004 的新請求，則只會刪除節點 0002 和 0004。系統不會刪除節點 0001、0003 和 0007。</p>
案例 2	Delete	建立	<p>新的建立請求 (待定或立即) 會取代待定的刪除請求。</p> <p>例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出建立節點的新請求，則會建立新的節點，而不會刪除節點 0001、0003 和 0007。</p>
案例 3	建立	Delete	<p>新的刪除請求 (待定或立即) 會取代待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出刪除節點 0003 的新請求，則不會建立新的節點，且會刪除節點 0003。</p>
案例 4	建立	建立	<p>新的建立請求會新增至待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出建立三個節點的新請求，則新請求會新增至待定請求，並建立五個節點。</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>如果新的建立請求設定為 Apply Immediately - Yes (立即套用 - 是)，則會立即執行所有建立請求。如果新的建立請求設定為 Apply Immediately - No (立即套用 - 否)，則所有建立請求都處於待定狀態。</p> </div>

若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。

AWS CLI 搭配 使用 ElastiCache

1. 識別您要移除IDs的節點。如需詳細資訊，請參閱[檢視 ElastiCache 叢集的詳細資訊](#)。
2. 使用 `decrease-replica-count` CLI 操作搭配要移除的節點清單，如下列範例所示。

若要使用命令列界面移除叢集中的節點，請搭配使用 `decrease-replica-count` 命令與下列參數：

- `--replication-group-id` 您想要從中移除節點的快取複寫群組 ID。
- `--new-replica-count` 參數可指定套用修改之後您要在此叢集中使用的節點數。
- `--replicas-to-remove` IDs 您要從此叢集中移除的節點清單。
- `--apply-immediately` 或 `--no-apply-immediately` 指定是否要立即移除這些節點，或等到下個維護時段再移除。
- `--region` 指定您要從中移除節點的叢集 AWS 區域。

Note

呼叫此作業時，您只能傳遞其中一個 `--replicas-to-remove` 或 `--new-replica-count` 參數。

若為 Linux、macOS 或 Unix：

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

針對 Windows：

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^
```

```
--region us-east-2 ^  
--apply-immediately
```

此操作會產生類似下列（JSON 格式）的輸出：

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test"  
  },  
  "Status": "modifying",  
  "PendingModifiedValues": {},  
  "MemberClusters": [  
    "node-test-001",  
    "node-test-002",  
    "node-test-003",  
    "node-test-004",  
    "node-test-005",  
    "node-test-006"  
  ],  
  "NodeGroups": [  
    {  
      "NodeGroupId": "0001",  
      "Status": "modifying",  
      "PrimaryEndpoint": {  
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",  
        "Port": 6379  
      },  
      "ReaderEndpoint": {  
        "Address": "node-test-  
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",  
        "Port": 6379  
      },  
      "NodeGroupMembers": [  
        {  
          "CacheClusterId": "node-test-001",  
          "CacheNodeId": "0001",  
          "ReadEndpoint": {  
            "Address": "node-  
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",  
            "Port": 6379  
          },  
          "PreferredAvailabilityZone": "us-west-2a",
```

```
        "CurrentRole": "primary"
    },
    {
        "CacheClusterId": "node-test-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-004",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-005",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
```

```

        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-006",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
"DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
}
}

```

或者，您可以呼叫 `decrease-replica-count`，不傳入 `--new-replica-count` 參數，而是可以傳遞 `--replicas-to-remove` 參數，如下所示：

若為 Linux、macOS 或 Unix：

```

aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately

```


針對 Windows :

```
aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^
  --region us-east-2 ^
  --apply-immediately
```

如需詳細資訊，請參閱 AWS CLI 主題 [decrease-replica-count](#)。

搭配 Valkey ElastiCache API 或 Redis 使用 OSS

若要使用 移除節點 ElastiCache API，請使用複寫群組 ID 和要移除的節點清單呼叫 DecreaseReplicaCount API 操作，如下所示：

- `ReplicationGroupId` 您想要從中移除節點的快取複寫群組 ID。
- `ReplicasToRemove` `ReplicasToRemove` 參數可指定套用修改之後您要在此叢集中使用的節點數。
- `ApplyImmediately` 指定是否要立即移除這些節點，或等到下個維護時段再移除。
- `Region` 指定您要從中移除節點的叢集 AWS 區域。

下列範例會從 `my-cluster` 叢集中立即移除節點 0004 和 0005。

```
https://elasticache.us-west-2.amazonaws.com/
  ?Action=DecreaseReplicaCount
  &ReplicationGroupId=my-replication-group
  &ApplyImmediately=true
  &ReplicasToRemove=node-test-003
  &Region us-east-2
  &Version=2014-12-01
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20141201T220302Z
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
  &X-Amz-Date=20141201T220302Z
  &X-Amz-SignedHeaders=Host
  &X-Amz-Expires=20141201T220302Z
  &X-Amz-Credential=<credential>
  &X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API主題 [DecreaseReplicaCount](#)。

搭配 Memcached ElastiCache API使用

若要使用 移除節點 ElastiCache API，請使用快取叢集 ID 和要移除的節點清單呼叫 `ModifyCacheCluster` API操作，如下所示：

- `CacheClusterId` 您想要從中移除節點的快取叢集 ID。
- `NumCacheNodes` `NumCacheNodes` 參數可指定套用修改之後您要在此叢集中使用的節點數。
- `CacheNodeIdsToRemove.member.n` 從叢集IDs中移除的節點清單。
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` 指定是否要立即移除這些節點，或等到下個維護時段再移除。
- `Region` 指定您要從中移除節點的叢集 AWS 區域。

下列範例會從 `my-cluster` 叢集中立即移除節點 `0004` 和 `0005`。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&ApplyImmediately=true  
&CacheNodeIdsToRemove.member.1=0004  
&CacheNodeIdsToRemove.member.2=0005  
&NumCacheNodes=3  
&Region us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API主題 [ModifyCacheCluster](#)。

取消 中的待定新增或刪除節點操作 ElastiCache

如果您選擇不立即套用 ElastiCache 叢集變更，操作會處於擱置狀態，直到下次維護時段執行為止。您可以取消任何等待中的操作。

取消等待中的操作

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇要取消待定新增或刪除節點操作 AWS 的區域。
3. 在導覽窗格中，選擇含有您要取消之等待中操作的叢集所執行的引擎。接著會出現執行所選引擎的叢集清單。
4. 在叢集清單中，選擇含有您要取消之等待中操作的叢集名稱，而非叢集名稱左側的方塊。
5. 若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。
6. 選擇 Nodes (節點) 標籤。
7. 若要取消所有等待中的操作，請選擇 Cancel Pending (取消待定)。隨即顯示 Cancel Pending (取消待定) 對話方塊。
8. 您可以選擇 Cancel Pending (取消待定) 按鈕以取消所有等待中的操作，或選擇 Cancel (取消) 來保留操作。

在中刪除叢集 ElastiCache

只要 ElastiCache 叢集處於可用狀態，無論您是否正在使用它，都會向您收取費用。若要停止產生費用，請刪除叢集。

Warning

刪除 ElastiCache 叢集時，會保留手動快照。您也可以在中刪除叢集之前建立最終快照。自動快照則不會保留。

使用 AWS Management Console

以下程序會從您的部署中刪除單一叢集。若要刪除多個叢集，請針對每個要刪除的叢集重複此程序。您不需要等待某個叢集完成刪除，即可開始刪除其他叢集。

刪除叢集

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在 ElastiCache 引擎儀表板中，選擇您要刪除的叢集正在執行的引擎。

接著會出現執行該引擎的所有叢集清單。

3. 要選擇欲刪除的叢集，請從叢集清單選擇叢集的名稱。

Important

您一次只能從 ElastiCache 主控台刪除一個叢集。選擇多個叢集會停用刪除操作。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 在 Delete Cluster (刪除叢集) 確認畫面中，選擇 Delete (刪除) 以刪除叢集，或選擇 Cancel (取消) 來保留叢集。

如果您選擇 Delete (刪除)，叢集的狀態就會變更為 deleting (正在刪除)。

一旦您的叢集不再列於叢集清單，您就不會再因此產生費用。

使用 AWS CLI 刪除 ElastiCache 叢集

下列程式碼會刪除 ElastiCache 快取叢集 `my-cluster`。

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

此 `delete-cache-cluster` CLI 動作只會刪除一個快取叢集。若要刪除多個快取叢集，請針對每個要刪除的快取叢集呼叫 `delete-cache-cluster`。您不需要等待某個快取叢集完成刪除，即可刪除其他叢集。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

針對 Windows：

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

如需詳細資訊，請參閱 ElastiCache 主題 AWS CLI 的 [delete-cache-cluster](#)。

使用 ElastiCache API

下列程式碼會刪除 `my-cluster` 叢集。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheCluster  
&CacheClusterId=my-cluster  
&Region us-east-2  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150202T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

DeleteCacheCluster API 操作只會刪除一個快取叢集。若要刪除多個快取叢集，請針對每個要刪除的快取叢集呼叫 DeleteCacheCluster。您不需要等待某個快取叢集完成刪除，即可刪除其他叢集。

如需詳細資訊，請參閱 ElastiCache API 參考主題 [DeleteCacheCluster](#)。

存取 ElastiCache 叢集或複寫群組

您的 Amazon ElastiCache 執行個體旨在透過 Amazon EC2執行個體存取。

如果您在 Amazon Virtual Private Cloud (AmazonVPC) 中啟動 ElastiCache 執行個體，您可以從相同 Amazon 中的 Amazon EC2執行個體存取執行個體 ElastiCacheVPC。或者，使用VPC對等，您可以從不同 Amazon EC2中的 Amazon 存取 ElastiCache 執行個體VPC。

如果您在 EC2 Classic 中啟動 ElastiCache 執行個體，則允許EC2執行個體透過授予與執行個體相關聯的 Amazon EC2安全群組存取快取安全群組來存取叢集。根據預設，僅有啟動叢集的帳戶才能存取該叢集。

主題

- [授權存取您的叢集或複寫群組](#)

授權存取您的叢集或複寫群組

您已將叢集啟動至 EC2-VPC

如果您將叢集啟動至 Amazon Virtual Private Cloud (Amazon VPC)，您只能從在相同 Amazon 中執行的 Amazon EC2執行個體連線至叢集 ElastiCacheVPC。在此情況下，您需要授權透過網路輸入至叢集。

Note

如果您有使用 Local Zones，請確定已啟用。如需詳細資訊，請參閱[啟用 Local Zones](#)。如此一來，您的 VPC 就會延伸到該 Local Zone，您的 VPC會將子網路視為任何其他可用區域和相關閘道、路由表和其他安全群組考量中的任何子網路。會自動調整。

將網路輸入從 Amazon VPC安全群組授予叢集

1. 登入 AWS Management Console 並在 開啟 Amazon EC2主控台<https://console.aws.amazon.com/ec2/>。
2. 在導覽窗格中，在 Network & Security (網路與安全) 下，選擇 Security Groups (安全群組)。
3. 從安全群組清單中，選擇 Amazon 的安全群組VPC。除非您建立安全群組以供 ElastiCache 使用，否則此安全群組將命名為預設。
4. 選擇 Inbound (傳入) 標籤，然後執行下列動作：

- a. 選擇編輯。
- b. 選擇新增規則。
- c. 在類型欄中，選擇自訂TCP規則。
- d. 在 Port range (連接埠範圍) 方塊中，輸入要用於叢集節點的連接埠號碼。此號碼必須與您啟動叢集時指定的號碼相同。Memcached 的預設連接埠是 **11211** Valkey 的預設連接埠，而 Redis OSS是 **6379**。
- e. 在來源方塊中，選擇具有連接埠範圍 (0.0.0.0/0) 的任何地方，以便您在 Amazon 中啟動的任何 Amazon EC2執行個體VPC都可以連線至 ElastiCache 節點。

 Important

將 ElastiCache 叢集開啟至 0.0.0.0/0 不會讓叢集暴露至網際網路，因為它沒有公有 IP 地址，因此無法從 外部存取VPC。不過，預設安全群組可以套用至客戶帳戶中的其他 Amazon EC2執行個體，而且這些執行個體可能具有公有 IP 地址。如果他們正巧在預設連接埠上執行某些項目，就可能會意外公開該服務。因此，我們建議您建立僅供 使用VPC的安全群組 ElastiCache。如需詳細資訊，請參閱[自訂安全群組](#)。

- f. 選擇 Save (儲存)。

當您在 Amazon 中啟動 Amazon EC2執行個體時VPC，該執行個體將能夠連線至 ElastiCache 叢集。

從外部存取 ElastiCache 資源 AWS

Amazon ElastiCache 是一種提供雲端記憶體內金鑰值存放區的 AWS 服務。該服務的設計僅限於從 內存取 AWS。不過，如果 ElastiCache 叢集託管在 內VPC，您可以使用 Network Address Translation (NAT) 執行個體來提供外部存取。

要求

您必須滿足下列要求，才能從外部存取您的 ElastiCache 資源 AWS：

- 叢集必須位於 內，VPC並透過 Network Address Translation (NAT) 執行個體存取。此需求沒有任何例外。
- NAT 執行個體必須在與叢集VPC相同的 中啟動。
- NAT 執行個體必須在與叢集分開的公有子網路中啟動。
- 彈性 IP 地址 (EIP) 必須與該NAT執行個體相關聯。iptables 的連接埠轉送功能用於將NAT執行個體上的連接埠轉送至 內的快取節點連接埠VPC。

考量事項

從外部存取 ElastiCache 資源時，請注意下列考量事項 ElastiCache。

- 用戶端會連線至NAT執行個體的 EIP和快取連接埠。NAT 執行個體上的連接埠轉送會將流量轉送至適當的快取叢集節點。
- 若新增或取代叢集節點，iptables 規則便需要更新以反映此變更。

限制

此方法應僅用於測試及開發用途。基於下列限制，不建議用於生產用途：

- NAT 執行個體正在用戶端和多個叢集之間充當代理。增加代理會影響快取叢集的效能。影響會隨著您透過NAT執行個體存取的快取叢集數量而增加。
- 從用戶端到NAT執行個體的流量未加密。因此，您應避免透過NAT執行個體傳送敏感資料。
- NAT 執行個體會新增維護另一個執行個體的額外負荷。
- NAT 執行個體做為單一失敗點。如需如何在 NAT上設定高可用性的詳細資訊VPC，請參閱 [Amazon VPCNAT執行個體的高可用性：範例](#)。

如何從外部存取 ElastiCache 資源 AWS

下列程序示範如何使用NAT執行個體連線至您的 ElastiCache 資源。

這些步驟假設下列事項：

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

接下來，您需要NAT相反方向：

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

您也需要啟用 IP 轉送 (預設為停用)：

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf  
sudo sysctl --system
```

- 您是透過以下設定存取 Memcached 叢集：
 - IP 地址 - 10.0.1.230
 - 預設 Memcached 連接埠 - 11211。
 - 安全群組 - *10\0\0\55*
- 您正在使用下列方式存取 Valkey 或 Redis OSS叢集：
 - IP 地址 - 10.0.1.230
 - 預設連接埠 – 6379
 - 安全群組 - sg-bd56b7da
 - AWS 執行個體 IP 地址 – sg-bd56b7da
- 您信任的用戶端 IP 地址為 198.51.100.27。
- 您的NAT執行個體具有彈性 IP 地址 203.0.113.73。
- 您的NAT執行個體具有安全群組 sg-ce56b7a9。

使用NAT執行個體連線至您的 ElastiCache 資源

1. 在與VPC快取叢集相同的 中建立NAT執行個體，但在公有子網路中建立執行個體。

根據預設，VPC精靈會啟動 cache.m1.small 節點類型。建議您根據需求選取節點大小。您必須使用 EC2 NAT AMI 才能 ElastiCache 從外部存取 AWS。

如需建立NAT執行個體的相關資訊，請參閱 使用者指南中的 [AWS VPC NAT執行個體](#)。

2. 建立快取叢集和NAT執行個體的安全群組規則。

NAT 執行個體安全群組和叢集執行個體應具有下列規則：

- 兩個傳入規則
 - 使用 Memcached 時，第一個規則是允許TCP從受信任用戶端連線至從NAT執行個體轉送的每個快取連接埠（11211 - 11213）。
 - 使用 Valkey 和 Redis OSS，第一個規則是允許TCP從受信任用戶端連線至從NAT執行個體轉送的每個快取連接埠（6379 - 6381）。
 - 允許SSH存取信任用戶端的第二個規則。

NAT 執行個體安全群組 - 使用 Memcached 的傳入規則

Type	通訊協定	連接埠範圍	來源
自訂TCP規則	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

NAT 執行個體安全群組 - 使用 Valkey 或 Redis 的傳入規則 OSS

Type	通訊協定	連接埠範圍	來源
自訂TCP規則	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- 使用 Memcached 時，允許TCP連線至快取連接埠（11211）的傳出規則。

NAT 執行個體安全群組 - 傳出規則

Type	通訊協定	連接埠範圍	目的地
自訂TCP規則	TCP	11211	sg-ce56b7a9 (叢集執行個體安全群組)

- 使用 Valkey 或 Redis 時OSS，允許TCP連線至快取連接埠的傳出規則（6379）。

NAT 執行個體安全群組 - 傳出規則

Type	通訊協定	連接埠範圍	目的地
自訂TCP規則	TCP	6379	sg-ce56b7a9 (叢集執行個體安全群組)

- 使用 Memcached 時，叢集安全群組的傳入規則，允許從NAT執行個體TCP連線到快取連接埠（11211）。

叢集執行個體安全群組 - 傳入規則

Type	通訊協定	連接埠範圍	來源
自訂TCP規則	TCP	11211	sg-bd56b7da (NAT安全群組)

- 使用 Valkey 或 Redis OSS，這是叢集安全群組的傳入規則，允許從NAT執行個體TCP連線到快取連接埠（6379）。

叢集執行個體安全群組 - 傳入規則

Type	通訊協定	連接埠範圍	來源
自訂TCP規則	TCP	6379	sg-bd56b7da (叢集安全群組)

3. 驗證規則。

- 確認受信任的用戶端能夠SSH存取NAT執行個體。
- 確認受信任的用戶端能夠從NAT執行個體連線至叢集。

4. Memcached

將 iptables 規則新增至 NAT 執行個體。

必須將 iptables 規則新增至叢集中每個節點的 NAT 資料表，以將快取連接埠從 NAT 執行個體轉送至叢集節點。範例看起來與以下內容相似：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
```

叢集中每個節點的連接埠號碼必須是唯一的。例如，若搭配連接埠 11211 到 11213 使用一個三節點 Memcached 叢集，規則看起來會與以下內容相似：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to
10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to
10.0.1.232:11211
```

確認信任的用戶端能夠連線到叢集。

信任的用戶端應連線至與 NAT 執行個體 EIP 相關聯的，以及對應至適當叢集節點的叢集連接埠。例如，的連線字串 PHP 可能如下所示：

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

telnet 用戶端也能用於驗證連線。例如：

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey 或 Redis OSS

將 iptables 規則新增至 NAT 執行個體。

必須將 iptables 規則新增至叢集中每個節點的 NAT 資料表，以將快取連接埠從 NAT 執行個體轉送至叢集節點。範例看起來與以下內容相似：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
```

叢集中每個節點的連接埠號碼必須是唯一的。例如，如果使用連接埠 6379 - 6381 使用三個節點 Redis OSS叢集，則規則會如下所示：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to
10.0.1.231:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to
10.0.1.232:6379
```

確認信任的用戶端能夠連線到叢集。

信任的用戶端應連線至與NAT執行個體EIP相關聯的，以及對應至適當叢集節點的叢集連接埠。例如，的連線字串PHP可能如下所示：

```
redis->connect( '203.0.113.73', 6379 );
redis->connect( '203.0.113.73', 6380 );
redis->connect( '203.0.113.73', 6381 );
```

telnet 用戶端也能用於驗證連線。例如：

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

5. 儲存 iptable 組態。

在測試及驗證過後儲存規則。如果您使用的是 Redhat 類型 Linux 發行版本 (例如 Amazon Linux)，我們建議使用以下命令：

```
service iptables save
```

相關主題

下列主題可能可以提供額外的資訊。

- [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)
- [從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
- [NAT 執行個體](#)
- [設定 ElastiCache 用戶端](#)
- [Amazon VPCNAT執行個體的高可用性：範例](#)

在 中尋找連線端點 ElastiCache

您的應用程式會使用端點連線至 ElastiCache 叢集。端點是節點或叢集的唯一位址。

要與 Valkey 或 Redis 搭配使用的端點 OSS

- 獨立節點，同時使用節點的端點進行讀取和寫入操作。
- Valkey 或 Valkey 或 Redis OSS (停用叢集模式) 叢集，請使用主要端點進行所有寫入操作。使用「讀取器端點」來將傳入連線平均分散在所有讀取複本間的端點中。使用個別節點端點進行讀取操作 (在 API/CLI 中，這些端點稱為讀取端點)。
- Valkey 或 Redis OSS (啟用叢集模式) 叢集，針對支援叢集模式啟用命令的所有操作使用叢集的組態端點。您必須使用支援 Valkey Cluster 或 Redis OSS Cluster (Redis OSS 3.2) 的用戶端。您仍然可以從個別節點端點讀取 (在 API/CLI 中，這些端點稱為讀取端點)。

以下各節可引導您探索執行引擎所需的端點。

要與 Memcached 搭配使用的端點

若要使用 Memcached 進行無 ElastiCache 伺服器快取，只需從主控台取得叢集端點DNS和連接埠即可。

從 中 AWS CLI，使用 `describe-serverless-caches` 命令來取得端點資訊。

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

上述操作的輸出應該看起來像這樣（JSON 格式）：

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
      "MajorEngineVersion": "1.6",
      "FullEngineVersion": "21",
      "SecurityGroupIds": [
        "sg-083eda453e1e51310"
      ],
      "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
      },
      "ARN": "<the ARN>",
      "SubnetIds": [
        "subnet-0cf759df15bd4dc65",
        "subnet-09e1307e8f1560d17"
      ],
      "SnapshotRetentionLimit": 0,
      "DailySnapshotTime": "03:00"
    }
  ]
}
```

執行個體型 Memcached 叢集：如果您使用自動探索功能，則可以使用叢集的組態端點來設定 Memcached 用戶端。這表示您必須使用支援 Automatic Discovery 的用戶端。

如果您並未使用 Automatic Discovery，則必須將用戶端設定為使用個別的節點端點以進行讀取和寫入。您也必須在新增和移除節點時追蹤這些端點。

尋找 Valkey 或 Redis OSS (停用叢集模式) 叢集的端點 (主控台)

如果 Valkery、Valkey 或 Redis OSS (停用叢集模式) 叢集只有一個節點，則節點的端點會同時用於讀取和寫入。如果 Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 叢集具有多個節點，則有三種類型的端點：主要端點、讀取器端點和節點端點。

主要端點是一律解析為叢集中主要節點DNS的名稱。主要端點不會受到您叢集變更的影響 (例如將僅供讀取複本提升至主要角色)。針對寫入活動，我們建議您的應用程式連線到主要端點。

讀取器端點會在 ElastiCache (Redis OSS) 叢集中的所有僅供讀取複本之間平均分割端點的傳入連線。其他因素 (例如應用程式建立連線或應用程式如何 (重新) 使用連線) 將決定流量分佈。隨著複本的新增或移除，讀取器端點會跟著叢集的變更即時保持在最新狀態。您可以將 ElastiCache (Redis OSS) 叢集的多個僅供讀取複本放置在不同的 AWS 可用區域 (AZ)，以確保讀取器端點的高可用性。

Note

讀取者端點並非負載平衡器。它是以循環方式解析為其中一個複本節點的 IP 地址DNS的記錄。

針對讀取活動，應用程式也可連線到叢集中的任何節點。與主要端點不同，節點端點會解析至特定端點。若您在叢集中進行變更 (例如新增或刪除複本)，您必須更新您應用程式中的節點端點。

尋找 Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 叢集的端點

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。

叢集畫面會顯示 Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 和 Valkey 或 Redis OSS (啟用叢集模式) 叢集的清單。

3. 若要尋找叢集的主要和/或讀取器端點，請選擇叢集名稱 (而非左側的按鈕)。

▼ Cluster details			
Cluster name	Description	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [lock icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [lock icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 叢集的主要和讀取器端點

如果叢集中只有一個節點，就不會有主要端點，因此您必須繼續下一個步驟。

- 如果 Valkery、Valkey 或 Redis OSS (停用叢集模式) 叢集有複本節點，您可以選擇叢集的名稱，然後選擇節點索引標籤，以尋找叢集的複本節點端點。

隨即顯示節點畫面，其顯示叢集中的每個節點、主要複本和複本，並與其端點一起列出。

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	[redacted]amazonaws.com
<input type="checkbox"/>	test-no-002	available	replica	6379	[redacted]amazonaws.com
<input type="checkbox"/>	test-no-003	available	replica	6379	[redacted]amazonaws.com

Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 叢集的節點端點

- 將端點複製到剪貼簿：
 - 逐一尋找您要複製的端點。
 - 選擇端點前面的複製圖示。

現在，端點即已複製到您的剪貼簿。如需使用端點連線至節點的相關資訊，請參閱 [連線至 Memcached 節點](#)。

Valkery 或 Valkey 或 Redis OSS (停用叢集模式) 主要端點看起來類似以下內容。其中的差異取決於是否啟用傳輸中加密。

未啟用傳輸中加密

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

已啟用傳輸中加密

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

尋找 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (主控台) 的端點

Valkey 或 Redis OSS (啟用叢集模式) 叢集具有單一組態端點。只要連線到組態端點，您的應用程式就可以探索叢集中每個碎片的主要和讀取端點。

尋找 Valkey 或 Redis OSS (啟用叢集模式) 叢集的端點

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集。

叢集畫面會顯示叢集清單。選擇您要連線的叢集。
3. 若要尋找叢集的組態端點，請選擇叢集名稱 (而非選項按鈕)。
4. Configuration endpoint (組態端點) 會顯示於 Cluster details (叢集詳細資訊) 中。若要加以複製，請選擇端點左側的複製圖示。

尋找叢集的端點 (主控台) (Memcached)

所有 Memcached 端點都是讀取/寫入端點。若要連線至 Memcached 叢集中的節點，您的應用程式可以使用每個叢集的端點，或搭配使用 Automatic Discovery 與叢集的組態端點。若要使用 Automatic Discovery，您必須使用支援 Automatic Discovery 的用戶端。

使用 Automatic Discovery 時，您的用戶端應用程式會使用組態端點連線至 Memcached 叢集。當您透過新增或移除節點以擴展叢集時，應用程式會自動「知道」叢集中的所有節點，並能連線至其中任何節

點。如果未使用 Automatic Discovery，應用程式就無法完成上述作業；您必須在每次新增或移除節點時手動更新應用程式中的端點。

若要複製端點，請選擇端點位址前面的複製圖示。如需使用端點連線至節點的相關資訊，請參閱 [連線至 Memcached 節點](#)。

組態端點和節點端點看起來非常類似。下列會將差異以粗體醒目提示。

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port # configuration endpoint  
contains "cfg"  
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

Important

如果您選擇 CNAME 為 Memcached 組態端點建立，為了讓自動探索用戶端將識別 CNAME 為組態端點，您必須包含在 .cfg. 中 CNAME。

尋找端點 (AWS CLI)

對於 Memcached，您可以使用 AWS CLI for Amazon ElastiCache 來探索節點和叢集的端點。

對於 Redis，您可以使用 AWS CLI for Amazon ElastiCache 來探索節點、叢集和複寫群組的端點。

主題

- [尋找節點和叢集的端點 \(AWS CLI\)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \(AWS CLI\)](#)

尋找節點和叢集的端點 (AWS CLI)

您可以使用 AWS CLI，透過 `describe-cache-clusters` 命令探索叢集及其節點的端點。對於 Valkey 或 Redis OSS 叢集，命令會傳回叢集端點。若是 Memcached 叢集，則命令會傳回組態端點。如果您包含選用參數 `--show-cache-node-info`，則命令也會傳回叢集中個別節點的端點。

Example

以下命令會擷取 Memcached 叢集 `mycluster` 的組態端點 (`ConfigurationEndpoint`) 和個別節點端點 (`Endpoint`)。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

針對 Windows：

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

上述操作的輸出應該看起來像這樣 (JSON 格式)。

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {
```

```
    "CacheNodeId": "0001",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0003",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.amazonaws.com"
},
```

```

    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "available",
    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled"
  }
]
}

```

⚠ Important

如果您選擇 CNAME 為 Memcached 組態端點建立，為了讓自動探索用戶端將識別 CNAME 為組態端點，您必須包含在 `.cfg` 中 CNAME。例如，`session.save_path` 參數 `php.ini` 檔案中的 `mycluster.cfg.local`。

Example

對於 Valkey 和 Redis OSS，下列命令會擷取單節點叢集 `mycluster` 的叢集資訊。

⚠ Important

參數 `--cache-cluster-id` 可與複寫群組中的單一節點 Valkey 或 Redis OSS (停用叢集模式) 叢集 ID 或特定節點 ID 搭配使用。複寫群組 `--cache-cluster-id` 的是 4 位數的值，例如 `0001`。如果 `--cache-cluster-id` 是複寫群組中叢集 (節點) 的 ID，則 `replication-group-id` 會包含在輸出中。

若為 Linux、macOS 或 Unix：

```

aws elasticache describe-cache-clusters \
  --cache-cluster-id redis-cluster \

```

```
--show-cache-node-info
```

針對 Windows :

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id redis-cluster ^  
  --show-cache-node-info
```

上述操作的輸出應該看起來像這樣 (JSON 格式)。

```
{  
  "CacheClusters": [  
    {  
      "CacheClusterStatus": "available",  
      "SecurityGroups": [  
        {  
          "SecurityGroupId": "sg-77186e0d",  
          "Status": "active"  
        }  
      ],  
      "CacheNodes": [  
        {  
          "CustomerAvailabilityZone": "us-east-1b",  
          "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",  
          "CacheNodeStatus": "available",  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Address": "redis-cluster.amazonaws.com",  
            "Port": 6379  
          },  
          "ParameterGroupStatus": "in-sync"  
        }  
      ],  
      "AtRestEncryptionEnabled": false,  
      "CacheClusterId": "redis-cluster",  
      "TransitEncryptionEnabled": false,  
      "CacheParameterGroup": {  
        "ParameterApplyStatus": "in-sync",  
        "CacheNodeIdsToReboot": [],  
        "CacheParameterGroupName": "default.redis3.2"  
      },  
      "NumCacheNodes": 1,  
      "PreferredAvailabilityZone": "us-east-1b",
```



```
    "AutoMinorVersionUpgrade": true,
    "Engine": "redis",
    "AuthTokenEnabled": false,
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
    "CacheSecurityGroups": [],
    "CacheSubnetGroupName": "default",
    "CacheNodeType": "cache.t2.small",
    "DataTiering": "disabled"
    "EngineVersion": "3.2.10",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
  }
]
}
```

如需詳細資訊，請參閱主題 [describe-cache-clusters](#)。

尋找 Valkey 或 Redis OSS 複寫群組的端點 (AWS CLI)

您可以使用 AWS CLI 來探索複寫群組及其叢集的端點，並使用 `describe-replication-groups` 命令。此命令會傳回複寫群組的主要端點、複寫群組中所有叢集 (節點) 的清單和其端點，以及讀取器端點。

下列操作會擷取複寫群組 `myreplgroup` 的主要端點與讀取器端點。使用主要端點來進行所有寫入操作。

```
aws elasticache describe-replication-groups \
  --replication-group-id myreplgroup
```

針對 Windows :

```
aws elasticache describe-replication-groups ^
  --replication-group-id myreplgroup
```

此操作的輸出應該看起來像這樣 (JSON 格式)。

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
```

```
"Description": "test",
"NodeGroups": [
  {
    "Status": "available",
    "NodeGroupMembers": [
      {
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-001.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-001"
      },
      {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-002.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-002"
      },
      {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-003.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-003"
      }
    ],
    "NodeGroupId": "0001",
    "PrimaryEndpoint": {
      "Port": 6379,
      "Address": "myreplgroup.amazonaws.com"
    },
    "ReaderEndpoint": {
      "Port": 6379,
      "Address": "myreplgroup-ro.amazonaws.com"
    }
  }
]
```

```
    }
  }
],
"ReplicationGroupId": "myreplgroup",
"AutomaticFailover": "enabled",
"SnapshottingClusterId": "myreplgroup-002",
"MemberClusters": [
  "myreplgroup-001",
  "myreplgroup-002",
  "myreplgroup-003"
],
"PendingModifiedValues": {}
}
]
}
```

如需詳細資訊，請參閱 命令參考 [describe-replication-groups](#) 中的 。AWS CLI

尋找端點 (ElastiCache API)

對於 Memcached，您可以使用 Amazon ElastiCache API 探索節點和叢集的端點。

對於 Redis，您可以使用 Amazon ElastiCache API 探索節點、叢集和複寫群組的端點。

主題

- [尋找節點和叢集的端點 \(ElastiCache API\)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \(ElastiCache API\)](#)

尋找節點和叢集的端點 (ElastiCache API)

您可以使用 ElastiCache API 來探索具有 DescribeCacheClusters 動作的叢集端點及其節點。對於 Valkey 或 Redis OSS 叢集，命令會傳回叢集端點。若是 Memcached 叢集，則命令會傳回組態端點。如果您包含選用參數 ShowCacheNodeInfo，則動作也會傳回叢集中個別節點的端點。

Example

對於 Memcached，下列命令會擷取 Memcached 叢集 mycluster 的組態端點 (ConfigurationEndpoint) 和個別節點端點 (Endpoint)。

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=mycluster  
  &ShowCacheNodeInfo=true  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Important

如果您選擇 CNAME 為 Memcached 組態端點建立，為了讓自動探索用戶端將 識別 CNAME 為組態端點，您必須包含在 .cfg. 中 CNAME。例如，session.save_path 參數 php.ini 檔案中的 mycluster.*cfg*.local。

尋找 Valkey 或 Redis OSS 複寫群組的端點 (ElastiCache API)

您可以使用 ElastiCache API 來探索複寫群組及其叢集的端點，以及 DescribeReplicationGroups 動作。此動作會傳回複寫群組的主要端點、複寫群組中所有叢集的清單和其端點，以及讀取器端點。

下列操作會擷取複寫群組的主要端點 () PrimaryEndpoint、讀取器端點 (ReaderEndpoint) 和個別節點端點 (ReadEndpoint) myreplgroup。使用主要端點來進行所有寫入操作。

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=myreplgroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeReplicationGroups](#)。

在 中 使用碎片 ElastiCache

碎片 (API/CLI : 節點群組) 是一到六個 ElastiCache 包含 Valkey 或 Redis OSS 節點的集合。Valkey 或 Redis OSS (停用叢集模式) 叢集永遠不會有多個碎片。使用碎片，您可以將大型資料庫分隔為更小、更快速且更易於管理的零件，稱為資料碎片。這可以透過在多個不同的區段中分佈操作來提高資料庫效率。使用碎片可以提供許多好處，包括改善的效能、可擴展性和成本效益。

您可以建立具有較高數量碎片和較低數量複本的叢集，每一叢集最高總計為 90 個節點。此叢集組態的範圍可以從 90 個碎片和 0 個複本到 15 個碎片和 5 個複本，這是允許的複本最大數量。叢集的資料會分割到叢集各個碎片中。如果一個碎片中有超過一個節點，碎片會實作複寫，其中一個節點為讀取/寫入主要節點，其他節點則為僅供讀取複本節點。

如果引擎版本 Valkey 7.2 或 Redis 5.0.6 或更新版本，節點或碎片限制可以提高到每個叢集最多 OSS 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 (每個碎片一個主要版本和 5 個複本) 到 500 個 (單一主要版本並且沒有複本) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路 CIDR 太小，範圍太小，或者子網路被其他叢集共用和大量使用。如需詳細資訊，請參閱 [建立子網路群組](#)。

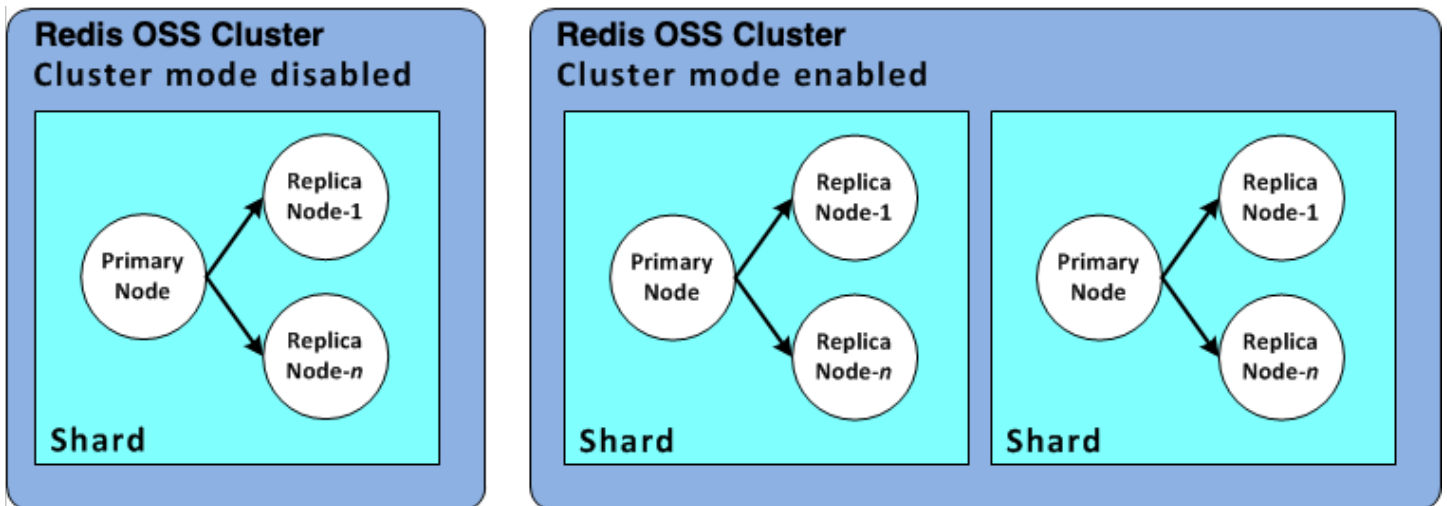
對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

當您使用 ElastiCache 主控台建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集時，您可以指定叢集中的碎片數量和碎片中的節點數量。如需詳細資訊，請參閱 [建立 Valkey 或 Redis OSS \(啟用叢集模式\) 叢集 \(主控台\)](#)。如果您使用 ElastiCache API 或 AWS CLI 建立叢集 (在 API 中稱為複寫群組 CLI)，您可以獨立設定碎片 (API/CLI：節點群組) 中的節點數量。如需詳細資訊，請參閱下列內容：

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

碎片中的每個節點都具有相同的運算、儲存體及記憶體規格。ElastiCache API 可讓您控制全碎片屬性，例如節點數量、安全設定和系統維護時段。



Valkey 或 Redis OSS 碎片組態

如需詳細資訊，請參閱 [Valkey 或 Redis 的離線重新共享 OSS \(啟用叢集模式\)](#) 和 [Valkey 或 Redis 的線上重新共享 OSS \(啟用叢集模式\)](#)。

尋找碎片的 ID

您可以使用 AWS Management Console、AWS CLI 或 [來尋找碎片的 ID ElastiCache API](#)。

使用 AWS Management Console

主題

- [針對 Valkey 或 Redis OSS \(停用叢集模式 \)](#)
- [針對 Valkey 或 Redis OSS \(已啟用叢集模式 \)](#)

針對 Valkey 或 Redis OSS (停用叢集模式)

Valkey 或 Redis OSS (停用叢集模式) 複寫群組碎片一律IDs為 0001。

針對 Valkey 或 Redis OSS (已啟用叢集模式)

下列程序使用 AWS Management Console 尋找 Valkey 或 Redis OSS (啟用叢集模式) 的複寫群組碎片 ID。

在 Valkey 或 Redis OSS (啟用叢集模式) 複寫群組中尋找碎片 ID

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS，然後選擇您要尋找碎片的 Valkey 或 Redis OSS (啟用叢集模式) 複寫群組的名稱IDs。
3. 在 Shard Name (碎片名稱) 欄中，碎片 ID 是碎片名稱的最後四位數。

使用 AWS CLI

若要尋找 Valkey 或 Redis (停用OSS叢集模式) 或 Valkey 或 Redis (啟用叢集模式) 複寫群組的碎片 OSS (節點群組) ID，請使用 AWS CLI 操作describe-replication-groups搭配下列選用參數。

- **--replication-group-id** - 選用參數，使用時會將輸出限制為指定複寫群組的詳細資訊。若忽略此參數，則最多會傳回 100 個複寫群組詳細資訊。

Example

此命令將傳回 sample-repl-group 的詳細資訊。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-replication-groups \  
  --replication-group-id sample-repl-group
```

針對 Windows :

```
aws elasticache describe-replication-groups ^  
  --replication-group-id sample-repl-group
```

此命令的輸出看起來會與以下內容相似。碎片（節點群組）ID 為 *highlighted* 以更輕鬆地找到它們。

```
{  
  "ReplicationGroups": [  
    {  
      "Status": "available",  
      "Description": "2 shards, 2 nodes (1 + 1 replica)",  
      "NodeGroups": [  
        {  
          "Status": "available",  
          "Slots": "0-8191",  
          "NodeGroupId": "0001",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2c",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-002"  
            }  
          ]  
        },  
        {  
          "Status": "available",  
          "Slots": "8192-16383",  
          "NodeGroupId": "0002",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2b",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0002-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",
```



```
        "CacheNodeId": "0001",
        "CacheClusterId": "sample-repl-group-0002-002"
    }
  ]
},
"ConfigurationEndpoint": {
  "Port": 6379,
  "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "sample-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "13:00-14:00",
"MemberClusters": [
  "sample-repl-group-0001-001",
  "sample-repl-group-0001-002",
  "sample-repl-group-0002-001",
  "sample-repl-group-0002-002"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
]
}
```

使用 ElastiCache API

若要尋找 Valkey 或 Redis (停用OSS叢集模式) 或 Valkey 或 Redis (啟用叢集模式) 複寫群組的碎片 OSS (節點群組) ID，請使用 AWS CLI 操作describe-replication-groups搭配下列選用參數。

- **ReplicationGroupId** - 選用參數，使用時會將輸出限制為指定複寫群組的詳細資訊。如果省略此參數，則最多的詳細資訊 **xxx** 複寫群組會傳回。

Example

此命令將傳回 sample-repl-group 的詳細資訊。

若為 Linux、macOS 或 Unix：

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroup  
&ReplicationGroupId=sample-repl-group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

比較 ValkeyOSS、Redis 和 Memcached 自行設計的快取

Amazon ElastiCache 支援 ValkeyOSS、Redis 和 Memcached 快取引擎。每個引擎都提供一些優點。使用本主題中的資訊來協助您選擇最符合您需求的引擎和版本。

Important

建立快取、自行設計的叢集或複寫群組後，您可以升級至較新的引擎版本，但無法降級至較舊的引擎版本。如果您想要使用較舊的引擎版本，則必須刪除現有的快取、自行設計的叢集或複寫群組，並使用較舊的引擎版本再次建立。

這兩個引擎表面上看起來很類似。每個引擎都是記憶體內金鑰/值存放區。不過，實際上有顯著的差異。

如果適用下列情況，請選擇 Memcached：

- 您需要盡可能最簡單的模型。
- 您需要執行具有多個核心或執行緒的大型節點。
- 您需要能夠向外和向內擴展，並視您系統需求增加和減少來新增和移除節點。
- 您需要快取物件。

OSS ElastiCache 如果下列條件適用於您，請選擇 Valkey 或 Redis：

- ElastiCache 搭配 Valkey 7.2 或 Redis 7.0 OSS版（增強版）

您想要使用[函數](#)、[陰影 Pub/Sub](#) 或[ACL改進](#)。如需詳細資訊，請參閱 [Redis 7.0 OSS版（增強版）](#)。

- ElastiCache（Redis OSS）6.2 版（增強版）

您希望能夠使用 r6gd 節點類型SSD在記憶體和 之間分層資料。如需詳細資訊，請參閱[資料分層](#)。

- ElastiCache (Redis OSS) 6.0 版 (增強版)

您想要使用角色型存取控制驗證使用者。

如需詳細資訊，請參閱 [Redis 6.0 OSS版 \(增強版 \)](#)。

- ElastiCache (Redis OSS) 5.0.0 版 (增強版)

您想要使用 [Redis OSS 串流](#)，這是一種日誌資料結構，可讓生產者即時附加新項目，也允許取用者以封鎖或非封鎖方式取用訊息。

如需詳細資訊，請參閱 [Redis 5.0.0 OSS版 \(增強版 \)](#)。

- ElastiCache (Redis OSS) 4.0.10 版 (增強版)


支援加密，並動態新增或移除 Valkey 或 Redis OSS (啟用叢集模式) 叢集中的碎片。

如需詳細資訊，請參閱 [Redis 4.0.10 OSS版 \(增強版 \)](#)。

以下版本已棄用、已達到或即將達到生命週期結束。

- ElastiCache (Redis OSS) 3.2.10 版 (增強版)

支援從 Valkey 或 Redis OSS (啟用叢集模式) 叢集動態新增或移除碎片的機能。

 Important

目前 ElastiCache (Redis OSS) 3.2.10 不支援加密。

如需詳細資訊，請參閱下列內容：

- [Redis 3.2.10 OSS版 \(增強版 \)](#)
- 線上轉載 Redis 的最佳實務OSS，如需詳細資訊，請參閱下列內容：
 - [最佳實務：線上重新分片](#)
 - [Valkey 或 Redis 的線上重新分割和碎片重新平衡 OSS \(已啟用叢集模式 \)](#)
- 如需擴展 Redis OSS叢集的詳細資訊，請參閱[擴展](#)。

- ElastiCache (Redis OSS) 3.2.6 版 (增強版)

如果您需要舊版 Redis 的功能OSS以及下列功能，請選擇 ElastiCache (Redis OSS) 3.2.6 :

- 傳輸中加密。如需詳細資訊，請參閱 [Amazon ElastiCache \(Redis OSS \) 傳輸中加密](#)。
- 靜態加密。如需詳細資訊，請參閱 [Amazon ElastiCache \(Redis OSS \) 靜態加密](#)。
- ElastiCache (Redis OSS) (啟用叢集模式) 3.2.4 版

如果您需要 Redis OSS 2.8.x 的功能加上下列功能，請選擇 Redis OSS 3.2.4 (叢集模式) :

- 您需要將資料分割到 2 至 500 個節點群組中 (僅限叢集模式)。
- 您需要地理空間索引 (叢集模式或非叢集模式)。
- 您不需要支援多個資料庫。
- ElastiCache (Redis OSS) (非叢集模式) 2.8.x 和 3.2.4 (增強型)

如果下列內容適用於您，請選擇 Redis OSS 2.8.x 或 Redis OSS 3.2.4 (非叢集模式) :

- 您需要複雜資料類型，例如字串、雜湊、清單、集合、有序集合及點陣圖。
- 您需要排序或排名記憶體內資料集。
- 您的金鑰存放區必須具備持久性。
- 您需要針對讀取密集型應用程式，將資料從主要複本複寫到一或多個僅供讀取複本。
- 您需要在主要節點失敗時自動容錯移轉。
- 您需要發佈和訂閱 (pub/sub) 功能，用於通知用戶端伺服器上的事件。
- 您需要自行設計叢集的備份和還原功能，以及無伺服器快取。
- 您需要支援多個資料庫。

Memcached、Valkey 或 Redis OSS (停用叢集模式) 和 Valkey 或 Redis OSS (啟用叢集模式) 的比較摘要

	Memcached	Valkey 或 Redis OSS (停用叢集模式)	Valkey 或 Redis OSS (啟用叢集模式)
引擎版本 +	1.4.5 及更新版本	4.0.10 和更新版本	4.0.10 和更新版本
資料類型	簡單	2.8.x - 複雜 * 複雜	3.2.x 和更新版本 - 複雜
資料分割	是	否	是
叢集可修改	是	是	3.2.10 和更新版本 - 有限
線上重新分片	否	否	3.2.10 版和更新版本
加密	傳輸中 1.6.12 及更新版本	4.0.10 和更新版本	4.0.10 和更新版本
資料分層	否	6.2 及更新版本	6.2 及更新版本
合規認證			
合規認證			
聯準會RAMP	是 - 1.6.12 和更新版本	4.0.10 和更新版本	4.0.10 和更新版本
HIPAA	是 - 1.6.12 和更新版本	4.0.10 和更新版本	4.0.10 和更新版本
PCI DSS	是	4.0.10 和更新版本	4.0.10 和更新版本
多執行緒	是	否	否
節點類型升級	否	是	是
引擎升級	是	是	是
高可用性 (複寫)	否	是	是

	Memcached	Valkey 或 Redis OSS (停用叢集模 式)	Valkey 或 Redis OSS (啟用叢集模 式)
自動容錯移轉	否	選用	必要
發佈/訂閱功能	否	是	是
有序集合	否	是	是
備份和還原	僅適用於 Serverless Memcached，不適用於自行設計的 Memcached 叢集	是	是
地理空間索引	否	4.0.10 和更新版本	是

備註：

字串、物件 (例如資料庫)

* 字串、集合、有序集合、清單、雜湊、點陣圖、hyperloglog

字串、集合、有序集合、清單、雜湊、點陣圖、hyperloglog、地理空間索引

+ 排除已棄用、已到達或即將達到生命週期結束的版本。

選擇適用於您叢集的引擎之後，我們建議您使用該引擎的最新版本。如需詳細資訊，請參閱[支援的節點類型](#)。

Valkey 或 Redis 的線上遷移 OSS

透過使用線上遷移，您可以將資料從 Amazon OSS 上的自我託管開放原始碼 Valkey 或 Redis 遷移 EC2 至 Amazon ElastiCache。

Note

在 r6gd 節點類型上執行的無 ElastiCache 伺服器快取或叢集不支援線上遷移。

概觀

若要將資料從在 Amazon OSS 上執行的開放原始碼 Valkey 或 Redis 遷移 EC2 至 Amazon ElastiCache，需要現有或新建立的 Amazon ElastiCache 部署。此部署必須有可以遷移的組態。此叢集也必須與您要使用的組態相符，包括如執行個體類型、碎片數目和複本數目等屬性。

線上遷移旨在將資料從 Amazon OSS 上的自我託管開放原始碼 Valkey 或 Redis 遷移 EC2 至 ElastiCache，而不是在 ElastiCache 叢集之間移動資料。

Important

強烈建議您先讀完下列段落，再開始線上遷移程序。

當您呼叫 StartMigration API 操作或 AWS CLI 命令時，遷移就會開始。遷移 Valkey 或 Redis OSS 叢集模式停用叢集時，遷移程序會讓 ElastiCache Valkey 或 Redis OSS 叢集的主要節點成為來源 Valkey 或 Redis OSS 主要叢集的複本。遷移啟用 Valkey 或 Redis OSS 叢集模式的叢集時，遷移程序會讓每個 ElastiCache 碎片的主要節點成為來源叢集具有相同插槽的對應碎片複本。

準備好用戶端變更後，請呼叫 CompleteMigration API 操作。API 此操作會使用主要節點和複本節點（如適用），將 ElastiCache 部署提升為主要 Valkey 或 Redis OSS 部署。現在，您可以重新導向用戶端應用程式，以開始將資料寫入 ElastiCache。在整個遷移過程中，您可以透過在 [Valkey 節點和主要節點上執行 valkey-cli INFO](#) 命令來檢查複寫狀態。ElastiCache

遷移步驟

以下主題說明遷移資料的程序：

- [準備您的來源和目標以進行遷移](#)
- [測試資料遷移](#)
- [開始遷移](#)
- [驗證資料遷移程序](#)
- [完成資料遷移](#)

準備您的來源和目標以進行遷移

透過這些步驟，您可以準備將資料從自我託管的 Valkey 或 Redis 來源遷移 EC2 至 ElastiCache，或從 Redis OSS 叢集遷移至 ElastiCache Valkey 叢集。

您必須確保在從 ElastiCache 主控台API或 開始遷移之前滿足下列所有四個先決條件 AWS CLI。

準備您的來源和目標 Valkey 或 Redis OSS節點以進行遷移

1. 識別目標 ElastiCache 部署，並確保您可以將資料遷移到其中。

現有或新建立的 ElastiCache 部署應符合下列遷移需求：

- 它使用 Valkey 或 Redis OSS引擎 5.0.6 版或更新版本。
- 它未啟用傳輸中加密。
- 已啟用多個可用區。
- 它有足夠的記憶體，可以容納來自 Valkey 或 Redis OSS叢集的資料。若要設定合適的保留記憶體設定，請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。
- 對於停用叢集模式，如果使用或 Valkey 或 Redis OSS5.0.6 OSS版之後使用 CLI或 CLI主控台，則可以直接從 Valkey 或 Redis 2.8.21 版之後遷移至 Valkey 或 Redis 5.0.6 OSS版之後。對於啟用叢集模式，如果使用或 Redis 5.0.6 OSS版之後，您可以使用 CLI或 CLI主控台，從任何啟用叢集模式的 Valkey 或 Redis OSS版本直接遷移至 Redis 5.0.6 OSS版之後。
- 來源與目標中的碎片數目相符。
- 它不屬於全域資料儲存的一部分。
- 它的資料分層已停用。

2. 請確定開放原始碼 Valkey 或 Redis OSS和 ElastiCache 部署的組態相容。

目標 ElastiCache 部署中的所有下列項目至少應與複寫的 Valkey 或 Redis OSS組態相容：

- 您的叢集不應AUTH已啟用。
- 組態protected-mode應設定為 no。
- 如果您的 Valkey 或 Redis OSS 組態中有bind組態，則應更新組態以允許 ElastiCache 節點的請求。
- ElastiCache 節點和 Valkey 或 Redis OSS叢集上的邏輯資料庫數目應該相同。此值是在 Valkey 或 Redis 組態databases中使用 OSS 設定。
- 執行資料修改的 Valkey 或 Redis OSS命令不應重新命名，以允許資料複寫成功。例如 sync、psyncinfo、config、command和 cluster。
- 若要將資料從 Valkey 或 Redis OSS叢集複寫至 ElastiCache，請確定有足夠的 CPU和 記憶體來處理此額外負載。此載入來自您的 Valkey 或 Redis OSS叢集建立RDB的檔案，並透過網路傳輸到 ElastiCache 節點。

- 執行下列 ElastiCache 動作，確定您的執行個體可以連線至：
 - 確認每個執行個體的 IP 地址都是私有。
 - 在與執行個體上的 Valkey 或 Redis 相同的虛擬私有雲端 (VPC) OSS 中指派或建立 ElastiCache 部署 (建議)。
 - 如果 VPCs 不同，請設定 VPC 對等，以允許節點之間的存取。如需 VPC 互連的詳細資訊，請參閱 [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)。
 - 連接至 Valkey 或 Redis OSS 執行個體的安全群組應允許來自 ElastiCache 節點的傳入流量。
- 確保您的應用程式可以在資料遷移完成後將流量導向 ElastiCache 節點。如需詳細資訊，請參閱 [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)。

測試資料遷移

完成所有先決條件之後，您可以使用 AWS Management Console ElastiCache API、或來驗證遷移設定 AWS CLI。下列範例顯示使用 CLI。

使用以下參數呼叫 test-migration 命令以測試遷移：

- replication-group-id – 做為資料遷移目標位置的複寫群組 ID。
- customer-node-endpoint-list – 做為資料遷移來源位置的端點清單。清單應只有一個元素。

以下是使用的範例 CLI。

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache 將在沒有任何實際資料遷移的情況下驗證遷移設定。

開始遷移

完成所有先決條件之後，您可以使用 AWS Management Console ElastiCache API、或開始資料遷移 AWS CLI。若叢集模式已啟用，在槽位遷移不同的情況下，則會先進行重新分片，再進行即時遷移。下列範例顯示使用 CLI。

Note

建議您使用 TestMigration API 來驗證遷移設定。不過這僅限選用。

使用以下參數呼叫 start-migration 命令以開始遷移：

- --replication-group-id – 目標 ElastiCache 複寫群組的識別碼
- --customer-node-endpoint-list – 具有 DNS 或 IP 地址的端點清單，以及執行來源 Valkey 或 Redis OSS 叢集的連接埠。此清單針對叢集模式已停用和叢集模式已啟用只能採用一個元素。如果您已啟用鏈結複寫，端點可以指向複本，而不是 Valkey 或 Redis OSS 叢集的主要節點。

以下是使用的範例 CLI。

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

當您執行此命令時，ElastiCache 主要節點（在每個碎片中）會自行設定為 Valkey 或 Redis OSS 執行個體的複本（在叢集中擁有相同插槽的對應碎片中）。ElastiCache 叢集的狀態變更為遷移，且資料開始從您的 Valkey 或 Redis OSS 執行個體遷移至 ElastiCache 主要節點。根據資料的大小和 Valkey 或 Redis OSS 執行個體上的載入，遷移可能需要一段時間才能完成。您可以在 [Valkey 執行個體和主要節點上執行 valkey-cli INFO](#) 命令，以檢查遷移進度。ElastiCache

成功複寫後，所有寫入您的 Valkey 或 Redis OSS 執行個體都會 ElastiCache 傳播到叢集。您可以使用 ElastiCache 節點進行讀取。不過，您無法寫入 ElastiCache 叢集。如果 ElastiCache 主要節點有與其連接的其他複本節點，這些複本節點會繼續從 ElastiCache 主要節點複寫。如此一來，來自 Valkey 或 Redis OSS 叢集的所有資料都會複寫至 ElastiCache 叢集中的所有節點。

如果 ElastiCache 主要節點無法成為 Valkey 或 Redis OSS 執行個體的複本，則會重試數次，然後再將自己提升回主要節點。然後，叢集的狀態 ElastiCache 會變更為可用的，並傳送有關無法啟動遷移的複寫群組事件。若要對此類失敗進行故障診斷，請執行以下檢查：

- 查看複寫群組事件。任何有關事件的特定資訊以修正遷移失敗的問題。
- 如果事件不提供任何特定資訊，請確保您已遵循 [準備您的來源和目標以進行遷移](#) 中的指導方針。
- 確保 VPC 和子網路的路由組態允許 ElastiCache 節點與 Valkey 或 Redis OSS 執行個體之間的流量。
- 確保連接至 Valkey 或 Redis OSS 執行個體的安全群組允許來自 ElastiCache 節點的輸入繫結流量。

- 如需複寫專用失敗的詳細資訊，請查看執行個體的 Valkey 或 Redis OSS 日誌。

驗證資料遷移程序

遷移開始後，您可以執行以下操作以追蹤其進度：

- 確認 Valkey 或 Redis OSS `master_link_status` 位於 ElastiCache 主節點的 INFO 命令輸出中 (s)。您也可以從 ElastiCache 主控台中找到此資訊。在 CloudWatch 指標下選取叢集和，觀察 Primary Link Health Status。一旦值達到 1，資料就會同步。
- 您可以在 Valkey 或 Redis OSS 執行個體上執行 INFO 命令，以檢查 ElastiCache 複本是否具有線上狀態。執行此操作，也可以提供有關複寫延遲的資訊。
- 在 Valkey 或 Redis OSS 執行個體上使用 [CLIENTLIST](#) 命令來驗證低用戶端輸出緩衝區。

資料遷移完成後，資料會與進入 Valkey 或 Redis OSS 叢集主要節點 (些) 的任何新寫入同步。

完成資料遷移

當您準備好切換到 ElastiCache 叢集時，請使用具有下列參數的 `complete-migration` CLI 命令：

- `--replication-group-id` - 複寫群組的識別碼。
- `--force` - 此值會強迫遷移停止且不確認資料是否正在同步。

以下是範例。

```
aws elasticache complete-migration --replication-group-id test-cluster
```

當您執行此命令時，ElastiCache 主要節點 (在每個碎片中) 會停止從 Valkey 或 Redis OSS 執行個體複寫，並將其提升為主要節點。此提升通常會在幾分鐘之內完成。若要確認提升至主要層級，請檢查事件 Complete Migration successful for test-cluster。此時，您可以指示應用程式 ElastiCache 寫入和讀取。ElastiCache 叢集狀態應該會從遷移到可用的。

如果升級為主要失敗，ElastiCache 主要節點會繼續從 Valkey 或 Redis OSS 執行個體複寫。ElastiCache 叢集持續處於遷移狀態，並傳送有關失敗的複寫群組事件訊息。若要就此失敗進行故障診斷，請進行以下檢查：

- 檢查複寫群組事件。使用有關事件的特定資訊以修正失敗的問題。

- 您或許會收到有關資料沒有同步的事件訊息。如果是這樣，請確定 ElastiCache 主要可以從您的 Valkey 或 Redis OSS執行個體複寫，而且兩者都是同步的。如果您仍要停止遷移，您可以使用 `force` 選項執行處理命令。
- 如果其中一個 ElastiCache 節點正在進行取代，您可能會收到事件訊息。您可以在取代完成後重試一次完成遷移步驟。

使用主控台執行線上資料遷移

您可以使用 AWS Management Console 將資料從叢集遷移到 Valkey 或 Redis OSS叢集。

使用主控台執行線上資料遷移

1. 登入主控台，並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 建立新的 Valkey 或 Redis OSS叢集，或選擇現有的叢集。確認叢集符合下列需求：
 - 您的引擎版本應為 Valkey 7.2 及更高版本，或 Redis OSS 5.0.6 或更高版本。
 - 您的叢集不應 AUTH 已啟用。
 - 組態 `protected-mode` 應設定為 `no`。
 - 如果您的 Valkey 或 Redis OSS 組態中有 `bind` 組態，則應更新組態以允許 ElastiCache 節點的請求。
 - ElastiCache 節點與 Valkey 或 Redis OSS叢集之間的資料庫數目應相同。此值是在引擎組態 `databases` 中使用 設定。
 - 執行資料修改的 Valkey 或 Redis OSS命令不應重新命名，以允許資料複寫成功。
 - 若要將資料從 Valkey 或 Redis OSS叢集複寫至 ElastiCache，請確定有足夠的 CPU 和 記憶體來處理此額外負載。此載入來自您的 Valkey 或 Redis OSS叢集建立 RDB 的檔案，並透過網路傳輸到 ElastiCache 節點。
 - 叢集的狀態為 `available` (可用)。
3. 選取叢集後，Actions (動作) 請 選擇 `Migrate Data from Endpoint` (從端點遷移資料)。
4. 在從端點遷移資料對話方塊中，輸入 IP 地址，以及您的 Valkey 或 Redis OSS叢集可用的連接埠。

Important

IP 位址必須完全相同。如果輸入錯誤的位址，遷移就會失敗。

5. 選擇 Start Migration (開始遷移)。

當叢集開始遷移時，叢集的狀態會變更為 Modifying (修改中)，然後 Migrating (遷移中)。

6. 在瀏覽窗格中選擇 Events (事件) 以監控遷移程序。

遷移程序進行期間，您隨時都可以停止遷移。若要執行此操作，請選擇您的叢集，並為 Actions (動作) 選擇 Stop Data Migration (停止資料遷移)。之後叢集就會進入 Available (可用) 狀態。

如果遷移成功，叢集的狀態就會變成 Available (可用)，而且事件記錄會顯示以下項目：

```
Migration operation succeeded for replication group ElastiCacheClusterName.
```

如果遷移失敗，叢集的狀態就會變成 Available (可用)，而且事件記錄會顯示以下項目：

```
Migration operation failed for replication group ElastiCacheClusterName.
```

選擇 的區域和可用區域 ElastiCache

您可以使用對應的端點指定區域和可用區域，為 ElastiCache 叢集提供額外的可擴展性和可靠性。

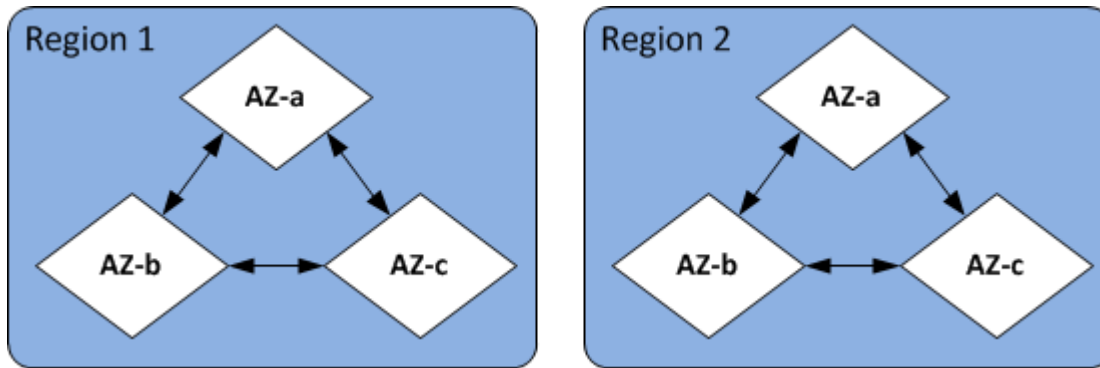
AWS 雲端運算資源存放在高可用性的資料中心設施中。為了提供額外的可擴展性和可靠性，這些資料中心設施會位在不同的實體位置。這些地點是依「區域」及「可用區域」分類。

AWS 區域很大，廣泛分散到不同的地理位置。可用區域是 AWS 區域中的不同位置，其設計旨在與其他可用區域中的故障隔離。它們為相同 AWS 區域中的其他可用區域提供廉價、低延遲的網路連線。

Important

每個區域都是完全獨立的。您啟動的任何 ElastiCache 活動（例如建立叢集）只會在目前的預設區域中執行。

若要在特定區域中建立或使用叢集，請使用對應的區域服務端點。如需了解服務端點，請參閱[支援的區域和端點](#)。



區域與可用區域

主題

- [使用 Memcached 的可用區域考量](#)
- [安置您的節點](#)
- [支援的區域和端點](#)
- [搭配 使用本機區域 ElastiCache](#)
- [搭配 使用 Outposts ElastiCache](#)

使用 Memcached 的可用區域考量

將 Memcached 節點散佈在一個區域內的多個可用區域，有助於保護您免受災難性故障的影響，例如可用區域內停電。

無伺服器快取

ElastiCache 無伺服器快取會建立跨多個可用區域的高可用性快取。您可以指定來自不同可用區域的子網路，並與建立無伺服器叢集VPC相同，或ElastiCache 將從您的預設 自動選擇子網路VPC。

設計您自己的 ElastiCache (Memcached) 叢集

Memcached 叢集最多可以有 300 個節點。當您建立或新增節點至 Memcached 叢集時，您可以為所有節點指定單一可用區域、允許為所有節點 ElastiCache 選擇單一可用區域、為每個節點指定可用區域，或允許為每個節點 ElastiCache 選擇可用區域。將新節點新增至現有的 Memcached 叢集時，您可以在不同的可用區域中建立新節點。快取節點一旦建立，就無法修改其可用區域。

如果您希望單一可用區域叢集中的叢集將節點分散到多個可用區域，ElastiCache 則可以在各種可用區域中建立新的節點。接著您可以刪除部分或所有原始快取節點。建議您使用此方法。

將 Memcached 節點從單一可用區域遷移至多個可用區域

1. 在需要的可用區域中建立新的快取節點，藉此修改叢集。在請求中執行下列操作：
 - 將 AZMode (CLI : - -az-mode) 設定為 cross-az。
 - 將 NumCacheNodes (CLI : - -num-cache-nodes) 設定為目前作用中快取節點的數量，加上您要建立的新快取節點數量。
 - 將 NewAvailabilityZones (CLI : - -new-availability-zones) 設定為您要在其中建立新快取節點的區域清單。若要讓 ElastiCache 決定每個新節點的可用區域，請勿指定清單。
 - 將 ApplyImmediately (CLI : - -apply-immediately) 設為 true。

Note

如果您沒有使用自動探索功能，請務必使用新的快取節點端點來更新用戶端應用程式。

繼續進行下一個步驟之前，請確定 Memcached 節點已完全建立且可用。

2. 移除原始可用區域中不再需要的節點，藉此修改叢集。在請求中執行下列操作：
 - 套用此修改後，將 NumCacheNodes (CLI : - -num-cache-nodes) 設定為您想要的作用中快取節點數量。
 - 將 CacheNodeIdsToRemove (CLI : - -nodes-to-remove) 設定為您要從叢集中移除的快取節點清單。

IDs 列出的快取節點數目必須等於目前作用中節點數目減去 中的值 NumCacheNodes。

- (選用) 將 ApplyImmediately (CLI : - -apply-immediately) 設為 true。

如果您未將 ApplyImmediately (CLI : - -apply-immediately) 設為 true，節點刪除會在下一個維護時段進行。

安置您的節點

Amazon ElastiCache 支援在單一或多個可用區域 () 中定位叢集的所有節點AZs。此外，如果您選擇在多個 AZs (建議) 中尋找節點，ElastiCache 您可以為每個節點選擇 AZ，或允許為您ElastiCache 選擇它們。

透過將節點放在不同的 AZs 中，您消除了單一 AZ 中發生故障的可能性，例如停電，將導致整個系統故障。測試已證明，在單一 AZ 中定位所有節點或將節點分散到多個 AZs 之間沒有顯著的延遲差異。

您可以在建立叢集時為每個節點指定 AZ，或在修改現有叢集時新增節點來完成此作業。如需詳細資訊，請參閱下列內容：

- [為 Memcached 建立叢集](#)
- [為 Valkey 或 Redis 建立叢集 OSS](#)
- [修改 ElastiCache 叢集](#)
- [將節點新增至 ElastiCache 叢集](#)

支援的區域和端點

Amazon ElastiCache 可在多個 AWS 區域中使用。這表示您可以在符合您需求的位置啟動 ElastiCache 叢集。例如，您可以在最接近客戶的 AWS 區域中啟動，或在特定 AWS 區域中啟動，以符合特定法律要求。

每個區域皆設計為與其他區域完全隔離。各個區域包含多個可用區域 (AZ)。ElastiCache 無伺服器快取會自動跨多個可用區域複寫資料 (除了 us-west-1，其中資料會在兩個可用區域中複寫)，以取得高可用性。設計自己的 ElastiCache 叢集時，您可以選擇在不同的 AZs 中啟動節點，以達到容錯能力。如需區域和可用區域的詳細資訊，請參閱本主題最上方的 [選擇的區域和可用區域 ElastiCache](#)。

ElastiCache 支援的區域

區域名稱/區域	端點	通訊協定
美國東部 (俄亥俄) 區域 us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
美國東部 (維吉尼亞北部) 區域 us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS
美國西部 (加利佛尼亞北部) 區域 us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS
美國西部 (奧勒岡) 區域 us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS
加拿大 (中部) 區域 ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS
加拿大 (西部) 區域 ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS
亞太區域 (雅加達) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS
亞太 (孟買) 區域 ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS
亞太區域 (海德拉巴) ap-south-2	elasticache.ap-south-2.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
亞太區域 (東京) 區域 ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS
亞太區域 (首爾) 區域 ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS
亞太 (大阪) 區域 ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS
亞太區域 (新加坡) 區域 ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS
亞太區域 (雪梨) 區域 ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS
歐洲 (法蘭克福) 區域 eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS
歐洲 (蘇黎世) 區域 eu-central-2	elasticache.eu-central-2.amazonaws.com	HTTPS
歐洲 (斯德哥爾摩) 區域 eu-north-1	elasticache.eu-north-1.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
Middle East (Bahrain) Region me-south-1	elasticache.me-south-1.amazonaws.com	HTTPS
中東 (UAE) 區域 me-central-1	elasticache.me-central-1.amazonaws.com	HTTPS
歐洲 (愛爾蘭) 區域 eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS
歐洲 (倫敦) 區域 eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS
歐洲 (巴黎) 區域 eu-west-3	elasticache.eu-west-3.amazonaws.com	HTTPS
Europe (Milan) Region eu-south-1	elasticache.eu-south-1.amazonaws.com	HTTPS
歐洲 (西班牙) 區域 eu-south-2	elasticache.eu-south-2.amazonaws.com	HTTPS
南美洲 (聖保羅) 區域 sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS
中國 (北京) 區域 cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS

區域名稱/區域	端點	通訊協定
中國 (寧夏) 區域 cn-northwest-1	elasticache.cn- northwest-1. amazonaws .com.cn	HTTPS
亞太區域 (香港) 區域 ap-east-1	elasticache.ap- east-1.amazo naws.com	HTTPS
非洲 (開普敦) 區域 af-south-1	elasticache.af- south-1.amaz onaws.com	HTTPS
以色列 (特拉維夫) 區 域 il-central-1	elasticache.il- central-1.am azonaws.com	HTTPS
AWS GovCloud (美 國西部) us-gov-west-1	elasticache.us- gov-west-1.a mazonaws.com	HTTPS
AWS GovCloud (美 國東部) us-gov-east-1	elasticache.us- gov-east-1.a mazonaws.com	HTTPS

如需搭配使用 AWS GovCloud (美國) 的資訊 ElastiCache，請參閱 [AWS GovCloud \(美國\) 區域中的服務：ElastiCache](#)。

有些區域支援一部分的節點類型。如需依 AWS 區域列出的支援節點類型資料表，請參閱 [支援的節點類型 \(依 AWS 區域\)](#)。

如需依區域區分的 AWS 產品和服務資料表，請參閱 [依區域區分的產品和服務](#)。

搭配 使用本機區域 ElastiCache

本機區域是地理位置接近您使用者的 AWS 區域延伸。您可以透過建立新的子網路並將其指派給 Local Zone，將任何虛擬私有雲端（VPC）從父 AWS 區域延伸到 Local Zones。當您在 Local Zone 中建立子網路時，您的 VPC 會延伸至該 Local Zone。本機區域中的子網路運作方式與 中的其他子網路相同 VPC。

透過使用 Local Zones，您可以將 ElastiCache 叢集等資源放置在靠近使用者的位置。

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

區域由 AWS 區域代碼表示，後面接著表示位置的識別符，例如 us-west-2-lax-1a。

目前可用的本地區域有 us-west-2-lax-1a 和 us-west-2-lax-1b。

下列限制適用於 ElastiCache Local Zones：

- 不支援全域資料存放區。
- 不支援線上遷移。
- 本地區域目前支援以下節點類型：
 - 最新一代：

M5 節點類

型：cache.m5.large、cache.m5.xlarge、cache.m5.2xlarge、cache.m5.4xlarge、cache.

R5 節點類

型：cache.r5.large、cache.r5.xlarge、cache.r5.2xlarge、cache.r5.4xlarge、cache.

T3 節點類型：cache.t3.micro、cache.t3.small、cache.t3.medium

啟用本地區域

1. 在 Amazon EC2 主控台中啟用 Local Zone。

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

2. 在 Local Zone 中建立子網。

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

3. 在 Local Zone 中建立 ElastiCache 子網路群組。

當您建立 ElastiCache 子網路群組時，請選擇本機區域的可用區域群組。

如需詳細資訊，請參閱[建立子網路群組](#)。

4. 建立在 Local Zone 中使用 ElastiCache 子網路的 ElastiCache (Memcached) 叢集。

如需詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

5. 在 Local Zone 中建立使用 ElastiCache 子網路的 ElastiCache (Redis OSS) 叢集。如需詳細資訊，請參閱下列其中一個主題：

- [建立 Valkey \(停用叢集模式 \) 叢集 \(主控台 \)](#)
- [建立 Valkey 或 Redis OSS \(啟用叢集模式 \) 叢集 \(主控台 \)](#)

搭配使用 Outposts ElastiCache

您可以搭配使用 AWS Outposts ElastiCache。Outposts 是完全受管的服務，可將 AWS 基礎設施、服務 APIs、和工具擴展到客戶內部部署。透過提供 AWS 受管基礎設施的本機存取權，AWS Outposts 可讓客戶使用與 AWS 區域相同的程式設計介面在內部部署中建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。Outpost 是部署在客戶站台的 AWS 運算和儲存容量集區。作為 AWS 區域的一部分 AWS 操作、監控和管理此容量。您可以在 Outpost 上建立子網路，並在建立 ElastiCache 叢集等 AWS 資源時指定子網路。

Note

此版本具有下列限制：

- ElastiCache for Outposts 僅支援 M5 和 R5 節點系列。
- 多可用區域 (不支援跨 Outpost 複寫)。
- 不支援即時遷移。
- 不支援本機快照。
- 引擎日誌和慢速日誌無法啟用。
- ElastiCache on Outposts 不支援 CoIP 。
- ElastiCache 下列區域不支援 for Outposts：cn-north-1、cn-northwest-1 和 ap-northeast-3。

搭配 ElastiCache 主控台使用 Outposts

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Valkey 快取、Redis OSS快取 或 Memcached 快取。
3. 如果您選擇 Valkey 快取，請選取建立 Valkey 快取。如果您選擇 Redis OSS快取，請選取建立 Redis OSS快取。如果您選擇 Memcached 快取，請選取建立 Memcached 快取。
4. 在叢集設定下，選取設計您自己的快取和叢集快取。將叢集模式設定為已停用。然後建立快取的名稱和選用描述。
5. 針對位置，選擇內部部署。
6. 在內部部署區段中，您會看到欄位 Outpost ID。輸入叢集執行位置的 ID。

叢集設定下的所有進一步設定都可以保留為預設值。

7. 在連線中，選取建立新的子網路群組，然後輸入 VPC ID。將其餘部分保留為預設值，然後選取下一步。

設定內部部署選項

您可以選取可用的 Outpost 來新增快取叢集，或者如果沒有可用的 Outpost，請使用下列步驟建立新的 Outpost：

在 On-Premises options (內部部署選項) 底下：

1. 在 Valkey 設定、Redis OSS設定 或 Memcached 設定下，視您偏好的引擎而定：
 - a. 名稱：輸入叢集的名稱
 - b. 描述：輸入叢集的描述。
 - c. 引擎版本相容性：引擎版本是以 AWS Outpost 區域為基礎
 - d. 連接埠：對於 Valkey 或 Redis OSS，接受預設連接埠 6379。對於 Memcached，接受預設連接埠 11211。如果您偏好使用不同的連接埠，請輸入連接埠號碼。
 - e. Parameter group (參數群組)：使用下拉式選單選取預設或自訂參數群組。
 - f. Node Type (節點類型)：可用的執行個體取決於 Outpost 可用性。如果使用 Valkey 或 Redis OSS，則的 Porting Assistant for。NET for Outposts 僅支援 M5 和 R5 節點系列。從下拉式清單中，選取 Outposts，然後選取您要用於此叢集的可用節點類型。然後選取 Save (儲存)。
 - g. Number of Replicas (複本數量)：輸入您要讓此複寫群組建立的僅供讀取複本數目。您必須至少有一個僅供讀取複本，且不能超過五個。預設值為 2。

僅供讀取複本的自動產生名稱會遵循與主叢集名稱相同的模式，結尾會加上破折號和連續的三位數字，開頭為 -002。例如，如果您的複寫群組名為 MyGroup，那麼次要複本的名稱就會是 MyGroup-002、MyGroup-003、MyGroup-004、MyGroup-005、MyGroup-006。

2. 在連線下：

a. Subnet Group (子網路群組)：從清單中選取 Create new (新建)。

- Name (名稱)：輸入子網路群組的名稱
- Description (描述)：輸入子網路群組的描述
- VPC ID：VPCID 應與 Outpost 相符VPC。如果您選擇 Outposts IDs上VPC沒有子網路的，則清單會傳回空白。
- Availability Zone or Outpost (可用區域或 Outpost)：選取您正在使用的 Outpost。
- Subnet ID (子網路 ID)：選取可用於 Outpost 的子網路 ID。如果沒有IDs可用的子網路，則需要建立它們。如需詳細資訊，請參閱「[建立子網路](#)」。

b. 選取建立。

檢視 Outpost 叢集詳細資訊

在清單頁面上，選取屬於 AWS Outpost 的叢集，並在檢視叢集詳細資訊時注意下列事項：

- 可用區域：這將代表 Outpost，使用 ARN (Amazon Resource Name) 和 AWS 資源編號。
- Outpost 名稱：AWS Outpost 的名稱。

搭配使用 Outposts AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 從命令列控制多個 AWS 服務，並透過指令碼自動化。您可以使用 AWS CLI 進行臨時 (一次性) 操作。

下載和設定 AWS CLI

AWS CLI 會在 Windows、macOS 或 Linux 上執行。請使用下列程序來下載並進行設定。

若要下載、安裝和設定 CLI

1. 在[AWS 命令列介面](#)網頁上下載 AWS CLI。
2. 請遵循 AWS Command Line Interface 使用者指南 [中的安裝 AWS CLI](#) 和 [設定 AWS CLI](#) 的指示。

搭配 Outposts 使用 AWS CLI

使用下列CLI操作建立使用 Outposts 的快取叢集：

- [create-cache-cluster](#) – 使用此操作，`outpost-mode` 參數接受一個值，指定快取叢集中的節點是在單一 Outpost 中建立還是跨多個 Outpost 建立。

Note

目前只支援 `single-outpost` 模式。

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id cache cluster id \  
  --outpost-mode single-outpost \  
  \
```

使用 ElastiCache

在本節中，您可以找到有關如何管理 ElastiCache 實作各種元件的詳細資訊。

主題

- [快照和還原](#)
- [在 中的引擎版本和升級 ElastiCache](#)
- [ElastiCache 最佳實務和快取策略](#)
- [在 中管理自行設計的叢集 ElastiCache](#)
- [擴展 ElastiCache](#)
- [適用於 Valkey 和 Redis JSON 的 入門 OSS](#)
- [標記您的 ElastiCache 資源](#)
- [使用 Amazon ElastiCache Well-Architected Lens](#)
- [使用的常見疑難排解步驟和最佳實務 ElastiCache](#)

快照和還原

執行 Valkey、Redis OSS 或 Serverless Memcached 的 Amazon ElastiCache 快取可以透過建立快照來備份其資料。您可以使用備份來還原快取或將資料植入新的快取。備份包含快取的中繼資料，以及快取中的所有資料。所有備份都會寫入 Amazon Simple Storage Service (Amazon S3)，該服務提供耐久性儲存空間。您可以隨時透過建立新的 Valkey、Redis 或 Serverless Memcached 快取來還原資料 OSS，並將其填入來自備份的資料。透過 ElastiCache，您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 和管理備份 ElastiCache API。

如果您打算刪除快取，而且有必要保留資料，您可以採取額外的預防措施。若要達成此目標，請先建立手動備份、確認其狀態為可用，然後刪除快取。這樣做可確保備份失敗時，快取資料仍然可用。您可以依照前述的最佳實務，重試進行備份。

主題

- [備份限制](#)
- [自行設計叢集之備份的效能影響](#)
- [排程自動備份](#)
- [取得手動備份](#)

- [建立最終備份](#)
- [描述備份](#)
- [複製備份](#)
- [匯出備份](#)
- [從備份還原到新的快取](#)
- [刪除備份](#)
- [標記備份](#)
- [教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)

備份限制

規劃或建立備份時，請考慮下列限制：

- 只有在 Valkey、Redis OSS 或 Serverless Memcached 上執行的快取才支援備份和還原。
- 對於 Valkey 或 Redis OSS (停用叢集模式) 叢集，cache.t1.micro 節點不支援備份和還原。所有其他快取節點類型則受到支援。
- 對於 Valkey 或 Redis OSS (啟用叢集模式) 叢集，所有節點類型都支援備份和還原。
- 在任何連續的 24 小時期間內，每個無伺服器快取建立的手動備份不得超過 24 個。對於 Valkey 和 Redis OSS 自行設計的叢集，您可以在叢集中為每個節點建立不超過 20 個手動備份。
- Valkey 或 Redis OSS (啟用叢集模式) 僅支援在叢集層級 (適用於 API 或 CLI 的複寫群組層級) 上進行備份。Valkey 或 Redis OSS (啟用叢集模式) 不支援在碎片層級 (適用於 API 或 CLI 的節點群組層級) 進行備份。
- 在備份過程中，您無法在無伺服器快取上執行任何其他 API 或 CLI 操作。您可以在備份期間在自我設計的叢集上執行 API 或 CLI 操作。
- 如果您使用具有資料分層的 Valkey 或 Redis OSS 快取，則無法將備份匯出至 Amazon S3。
- 只能將使用 r6gd 節點類型的叢集備份還原到使用 r6gd 節點類型的叢集。

自行設計叢集之備份的效能影響

無伺服器快取上的備份對於應用程式來說公開透明，不會影響效能。然而，在建立自行設計叢集的備份時，根據可用的保留記憶體而定，可能會對效能造成某種程度的影響。自我設計叢集的備份不適用於 ElastiCache (Memcached)，但可用於 ElastiCache (Redis OSS)。

以下是提升自行設計叢集之備份效能的準則。

- 設定 `reserved-memory-percent` 參數 – 若要減少過多分頁，建議您設定 `reserved-memory-percent` 參數。此參數可防止 Valkey 和 Redis OSS 耗用節點的所有可用記憶體，並可協助減少分頁數量。您也可以直接使用更大的節點來提升效能。如需預留記憶體和 `reserved-memory-percent` 參數的詳細資訊，請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。
- 從僅供讀取複本建立備份：如果您在具有多個節點的節點群組 OSS 中執行 Valkey 或 Redis，您可以從主要節點或其中一個僅供讀取複本取得備份。由於期間需要的系統資源 BGSAVE，我們建議您從其中一個僅供讀取複本建立備份。從複本建立備份時，主要節點不會受到 BGSAVE 資源需求的影響。主要節點可繼續服務請求而不會變慢。

若要執行此操作，請參閱「[建立手動備份 \(主控台\)](#)」，並在 Create Backup (建立備份) 視窗裡的 Cluster Name (叢集名稱) 欄位中，選擇複本而非預設主節點。

如果您刪除複寫群組並請求最終備份，ElastiCache 請務必從主要節點取得備份。這可確保在刪除複寫群組之前，先擷取最新的 Valkey 或 Redis OSS 資料。

排程自動備份

您可以為任何 Valkey 或 Redis 無OSS伺服器快取或自行設計的叢集啟用自動備份。啟用自動備份時，會每天 ElastiCache 建立快取的備份。不會對快取造成任何影響，而且變更會立即生效。自動備份可協助防止資料遺失。如果發生失敗，您可以建立新的快取，從最近的備份還原您的資料。這會產生暖啟動的快取，並預先載入您的資料以供使用。如需詳細資訊，請參閱[從備份還原到新的快取](#)。

您可以為任何 Memcached Serverless 快取啟用自動備份。啟用自動備份時，會每天 ElastiCache 建立快取的備份。不會對快取造成任何影響，而且變更會立即生效。自動備份可協助防止資料遺失。如果發生失敗，您可以建立新的快取，從最近的備份還原您的資料。這會產生暖啟動的快取，並預先載入您的資料以供使用。如需詳細資訊，請參閱[從備份還原到新的快取](#)。

當您排程自動備份時，您應該規劃下列設定：

- 備份開始時間 – ElastiCache 開始建立備份時的一天中時間。您可以將備份時段設定在最方便的任何時間。如果您未指定備份時段，會自動 ElastiCache 指派備份時段。
- 備份保留期限 - 要在 Amazon S3 中保留備份的天數。例如，如果您將保留期限設為 5，則今天進行的備份會保留 5 天。超過保留期限之後，則會自動刪除備份。

保留期上限為 35 天。如果將備份保留期限設為 0，則會停用快取的自動備份。

當您排程自動備份時，ElastiCache 將開始建立備份。您可以將備份時段設定在最方便的任何時間。如果您未指定備份時段，會自動 ElastiCache 指派備份時段。

您可以使用 ElastiCache 主控台 AWS CLI、或 [AWS Management Console](#)，在建立新的快取或更新現有快取時啟用或停用自動備份 ElastiCache API。對於 Valkey 和 Redis OSS，方法是勾選進階 Valkey 設定或進階 Redis OSS 設定區段中的啟用自動備份方塊。對於 Memcached，方法是勾選進階 Memcached 設定區段中的啟用自動備份方塊。

取得手動備份

除了自動備份，您還可以隨時建立「手動」備份。不同於自動備份會在指定的保留期之後自動刪除，手動備份沒有過了一段時間會自動刪除的保留期。即使您刪除快取，該快取中的任何手動備份仍會保留。如果您不想再保留手動備份，則必須自行明確予以刪除。

除了直接建立手動備份，您還可以利用下列其中一種方式來建立手動備份：

- [複製備份](#)。與來源備份是自動或手動建立無關。
- [建立最終備份](#)。在刪除叢集或節點前立即建立一個備份。

您可以使用 AWS Management Console、AWS CLI 或 建立快取的手動備份 ElastiCache API。

建立手動備份 (主控台)

建立快取的備份 (主控台)

1. 登入 AWS Management Console 並在 開啟 Amazon EC2主控台<https://console.aws.amazon.com/ec2/>。
2. 從導覽窗格中，根據您的偏好設定，選擇 Valkey 快取、Redis OSS快取 或 Memcached 快取。
3. 選擇您要備份之快取名稱左側的方塊。
4. 選擇 Backup (備份)。
5. 在 Create Backup (建立備份) 對話方塊的 Backup Name (備份名稱) 方塊中，輸入您的備份名稱。我們建議名稱指出備份的叢集，以及建立備份的日期和時間。

叢集命名限制條件如下：

- 必須包含 1-40 個英數字元或連字號。
 - 必須以字母開頭。
 - 不能連續包含兩個連字號。
 - 結尾不能是連字號。
6. 選擇 Create Backup (建立備份)。

叢集的狀態會變更為「快照中」。

建立手動備份 (AWS CLI)

使用 手動備份無伺服器快取 AWS CLI

若要使用 建立快取的手動備份 AWS CLI，請使用 `create-serverless-snapshot` AWS CLI 操作搭配下列參數：

- `--serverless-cache-name` - 您要備份之無伺服器快取的名稱。
- `--serverless-cache-snapshot-name` - 要建立的快照名稱。

若為 Linux、macOS 或 Unix：

- ```
aws elasticache create-serverless-snapshot \
 --serverless-cache-name CacheName \
 --serverless-cache-snapshot-name bkup-20231127
```

針對 Windows：

- ```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

使用 手動備份自我設計叢集 AWS CLI

若要使用 建立自訂叢集的手動備份 AWS CLI，請使用 `create-snapshot` AWS CLI 操作搭配下列參數：

- `--cache-cluster-id`
 - 如果您要備份的叢集沒有複本節點，`--cache-cluster-id`就是您要備份的叢集名稱，例如 *mycluster*。
 - 如果您要備份的叢集含有一或多個複本節點，`--cache-cluster-id`是您要用於備份之叢集中的節點名稱。例如，名稱可能是 *mycluster-002*。

只有在備份 Valkey 或 Redis OSS (停用叢集模式) 叢集時，才使用此參數。

- `--replication-group-id` – 用作備份來源的 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (CLI/API : 複寫群組) 的名稱。備份 Valkey 或 Redis OSS (啟用叢集模式) 叢集時，請使用此參數。
- `--snapshot-name` - 要建立的快照名稱。

叢集命名限制條件如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

範例 1：備份沒有複本節點的 Valkey 或 Redis OSS (停用叢集模式) 叢集

下列 AWS CLI 操作 `bkup-20150515` 會從 `myNonClusteredRedis` 沒有僅供讀取複本的 Valkey 或 Redis OSS (停用叢集模式) 叢集建立備份。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

針對 Windows：

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis ^  
  --snapshot-name bkup-20150515
```

範例 2：使用複本節點備份 Valkey 或 Redis OSS (停用叢集模式) 叢集

下列 AWS CLI 操作 `bkup-20150515` 會從 Valkey 或 Redis OSS (停用叢集模式) 叢集 建立備份 `myNonClusteredRedis`。此備份具有一個或多個僅供讀取複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis-001 \  
  --snapshot-name bkup-20150515
```



```
--snapshot-name bkup-20150515
```

針對 Windows :

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis-001 ^  
  --snapshot-name bkup-20150515
```

輸出範例：使用複本節點備份 Valkey 或 Redis OSS (停用叢集模式) 叢集

此操作的輸出將會如下所示。

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x",  
    "VpcId": "vpc-91280df6",  
    "CacheClusterId": "myNonClusteredRedis-001",  
    "SnapshotRetentionLimit": 0,  
    "NumCacheNodes": 1,  
    "SnapshotName": "bkup-20150515",  
    "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",  
    "AutoMinorVersionUpgrade": true,  
    "PreferredAvailabilityZone": "us-east-1c",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "08:30-09:30",  
    "EngineVersion": "6.0",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "CacheNodeId": "0001",  
        "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"  
      }  
    ],  
    "CacheSubnetGroupName": "default",  
    "Port": 6379,  
    "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",  
    "CacheNodeType": "cache.m3.2xlarge",  
    "DataTiering": "disabled"  
  }  
}
```

範例 3：備份 Valkey 或 Redis 的叢集 OSS (已啟用叢集模式)

下列 AWS CLI 操作 `bkup-20150515` 會從 Valkey 或 Redis OSS (啟用叢集模式) 叢集 建立備份 `myClusteredRedis`。注意使用 `--replication-group-id` (而非 `--cache-cluster-id`) 識別來源。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-snapshot \  
  --replication-group-id myClusteredRedis \  
  --snapshot-name bkup-20150515
```

針對 Windows：

```
aws elasticache create-snapshot ^  
  --replication-group-id myClusteredRedis ^  
  --snapshot-name bkup-20150515
```

輸出範例：備份 Valkey 或 Redis OSS (啟用叢集模式) 叢集

此操作的輸出將會如下所示。

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x.cluster.on",  
    "VpcId": "vpc-91280df6",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0001"  
      },  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0002"  
      }  
    ],  
    "NumNodeGroups": 2,  
    "SnapshotName": "bkup-20150515",  
    "ReplicationGroupId": "myClusteredRedis",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 1,  
  }  
}
```

```
"AutomaticFailover": "enabled",
"SnapshotStatus": "creating",
"SnapshotSource": "manual",
"SnapshotWindow": "10:00-11:00",
"EngineVersion": "6.0",
"CacheSubnetGroupName": "default",
"ReplicationGroupDescription": "2 shards 2 nodes each",
"Port": 6379,
"PreferredMaintenanceWindow": "sat:03:30-sat:04:30",
"CacheNodeType": "cache.r3.large",
"DataTiering": "disabled"
}
}
```

相關主題

如需詳細資訊，請參閱 AWS CLI 命令參考中的 [create-snapshot](#)。

建立最終備份

您可以使用 ElastiCache 主控台、AWS CLI 或 建立最終備份 ElastiCache API。

建立最終備份 (主控台)

您可以使用 ElastiCache 主控台，在刪除 Valkey 或 Redis 無OSS伺服器快取、Valkey 或 Redis OSS 自行設計的叢集，或 Memcached 無伺服器快取時建立最終備份。

若要在刪除快取時建立最終備份，請在刪除對話方塊的建立備份下選擇是，並為備份命名。

相關主題

- [使用 AWS Management Console](#)
- [刪除複寫群組 \(主控台\)](#)

建立最終備份 (AWS CLI)

您可以使用 刪除快取時建立最終備份 AWS CLI。

主題

- [刪除 Valkey 快取、Redis OSS快取或 Memcached 無伺服器快取時](#)
- [刪除沒有僅供讀取複本的 Valkey 或 Redis OSS叢集時](#)
- [刪除具有僅供讀取複本的 Valkey 或 Redis OSS叢集時](#)

刪除 Valkey 快取、Redis OSS快取或 Memcached 無伺服器快取時

若要建立最終備份，請使用 delete-serverless-cache AWS CLI 操作搭配下列參數。

- --serverless-cache-name - 要刪除的快取名稱。
- --final-snapshot-name - 備份的名稱。

下列程式碼會在刪除快取 myserverlesscache 時建立最終備份 bkup-20231127-final。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-serverless-cache \  
    --serverless-cache-name myserverlesscache \  
    --final-snapshot-name bkup-20231127-final
```

```
--final-snapshot-name bkup-20231127-final
```

針對 Windows :

```
aws elasticache delete-serverless-cache ^  
  --serverless-cache-name myserverlesscache ^  
  --final-snapshot-name bkup-20231127-final
```

如需詳細資訊，請參閱 命令參考 [delete-serverless-cache](#) 中的 。 AWS CLI

刪除沒有僅供讀取複本的 Valkey 或 Redis OSS叢集時

若要為沒有僅供讀取複本自我設計叢集建立最終備份，請使用 delete-cache-cluster AWS CLI 操作搭配下列參數。

- --cache-cluster-id - 正在刪除的叢集名稱。
- --final-snapshot-identifier - 備份的名稱。

下列程式碼會在刪除叢集 myRedisCluster 時建立最終備份 bkup-20150515-final。

若為 Linux、macOS 或 Unix :

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id myRedisCluster \  
  --final-snapshot-identifier bkup-20150515-final
```

針對 Windows :

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id myRedisCluster ^  
  --final-snapshot-identifier bkup-20150515-final
```

如需詳細資訊，請參閱 命令參考 [delete-cache-cluster](#) 中的 。 AWS CLI

刪除具有僅供讀取複本的 Valkey 或 Redis OSS叢集時

若要在刪除複寫群組時建立最終備份，請使用 delete-replication-group AWS CLI 操作搭配下列參數：

- --replication-group-id - 正在刪除的複寫群組名稱。

- `--final-snapshot-identifier` - 最終備份的名稱。

下列程式碼會在刪除複寫群組 `myReplGroup` 時建立最終備份 `bkup-20150515-final`。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-replication-group \  
  --replication-group-id myReplGroup \  
  --final-snapshot-identifier bkup-20150515-final
```

針對 Windows：

```
aws elasticache delete-replication-group ^  
  --replication-group-id myReplGroup ^  
  --final-snapshot-identifier bkup-20150515-final
```

如需詳細資訊，請參閱 [命令參考 delete-replication-group](#) 中的。AWS CLI

描述備份

下列程序說明如何顯示您的備份清單。如果您需要，也可以檢視特定備份的詳細資訊。

描述備份 (主控台)

使用 顯示備份 AWS Management Console

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格，選擇 Backups (備份)。
3. 若要查看特定備份的詳細資訊，請選擇備份名稱左側的方塊。

說明無伺服器備份 (AWS CLI)

若要顯示無伺服器備份的清單，以及特定備份的選擇性詳細資訊，請使用 `describe-serverless-cache-snapshots` CLI 操作。

範例

下列操作使用參數 `--max-records` 列出最多 20 個與您的帳戶相關的備份。省略參數 `--max-records` 會列出最多 50 個備份。

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

下列操作使用參數 `--serverless-cache-name` 單獨列出與快取 `my-cache` 相關聯的備份。

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

下列操作使用參數 `--serverless-cache-snapshot-name` 顯示備份 `my-backup` 的詳細資訊。

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

如需詳細資訊，請參閱 AWS CLI 命令參考 [describe-serverless-cache-snapshots](#) 中的。

說明自行設計叢集備份 (AWS CLI)

若要顯示自行設計的叢集備份清單，以及特定備份的選擇性詳細資訊，請使用 `describe-snapshots` CLI 操作。

範例

下列操作使用參數 `--max-records` 列出最多 20 個與您的帳戶相關的備份。省略參數 `--max-records` 會列出最多 50 個備份。

```
aws elasticache describe-snapshots --max-records 20
```

下列操作使用參數 `--cache-cluster-id` 只列出與叢集 `my-cluster` 相關的備份。

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

下列操作使用參數 `--snapshot-name` 顯示備份 `my-backup` 的詳細資訊。

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

如需詳細資訊，請參閱 AWS CLI 命令參考中的 [describe-snapshots](#)。

複製備份

您可以建立任何備份的副本，不論是自動或手動建立的備份。您也可以匯出備份，以便從外部存取備份 ElastiCache。如需匯出備份的指導方針，請參閱[匯出備份](#)。

下列步驟說明如何複製備份。

複製備份 (主控台)

複製備份 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看您的備份清單，請從左側導覽窗格，選擇 Backups (備份)。
3. 從備份清單，選擇您要複製之備份名稱左側的方塊。
4. 選擇動作、複製。
5. 在 New backup name (新的備份名稱) 方塊中，輸入新的備份名稱。
6. 請選擇 Copy (複製)。

複製無伺服器備份 (AWS CLI)

若要複製無伺服器快取的備份，請使用 `copy-serverless-cache-snapshot` 操作。

參數

- `--source-serverless-cache-snapshot-name` - 要複製的備份名稱。
- `--target-serverless-cache-snapshot-name` - 備份的複本名稱。

下列範例會建立自動備份的複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache copy-serverless-cache-snapshot \  
  --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \  
  --target-serverless-cache-snapshot-name my-backup-copy
```

針對 Windows：

```
aws elasticache copy-serverless-cache-snapshot ^
  --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^
  --target-serverless-cache-snapshot-name my-backup-copy
```

如需詳細資訊，請參閱 AWS CLI 中的 [copy-serverless-cache-snapshot](#)。

複製自行設計的叢集備份 (AWS CLI)

若要複製自行設計叢集的備份，請使用 `copy-snapshot` 操作。

參數

- `--source-snapshot-name` - 要複製的備份名稱。
- `--target-snapshot-name` - 備份的複本名稱。
- `--target-bucket` - 保留以供匯出備份。建立備份複本時，請勿使用此參數。如需詳細資訊，請參閱 [匯出備份](#)。

下列範例會建立自動備份的複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \  
  --target-snapshot-name my-backup-copy
```

針對 Windows：

```
aws elasticache copy-snapshot ^
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^
  --target-snapshot-name my-backup-copy
```

如需詳細資訊，請參閱 AWS CLI 中的 [copy-snapshot](#)。

匯出備份

Amazon ElastiCache 支援將 ElastiCache (RedisOSS) 備份匯出至 Amazon Simple Storage Service (Amazon S3) 儲存貯體，可讓您從外部存取 ElastiCache。您可以使用 ElastiCache 主控台、AWS CLI或 匯出備份 ElastiCache API。

如果您需要在另一個 AWS 區域中啟動叢集，匯出備份會很有幫助。您可以在一個 AWS 區域中匯出資料，將 .rdb 檔案複製到新 AWS 區域，然後使用該 .rdb 檔案來植入新快取，而不是等待新叢集透過使用填入。如需植入新叢集的資訊，請參閱[教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。另一個您可能想要匯出快取資料的原因是使用 .rdb 檔案進行離線處理。

Important

- ElastiCache 備份和您要將其複製到的 Amazon S3 儲存貯體必須位於相同的 AWS 區域中。
雖然複製到 Amazon S3 儲存貯體的備份已經過加密，但強烈建議您不要授權其他人存取要存放備份的 Amazon S3 儲存貯體。
- 使用資料分層的叢集不支援將備份匯出到 Amazon S3。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。
- 匯出備份適用於 Valkey 和 Redis OSS自行設計的叢集、Serverless Valkey 和 Redis OSS以及 Serverless Memcached。匯出備份不適用於自行設計的 Memcached 叢集。

在將備份匯出至 Amazon S3 儲存貯體之前，您必須在與備份相同的 AWS 區域中擁有 Amazon S3 儲存貯體。授予對儲存貯體的 ElastiCache 存取權。前兩個步驟示範如何執行此操作。

建立 Amazon S3 儲存貯體

下列步驟使用 Amazon S3 主控台建立 Amazon S3 儲存貯體，您可以在其中匯出和存放 ElastiCache 備份。

建立 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。
2. 選擇 Create Bucket (建立儲存貯體)。
3. 在 Create a Bucket - Select a Bucket Name and Region (建立儲存貯體 - 選取儲存貯體名稱和區域) 中，執行下列動作：

- a. 在 Bucket Name (儲存貯體名稱) 中，輸入 Amazon S3 儲存貯體的名稱。

Amazon S3 儲存貯體的名稱必須符合 DNS。否則，ElastiCache 無法存取您的備份檔案。DNS 合規規則包括：

- 名稱長度須為 3 到 63 個字元。
 - 名稱必須是一連串一或多個標籤，並以句號 (.) 分隔，其中每個標籤：
 - 以小寫字母或數字開頭。
 - 以小寫字母或數字結尾。
 - 僅包含小寫字母、數字和破折號。
 - 不得使用 IP 地址格式 (例如 192.0.2.0)。
- b. 在區域清單中，為您的 Amazon S3 儲存貯體選擇 AWS 區域。此 AWS 區域必須與您要匯出的 ElastiCache 備份 AWS 區域相同。
 - c. 選擇 Create (建立)。

如需有關建立 Amazon S3 儲存貯體的詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的[建立儲存貯體](#)。

授予對 Amazon S3 儲存貯體的 ElastiCache 存取權

若要 ElastiCache 讓 能夠將快照複製到 Amazon S3 儲存貯體，您必須更新儲存貯體政策以授予儲存貯體的 ElastiCache 存取權。

Warning


即使複製到 Amazon S3 儲存貯體的備份已經過加密，有權存取 Amazon S3 儲存貯體的所有人還是可以存取您的資料。因此，強烈建議您設定 IAM 政策，以防止未經授權存取此 Amazon S3 儲存貯體。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[管理存取權](#)。

若要在 Amazon S3 儲存貯體上建立適當許可，請執行下列步驟。

授予對 S3 儲存貯體的 ElastiCache 存取權

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。

2. 選擇您要複製備份的目的地 Amazon S3 儲存貯體名稱。這應該是您在[建立 Amazon S3 儲存貯體](#)中建立的 S3 儲存貯體。
3. 選擇許可索引標籤，然後在許可下，選擇存取控制清單（ACL），然後選擇編輯。
4. 透過下列選項，新增承授者正規 ID
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353：
 - Objects (物件)：List (列出)、Write (寫入)
 - 儲存貯體ACL：讀取、寫入

 Note

- 對於 PDT GovCloud 區域，正式 ID 為
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6。
- 對於 OSU GovCloud 區域，正式 ID 為
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c。

5. 選擇 Save (儲存)。

匯出 ElastiCache 備份

現在，您已建立 S3 儲存貯體並授予存取它的 ElastiCache 許可。接下來，您可以使用 ElastiCache 主控台 AWS CLI、或 ElastiCache API 將快照匯出至其中。下列範例假設來電者的 IAM 身分具有下列額外的 S3 特定 IAM 許可。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
      "s3:PutObject",
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  }]
}
```

針對選擇加入區域，以下是 S3 儲存貯體已更新政策可能有的外觀範例。(以下範例使用亞太區域 (香港) 區域。)

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticache.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    },
    {
      "Sid": "Stmt15399484",
      "Effect": "Allow",
      "Principal": {
        "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    }
  ]
}
```

匯出 ElastiCache 備份 (主控台)

下列步驟使用 ElastiCache 主控台將備份匯出至 Amazon S3 儲存貯體，以便您可以從外部存取備份 ElastiCache。Amazon S3 儲存貯體必須與 ElastiCache 備份位於相同的 AWS 區域。

將 ElastiCache 備份匯出至 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看您的備份清單，請從左側導覽窗格，選擇 Backups (備份)。
3. 從備份清單，選擇您要匯出之備份名稱左側的方塊。
4. 請選擇 Copy (複製)。
5. 在 Create a Copy of the Backup? (是否建立備份複本?) 中，執行下列動作：

- a. 在 New backup name (新的備份名稱) 方塊中，輸入新的備份名稱。

名稱必須介於 1 到 1,000 個字元之間，並且能夠 UTF-8 編碼。

ElastiCache 會將執行個體識別碼和 .rdb 新增至您在此處輸入的值。例如，如果您輸入 my-exported-backup，ElastiCache 會建立 my-exported-backup-0001.rdb。

- b. 在 Target S3 Location (目標 S3 位置) 清單中，選擇您要複製備份的目的地 Amazon S3 儲存貯體名稱 (您在 [建立 Amazon S3 儲存貯體](#) 中建立的儲存貯體)。

目標 S3 位置必須是備份 AWS 區域中的 Amazon S3 儲存貯體，具有下列許可，匯出程序才能成功。

- 物件存取權 - Read (讀取) 和 Write (寫入)。
- 許可存取權 - Read (讀取)。

如需詳細資訊，請參閱[授予對 Amazon S3 儲存貯體的 ElastiCache 存取權](#)。

- c. 請選擇 Copy (複製)。

Note

如果您的 S3 儲存貯體沒有將備份 ElastiCache 匯出到其中所需的許可，您會收到下列其中一個錯誤訊息。請返回[授予對 Amazon S3 儲存貯體的 ElastiCache 存取權](#)，新增指定的許可並重新嘗試匯出您的備份。

- ElastiCache 尚未在 S3 儲存貯體上授予 READ 許可 %s。

解決方式：新增儲存貯體的 Read (讀取) 許可。

- ElastiCache 尚未在 S3 儲存貯體上授予 WRITE 許可 %s。

解決方式：新增儲存貯體的 Write (寫入) 許可。

- ElastiCache 尚未在 S3 儲存貯體上授予 READ_ACP 許可 %s。

解決方式：新增儲存貯體的 Read (讀取) 許可存取。

如果您想要將備份複製到另一個 AWS 區域，請使用 Amazon S3 將其複製。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的[複製物件](#)。

匯出無 ElastiCache 伺服器備份 (AWS CLI)

匯出無伺服器快取的備份

使用具有下列參數 `export-serverless-cache-snapshot` CLI 的操作，將備份匯出至 Amazon S3 儲存貯體：

參數

- `--serverless-cache-snapshot-name` - 要複製的備份名稱。
- `--s3-bucket-name` - 您要匯出備份的 Amazon S3 儲存貯體名稱。系統會在指定的儲存貯體中建立一個備份複本。

`--s3-bucket-name` 必須是備份 AWS 區域中的 Amazon S3 儲存貯體，具有下列匯出程序的許可，才能成功執行。

- 物件存取權 - Read (讀取) 和 Write (寫入)。
- 許可存取權 - Read (讀取)。

下列操作會將備份複製到 `my-s3-bucket`。

若為 Linux、macOS 或 Unix：

```
aws elasticache export-serverless-cache-snapshot \  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \  
  --s3-bucket-name my-s3-bucket
```

針對 Windows：

```
aws elasticache export-serverless-cache-snapshot ^
```



```
--serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^  
--s3-bucket-name my-s3-bucket
```

匯出自行設計的 ElastiCache 叢集備份 (AWS CLI)

匯出自行設計叢集的備份

使用具有下列參數 `copy-snapshot` CLI 的操作，將備份匯出至 Amazon S3 儲存貯體：

參數

- `--source-snapshot-name` - 要複製的備份名稱。
- `--target-snapshot-name` - 備份的複本名稱。

名稱必須介於 1 到 1,000 個字元之間，並且能夠 UTF-8 編碼。

ElastiCache 會將執行個體識別符和 `.rdb` 新增至您在此處輸入的值。例如，如果您輸入 `my-exported-backup`，ElastiCache 會建立 `my-exported-backup-0001.rdb`。

- `--target-bucket` - 您要匯出備份的 Amazon S3 儲存貯體名稱。系統會在指定的儲存貯體中建立一個備份複本。

`--target-bucket` 必須是備份 AWS 區域中的 Amazon S3 儲存貯體，具有下列匯出程序的許可，才能成功。

- 物件存取權 - Read (讀取) 和 Write (寫入)。
- 許可存取權 - Read (讀取)。

如需詳細資訊，請參閱 [授予對 Amazon S3 儲存貯體的 ElastiCache 存取權](#)。

下列操作會將備份複製到 `my-s3-bucket`。

若為 Linux、macOS 或 Unix：

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \  
  --target-snapshot-name my-exported-backup \  
  --target-bucket my-s3-bucket
```

針對 Windows：

```
aws elasticache copy-snapshot ^
```

```
--source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^  
--target-snapshot-name my-exported-backup ^  
--target-bucket my-s3-bucket
```

從備份還原到新的快取

您可以將現有的備份從 Valkey 還原至新的 Valkey 快取或自我設計的叢集，並將現有的 Redis OSS 備份還原至新的 Redis OSS快取或自我設計的叢集。您也可以將現有的 Memcached 無伺服器快取記憶體還原至新的 Memcached 無伺服器快取。

將備份還原到無伺服器快取 (主控台)

Note

ElastiCache Serverless 支援與 Valkey 7.2 及更高OSS版本相容的RDB檔案，以及介於 5.0 和最新版本之間的 Redis 版本。

將備份還原到無伺服器快取 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 從導覽窗格，選擇 Backups (備份)。
3. 在備份清單中，選擇您要還原之備份名稱左側的方塊。
4. 選擇動作，然後選擇還原。
5. 輸入新的無伺服器快取的名稱，以及選用的說明。
6. 按一下建立以建立新的快取，並從備份匯入資料。

將備份還原到自行設計的叢集 (主控台)

將備份還原到自行設計的叢集 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 從導覽窗格，選擇 Backups (備份)。
3. 在備份清單中，選擇您要從中還原之備份名稱左側的方塊。
4. 選擇動作，然後選擇還原。
5. 選擇設計您自己的快取並自訂叢集設定，例如節點類型、大小、碎片數量、複本、AZ 放置和安全性設定。
6. 選擇建立以建立新的自行設計快取，並從備份匯入資料。

將備份還原到無伺服器快取 (AWS CLI)

Note

ElastiCache Serverless 支援與 Valkey 7.2 及更高OSS版本相容的RDB檔案，以及介於 5.0 和最新版本之間的 Redis 版本。

將備份還原到新的無伺服器快取 (AWS CLI)

下列 AWS CLI 範例使用 建立新的快取`create-serverless-cache`，並從備份匯入資料。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-serverless-cache \  
  
    --serverless-cache-name CacheName \  
    --engine redis \  
    --snapshot-arns-to-restore Snapshot-ARN
```

針對 Windows：

```
aws elasticache create-serverless-cache ^ \  
  
    --serverless-cache-name CacheName ^ \  
    --engine redis ^ \  
    --snapshot-arns-to-restore Snapshot-ARN
```

將備份還原到自行設計的叢集 (AWS CLI)

將備份還原到自行設計的叢集 (AWS CLI)

您可以透過兩種方式還原 Valkey 或 Redis OSS (停用叢集模式) 備份。

- ```
aws elasticache create-serverless-cache \
 --serverless-cache-name CacheName \
 --engine redis \
 --snapshot-arns-to-restore Snapshot-ARN
```

- 針對 Windows：

```
aws elasticache create-serverless-cache ^
```

```
--serverless-cache-name CacheName ^
--engine redis ^
--snapshot-arns-to-restore Snapshot-ARN
```

將備份還原到自行設計的叢集 (AWS CLI)

將備份還原到自行設計的叢集 (AWS CLI)

您可以還原 Valkey 或 Redis 無OSS伺服器快取備份，也可以還原 Valkey 或 Redis OSS自行設計的叢集。

您可以透過兩種方式還原 Valkey 或 Redis 無OSS伺服器快取備份。

- 您可以使用 AWS CLI 操作還原至單節點 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集 `create-cache-cluster`。
- 您可以使用僅供讀取複本 ( 複寫群組 ) 還原至 Valkey 或 Redis OSS叢集。若要這麼做，您可以搭配 AWS CLI 操作使用 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 Redis OSS ( 啟用叢集模式 ) `create-replication-group`。在此情況下，您可以使用 Valkey 或 Redis OSS `.rdb` 檔案植入還原。如需植入新的自行設計叢集的詳細資訊，請參閱 [教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。

您可以透過兩種方式還原 Valkey 或 Redis OSS ( 停用叢集模式 ) 備份。

- 您可以使用 AWS CLI 操作還原至單節點 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集 `create-cache-cluster`。
- 您可以使用僅供讀取複本 ( 複寫群組 ) 還原至 Valkey 或 Redis OSS叢集。若要這麼做，您可以搭配 AWS CLI 操作使用 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 Redis OSS ( 啟用叢集模式 ) `create-replication-group`。在此情況下，您可以使用 Valkey 或 Redis OSS `.rdb` 檔案植入還原。如需植入新的自行設計叢集的詳細資訊，請參閱 [教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。

使用 `create-cache-cluster` 或 `create-replication-group` 操作時，請務必包含參數 `--snapshot-name` 或 `--snapshot-arn`，以將備份資料植入新叢集或複寫群組。

## 刪除備份

自動備份會在超過保留期限之後自動刪除。如果您刪除叢集，也會刪除其所有自動備份。如果您刪除複寫群組，也會從該群組中的叢集刪除所有自動備份。

ElastiCache 提供刪除API操作，可讓您隨時刪除備份，無論備份是自動還是手動建立。由於手動備份沒有留存期限，因此手動刪除是唯一的移除方式。

您可以使用 ElastiCache 主控台、AWS CLI或 刪除備份 ElastiCache API。

### 刪除備份 (主控台)

下列程序會使用 ElastiCache 主控台刪除備份。

#### 刪除備份

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇備份。

Backups (備份) 畫面會隨即顯示，其中包含您的備份清單。

3. 選擇您要刪除之備份名稱左側的方塊。
4. 選擇 刪除。
5. 如果要刪除此備份，請在 Delete Backup (刪除備份) 確認畫面上選擇 Delete (刪除)。狀態會變更為「刪除中」。

### 刪除無伺服器備份 (AWS CLI)

使用 delete-snapshot AWS CLI 操作搭配下列參數來刪除無伺服器備份。

- `--serverless-cache-snapshot-name` - 要刪除的備份名稱。

下列程式碼會刪除備份 myBackup。

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

如需詳細資訊，請參閱 命令參考 [delete-serverless-cache-snapshot](#) 中的。AWS CLI

### 複製自行設計的叢集備份 (AWS CLI)

使用 delete-snapshot AWS CLI 操作搭配下列參數來刪除自行設計的叢集備份。

- `--snapshot-name` - 要刪除的備份名稱。

下列程式碼會刪除備份 myBackup。

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

如需詳細資訊，請參閱 AWS CLI 命令參考中的 [delete-snapshot](#)。

## 標記備份

您可以用標籤的形式將自己的中繼資料指派給每個備份。標籤可讓您以不同的方式分類備份，例如依據目的、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為—您可以依據先前指派的標籤，快速識別特定的資源。如需詳細資訊，請參閱[您可以標記的資源](#)。

成本分配標籤是追蹤跨多個 AWS 服務成本的一種方式，方法是將發票上的費用依標籤值分組。若要進一步了解成本分配標籤，請參閱[使用成本分配標籤](#)。

使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 您可以在備份上新增、列出、修改、移除或複製成本分配標籤。如需詳細資訊，請參閱[使用成本配置標籤監控成本](#)。

## 教學課程：使用外部建立的備份來尋找新的自我設計叢集

當您建立新的 Valkey 或 Redis OSS 自行設計的叢集時，您可以使用 Valkey 或 Redis OSS .rdb 備份檔案的資料來植入叢集。如果您目前在外部管理 Valkey 或 Redis OSS 執行個體，ElastiCache 並想要使用現有的 Valkey 或 Redis OSS 資料填入新的 ElastiCache (RedisOSS) 自我設計叢集，則查看叢集非常有用。

若要從在 Amazon 中建立的 Valkey 或 Redis OSS 備份植入新的 Valkey 或 Redis OSS 自行設計叢集 ElastiCache，請參閱 [從備份還原到新的快取](#)。

當您使用 Valkey 或 Redis OSS .rdb 檔案植入新的自我設計叢集時，您可以執行下列動作：

- 從非分割叢集升級至執行 Redis 3.2.4 OSS 版的 Valkey 或 Redis OSS (啟用叢集模式) 自我設計的叢集。
- 在新的自我設計叢集中指定許多碎片 (在 API 和中稱為節點群組 CLI)。這個數字可以與用來建立備份檔案之自行設計叢集中的碎片數不同。
- 為新的自行設計叢集指定不同的節點類型 - 大於或小於叢集中用來建立備份的節點。如果您擴展到較小的節點類型，請確定新的節點類型有足夠的記憶體來儲存資料和 Valkey 或 Redis OSS 額外負荷。如需詳細資訊，請參閱 [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)。
- 在新 Valkey 或 Redis OSS (啟用叢集模式) 叢集的插槽中分發金鑰的方式，與用來建立備份檔案的叢集不同。

### Note

您無法從 Valkey 或 Redis OSS (啟用叢集模式) 叢集建立的 .rdb 檔案植入 Valkey 或 Redis OSS (停用叢集模式) 叢集。

### Important

- 您必須確保 Valkey 或 Redis OSS 備份資料不超過節點的資源。例如，您無法將具有 5 GB Valkey 或 Redis OSS 資料的 .rdb 檔案上傳至具有 2.9 GB 記憶體的 cache.m3.medium 節點。

如果備份太大，產生的叢集狀態會是 `restore-failed`。如果發生此情況，您必須刪除叢集並重新開始。



如需節點類型和規格的完整清單，請參閱 [Redis OSS節點類型特定參數](#) 和 [Amazon ElastiCache 產品功能和詳細資訊](#)。

- 您只能使用 Amazon S3 伺服器端加密 OSS ( SSE-S3 ) 來加密 Valkey 或 Redis .rdb 檔案。如需詳細資訊，請參閱 [使用伺服器端加密保護資料](#)。

接下來，您可以找到主題，引導您將叢集從外部遷移OSS至 ElastiCache Valkey 或 Redis ElastiCache ( Redis OSS )。

遷移至 ElastiCache ( Redis OSS )

- [步驟 1：建立 Valkey 或 Redis OSS 備份](#)
- [步驟 2：建立 Amazon S3 儲存貯體和資料夾](#)
- [步驟 3：將備份上傳到 Amazon S3](#)
- [步驟 4：授予 .rdb 檔案的 ElastiCache 讀取存取權](#)

從外部服務遷移至 ElastiCache ( Redis OSS )。

- [步驟 1：建立 Valkey 或 Redis OSS 備份](#)
- [步驟 2：建立 Amazon S3 儲存貯體和資料夾](#)
- [步驟 3：將備份上傳到 Amazon S3](#)
- [步驟 4：授予 .rdb 檔案的 ElastiCache 讀取存取權](#)

## 步驟 1：建立 Valkey 或 Redis OSS 備份

建立 Valkey 或 Redis OSS 備份以植入您的 ElastiCache ( RedisOSS ) 執行個體

1. 連線到現有的 Valkey 或 Redis OSS執行個體。
2. 執行 BGSAVE或 SAVE操作以建立備份。記下您的 .rdb 檔案位置。

BGSAVE 是非同步的，不會封鎖其他用戶端的處理。如需詳細資訊，請參閱 Valkey [BGSAVE](#) 網站上的。

SAVE 是同步的，並會封鎖其他處理序直到完成為止。如需詳細資訊，請參閱 Valkey [SAVE](#) 網站上的。

如需建立備份的詳細資訊，請參閱 Valkey 網站上的[持久性](#)。

## 步驟 2：建立 Amazon S3 儲存貯體和資料夾

建立備份檔案之後，您需要將它上傳到 Amazon S3 儲存貯體中的資料夾。若要執行此操作，您必須先擁有 Amazon S3 儲存貯體，且該儲存貯體中有資料夾。如果您已有具備適當許可的 Amazon S3 儲存貯體和資料夾，您可以跳到「[步驟 3：將備份上傳到 Amazon S3](#)」。

### 建立 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。
2. 依照 Amazon Simple Storage Service 使用者指南中的[建立儲存貯體](#)提供的指引操作，建立 Amazon S3 儲存貯體。

Amazon S3 儲存貯體的名稱必須符合 DNS。否則，ElastiCache 無法存取您的備份檔案。DNS 合規規則包括：

- 名稱長度須為 3 到 63 個字元。
- 名稱必須是一連串一或多個標籤，並以句號 (.) 分隔，其中每個標籤：
  - 以小寫字母或數字開頭。
  - 以小寫字母或數字結尾。
  - 僅包含小寫字母、數字和破折號。
- 不得使用 IP 地址格式 (例如 192.0.2.0)。

您必須在與新 ElastiCache ( RedisOSS ) 叢集相同的 AWS 區域中建立 Amazon S3 儲存貯體。此方法可確保從 Amazon S3 ElastiCache 讀取 .rdb 檔案時的最高資料傳輸速度。

#### Note

為了讓您的資料盡可能保持安全，請盡可能限制您 Amazon S3 儲存貯體的許可。同時，許可仍然需要允許儲存貯體及其內容用於植入新的 Valkey 或 Redis OSS 叢集。

### 在 Amazon S3 儲存貯體中新增資料夾

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。

2. 選擇您要上傳 .rdb 檔案的目的地儲存貯體名稱。
3. 選擇 Create folder (建立資料夾)。
4. 輸入您的新資料夾名稱。
5. 選擇 Save (儲存)。

記下儲存貯體名稱和資料夾名稱。

### 步驟 3：將備份上傳到 Amazon S3

現在，上傳您在[步驟 1：建立 Valkey 或 Redis OSS 備份](#)中建立的 .rdb 檔案上傳到[在 步驟 2：建立 Amazon S3 儲存貯體和資料夾](#)中建立的 Amazon S3 儲存貯體和資料夾。如需此任務的詳細資訊，請參閱[將物件新增至儲存貯體](#)。在步驟 2 到 3 之間，選擇您已建立的資料夾名稱。

將 .rdb 檔案上傳到 Amazon S3 資料夾

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。
2. 選擇您在步驟 2 中建立的 Amazon S3 儲存貯體名稱。
3. 選擇您在步驟 2 中建立的資料夾名稱。
4. 選擇上傳。
5. 選擇 Add files (新增檔案)。
6. 瀏覽至您要上傳的一或多個檔案，然後選擇一或多個檔案。若要選擇多個檔案，請按住 Ctrl 鍵並選擇每個檔案名稱。
7. 選擇 Open (開啟)。
8. 確認 Upload (上傳) 對話方塊中所列的一或多個檔案正確，然後選擇 Upload (上傳)。

記下 .rdb 檔案的路徑。例如，如果您的儲存貯體名稱為 myBucket 且路徑為 myFolder/redis.rdb，請輸入 myBucket/myFolder/redis.rdb。您需要此路徑，才能將此備份資料植入新叢集。

如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的[儲存貯體限制與局限](#)。

### 步驟 4：授予 .rdb 檔案的 ElastiCache 讀取存取權

現在，授予 .rdb 備份檔案的 ElastiCache 讀取存取權。視儲存貯體位於預設 AWS 區域或選擇加入 AWS 區域而定，您會以不同的方式授予備份檔案的 ElastiCache 存取權。

AWS 2019 年 3 月 20 日之前引入的區域預設為啟用。您可以立即開始在這些 AWS 區域中工作。2019 年 3 月 20 日之後推出的區域 (例如亞太區域 (香港) 和 中東 (巴林)) 預設為停用。您必須先啟用或選擇加入這些區域才能使用它們，如 AWS 一般參考 中的[管理 AWS 區域](#)所說明。

根據您的 AWS 區域選擇您的方法：

- 對於預設區域，請使用[授予預設區域中 .rdb 檔案的 ElastiCache 讀取存取權](#)中的程序。
- 對於選擇加入區域，請使用[授予選擇加入區域中 .rdb 檔案的 ElastiCache 讀取存取權](#)中的程序。

### 授予預設區域中 .rdb 檔案的 ElastiCache 讀取存取權

AWS 2019 年 3 月 20 日之前引入的區域預設為啟用。您可以立即開始在這些 AWS 區域中工作。2019 年 3 月 20 日之後推出的區域 (例如亞太區域 (香港) 和 中東 (巴林)) 預設為停用。您必須先啟用或選擇加入這些區域才能使用它們，如 AWS 一般參考 中的[管理 AWS 區域](#)所說明。

在預設啟用的 AWS 區域中授予備份檔案的 ElastiCache 讀取存取權

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。
2. 選擇包含您 .rdb 檔案的 S3 儲存貯體名稱。
3. 選擇包含您 .rdb 檔案的資料夾名稱。
4. 選擇您的 .rdb 備份檔案名稱。所選檔案的名稱將會顯示在頁面頂端的標籤上方。
5. 選擇許可。
6. 如果IDs下列清單中的 aws-scs-s3 個唯讀或其中一個標準未列為使用者，請執行下列動作：
  - a. 在存取其他 AWS 帳戶 下，選擇新增承授者。
  - b. 在方塊中，新增 AWS 區域的正式 ID，如下所示：
    - AWS GovCloud (美國西部) 區域：

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

#### Important

備份必須位於 中的 S3 儲存貯體中，AWS GovCloud (US) 才能下載到 中的 Valkey 或 Redis OSS叢集 AWS GovCloud (US)。

- AWS 預設啟用的區域：

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. 針對下列項目選擇 Yes (是) 以設定儲存貯體的許可：
    - 列出/寫入物件
    - 讀取/寫入物件ACL許可
  - d. 選擇 Save (儲存)。
7. 選擇 Overview (概觀)，然後選擇 Download (下載)。

授予選擇加入區域中 .rdb 檔案的 ElastiCache 讀取存取權

AWS 2019 年 3 月 20 日之前引入的區域預設為啟用。您可以立即開始在這些 AWS 區域中工作。2019 年 3 月 20 日之後推出的區域 (例如亞太區域 (香港) 和 中東 (巴林)) 預設為停用。您必須先啟用或選擇加入這些區域才能使用它們，如 AWS 一般參考 中的[管理 AWS 區域](#)所說明。

現在，授予 .rdb 備份檔案的 ElastiCache 讀取存取權。

授予備份檔案的 ElastiCache 讀取存取權

1. 登入 AWS Management Console 並在 開啟 Amazon S3 主控台<https://console.aws.amazon.com/s3/>。
2. 選擇包含您 .rdb 檔案的 S3 儲存貯體名稱。
3. 選擇包含您 .rdb 檔案的資料夾名稱。
4. 選擇您的 .rdb 備份檔案名稱。所選檔案的名稱將會顯示在頁面頂端的標籤上方。
5. 選擇許可索引標籤標籤。
6. 在 Permissions (許可) 中，選擇 Bucket policy (儲存貯體政策)，然後選擇 Edit (編輯)。
7. 更新政策以授予執行操作 ElastiCache 所需的許可：
  - 將 [ "Service" : "*region-full-name*.elasticache-snapshot.amazonaws.com" ] 新增至 Principal。
  - 新增下列將快照匯出至 Amazon S3 儲存貯體所需的許可：
    - "s3:GetObject"
    - "s3:ListBucket"
    - "s3:GetBucketAcl"

以下是已更新政策可能有的外觀範例。

```
{
 "Version": "2012-10-17",
 "Id": "Policy15397346",
 "Statement": [
 {
 "Sid": "Stmt15399483",
 "Effect": "Allow",
 "Principal": {
 "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
 },
 "Action": [
 "s3:GetObject",
 "s3:ListBucket",
 "s3:GetBucketAcl"
],
 "Resource": [
 "arn:aws:s3:::example-bucket",
 "arn:aws:s3:::example-bucket/backup1.rdb",
 "arn:aws:s3:::example-bucket/backup2.rdb"
]
 }
]
}
```

## 8. 選擇 Save changes (儲存變更)。

### 使用 .rdb 檔案資料種子 ElastiCache 叢集

現在，您已準備好建立 ElastiCache 叢集，並使用 .rdb 檔案的資料將其植入。若要建立叢集，請依照為 [Valkey 或 Redis 建立叢集 OSS](#) 或 [從頭開始建立 Valkey 或 Redis OSS 複寫群組](#) 中的指示進行。請務必選擇 Valkey 或 Redis OSS 作為叢集引擎。

您用來判斷上傳到 Amazon S3 的備份 ElastiCache 位置的方法，取決於您用來建立叢集的方法：

### 使用 .rdb 檔案資料種子 ElastiCache (Redis OSS) 叢集或複寫群組

- 使用 ElastiCache 主控台

選取 Cluster settings (叢集設定) 時，請選擇 Restore from backups (從備份還原) 做為叢集建立方法，然後在 Backup source (備份來源) 區段中，選擇 Other backups (其他備份) 做為您的 Source (來源)。在種子RDB檔案 S3 位置方塊中，輸入檔案的 Amazon S3 路徑 ( Amazon S3 )。如果您有多個 .rdb 檔案，請以逗號分隔清單輸入每個檔案的路徑。Amazon S3 路徑看起來像 *myBucket/myFolder/myBackupFilename.rdb*。

- 使用 AWS CLI

如果您使用 `create-cache-cluster` 或 `create-replication-group` 操作，請使用 參數 `--snapshot-arns` 來指定 ARN 每個 .rdb 檔案的完整資格。例如：`arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`。必須 ARN 解析為您儲存在 Amazon S3 中的備份檔案。

- 使用 ElastiCache API

如果您使用 `CreateCacheCluster` 或 `CreateReplicationGroup` ElastiCache API 操作，請使用 參數 `SnapshotArns` 來指定 ARN 每個 .rdb 檔案的完整資格。例如：`arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`。必須 ARN 解析為您儲存在 Amazon S3 中的備份檔案。

### Important

植入 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集時，您必須在新的叢集或複寫群組中設定每個節點群組 ( 碎片 )。使用 參數 `--node-group-configuration` ( API : `NodeGroupConfiguration` ) 執行此操作。如需詳細資訊，請參閱下列內容：

- CLI AWS CLI 參考 [create-replication-group](#) 中的：
- API 參考 [CreateReplicationGroup](#) 中的 ElastiCache API：

在建立叢集的過程中，Valkey 或 Redis OSS 備份中的資料會寫入叢集。您可以檢視 ElastiCache 事件訊息來監控進度。若要執行此操作，請參閱 ElastiCache 主控台，然後選擇快取事件。您也可以使用 AWS ElastiCache 命令列介面 或 ElastiCache API 來取得事件訊息。如需詳細資訊，請參閱 [檢視 ElastiCache 事件](#)。



## 在 中的引擎版本和升級 ElastiCache

本節涵蓋支援的 Valkey、Redis OSS和 Memcached 引擎以及如何升級。請注意，Redis OSS 7.2 提供的所有功能預設可在 Valkey 7.2 及更高版本中使用。您也可以從一些 ElastiCache 現有的 Redis OSS引擎升級至 Valkey 引擎。

### 主題

- [的版本管理 ElastiCache](#)
- [如何升級引擎版本](#)
- [如何觸發從 Redis OSS 到 Valkey 的跨引擎升級](#)
- [支援的引擎和版本](#)
- [與 Valkey 的主要版本行為和相容性差異](#)
- [與 Redis 的主要版本行為和相容性差異 OSS](#)
- [解決封鎖的 Valkey 或 Redis OSS引擎升級](#)

## 的版本管理 ElastiCache

管理您要如何更新為 Valkey、Redis 和 Memcached 引擎更新的 ElastiCache 快取OSS和自行設計的叢集。

### ElastiCache Serverless Cache 的版本管理

管理無 ElastiCache 伺服器快取升級和升級的時間，並根據自己的條款與條件執行版本升級。

ElastiCache Serverless 會自動將最新的 MINOR和 PATCH 軟體版本套用至快取，而不會對您的應用程式造成任何影響或停機。您不需要進行任何動作。

當新MAJOR版本可用時，ElastiCache Serverless 會在主控台中傳送通知，並在 中傳送事件給您 EventBridge。您可以選擇使用主控台、或 修改快取API，然後選擇最新的引擎版本CLI，以將快取升級至最新的主要版本。

### 自我設計 ElastiCache 叢集的版本管理

使用自行設計的 ElastiCache 叢集時，您可以控制為快取叢集供電的軟體何時升級至 支援的新版本 ElastiCache 。您可以控制何時將快取升級至最新的可用 MINOR、 MAJOR和 PATCH版本。您可以修改引擎版本並指定新的引擎版本，以初始化叢集或複寫群組的引擎版本升級。



您可以控制為快取叢集供電的通訊協定相容軟體是否和何時升級至支援的新版本 ElastiCache。這一層控制可讓您維持特定版本的相容性、在部署至生產環境前先利用您的應用程式測試新版本，並根據自己的期限和時間表執行版本升級。

由於版本升級可能涉及一定的相容性風險，因此這些升級不會自行啟動，必須由您啟動。

## Valkey 和 Redis OSS叢集

### Note

- 如果 Valkey 或 Redis OSS叢集在一或多個區域中複寫，則會針對次要區域升級引擎版本，然後針對主要區域升級引擎版本。
- ElastiCache (Redis OSS) 版本會以語意版本識別，該版本包含 MAJOR和 MINOR元件。例如，在 Redis OSS 6.2 中，主要版本為 6，次要版本為 2。操作自行設計的叢集時，ElastiCache (Redis OSS) 也會公開PATCH元件，例如 Redis OSS 6.2.1，且修補程式版本為 1。

MAJOR 版本適用於API不相容的變更，而MINOR版本適用於以向下相容方式新增的新功能。PATCH 版本適用於向下相容的錯誤修正和非功能變更。

使用 Valkey 和 Redis OSS，您可以透過修改並指定新的引擎版本，啟動叢集或複寫群組的引擎版本升級。如需詳細資訊，請參閱[修改複寫群組](#)。

## Memcached

使用 Memcached，若要升級至較新的版本，您必須修改快取叢集，並指定要使用的新引擎版本。升級至更新版本的 Memcached 是一種破壞性程序，您會失去所有資料，並從空白快取開始。如需詳細資訊，請參閱[修改 ElastiCache 叢集](#)。

從舊版 Memcached 升級到 Memcached 1.4.33 版或更新版本時，請注意下列要求。CreateCacheCluster 和 ModifyCacheCluster 必須滿足下列條件：

- 如果  $\text{slab\_chunk\_max} > \text{max\_item\_size}$ 。
- 如果  $\text{max\_item\_size} \bmod \text{slab\_chunk\_max} \neq 0$ 。
- 如果  $\text{max\_item\_size} > ((\text{max\_cache\_memory} - \text{memcached\_connections\_overhead}) / 4)$ 。

$(\text{max\_cache\_memory} - \text{memcached\_connections\_overhead})$  值是節點可供資料使用的記憶體。如需詳細資訊，請參閱[Memcached 連線額外負荷](#)。

## 使用自行設計的叢集時的升級考量

### Note

下列考量僅適用於升級自行設計之叢集的情況。它們不適用於 ElastiCache Serverless。

### Valkey 和 Redis OSS 考量事項

升級自我設計的 Valkey 或 Redis OSS 叢集時，請考慮下列事項。

- 引擎版本管理功能是為了讓您能夠盡可能控制執行修補的方式，不過，ElastiCache 保留在系統或快取軟體中不太可能發生重大安全漏洞時，代表您修補叢集的權利。
- 從 Valkey 7.2 和 Redis OSS 6.0 開始，ElastiCache 將為每個次要版本提供單一版本，而不是提供多個修補程式版本。
- 從 Redis OSS 引擎 5.0.6 版開始，您可以在最短的停機時間下升級叢集版本。叢集在整個升級過程中都可供讀取，而在過程的多數時間也可供寫入，除了在容錯移轉操作中會有幾秒可能無法寫入。
- 您也可以使用 5.0.6 之前的版本升級 ElastiCache 叢集。涉及的程序相同，但可能會在傳播期間產生較長的容錯移轉時間 DNS (30s-1m)。
- 從 Redis OSS 7 開始，ElastiCache 支援在 Valkey 或 Redis OSS (停用叢集模式) 與 Valkey 或 Redis OSS (啟用叢集模式) 之間切換。
- Amazon ElastiCache (Redis OSS) 引擎升級程序旨在盡最大努力保留現有資料，並需要成功的 Redis OSS 複寫。
- 升級引擎時，ElastiCache 會終止現有的用戶端連線。為了在引擎升級期間將停機時間降至最低，我們建議您為 [Redis OSS 用戶端實作最佳實務](#)，並執行錯誤重試和指數退避，以及將 [維護期間的停機時間降至最低](#) 的最佳實務。
- 升級引擎時，您無法直接從 Valkey 或 Redis OSS (停用叢集模式) 升級到 Valkey 或 Redis OSS (啟用叢集模式)。下列程序說明如何從 Valkey 或 Redis OSS (停用叢集模式) 升級到 Valkey 或 Redis OSS (啟用叢集模式)。

從 Valkey 或 Redis OSS ( 停用叢集模式 ) 升級到 Valkey 或 Redis OSS ( 啟用叢集模式 ) 引擎版本

1. 備份您的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集或複寫群組。如需詳細資訊，請參閱[取得手動備份](#)。
  2. 使用備份建立並植入具有一個碎片 OSS ( 節點群組 ) 的 Valkey 或 Redis ( 啟用叢集模式 ) 叢集。指定新的引擎版本，並在建立叢集或複寫群組時啟用叢集模式。如需詳細資訊，請參閱[教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。
  3. 刪除舊的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集或複寫群組。如需詳細資訊，請參閱 [在中刪除叢集 ElastiCache](#) 或 [刪除複寫群組](#)。
  4. 將新的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集或複寫群組擴展到您需要的碎片數量 ( 節點群組 )。如需詳細資訊，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)
- 升級主要引擎版本 (例如從 5.0.6 升級到 6.0) 時，還需要選擇一個與新引擎版本相容的新參數群組。
  - 對於停用多可用區域之單一 Redis OSS叢集和叢集，建議OSS如 所述為 Redis 提供足夠的記憶體[確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)。在這類案例中，主要叢集將無法在升級期間處理服務的請求。
  - 對於啟用多可用區域功能的 Redis OSS叢集，我們也建議您在低傳入寫入流量期間排程引擎升級。升級至 Redis OSS 5.0.6 或更新版本時，主要叢集在升級過程中仍可繼續用於服務請求。

系統會依下列方式處理及修補含多個碎片的叢集和複寫群組：

- 所有碎片都會平行處理。不論任何時間，一個碎片上都只能執行一項升級操作。
- 在每個碎片中，都會先處理所有複本，再處理主要複本。如果某個碎片中的複本較少，則該碎片中的主要複本可能會在其他碎片的複本處理完成前就已處理。
- 跨所有碎片時，則會循序處理主要節點。一次只會升級一個主要節點。
- 如果您目前的叢集和複寫群組已啟用加密功能，就無法升級到不支援加密的引擎版本，例如從 3.2.6 升級到 3.2.10。

## Memcached 考量事項

升級自我設計的 Memcached 叢集時，請考慮下列事項。

- 引擎版本管理功能是為了讓您能夠盡可能控制執行修補的方式，不過，ElastiCache 保留在系統或快取軟體中不太可能發生重大安全漏洞時，代表您修補叢集的權利。
- 由於 Memcached 引擎不支援永久儲存，因此 Memcached 引擎版本升級一定是破壞性程序，並會清除叢集中的所有快取資料。

## 如何升級引擎版本

### Valkey 和 Redis OSS

透過 Valkey 和 Redis OSS，您可以使用 ElastiCache 主控台、或 ElastiCache API 修改叢集或複寫群組的版本升級 AWS CLI，並指定較新的引擎版本。如需詳細資訊，請參閱下列主題。

| 如何修改叢集和複寫群組                                           |                                           |
|-------------------------------------------------------|-------------------------------------------|
| 叢集                                                    | 複寫群組                                      |
| <a href="#">使用 ElastiCache AWS Management Console</a> | <a href="#">使用 AWS Management Console</a> |
| <a href="#">AWS CLI 搭配 使用 ElastiCache</a>             | <a href="#">使用 AWS CLI</a>                |
| <a href="#">使用 ElastiCache API</a>                    | <a href="#">使用 ElastiCache API</a>        |

### Memcached

使用 Memcached，若要開始叢集的版本升級，請修改它並指定較新的引擎版本。您可以使用 ElastiCache 主控台、AWS CLI 或來執行此操作 ElastiCache API：

- 若要使用 AWS Management Console，請參閱 – [使用 ElastiCache AWS Management Console](#)。
- 若要使用 AWS CLI，請參閱 [AWS CLI 搭配 使用 ElastiCache](#)。
- 若要使用 ElastiCache API，請參閱 [使用 ElastiCache API](#)。

## 如何觸發從 Redis OSS 到 Valkey 的跨引擎升級

您可以使用主控台或將現有的 Redis OSS 複寫群組（v4 或之後）升級至 Valkey API 引擎 CLI。

### Note

如果您想要將現有的 Redis OSS（停用叢集模式）單節點叢集升級至 Valkey 引擎，您必須先遵循下列先決條件步驟，將其新增至複寫群組：[使用現有叢集建立複寫群組](#)。

如果您有使用預設快取參數群組的現有 Redis OSS 複寫群組，可以透過使用 modify-replication-group 指定新的引擎和引擎版本來升級至 Valkey API。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2
```

如果您將自訂快取參數群組套用至要升級的現有 redis 複寫群組，您也需要在請求中傳遞自訂 Valkey 快取參數群組。輸入 Valkey 自訂參數群組必須與現有的 Redis 自訂參數群組具有相同的 Redis 靜態參數值。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2 \
 --cache-parameter-group-name myParamGroup
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2 ^
 --cache-parameter-group-name myParamGroup
```

## Redis OSS至 Valkey 無 ElastiCache 伺服器跨引擎升級

您可以使用主控台，API或使用 指定新的引擎和主要引擎版本CLI，將 modify-serverless-cache現有的 Redis 無OSS伺服器快取升級至 Valkey 引擎API。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-serverless-cache \
 --serverless-cache-name myCluster \
 --engine valkey \
 --major-engine-version 7
```

針對 Windows :

```
aws elasticache modify-serverless-cache ^
 --serverless-cache-name myCluster ^
 --engine valkey ^
 --major-engine-version 7
```

## 支援的引擎和版本

ElastiCache 無伺服器快取支援 Valkey 7.2 及更高版本、Redis 7.0 OSS版，以及 Memcached 1.6 及更高版本。

ElastiCache 自我設計的快取支援 Valkey 7.2 及更高版本、所有 Redis 4OSS.0.10 及更高版本，以及 Memcached 1.4.5 及更高版本。

自行設計的 ElastiCache 叢集支援下列 Valkey 版本：

- [支援的 Valkey 版本](#)
- [支援的 Redis OSS版本](#)
- [Redis OSS版本生命週期結束排程](#)
- [支援的 ElastiCache \( Memcached \) 版本](#)

## 支援的 Valkey 版本

支援的 Valkey 版本如下。請注意，根據預設，Valkey 支援 Redis 7.2 OSS 中可用的大多數功能。

### ElastiCache ( Valkey ) 7.2.6 版

2024 年 10 月 10 日，ElastiCache Valkey 7.2.6 發行。以下是 Valkey 7.2 中引入的一些新功能（相較於 Redis OSS 7.1）：

- ZRANK 和 ZREVRANK 命令的新 WITHSCORE 選項
- CLIENT NO-TOUCH 可讓用戶端在不影響 LRU/LFU 金鑰的情況下執行命令。
- 根據複寫，CLUSTERMYSHARDID 將節點的碎片 ID 傳回至叢集模式中邏輯分組節點的新命令。

- 各種資料類型的效能和記憶體最佳化。

以下是 Valkey 7.2 和 Redis OSS 7.1 ( 或 7.0 ) 之間可能中斷的行為變更：

- PUBLISH 使用同時訂閱相同頻道的RESP3用戶端呼叫時，順序會變更，並在發佈的訊息之前傳送回覆。
- 指令碼的用戶端追蹤現在會追蹤指令碼讀取的金鑰，而不是 / EVAL 的呼叫者宣告的金鑰FCALL。
- 凍結時間取樣會在命令執行期間和指令碼中發生。
- 解除封鎖命令時，會重新評估 ACL、OOM等檢查和其他檢查。
- ACL 失敗錯誤訊息文字和錯誤代碼統一。
- 當金鑰不再存在時所釋出的封鎖串流命令，會攜帶不同的錯誤碼 ( -NOGROUP 或 -WRONGTYPE 而非 -UNBLOCKED )。
- 只有在命令實際執行時，才會針對封鎖的命令更新命令統計資料。
- ACL 使用者的內部儲存不再移除備援命令和類別規則。這可能會改變這些規則在 SAVE、ACL ACLGETUSER和 ACL 中顯示的方式LIST。
- 如果可能SNI，為 TLS型複寫使用而建立的任何用戶端連線。
- XINFO STREAM：看到的時間回應欄位現在表示上次嘗試的互動，而不是上次成功的互動。新的作用中時間回應欄位現在表示上次成功的互動。
- XREADGROUP 和 X【AUTO】CLAIM 會建立取用者，無論其是否能夠執行一些讀取/宣告。【TBD - 這裡的「它」是什麼？】
- ACL 預設 LIST/ 中新建立的使用者集 sanitize-payload ACL 旗標GETUSER。
- 除非成功，否則HELLO命令不會影響用戶端狀態。
- NAN 回應會標準化為單一 nan 類型，類似於 inf 的目前行為。

如需 Valkey 的詳細資訊，請參閱 [Valkey](#)

如需 Valkey 7.2 版本的詳細資訊，請參閱上的 [Redis OSS 7.2.4 版本備註](#) ( Valkey 7.2 包含從 Redis OSS到 7.2.4 版的所有變更 ) 和 [Valkey 7.2 版本備註](#) GitHub。

## 支援的 Redis OSS版本

ElastiCache 無伺服器快取和自行設計的快取支援所有 Redis 7.1 版和之前OSS版本。

- [ElastiCache \( Valkey \) 7.2.6 版](#)
- [ElastiCache \( Redis OSS \) 7.1 版 \( 增強版 \)](#)



自行設計的 ElastiCache 叢集支援下列 Valkey 和 Redis OSS 版本：

- [ElastiCache \(Redis OSS\) 7.1 版 \(增強版\)](#)
- [ElastiCache \(Redis OSS\) 7.0 版 \(增強版\)](#)
- [ElastiCache \(Redis OSS\) 6.2 版 \(增強版\)](#)
- [ElastiCache \(Redis OSS\) 6.0 版 \(增強版\)](#)
- [ElastiCache \(Redis OSS\) 5.0.6 版 \(增強版\)](#)
- [ElastiCache \(Redis OSS\) 5.0.5 版 \(已棄用，請使用 5.0.6 版\)](#)
- [ElastiCache \(Redis OSS\) 5.0.4 版 \(已棄用，請使用 5.0.6 版\)](#)
- [ElastiCache \(Redis OSS\) 5.0.3 版 \(已棄用，請使用 5.0.6 版\)](#)
- [ElastiCache \(Redis OSS\) 5.0.0 版 \(已棄用，請使用 5.0.6 版\)](#)
- [ElastiCache \(Redis OSS\) 4.0.10 版 \(增強版\)](#)
- [過去生命週期結束 \(EOL\) 版本 \(3.x\)](#)
- [過去生命週期結束 \(EOL\) 版本 \(2.x\)](#)

### ElastiCache (Redis OSS) 7.1 版 (增強版)

此版本包含效能改善，可讓工作負載提高輸送量並降低操作延遲。ElastiCache 7.1 推出[兩個主要增強功能](#)：

我們擴展了增強型 I/O 執行緒功能，現在也能夠處理呈現層邏輯。依呈現層，我們是指增強型 I/O 執行緒，現在不僅讀取用戶端輸入，也剖析輸入為 Redis OSS 二進位命令格式。然後再將其轉發到主執行緒來執行，藉此實現效能增益。改進了 Redis OSS 記憶體存取模式。來自許多資料結構操作的執行步驟為交錯式，以確保平行記憶體存取並降低記憶體存取延遲。在 Graviton3-based R7g.4xlarge 或更大 ElastiCache 版本上執行時，客戶可以實現每個節點每秒超過 100 萬個請求。與 ElastiCache (Redis) v7.0 相比，透過 (Redis OSS) v7OSS.1 的效能改進，客戶可以實現多 100% 的輸送量和低 50% ElastiCache 的 P99 延遲。這些增強功能會在節點大小上啟用，其中至少包含 8 個實體核心 (2xlarge 在 Graviton 上，在 x86 4xlarge 上)，無論 CPU 類型為何，而且不需要變用戶端。

#### Note

ElastiCache v7.1 與 Redis OSS v7.0 相容。

### ElastiCache (Redis OSS) 7.0 版 (增強版)

ElastiCache (Redis OSS) 7.0 新增了許多改進和對新功能的支援：



- [Functions](#) : ElastiCache ( Redis OSS ) 7 新增對 Redis OSS Functions 的支援，並提供受管體驗，讓開發人員可以使用儲存在叢集上的 ElastiCache 應用程式邏輯執行 [LUA 指令碼](#)，而無需用戶端透過每個連線重新將指令碼傳送至伺服器。
- [ACL 改進](#) : Valkey OSS 和 Redis 7 新增對下一版本存取控制清單 ( ) 的支援 ACLs。用戶端現在可以在 Valkey 和 Redis 中指定特定金鑰或金鑰空間的多組許可 OSS。
- [Sharded Pub/Sub](#) : ElastiCache 搭配 Valkey OSS 和 Redis 7 新增了支援，以在啟用 ElastiCache 叢集模式 ( ) 中執行時以碎片方式執行 Pub/Sub 功能 CME。Pub/Sub 功能可讓發佈者向頻道上任何數量的訂閱者發出訊息。通道與 ElastiCache 叢集中的碎片繫結，無需在碎片之間傳播通道資訊，從而提高可擴展性。
- 增強型 I/O 多工處理 : ElastiCache 搭配 Valkey 和 Redis OSS 7 推出增強型 I/O 多工處理，可為與 ElastiCache 叢集具有許多並行用戶端連線的高輸送量工作負載提供更高的輸送量和更低的延遲。例如，使用 r6g.xlarge 節點叢集並執行 5200 個並行用戶端時，相較於 ElastiCache Redis 第 6 OSS 版，您最多可以提高 72% 的輸送量 ( 每秒的讀取和寫入操作 )，並減少 71% 的 P99 延遲。

如需 Valkey 的詳細資訊，請參閱 [Valkey](#)。如需 Redis OSS 7.0 版本的詳細資訊，請參閱 OSS 上的 [Redis OSS 7.0 版本備註](#) GitHub。

## ElastiCache ( Redis OSS ) 6.2 版 ( 增強版 )

ElastiCache ( Redis OSS ) 6.2 包括使用 x86 節點類型搭配 8 個 vCPUs 以上節點或 Graviton2 節點類型搭配 4 個 vCPUs 以上啟用 TLS 的叢集的效能改善。這些增強功能透過將加密卸載至其他來改善輸送量並減少用戶端連線建立時間 vCPUs。使用 Redis OSS 6.2，您也可以使用存取控制清單 ( ACL ) 規則來管理對 Pub/Sub 頻道的存取。

在此版本中，我們也推出對包含本機連接 NVMe 之叢集節點資料分層的支援 SSD。如需詳細資訊，請參閱 [中的資料分層 ElastiCache](#)。

Redis OSS 引擎 6.2.6 版也推出對原生 JavaScript Object Notation ( JSON ) 格式的支援，這是一種簡單、無結構描述的方式，可編碼 Redis OSS 叢集內的複雜資料集。透過 JSON 支援，您可以 OSS APIs 為透過操作的應用程式利用和 Redis 的效能 JSON。如需詳細資訊，請參閱 [入門 JSON](#)。也包含 JSON 相關指標，JsonBasedCmds 以及 JsonBasedCmdsLatency，這些指標會納入 CloudWatch 以監控此資料類型的使用。如需詳細資訊，請參閱 [Valkey 和 Redis 的指標 OSS](#)。

您可以使用 6.2. ElastiCache ( Redis OSS ) 會自動叫用可用的 Redis OSS 6.2 的偏好修補程式版本。例如，建立/修改快取叢集時，您可以將 `--engine-version` 參數設為 6.2。叢集將在建立/修改時，以目前可用的 Redis OSS 6.2 偏好修補程式版本啟動。在 中指定引擎版本 6.x API 會導致 Redis 6 OSS 的最新次要版本。

對於現有的 6.0 叢集，您可以透過在 `CreateCacheCluster`、`CreateReplicationGroup` 或 `ModifyReplicationGroup` 中將 `AutoMinorVersionUpgrade` 參數設定為 `yes`，選擇加入下一個自動次要版本升級 APIs。ElastiCache ( Redis OSS ) 將使用自助更新，將現有 6.0 叢集的次要版本升級至 6.2。如需詳細資訊，請參閱 [Amazon 中的自助式更新 ElastiCache](#)。

呼叫時 `DescribeCacheEngineVersions` API，`EngineVersion` 參數值會設定為 6.2，而具有修補程式版本的實際引擎版本會在 `CacheEngineVersionDescription` 欄位中傳回。

如需 Redis OSS 6.2 版本的詳細資訊，請參閱 [上的 Redis OSS 6.2 版本備註](#) OSS GitHub。

### ElastiCache ( Redis OSS ) 6.0 版 ( 增強版 )

Amazon ElastiCache ( Redis OSS ) 推出下一個版本的 Redis OSS 引擎，包括 [使用角色型存取控制驗證使用者](#)、用戶端快取和重大的操作改進。

從 Redis OSS 6.0 開始，ElastiCache ( Redis OSS ) 將為每個 Redis OSS 次要版本提供單一版本，而不是提供多個修補程式版本。ElastiCache ( Redis OSS ) 會自動管理執行中快取叢集的修補程式版本，確保效能改善且安全性增強。

您也可以將 `AutoMinorVersionUpgrade` 參數設定為 `yes` 和 ElastiCache ( Redis OSS )，以選擇加入下一個自動次要版本升級，並透過自助更新來管理次要版本升級。如需詳細資訊，請參閱 [中的服務更新 ElastiCache](#)。

您可以使用指定引擎版本 6.0。ElastiCache ( Redis OSS ) 會自動叫用可用的 Redis OSS 6.0 的偏好修補程式版本。例如，建立/修改快取叢集時，您可以將 `--engine-version` 參數設為 6.0。叢集將在建立/修改時，以目前可用的 Redis OSS 6.0 偏好修補程式版本啟動。任何具有特定修補程式版本值的請求都將遭到拒絕、擲回例外狀況，且處理會失敗。

呼叫時 `DescribeCacheEngineVersions` API，`EngineVersion` 參數值會設定為 6.0，而具有修補程式版本的實際引擎版本會在 `CacheEngineVersionDescription` 欄位中傳回。

如需 Redis OSS 6.0 版本的詳細資訊，請參閱 OSS 上的 [Redis OSS 6.0 版本備註](#) GitHub。

### ElastiCache ( Redis OSS ) 5.0.6 版 ( 增強版 )

Amazon ElastiCache ( Redis OSS ) 推出下一個版本的 Redis OSS 引擎，其中包括錯誤修正和下列累積更新：

- 特定條件下的引擎穩定性保證。

- 改善的 Hyperloglog 錯誤處理。
- 改進交握命令，進行可靠複寫。
- 以 XCLAIM 命令持續追蹤訊息遞送。
- 改善物件中的 LFU 欄位管理。
- 改進使用 ZPOP 時的交易管理。
- 重新命名命令的能力：稱為 `rename-commands` 的參數，可讓您重新命名可能造成意外資料遺失的潛在危險或昂貴的 Redis OSS 命令，例如 `FLUSHALL` 或 `FLUSHDB`。這類似於開放原始碼 Redis 中的重新命名命令組態 OSS。不過，ElastiCache 透過提供完全受管的工作流程來改善體驗。命令名稱的變更會立即套用，並自動傳播至命令清單所在叢集中的所有節點。使用者無需介入重啟節點及其他任何操作。

以下範例說明如何修改現有參數群組。此群組包含 `rename-commands` 參數，這是由空格分隔的命令清單，可供您重新命名。

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

這個範例使用重新命名 - 命令參數來將 `flushall` 命令重新命名為 `restrictedflushall`。

若要重新命名多個命令，請使用下列參數：

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall flushdb restrictedflushdb'" --region region
```

若要恢復任何變更，請重新執行命令，並從所要保留的 `ParameterValue` 清單中排除重新命名的值，如下所示：

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

在這種情況下，`flushall` 命令將會重新命名為 `restrictedflushall`，任何其他重新命名的命令則恢復為其原始的命令名稱。

**Note**

命令的重新命名功能具有下列限制：

- 所有重新命名的命令應採用英數字元。
- 新的命令名稱最大長度是 20 個英數字元。
- 為命令重新命名時，請確實更新叢集所關聯的參數群組。
- 若要完全防止使用特定命令，請使用關鍵字 `blocked`，如下所示：

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall blocked'" --region region
```

進一步了解參數變更和可以重新命名的命令清單，請參閱 [Redis OSS 5.0.3 參數變更](#)。

- Redis OSS Streams：此模型為日誌資料結構，可讓生產者即時附加新項目。同時也允許消費者以封鎖或非封鎖方式取用訊息。串流也允許消費者群組，其代表以合作方式取用相同訊息串流的一組用戶端，類似 [Apache Kafka](#)。如需詳細資訊，請參閱 [串流](#)。
- 支援一系列的串流命令，例如 XADD、XRANGE 和 XREAD。如需詳細資訊，請參閱 [串流命令](#)。
- 多個新的和重新命名的參數。如需詳細資訊，請參閱 [Redis OSS 5.0.0 參數變更](#)。
- 新的 Redis OSS 指標 `StreamBasedCmds`。
- Redis OSS 節點的快照時間略快。

**Important**

ElastiCache (Redis OSS) 從 [Redis OSS 開放原始碼 5.0.1 版](#) 回溯匯入了兩個關鍵錯誤修正。下列為其說明：

- RESTORE 某些金鑰已過期時，回應不相符。
- XCLAIM 命令可能傳回錯誤的項目，或取消同步通訊協定。

這兩種錯誤修正都包含在 Redis OSS 引擎 5.0.0 版的 ElastiCache (Redis OSS) 支援中，並在未來版本更新中耗用。

如需詳細資訊，請參閱 上的 [Redis OSS 5.0.6 版本備註](#) OSS GitHub。

ElastiCache ( Redis OSS ) 5.0.5 版 ( 已棄用，請使用 5.0.6 版 )

Amazon ElastiCache ( Redis OSS ) 推出下一個版本的 Redis OSS引擎；它包含所有計劃操作期間自動容錯移轉叢集 ElastiCache ( RedisOSS ) 的線上組態變更。您現在可以擴展叢集、升級 Redis OSS引擎版本並套用修補程式和維護更新，同時叢集會保持連線，並繼續提供傳入的請求。它也包括錯誤修正。

如需詳細資訊，請參閱 上的 [Redis OSS 5.0.5 Redis 版本備註](#) GitHub。 OSS

ElastiCache ( Redis OSS ) 5.0.4 版 ( 已棄用，請使用 5.0.6 版 )

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個 Redis OSS引擎版本 ElastiCache。包含下列增強功能：

- 特定條件下的引擎穩定性保證。
- 改善的 Hyperloglog 錯誤處理。
- 改進交握命令，進行可靠複寫。
- 以 XCLAIM 命令持續追蹤訊息遞送。
- 改善物件中的 LFU 欄位管理。
- 改進使用 ZPOP 時的交易管理。

如需詳細資訊，請參閱 上的 [Redis OSS 5.0.4 Redis 版本備註](#) GitHub。 OSS

ElastiCache ( Redis OSS ) 5.0.3 版 ( 已棄用，請使用 5.0.6 版 )

Amazon ElastiCache ( Redis OSS ) 推出 Amazon ElastiCache 支援的下一個 Redis OSS引擎版本，其中包含錯誤修正。

ElastiCache ( Redis OSS ) 5.0.0 版 ( 已棄用，請使用 5.0.6 版 )

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個主要 Redis OSS引擎版本 ElastiCache。ElastiCache ( Redis OSS ) 5.0.0 為下列改進提供支援：

- Redis OSS Streams：此模型為日誌資料結構，可讓生產者即時附加新項目。同時也允許消費者以封鎖或非封鎖方式取用訊息。串流也允許消費者群組，其代表以合作方式取用相同訊息串流的一組用戶端，類似 [Apache Kafka](#)。如需詳細資訊，請參閱[串流](#)。
- 支援一系列的串流命令，例如 XADD、XRANGE 和 XREAD。如需詳細資訊，請參閱[串流命令](#)。
- 多個新的和重新命名的參數。如需詳細資訊，請參閱[Redis OSS 5.0.0 參數變更](#)。

- 新的 Redis OSS 指標 StreamBasedCmds。
- Redis OSS節點的快照時間略快。

## ElastiCache ( Redis OSS ) 4.0.10 版 ( 增強版 )

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個主要 Redis OSS引擎版本 ElastiCache。ElastiCache ( Redis OSS ) 4.0.10 帶來下列改進支援：

- 在單一 ElastiCache ( Redis OSS ) 版本中調整大小和加密的線上叢集。如需詳細資訊，請參閱下列內容：
  - [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)
  - [Valkey 或 Redis 的線上重新共享 OSS \( 啟用叢集模式 \)](#)
  - [Amazon 中的資料安全 ElastiCache](#)
- 多個新的參數。如需詳細資訊，請參閱[Redis OSS 4.0.10 參數變更](#)。
- 支援記憶體的一系列命令，例如 MEMORY。如需詳細資訊，請參閱[命令 \( 搜尋 \)](#) MEMO。
- 支援線上記憶體重組，可保障更有效的記憶體使用率，讓您的資料可使用更多記憶體。
- 支援非同步排清和刪除。ElastiCache ( Redis OSS ) 支援類似 UNLINKFLUSHDB和 的命令FLUSHALL，以在與主執行緒不同的執行緒中執行。這麼做可用非同步方式釋放記憶體，藉以提升應用程式的效能與回應時間。
- 新的 Redis OSS 指標 ActiveDefragHits。如需詳細資訊，請參閱 [Redis 的指標。OSS](#)

執行 Redis 3.2.10 OSS版的 Valkey 或 Redis OSS ( 停用叢集模式 ) 使用者可以使用主控台透過線上升級來升級叢集。

## 比較 ElastiCache ( RedisOSS ) 叢集大小調整和加密支援

| 功能         | 3.2.6 | 3.2.10 | 4.0.10 和更新版本 |
|------------|-------|--------|--------------|
| 線上叢集大小調整 * | 否     | 是      | 是            |
| 傳輸中加密 **   | 是     | 否      | 是            |
| 靜態加密 **    | 是     | 否      | 是            |

\* 新增、移除和重新平衡碎片。



| 功能 | 3.2.6 | 3.2.10 | 4.0.10 和更新版本 |
|----|-------|--------|--------------|
|----|-------|--------|--------------|

\*\* 聯準會 RAMP、HIPAA和 PCI DSS 相容應用程式所需的。如需詳細資訊，請參閱[Amazon 的合規驗證 ElastiCache](#)。

過去生命週期結束 (EOL) 版本 (3.x)

ElastiCache (Redis OSS) 3.2.10 版 (增強版)

Amazon ElastiCache (Redis OSS) 推出 Amazon 支援的下一個主要 Redis OSS引擎版本 ElastiCache。ElastiCache (Redis OSS) 3.2.10 引入線上叢集調整大小，以在叢集繼續提供傳入 I/O 請求時，從叢集新增或移除碎片。ElastiCache (Redis OSS) 3.2.10 使用者除了能夠加密其資料 OSS之外，還擁有舊版 Redis 的所有功能。此能力目前僅於 3.2.6 版提供。

比較 ElastiCache (RedisOSS) 3.2.6 版和 3.2.10

| 功能         | 3.2.6 | 3.2.10 |
|------------|-------|--------|
| 線上叢集大小調整 * | 否     | 是      |
| 傳輸中加密 **   | 是     | 否      |
| 靜態加密 **    | 是     | 否      |

\* 新增、移除和重新平衡碎片。

\*\* 聯準會 RAMP、HIPAA和 PCI DSS 相容應用程式所需的。如需詳細資訊，請參閱[Amazon 的合規驗證 ElastiCache](#)。

如需詳細資訊，請參閱下列內容：

- [Valkey 或 Redis 的線上重新共享 OSS \(啟用叢集模式\)](#)
- [線上叢集大小調整](#)

## ElastiCache ( Redis OSS ) 3.2.6 版 ( 增強版 )

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個主要 Redis OSS引擎版本 ElastiCache。ElastiCache ( Redis OSS ) 3.2.6 使用者擁有舊版 Redis 的所有功能，OSS以及加密其資料的選項。如需詳細資訊，請參閱下列內容：

- [ElastiCache 傳輸中加密 \( TLS \)](#)
- [中的靜態加密 ElastiCache](#)
- [Amazon 的合規驗證 ElastiCache](#)

## ElastiCache ( Redis OSS ) 3.2.4 版 ( 增強版 )

Amazon ElastiCache ( Redis OSS ) 3.2.4 版推出 Amazon 支援的下一個主要 Redis OSS引擎版本 ElastiCache。ElastiCache ( Redis OSS ) 3.2.4 使用者擁有OSS可用的 Redis 舊版的所有功能，以及可在叢集模式或非叢集模式下執行的選項。下表提供摘要。

## 比較 Redis OSS 3.2.4 非叢集模式和叢集模式

| 功能     | 非叢集模式               | 叢集模式                           |
|--------|---------------------|--------------------------------|
| 資料分割   | 否                   | 是                              |
| 地理空間索引 | 是                   | 是                              |
| 變更節點類型 | 是                   | 是 *                            |
| 複本擴展   | 是                   | 是 *                            |
| 擴展     | 否                   | 是 *                            |
| 資料庫支援  | 多個                  | 單一                             |
| 參數群組   | default.redis3.2 ** | default.redis3.2.cluster.on ** |

\* 請參閱「[從備份還原到新的快取](#)」

\*\* 或其衍生內容。



**備註：**

- 磁碟分割 - 將您的資料分割到 2 至 500 個節點群組 (碎片) 中，並對每個節點群組提供複寫支援。
- 地理空間索引 – Redis OSS 3.2.4 透過六個GEO命令引入對地理空間索引的支援。如需詳細資訊，請參閱 Valkey 命令頁面上的 Redis OSS GEO\* 命令文件命令：( 已針對 篩選GEO )。 [GEO](#)

如需 Redis 3 OSS 其他功能的相關資訊，請參閱 [Redis OSS 3.2 版本備註](#)和 [Redis OSS 3.0 版本備註](#)。

目前 ElastiCache 受管的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 不支援下列 Redis OSS 3.2 功能：

- 複本移轉
- 叢集重新平衡
- Lua 除錯器

ElastiCache 停用下列 Redis OSS 3.2 管理命令：

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`
- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumepoch`
- `cluster set-config-epoch`
- `cluster reset`

如需 Redis OSS 3.2.4 參數的相關資訊，請參閱 [Redis OSS 3.2.4 參數變更](#)。

## 過去生命週期結束 ( EOL ) 版本 ( 2.x )

### ElastiCache ( Redis OSS ) 2.8.24 版 ( 增強版 )

自 2.8.23 版以來新增的 Redis OSS 改進包括錯誤修正和不良記憶體存取地址的記錄。如需詳細資訊，請參閱 [Redis OSS 2.8 版本備註](#)。

### ElastiCache ( Redis OSS ) 2.8.23 版 ( 增強版 )

自 2.8.22 版以來新增的 Redis OSS 改進包含錯誤修正。如需詳細資訊，請參閱 [Redis OSS 2.8 版本備註](#)。此版本也包括對新參數 `close-on-slave-write` 的支援，若啟用此參數，則會在用戶端嘗試寫入唯讀複本時中斷其連線。

如需 Redis OSS 2.8.23 參數的詳細資訊，請參閱 ElastiCache 使用者指南 [Redis OSS 2.8.23 \( 增強 \) 新增的參數](#) 中的。

### ElastiCache ( Redis OSS ) 2.8.22 版 ( 增強版 )

自 2.8.21 版以來新增的 Redis OSS 改進包括以下內容：

- 支援無延伸備份和同步，其可讓您為備份額外負荷配置較少的記憶體，而為應用程式配置較多的記憶體。如需詳細資訊，請參閱 [同步與備份的實作方式](#)。此無延伸程序可能會影響延遲和輸送量。具有高寫入傳輸量時，當複本重新同步時，有可能在整段同步期間都無法連線。
- 如果發生容錯移轉，由於複本會盡可能與主要節點執行部分同步，而不需進行完整同步，因此現在可以更快恢復複寫群組。此外，主要節點和複本不再需要於同步期間使用磁碟，因此可提供更快的速度。
- 支援兩個新 CloudWatch 指標。
  - `ReplicationBytes` - 複寫群組的主要叢集傳送給僅供讀取複本的位元組數。
  - `SaveInProgress` - 二進位值，其指出背景儲存程序是否正在執行。

如需詳細資訊，請參閱 [監控與 CloudWatch 指標搭配使用](#)。

- 複寫 PSYNC 行為中的許多關鍵錯誤修正。如需詳細資訊，請參閱 [Redis OSS 2.8 版本備註](#)。
- 為了在多可用區域複寫群組中維持增強的複寫效能，並提高叢集穩定性，不再支援非 ElastiCache 複寫。
- 為了提升複寫群組中主要叢集和複本之間的資料一致性，複本不會再獨立於主要叢集而移出鍵。
- Redis 2.8.22 版 `appendonly` 及更新 OSS 版本 `appendfsync` 不支援 Redis OSS 組態變數和。
- 在記憶體不足的情況下，含大型輸出緩衝區的用戶端與複本叢集的連線可能會中斷。如果中斷連線，用戶端就需要重新連線。這類情況最可能發生在 PUBSUB 用戶端。

## ElastiCache ( Redis OSS ) 2.8.21 版

自 2.8.19 版以來新增的 Redis OSS改進包含許多錯誤修正。如需詳細資訊，請參閱 [Redis OSS 2.8 版本備註](#)。

## ElastiCache ( Redis OSS ) 2.8.19 版

自 2.8.6 版以來新增的 Redis OSS改進包括下列各項：

- 支援 HyperLogLog。如需詳細資訊，請參閱 [Redis OSS新資料結構：HyperLogLog](#)。
- 有序集合資料類型現已支援使用新命令 ZRANGEBYLEX、ZLEXCOUNT 和 ZREMRANGEBYLEX 的詞典編纂範圍查詢。
- 為了防止主要節點將過時資料傳送至複本節點，如果背景儲存（bgsave）子程序中止，則主節點會SYNC失敗。
- 支援 HyperLogLogBasedCommands CloudWatch 指標。如需詳細資訊，請參閱[Valkey 和 Redis 的指標 OSS](#)。

## ElastiCache ( Redis OSS ) 2.8.6 版

自 2.6.13 版以來新增的 Redis OSS改進包括以下內容：

- 提升僅供讀取複本的復原能力與容錯能力。
- 支援部分重新同步。
- 支援使用者定義且必須隨時可供使用的僅供讀取複本最低數量。
- 完整支援 pub/sub - 可通知用戶端伺服器相關事件。
- 系統會自動偵測主要節點故障，並將主要節點容錯移轉至次要節點。

## ElastiCache ( Redis OSS ) 2.6.13 版

Redis 2.6.13 OSS版是 Amazon ElastiCache ( Redis OSS ) OSS支援的 Redis 初始版本。Redis 2.6.13 OSS 不支援多可用區。

## Redis OSS版本生命週期結束排程

本節定義公告舊主要版本的生命週期終止（EOL）日期。這可讓您針對未來做出版本和升級決策。

**Note**

ElastiCache ( Redis OSS ) 從 5.0.0 到 5.0.5 的修補程式版本已棄用。使用 5.0.6 版 (或更高版)。

下表摘要說明每個版本及其公告EOL日期，以及建議的升級目標版本。

## 過去 EOL

| 來源次要版本                                                 | 建議升級目標                                                                                   | EOL 日期          |
|--------------------------------------------------------|------------------------------------------------------------------------------------------|-----------------|
| 3.2.4 版、3.2.6 版和 3.2.10 版                              | 6.2 版或更高版本                                                                               | 2023 年 7 月 31 日 |
|                                                        | <b>Note</b><br>對於 US-ISO-EA ST-1、US-ISO-WE ST-1 和 US-ISOB-E AST-1 區域，我們建議使用 5.0.6 或更高版本。 |                 |
| 2.8.24、2.8.23、2.8.22、2.8.21、2.8.19、2.8.12、2.8.6、2.6.13 | 6.2 版或更高版本                                                                               | 2023 年 1 月 13 日 |
|                                                        | <b>Note</b><br>對於 US-ISO-EA ST-1、US-ISO-WE ST-1 和 US-ISOB-E AST-1 區域，我們建                 |                 |

| 來源次要版本 | 建議升級目標                                                                                                                                     | EOL 日期 |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
|        | <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; text-align: center;">           議使用 5.0.6 或更高版本。         </div> |        |

## 支援的 ElastiCache ( Memcached ) 版本

ElastiCache 支援下列 Memcached 版本並升級至較新的版本。升級至較新版本時，請特別注意必須滿足的相關條件，以免升級失敗。

ElastiCache 適用於 Memcached 版本

- [Memcached 1.6.22 版](#)
- [Memcached 1.6.17 版](#)
- [Memcached 1.6.12 版](#)
- [Memcached 1.6.6 版](#)
- [Memcached 1.5.16 版](#)
- [Memcached 1.5.10 版](#)
- [Memcached 1.4.34 版](#)
- [Memcached 1.4.33 版](#)
- [Memcached 1.4.24 版](#)
- [Memcached 1.4.14 版](#)
- [Memcached 1.4.5 版](#)

### Memcached 1.6.22 版

ElastiCache ( Memcached ) 新增對 Memcached 1.6.22 版的支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.6.18](#) 的累積更新。

如需詳細資訊，請參閱 上的 Memcached 的 [ReleaseNotes1622](#) GitHub。

## Memcached 1.6.17 版

ElastiCache ( Memcached ) 新增對 Memcached 1.6.17 版的支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.6.17](#) 的累積更新。

如需詳細資訊，請參閱 上的 Memcached 的 [ReleaseNotes1617](#) GitHub。

## Memcached 1.6.12 版

ElastiCache ( Memcached ) 新增對 Memcached 1.6.12 版和傳輸中加密的支援。其中包括 [Memcached 1.6.6 版](#) 的錯誤修正與累積更新。

如需詳細資訊，請參閱 上的 Memcached 的 [ReleaseNotes1612](#) GitHub。

## Memcached 1.6.6 版

ElastiCache ( Memcached ) 新增對 Memcached 1.6.6 版的支援。它不包含新功能，但包含來自 [Memcached 1.5.16](#) 的錯誤修正和累積更新。ElastiCache ( Memcached ) 不包含對 [Extstore](#) 的支援。

如需詳細資訊，請參閱 上的 Memcached 的 [ReleaseNotes166](#) GitHub。

## Memcached 1.5.16 版

ElastiCache for Memcached 新增對 Memcached 1.5.16 版的支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.5.14](#) 與 [Memcached 1.5.15](#) 的累積更新。

如需詳細資訊，請參閱 上的 [Memcached 的 Memcached 1.5.16 版本備註](#) GitHub。

## Memcached 1.5.10 版

ElastiCache for Memcached 1.5.10 支援下列 Memcached 功能：

- 自動化 slab 重新平衡。
- 使用 murmur3 演算法進行更快速的雜湊資料表查閱。
- 分段LRU演算法。
- LRU 爬蟲程式至背景回收記憶體。
- --enable-seccomp：編譯時間選項。

它也推出 no\_modern 和 inline\_ascii\_resp 方法。如需詳細資訊，請參閱 [Memcached 1.5.10 參數變更](#)。

自 1.4.34 版 Memcached 以來新增 ElastiCache 的 Memcached 改進包括以下內容：

- 累積修正，例如的 ASCII multiget、CVE-2017-9951 和限制爬蟲metadumper。
- 在達到連線限制時關閉連線來獲得更好的連線管理。
- 對於 1MB 以上的項目大小，改善的項目大小管理。
- 透過將每一項目的記憶體需求減少一些位元組，獲得更好的效能和記憶體負擔改善。

如需詳細資訊，請參閱 上的 [Memcached 的 Memcached 1.5.10 版本備註](#) GitHub。

### Memcached 1.4.34 版

ElastiCache for Memcached 1.4.34 版未新增任何新功能至 1.4.33 版。1.4.34 版是一款錯誤修正版本，其比一般類似版本的涵蓋範圍更大。

如需詳細資訊，請參閱 上的 [Memcached 的 Memcached 1.4.34 版本備註](#) GitHub。

### Memcached 1.4.33 版

從 Memcached 1.4.24 版起新增的改良功能，包括下列項目：

- 可傾印特定 slab 類別、slab 類別清單或所有 slab 類別的所有中繼資料。如需詳細資訊，請參閱 [Memcached 1.4.31 版本備註](#)。
- 已改善對預設超過 1 megabyte (MB) 大型項目的支援。如需詳細資訊，請參閱 [Memcached 1.4.29 版本備註](#)。
- 可指定在詢問關閉用戶端之前用戶端可閒置的時間。

可動態增加 Memcached 可用的記憶體容量，而不需重新啟動叢集。如需詳細資訊，請參閱 [Memcached 1.4.27 版本備註](#)。

- 現已支援 fetchers、mutations 和 evictions 的記錄功能。如需詳細資訊，請參閱 [Memcached 1.4.26 版本備註](#)。
- 釋放的記憶體可以回收到全域集區，並重新指派給新的 slab 類別。如需詳細資訊，請參閱 [Memcached 1.4.25 版本備註](#)。
- 多項錯誤修正。
- 一些新的命令和參數。如需清單，請參閱 [Memcached 1.4.33 版新增參數](#)。

### Memcached 1.4.24 版

從 Memcached 1.4.14 版起新增的改良功能，包括下列項目：

- 使用背景程序進行至少最近使用的（LRU）管理。
- 已新增將 jenkins 或 murmur3 做為雜湊演算法的選項。
- 一些新的命令和參數。如需清單，請參閱[Memcached 1.4.24 版新增參數](#)。
- 多項錯誤修正。

### Memcached 1.4.14 版

從 Memcached 1.4.5 版起新增的改良功能，包括下列項目：

- 增強型 slab 重新平衡功能。
- 高效能與可擴展性提升。
- 引進 touch 命令，以更新現有項目的過期時間，而不需擷取該項目。
- 自動探索 - 可讓用戶端程式自動判斷快取叢集中的所有快取節點，並初始化及維持與這些節點的連線。

### Memcached 1.4.5 版

Memcached 1.4.5 版是 Amazon ElastiCache（Memcached）支援的初始引擎和版本。



## 與 Valkey 的主要版本行為和相容性差異

Valkey 7.2.6 與 Redis 7.2.5 OSS 的先前版本具有類似的相容性差異。如需最新的 Valkey 支援版本，請參閱 [支援的引擎和版本](#)。

## 與 Redis 的主要版本行為和相容性差異 OSS

### Important

以下頁面的結構目的是指出版本控制之間所有不相容的差異，並告知您升級至較新版本控制時應注意的事項。此清單包含您在升級時可能遇到的版本控制不相容問題。

您可以直接從目前的 Redis OSS 版本升級到最新的 Redis OSS 版本，而不需要循序升級。例如，您可以直接從 Redis 3.0 OSS 版升級至 7.0 版。

Redis OSS 版本會以語意版本識別，該版本包含 MAJOR、MINOR 和 PATCH 元件。例如，在 Redis OSS 4.0.10 中，主要版本為 4，次要版本為 0，修補程式版本為 10。這些值一般會根據下列慣例遞增：

- MAJOR 版本適用於 API 不相容的變更
- MINOR 版本適用於以向下相容方式新增的新功能
- PATCH 版本適用於向下相容的錯誤修正和非功能變更

我們建議始終使用指定 MAJOR.MINOR 版本中的最新修補程式版本，以便獲得最新的效能和穩定性改進。從 Redis OSS 6.0 開始，ElastiCache (Redis OSS) 將為每個 Redis OSS 次要版本提供單一版本，而不是提供多個修補程式版本。ElastiCache (Redis OSS) 會自動管理執行中快取叢集的修補程式版本，以確保效能提升和安全性提升。

同時建議您定期升級至最新主要版本，因為大部分主要改進功能並未向後移植至舊版。隨著將可用性 ElastiCache 擴展到新 AWS 區域，ElastiCache (Redis OSS) 支援新區域的兩個最新的 MAJOR.MINOR 版本。例如，如果新 AWS 區域啟動和最新的 MAJOR.MINOR ElastiCache (Redis OSS) 版本為 7.0 和 6.2，ElastiCache (Redis OSS) 將在新 AWS 區域中支援 7.0 和 6.2 版。隨著較新的 MAJOR.MINOR 版本 ElastiCache (Redis OSS) 發行，ElastiCache 將繼續為新發行的 ElastiCache (Redis OSS) 版本新增支援。若要進一步了解如何選擇的區域 ElastiCache，請參閱 [選擇區域和可用區域](#)。

進行跨主要或次要版本的升級時，請考慮下列清單，其中包括 OSS 隨著時間的推移，與 Redis 發行的行為和向後不相容的變更。

## Redis OSS 7.0 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 7.0 版本備註](#)。

- SCRIPT LOAD 與 SCRIPT FLUSH 不再傳播到複本。如果您需要一些指令碼的耐久性，建議您考慮使用 [Redis OSS函數](#)。
- Pubsub 頻道現在預設為封鎖新ACL使用者。
- LCS 命令已取代 STRALGO 命令。
- 已變更 ACL GETUSER 的格式，所有欄位皆會顯示標準存取字串模式。若您使用 ACL GETUSER 進行自動化，則應確認其能處理任一格式。
- SELECT、WAIT、ROLE、LASTSAVE、READONLY、READWRITE和的ACL類別ASKING已變更。
- 現在，每個子命令的命令統計資料由 INFO 命令顯示，而非顯示於頂層容器命令。
- 在部分邊緣情況下，LPOP、RPOP、ZPOPMIN 和 ZPOPMAX 命令的傳回值已變更。若您使用這些命令，則應確認版本備註，並評估您是否受到影響。
- SORT 和 SORT\_RO 命令現在需存取整個金鑰空間，以使用 GET 和 BY 引數。

## Redis OSS 6.2 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 6.2 版本備註](#)。

- TIME、、ECHOROLE和 LASTSAVE命令的ACL旗標已變更。這可能會導致先前允許的命令遭拒，反之亦然。

### Note

這些命令都不會修改或授予資料存取權。

- 從 Redis OSS 6.0 升級時，從地圖回應傳回至 lua 指令碼的索引鍵/值對順序會變更。如果您的指令碼使用或redis.setresp()傳回映射（Redis 6.0 OSS 中新增），請考慮指令碼可能會在升級時中斷的影響。

## Redis OSS 6.0 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 6.0 版本備註](#)。

- 允許的資料庫數量上限已從 120 萬降低至 1 萬。預設值為 16，我們不鼓勵使用大於此的值，因為發現效能和記憶體問題。

- 將 `AutoMinorVersionUpgrade` 參數設定為是，且 ElastiCache ( Redis OSS ) 將透過自助更新管理次要版本升級。這將透過自助更新行銷活動，經由標準客戶通知管道來處理。如需詳細資訊，請參閱 [中的自助式更新 ElastiCache](#)。

## Redis OSS 5.0 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 5.0 版本備註](#)。

- 指令碼是依效果複寫，而不是在複本上重新執行指令碼。這一般會改善效能，但可能會增加在主要項目和複本之間複寫的資料量。有一個選項可還原至 ElastiCache ( Redis OSS ) 5.0 中僅可用的先前行為。
- 如果您從 Redis OSS 4.0 升級，LUA 指令碼中的某些命令會以與舊版不同的順序傳回引數。在 Redis OSS 4.0 中，Redis OSS 會以文字方式排序一些回應，讓回應具有決定性，當指令碼依效果複寫時，不會套用此排序。
- 在 Redis OSS 5.0.3 及更高版本，ElastiCache ( Redis OSS ) 會將部分 IO 工作卸載至執行個體類型上超過 4 的背景核心 VCPUs。這可能會變更 Redis 的效能特性，OSS 並變更某些指標的值。如需詳細資訊，請參閱 [應監控哪些指標？](#)，了解您是否需要變更您監控的指標。

## Redis OSS 4.0 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 4.0 版本備註](#)。

- 慢速日誌現在會記錄另外兩個引數，即用戶端名稱和位址。除非您明確依賴每個包含 3 個值的慢速日誌項目，否則此變更應回溯相容。
- `CLUSTER NODES` 命令現在會傳回略為不同的格式，不回溯相容。建議客戶不要使用此命令了解叢集中的節點，而應使用 `CLUSTER SLOTS`。

## 過去 EOL

### Redis OSS 3.2 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 3.2 版本備註](#)。

- 此版本沒有需注意的相容性變更。

如需詳細資訊，請參閱 [Redis OSS 版本生命週期結束排程](#)。

## Redis OSS 2.8 行為和向後不相容的變更

如需變更的完整清單，請參閱 [Redis OSS 2.8 版本備註](#)。

- 從 Redis OSS 2.8.22 開始，OSS AOF ElastiCache (Redis OSS) 不再支援 Redis。資料需要持久保存時，建議使用 MemoryDB。
- 從 Redis OSS 2.8.22 開始，ElastiCache (Redis OSS) 不再支援將複本連接至中託管的主要複本 ElastiCache。升級時，外部複本將中斷連線，且無法重新連線。我們建議您使用用戶端快取，可在 Redis 6.0 OSS 中取得，作為外部複本的替代方案。
- 如果沒有金鑰，TTL 和 PTTTL 命令現在會傳回 -2，而如果有金鑰，但沒有關聯的過期，則傳回 -1。Redis OSS 2.6 和先前用來針對這兩個條件傳回 -1 的版本。
- 如果不使用 STORE 選項，具 ALPHA 的 SORT 現在會根據地區定序設定排序。

如需詳細資訊，請參閱 [Redis OSS 版本生命週期結束排程](#)。

## 解決封鎖的 Valkey 或 Redis OSS 引擎升級

如下表所示，如果您有待處理的擴展操作，則會封鎖您的 Valkey 或 Redis OSS 引擎升級操作。

| 待定作業      | 封鎖的作業  |
|-----------|--------|
| 向上擴展      | 立即升級引擎 |
| 引擎升級      | 立即向上擴展 |
| 向上擴展與升級引擎 | 立即向上擴展 |
|           | 立即升級引擎 |

### 若要解決封鎖的 Redis OSS 引擎升級

- 執行以下任意一項：
  - 透過清除立即套用核取方塊，為下一個維護時段排程 Redis OSS 引擎升級操作。
    - 搭配 CLI 使用 `--no-apply-immediately`。搭配 API 使用 `ApplyImmediately=false`。
  - 請等到下一個維護時段（或之後）執行 Redis OSS 引擎升級操作。
  - 使用選取的立即套用核取方塊，將 Redis OSS 擴展操作新增至此叢集修改。

搭配 CLI 使用 `--apply-immediately`。搭配 API 使用 `ApplyImmediately=true`。

此方法會立即執行引擎升級，而取消下一個維護時段的引擎升級。

## ElastiCache 最佳實務和快取策略

您可以在下方找到 Amazon 的建議最佳實務 ElastiCache。遵循這些內容可改善快取的效能和可靠性。

### 主題

- [整體最佳實務](#)
- [支援和限制的 Valkey、Redis OSS 和 Memcached 命令](#)
- [Valkey 和 Redis OSS 組態和限制](#)
- [IPv6 Valkey、Redis OSS 和 Memcached 的用戶端範例](#)
- [用戶端的最佳實務 \(Valkey 和 RedisOSS\)](#)
- [用戶端的最佳實務 \(Memcached\)](#)
- [TLS 已啟用雙堆疊 ElastiCache 叢集](#)
- [管理 Valkey 和 Redis 的預留記憶體 OSS](#)
- [使用 Valkey 和 Redis OSS 自行設計的叢集時的最佳實務](#)
- [Memcached 的快取策略](#)

## 整體最佳實務

您可以在下方找到在 中使用 Valkey、Redis OSS 和 Memcached 介面的最佳評價的相關資訊 ElastiCache。

- 使用啟用叢集模式的組態 – 啟用叢集模式可讓快取水平擴展，以達到比停用叢集模式的組態更高的儲存和輸送量。ElastiCache serverless 僅適用於啟用叢集模式的組態。
- 使用長時間連線 – 建立新連線非常昂貴，而且需要快取的時間 CPU 和資源。盡可能重複使用連線 (例如使用連線集區)，以將此費用分攤至多個命令。
- 從複本中讀取 – 如果您使用的是無 ElastiCache 伺服器或已佈建僅供讀取複本 (自行設計的叢集)，會將僅供讀取導向複本，以達到更好的可擴展性和/或更低的延遲。從複本讀取的內容最終會與主節點保持一致。

在自行設計的叢集中，請避免將讀取請求導向單一讀取複本，因為節點失敗時，讀取可能會暫時無法使用。請設定用戶端將讀取請求導向到至少兩個讀取複本，或將讀取導向單一複本和主節點。

在無 ElastiCache 伺服器中，從複本連接埠（6380）讀取將盡可能將讀取導向用戶端的本機可用區域，從而降低擷取延遲。發生故障時，它將自動回復至其他節點。

- 避免費時命令 - 避免執行任何需要大量運算或輸入/輸出的操作，例如 KEYS 和 SMEMBERS 命令。我們建議此方法，因為這些操作會增加叢集上的負載，並會影響叢集的效能。請改為使用 SCAN 和 SSCAN 命令。
- 遵循 Lua 最佳實務 - 避免長時間執行的 Lua 指令碼，並且一律預先宣告用於 Lua 指令碼的索引鍵。我們建議此方法來判斷 Lua 指令碼並未使用跨位置命令。確認用於 Lua 指令碼中的鍵屬於相同位置。
- 使用碎片 pub/sub – 使用 Valkey 或 Redis OSS 支援高輸送量的 pub/sub 工作負載時，建議您使用 [碎片 pub/sub](#)（搭配 Valkey 使用，搭配 Redis OSS 7 或更新版本）。在啟用叢集模式的叢集中的傳統 pub/sub 會將訊息廣播到叢集中的所有節點，因此可能導致高 EngineCPUUtilization。請注意，在無 ElastiCache 伺服器中，傳統 pub/sub 命令內部使用碎片 pub/sub 命令。

## 支援和限制的 Valkey、Redis OSS和 Memcached 命令

### 支援的 Valkey 和 Redis OSS命令

#### 支援的 Valkey 和 Redis OSS命令

無伺服器快取支援下列 Valkey 和 Redis OSS命令。除了這些命令之外，還支援這些 [支援的 Valkey 和 Redis OSS命令](#)。

#### 點陣圖命令

- BITCOUNT

計算字串中設定的位元數 (母體計數)

#### [進一步了解](#)

- BITFIELD

對字串執行任意 bitfield integer 操作。

#### [進一步了解](#)

- BITFIELD\_RO

對字串執行任意唯讀 bitfield integer 操作。

### [進一步了解](#)

- BITOP

對多個字串執行位元操作，並且儲存結果。

### [進一步了解](#)

- BITPOS

尋找第一個設定 (1) 或清除 (0) 位元。

### [進一步了解](#)

- GETBIT

依偏移傳回位元值。

### [進一步了解](#)

- SETBIT

設定或清除字串值偏移處的位元。建立金鑰 (如不存在)。

### [進一步了解](#)

## 叢集管理命令

- CLUSTER COUNTKEYSINSLOT

傳回雜湊位置中的金鑰數目。

### [進一步了解](#)

- CLUSTER GETKEYSINSLOT

傳回雜湊位置中的金鑰名稱。

### [進一步了解](#)

- CLUSTER INFO

傳回節點狀態的相關資訊。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- CLUSTER KEYSLOT

傳回金鑰的雜湊位置。

[進一步了解](#)

- CLUSTER MYID

傳回節點的 ID。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- CLUSTER NODES

傳回節點的叢集組態。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- CLUSTER REPLICAS

列出主節點的複本節點。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- CLUSTER SHARDS

傳回叢集位置對碎片的映射。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- CLUSTER SLOTS

傳回叢集位置對節點的映射。在無伺服器快取中，傳回已對用戶端公開的單一虛擬「碎片」的狀態。

[進一步了解](#)

- READONLY

為與 Valkey 或 Redis OSS叢集複本節點的連線啟用唯讀查詢。

[進一步了解](#)

- READWRITE

啟用對 Valkey 或 Redis OSS叢集複本節點連線的讀寫查詢。



[進一步了解](#)

## 連線管理命令

- AUTH

驗證連線。

[進一步了解](#)

- CLIENT GETNAME

傳回連線的名稱。

[進一步了解](#)

- CLIENT REPLY

指示伺服器是否回覆命令。

[進一步了解](#)

- CLIENT SETNAME

設定連線名稱。

[進一步了解](#)

- ECHO

傳回所指定的字串。

[進一步了解](#)

- HELLO

使用 Valkey 或 Redis OSS 伺服器交握。

[進一步了解](#)

- PING

傳回伺服器的活動回應。

[進一步了解](#)

- QUIT

關閉連線。

[進一步了解](#)

- RESET

重設連線。

[進一步了解](#)

- SELECT

變更選取的資料庫。

[進一步了解](#)

## 一般命令

- COPY

將金鑰的值複製到新金鑰。

[進一步了解](#)

- DEL

刪除一或多個金鑰。

[進一步了解](#)

- DUMP

傳回儲存於金鑰之值的序列化表示。

[進一步了解](#)

- EXISTS

判斷有一或多個值存在。

[進一步了解](#)

- EXPIRE

支援和限制的 Valkey、Redis OSS和 Memcached 命令

設定金鑰的到期時間 (以秒為單位)。

### [進一步了解](#)

- EXPIREAT

將金鑰的到期時間設為 Unix 時間戳記。

### [進一步了解](#)

- EXPIRETIME

以 Unix 時間戳記傳回金鑰的到期時間。

### [進一步了解](#)

- PERSIST

移除金鑰的到期時間。

### [進一步了解](#)

- PEXPIRE

設定金鑰的到期時間 (以毫秒為單位)。

### [進一步了解](#)

- PEXPIREAT

將金鑰的到期時間設為 Unix 毫秒時間戳記。

### [進一步了解](#)

- PEXPIRETIME

以 Unix 毫秒時間戳記傳回金鑰的到期時間。

### [進一步了解](#)

- PTTL

傳回以毫秒為單位的金鑰到期時間。

### [進一步了解](#)

- RANDOMKEY

從資料庫傳回隨機金鑰名稱。

### [進一步了解](#)

- RENAME

重新命名金鑰並覆寫目的地。

### [進一步了解](#)

- RENAMENX

僅在目標金鑰名稱不存在時重新命名金鑰。

### [進一步了解](#)

- RESTORE

從值的序列化表示建立金鑰。

### [進一步了解](#)

- SCAN

逐一查看資料庫中的金鑰名稱。

### [進一步了解](#)

- SORT

排序清單中、集合中或已排序集合中的元素，並選擇性地儲存結果。

### [進一步了解](#)

- SORT\_RO

傳回清單、集合或已排序清單的已排序元素。

### [進一步了解](#)

- TOUCH

傳回在更新上次存取時間後，所指定金鑰當中現有的金鑰數目。

### [進一步了解](#)

- TTL

傳回以秒為單位的金鑰到期時間。

### [進一步了解](#)

- TYPE

判斷儲存於金鑰的值類型。

### [進一步了解](#)

- UNLINK

以非同步方式刪除一或多個金鑰。

### [進一步了解](#)

## 地理空間命令

- GEOADD

將一或多個成員新增到地理空間索引。建立金鑰 (如不存在)。

### [進一步了解](#)

- GEODIST

傳回兩個地理空間索引成員之間的距離。

### [進一步了解](#)

- GEOHASH

從地理空間索引傳回成員作為 geohash 字串。

### [進一步了解](#)

- GEOPOS

從地理空間索引傳回成員的經度和緯度。

### [進一步了解](#)

- GEORADIUS

查詢距離某個座標一定距離內的成員的地理空間索引，並選擇性地儲存結果。

[進一步了解](#)

- GEORADIUS\_RO

傳回距離某個座標一定距離內的地理空間索引的成員。

[進一步了解](#)

- GEORADIUSBYMEMBER

查詢距離某個成員一定距離內的成員的地理空間索引，並選擇性地儲存結果。

[進一步了解](#)

- GEORADIUSBYMEMBER\_RO

傳回距離某個成員一定距離內的地理空間索引的成員。

[進一步了解](#)

- GEOSEARCH

查詢方形或圓形區域內成員的地理空間索引。

[進一步了解](#)

- GEOSEARCHSTORE

查詢方形或圓形區域內成員的地理空間索引，並選擇性地儲存結果。

[進一步了解](#)

## 雜湊命令

- HDEL

從雜湊中刪除一或多個欄位及其值。刪除沒有剩餘任何欄位的雜湊。

[進一步了解](#)

- HEXISTS

判斷雜湊中是否有欄位存在。

[進一步了解](#)

- HGET

傳回雜湊中某個欄位的值。

[進一步了解](#)

- HGETALL

傳回雜湊中的所有欄位和值。

[進一步了解](#)

- HINCRBY

按某個數字遞增雜湊中欄位的整數值。如果欄位不存在，則使用 0 作為初始值。

[進一步了解](#)

- HINCRBYFLOAT

按某個數字遞增欄位的浮點值。如果欄位不存在，則使用 0 作為初始值。

[進一步了解](#)

- HKEYS

傳回雜湊中的所有欄位。

[進一步了解](#)

- HLEN

傳回雜湊中的欄位數。

[進一步了解](#)

- HMGET

傳回雜湊中所有欄位的值。

[進一步了解](#)

- HMSET

設定多個欄位的值。

[進一步了解](#)

- HRANDFIELD

傳回雜湊中的一或多個隨機欄位。

[進一步了解](#)

- HSCAN

逐一查看雜湊的欄位和值。

[進一步了解](#)

- HSET

建立或修改雜湊中欄位的值。

[進一步了解](#)

- HSETNX

僅在欄位不存在時設定雜湊中欄位的值。

[進一步了解](#)

- HSTRLEN

傳回欄位值的長度。

[進一步了解](#)

- HVALS

傳回雜湊中的所有值。

[進一步了解](#)

## HyperLogLog 命令

- PFADD

將元素新增至 HyperLogLog 金鑰。建立索引鍵 (如不存在)。

[進一步了解](#)

- PFCOUNT



傳回 HyperLogLog 鍵 ( s ) 觀察到的 set ( s ) 近似基數。

### [進一步了解](#)

- PFMERGE

將一或多個 HyperLogLog 值合併至單一金鑰。

### [進一步了解](#)

## 清單命令

- BLMOVE

從清單彈現元素、將其推送到另一個清單並傳回。封鎖直到元素可用為止。最後一個元素移動後刪除清單。

### [進一步了解](#)

- BLMPOP

彈現多個清單之一的第一個元素。封鎖直到元素可用為止。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- BLPOP

移除並傳回清單中的第一個元素。封鎖直到元素可用為止。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- BRPOP

移除並傳回清單中的最後一個元素。封鎖直到元素可用為止。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- BRPOPLPUSH

從清單彈現元素、將其推送到另一個清單並傳回。封鎖直到元素可用為止。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- LINDEX

按索引傳回清單中的元素。

### [進一步了解](#)

- LINSERT

將某一個元素插入清單中另一個元素之前或之後。

### [進一步了解](#)

- LLEN

傳回清單的長度。

### [進一步了解](#)

- LMOVE

從某一個清單彈現元素並推送至另一個清單後，傳回該元素。最後一個元素移動後刪除清單。

### [進一步了解](#)

- LMPOP

從清單中移除多個元素後，傳回這些元素。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- LPOP

從清單中移除第一個元素後將該元素傳回。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- LPOS

傳回清單中相符元素的索引。

### [進一步了解](#)

- LPUSH

在清單前面加上一或多個元素。建立鍵 (如不存在)。

### [進一步了解](#)

- LPUSHX

僅於清單存在時，在清單前面加上一或多個元素。

### [進一步了解](#)

- LRANGE

從清單中傳回某一範圍的元素。

### [進一步了解](#)

- LREM

從清單中移除元素。最後一個元素移除後刪除清單。

### [進一步了解](#)

- LSET

按索引設定清單中元素的值。

### [進一步了解](#)

- LTRIM

移除清單兩端的元素。若元素都已修剪，則刪除清單。

### [進一步了解](#)

- RPOP

傳回並移除清單中的最後幾個元素。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- RPOPLPUSH

移除清單中的最後一個元素並推送至另一個清單後，傳回該元素。最後一個元素彈現後刪除清單。

### [進一步了解](#)

- RPUSH

附加一或多個元素至清單。建立鍵 (如不存在)。

### [進一步了解](#)

- RPUSHX

僅於清單存在時附加元素至清單。

[進一步了解](#)

## Pub/Sub 命令

### Note

PUBSUB 命令會在內部使用碎片 PUBSUB，因此會混合頻道名稱。

- PUBLISH

將訊息發佈至通道。

[進一步了解](#)

- PUBSUB CHANNELS

傳回作用中通道。

[進一步了解](#)

- PUBSUB NUMSUB

將訂閱用戶數傳回至通道。

[進一步了解](#)

- PUBSUB SHARDCHANNELS

傳回作用中碎片通道。

[PUBSUB-SHARDCHANNELS](#)

- PUBSUB SHARDNUMSUB

傳回碎片通道的訂閱用戶數。

[PUBSUB-SHARDNUMSUB](#)

- SPUBLISH

將訊息發佈至碎片通道

[進一步了解](#)

- SSUBSCRIBE

偵聽發佈至碎片通道的訊息。

[進一步了解](#)

- SUBSCRIBE

偵聽發佈至通道的訊息。

[進一步了解](#)

- SUNSUBSCRIBE

停止偵聽發佈至碎片通道的訊息。

[進一步了解](#)

- UNSUBSCRIBE

停止偵聽發佈至通道的訊息。

[進一步了解](#)

## 指令碼命令

- EVAL

執行伺服器端 Lua 指令碼。

[進一步了解](#)

- EVAL\_RO

執行唯讀伺服器端 Lua 指令碼。

[進一步了解](#)

- EVALSHA

依摘要執行伺服器端 Lua SHA1 指令碼。

[進一步了解](#)

- EVALSHA\_RO

依摘要執行唯讀伺服器端 Lua SHA1 指令碼。

[進一步了解](#)

- SCRIPT EXISTS

判斷指令碼快取中是否存在伺服器端 Lua 指令碼。

[進一步了解](#)

- SCRIPT FLUSH

目前無操作的指令碼快取是由服務進行管理。

[進一步了解](#)

- SCRIPT LOAD

將伺服器端 Lua 指令碼載入指令碼快取中。

[進一步了解](#)

## 伺服器管理命令

- ACL CAT

列出ACL類別或類別內的命令。

[進一步了解](#)

- ACL GENPASS

產生偽隨機的安全密碼，可用來識別ACL使用者。

[進一步了解](#)

- ACL GETUSER

列出使用者的ACL規則。

[進一步了解](#)

- ACL LIST

傾印ACL檔案格式的有效規則。

### [進一步了解](#)

- ACL USERS

列出ACL所有使用者。

### [進一步了解](#)

- ACL WHOAMI

傳回目前連線的已驗證使用者名稱。

### [進一步了解](#)

- DBSIZE

傳回目前選取的資料庫中的索引鍵數目。此操作不保證在所有位置都是原子。

### [進一步了解](#)

- COMMAND

傳回所有命令的詳細資訊。

### [進一步了解](#)

- COMMAND COUNT

傳回命令計數。

### [進一步了解](#)

- COMMAND DOCS

傳回有關一個、多個或所有命令的文件資訊。

### [進一步了解](#)

- COMMAND GETKEYS

從任意命令中擷取索引鍵名稱。

### [進一步了解](#)

- COMMAND GETKEYSANDFLAGS

擷取任意命令的索引鍵名稱和存取旗標。

### [進一步了解](#)

- COMMAND INFO

傳回有關一個、多個或所有命令的資訊。

### [進一步了解](#)

- COMMAND LIST

傳回命令名稱清單。

### [進一步了解](#)

- FLUSHALL

移除所有資料庫中的所有索引鍵。此操作不保證在所有位置都是原子。

### [進一步了解](#)

- FLUSHDB

移除目前資料庫中的所有索引鍵。此操作不保證在所有位置都是原子。

### [進一步了解](#)

- INFO

傳回有關伺服器的資訊和統計資料。

### [進一步了解](#)

- LOLWUT

顯示電腦圖片和 Valkey 或 Redis OSS 版本。

### [進一步了解](#)

- ROLE

傳回複寫角色。

### [進一步了解](#)

- TIME



傳回伺服器時間。

[進一步了解](#)

## 設定命令

- SADD

將一或多個成員新增至集合。建立鍵 (如不存在)。

[進一步了解](#)

- SCARDT

傳回集合中的成員數。

[進一步了解](#)

- SDIFF

傳回多個集合的差異。

[進一步了解](#)

- SDIFFSTORE

將多個集合的差異儲存在索引鍵中。

[進一步了解](#)

- SINTER

傳回多個集合的交集。

[進一步了解](#)

- SINTERCARD

傳回多個集合之交集的成員數。

[進一步了解](#)

- SINTERSTORE

將多個集合的交集儲存在索引鍵中。

[進一步了解](#)

- SISMEMBER

判斷成員是否屬於集合。

[進一步了解](#)

- SMEMBERS

傳回集合的所有成員。

[進一步了解](#)

- SMISMEMBER

判斷多個成員是否屬於某個集合。

[進一步了解](#)

- SMOVE

將成員從某一個集合移動到另一個集合。

[進一步了解](#)

- SPOP

從集合中移除一或多個隨機成員後傳回這些成員。最後一個成員彈現後刪除集合。

[進一步了解](#)

- SRANDMEMBER

從集合中取得一或多個隨機成員

[進一步了解](#)

- SREM

從集合中移除一或多個成員。最後一個成員移除後刪除集合。

[進一步了解](#)

- SSCAN

逐一查看集合的成員。

[進一步了解](#)

- SUNION

傳回多個集合的聯集。

[進一步了解](#)

- SUNIONSTORE

將多個集合的聯集儲存在索引鍵中。

[進一步了解](#)

已排序集合命令

- BZMPOP

按分數從一或多個已排序集合移除並傳回成員。封鎖直到成員可用為止。最後一個成員彈現後刪除已排序集合。

[進一步了解](#)

- BZPOPMAX

從一或多個已排序集合移除並傳回分數最高的成員。封鎖直到成員可用為止。最後一個成員彈現後刪除已排序集合。

[進一步了解](#)

- BZPOPMIN

從一或多個已排序集合移除並傳回分數最低的成員。封鎖直到成員可用為止。最後一個成員彈現後刪除已排序集合。

[進一步了解](#)

- ZADD

將一或多個成員新增至已排序集合，或更新其分數。建立索引鍵 (如不存在)。

[進一步了解](#)

- ZCARD

傳回已排序集合中的成員數。

### [進一步了解](#)

- ZCOUNT

傳回已排序集合中分數落在某個範圍內的成員數。

### [進一步了解](#)

- ZDIFF

傳回多個已排序集合的差異。

### [進一步了解](#)

- ZDIFFSTORE

將多個已排序集合的差異儲存在索引鍵中。

### [進一步了解](#)

- ZINCRBY

遞增排列已排序集合中成員的分數。

### [進一步了解](#)

- ZINTER

傳回多個已排序集合的交集。

### [進一步了解](#)

- ZINTERCARD

傳回多個已排序集合之交集的成員數。

### [進一步了解](#)

- ZINTERSTORE

將多個已排序集合的交集儲存在索引鍵中。

### [進一步了解](#)

- ZLEXCOUNT

傳回已排序集合中某一字典順序範圍內的成員數。

### [進一步了解](#)

- ZMPOP

從一或多個已排序集合移除最高得分或最低得分成員後，傳回該成員。最後一個成員彈現後刪除已排序集合。

### [進一步了解](#)

- ZMSCORE

傳回已排序集合中一或多個成員的分數。

### [進一步了解](#)

- ZPOPMAX

從已排序集合移除最高得分成員後，傳回這些成員。最後一個成員彈現後刪除已排序集合。

### [進一步了解](#)

- ZPOPMIN

從已排序集合移除最低得分成員後，傳回這些成員。最後一個成員彈現後刪除已排序集合。

### [進一步了解](#)

- ZRANDMEMBER

從已排序集合中傳回一或多個隨機成員。

### [進一步了解](#)

- ZRANGE

傳回已排序集合中某個索引範圍內的成員。

### [進一步了解](#)

- ZRANGEBYLEX

傳回已排序集合中某一字典順序範圍內的成員。

### [進一步了解](#)

- ZRANGEBYSCORE

傳回已排序集合中某個分數範圍內的成員。

[進一步了解](#)

- ZRANGESTORE

將已排序集合中某個範圍的成員儲存到索引鍵中。

[進一步了解](#)

- ZRANK

傳回依分數遞增順序排列的已排序集合中成員的索引。

[進一步了解](#)

- ZREM

從已排序集合中移除一或多個成員。若所有成員都已移除，則刪除已排序集合。

[進一步了解](#)

- ZREMRANGEBYLEX

移除已排序集合中某一字典順序範圍內的成員。若所有成員都已移除，則刪除已排序集合。

[進一步了解](#)

- ZREMRANGEBYRANK

移除已排序集合中某個索引範圍內的成員。若所有成員都已移除，則刪除已排序集合。

[進一步了解](#)

- ZREMRANGEBYSCORE

移除已排序集合中某個分數範圍內的成員。若所有成員都已移除，則刪除已排序集合。

[進一步了解](#)

- ZREVRANGE

依反向順序傳回已排序集合中某個索引範圍內的成員。

[進一步了解](#)

- ZREVRANGEBYLEX

依反向順序傳回已排序集合中某一字典順序範圍內的成員。

[進一步了解](#)

- ZREVRANGEBYSCORE

依反向順序傳回已排序集合中某個分數範圍內的成員。

[進一步了解](#)

- ZREVRANK

傳回依分數遞減順序排列的已排序集合中成員的索引。

[進一步了解](#)

- ZSCAN

逐一查看已排序集合的成員和分數。

[進一步了解](#)

- ZSCORE

傳回已排序集合中成員的分數。

[進一步了解](#)

- ZUNION

傳回多個已排序集合的聯集。

[進一步了解](#)

- ZUNIONSTORE

將多個已排序集合的聯集儲存在索引鍵中。

[進一步了解](#)

## 串流命令

- XACK

支援和限制的 Valkey、Redis OSS和 Memcached 命令

傳回已由串流的取用者群組成員成功確認的訊息數。

### [進一步了解](#)

- XADD

將新訊息附加至串流。建立索引鍵 (如不存在)。

### [進一步了解](#)

- XAUTOCLAIM

變更或取得取用者群組中訊息的擁有權，就像訊息傳遞至取用者群組成員一樣。

### [進一步了解](#)

- XCLAIM

變更或取得取用者群組中訊息的擁有權，就像訊息傳遞至取用者群組成員一樣。

### [進一步了解](#)

- XDEL

從串流中移除訊息後傳回訊息數。

### [進一步了解](#)

- XGROUP CREATE

建立取用者群組。

### [進一步了解](#)

- XGROUP CREATECONSUMER

建立取用者群組中的取用者。

### [進一步了解](#)

- XGROUP DELCONSUMER

從取用者群組刪除取用者。

### [進一步了解](#)

- XGROUP DESTROY



銷毀取用者群組。

### [進一步了解](#)

- XGROUP SETID

設定取用者群組的上次傳遞 ID。

### [進一步了解](#)

- XINFO CONSUMERS

傳回取用者群組中取用者的清單。

### [進一步了解](#)

- XINFO GROUPS

傳回串流的取用者群組清單。

### [進一步了解](#)

- XINFO STREAM

傳回串流的相關資訊。

### [進一步了解](#)

- XLEN

傳回串流中的訊息數。

### [進一步了解](#)

- XPENDING

從串流取用者群組的待處理項目清單中傳回資訊和項目。

### [進一步了解](#)

- XRANGE

傳回 範圍內串流的訊息IDs。

### [進一步了解](#)

- XREAD

傳回來自多個串流且IDs大於所請求串流的訊息。封鎖直到訊息可用為止。

### [進一步了解](#)

- XREADGROUP

從串流傳回群組中取用者的新訊息和歷史訊息。封鎖直到訊息可用為止。

### [進一步了解](#)

- XREVRANGE

IDs 以相反順序傳回 範圍內串流的訊息。

### [進一步了解](#)

- XTRIM

從串流開頭刪除訊息。

### [進一步了解](#)

## 字串命令

- APPEND

將字串附加至索引鍵的值。建立索引鍵 (如不存在)。

### [進一步了解](#)

- DECR

以 1 為單位遞減索引鍵的整數值。如果索引鍵不存在，則使用 0 作為初始值。

### [進一步了解](#)

- DECRBY

從索引鍵的整數值遞減某個數字。如果索引鍵不存在，則使用 0 作為初始值。

### [進一步了解](#)

- GET

傳回索引鍵的字串值。

### [進一步了解](#)

- GETDEL

刪除索引鍵後，傳回索引鍵的字串值。

### [進一步了解](#)

- GETEX

設定到期時間後，傳回索引鍵的字串值。

### [進一步了解](#)

- GETRANGE

傳回儲存於索引鍵之字串的子字串。

### [進一步了解](#)

- GETSET

設定新值後，傳回索引鍵的前一個字串值。

### [進一步了解](#)

- INCR

以 1 為單位遞增索引鍵的整數值。如果索引鍵不存在，則使用 0 作為初始值。

### [進一步了解](#)

- INCRBY

以某個數字為單位遞增索引鍵的整數值。如果索引鍵不存在，則使用 0 作為初始值。

### [進一步了解](#)

- INCRBYFLOAT

以某個數字為單位遞增索引鍵的浮點值。如果索引鍵不存在，則使用 0 作為初始值。

### [進一步了解](#)

- LCS

尋找最長的常用子字串。

[進一步了解](#)

- MGET

以原子方式傳回一或多個索引鍵的字串值。

[進一步了解](#)

- MSET

以原子方式建立或修改一或多個索引鍵的字串值。

[進一步了解](#)

- MSETNX

僅在所有索引鍵都不存在時，以原子方式修改一或多個索引鍵的字串值。

[進一步了解](#)

- PSETEX

設定索引鍵的字串值和毫秒為單位的期時間。建立索引鍵 (如不存在)。

[進一步了解](#)

- SET

設定索引鍵的字串值，並忽略其類型。建立索引鍵 (如不存在)。

[進一步了解](#)

- SETEX

設定索引鍵的字串值和到期時間。建立索引鍵 (如不存在)。

[進一步了解](#)

- SETNX

僅於索引鍵不存在時，才設定索引鍵的字串值。

[進一步了解](#)

- SETRANGE

依某個偏移量將字串值的一部分覆寫為另一個字串值。建立索引鍵 (如不存在)。

[進一步了解](#)

- STRLEN

傳回字串值的長度。

[進一步了解](#)

- SUBSTR

從字串值傳回子字串。

[進一步了解](#)

## 交易命令

- DISCARD

捨棄交易。

[進一步了解](#)

- EXEC

執行交易中的所有命令。

[進一步了解](#)

- MULTI

開始交易。

[進一步了解](#)

## 受限的 Valkey 和 Redis OSS 命令

若要提供受管服務體驗，ElastiCache 會限制存取某些需要進階權限的快取引擎特定命令。執行 Redis 的快取無法使用下列命令：

- `acl setuser`
- `acl load`
- `acl save`

- `acl deluser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster addslotsrange`
- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

此外，無伺服器快取無法使用下列命令：

- `acl log`
- `client caching`
- `client getredir`
- `client id`

- `client info`
- `client kill`
- `client list`
- `client no-evict`
- `client pause`
- `client tracking`
- `client trackinginfo`
- `client unblock`
- `client unpause`
- `cluster count-failure-reports`
- `fcall`
- `fcall_ro`
- `function`
- `function delete`
- `function dump`
- `function flush`
- `function help`
- `function kill`
- `function list`
- `function load`
- `function restore`
- `function stats`
- `keys`
- `lastsave`
- `latency`
- `latency doctor`
- `latency graph`
- `latency help`
- `latency histogram`
- `latency history`

- latency latest
- latency reset
- memory
- memory doctor
- memory help
- memory malloc-stats
- memory purge
- memory stats
- memory usage
- monitor
- move
- object
- object encoding
- object freq
- object help
- object idletime
- object refcount
- pfdebug
- pfselftest
- psubscribe
- pubsub numpat
- punsubscribe
- script kill
- slowlog
- slowlog get
- slowlog help
- slowlog len
- slowlog reset
- swapdb
- unwatch



- wait
- watch

## 支援的 Memcached 命令

ElastiCache Serverless for Memcached 支援開放原始碼 memcached 1.6 中的所有 memcached [命令](#)，但下列除外：

- 用戶端連線需要 TLS，因為不支援結果UDP通訊協定。
- 不支援二進位通訊協定，因為已在 memcached 1.6 中正式[棄用](#)。
- GET/GETS 命令限制為 16KB，以避免可能遭到對伺服器進行大量金鑰擷取的 DoS 攻擊。
- 延遲的 flush\_all 命令將遭到拒絕，並顯示 CLIENT\_ERROR。
- 不支援設定引擎或揭露有關引擎狀態或日誌之內部資訊的命令，例如：
  - 對於 STATS 命令，僅支援 stats 和 stats reset。其他變化都將傳回 ERROR
  - lru / lru\_crawler - 修改 LRU和LRU爬蟲程式設定
  - watch - 監看 memcached 伺服器日誌
  - verbosity - 設定伺服器日誌層級
  - me - 不支援中繼偵錯 ( me ) 命令

## Valkey 和 Redis OSS組態和限制

Valkey 和 Redis OSS引擎各提供許多組態參數，其中一些可在 ElastiCache ( Redis OSS ) 中修改，而其中一些無法修改以提供穩定的效能和可靠性。

### 無伺服器快取

對於無伺服器快取，不會使用參數群組，且無法修改所有 Valkey 或 Redis OSS組態。已備妥下列 Valkey 或 Redis OSS 參數：

| 名稱                 | 詳細資訊        | 描述                        |
|--------------------|-------------|---------------------------|
| acl-pubsub-default | allchannels | 快取上ACL使用者的預設 pubsub 頻道許可。 |

| 名稱                                  | 詳細資訊                               | 描述                                                                                              |
|-------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------|
| client-output-buffer-limit          | normal 0 0 0<br>pubsub 32mb 8mb 60 | 一般用戶端沒有緩衝區限制。PUB如果 /SUB 用戶端違反 32MiB 待處理項目，或違反 8MiB 待處理項目達 60 秒，則會中斷連線。                          |
| client-query-buffer-limit           | 1 GiB                              | 單一用戶端查詢緩衝區的大小上限。此外，用戶端無法發出包含超過 4,000 個引數的請求。                                                    |
| cluster-allow-pubsubshard-when-down | yes                                | 如此即使快取在部分無法運作的狀況下，仍能處理 pubsub 流量。                                                               |
| cluster-allow-reads-when-down       | yes                                | 如此即使快取在部分無法運作的狀況下，仍能處理讀取流量。                                                                     |
| cluster-enabled                     | yes                                | 所有無伺服器快取都已啟用叢集模式，如此就能以公開透明的方式在多個後端碎片之間分割其資料。所有位置都會以單一虛擬節點擁有的形式對用戶端呈現。                           |
| cluster-require-full-coverage       | no                                 | 當金鑰空間部分無法運作時 (也就是至少有一個雜湊位置無法存取)，快取將會繼續針對仍涵蓋的金鑰空間部分接受查詢。整個金鑰空間在 cluster slots 中一律為單一虛擬節點「涵蓋」的狀態。 |

| 名稱                     | 詳細資訊                     | 描述                                                                                                                                                                                                                                                  |
|------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| lua-time-limit         | 5000                     | <p>Lua 指令碼在 之前的最大執行時間，以毫秒為單位 ElastiCache ，採取行動來停止指令碼。</p> <p>lua-time-limit 如果超過 ，所有 Valkey 或 Redis OSS命令可能會傳回格式 ____-BUSY 的錯誤。由於此狀態可能會導致許多基本 Valkey 或 Redis OSS操作的干擾，ElastiCache 會先發出SCRIPTKILL命令。如果失敗，ElastiCache 會強制重新啟動 Valkey 或 Redis OSS。</p> |
| maxclients             | 65000                    | 一次可連線至快取的用戶端數量上限。進一步建立的連線不一定會成功。                                                                                                                                                                                                                    |
| maxmemory-policy       | volatile-lru             | 當達到快取的記憶體限制時，具有TTL集合的項目會在下列 least-recently-used ( LRU ) 估算後被清除。                                                                                                                                                                                     |
| notify-keyspace-events | (空白字串)                   | 無伺服器快取目前不支援金鑰空間事件。                                                                                                                                                                                                                                  |
| port                   | 主要連接埠：6379<br>讀取連接埠：6380 | 無伺服器快取會公告兩個相同主機名稱的連接埠。主要連接埠允許寫入和讀取，而讀取連接埠則允許使用 READONLY 命令的較低延遲的最終一致讀取。                                                                                                                                                                             |
| proto-max-bulk-len     | 512 MiB                  | 單一元素請求的大小上限。                                                                                                                                                                                                                                        |
| timeout                | 0                        | 用戶端不會在特定閒置時間強制中斷連線，但可能會為了進行負載平衡，在穩定狀態期間中斷連線。                                                                                                                                                                                                        |

此外也設有下列限制：

| 名稱        | 詳細資訊   | 描述                                                                                                                    |
|-----------|--------|-----------------------------------------------------------------------------------------------------------------------|
| 金鑰名稱長度    | 4 KiB  | 單一 Valkey 或 Redis OSS 金鑰或頻道名稱的大小上限。若用戶端參考的金鑰大於此值，則會收到錯誤。                                                              |
| Lua 指令碼大小 | 4 MiB  | 單一 Valkey 或 Redis OSS Lua 指令碼的大小上限。若嘗試載入的 Lua 指令碼大於此值，則會收到錯誤。                                                         |
| 位置大小      | 32 GiB | 單一 Valkey 或 Redis OSS 雜湊插槽的大小上限。嘗試在單一 Valkey 或 Redis OSS 插槽上設定超過此值的資料的用戶端將觸發該插槽上的清除政策，而且如果無法清除任何金鑰，將收到記憶體不足 (OOM) 錯誤。 |

## 自行設計的叢集

對於自行設計的叢集，請參閱 [Valkey 和 Redis OSS 參數](#) 當中所述的組態參數預設值，並了解哪些可設定。預設值為一般建議，除非您的使用案例較為特殊，需要覆寫這些值。

## IPv6 Valkey、Redis OSS 和 Memcached 的用戶端範例

ElastiCache 與 Valkey、Redis OSS 和 Memcached 相容。這表示支援 IPv6 連線的用戶端應該能夠連線至 IPv6 已啟用 ElastiCache ( Memcached ) 叢集。與 IPv6 已啟用的資源互動時，有一些值得注意的注意事項。

您可以在 AWS 資料庫部落格上檢視 [Valkey 和 Redis 用戶端部落格文章的最佳實務](#)，以取得有關為 ElastiCache 資源設定 Valkey 和 Redis OSS 用戶端的建議。

以下是使用常用開放原始碼用戶端程式庫與 IPv6 已啟用 ElastiCache 資源互動的最佳實務。

### 使用 Valkey 和 Redis 驗證的用戶端 OSS

ElastiCache 與 Valkey 和開放原始碼 Redis 相容 OSS。這表示支援 IPv6 連線的 Valkey 和開放原始碼 Redis OSS 用戶端應該能夠連線至 IPv6 已啟用的 ElastiCache ( RedisOSS ) 叢集。此外，一些最熱門的 Python 和 Java 用戶端已經過專門測試和驗證，可與所有支援的網路類型組態搭配使用 ( IPv4 僅限、IPv6 僅限 和 Dual Stack )

下列用戶端已特別經過驗證，可搭配 Valkey 和 Redis 的所有支援網路類型組態使用OSS。

已驗證的用戶端：

- [Redis Py \(\) – 4.1.2](#)
- [生菜 – 版本：6.1.6。RELEASE](#)
- [Jedis – 版本：3.6.0](#)

## 用戶端的最佳實務 ( Valkey 和 RedisOSS )

了解常見案例的最佳實務，並遵循一些最熱門開放原始碼 Valkey 和 Redis OSS用戶端程式庫 ( redis-py、PHPRedis、和 Lettuce ) 的程式碼範例，以及使用常用開放原始碼 Memcached 用戶端程式庫與 ElastiCache 資源互動的最佳實務。

主題

- [大量的連線 \( Valkey 和 RedisOSS \)](#)
- [叢集用戶端探索和指數退避 \( Valkey 和 RedisOSS \)](#)
- [設定用戶端逾時 \( Valkey 和 RedisOSS \)](#)
- [設定伺服器端閒置逾時 \( Valkey 和 RedisOSS \)](#)
- [Lua 指令碼](#)
- [儲存大型複合項目 \( Valkey 和 RedisOSS \)](#)
- [Lettuce 用戶端組態 \( Valkey 和 RedisOSS \)](#)
- [為雙堆疊叢集設定偏好的通訊協定 \( Valkey 和 RedisOSS \)](#)

## 大量的連線 ( Valkey 和 RedisOSS )

無伺服器快取和個別 ElastiCache ( RedisOSS ) 節點支援最多 65,000 個並行用戶端連線。然而，為了獲得最佳效能，我們建議用戶端應用程式不要以如此大量的連線持續操作。Valkey 和 Redis OSS 都具有單一執行緒程序，以事件迴圈為基礎，其中傳入的用戶端請求會依序處理。這表示，特定用戶端的回應時間會隨著連線的用戶端數量增加而變長。

您可以採取下列動作，以避免遇到 Valkey 或 Redis OSS 伺服器上的連線瓶頸：

- 從讀取複本執行讀取操作。這可以透過在停用叢集模式下使用 ElastiCache 讀取器端點，或使用複本在啟用叢集模式下讀取，包括無伺服器快取來完成。

- 將寫入流量分散到多個主節點。您可以透過兩種方式進行。您可以使用多碎片 Valkey 或 Redis OSS 叢集搭配具備叢集模式功能的用戶端。您也可以利用用戶端分片寫入多個停用叢集模式的主節點。無伺服器快取中會自動進行此操作。
- 在用戶端程式庫中使用連線集區 (可用時)。

一般而言，與一般 Valkey 或 Redis OSS 命令相比，建立 TCP 連線是一項運算昂貴的操作。例如，在重複使用現有連線時，處理 SET/GET 請求的速度會更快。使用大小有限的用戶端連線集區可減輕連線管理的額外負荷。此外還限制了來自用戶端應用程式的並行傳入連線數。

的下列程式碼範例 PHPRedis 顯示為每個新使用者請求建立新的連線：

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
 //ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

我們在連接至 Graviton2 ( m6g.2xlargeEC2 ) ( ElastiCache Redis ) 節點的 Amazon Elastic Compute Cloud ( Amazon OSS ) 執行個體的迴圈中對此程式碼進行了基準測試。我們將用戶端和伺服器放在相同可用區域內。整個操作的平均延遲為 2.82 毫秒。

當我們更新程式碼並使用持續連線和連線集區時，整個操作的平均延遲為 0.21 毫秒：

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
 // ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

必要的 redis.ini 組態：

- redis.pconnect.pooling\_enabled=1
- redis.pconnect.connection\_limit=10

下列程式碼是 [Redis-py 連線集區](#) 的範例：

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
conn.set(key, value)
```

下列程式碼是 [Lettuce 連線集區](#) 的範例：

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
 ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
 GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
 RedisCommands syncCommands = connection.sync();
 syncCommands.set(key, value);
}
```

## 叢集用戶端探索和指數退避 ( Valkey 和 RedisOSS )

在啟用叢集模式下連線至 ElastiCache Valkey 或 Redis OSS 叢集時，對應的用戶端程式庫必須具備叢集感知能力。用戶端必須取得雜湊位置與叢集中相對應節點的對應，才能將請求傳送至正確的節點，並避免處理叢集重新導向的額外效能負荷。因此，用戶端必須在兩種不同的情況下，探索位置和所對應節點的完整清單：

- 用戶端已初始化，且必須填入初始位置組態
- 從伺服器接收 MOVED 重新導向，例如當複本接管先前主節點提供的所有插槽時，容錯移轉的情況，或當插槽從來源主節點移至目標主節點時，重新分割

用戶端探索通常透過向 Valkey CLUSTERSLOT 或 Redis OSS 伺服器發出 或 CLUSTER NODE 命令來完成。建議您使用此 CLUSTERSLOT 方法，因為它會將一組插槽範圍和相關聯的主要節點和複本節點傳回給用戶端。這不需要從用戶端進行額外剖析，而且較有效率。

根據叢集拓撲，CLUSTERSLOT 命令的回應大小可能會根據叢集大小而有所不同。具有較多節點的較大型叢集會產生較大型的回應。因此，請務必確保執行叢集拓撲探索的用戶端數量不會無限增加。例如，當用戶端應用程式啟動或中斷來自伺服器的連線，而且必須執行叢集探索時，常見的錯誤是，用戶端應用程式發出數個重新連線和探索請求，但未在重試時加上指數退避。這可能會導致 Valkey 或 Redis OSS 伺服器長時間沒有回應，使用 CPU 率為 100%。如果每個 CLUSTERSLOT 命令必須處理叢

集匯流排中的大量節點，則中斷時間會延長。我們在過去觀察到多個用戶端中斷，因為此行為涉及多種不同語言，包括 Python ( redis-py-cluster ) 和 Java ( Lettuce 和 Redisson ) 。

在無伺服器快取中，許多問題會自動緩解，因為公告的叢集拓撲是靜態的，並且由兩個項目組成：寫入端點和讀取端點。使用快取端點時，叢集探索也會自動分散到多個節點上。不過，以下建議仍很實用。

為了減輕突然湧入連線和探索請求所造成的影響，以下是我們的建議做法：

- 實作具有大小限制的用戶端連線集區，以限制來自用戶端應用程式的並行傳入連線數。
- 當用戶端因逾時而中斷與伺服器的連線時，請使用指數退避和抖動進行重試。這樣有助於避免多個用戶端同時癱瘓伺服器。
- 使用位於 [在中尋找連線端點 ElastiCache](#) 的指南尋找叢集端點來執行叢集探索。這樣做就能將探索負載分散到叢集中的所有節點 (最多 90 個)，而不會集中在叢集中少數幾個硬式編碼的種子節點。

以下是 redis-py、PHPRedis 和 Lettuce 中指數退避重試邏輯的一些程式碼範例。

退避邏輯範例 1：redis-py

redis-py 有內建的重試機制，會在失敗後立即重試一次。此機制可以透過建立 [Redis OSS](#) 物件時提供的 `retry_on_timeout` 引數來啟用。我們在這裡示範搭配指數退避和抖動的自動重試機制。我們已提交了提取請求，以在 [redis-py \(#1494\)](#) 中以原生方式實作指數退避。未來可能不需要手動實作。

```
def run_with_backoff(function, retries=5):
 base_backoff = 0.1 # base 100ms backoff
 max_backoff = 10 # sleep for maximum 10 seconds
 tries = 0
 while True:
 try:
 return function()
 except (ConnectionError, TimeoutError):
 if tries >= retries:
 raise
 backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
 print(f"sleeping for {backoff:.2f}s")
 sleep(backoff)
 tries += 1
```

您可以使用下面的程式碼來設定值：

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
```



```
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

根據您的工作負載而定，您可能需要針對延遲敏感的工作負載，將基本退避值從 1 秒變更為數十或數百毫秒。

### 退避邏輯範例 2：PHPRedis

PHPRedis 具有內建重試機制，可重試（不可設定）最多 10 次。您可設定兩次重試之間的延遲（從第二次重試開始使用抖動）。如需詳細資訊，請參閱下列[範例程式碼](#)。我們已提交提取請求，以在[PHPRedis \(#1986\)](#) 中原生實作指數退避，該請求已合併並[記載](#)。對於最新版本的，不需要手動實作 PHPRedis，但我們在此處包含先前版本之的參考。目前，下列程式碼範例會設定重試機制的延遲：

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
 return; // ERROR: connection failed
}
$client->set($key, $value);
```

### 退避邏輯範例 3：Lettuce

Lettuce 採用以[指數退避和抖動](#)文章中所述指數退避策略為基礎的內建重試機制。以下程式碼摘錄顯示完整的抖動方法：

```
public static void main(String[] args)
{
 ClientResources resources = null;
 RedisClient client = null;

 try {
 resources = DefaultClientResources.builder()
 .reconnectDelay(Delay.fullJitter(
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(5), // maximum 5 second delay
 100, TimeUnit.MILLISECONDS) // 100 millisecond base
).build();

 client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
 client.setOptions(ClientOptions.builder()
```

```
.socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
100 millisecond connection timeout
.timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
5 second command timeout
.build());

 // use the connection pool from above example
} finally {
 if (connection != null) {
 connection.close();
 }

 if (client != null){
 client.shutdown();
 }

 if (resources != null){
 resources.shutdown();
 }
}
}
```

## 設定用戶端逾時 ( Valkey 和 RedisOSS )

### 設定用戶端逾時

適當地設定用戶端逾時，讓伺服器有足夠的時間來處理請求並產生回應。此外還能在無法建立伺服器連線時，讓伺服器快速檢錯。某些 Valkey 或 Redis OSS 命令可能比其他命令更昂貴。例如，Lua 指令碼或 MULTI/EXEC 交易包含多個必須以原子方式執行的命令。一般而言，建議設定較長的用戶端逾時，以避免用戶端還未收到來自伺服器的回應就已逾時，包括下列情況：

- 在多個索引鍵之間執行命令
- 執行 MULTI/EXEC 交易或包含多個個別 Valkey 或 Redis OSS 命令的 Lua 指令碼
- 讀取較大的值
- 執行封鎖操作，例如 BLPOP

如果發生封鎖操作，例如 BLPOP，最佳實務是將命令逾時設定為低於通訊端逾時的數字。

以下是在 redis-py、PHPRedis 和 Lettuce 中實作用戶端逾時的程式碼範例。

## 逾時組態範例 1 : redis-py

下列程式碼範例使用 redis-py :

```
connect to Redis server with a 100 millisecond timeout
give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
print(res) # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
 # less than the 2 second socket timeout
print(res)
```

## 逾時組態範例 2 : PHPRedis

以下是使用的程式碼範例PHPRedis :

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
 return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n"; // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n"; // less than the 2 second socket timeout
```

## 逾時組態範例 3 : Lettuce

下列程式碼範例使用 Lettuce :

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
```

```
{
 RedisClient client = null;
 StatefulRedisConnection<String, String> connection = null;
 try {
 client = RedisClient.create(RedisURI.create(HOST, PORT));
 client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
 2 second command timeout
 .build());

 // use the connection pool from above example

 commands.set("key", "value"); // will timeout after 2 seconds
 commands.blpop(1, "list"); // BLPPOP with 1 second timeout
 } finally {
 if (connection != null) {
 connection.close();
 }

 if (client != null){
 client.shutdown();
 }
 }
}
```

## 設定伺服器端閒置逾時 ( Valkey 和 RedisOSS )

我們發現了下列情況：客戶的應用程式有大量連線的閒置用戶端，但未主動傳送命令。在這類情況下，大量閒置的用戶端可能造成您耗盡全部 65,000 個連線。為避免這類情況發生，請在伺服器上透過 [Valkey 和 Redis OSS 參數](#) 設定適當的逾時設定。這樣做可確保伺服器主動中斷閒置用戶端的連線，以避免連線數目增加。無伺服器快取不適用此設定。

### Lua 指令碼

Valkey 和 Redis OSS 支援超過 200 個命令，包括執行 Lua 指令碼的命令。不過，在 Lua 指令碼方面，有幾個陷阱可能會影響 Valkey 或 Redis 的記憶體和可用性 OSS。

#### 未參數化的 Lua 指令碼

每個 Lua 指令碼都會在執行之前快取在 Valkey 或 Redis OSS 伺服器上。無參數 Lua 指令碼是唯一的，這可能會導致 Valkey 或 Redis OSS 伺服器儲存大量 Lua 指令碼並耗用更多記憶體。若要緩解這

種情況，請確保所有 Lua 指令碼都已參數化，並定期執行SCRIPTFLUSH以視需要清除快取的 Lua 指令碼。

以下範例說明如何使用參數化指令碼。首先將示範非參數化的方法來產生三個不同的已快取 Lua 指令碼，這是不建議的方法：

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

建議改用下列模式來建立可接受所傳遞參數的單一指令碼：

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

### 長時間執行的 Lua 指令碼

Lua 指令碼可以以原子方式執行多個命令，因此比一般 Valkey 或 Redis OSS命令需要更長的時間才能完成。如果 Lua 指令碼只執行唯獨操作，就可以讓它中途停止執行。然而，一旦 Lua 指令碼執行寫入操作，就無法讓它終止，而必須執行到完成為止。長時間執行且正在靜音的 Lua 指令碼可能會導致 Valkey 或 Redis OSS 伺服器長時間沒有回應。為了解決此問題，請避免長時間執行的 Lua 指令碼，並且在生產前環境中測試指令碼。

### 具有隱匿寫入的 Lua 指令碼

Lua 指令碼有幾種方式可以繼續將新資料寫入 Valkey 或 Redis，OSS即使 Valkey 或 Redis OSS 超過 maxmemory：

- 指令碼會在 Valkey 或 Redis OSS 伺服器低於 時啟動maxmemory，並在 內包含多個寫入操作
- 指令碼的第一個寫入命令不會耗用記憶體（例如 DEL），其次是消耗記憶體的更多寫入操作
- 您可以在 以外的 Valkey 或 Redis OSS 伺服器上設定適當的驅逐政策，以緩解此問題noeviction。這可讓 Redis OSS 在 Lua 指令碼之間逸出項目並釋放記憶體。

### 儲存大型複合項目（Valkey 和 RedisOSS）

在某些情況下，應用程式可能會在 Valkey 或 Redis 中存放大型複合項目 OSS（例如多 GB 雜湊資料集）。這不是建議的做法，因為它通常會導致 Valkey 或 Redis 中的效能問題OSS。例如，用戶端可以執行HGETALL命令來擷取整個多 GB 雜湊集合。這可能會對 Valkey 或 Redis OSS 伺服器產生重大記

記憶體壓力，以緩衝用戶端輸出緩衝區中的大型項目。此外，對於叢集模式下的插槽遷移，ElastiCache 不會遷移包含序列化大小大於 256 MB 之項目的插槽。

為了解決大型項目的問題，以下是我們的建議做法：

- 將大型複合項目分割成多個較小的項目。例如，將大型雜湊集合分割成個別的索引鍵-值欄位，且索引鍵名稱結構描述適當反映集合，例如，在索引鍵名稱中使用通用首碼來識別項目集合。如果您必須以原子方式存取相同集合中的多個欄位，您可以使用 MGET 命令來擷取相同命令中的多個鍵值。
- 如果您評估了所有選項，但仍無法分割大型集合資料集，請嘗試使用對集合中的資料子集而非整個集合執行操作的命令。避免採用要求您在同一命令中以原子方式擷取整個多 GB 集合的使用案例。其中一個範例是使用 HGET 或 HMGET 命令 HGETALL，而不是雜湊集合。

## Lettuce 用戶端組態 ( Valkey 和 RedisOSS )

本節說明建議的 Java 和 Lettuce 組態選項，以及這些選項如何套用至 ElastiCache 叢集。

本節中的建議已使用 Lettuce 版本 6.2.2 進行測試。

### 主題

- [範例：TLS已啟用叢集模式的 Lettuce 組態](#)
- [範例：停用、TLS啟用叢集模式的 Lettuce 組態](#)

## Java DNS快取 TTL

Java 虛擬機器 ( JVM ) 會快取DNS名稱查詢。當將主機名稱JVM解析為 IP 地址時，它會快取 IP 地址一段時間，稱為 time-to-live ( TTL )。

TTL 值的選擇是在延遲和對變更的回應能力之間取得權衡。透過較短的 TTLs，DNS解決者會DNS更快地通知叢集中的更新。這可讓您的應用程式更快回應叢集所經歷的替換或其他工作流程。不過，如果 TTL 太低，則會增加查詢磁碟區，進而增加應用程式的延遲。雖然沒有正確的TTL值，但建議您考慮在設定TTL值時，等待變更生效的時間長度。

由於 ElastiCache 節點使用可能會變更DNS的名稱項目，建議您JVM將設定為 5 到 10 秒TTL的低點。這可確保當節點的 IP 地址變更時，您的應用程式將能夠透過重新查詢DNS項目來接收和使用資源的新 IP 地址。

在某些 Java 組態上，TTL會設定JVM預設值，因此在JVM重新啟動之前，永遠不會重新整理DNS項目。

如需如何設定 JVM 的詳細資訊TTL，請參閱[如何設定 JVM TTL](#)。

## Lettuce 版本

建議使用 Lettuce 6.2.2 或更新版本。

## 端點

當您使用已啟用叢集模式的叢集時，請將 `redisUri` 設為叢集組態端點。此DNS查詢會URI傳回叢集中所有可用節點的清單，並在叢集初始化期間隨機解析為其中一個節點。如需拓撲重新整理運作方式的詳細資訊，請參閱本主題[dynamicRefreshResources](#)稍後的。

## SocketOption

啟用 [KeepAlive](#)。啟用此選項可減少在命令執行期間處理失敗連線的需求。

請務必根據應用程式需求和工作負載設定[連線逾時](#)。如需詳細資訊，請參閱本主題稍後的 [Timeouts \(逾時\)](#) 章節。

`ClusterClientOption`：啟用叢集模式的用戶端選項

連線中斷[AutoReconnect](#)時啟用。

設定 [CommandTimeout](#)。如需更多詳細資料，請參閱本主題稍後的 [Timeouts \(逾時\)](#) 章節。

[nodeFilter](#) 設定為從拓撲篩選失敗的節點。Lettuce 會儲存用戶端「分割區」（也稱為碎片）中「叢集節點」輸出中發現的所有節點（包括具有 PFAIL/FAIL 狀態的節點）。在建立叢集拓撲的過程中，它會嘗試連線到所有的分割區節點。當節點因任何原因被取代時，新增故障節點的這種 Lettuce 行為可能會導致連線錯誤（或警告）。

例如，在容錯移轉完成且叢集啟動復原程序後，當 `clusterTopology` 重新整理時，叢集匯流排節點映射會在短時間內將下節點列為FAIL節點，然後再從拓撲中完全移除。在此期間，Lettuce 用戶端會將其視為運作良好的節點，並持續與其連線。這會在重試耗盡後導致失敗。

例如：

```
final ClusterClientOptions clusterClientOptions =
 ClusterClientOptions.builder()
 ... // other options
 .nodeFilter(it ->
 ! (it.is(RedisClusterNode.NodeFlag.FAIL)
 || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
 || it.is(RedisClusterNode.NodeFlag.HANDSHAKE))
```



```
 || it.is(RedisClusterNode.NodeFlag.NOADDR)))
 .validateClusterNodeMembership(false)
 .build();
redisClusterClient.setOptions(clusterClientOptions);
```

### Note

節點篩選最好搭配 `DynamicRefreshSources` 設定為 `true` 使用。否則，如果拓撲檢視是從單一有問題的種子節點中取得，並將某些碎片的主節點視為故障，則它將篩除此主節點，這將導致槽不被涵蓋。擁有多個種子節點（當 `DynamicRefreshSources` 為 `true`）可降低此問題的可能性，因為至少一些種子節點在新升級的主要節點容錯移轉之後，應該具有更新的拓撲檢視。

`ClusterTopologyRefreshOptions`：用於控制叢集拓撲的選項，以重新整理已啟用叢集模式的用戶端

### Note

已停用叢集模式的叢集不支援叢集探索命令，也不相容於所有用戶端動態拓撲探索功能。與停用的叢集模式與 Lettuce 的 ElastiCache 不相容 `MasterSlaveTopologyRefresh`。相反地，對於停用的叢集模式，您可以設定 `StaticMasterReplicaTopologyProvider` 並提供叢集讀取和寫入端點。如需連接至已停用叢集模式之叢集的詳細資訊，請參閱 [尋找 Valkey 或 Redis OSS \( 停用叢集模式 \) 叢集的端點 \( 主控台 \)](#)。如果您想要使用 Lettuce 的動態拓撲探索功能，可以使用與現有叢集相同的碎片組態來建立已啟用叢集模式的叢集。不過，對於已啟用叢集模式的叢集，建議至少設定 3 個具有至少 1 個複本的碎片，以支援快速容錯移轉。

啟用 [enablePeriodicRefresh](#)。這會啟用定期叢集拓撲更新，以便用戶端以的間隔更新叢集拓撲 `refreshPeriod`（預設：60 秒）。停用時，用戶端只有在嘗試對叢集執行命令發生錯誤時，叢集拓撲才會更新。

啟用此選項後，您可以將此工作新增至背景任務，以減少與重新整理叢集拓撲相關的延遲。雖然拓撲重新整理是在背景工作中執行，但對於具有許多節點的叢集而言，可能會有些慢。這是因為所有節點都在查詢其檢視以取得最新的叢集檢視。如果您執行的是大型叢集，則可能需要增加期間。

啟用 [enableAllAdaptiveRefreshTriggers](#)。這會啟用適應性拓撲重新整理，其使用所有 [觸發條件](#)：`MOVED_REDIRECT`、`ASK_REDIRECT`、`PERSISTENT_RECONNECTS`、`UNCOVERED_SLOT`、`UNK`。自適應重新整理觸發程序會根據 Valkey 或 Redis OSS 叢集操作期間發生的事件啟動拓撲檢視更新。啟



用此選項會在上述觸發程序發生時立即重新整理拓撲。適應性觸發程序重新整理會使用逾時來限制速率，因為事件可能會大規模發生 (更新之間的預設逾時：30)。

啟用 [closeStaleConnections](#)。如此可在重新整理叢集拓撲時關閉過時的連線。只有在 [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled \( \)](#) 為 true 時，才會生效。啟用後，用戶端可以關閉過時的連線，並在背景建立新連線。這可減少在命令執行期間處理失敗連線的需求。

啟用 [dynamicRefreshResources](#)。我們建議為小型叢集啟用 `dynamicRefreshResources`，並為大型叢集停用它。 `dynamicRefreshResources` 啟用從提供的種子節點探索叢集節點 (例如，叢集組態端點)。它會使用所有探索到的節點做為重新整理叢集拓撲的來源。

使用動態重新整理查詢所有探索到的叢集拓撲節點，並嘗試選擇最準確的叢集檢視。如果其設定為 false，則只會使用初始種子節點做為拓撲探索的來源，而且只會取得初始種子節點的用戶端數量。在停用時，如果叢集配置端點解析為故障的節點，則嘗試重新整理叢集檢視會失敗，並導致例外狀況。這種情況可能會發生，因為從叢集組態端點移除故障節點的項目需要一些時間。因此，組態端點仍可在短時間內隨機解析為故障的節點。

但是，在其啟用後，我們會使用從叢集檢視接收到的所有叢集節點，來查詢其目前的檢視。因為我們會從該檢視中篩選出故障的節點，所以拓撲重新整理將會成功。但是，如果 `dynamicRefreshSources` 為 true，Lettuce 會查詢所有節點以取得叢集檢視，然後比較結果。因此，對於具有大量節點的叢集來說，它可能很昂貴。建議您在多節點的叢集時關閉此功能。

```
final ClusterTopologyRefreshOptions topologyOptions =
 ClusterTopologyRefreshOptions.builder()
 .enableAllAdaptiveRefreshTriggers()
 .enablePeriodicRefresh()
 .dynamicRefreshSources(true)
 .build();
```

## ClientResources

[DnsResolver](#) 使用 設定 [DirContextDnsResolver](#)。DNS 解析程式是以 Java 的 `com.sun.jndi.dns` 為基礎 `DnsContextFactory`。

[reconnectDelay](#) 使用指數退避和全抖動來設定。Lettuce 具有以指數退避策略為基礎的內建重試機制。如需詳細資訊，請參閱 AWS 架構部落格上的 [指數退避和抖動](#)。如需重試退避策略重要性的詳細資訊，請參閱 AWS 資料庫部落格上 [最佳實務部落格文章](#) 的退避邏輯區段。

```
ClientResources clientResources = DefaultClientResources.builder()
 .dnsResolver(new DirContextDnsResolver())
 .reconnectDelay(
```

```

 Delay.fullJitter(
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(10), // maximum 10 second delay
 100, TimeUnit.MILLISECONDS) // 100 millisecond base
 .build();

```

## 逾時

使用低於指令逾時的連線逾時值。Lettuce 使用延遲連接建立。因此，如果連線逾時高於指令逾時，而 Lettuce 嘗試連線到健康狀態不良的節點，且永遠超過指令逾時，則您可能會在拓撲重新整理後發生一段時間的持續性失敗。

針對不同的指令使用動態指令逾時。我們建議您根據指令預期持續時間設定指令逾時。例如，針對透過多個金鑰反覆運算的命令使用較長的逾時，例如 FLUSHDB、KEYS、SMEMBERS 或 Lua FLUSHALL 指令碼。針對單一金鑰命令使用較短的逾時，例如 GET、SET 和 HSET。

### Note

下列範例中設定的逾時，適用於執行 SET/GET 命令的測試，其金鑰和值的長度上限為 20 位元組。在指令很複雜或金鑰和值較大時，處理時間可能會更長。您應該根據應用程式的使用案例設定逾時。

```

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

```

```

SocketOptions socketOptions = SocketOptions.builder()
 .connectTimeout(CONNECT_TIMEOUT)
 .build();

```

```

class DynamicClusterTimeout extends TimeoutSource {
 private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
 .add(CommandType.FLUSHDB)
 .add(CommandType.FLUSHALL)
 .add(CommandType.CLUSTER)
 .add(CommandType.INFO)
 .add(CommandType.KEYS)
 .build();

```

```

private final Duration defaultCommandTimeout;
private final Duration metaCommandTimeout;

DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
{
 defaultCommandTimeout = defaultTimeout;
 metaCommandTimeout = metaTimeout;
}

@Override
public long getTimeout(RedisCommand<?, ?, ?> command) {
 if (META_COMMAND_TYPES.contains(command.getType())) {
 return metaCommandTimeout.toMillis();
 }
 return defaultCommandTimeout.toMillis();
}
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
 .timeoutSource(
 new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
 .build();

```

### 範例：TLS已啟用叢集模式的 Lettuce 組態

#### Note

下列範例中的逾時適用於執行 SET/GET 命令的測試，其金鑰和值長度最多為 20 個位元組。在指令很複雜或金鑰和值較大時，處理時間可能會更長。您應該根據應用程式的使用案例設定逾時。

```

// Set DNS cache TTL
public void setJVMProperties() {
 java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);

```

```
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
 cluster configuration endpoint
final RedisURI redisUriCluster =
 RedisURI.Builder.redis(clusterConfigurationEndpoint)
 .withPort(6379)
 .withSsl(true)
 .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
 .reconnectDelay(
 Delay.fullJitter(
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(10), // maximum 10 second delay
 100, TimeUnit.MILLISECONDS)) // 100 millisecond base
 .dnsResolver(new DirContextDnsResolver())
 .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
 RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
 private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
 .add(CommandType.FLUSHDB)
 .add(CommandType.FLUSHALL)
 .add(CommandType.CLUSTER)
 .add(CommandType.INFO)
 .add(CommandType.KEYS)
 .build();

 private final Duration metaCommandTimeout;
 private final Duration defaultCommandTimeout;

 DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
 {
 defaultCommandTimeout = defaultTimeout;
 }
}
```

```
 metaCommandTimeout = metaTimeout;
 }

 @Override
 public long getTimeout(RedisCommand<?, ?, ?> command) {
 if (META_COMMAND_TYPES.contains(command.getType())) {
 return metaCommandTimeout.toMillis();
 }
 return defaultCommandTimeout.toMillis();
 }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
 .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
META_COMMAND_TIMEOUT))
 .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
 ClusterTopologyRefreshOptions.builder()
 .enableAllAdaptiveRefreshTriggers()
 .enablePeriodicRefresh()
 .dynamicRefreshSources(true)
 .build();

// Configure the socket options
final SocketOptions socketOptions =
 SocketOptions.builder()
 .connectTimeout(CONNECT_TIMEOUT)
 .keepAlive(true)
 .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
 ClusterClientOptions.builder()
 .topologyRefreshOptions(topologyOptions)
 .socketOptions(socketOptions)
 .autoReconnect(true)
 .timeoutOptions(timeoutOptions)
 .nodeFilter(it ->
 ! (it.is(RedisClusterNode.NodeFlag.FAIL)
 || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
 || it.is(RedisClusterNode.NodeFlag.NOADDR)))
 .validateClusterNodeMembership(false)
```

```

 .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
 redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();

```

範例：停用、TLS啟用叢集模式的 Lettuce 組態

### Note

下列範例中的逾時適用於執行 SET/GET 命令的測試，其金鑰和值長度最多為 20 個位元組。在指令很複雜或金鑰和值較大時，處理時間可能會更長。您應該根據應用程式的使用案例設定逾時。

```

// Set DNS cache TTL
public void setJVMPProperties() {
 java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

 RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
 DefaultClientResources.builder()
 .dnsResolver(new DirContextDnsResolver())
 .reconnectDelay(
 Delay.fullJitter(

```

```
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(10), // maximum 10 second delay
 100,
 TimeUnit.MILLISECONDS)) // 100 millisecond base
 .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
 private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
 .add(CommandType.FLUSHDB)
 .add(CommandType.FLUSHALL)
 .add(CommandType.INFO)
 .add(CommandType.KEYS)
 .build();

 private final Duration metaCommandTimeout;
 private final Duration defaultCommandTimeout;

 DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
 {
 defaultCommandTimeout = defaultTimeout;
 metaCommandTimeout = metaTimeout;
 }

 @Override
 public long getTimeout(RedisCommand<?, ?, ?> command) {
 if (META_COMMAND_TYPES.contains(command.getType())) {
 return metaCommandTimeout.toMillis();
 }
 return defaultCommandTimeout.toMillis();
 }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
 .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
 .build();

final SocketOptions socketOptions =
 SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =
```

```
ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);
```

## 為雙堆疊叢集設定偏好的通訊協定 ( Valkey 和 RedisOSS )

對於啟用叢集模式的 Valkey 或 Redis OSS叢集，您可以控制通訊協定用戶端將使用 IP Discovery 參數連線到叢集中的節點。IP Discovery 參數可以設定為 IPv4或 IPv6。

對於 Valkey 或 Redis OSS叢集，IP 探索參數會設定[叢集插槽 \( \)](#)、[叢集碎片 \( \)](#) 和[叢集節點 \( \)](#) 輸出中使用的 IP 通訊協定。用戶端會使用這些命令來探索叢集拓撲。用戶端使用這些命令IPs中的 來連線到叢集中的其他節點。

變更 IP 探索不會導致連線用戶端停機。但是，該變更需要一些傳播時間。若要判斷變更何時完全傳播至 Valkey 或 Redis OSS叢集，請監控 的輸出cluster slots。一旦叢集插槽命令報告傳回的所有節點IPs使用新通訊協定，變更即完成傳播。

Redis-Py 範例：

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
nodes):
 nodes = set()

 # This refreshes the cluster topology and will discovery any node updates.
 # Under the hood it calls cluster slots
 cluster.nodes_manager.initialize()
 for node in cluster.get_nodes():
 nodes.add(node.host)
 self.logger.info(nodes)

 time.sleep(1)
```

Lettuce 範例：

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
6379));
```



```
Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
 // Check for any changes in the cluster topology.
 // Under the hood this calls cluster slots
 clusterClient.refreshPartitions();
 Set<String> nodes = new HashSet<>();

 for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
 nodes.add(node.getUri().getHost());
 }

 Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
 try {
 return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
 } catch (UnknownHostException ignored) {}
 return false;
}));
```

## 用戶端的最佳實務 ( Memcached )

### 設定您的 ElastiCache 用戶端以進行有效的負載平衡 ( Memcached )

#### Note

本節適用於自行設計的多節點 Memcached 叢集。

若要有效使用多個 ElastiCache Memcached 節點，您需要能夠將快取金鑰分散到節點。其中一種針對擁有  $n$  個節點叢集進行負載平衡的簡易方式，便是計算物件索引鍵的雜湊，並以  $n$  對結果進行模除以取得餘數 -  $\text{hash}(\text{key}) \bmod n$ 。產生的值 (0 到  $n-1$ ) 便是您置放物件的節點數。

只要節點數 ( $n$ ) 為常數，此方法便相當簡易且運作良好。但是，每當您新增節點到叢集，或從叢集移除節點時，需要移動的鍵數便是  $(n - 1) / n$  (其中  $n$  是新的節點數)。因此，此方法會導致移動大量的鍵，引發大量的初始快取遺漏，尤其是在節點數越來越大時。從 1 個擴展到 2 個節點會導致  $(2-1)/2$  (50%) 的索引鍵移動，這是最佳情況。從 9 個擴展到 10 個節點會導致  $(10-1)/10$  (90%) 的索引鍵移動。若您

因為流量的峰值而向上擴展，您不會希望發生大量的快取遺漏。大量的快取遺漏會導致資料庫遭受衝擊，尤其是藉時資料庫已經因為流量峰值而超載了。

此難題的解決方案便是一致性雜湊。一致雜湊所使用的演算法，會讓節點新增到叢集或從叢集移除節點時需移動的索引鍵數約為  $1/n$  (其中  $n$  是新節點的數量)。從 1 擴展到 2 個節點會導致  $1/2$  (50%) 的鍵移動，此為最壞情況。從 9 擴展到 10 個節點會導致  $1/10$  (10%) 的鍵移動。

做為使用者，您可以控制要用於多節點叢集的雜湊演算法。建議您設定用戶端，以使用一致性雜湊。所幸，在大多數的熱門語言中，有許多 Memcached 用戶端程式庫都實作了一致性雜湊。檢查您使用的程式庫文件，查看其是否支援一致性雜湊以及其實作方式。

如果您在 Java、PHP 或 .NET 中工作，我們建議您使用其中一個 Amazon ElastiCache 用戶端程式庫。

### 使用 Java 的一致性雜湊

ElastiCache Memcached Java 用戶端是以開放原始碼 spymemcached Java 用戶端為基礎，該用戶端內建一致的雜湊功能。程式庫包含實作一致雜湊的 `KetamaConnectionFactory` 類別。根據預設，spymemcached 中會關閉一致性雜湊。

如需詳細資訊，請參閱位於 `KetamaConnectionFactory` 的文件 [KetamaConnectionFactory](#)。

### PHP 搭配 Memcached 使用時一致的雜湊

ElastiCache Memcached PHP 用戶端是圍繞內建 Memcached PHP 程式庫的包裝函式。根據預設，Memcached PHP 程式庫會關閉一致的雜湊。

請使用以下程式碼來開啟一致性雜湊。

```
$m = new Memcached();
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

除了上述程式碼之外，我們建議也在您的 `php.ini` 檔案中開啟 `memcached.sess_consistent_hash`。

如需詳細資訊，請參閱 PHP <https://http://php.net/manual/en/memcached.configuration.php> Memcached 的執行期組態文件。請特別注意 `memcached.sess_consistent_hash` 參數。

### 使用與 .NET Memcached 使用的一致雜湊

ElastiCache Memcached .NET 用戶端是圍繞 Enyim Memcached 的包裝。根據預設，Enyim Memcached 用戶端會開啟一致性雜湊。

如需詳細資訊，請參閱位於 <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator> memcached/locator 的文件。

## 使用 Memcached 驗證的用戶端

下列用戶端已特別經過驗證，可搭配 Memcached 的所有支援網路類型組態使用。

已驗證的用戶端：

- 適用於 [AWS ElastiCache Php 的 Cluster Client Memcached](#) – [\\*3.6.2 版](#)
- 適用於 [AWS ElastiCache Java 的 Cluster Client Memcached](#) – Github 上的最新主控端

## 為雙堆疊叢集設定偏好的通訊協定 ( Memcached )

針對 Memcached 叢集，您可以使用 IP 探索參數，控制用戶端用來連線至叢集中節點的通訊協定。IP Discovery 參數可以設定為 IPv4 或 IPv6。

IP 探索參數可控制設定取得叢集輸出中使用的 IP 通訊協定。而這將決定支援 ElastiCache ( Memcached ) 叢集自動探索的用戶端所使用的 IP 通訊協定。

變更 IP 探索不會導致連線用戶端停機。但是，該變更需要一些傳播時間。

監控 Java 的 `getAvailableNodeEndpoints` 輸出，針對 Php 則監控 `getServerList` 的輸出。一旦這些函數的輸出 IPs 回報叢集中所有使用更新通訊協定的節點，變更即完成傳播。

Java 範例：

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
 nodes =
 client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toSet());

 Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
 try {
```

```
 return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
 } catch (UnknownHostException ignored) {}
 return false;
}));
```

Php 範例 :

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
 # The PHP memcached client only updates the server list if the polling interval has
 expired and a
 # command is sent
 $client->get('test');

 $nodes = $client->getServerList();

 sleep(1);
 $target_ips_count = 0;

 // For IPv4 use FILTER_FLAG_IPV4
 $target_ips_count = count(array_filter($nodes, function($node) { return
 filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));
} while (count($nodes) !== $target_ips_count);
```

在更新 IP 探索之前建立的任何現有用戶端連線仍會使用舊通訊協定進行連線。一旦在叢集探索命令的輸出中偵測到變更，所有經過驗證的用戶端都會使用新的 IP 通訊協定自動重新連線至叢集。但是，這仍取決於用戶端的操作。

## TLS 已啟用雙堆疊 ElastiCache 叢集

ElastiCache 叢集啟用 TLS 時，叢集探索函數 ( cluster slots、cluster shards 和 cluster nodes 代表 Redis ) 或 config get cluster Memcached 傳回主機名稱，而非 IPs。然後，會使用主機名稱來取代 IPs 連接至 ElastiCache 叢集，並執行交 TLS 握。這表示用戶端不會受到 IP 探索參數的影響。對於 TLS 已啟用的叢集，IP Discovery 參數不會影響偏好的 IP 通訊協定。相反地，使用的 IP 通訊協定將取決於用戶端在解析 DNS 主機名稱時偏好的 IP 通訊協定。

## Java 用戶端

從同時支援 IPv4 和 的 Java 環境連線時 IPv6，Java 預設會偏好 IPv4 而不是向 IPv6 後相容。不過，IP 通訊協定偏好設定可透過 JVM 引數設定。若要偏好 IPv4，JVM 接受 `-Djava.net.preferIPv4Stack=true` 和 偏好 IPv6 設定 `-Djava.net.preferIPv6Stack=true`。設定 `-Djava.net.preferIPv4Stack=true` 表示 JVM 將不再進行任何 IPv6 連線。對於 Valkey 或 Redis OSS，這包括對其他非 Valkey 和非 Redis OSS 應用程式的應用程式。

## 主機層級偏好

一般而言，如果用戶端或用戶端執行階段不提供設定 IP 通訊協定偏好設定的組態選項，則在執行 DNS 解析時，IP 通訊協定將取決於主機的組態。根據預設，大多數主機偏好 IPv6 偏好 IPv4，但此偏好可在主機層級設定。這將影響來自該主機的所有 DNS 請求，而不僅僅是對 ElastiCache 叢集的請求。

## Linux 主機

針對 Linux，可以透過修改 `gai.conf` 檔案，設定 IP 通訊協定偏好設定。您可以在 `/etc/gai.conf` 下找到 `gai.conf` 檔案。若沒有指定的 `gai.conf`，則應在 `/usr/share/doc/glibc-common-x.xx/gai.conf` 之下提供一個範例，可以複製到 `/etc/gai.conf`，且預設組態應取消註解。若要更新連線到 ElastiCache 叢集 IPv4 時偏好的組態，請將包含叢集 CIDR 的範圍優先順序更新 IPs 為高於預設 IPv6 連線的優先順序。預設 IPv6 連線的優先順序為 40。例如，假設叢集位於具有 `172.31.0.0 : 0/16` CIDR 的子網路中，下面的組態會導致用戶端偏好與該叢集的 IPv4 連線。

```
label ::1/128 0
label ::/0 1
label 2002::/16 2
label ::/96 3
label ::ffff:0:0/96 4
label fec0::/10 5
label fc00::/7 6
label 2001:0::/32 7
label ::ffff:172.31.0.0/112 8
#
This default differs from the tables given in RFC 3484 by handling
(now obsolete) site-local IPv6 addresses and Unique Local Addresses.
The reason for this difference is that these addresses are never
NATed while IPv4 site-local addresses most probably are. Given
the precedence of IPv6 over IPv4 (see below) on machines having only
site-local IPv4 and IPv6 addresses a lookup for a global address would
see the IPv6 be preferred. The result is a long delay because the
site-local IPv6 addresses cannot be used while the IPv4 address is
```

```
(at least for the foreseeable future) NATed. We also treat Teredo
tunnels special.
#
precedence <mask> <value>
Add another rule to the RFC 3484 precedence table. See section 2.1
and 10.3 in RFC 3484. The default is:
#
precedence ::1/128 50
precedence ::/0 40
precedence 2002::/16 30
precedence ::/96 20
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100
```

若需更多 `gai.conf` 的相關資訊，請前往 [Linux 主頁](#)

## Windows 主機

Windows 主機的程式十分類似。針對 Windows 主機，您可以執行 `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`。此操作與在 Linux 主機上修改 `gai.conf` 檔案的效果相同。

這將更新偏好設定政策，以偏好在指定CIDR範圍內透過IPv4連線IPv6進行連線。例如，假設叢集位於執行 `172.31.0.0 : 0/16` 的子網路中，執行 `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15`，則會導致下列優先順序表，這會導致用戶端在連線至叢集IPv4時更喜歡。

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...

Precedence Label Prefix

100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```



## 管理 Valkey 和 Redis 的預留記憶體 OSS

預留記憶體是為非資料使用預留的記憶體。執行備份或容錯移轉時，Valkey 和 Redis OSS會使用可用的記憶體，在叢集的資料寫入 .rdb 檔案時記錄對叢集的寫入操作。如果您沒有足夠的記憶體可供所有寫入使用，此程序會失敗。接下來，您可以找到管理 ElastiCache（Redis OSS）預留記憶體的選項，以及如何套用這些選項的相關資訊。

### 主題

- [您需要多少預留記憶體？](#)
- [用於管理預留記憶體的參數](#)
- [指定您的預留記憶體管理參數](#)

### 您需要多少預留記憶體？

如果您在 2.8.22 OSS之前執行 Redis 版本，則保留比執行 Redis 2.8.22 OSS 或更新版本更多的備份和容錯移轉記憶體。此要求是由於 ElastiCache（Redis OSS）實作備份程序的不同方式所致。經驗法則是為 2.8.22 之前的版本保留節點類型maxmemory值的一半OSS，為 Redis 2.8.22版及更新OSS版本保留四分之一。

由於 ElastiCache 實作備份和複寫程序的方式不同，因此經驗法則是使用 reserved-memory-percent 參數保留節點類型maxmemory值的 25%。這是預設值，建議大多數情況下使用。

當爆量微執行個體和小型執行個體類型在接近maxmemory限制的情況下運作時，它們可能會遇到交換用量。為了在備份、複寫和高流量期間改善這些執行個體類型的操作可靠性，我們建議將reserved-memory-percent參數的值在小型執行個體類型上增加到 30%，在微型執行個體類型上增加到 50%。

對於具有資料分層的 ElastiCache 叢集上的寫入密集型工作負載，我們建議將增加reserved-memory-percent到節點可用記憶體的 50%。

如需詳細資訊，請參閱下列內容：

- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)
- [同步與備份的實作方式](#)
- [中的資料分層 ElastiCache](#)



## 用於管理預留記憶體的參數

截至 2017 年 3 月 16 日，Amazon ElastiCache 提供兩個互斥參數來管理您的 Valkey 或 Redis OSS 記憶體，`reserved-memory` 以及 `reserved-memory-percent`。這些參數都不是 Valkey 或 Redis OSS 分佈的一部分。

視您成為 ElastiCache 客戶的時間而定，其中一個參數是預設記憶體管理參數。當您建立新的 Valkey 或 Redis OSS 叢集或複寫群組並使用預設參數群組時，此參數適用。

- 對於在 2017 年 3 月 16 日之前啟動的客戶 – 當您使用預設參數群組建立 Redis OSS 叢集或複寫群組時，您的記憶體管理參數是 `reserved-memory`。這個情況下會保留零 (0) 位元組的記憶體。
- 對於在 2017 年 3 月 16 日或之後啟動的客戶 – 當您使用預設參數群組建立 Valkey 或 Redis OSS 叢集或複寫群組時，您的記憶體管理參數為 `reserved-memory-percent`。在這種情況下，會保留節點 `maxmemory` 值的 25% 用於非資料目的。

讀取兩個 Valkey 或 Redis OSS 記憶體管理參數後，您可能偏好使用非預設或非預設值的參數。若是如此，您可以變更為另一個預留記憶體管理參數。

若要變更該參數的值，您可以建立自訂參數群組並進行修改，以使用您偏好的記憶體管理參數和值。然後，每當您建立新的 Valkey 或 Redis OSS 叢集或複寫群組時，即可使用自訂參數群組。若是現有的叢集或複寫群組，您可以進行修改來使用您的自訂參數群組。

如需詳細資訊，請參閱下列內容：

- [指定您的預留記憶體管理參數](#)
- [建立 ElastiCache 參數群組](#)
- [修改 ElastiCache 參數群組](#)
- [修改 ElastiCache 叢集](#)
- [修改複寫群組](#)

### `reserved-memory` 參數

在 2017 年 3 月 16 日之前，所有 ElastiCache ( Redis OSS ) 預留記憶體管理都是使用參數完成 `reserved-memory`。`reserved-memory` 的預設值為 0。此預設值不會保留 Valkey 或 Redis OSS 額外負荷的記憶體 OSS，並允許 Valkey 或 Redis 使用資料來耗用節點的所有記憶體。

您必須建立自訂參數群組，才能變更 reserved-memory，以便有足夠的記憶體可供備份和容錯移轉使用。在此自訂參數群組中，您可以針對叢集和叢集節點類型上執行的 Valkey 或 Redis OSS 版本 reserved-memory，將設定為適當的值。如需詳細資訊，請參閱 [您需要多少預留記憶體？](#)

參數 reserved-memory 為特定 ElastiCache 參數，且不屬於一般 Redis OSS 分佈。

下列程序說明如何使用 reserved-memory 來管理 Valkey 或 Redis OSS 叢集上的記憶體。

使用 reserved-memory 來保留記憶體

1. 建立自訂參數群組，指定與您所執行引擎版本相符的參數群組系列，例如指定 redis2.8 參數群組系列。如需詳細資訊，請參閱 [建立 ElastiCache 參數群組](#)。

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name redis6x-m3xl \
 --description "Redis OSS 2.8.x for m3.xlarge node type" \
 --cache-parameter-group-family redis6.x
```

2. 計算要保留給 Valkey 或 Redis OSS 額外負荷的記憶體位元組數。您可以在 [Redis OSS 節點類型特定參數](#) 中，找到您節點類型的 maxmemory 值。
3. 修改自訂參數群組，將參數 reserved-memory 設為您在上一個步驟中計算得到的位元組數。下列 AWS CLI 範例假設您正在 2.8.22 OSS 之前執行 Redis 版本，且需要保留節點的一半 maxmemory。如需詳細資訊，請參閱 [修改 ElastiCache 參數群組](#)。

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis28-m3xl \
 --parameter-name-values "ParameterName=reserved-memory,
 ParameterValue=7130316800"
```

您所使用的每個節點類型都需要個別的自訂參數群組，因為每個節點類型都有不同的 maxmemory 值。因此，每個節點類型都需要不同的 reserved-memory 值。

4. 修改 Redis OSS 叢集或複寫群組以使用您的自訂參數群組。

下列 CLI 範例會修改叢集 my-redis-cluster，以立即 redis28-m3xl 開始使用自訂參數群組。如需詳細資訊，請參閱 [修改 ElastiCache 叢集](#)。

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cluster \
 --cache-parameter-group-name redis28-m3xl \
 --apply-immediately
```

下列CLI範例會修改複寫群組my-redis-repl-grp，以立即redis28-m3x1開始使用自訂參數群組。如需更多詳細資訊，[修改複寫群組](#)。

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-repl-grp \
 --cache-parameter-group-name redis28-m3x1 \
 --apply-immediately
```

## 參數 reserved-memory-percent

Amazon 於 2017 年 3 月 16 日 ElastiCache 推出 參數，reserved-memory-percent 並在所有 版本上提供 ElastiCache ( Redis OSS )。reserved-memory-percent 旨在簡化您所有叢集上的預留記憶體管理工作。其做法是讓您針對每個參數群組系列 (例如 redis2.8)，各使用一個參數群組來管理叢集的預留記憶體，而不論節點類型為何。reserved-memory-percent 的預設值為 25 (25%)。

參數 reserved-memory-percent 為特定參數，且 ElastiCache 不屬於一般 Redis OSS 分佈。

如果您的叢集使用 r6gd 系列的節點類型，且您的記憶體使用量達到 75%，則會自動觸發資料分層。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

若要使用 保留記憶體 reserved-memory-percent

若要使用 reserved-memory-percent 來管理 ElastiCache ( Redis OSS ) 叢集上的記憶體，請執行下列其中一項操作：

- 如果您執行 Redis OSS 2.8.22 或更新版本，請將預設參數群組指派給叢集。預設值 25% 應當足夠。如果為較舊的版本，請依照以下說明的步驟變更該值。
- 如果您在 2.8.22 OSS 之前執行 Redis 版本，則可能需要保留比 reserved-memory-percent 的預設 25% 更多的記憶體。若要完成此操作，請遵循下列程序：

若要變更 的百分比值 reserved-memory-percent

1. 建立自訂參數群組，指定與您所執行引擎版本相符的參數群組系列，例如指定 redis2.8 參數群組系列。必須有一個自訂參數群組，因為您無法修改預設參數群組。如需詳細資訊，請參閱[建立 ElastiCache 參數群組](#)。

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name redis28-50 \
 --description "Redis OSS 2.8.x 50% reserved" \
 --apply-immediately
```

```
--cache-parameter-group-family redis2.8
```

由於 `reserved-memory-percent` 是以節點 `maxmemory` 的百分比為單位來預留記憶體，因此您不需要每個節點類型都有一個自訂參數群組。

2. 修改自訂參數群組，將 `reserved-memory-percent` 設為 50 (50%)。如需詳細資訊，請參閱 [修改 ElastiCache 參數群組](#)。

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis28-50 \
 --parameter-name-values "ParameterName=reserved-memory-percent,
 ParameterValue=50"
```

3. 針對執行 OSS Redis 早於 2.8.22 版本的任何 Redis OSS 叢集或複寫群組，使用此自訂參數群組。

下列 CLI 範例會修改 Redis OSS 叢集 `my-redis-cluster`，以立即 `redis28-50` 開始使用自訂參數群組。如需詳細資訊，請參閱 [修改 ElastiCache 叢集](#)。

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cluster \
 --cache-parameter-group-name redis28-50 \
 --apply-immediately
```

下列 CLI 範例會修改 Redis OSS 複寫群組 `my-redis-repl-grp`，以立即 `redis28-50` 開始使用自訂參數群組。如需詳細資訊，請參閱 [修改複寫群組](#)。

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-repl-grp \
 --cache-parameter-group-name redis28-50 \
 --apply-immediately
```

## 指定您的預留記憶體管理參數

如果您是 2017 年 3 月 16 日目前的 ElastiCache 客戶，您的預設預留記憶體管理參數 `reserved-memory` 為預留記憶體的零 (0) 位元組。如果您在 2017 年 3 月 16 日之後成為 ElastiCache 客戶，您的預設預留記憶體管理參數為 25% `reserved-memory-percent` 的節點記憶體預留。無論您在何時建立 ElastiCache (Redis OSS) 叢集或複寫群組，都是如此。不過，您可以使用或 AWS CLI 變更預留記憶體管理參數 ElastiCache API。

參數 `reserved-memory` 和 `reserved-memory-percent` 不會同時存在。每個參數群組一律只有一個參數，但絕不能兩者都有。您可以透過修改參數群組，來變更參數群組用於預留記憶體管理的參數。此參數群組必須是自訂參數群組，因為您無法修改預設參數群組。如需詳細資訊，請參閱[建立 ElastiCache 參數群組](#)。

若要指定 `reserved-memory-percent`

若要使用 `reserved-memory-percent` 做為您的預留記憶體管理參數，請使用 `modify-cache-parameter-group` 命令修改自訂參數群組。使用 `parameter-name-values` 參數來指定 `reserved-memory-percent` 和其值。

下列CLI範例會修改自訂參數群組，`redis32-cluster-on`以便使用 `reserved-memory-percent`管理預留記憶體。您必須指派一個值給 `ParameterValue`，參數群組才能使用 `ParameterName` 參數進行預留記憶體管理。如需詳細資訊，請參閱[修改 ElastiCache 參數群組](#)。

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis32-cluster-on \
 --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

指定 `reserved-memory`

若要使用 `reserved-memory` 做為您的預留記憶體管理參數，請使用 `modify-cache-parameter-group` 命令修改自訂參數群組。使用 `parameter-name-values` 參數來指定 `reserved-memory` 和其值。

下列CLI範例會修改自訂參數群組，`redis32-m3x1`以便使用 `reserved-memory`管理預留記憶體。您必須指派一個值給 `ParameterValue`，參數群組才能使用 `ParameterName` 參數進行預留記憶體管理。由於引擎版本是 2.8.22 以後的版本，因此我們會將值設為 `3565158400`，這是 `cache.m3.xlarge` 的 25% `maxmemory`。如需詳細資訊，請參閱[修改 ElastiCache 參數群組](#)。

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis32-m3x1 \
 --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

## 使用 Valkey 和 Redis OSS自行設計的叢集時的最佳實務

在 Valkey 或 Redis 中使用自行設計的叢集時，要記住多可用區域用途、具有足夠的記憶體、叢集大小調整和盡可能減少停機時間都是有用的概念OSS。建議您檢閱並遵循這些最佳實務。

主題

- [使用異地同步備份將停機時間降至最低](#)
- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)
- [線上叢集大小調整](#)
- [將維護期間的停機時間縮到最短](#)

## 使用異地同步備份將停機時間降至最低

有些執行個體可能需要 ElastiCache Valkey 或 Redis OSS 取代主要節點；這些執行個體包括特定類型的計劃維護，以及主要節點或可用區域故障的不太可能事件。

此取代會導致叢集出現一些停機情況，但如果啟用異地同步備份，停機時間可降至最低。主節點的角色會自動容錯移轉到其中一個僅供讀取複本。不需要建立新的主節點和佈建，因為 ElastiCache 會以透明方式處理。此容錯移轉及複本提升可確保您能在提升完成時立即繼續寫入新的主要節點。

請參閱 [搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS](#)，以進一步了解多可用區並盡可能減少停機時間。

## 確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照

Valkey 7.2 及更新版本中的快照和同步，以及 Redis 2.8.22 及更新 OSS 版本

Valkey 預設支援快照和同步。Redis OSS 2.8.22 推出無叉儲存程序，可讓您將更多記憶體配置到應用程式的使用，而不會在同步和儲存期間增加交換用量。如需詳細資訊，請參閱 [同步與備份的實作方式](#)。

### 2.8.22 版之前的 Redis OSS 快照和同步

當您使用 ElastiCache (Redis OSS) 時，Redis 會在許多情況下 OSS 呼叫背景寫入命令：

- 建立備份的快照時。
- 和複寫群組中的主要節點同步複本時。
- 為 Redis 啟用僅附加檔案功能 (AOF) 時 OSS。
- 將複本提升為主節點時 (這會引發主節點/複本節點同步)。

每當 Redis OSS 執行背景寫入程序時，您必須有足夠的可用記憶體來容納程序額外負荷。無法擁有足夠的記憶體會造成程序失敗。因此，在建立 Redis OSS 叢集時，請務必選擇具有足夠記憶體的節點執行個體類型。



## 使用 Valkey 和 Redis 的背景寫入程序與記憶體用量 OSS

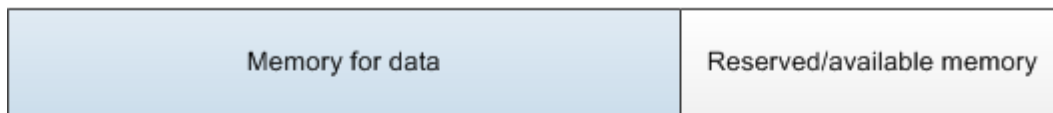
每當呼叫背景寫入程序時，Valkey 和 Redis 都會OSS放棄其程序（請記住，這些引擎是單一螺紋）。一個叉子會持續將資料保留到 Redis .rdb OSS 快照檔案中的磁碟。另一個分支則會服務所有讀取及寫入操作。為了確保您的快照是 point-in-time 快照，所有資料更新和新增都會寫入與資料區域分開的可用記憶體區域。

只要您擁有足夠的可用記憶體，可在資料持久存放至磁碟時用於記錄所有寫入操作，便不會發生記憶體不足的問題。若您的狀況符合以下內容中的任何一項，便可能會發生記憶體不足的問題：

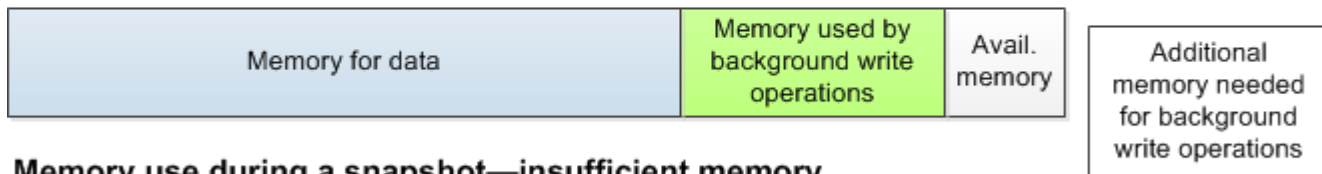
- 您的應用程式執行許多寫入操作，因此需要大量的可用記憶體來接受新的或更新後的資料。
- 您只有極少的可用記憶體可用於寫入新的或更新後的資料。
- 您擁有大型的資料集，需要長時間才能持久存放至磁碟，因此需要大量的寫入操作。

下圖說明執行背景寫入程序時的記憶體使用方式。

### Memory use prior to a snapshot



### Memory use during a snapshot—sufficient memory



### Memory use during a snapshot—insufficient memory



如需執行備份時對效能造成影響的資訊，請參閱[自行設計叢集之備份的效能影響](#)

如需 Valkey 和 Redis 如何OSS執行快照的詳細資訊，請參閱 <http://valkey.io> 。

如需區域及可用區域的詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。

### 在執行背景寫入時避免耗盡記憶體

每當呼叫 BGSAVE 或 BGREWRITEAOF 等背景寫入程序時，為了防止程序失敗，您必須擁有比寫入操作在程序期間耗用的更多記憶體。最壞的情況是，在背景寫入操作期間，每個記錄都會更新，並將一些新

記錄新增至快取。因此，建議您將 2.8.22 之前的 Redis OSS 版本 `reserved-memory-percent` 設為 50 ( 50% )，而 Valkey 和所有 Redis 2.8.22 及更新 OSS 版本設為 25 ( 25% )。

`maxmemory` 值指出您可用於資料及操作額外負荷的可用記憶體。因為您無法修改預設參數群組中的 `reserved-memory` 參數，因此您必須為叢集建立自訂參數群組。的預設值 `reserved-memory` 為 0，這允許 Redis 使用資料消耗 OSS 所有記憶體上限，可能留下太少記憶體供其他用途使用，例如背景寫入程序。針對根據節點執行個體類型的 `maxmemory` 值，請參閱 [Redis OSS 節點類型特定參數](#)。

您也可以使用 `reserved-memory` 參數來減少方塊上使用的記憶體量。

如需中 Valkey 和 Redis 特定參數的詳細資訊 ElastiCache，請參閱 [Valkey 和 Redis OSS 參數](#)。

如需建立及修改參數群組的資訊，請參閱 [建立 ElastiCache 參數群組](#) 及 [修改 ElastiCache 參數群組](#)。

## 線上叢集大小調整

「重新碎片」涉及將碎片或節點新增到您的叢集或從叢集移除碎片或節點，並重新分配鍵的空間。多項事物會對重新碎片操作造成影響，例如叢集上的負載、記憶體使用率，以及整體資料大小。為了取得最佳體驗，我們建議您遵循統一工作負載模式分佈的整體叢集最佳實務。此外，我們建議您採取以下步驟。

在初始化重新碎片前，我們建議以下內容：

- 測試您的應用程式 - 在預備環境中測試應用程式於重新分片期間的行為 (若可能的話)。
- 提早取得擴展問題的通知 - 重新分片是一項需要大量運算的操作。因此，我們建議在重新分割期間，將多核心執行個體的使用 CPU 率保持在 80% 以下，將單一核心執行個體的使用率保持在 50% 以下。在應用程式開始觀察擴展問題之前，監控 ElastiCache (Redis OSS) 指標並啟動重新分配。可進行追蹤的有用指標包括 `CPUUtilization`、`NetworkBytesIn`、`NetworkBytesOut`、`CurrConnections`、`NewConnections` 及 `BytesUsedForCacheItems`。
- 在向內擴展前確保有足夠的可用記憶體 - 若您要向內擴展，請確保碎片上保留的可用記憶體至少是待移除碎片所使用記憶體的 1.5 倍。
- 在離峰時段期間起始重新分片程序 - 此做法有助於減少重新分片作業期間的延遲，以及對用戶端造成的輸送量影響。它也有助於更快完成重新碎片，因為有更多的資源可用於重新分配位置。
- 檢閱用戶端逾時行為 - 有些用戶端可能會在線上叢集調整大小期間產生較高的延遲。使用較高的逾時設定您的用戶端程式庫，有助於解決該情況，即使伺服器上負載較高，系統仍有時間連線。在某些情況下，您可能想要對伺服器開啟大量連線。在這些情況下，請考慮將指數退避新增到重新連線邏輯。這麼做可幫助防止爆量的新連線同時衝擊伺服器。



- 在每個碎片上載入函數 – 向外擴展叢集時，ElastiCache 會自動將其中一個現有節點（隨機選取）中載入的函數複製到新的節點。如果您的叢集具有 Valkey 7.2 及更高版本，或 Redis OSS 7.0 或更高版本，且您的應用程式使用[函數](#)，建議您先將所有函數載入所有碎片，然後再向外擴展，這樣您的叢集就不會在不同的碎片上最後出現不同的函數。

在重新碎片之後，請注意以下內容：

- 若目標碎片上的可用記憶體不足，向內擴展可能會僅部分成功。若發生這種結果，請檢閱可用記憶體，並視需要重試操作。目標碎片上的資料不會遭刪除。
- 擁有大量項目的位置不會進行遷移。特別是擁有大於 256 MB 項目位置的後序列化不會進行遷移。
- 在重新分片操作期間，Lua 指令碼內不支援 FLUSHALL 和 FLUSHDB 命令。在 Redis OSS 6 之前，如果命令在要遷移的插槽上操作，則不支援該 BRPOPLPUSH 命令。

## 將維護期間的停機時間縮到最短

叢集模式組態在受管或未受管操作期間具有最佳的可用性。我們建議您使用連線到叢集探索端點的叢集模式支援用戶端。若是停用叢集模式，建議您針對所有寫入作業使用主要端點。

針對讀取活動，應用程式也可連線到叢集中的任何節點。與主要端點不同，節點端點會解析至特定端點。若您在叢集中進行變更（例如新增或刪除複本），您必須更新您應用程式中的節點端點。這就是為什麼停用叢集模式的原因，我們建議您使用讀取器端點進行讀取活動。

如果在叢集中啟用 AutoFailover，則主要節點可能會變更。因此，應用程式應確認節點的角色並更新所有讀取端點。這麼做可確保不會對主節點造成過重的負載。在 AutoFailover 停用的情況下，節點的角色不會變更。不過，與啟用的叢集 AutoFailover 相比，受管或未受管操作的停機時間較高。

避免將讀取請求導向到單一讀取複本節點，因為其不可用的情況可能導致讀取中斷。從主要複本進行後援讀取，或確定您至少有兩個讀取複本，以避免在維護期間發生任何讀取中斷。

## Memcached 的快取策略

在下列主題中，您可以找到填充和維護 Memcached 快取的策略。

您實作用於填入和維護快取的策略，取決於您要快取的資料，以及該資料的存取模式。例如，您可能不會想對遊戲網站前十名排行榜和趨勢新聞報導使用相同的策略。在本節的其餘部分，我們會討論常見的快取維護策略，以及其優點和缺點。

### 主題

- [延遲載入](#)
- [全部寫入](#)
- [新增 TTL](#)
- [相關主題](#)

### 延遲載入

如同名稱所隱含的意義，延遲載入是一項快取策略，只有在必要時才會將資料載入快取中。它的運作方式如下。

Amazon ElastiCache 是記憶體內鍵值存放區，位於您的應用程式與其存取的資料存放區（資料庫）之間。每當您的應用程式請求資料時，它都會先向 ElastiCache 快取提出請求。如果資料存在於快取中且為最新版本，會將資料ElastiCache 傳回至您的應用程式。如果資料不存在於快取中或已過期，則應用程式會向資料存放區請求資料。接著資料存放區會將資料傳回至應用程式。。應用程式會將從存放區收到的資料寫入快取中。如此一來，下次收到請求時就能更快速擷取這些資料。

當資料存在於快取中且未過期，就會發生快取命中：

1. 應用程式向快取請求資料。
2. 快取將資料傳回給應用程式。

當資料不存在於快取中已過期，就會發生快取未命中：

1. 應用程式向快取請求資料。
2. 快取中沒有所請求的資料，因此傳回 null。
3. 應用程式請求並收到來自資料庫的資料。
4. 應用程式以新資料更新快取。

## 延遲載入的優點和缺點

延遲載入的優點如下：

- 只會將請求的資料加以快取。

因為大部分資料永遠不會被請求，延遲載入可避免將快取填滿未請求的資料。

- 節點故障不會成為應用程式的嚴重問題。

當節點故障並且由新的空白節點取代時，應用程式會繼續運作，只是會增加延遲。對新節點提出請求時，每次快取未命中都會產生一個資料庫查詢。同時會將資料複本新增至快取中，以便從快取擷取後續的請求。

延遲載入的缺點如下：

- 快取遺漏會有懲罰。每次快取未命中都會產生 3 趟行程：

1. 向快取初始請求資料
2. 查詢資料庫來取得資料
3. 將資料寫入快取

這些未命中可能會造成應用程式取得資料的時間出現明顯的延遲。

- 資料過時。

如果只在快取未命中時將資料寫入快取，則快取中的資料可能變得過時。當資料庫中的資料變更時，並不會更新快取，因此會產生此結果。若要解決這個問題，您可以使用 [全部寫入](#) 和 [新增 TTL](#) 策略。

## 延遲載入虛擬程式碼範例

以下是延遲載入邏輯的虛擬程式碼範例。

```
// *****
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
// retrieved from the database,
// added to the cache, and
// returned to the application
```

```
// *****
get_customer(customer_id)

 customer_record = cache.get(customer_id)
 if (customer_record == null)

 customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",
customer_id)
 cache.set(customer_id, customer_record)

 return customer_record
```

在此範例中，取得資料的應用程式程式碼如下。

```
customer_record = get_customer(12345)
```

## 全部寫入

每當有資料寫入資料庫，全部寫入 (write-through) 策略就會在快取中新增或更新資料。

### 全部寫入的優點和缺點

全部寫入的優點如下：

- 快取中的資料絕不會過時。

快取中的資料每次寫入資料庫時都會進行更新，因此快取中的資料永遠是最新的。

- 寫入懲罰與讀取懲罰。

每次寫入都牽涉到兩個來回行程：

1. 一個寫入快取
2. 一個寫入資料庫

這會對程序增加延遲。也就是說，相較於擷取資料，最終使用者一般更能容忍更新資料時的延遲。固有的想法是更新需要的動作較多因此會花費較長時間。

全部寫入的缺點如下：

- 遺漏資料。

如果啟動新節點，無論是因為節點故障或水平擴展，都會發生資料未命中。這些資料會繼續處於未命中狀態，直到在資料庫上新增或更新為止。您可以藉由搭配全部寫入實作[延遲載入](#)來盡可能減少此情形。

- 快取流失。

大部分資料都永遠不會被讀取，導致資源的浪費。透過[新增存留時間 \(TTL\) 值](#)，您可以將浪費的空間降到最低。

## 全部寫入虛擬程式碼範例

下列是全部寫入邏輯的虛擬程式碼範例。

```
// *****
// function that saves a customer's record.
// *****
save_customer(customer_id, values)

 customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
 cache.set(customer_id, customer_record)
 return success
```

在此範例中，取得資料的應用程式程式碼如下。

```
save_customer(12345, {"address": "123 Main"})
```

## 新增 TTL

延遲載入允許使用過時資料，但不會因空白節點而失敗。全部寫入可確保資料總是保持最新狀態，但可能因為空白節點而失敗，且可能對快取填入多餘的資料。透過為每個寫入新增存留時間 (TTL) 值，您可以擁有每個策略的優點。同時可以大幅避免將快取塞滿多餘的資料。

存留時間 (TTL) 是整數值，可指定金鑰過期前的秒數。Valkey 或 Redis OSS 可以為此值指定秒或毫秒。Memcached 需以秒為單位指定此值。應用程式嘗試讀取過期的索引鍵時，系統會視為找不到索引鍵。資料庫會查詢索引鍵並更新快取。這種方法並不能保證值不會過時。不過會確保資料不致於過時太久，且需要偶爾從資料庫重新整理快取中的值。

如需詳細資訊，請參閱 [Valkey 和 Redis OSS 命令](#) 或 [Memcached set 命令](#)。

## TTL 虛擬程式碼範例

以下是具有 的寫入邏輯的虛擬程式碼範例TTL。

```
// *****
// function that saves a customer's record.
// The TTL value of 300 means that the record expires
// 300 seconds (5 minutes) after the set command
// and future reads will have to query the database.
// *****
save_customer(customer_id, values)

 customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
 cache.set(customer_id, customer_record, 300)

return success
```

以下是使用 延遲載入邏輯的虛擬程式碼範例TTL。

```
// *****
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
// retrieved from the database,
// added to the cache, and
// returned to the application.
// The TTL value of 300 means that the record expires
// 300 seconds (5 minutes) after the set command
// and subsequent reads will have to query the database.
// *****
get_customer(customer_id)

 customer_record = cache.get(customer_id)

 if (customer_record != null)
 if (customer_record.TTL < 300)
 return customer_record // return the record and exit function

// do this only if the record did not exist in the cache OR
// the TTL was >= 300, i.e., the record in the cache had expired.
 customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
 cache.set(customer_id, customer_record, 300) // update the cache
```

```
return customer_record // return the newly retrieved record and exit
function
```

在此範例中，取得資料的應用程式程式碼如下。

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

## 相關主題

- [記憶體內資料存放區](#)
- [選擇引擎和版本](#)
- [擴展 ElastiCache](#)

## 在 中管理自行設計的叢集 ElastiCache

ElastiCache 提供兩種部署選項，即無伺服器快取和自行設計的叢集。每個都有自己的功能和要求。

本節包含主題，可協助您管理自行設計的叢集。

### Note

這些主題不適用於 ElastiCache Serverless。

## 主題

- [Auto Scaling Valkey 和 Redis OSS叢集](#)
- [修改叢集模式](#)
- [使用全域資料存放區跨 AWS 區域複寫](#)
- [使用複寫群組的高可用性](#)
- [管理 ElastiCache 叢集維護](#)
- [使用參數群組設定引擎 ElastiCache 參數](#)

## Auto Scaling Valkey 和 Redis OSS叢集

### 必要條件

ElastiCache Auto Scaling 僅限於下列項目：

- 執行 Valkey 7.2 以上版本或執行 Redis OSS引擎 6.0 以上版本的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集
- 執行 Valkey 7.2 以上版本或執行 Redis OSS引擎 7.0.7 以上版本的資料分層 ( 啟用叢集模式 ) 叢集
- 執行個體大小 - 大型、XLarge、2XLarge
- 執行個體類型系列 - R7g、R6g、R6gd、R5、M7g、M6g、M5、C7gn
- 在全球資料存放區、Outpost 或本機區域中執行的叢集 ElastiCache 不支援 Auto Scaling in。

### 使用 Valkey 或 Redis ElastiCache Auto Scaling自動管理容量 OSS

ElastiCache 使用 Valkey 或 Redis 自動擴展OSS是自動增加或減少 ElastiCache 服務中所需碎片或複本的能力。ElastiCache 利用 Application Auto Scaling 服務提供此功能。如需詳細資訊，請參閱 [Application Auto Scaling](#)。若要使用自動擴展，您可以定義並套用擴展政策，該政策使用您指派的 CloudWatch 指標和目標值。ElastiCache 自動擴展使用政策來增加或減少回應實際工作負載的執行個體數量。

您可以使用 AWS Management Console，根據預先定義的指標套用擴展政策。predefined metric 是在列舉中定義，如此您就可以在程式碼中依名稱指定指標，或是在 AWS Management Console中使用指標。使用 AWS Management Console時無法選擇自訂指標。或者，您可以使用 AWS CLI 或 Application Auto ScalingAPI，根據預先定義的或自訂指標套用擴展政策。

ElastiCache 搭配 Valkey 或 Redis OSS支援擴展下列維度：

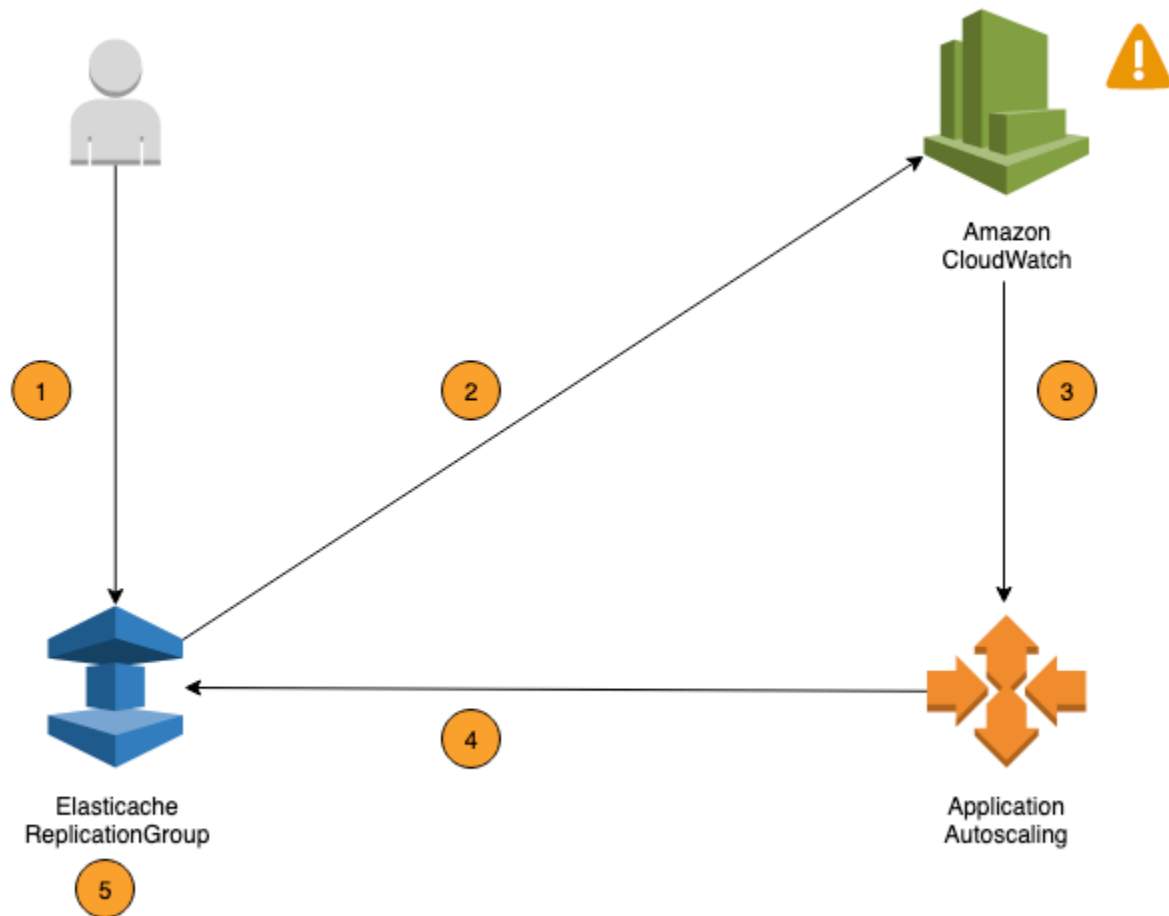
- 碎片 - 自動新增/移除叢集中的碎片，類似於手動線上重新分片。在此情況下，ElastiCache自動擴展會觸發代表您擴展。
- 複本 - 自動新增/移除叢集中的複本，類似於手動增加/減少複本作業。ElastiCache 透過 Valkey 或 Redis OSS自動擴展，在叢集中的所有碎片中均勻地新增/移除複本。

ElastiCache 搭配 Valkey 或 Redis OSS支援下列類型的自動擴展政策：

- [目標追蹤擴展政策](#) - 根據特定指標的目標值，增加或減少服務執行的碎片/複本數。這與您運用電熱器維持家中溫度的方式很類似。您只需選取溫度，電熱器會自行執行其餘操作。



- [ElastiCache 使用 Valkey 或 Redis OSS自動擴展的應用程式排程擴展](#) – 根據日期和時間增加或減少服務執行的碎片/複本數量。



下列步驟摘要說明 ElastiCache 使用 Valkey 或 Redis OSS自動擴展程序的，如上圖所示：

1. 您可以為複寫群組建立 ElastiCache 自動擴展政策。
2. ElastiCache 使用 Valkey 或 Redis 自動擴展會代表您OSS建立一對 CloudWatch 警示。每對指標皆代表指標的上限與下限。當叢集的實際使用率偏離您的目標使用率持續一段時間時，就會觸發這些 CloudWatch 警示。您可以在主控台中檢視警示。
3. 如果設定的指標值超過您的目標使用率（或低於目標）達特定時間長度，CloudWatch 會觸發警示來叫用自動擴展來評估擴展政策。
4. ElastiCache 透過 Valkey 或 Redis OSS自動擴展，發出修改請求來調整叢集容量。

5. ElastiCache 使用 Valkey 或 Redis OSS處理修改請求，動態增加（或減少）叢集碎片/複本容量，以便接近您的目標使用率。

若要了解如何 ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling，假設您有名為的叢集UsersCluster。透過監控的 CloudWatch 指標UsersCluster，您可以判斷當流量處於其峰值時，叢集所需的最大碎片，以及當流量處於其最低點時所需的最小碎片。您也可以決定UsersCluster叢集CPU使用的目標值。ElastiCache 自動擴展使用其目標追蹤演算法，以確保的佈建碎片會視需要UsersCluster進行調整，以便利用率保持在目標值或接近目標值。

#### Note

擴展可能需要很長的時間，並且需要額外的叢集資源才能讓碎片重新平衡。ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 只有在實際工作負載持續幾分鐘保持升高（或降低）時，才能修改資源設定。自動擴展目標追蹤演算法會長期嘗試將目標使用率保持在所選值或接近所選值。

## Auto Scaling 政策

擴展政策具有下列元件：

- 目標指標 – ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 來決定擴展時間和數量的 CloudWatch 指標。
- 最大和最小容量 - 進行擴展的碎片或複本數量下限與上限。

#### Important

建立自動擴展政策時，如果目前容量高於設定的最大容量，我們會在政策建立 MaxCapacity 期間 scaleIn 前往。同樣地，如果目前容量低於設定的最小容量，我們會 scaleOut 前往 MinCapacity。

- 冷卻時間 - 完成規模縮減或規模擴展活動後，可以開始進行另一個水平擴展活動之前需等待的時間長度 (秒)。
- 服務連結角色 – 連結至特定 AWS 服務的 AWS Identity and Access Management (IAM) 角色。服務連結角色包含服務 AWS 代表您呼叫其他服務所需的所有許可。ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 會自動為您產生此角色 AWSServiceRoleForApplicationAutoScaling\_ElastiCacheRG。

- 啟用或停用規模縮減活動 - 為政策啟用或停用規模縮減活動的功能。

## 主題

- [Auto Scaling 的目標指標](#)
- [容量下限和上限](#)
- [冷卻時間](#)
- [啟用或停用規模縮減活動](#)

## Auto Scaling 的目標指標

在此類型的政策中，預先定義的或自訂指標和指標的目標值是在目標追蹤擴展政策組態中指定。ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 會建立和管理觸發擴展政策的 CloudWatch 警示，並根據指標和目標值計算擴展調整。規模調整政策會視需要新增或移除碎片/複本，讓指標保持在等於或接近指定目標值。除了讓指標保持在接近目標值之外，目標追蹤規模調整政策也會配合因為變更工作負載所造成的指標波動而進行調整。這樣的政策也能將資料庫叢集可用碎片/複本數量的快速波動減到最低。

例如，想想看當規模調整政策使用預先定義的平均 `ElastiCachePrimaryEngineCPUUtilization` 指標時的情況。此類政策可以保持 CPU 或接近指定的使用率百分比，例如 70%。

### Note

在每個叢集上，您只能為每一個目標指標建立一個 Auto Scaling 政策。

## 容量下限和上限

### 碎片

您可以使用 ElastiCache Valkey 或 Redis OSS 自動擴展，指定可以由擴展的碎片數量上限。此值必須小於或等於 250，而下限為 1。您也可以指定要透過自動擴展管理的碎片數量下限。此值下限為 1，而且必須小於或等於指定的碎片數量上限值 250。

### 複本

您可以使用 ElastiCache Valkey 或 Redis OSS 自動擴展指定要由管理的複本數量上限。此值必須等於或大於 5。您也可以指定要透過自動擴展管理的複本數量下限。此值下限為 1，而且必須小於或等於指定的複本數量上限值 5。

若要判斷一般流量所需的碎片/複本數量上限和下限，請使用預期的傳輸流量來傳送到您的模型，以測試您的 Auto Scaling 組態設定。

#### Note

ElastiCache 使用 Valkey 或 Redis OSS 自動擴展政策會增加叢集容量，直到達到您定義的大小上限，或直到套用服務限制為止。若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

#### Important

在沒有流量時縮減規模。如果變體的流量變成零，ElastiCache 則使用 Valkey 或 Redis OSS 會自動擴展至指定的執行個體數量下限。

## 冷卻時間

藉由新增會影響調整叢集規模的冷卻時間，您可以調整目標追蹤擴展政策的回應能力。冷卻時間會封鎖後續的擴展或縮減請求，直到冷卻時間到期。這可減緩 ElastiCache 使用 Valkey 或 Redis OSS 叢集進行縮減請求時，刪除您中的碎片/複本，以及建立縮減請求的碎片/複本。您可以指定下列其中一種冷卻時間：

- 縮減活動可減少叢集中的碎片/複本數量。規模縮減冷卻時間會指定在規模縮減動作完成之後，另一個規模縮減動作可以再開始執行之前的等待時間長度 (秒)。
- 橫向擴展活動會增加叢集中的碎片/複本數量。橫向擴展冷卻時間會指定在橫向擴展動作完成之後，可以再開始執行另一個橫向擴展動作之前的等待時間長度 (秒)。

若未指定規模縮減或水平擴展的冷卻時間，則水平擴展的預設值是 600 秒，規模縮減為 900 秒。

## 啟用或停用規模縮減活動

您可以啟用或停用政策的規模縮減動作。啟用規模縮減活動可讓規模調整政策刪除碎片/複本。規模縮減動作啟用時，規模調整政策中的規模縮減冷卻時間會套用至規模縮減動作。停用規模縮減活動可防止規模調整政策刪除碎片/複本。

**Note**

橫向擴展活動一律會啟用，以便擴展政策可以視需要 ElastiCache 使用 Valkey 或 Redis OSS 碎片/複本建立。

## IAM Auto Scaling 所需的許可

ElastiCache 透過 Valkey 或 Redis OSS Auto Scaling，可透過 ElastiCache CloudWatch、和 Application Auto Scaling 的組合實現 APIs。叢集是使用 ElastiCache ( Redis OSS ) 建立和更新，警示是使用 建立 CloudWatch，擴展政策是使用 Application Auto Scaling 建立。除了建立和更新叢集的標準 IAM 許可外，存取 ElastiCache Auto Scaling 設定 IAM 的使用者必須具有支援動態擴展之服務的合適許可。IAM 使用者必須具有使用下列範例政策中所示動作的許可：

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "application-autoscaling:*",
 "elasticache:DescribeReplicationGroups",
 "elasticache:ModifyReplicationGroupShardConfiguration",
 "elasticache:IncreaseReplicaCount",
 "elasticache:DecreaseReplicaCount",
 "elasticache:DescribeCacheClusters",
 "elasticache:DescribeCacheParameters",
 "cloudwatch:DeleteAlarms",
 "cloudwatch:DescribeAlarmHistory",
 "cloudwatch:DescribeAlarms",
 "cloudwatch:DescribeAlarmsForMetric",
 "cloudwatch:GetMetricStatistics",
 "cloudwatch:ListMetrics",
 "cloudwatch:PutMetricAlarm",
 "cloudwatch:DisableAlarmActions",
 "cloudwatch:EnableAlarmActions",
 "iam:CreateServiceLinkedRole",
 "sns:CreateTopic",
 "sns:Subscribe",
 "sns:Get*",
 "sns:List*"
],
 }
],
}
```

```
 "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-cluster"
 }
}
}
```

## 服務連結角色

ElastiCache 使用 Valkey 或 Redis OSS 自動擴展服務的 也需要描述叢集和 CloudWatch 警示的許可，以及代表您修改 ElastiCache 目標容量的許可。如果您為叢集啟用 Auto Scaling，它會建立名為 的服務連結角色 `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`。此服務連結角色會授予 ElastiCache 自動擴展許可，以描述政策的警示、監控機群的目前容量，以及修改機群的容量。服務連結角色是 ElastiCache 自動擴展的預設角色。如需詳細資訊，請參閱 Application Auto Scaling 使用者指南中的 [ElastiCache \( Redis OSS \) 自動擴展的服務連結角色](#)。

## Auto Scaling 最佳實務

註冊 Auto Scaling 前，我們建議以下事項：

1. 僅使用一個追蹤指標 – 識別叢集是否具有 CPU 或資料密集型工作負載，並使用對應的預先定義指標來定義擴展政策。
  - 引擎 CPU：ElastiCachePrimaryEngineCPUUtilization ( 碎片維度 ) 或 ElastiCacheReplicaEngineCPUUtilization ( 複本維度 )
  - 資料庫使用量：ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage 此擴展政策在叢集上將 `maxmemory-policy` 設為 `noeviction` 時效果最佳。

我們建議您避免叢集上每個維度的多個政策。如果任何目標追蹤政策已準備好進行擴展，ElastiCache 則透過 Valkey 或 Redis OSS Auto 擴展將擴展可擴展的目標，但只有在所有目標追蹤政策 ( 啟用擴展部分 ) 都準備好擴展時，才會擴展。如果多個政策指示可擴展的目標在同一時間進行水平擴展或縮減，則會根據針對水平擴展和縮減提供最大容量的政策進行擴展。

2. 用於目標追蹤的自訂指標 – 使用自訂指標進行目標追蹤時需小心謹慎，因為 Auto Scaling 最適合用於根據為策略所選指標的根據為策略所選指標的變更，依比例擴展/縮減規模。如果這些指標未與用於政策建立的擴展操作比例地變化，則可能會導致持續的橫向擴展或縮減操作，這可能會影響可用性或成本。

對於資料分層叢集 (r6gd 系列執行個體類型)，請避免使用記憶體指標進行擴展。

3. 排程規模調整 - 如果您發現工作負載具確定性 (會在特定時間達到高/低負載)，建議您使用排程規模調整功能，根據需求設定目標容量。目標追蹤可在需要更多資源時橫向擴展，並在需要較少資源時縮減，最適合用於讓非確定性工作負載和叢集以所需的目標指標運作。

4. 停用縮減 - 針對目標追蹤自動調整規模的功能最適合用於工作負載逐漸增加/減少的叢集，因為指標的峰值/低谷可能會觸發連續的橫向擴展/縮減振盪。若要避免這種振盪，可以從停用縮減功能開始，您之後可以隨時根據需求手動縮減。
5. 測試您的應用程式 - 建議您使用預估的最小/最大工作負載來測試應用程式，以確定叢集所需的絕對碎片/複本數下限與上限，同時建立規模調整政策以避免可用性問題。自動調整規模可以橫向擴展至為目標設定的最大值，也可縮減至最小值。
6. 定義目標值 - 您可以分析叢集使用率在四週期間內的對應 CloudWatch 指標，以判斷目標值閾值。如果您仍不確定要選擇哪個值，建議從支援的最小預先定義指標值開始。
7. AutoScaling 在目標追蹤上，最適合工作負載分佈在碎片/複本維度的叢集。不均勻分配可能會導致：
  - 由於幾個熱碎片/複本上出現工作負載峰值/低谷，因此在沒有需要時進行擴展。
  - 即使具有熱碎片/複本，但由於整體平均接近目標，因此有需要時不進行縮放。

#### Note

擴展叢集時，ElastiCache 會自動將其中一個現有節點（隨機選取）中載入的函數複寫至新的節點。如果您的叢集具有 Valkey OSS 或 Redis 7.0 或更新版本，且您的應用程式使用[函數](#)，建議您先將所有函數載入所有碎片，然後再向外擴展，這樣您的叢集就不會最終在不同的碎片上產生不同的函數。

註冊之後 AutoScaling，請注意下列事項：

- 支援自動調整規模的組態具有限制，因此建議您不要變更已針對自動調整規模註冊之複寫群組的組態。範例如下：
  - 手動將執行個體類型修改為不支援的類型。
  - 為複寫群組與全域資料存放區建立關聯。
  - 變更 ReservedMemoryPercent 參數。
  - 手動將碎片/複本增加/減少至超出政策建立期間設定的容量上限/下限。

## 搭配碎片使用 Auto Scaling

使用 ElastiCache 時 AutoScaling，您可以將追蹤和排程政策與 Valkey 或 Redis OSS 引擎搭配使用。



以下提供目標追蹤和排程政策的詳細資訊，以及如何使用 AWS Management Console AWS CLI 和 套用這些政策APIs。

## 主題

- [目標追蹤擴展政策](#)
- [新增擴展原則](#)
- [註冊可擴展的目標](#)
- [定義擴展政策](#)
- [停用規模縮減活動](#)
- [套用擴展原則](#)
- [編輯擴展原則](#)
- [刪除擴展原則](#)
- [AWS CloudFormation 用於 Auto Scaling 政策](#)
- [排程擴展](#)

## 目標追蹤擴展政策

使用目標追蹤擴展政策，您可以選取指標及設定目標值。ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 會建立和管理觸發擴展政策的 CloudWatch 警示，並根據指標和目標值計算擴展調整。規模調整政策會視需要新增或移除複本，讓指標保持在等於或接近指定目標值。除了讓指標保持在接近目標值之外，目標追蹤規模調整政策也會配合指標中，因為負載模式波動所造成的波動調整，並將機群容量中的快速波動降到最低。

例如，想想看當規模調整政策使用預先定義的平均

ElastiCachePrimaryEngineCPUUtilization 指標搭配所設定目標值時的情況。此類政策可以保持CPU使用率為或接近指定的目標值。

## 預先定義的指標

預先定義的指標是參考指定 CloudWatch 指標的特定名稱、維度和統計資料（average）的結構。Auto Scaling 政策會為您的叢集定義下列其中一項預先定義的指標：



| 預先定義的指標名稱                                                 | CloudWatch 指標名稱                                | CloudWatch 指標維度             | 不合格執行個體類型 |
|-----------------------------------------------------------|------------------------------------------------|-----------------------------|-----------|
| ElastiCachePrimaryEngineCPUUtilization                    | EngineCPUUtilization                           | ReplicationGroupId, 角色 = 主要 | 無         |
| ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage | DatabaseCapacityUsageCountedForEvictPercentage | Valkey 或 Redis OSS 複寫群組指標   | 無         |
| ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage   | DatabaseMemoryUsageCountedForEvictPercentage   | Valkey 或 Redis OSS 複寫群組指標   | R6gd      |

### 資料分層執行個體類型無法使用

ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage，因為這些執行個體類型會將資料同時儲存在記憶體和 SSD。資料分層執行個體的預期使用案例是 100% 的記憶體用量，並視需要填滿 SSD。

### 碎片的 Auto Scaling 條件

當服務偵測到預先定義指標等於或大於目標設定，便會自動增加碎片容量。ElastiCache 使用 Valkey 或 Redis 將叢集碎片 OSS 擴展為等於兩個數字中較大者的數量：與目標的百分比變化和目前碎片的 20%。對於縮減，除非整體指標值低於定義目標的 75%，否則 ElastiCache 不會自動縮減。

以下提供水平擴展的範例，如果你有 50 個碎片，

- 如果您的目標違規率為 30%，ElastiCache 則 Valkey 或 Redis OSS 會縮減 30%，因此每個叢集會產生 65 個碎片。
- 如果您的目標違規率為 10%，ElastiCache 則依預設，Valkey 或 Redis 會橫向 OSS 擴展至少 20%，因此每個叢集會產生 60 個碎片。

對於縮減範例，如果您已選取目標值 60%，ElastiCache 則使用 Valkey 或 Redis 時，在指標小於或等於 45%（低於目標 60% 的 25%）之前 OSS，不會自動縮減。

## Auto Scaling 考量事項

請謹記以下幾點考量：

- 目標追蹤擴展政策假設在指定的指標超過目標值時，應執行向外擴展。當指定的指標低於目標值時，您無法使用目標追蹤擴展政策來擴展。ElastiCache 使用 Valkey 或 Redis 將碎片 OSS 擴展到叢集中現有碎片目標的至少 20% 偏差。
- 所指定指標的資料不足時，目標追蹤擴展政策不會執行擴展。政策不會執行縮減，因為縮減不會將資料不足解釋為低使用率。
- 您可能會看到目標值與實際指標資料點之間有些差距。這是因為 ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 時，在決定要新增或移除的容量時，一律會四捨五入或向下捨入，以保守的方式運作。這樣可防止新增不足的容量，或移除過多的容量。
- 為了確保應用程式可用性，服務可以根據指標依比例快速水平擴展，但需以更保守的方式縮減規模。
- 您可以針對 ElastiCache 具有 Valkey 或 Redis OSS 叢集的擁有多個目標追蹤擴展政策，前提是每個政策都使用不同的指標。ElastiCache（Redis OSS）Auto Scaling 的用意是一律優先考慮可用性，因此其行為會因目標追蹤政策是否已準備好橫向擴展或橫向擴展而有所不同。如果任何目標追蹤政策已準備好向外擴展，它就會將服務向外擴展，但只有在所有目標追蹤政策（已啟用向內擴展部分）都已準備好要向內擴展時才會向內擴展。
- 請勿編輯或刪除 ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 管理的目標追蹤擴展政策的 CloudWatch 警示。當您刪除擴展政策時，ElastiCache Auto Scaling 會自動刪除警示。
- ElastiCache Auto Scaling 不會阻止您手動修改叢集碎片。這些手動調整不會影響連接到擴展政策的任何現有 CloudWatch 警示，但可能會影響可能觸發這些 CloudWatch 警示的指標。
- 這些由 Auto Scaling 管理的 CloudWatch 警示是透過叢集中所有碎片的 AVG 指標來定義。因此，使用熱碎片可能會導致以下任一情況：
  - 因幾個觸發 CloudWatch 警示的熱碎片載入而不需要擴展
  - 由於 AVG 彙總了影響警示不違反的所有碎片，因此在需要時不會擴展。

- ElastiCache 對於每個叢集的節點，使用 Valkey 或 Redis OSS 預設限制仍然適用。因此，當選擇使用 Auto Scaling 且您希望節點數上限超過預設配額時，請至 [AWS 服務配額](#) 請求增加配額，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。
- 請確定您的 中有足夠的 ENIs ( 彈性網路介面 ) VPC，這是橫向擴展期間所需的。如需詳細資訊，請參閱 [彈性網路介面](#)。
- 如果 提供的容量不足 EC2，ElastiCache Auto Scaling 將無法擴展，且會在容量可用時延遲。
- ElastiCache ( Redis OSS ) 在縮減期間 Auto Scaling 不會移除項目大小大於 256 MB 後序列化的插槽碎片。
- 進行規模縮減期間，如果產生的碎片組態上的可用記憶體不足，就不會移除碎片。

## 新增擴展原則

您可以使用 新增擴展政策 AWS Management Console。

若要將 Auto Scaling 政策新增至 ElastiCache 具有 Valkey 或 Redis OSS 叢集的

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇要新增政策至其中的叢集 (請選擇叢集名稱而非其左側的按鈕)。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
5. 選擇 Dynamic scaling (動態擴展)。
6. 在 Policy Name (政策名稱) 輸入政策名稱。
7. 在 Scalable Dimension (可擴展的維度) 選擇 shards (碎片)。
8. 目標指標請選擇下列其中一個：
  - 根據平均 CPU 使用率建立政策的主要 CPU 使用率。
  - Memory (記憶體)，用於根據平均資料庫記憶體建立政策。
  - 根據平均資料庫容量使用情況建立政策的容量。容量指標包括資料分層執行個體的記憶體和 SSD 使用率，以及所有其他執行個體類型的記憶體使用率。
9. 針對目標值，選擇大於或等於 35 且小於或等於 70 的值。自動擴展會在 ElastiCache 碎片中維護所選目標指標的此值：
  - 主要 CPU 使用率：維持主要節點 EngineCPUUtilization 指標的目標值。
  - 記憶體：維護 DatabaseMemoryUsageCountedForEvictPercentage 指標結果的目標值

- 維護 DatabaseCapacityUsageCountedForEvictPercentage 指標結果目標值的容量，系統會新增或移除叢集碎片，讓指標接近指定的值。
10. (選用) 主控台不支援規模縮減或水平擴展冷卻時間。使用 AWS CLI 修改冷卻時間值。
  11. 對於最小容量，輸入 ElastiCache Auto Scaling 政策需要維護的碎片數量下限。
  12. 針對最大容量，輸入 ElastiCache Auto Scaling 政策需要維護的最大碎片數量。此值必須等於或大於 250。
  13. 選擇 Create (建立)。

### 註冊可擴展的目標

在將 Auto Scaling 與 ElastiCache 具有 Valkey 或 Redis OSS 叢集的 搭配使用之前，您可以透過 ElastiCache 自動擴展註冊叢集。您這樣做是為了定義要套用至該 cluster. ElastiCache auto scaling 的擴展維度和限制，並沿著 `elasticache:replication-group:NodeGroups` 可擴展維度動態擴展叢集，其代表叢集碎片的數量。

### 使用 AWS CLI

若要 ElastiCache 向 Valkey 或 Redis OSS 叢集註冊您的，請使用具有下列參數的 [register-scalable-target](#) 命令：

- `--service-namespace` – 將此值設定為 `elasticache`
- `--resource-id` – 叢集的資源識別碼。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--scalable-dimension` – 將此值設定為 `elasticache:replication-group:NodeGroups`。
- `--max-capacity` – 要透過 ElastiCache 自動擴展管理的碎片數量上限。如需了解有關 `--min-capacity`、`--max-capacity` 以及叢集中碎片數目之間的關係資訊，請參閱「[容量下限和上限](#)」。
- `--min-capacity` – 要透過 ElastiCache 自動擴展管理的碎片數量下限。如需了解有關 `--min-capacity`、`--max-capacity` 以及叢集中碎片數目之間的關係資訊，請參閱「[容量下限和上限](#)」。

## Example

在下列範例中，您會 ElastiCache 向名為的 Valkey 或 Redis OSS叢集註冊 myscalablecluster。註冊中表明應該動態擴展叢集，使其具有 1 到 10 個碎片。

若為 Linux、macOS 或 Unix：

```
aws application-autoscaling register-scalable-target \
 --service-namespace elasticache \
 --resource-id replication-group/myscalablecluster \
 --scalable-dimension elasticache:replication-group:NodeGroups \
 --min-capacity 1 \
 --max-capacity 10 \

```

針對 Windows：

```
aws application-autoscaling register-scalable-target ^\
 --service-namespace elasticache ^\
 --resource-id replication-group/myscalablecluster ^\
 --scalable-dimension elasticache:replication-group:NodeGroups ^\
 --min-capacity 1 ^\
 --max-capacity 10 ^\

```

## 使用 API

若要註冊 ElastiCache 叢集，請使用 [register-scalable-target](#) 命令搭配下列參數：

- ServiceNamespace – 將此值設定為 elasticache。
- ResourceID – ElastiCache 叢集的資源識別符。對於此參數，資源類型為 ReplicationGroup，唯一識別符為叢集的名稱，例如 replication-group/myscalablecluster。
- ScalableDimension – 將此值設定為 elasticache:replication-group:NodeGroups。
- MinCapacity – 要透過 ElastiCache 自動擴展管理的碎片數量下限。如需了解 --min-capacity、--max-capacity 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。
- MaxCapacity – 要透過 ElastiCache 自動擴展管理的碎片數量上限。如需了解 --min-capacity、--max-capacity 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。

## Example

在下列範例中，您會 ElastiCache 向以 Application Auto Scaling 命名myscalablecluster的 Valkey 或 Redis OSS叢集註冊 API。此註冊中表明應該動態擴展叢集，使其具有 1 到 5 個複本。

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:NodeGroups",
 "MinCapacity": 1,
 "MaxCapacity": 5
}
```

## 定義擴展政策

目標追蹤擴展政策組態由JSON區塊表示，其中定義指標和目標值。您可以將擴展政策組態儲存為文字檔案中的JSON區塊。叫用 AWS CLI 或 Application Auto Scaling 時，請使用該文字檔案API。如需政策組態語法的詳細資訊，請參閱 Application Auto Scaling API 參考[TargetTrackingScalingPolicyConfiguration](#)中的。

您可使用下列選項來定義目標追蹤擴展政策的組態：

### 主題

- [使用預先定義的指標](#)
- [使用自訂的指標](#)
- [使用冷卻時間](#)

### 使用預先定義的指標

透過使用預先定義的指標，您可以快速為 ElastiCache 具有 Valkey 或 Redis OSS叢集的 定義目標追蹤擴展政策，該叢集可與 ElastiCache ( Redis OSS ) Auto Scaling 中的目標追蹤搭配使用。

目前，在 NodeGroup Auto Scaling 中 ElastiCache 支援下列預先定義的指標：

- ElastiCachePrimaryEngineCPUUtilization – 叢集中所有主要節點的 EngineCPUUtilization CloudWatch 中指標的平均值。

- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage` – 叢集 CloudWatch 中所有主要節點的 `DatabaseMemoryUsageCountedForEvictPercentage` 中指標的平均值。
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` – 叢集 CloudWatch 中所有主要節點的 `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` 中指標的平均值。

如需 `EngineCPUUtilization`、`DatabaseMemoryUsageCountedForEvictPercentage` 和 `DatabaseCapacityUsageCountedForEvictPercentage` 指標的詳細資訊，請參閱 [監控與 CloudWatch 指標搭配使用](#)。若要在您的規模調整政策中使用預先定義的指標，請為規模調整政策建立目標追蹤組態設定。此組態必須包含預先定義指標 `PredefinedMetricSpecification` 的，以及該指標 `TargetValue` 目標值的。

### Example

下列範例說明 ElastiCache 使用 Valkey 或 Redis OSS 叢集進行目標追蹤擴展的典型政策組態。在此組態中，`ElastiCachePrimaryEngineCPUUtilization` 預先定義的指標用於根據叢集中所有主要節點的平均 CPU 使用率 40% 來調整叢集。

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 }
}
```

### 使用自訂的指標

使用自訂的指標，您可以定義目標追蹤規模調整政策來滿足您的自訂需求。您可以根據隨擴展比例變化的任何 ElastiCache 指標來定義自訂指標。並非所有 ElastiCache 指標都適用於目標追蹤。指標必須是有效的使用率指標，而且能夠表示執行個體的忙碌程度。指標的值必須根據叢集中碎片的數量依比例增加或減少。若要使用指標資料來依比例水平擴展或縮減碎片的數量，這樣子成比例的增加或減少是必要的。

### Example

下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，自訂指標會根據名為 `my-db-cluster` 的叢集中所有碎片的平均 CPU 使用率 50% 來調整 ElastiCache ( Redis OSS ) 叢集 `my-db-cluster`。

```
{
```



```
"TargetValue": 50,
"CustomizedMetricSpecification":
{
 "MetricName": "EngineCPUUtilization",
 "Namespace": "AWS/ElastiCache",
 "Dimensions": [
 {
 "Name": "RelicationGroup","Value": "my-db-cluster"
 },
 {
 "Name": "Role","Value": "PRIMARY"
 }
],
 "Statistic": "Average",
 "Unit": "Percent"
}
}
```

## 使用冷卻時間

您可以指定一個值 (單位為秒)，讓 `ScaleOutCooldown` 新增冷卻時間以便水平擴展您的叢集。同樣的，您可以指定一個值 (單位為秒)，讓 `ScaleInCooldown` 新增冷卻時間以便為您的叢集縮減規模。如需詳細資訊，請參閱 [Application Auto Scaling API 參考 `TargetTrackingScalingPolicyConfiguration`](#) 中的。

下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，`ElastiCachePrimaryEngineCPUUtilization` 預先定義的指標用於根據該叢集中所有主要節點的平均CPU使用率 40% 來調整 ElastiCache (RedisOSS) 叢集。這個組態設定分別提供了 10 分鐘的規模縮減冷卻時間，和 5 分鐘的橫向擴展冷卻時間。

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 },
 "ScaleInCooldown": 600,
 "ScaleOutCooldown": 300
}
```



## 停用規模縮減活動

您可以停用縮減活動，防止叢集中的目標追蹤擴展政策組態擴展。停用規模縮減的動作，可防止規模調整政策刪除碎片，同時讓規模調整政策仍然能視需要建立碎片。

您可以為 `DisableScaleIn` 指定布林值，以啟用或停用叢集的規模縮減活動。如需詳細資訊，請參閱 Application Auto Scaling API 參考 [TargetTrackingScalingPolicyConfiguration](#) 中的。

下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，`ElastiCachePrimaryEngineCPUUtilization` 預先定義的指標會根據該 OSS 叢集中所有主要節點的平均 CPU 使用率 40%，調整 ElastiCache 具有 Valkey 或 Redis 叢集的。此組態設定停用了規模調整政策的規模縮減動作。

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 },
 "DisableScaleIn": true
}
```

## 套用擴展原則

ElastiCache 使用 Valkey 或 Redis OSS 自動擴展註冊叢集並定義擴展政策後，您可以將擴展政策套用至已註冊叢集。若要將擴展政策套用至 ElastiCache (Redis OSS) 叢集，您可以使用 AWS CLI 或 Application Auto Scaling API。

### 使用 套用擴展政策 AWS CLI

若要 ElastiCache 使用 Valkey 或 Redis OSS 叢集將擴展政策套用至您的，請使用 [put-scaling-policy](#) 命令搭配下列參數：

- `--policy-name` - 規模調整政策的名稱。
- `--policy-type` - 將此值設為 `TargetTrackingScaling`。
- `--resource-id` - 資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設為 `elasticache:replication-group:NodeGroups`。

- `--target-tracking-scaling-policy-configuration` – 用於叢集的目標追蹤擴展政策組態。

在下列範例中，您會將名為 `myscalablepolicy` 的目標追蹤擴展政策套用到 ElastiCache 使用 ElastiCache 自動擴展名為 Valkey 或 Redis OSS叢集 `myscalablecluster` 的。做法是使用儲存於 `config.json` 檔案中的政策組態設定。

針對 Linux、macOS 或 Unix：

```
aws application-autoscaling put-scaling-policy \
 --policy-name myscalablepolicy \
 --policy-type TargetTrackingScaling \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:NodeGroups \
 --target-tracking-scaling-policy-configuration file://config.json
```

針對 Windows：

```
aws application-autoscaling put-scaling-policy ^\
 --policy-name myscalablepolicy ^\
 --policy-type TargetTrackingScaling ^\
 --resource-id replication-group/myscalablecluster ^\
 --service-namespace elasticache ^\
 --scalable-dimension elasticache:replication-group:NodeGroups ^\
 --target-tracking-scaling-policy-configuration file://config.json
```

## 使用 套用擴展政策 API

若要 ElastiCache 使用 Valkey 或 Redis OSS叢集將擴展政策套用至您的 `ReplicationGroup`，請使用具有下列參數的 [PutScalingPolicy](#) AWS CLI 命令：

- `--policy-name` - 規模調整政策的名稱。
- `--resource-id` – 資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設為 `elasticache:replication-group:NodeGroups`。
- `--target-tracking-scaling-policy-configuration` – 用於叢集的目標追蹤擴展政策組態。

在下列範例中，您會將名為 `myscalablepolicy` 的目標追蹤擴展政策套用到 ElastiCache 使用 ElastiCache 自動擴展名為 Valkey 或 Redis OSS 叢集 `myscalablecluster` 的。您使用的政策組態設定，是以 `ElastiCachePrimaryEngineCPUUtilization` 這個預先定義的指標為根據。

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "PolicyName": "myscalablepolicy",
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:NodeGroups",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 }
 }
}
```

## 編輯擴展原則

您可以使用 AWS Management Console、AWS CLI 或 Application Auto Scaling 來編輯擴展政策 API。

使用 編輯擴展政策 AWS Management Console

使用 ElastiCache Valkey 或 Redis OSS 叢集編輯 的 Auto Scaling 政策

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇適當的引擎。
3. 選擇要新增政策至其中的叢集 (請選擇叢集名稱而非其左側的按鈕)。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。

5. 根據 Scaling policies (擴展政策)，找到您要變更的自動擴展政策，並選擇 Modify (修改)。
6. 對政策進行必要的變更。
7. 選擇 Modify (修改)。

### 使用 AWS CLI 和 編輯擴展政策 API

您可以使用 AWS CLI 或 Application Auto Scaling API來編輯擴展政策，方法與套用擴展政策相同：

- 使用時 AWS CLI，請在 `--policy-name` 參數中指定您要編輯的政策名稱。針對您想要變更的參數指定新的參數值。
- 使用 Application Auto Scaling 時 API，請在 `PolicyName` 參數中指定您要編輯的政策名稱。針對您想要變更的參數指定新的參數值。

如需詳細資訊，請參閱[套用擴展原則](#)。

### 刪除擴展原則

您可以使用 AWS Management Console、AWS CLI或 Application Auto Scaling 刪除擴展政策API。

### 使用 刪除擴展政策 AWS Management Console

#### 刪除 ElastiCache ( Redis OSS ) 叢集的 Auto Scaling 政策

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇要您想編輯其擴展政策的叢集 (請選擇叢集名稱而非其左側的按鈕)。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
5. 在 Scaling Policies (擴展政策) 區段中，選擇自動擴展政策，然後選擇 Delete (刪除)。

### 使用 刪除擴展政策 AWS CLI

若要 ElastiCache 使用 Valkey 或 Redis OSS叢集將擴展政策刪除到，請使用具有下列參數的 [delete-scaling-policy](#) AWS CLI 命令：

- `--policy-name` - 規模調整政策的名稱。

- `--resource-id` – 資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設為 `elasticache:replication-group:NodeGroups`。

在下列範例中，您會 `myscalablepolicy` 從名為 `myscalablecluster` 的叢集中刪除名為 `myscalablecluster` 的目標追蹤擴展政策。

若為 Linux、macOS 或 Unix：

```
aws application-autoscaling delete-scaling-policy \
 --policy-name myscalablepolicy \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:NodeGroups
```

針對 Windows：

```
aws application-autoscaling delete-scaling-policy ^\
 --policy-name myscalablepolicy ^\
 --resource-id replication-group/myscalablecluster ^\
 --service-namespace elasticache ^\
 --scalable-dimension elasticache:replication-group:NodeGroups
```

使用 刪除擴展政策 API

若要 ElastiCache 使用 Valkey 或 Redis OSS 叢集將擴展政策刪除到 `myscalablecluster`，請使用具有下列參數的 [DeleteScalingPolicy](#) AWS CLI 命令：

- `--policy-name` - 規模調整政策的名稱。
- `--resource-id` – 資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設為 `elasticache:replication-group:NodeGroups`。

在下列範例中，您會 `myscalablepolicy` 從名為 `myscalablecluster` 的叢集中刪除名為 `myscalablecluster` 的目標追蹤擴展政策。

```

POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "PolicyName": "myscalablepolicy",
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:NodeGroups"
}

```

## AWS CloudFormation 用於 Auto Scaling 政策

此程式碼片段說明如何建立目標追蹤政策，並使用 [AWS :: ElastiCache :: ReplicationGroup](#) 資源將其套用至 [AWS :: ApplicationAutoScaling :: 資源ScalableTarget](#)。其使用 [Fn::Join](#) 和 [Ref](#) 內部函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::ElastiCache::ReplicationGroup](#) 資源的邏輯名稱。

```

ScalingTarget:
 Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
 Properties:
 MaxCapacity: 3
 MinCapacity: 1
 ResourceId: !Sub replication-group/${logicalName}
 ScalableDimension: 'elasticache:replication-group:NodeGroups'
 ServiceNamespace: elasticache
 RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
 Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
 Properties:
 ScalingTargetId: !Ref ScalingTarget
 ServiceNamespace: elasticache
 PolicyName: testpolicy
 PolicyType: TargetTrackingScaling

```

```
ScalableDimension: 'elasticache:replication-group:NodeGroups'
TargetTrackingScalingPolicyConfiguration:
 PredefinedMetricSpecification:
 PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
 TargetValue: 40
```

## 排程擴展

按照排程進行擴展讓您能夠因應可預測的需求變化，據以擴展您的應用程式。若要使用排程擴展，您可以建立排程動作，該動作 ElastiCache 會與 Valkey 或 Redis 一起在特定時間 OSS 執行擴展活動。建立排程動作時，您可以指定現有的 ElastiCache (Redis OSS) 叢集，以及應進行擴展活動的時間、最小容量和最大容量。您可以建立僅擴展一次或依週期性排程擴展的排程動作。

您只能為已存在的 ElastiCache (Redis OSS) 叢集建立排程動作。您無法在建立叢集的同時建立排程動作。

如需排程動作建立、管理和刪除作業相關術語的詳細資訊，請參閱[排程動作建立、管理和刪除作業常用的命令](#)

若要建立週期性排程：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇您要新增政策的叢集。
4. 從 Actions (動作) 下拉式選單中選擇 Manage Auto Scaling policies (管理 Auto Scaling 政策)。
5. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
6. 在 Auto scaling policies (Auto Scaling 政策) 區段中，會出現 Add Scaling policy (新增規模調整政策) 對話方塊。選擇 Scheduled scaling (排程規模調整)。
7. 在 Policy Name (政策名稱) 輸入政策的名稱。
8. 在 Scalable Dimension (可擴展的維度) 選擇 Shards (碎片)。
9. 在 Target Shards (目標碎片) 選擇值。
10. 在 Recurrence (重複) 選擇 Recurring (週期性)。
11. 在 Frequency (頻率) 選擇各自的值。
12. 在 Start Date (開始日期) 和 Start time (開始時間) 選擇政策生效的時間。
13. 選擇 Add Policy (新增政策)。

若要建立一次性排程動作：

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇您要新增政策的叢集。
4. 從 Actions (動作) 下拉式選單中選擇 Manage Auto Scaling policies (管理 Auto Scaling 政策)。
5. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
6. 在 Auto scaling policies (Auto Scaling 政策) 區段中，會出現 Add Scaling policy (新增規模調整政策) 對話方塊。選擇 Scheduled scaling (排程規模調整)。
7. 在 Policy Name (政策名稱) 輸入政策的名稱。
8. 在 Scalable Dimension (可擴展的維度) 選擇 Shards (碎片)。
9. 在 Target Shards (目標碎片) 選擇值。
10. 在 Recurrence (重複)，選擇 Once (一次)。
11. 在 Start Date (開始日期) 和 Start time (開始時間) 選擇政策生效的時間。
12. 在 End Date (結束日期) 選擇政策效用結束的日期。
13. 選擇 Add Policy (新增政策)。

刪除排程動作

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇您要新增政策的叢集。
4. 從 Actions (動作) 下拉式選單中選擇 Manage Auto Scaling policies (管理 Auto Scaling 政策)。
5. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
6. 在 Auto Scaling policies (Auto Scaling 政策) 區段中，選擇 Auto Scaling 政策，然後從 Actions (動作) 對話中選擇 Delete (刪除)。

使用 AWS CLI 管理排定的擴展

使用下列應用程式自動擴展 APIs：

- [put-scheduled-action](#)



- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

使用 AWS CloudFormation 建立排程動作

此程式碼片段說明如何建立目標追蹤政策，並使用 [AWS::ElastiCache::ReplicationGroup](#) 資源將其套用至 [AWS::ApplicationAutoScaling::ScalableTarget](#)。其使用 [Fn::Join](#) 和 [Ref](#) 內部函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::ElastiCache::ReplicationGroup](#) 資源的邏輯名稱。

```
ScalingTarget:
 Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
 Properties:
 MaxCapacity: 3
 MinCapacity: 1
 ResourceId: !Sub replication-group/${logicalName}
 ScalableDimension: 'elasticache:replication-group:NodeGroups'
 ServiceNamespace: elasticache
 RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
 ScheduledActions:
 - EndTime: '2020-12-31T12:00:00.000Z'
 ScalableTargetAction:
 MaxCapacity: '5'
 MinCapacity: '2'
 ScheduledActionName: First
 Schedule: 'cron(0 18 * * ? *)'
```

## 搭配複本使用自動擴展

ElastiCache 複寫群組可以設定一或多個快取，以做為單一邏輯節點。

以下提供目標追蹤和排程政策的詳細資訊，以及如何使用 AWS Management Console AWS CLI 和 套用這些政策 APIs。

### 目標追蹤擴展政策

使用目標追蹤擴展政策，您可以選取指標及設定目標值。ElastiCache 使用 Valkey 或 Redis OSS AutoScaling 建立和管理觸發擴展政策的 CloudWatch 警示，並根據指標和目標值計算擴展調整。規模

調整政策會視需要平均在所有碎片間新增或移除複本，以讓指標保持在等於或接近指定目標值。除了讓指標保持在接近目標值之外，目標追蹤規模調整政策也會配合指標中，因為負載模式波動所造成的波動調整，並將機群容量中的快速波動降到最低。

## 複本的 Auto Scaling 條件

Auto Scaling 政策會為您的叢集定義下列預先定義的指標：

`ElastiCacheReplicaEngineCPUUtilization`：ElastiCache 用於觸發自動擴展操作的所有複本彙總的AVG引擎CPU使用率閾值。您可以將使用率目標設定為 35% 到 70% 之間。

當服務偵測到您的`ElastiCacheReplicaEngineCPUUtilization`指標等於或大於目標設定時，它會自動增加碎片上的複本。ElastiCache 使用 Valkey 或 Redis 會將叢集複本OSS擴展為等於兩個數字中較大者的數量：與目標的百分比變化和一個複本。對於縮減，ElastiCache 使用 Valkey 或 Redis OSS不會自動縮減，除非整體指標值低於您定義目標的 75%。

以下提供水平擴展的範例，如果有 5 個碎片，各 1 個複本：

如果您的目標違規率為 30%，ElastiCache 則 Valkey 或 Redis 會跨所有碎片OSS縮減 1 個複本（ $\max(0.3, \text{預設 } 1)$ ）。這會產生 5 個碎片，每個碎片有 2 個複本，

對於縮減範例，如果您已選取目標值 60%，ElastiCache 則使用 Valkey 或 Redis 時，在指標小於或等於 45%（低於目標 60% 的 25%）之前OSS，不會自動縮減。

## Auto Scaling 考量事項

請謹記以下幾點考量：

- 目標追蹤擴展政策假設在指定的指標超過目標值時，應執行向外擴展。當指定的指標低於目標值時，您無法使用目標追蹤擴展政策來擴展。ElastiCache 使用 Valkey 或 Redis 會將複本OSS擴展至叢集中所有碎片中現有複本的最大值（與目標的偏差為 %，預設值為 1）。
- 所指定指標的資料不足時，目標追蹤擴展政策不會執行擴展。政策不會執行向內擴展，因為向內擴展不會將資料不足解釋為低使用率。
- 您可能會看到目標值與實際指標資料點之間有些差距。這是因為 ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 時，在決定要新增或移除的容量時，一律會四捨五入或向下捨入，以保守的方式運作。這樣可防止新增不足的容量，或移除過多的容量。
- 為了確保應用程式的可用性，此服務可以配合指標依比例快速擴展規模，但是以更漸進的方式在叢集中碎片間最多縮減規模 1 個複本。
- 您可以針對 ElastiCache 具有 Valkey 或 Redis OSS叢集的擁有多個目標追蹤擴展政策，前提是每個政策都使用不同的指標。Auto Scaling 的目的在於一律優先考慮可用性，因此其行為會因目標追蹤政

策是否已準備好橫向擴展或橫向擴展而有所不同。如果任何目標追蹤政策已準備好向外擴展，它就會將服務向外擴展，但只有在所有目標追蹤政策 (已啟用向內擴展部分) 都已準備好要向內擴展時才會向內擴展。

- 請勿編輯或刪除 ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 管理的目標追蹤擴展政策的 CloudWatch 警示。當您刪除擴展政策或刪除叢集時，Auto Scaling 會自動刪除警示。
- ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 不會阻止您在碎片之間手動修改複本。這些手動調整不會影響連接到擴展政策的任何現有 CloudWatch 警示，但可能會影響可能觸發這些 CloudWatch 警示的指標。
- 這些由 Auto Scaling 管理的 CloudWatch 警示是透過叢集中所有碎片的AVG指標來定義。因此，使用熱碎片可能會導致以下任一情況：
  - 因幾個觸發 CloudWatch 警示的熱碎片載入而不需要擴展
  - 由於AVG彙總了影響警示不違反的所有碎片，因此在需要時不會擴展。
- ElastiCache 對於每個叢集的節點，使用 Valkey 或 Redis OSS 預設限制仍然適用。因此，當選擇使用 Auto Scaling 且您希望節點數上限超過預設配額時，請至 [AWS 服務配額](#) 請求增加配額，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。
- 請確定您的 中有足夠的 ENIs ( 彈性網路介面 ) VPC，這是橫向擴展期間所需的。如需詳細資訊，請參閱[彈性網路介面](#)。
- 如果 提供的容量不足EC2，ElastiCache 則使用 Valkey 或 Redis OSS Auto Scaling 時，在容量可用或將叢集手動修改為具有足夠容量的執行個體類型之前，不會橫向擴展。
- ElastiCache 使用 Valkey 或 Redis OSS Auto Scaling 不支援以ReservedMemoryPercent不到 25% 的叢集擴展複本。如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

## 新增擴展原則

您可以使用 新增擴展政策 AWS Management Console。

使用 新增擴展政策 AWS Management Console

若要 ElastiCache 使用 Valkey 或 Redis 將自動擴展政策新增至 OSS

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 選擇要新增政策至其中的叢集 (請選擇叢集名稱而非其左側的按鈕)。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。

5. 選擇 Dynamic scaling (動態擴展)。
6. 根據 Scaling policies (擴展政策)，選擇 Add dynamic scaling (新增動態擴展)。
7. 在 Policy Name (政策名稱) 輸入政策的名稱。
8. 在 Scalable Dimension (可擴展的維度)，從對話方塊中選取 Replicas (複本)。
9. 針對目標值，輸入您要在 ElastiCache 複本上維護的平均CPU使用率百分比。此值必須  $\geq 35$  且  $\leq 70$ 。系統會新增或移除叢集複本，讓指標接近指定的值。
10. (選用) 主控台不支援規模縮減或水平擴展冷卻時間。使用 AWS CLI 修改冷卻值。
11. 對於最小容量，輸入 ElastiCache 具有 Valkey 或 Redis OSS Auto Scaling 政策的 需要維護的複本數量下限。
12. 對於最大容量，輸入需要維護 ElastiCache 的具有 Valkey 或 Redis OSS Auto Scaling 政策的複本數量上限。此值必須  $\geq 5$ 。
13. 選擇 Create (建立)。

## 註冊可擴展的目標

您可以根據預先定義的指標或自訂指標，套用規模調整政策。若要這麼做，您可以使用 AWS CLI 或 Application Auto Scaling API。第一步是使用 Auto Scaling ElastiCache 向 Valkey 或 Redis OSS 複寫群組註冊您的。Auto Scaling

您必須先 ElastiCache 使用 Valkey 或 Redis ElastiCache 自動擴展向 註冊叢集，才能搭配叢集使用 OSS 自動擴展。您可以這樣做來定義要套用至該叢集的擴展維度和限制。ElastiCache 使用 Valkey 或 Redis OSS 自動擴展會沿著 `elasticache:replication-group:Replicas` 可擴展維度動態擴展叢集，這代表每個碎片的叢集複本數量。

## 使用 CLI

若要註冊 ElastiCache 叢集，請使用 [register-scalable-target](#) 命令搭配下列參數：

- `--service-namespace` - 將此值設為 `elasticache`。
- `--resource-id` - ElastiCache 叢集的資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--scalable-dimension` - 將此值設定為 `elasticache:replication-group:Replicas`。
- `--min-capacity` - 使用 ElastiCache Valkey 或 Redis OSS 自動擴展管理的複本數目下限。如需了解 `--min-capacity`、`--max-capacity` 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。
- `--max-capacity` - 使用 ElastiCache Valkey 或 Redis OSS 自動擴展管理的複本數量上限。如需了解 `--min-capacity`、`--max-capacity` 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。

## Example

在下列範例中，您會 ElastiCache 向名為的 Valkey 或 Redis OSS叢集註冊 myscalablecluster。註冊中表明應該動態擴展叢集，使其具有 1 到 5 個複本。

若為 Linux、macOS 或 Unix：

```
aws application-autoscaling register-scalable-target \
 --service-namespace elasticache \
 --resource-id replication-group/myscalablecluster \
 --scalable-dimension elasticache:replication-group:Replicas \
 --min-capacity 1 \
 --max-capacity 5 \

```

針對 Windows：

```
aws application-autoscaling register-scalable-target ^\
 --service-namespace elasticache ^\
 --resource-id replication-group/myscalablecluster ^\
 --scalable-dimension elasticache:replication-group:Replicas ^\
 --min-capacity 1 ^\
 --max-capacity 5 ^\

```

## 使用 API

若要註冊 ElastiCache 叢集，請使用 [register-scalable-target](#) 命令搭配下列參數：

- ServiceNamespace – 將此值設定為 elasticache。
- ResourceID – ElastiCache 叢集的資源識別碼。對於此參數，資源類型為 ReplicationGroup，唯一識別符為叢集的名稱，例如 replication-group/myscalablecluster。
- ScalableDimension – 將此值設定為 elasticache:replication-group:Replicas。
- MinCapacity – 使用 ElastiCache Valkey 或 Redis OSS自動擴展管理的複本數目下限。如需了解 --min-capacity、--max-capacity 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。
- MaxCapacity – 使用 ElastiCache Valkey 或 Redis OSS自動擴展管理的複本數量上限。如需了解 --min-capacity、--max-capacity 以及叢集中複本數之間的關係，請參閱「[容量下限和上限](#)」。

## Example

在下列範例中，您可以使用 Application Auto Scaling 註冊名為 myscalablecluster的叢集API。此註冊中表明應該動態擴展叢集，使其具有 1 到 5 個複本。

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:Replicas",
 "MinCapacity": 1,
 "MaxCapacity": 5
}
```

## 定義擴展政策

目標追蹤擴展政策組態由JSON區塊表示，其中定義指標和目標值。您可以將擴展政策組態儲存為文字檔案中的JSON區塊。叫用 AWS CLI 或 Application Auto Scaling 時，請使用該文字檔案API。如需政策組態語法的詳細資訊，請參閱 Application Auto Scaling 參考 [TargetTrackingScalingPolicyConfiguration](#) 中的 [Auto Scaling API](#)

您可使用下列選項來定義目標追蹤擴展政策的組態：

### 主題

- [使用預先定義的指標](#)
- [編輯擴展原則](#)
- [刪除擴展原則](#)
- [AWS CloudFormation 用於 Auto Scaling 政策](#)
- [排程擴展](#)

### 使用預先定義的指標

目標追蹤擴展政策組態由JSON區塊表示，其中定義指標和目標值。您可以將擴展政策組態儲存為文字檔案中的JSON區塊。叫用 AWS CLI 或 Application Auto Scaling 時，請使用該文字檔案API。如需政策組態語法的詳細資訊，請參閱 Application Auto Scaling 參考 [TargetTrackingScalingPolicyConfiguration](#) 中的 [Auto Scaling API](#)



您可使用下列選項來定義目標追蹤擴展政策的組態：

## 主題

- [使用預先定義的指標](#)
- [使用自訂的指標](#)
- [使用冷卻時間](#)
- [停用規模縮減活動](#)
- [ElastiCache 使用 Valkey 或 Redis OSS 叢集將擴展政策套用至](#)

## 使用預先定義的指標

透過使用預先定義的指標，您可以快速為 ElastiCache 具有 Valkey 或 Redis OSS 叢集的 定義目標追蹤擴展政策，該叢集可與 ElastiCache 具有 Valkey 或 Redis OSS Auto Scaling 的目標追蹤搭配使用。目前，在 ElastiCache Replicas Auto Scaling 中 ElastiCache 支援下列預先定義的指標：

ElastiCacheReplicaEngineCPUUtilization – 叢集 CloudWatch 中所有複本中 EngineCPUUtilization 指標的平均值。您可以在 CloudWatch ElastiCache ReplicationGroupId, Role 中找到必要 ReplicationGroupId 和角色複本的彙總指標值。

若要在您的規模調整政策中使用預先定義的指標，請為規模調整政策建立目標追蹤組態設定。此組態設定必須加入用於預先定義指標的 PredefinedMetricSpecification，以及用於該指標目標值的 TargetValue。

## 使用自訂的指標

使用自訂的指標，您可以定義目標追蹤規模調整政策來滿足您的自訂需求。您可以根據任何 ElastiCache 具有 Valkey 或 Redis 指標的 來定義自訂指標，該OSS指標會隨著擴展比例而變更。並非所有 ElastiCache 指標都適用於目標追蹤。指標必須是有效的使用率指標，而且能夠表示執行個體的忙碌程度。指標的值必須根據叢集中複本的數量依比例增加或減少。若要使用指標資料來依比例增加或減少碎片數量，這樣子成比例的增加或減少是必要的。

## Example

下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，自訂指標會根據名為 的叢集中所有複本的平均CPU使用率 50% 來調整叢集my-db-cluster。

```
{"TargetValue": 50,
```

```
"CustomizedMetricSpecification":
{
 "MetricName": "EngineCPUUtilization",
 "Namespace": "AWS/ElastiCache",
 "Dimensions": [
 { "Name": "RelocationGroup", "Value": "my-db-cluster" },
 { "Name": "Role", "Value": "REPLICA" }
],
 "Statistic": "Average",
 "Unit": "Percent"
}
}
```

## 使用冷卻時間

您可以指定一個值 (單位為秒)，讓 `ScaleOutCooldown` 新增冷卻時間以便水平擴展您的叢集。同樣的，您可以指定一個值 (單位為秒)，讓 `ScaleInCooldown` 新增冷卻時間以便為您的叢集縮減規模。如需 `ScaleInCooldown` 和 `ScaleOutCooldown` 的詳細資訊，請參閱 [Application Auto Scaling 參考 TargetTrackingScalingPolicyConfiguration](#) 中的。Auto Scaling API 下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，`ElastiCacheReplicaEngineCPUUtilization` 預先定義的指標用於根據該叢集中所有複本的平均 CPU 使用率 40% 來調整叢集。這個組態設定分別提供了 10 分鐘的規模縮減冷卻時間，和 5 分鐘的規模擴展冷卻時間。

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
 },
 "ScaleInCooldown": 600,
 "ScaleOutCooldown": 300
}
```

## 停用規模縮減活動

您可以透過停用縮減活動，防止目標追蹤擴展政策組態 `ElastiCache` 在中使用 `Valkey` 或 `Redis OSS` 叢集進行擴展。停用規模縮減的動作，可防止規模調整政策刪除複本，同時讓規模調整政策仍然能視需要新增複本。

您可以為 `DisableScaleIn` 指定布林值，以啟用或停用叢集的規模縮減活動。如需的詳細資訊 `DisableScaleIn`，請參閱 [Application Auto Scaling 參考 TargetTrackingScalingPolicyConfiguration](#) 中的。Auto Scaling API



## Example

下列的範例描述規模調整政策的目標追蹤組態設定。在此組態中，ElastiCacheReplicaEngineCPUUtilization預先定義的指標會根據該叢集中所有複本的平均CPU使用率 40% 來調整叢集。此組態設定停用了規模調整政策的規模縮減動作。

```
{"TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"},
 "DisableScaleIn": true
}
```

ElastiCache 使用 Valkey 或 Redis OSS叢集將擴展政策套用至

ElastiCache 使用 Valkey 或 Redis OSS自動擴展和定義擴展政策註冊叢集之後，您可以將擴展政策套用至已註冊叢集。若要將擴展政策套用至 ElastiCache 具有 Valkey 或 Redis OSS叢集的，您可以使用 AWS CLI 或 Application Auto Scaling API。

### 使用 AWS CLI

若要 ElastiCache 使用 Valkey 或 Redis OSS叢集將擴展政策套用至您的，請使用 [put-scaling-policy](#) 命令搭配下列參數：

- `--policy-name` - 規模調整政策的名稱。
- `--policy-type` - 將此值設為 `TargetTrackingScaling`。
- `--resource-id` - 叢集的資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設定為 `elasticache:replication-group:Replicas`。
- `--target-tracking-scaling-policy-configuration` - 用於叢集的目標追蹤擴展政策組態。

## Example

在下列範例中，您將名為的目標追蹤擴展政策套用 `myscalablepolicy` 到 `myscalablecluster` ElastiCache 使用 Valkey 或 Redis OSS自動擴展名為的叢集。做法是使用儲存於 `config.json` 檔案中的政策組態設定。

## 針對 Linux、macOS 或 Unix：

```
aws application-autoscaling put-scaling-policy \
 --policy-name myscalablepolicy \
 --policy-type TargetTrackingScaling \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:Replicas \
 --target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
 },
 "DisableScaleIn": true
}
```

## 針對 Windows：

```
aws application-autoscaling put-scaling-policy ^
 --policy-name myscalablepolicy ^
 --policy-type TargetTrackingScaling ^
 --resource-id replication-group/myscalablecluster ^
 --service-namespace elasticache ^
 --scalable-dimension elasticache:replication-group:Replicas ^
 --target-tracking-scaling-policy-configuration file://config.json
```

## 使用 API

若要使用 Application Auto Scaling 將擴展政策套用至 ElastiCache 具有 Valkey 或 Redis OSS 叢集的 API，請使用 [PutScalingPolicy](#) Application Auto Scaling API 操作搭配下列參數：

- PolicyName – 擴展政策的名稱。
- PolicyType – 將此值設定為 TargetTrackingScaling。
- ResourceID – 叢集的資源識別符。對於此參數，資源類型為 ReplicationGroup，唯一識別符為 ElastiCache (Redis OSS) 叢集的名稱，例如 replication-group/myscalablecluster。
- ServiceNamespace – 將此值設定為 elasticache。

- ScalableDimension – 將此值設定為 elasticache:replication-group:Replicas。
- TargetTrackingScalingPolicyConfiguration – 用於叢集的目標追蹤擴展政策組態。

## Example

在下列範例中，您將名為 的目標追蹤擴展政策套用scalablepolicy到myscalablecluster ElastiCache 使用 Valkey 或 Redis OSS自動擴展名為 的叢集。您使用的政策組態設定，是以 ElastiCacheReplicaEngineCPUUtilization 這個預先定義的指標為根據。

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "PolicyName": "myscalablepolicy",
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:Replicas",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
 }
 }
}
```

## 編輯擴展原則

您可以使用 AWS Management Console、AWS CLI 或 Application Auto Scaling 來編輯擴展政策 API。

使用 編輯擴展政策 AWS Management Console

您只能使用 AWS Management Console 來編輯具有預先定義指標類型的政策

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS
3. 選擇要新增政策至其中的叢集 (請選擇叢集名稱而非其左側的按鈕)。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
5. 根據 Scaling policies (擴展政策)，找到您要變更的自動擴展政策，並選擇 Modify (修改)。
6. 對政策進行必要的變更。
7. 選擇 Modify (修改)。
8. 對政策進行變更。
9. 選擇 Modify (修改)。

使用 AWS CLI 或 Application Auto Scaling 編輯擴展政策 API

您可以使用 AWS CLI 或 Application Auto Scaling API 來編輯擴展政策，方式與套用擴展政策相同：

- 使用 Application Auto Scaling 時 API，請在 PolicyName 參數中指定您要編輯的政策名稱。針對您想要變更的參數指定新的參數值。

如需詳細資訊，請參閱 [ElastiCache 使用 Valkey 或 Redis OSS 叢集將擴展政策套用至](#)。

## 刪除擴展原則

您可以使用 AWS Management Console、AWS CLI 或 Application Auto Scaling 刪除擴展政策 API

使用 刪除擴展政策 AWS Management Console

您只能使用 AWS Management Console 來編輯具有預先定義指標類型的政策

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS
3. 選擇您要刪除其 Auto Scaling 政策的叢集。
4. 選擇 Auto Scaling policies (Auto Scaling 政策) 索引標籤。
5. 在 Scaling Policies (擴展政策) 區段中，選擇自動擴展政策，然後選擇 Delete (刪除)。

使用 AWS CLI 或 Application Auto Scaling 刪除擴展政策 API

您可以使用 AWS CLI 或 Application Auto Scaling 從 ElastiCache 叢集API中刪除擴展政策。

## CLI

若要 ElastiCache 使用 Valkey 或 Redis OSS叢集從 中刪除擴展政策，請使用 [delete-scaling-policy](#) 命令搭配下列參數：

- `--policy-name` - 規模調整政策的名稱。
- `--resource-id` - 叢集的資源識別符。對於此參數，資源類型為 `ReplicationGroup`，唯一識別符為叢集的名稱，例如 `replication-group/myscalablecluster`。
- `--service-namespace` - 將此值設為 `elasticache`。
- `--scalable-dimension` - 將此值設定為 `elasticache:replication-group:Replicas`。

## Example

在下列範例中，您會 `myscalablepolicy` 從 刪除名為 的目標追蹤擴展政策ELC；叢集名為 `myscalablecluster`。

若為 Linux、macOS 或 Unix：

```
aws application-autoscaling delete-scaling-policy \
 --policy-name myscalablepolicy \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:Replicas \

```

針對 Windows：

```
aws application-autoscaling delete-scaling-policy ^\
 --policy-name myscalablepolicy ^\
 --resource-id replication-group/myscalablecluster ^\
 --service-namespace elasticache ^\
 --scalable-dimension elasticache:replication-group:Replicas ^\

```

## API

若要 ElastiCache 使用 Valkey 或 Redis OSS叢集從 中刪除擴展政策，請使用 [DeleteScalingPolicy](#) Application Auto Scaling API操作搭配下列參數：

- PolicyName – 擴展政策的名稱。
- ResourceID – 叢集的資源識別符。對於此參數，資源類型為 ReplicationGroup，唯一識別符為叢集的名稱，例如 replication-group/myscalablecluster。
- ServiceNamespace – 將此值設定為 elasticache。
- ScalableDimension – 將此值設定為 elasticache:replication-group:Replicas。

在下列範例中，您會myscalablepolicy從myscalablecluster使用 Application Auto Scaling 命名的叢集中刪除名為的目標追蹤擴展政策API。

```
POST / HTTP/1.1
>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "PolicyName": "myscalablepolicy",
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalablecluster",
 "ScalableDimension": "elasticache:replication-group:Replicas"
}
```

## AWS CloudFormation 用於 Auto Scaling 政策

此程式碼片段說明如何建立排程動作，並使用 [AWS :: ElastiCache :: ReplicationGroup](#) 資源將其套用至 [AWS :: ApplicationAutoScaling :: 資源ScalableTarget](#)。其使用 [Fn::Join](#) 和 [Ref](#) 內部函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::ElastiCache::ReplicationGroup](#) 資源的邏輯名稱。

```
ScalingTarget:
 Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
 Properties:
 MaxCapacity: 0
 MinCapacity: 0
 ResourceId: !Sub replication-group/${logicalName}
 ScalableDimension: 'elasticache:replication-group:Replicas'
```

```
ServiceNamespace: elasticache
RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
 Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
 Properties:
 ScalingTargetId: !Ref ScalingTarget
 ServiceNamespace: elasticache
 PolicyName: testpolicy
 PolicyType: TargetTrackingScaling
 ScalableDimension: 'elasticache:replication-group:Replicas'
 TargetTrackingScalingPolicyConfiguration:
 PredefinedMetricSpecification:
 PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
 TargetValue: 40
```

## 排程擴展

按照排程進行擴展讓您能夠因應可預測的需求變化，據以擴展您的應用程式。若要使用排程擴展，您可以建立排程動作，該動作 ElastiCache 會與 Valkey 或 Redis 一起在特定時間OSS執行擴展活動。建立排程動作時，您可以指定 ElastiCache 具有 Valkey 或 Redis OSS叢集的現有、何時應進行擴展活動、最小容量和最大容量。您可以建立僅擴展一次或依週期性排程擴展的排程動作。

您只能 ElastiCache 使用已存在的 Valkey 或 Redis OSS叢集建立的排程動作。您無法在建立叢集的同時建立排程動作。

如需排程動作建立、管理和刪除作業相關術語的詳細資訊，請參閱[排程動作建立、管理和刪除作業常用的命令](#)

建立一次性排程動作：

類似於碎片維度。請參閱 [排程擴展](#)。

刪除排程動作

類似於碎片維度。請參閱 [排程擴展](#)。

使用 AWS CLI 管理排定的擴展

使用下列應用程式自動擴展 APIs：

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

使用 AWS CloudFormation 建立 Auto Scaling 政策

此程式碼片段說明如何使用 [AWS :: ElastiCache :: ReplicationGroup](#) 資源建立排程動作，並將其套用於 [AWS :: ApplicationAutoScaling :: ScalableTarget](#) 資源。其使用 [Fn::Join](#) 和 [Ref](#) 內部函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::ElastiCache::ReplicationGroup](#) 資源的邏輯名稱。

```
ScalingTarget:
 Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
 Properties:
 MaxCapacity: 0
 MinCapacity: 0
 ResourceId: !Sub replication-group/${logicalName}
 ScalableDimension: 'elasticache:replication-group:Replicas'
 ServiceNamespace: elasticache
 RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
 ScheduledActions:
 - EndTime: '2020-12-31T12:00:00.000Z'
 ScalableTargetAction:
 MaxCapacity: '5'
 MinCapacity: '2'
 ScheduledActionName: First
 Schedule: 'cron(0 18 * * ? *)'
```

## 修改叢集模式

Valkey 和 Redis OSS 是分散式記憶體內資料庫，支援分割和複寫。ElastiCache Valkey 和 Redis OSS 叢集是分散式實作，允許跨多個節點分割資料。ElastiCache (Redis OSS) 叢集有兩種操作模式：啟用叢集模式 (CME) 和停用叢集模式 (CMD)。在中 CME，Valkey 和 Redis OSS 引擎可用作具有多個碎片和節點的分散式資料庫，而在中 CMD，Valkey 和 Redis OSS 可用作單一節點。

在從 遷移 CMD 至 之前 CME，必須符合下列條件：



**⚠ Important**

叢集模式組態只能從停用叢集模式變更為啟用叢集模式。還原此組態是不可能的。

- 叢集可能只有資料庫 0 中的金鑰。
- 應用程式必須使用能夠使用叢集通訊協定並使用組態端點的 Valkey 或 Redis OSS 用戶端。
- 必須在至少有 1 個複本的叢集上啟用自動容錯移轉。
- 遷移所需的最低引擎版本為 Valkey 7.2 及更高版本，或 Redis OSS 7.0 及更高版本。

若要從 遷移CMD至 CME，必須將叢集模式組態從停用的叢集模式變更為啟用的叢集模式。這是一個兩步驟的程序，可確保遷移程序期間的叢集可用性。

**📘 Note**

您需要提供具有啟用叢集組態的參數群組，也就是說，已啟用叢集的參數設為 yes。如果您使用預設參數群組，ElastiCache (Redis OSS) 會自動挑選具有啟用叢集組態的對應預設參數群組。CMD 叢集啟用叢集no的參數值設定為 。當叢集移至相容模式時，做為修改動作的一部分，啟用叢集的參數值會隨之更新為 yes。

如需詳細資訊，請參閱 [使用參數群組設定引擎 ElastiCache 參數](#)

1. 準備 – 建立測試CME叢集，並確保您的堆疊已準備好使用它。ElastiCache (Redis OSS) 無法驗證您的準備狀態。如需詳細資訊，請參閱 [為 Valkey 或 Redis 建立叢集 OSS](#)。
2. 將現有CMD叢集組態修改為叢集模式相容 – 在此模式下，將部署單一碎片，且 ElastiCache (Redis OSS) 將作為單一節點使用，但也可作為單一碎片叢集使用。相容模式表示用戶端應用程式可以使用任一通訊協定與叢集進行通訊。在此模式下，應用程式必須重新設定為開始使用 Valkey 或 Redis OSS 叢集通訊協定和組態端點。若要將 Valkey 或 Redis OSS 叢集模式變更為與叢集模式相容的模式，請遵循下列步驟：

**📘 Note**

在相容模式下，叢集不允許使用其他修改操作，例如擴展和引擎版本。此外，在 [ModifyReplicationGroup](#) 請求中定義叢集模式參數時，無法修改參數 ( 除 `cacheParameterGroupName` )。

- a. 使用 AWS Management Console，請參閱 [修改複寫群組](#) 並將叢集模式設定為相容
- b. 使用 API，請參閱 [並將 ClusterMode 參數 ModifyReplicationGroup 更新為 compatible。](#)
- c. 使用 AWS CLI，請參閱 [並將 cluster-mode 參數 modify-replication-group 更新為 compatible。](#)

將 Valkey 或 Redis OSS 叢集模式變更為叢集模式相容之後，[DescribeReplicationGroups](#) API 將傳回 ElastiCache (RedisOSS) 叢集組態端點。叢集配置端點是單一端點，可供應用程式用來連線至叢集。如需詳細資訊，請參閱 [在中尋找連線端點 ElastiCache。](#)

3. 將叢集組態修改為啟用叢集模式 – 一旦叢集模式設為叢集模式相容之後，第二個步驟即是將叢集組態修改為啟用叢集模式。在此模式下，單一碎片正在執行，客戶現在可以擴展其叢集或修改其他叢集組態。

若要將叢集模式變更為已啟用，請依照下列步驟進行：

開始之前，請確定您的 Valkey 或 Redis OSS 用戶端已使用叢集通訊協定遷移至 [Valkey](#)，且叢集的組態端點未使用。

- a. 使用 AWS Management Console，請參閱 [並將叢集模式 修改複寫群組](#) 設定為已啟用。
- b. 使用 API，請參閱 [並將 ClusterMode 參數 ModifyReplicationGroup 更新為 enabled。](#)
- c. 使用 AWS CLI，請參閱 [並將 cluster-mode 參數 modify-replication-group 更新為 enabled。](#)

將叢集模式變更為啟用後，端點將根據 Valkey 或 Redis OSS 叢集規格進行設定。[DescribeReplicationGroups](#) API 會將叢集模式參數傳回為 `enabled` 而叢集端點現在可供應用程式用來連線至叢集。

請注意，叢集模式變更為啟用後，叢集端點將會變更。請務必使用新端點更新您的應用程式。

您也可以選擇從叢集模式相容還原為停用的叢集模式 (CMD)，並保留原始組態。

從叢集模式相容將叢集組態修改為停用叢集模式

1. 使用 AWS Management Console，請參閱 [修改複寫群組](#) 並將叢集模式設定為已停用
2. 使用 API，請參閱 [並將 ClusterMode 參數 ModifyReplicationGroup 更新為 disabled。](#)
3. 使用 AWS CLI，請參閱 [並將 cluster-mode 參數 modify-replication-group 更新為 disabled。](#)

將叢集模式變更為停用後，[DescribeReplicationGroups](#) API 會將叢集模式參數傳回為 `disabled`。

## 使用全域資料存放區跨 AWS 區域複寫

### Note

全域資料儲存目前僅適用於自行設計的叢集。

使用全球資料存放區功能，您可以跨 AWS 區域使用完全受管、快速、可靠且安全的 Valkey 或 Redis OSS 叢集複寫。使用此功能，您可以建立跨區域僅供讀取複本叢集，以跨 AWS 區域啟用低延遲讀取和災難復原。

在以下各節中，您可以找到如何使用全域資料存放區的描述。

### 主題

- [概觀](#)
- [先決條件和限制](#)
- [使用全域資料存放區 \(主控台\)](#)
- [使用全域資料存放區 \(CLI\)](#)

### 概觀

每個全域資料存放區都是一個或多個叢集的集合，用於彼此複寫。

全域資料存放區由下列項目組成：

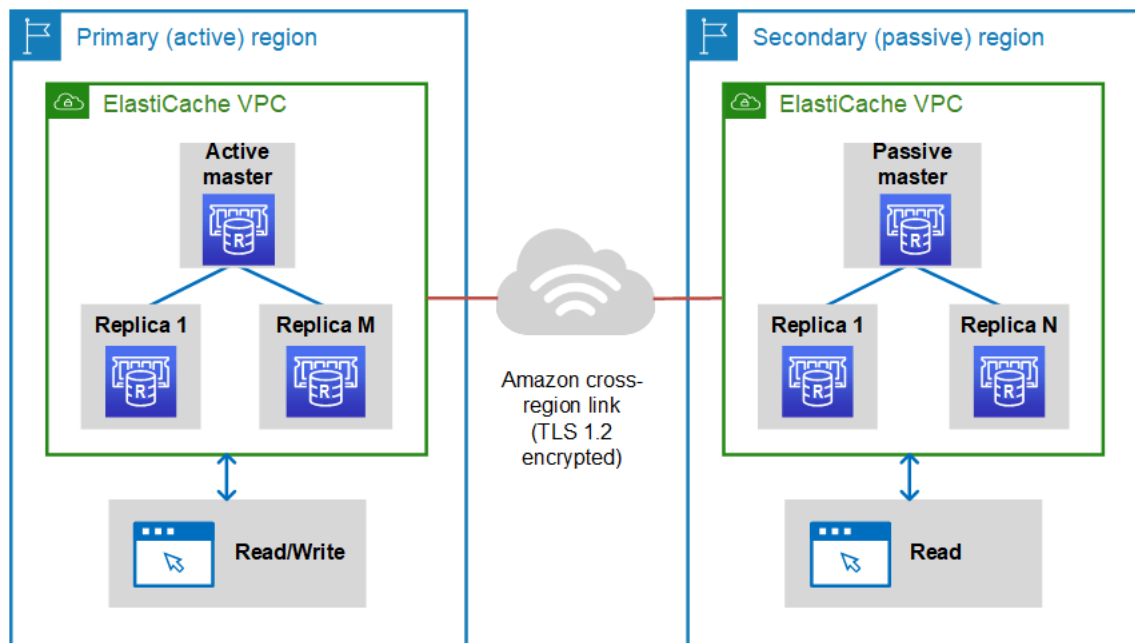
- **主要 (主動) 叢集** - 主要叢集接受複寫到全域資料存放區內所有叢集的寫入。主要叢集也接受讀取要求。
- **次要 (被動) 叢集** - 次要叢集只接受讀取要求，並從主要叢集複寫資料更新。次要叢集需要位於與主要叢集不同的 AWS 區域。

當您 ElastiCache 使用 Valkey 或 Redis 在中建立全域資料存放區時 OSS，其會自動將您的資料從主要叢集複寫到次要叢集。您可以選擇應該複寫 Valkey 或 Redis OSS 資料的 AWS 區域，然後在該 AWS 區域中建立次要叢集。ElastiCache 然後設定和管理兩個叢集之間的資料自動、非同步複寫。

使用適用於 Valkey 或 Redis 的全域資料存放區 OSS 具有下列優點：

- 地理效能 – 透過在其他 AWS 區域中設定遠端複本叢集並在它們之間同步資料，您可以降低該 AWS 區域中的資料存取延遲。全域資料存放區可以透過跨 AWS 區域提供低延遲的地理本機讀取，協助提高應用程式的回應能力。
- 災難復原 - 如果全域資料存放區中的主要叢集發生降級，您可以將次要叢集升級為新的主要叢集。您可以連線至任何包含次要叢集 AWS 的區域來執行此操作。

下圖顯示了全域資料存放區如何運作。



## 先決條件和限制

在開始使用全域資料存放區之前，請注意下列事項：

- 全球資料存放區支援以下 AWS 區域：亞太區域（首爾、東京、新加坡、雪梨、孟買和大阪）、歐洲（法蘭克福、巴黎、倫敦、愛爾蘭和斯德哥爾摩）、美國東部（維吉尼亞北部和俄亥俄）、美國西部（加州北部和奧勒岡北部）、南美洲（聖保羅）、AWS GovCloud（美國西部和美國東部）、加拿大（中部）區域、中國（北京和寧夏）
- 全域資料存放區中的所有叢集（主要和次要）應具有相同數量的主節點、節點類型、引擎版本和碎片數量（在啟用叢集模式的情況下）。全域資料存放區中的每個叢集都可以擁有不同數量的僅供讀取複本，以容納該叢集的本機讀取流量。

如果您計劃使用現有的單一節點叢集，則必須啟用複寫。

- 大型和更高大小的執行個體支援全域資料存放區。
- 您可以為主要叢集設定複寫，從一個 AWS 區域到最多兩個其他 AWS 區域的次要叢集。

**Note**

例外情況是中國 (北京) 區域和中國 (寧夏) 區域，在兩個區域之間只能進行複寫。

- 您只能在 VPC 叢集中使用全域資料存放區。如需詳細資訊，請參閱在 [Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)。當您使用 EC2-Classic 時，不支援全域資料存放區。如需詳細資訊，請參閱 Amazon 使用者指南中的 [EC2-Classic](#)。 EC2

**Note**

目前無法在 [搭配使用本機區域 ElastiCache](#) 使用全域資料存放區。

- ElastiCache 不支援從一個區域自動容錯移轉到另一個 AWS 區域。必要時，您可以手動升級次要叢集。如需範例，請參閱[將次要叢集升級為主要叢集](#)。
- 若要從現有資料啟動，請使用現有叢集做為主要叢集以建立全域資料存放區。我們不支援將現有叢集新增為次要叢集。將叢集新增為次要叢集的程序會清除資料，這可能會導致資料遺失。
- 當您修改屬於全域資料存放區之叢集的本機參數群組時，參數更新會套用至所有叢集。
- 您可以利用垂直方式 (擴展和縮減) 和水平方式 (橫向縮減和擴展) 來擴展區域叢集。您可以透過修改全域資料存放區來擴展叢集。全域資料存放區中的所有區域叢集隨後便會進行擴展，而不會發生中斷。如需詳細資訊，請參閱[擴展 ElastiCache](#)。
- 全域資料存放區支援[靜態加密](#)、[傳輸中的加密](#) 和 [AUTH](#)。
- 全域資料存放區不支援網際網路通訊協定第 6 版 (IPv6)。
- 全域資料存放區支援 AWS KMS 金鑰。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [AWS KMS 金鑰管理服務概念](#)。

**Note**

全域資料存放區支援[發佈/訂閱簡訊](#)，規定如下：

- 若是停用叢集模式，完全支援發佈/訂閱簡訊。在主要 AWS 區域的主要叢集上發佈的事件會傳播到次要 AWS 區域。
- 若是啟用叢集模式，適用下列條件：
  - 對於不在鍵空間中的已發佈事件，只有相同 AWS 區域中的訂閱者會收到事件。
  - 對於已發佈的鍵空間事件，所有 AWS 區域中的訂閱者都會收到事件。

## 使用全域資料存放區 (主控台)

若要使用主控台建立全域資料存放區，請遵循下列兩個步驟程序：

1. 使用現有叢集或建立新叢集，來建立主要叢集。引擎必須為 Valkey 7.2 或更新版本，或 Redis OSS 5.0.6 或更新版本。
2. 在不同 AWS 區域中新增最多兩個次要叢集，再次使用 Valkey 7.2 或更新版本，或 Redis 5.0.6 OSS 引擎或更新版本。

下列程序會指引您如何為 Valkey 或 Redis 建立全域資料存放區，OSS並使用 ElastiCache 主控台執行其他操作。

### 主題

- [使用現有叢集建立全域資料存放區](#)
- [使用新的主要叢集建立新的全域資料存放區](#)
- [檢視全域資料存放區詳細資訊](#)
- [將區域新增至全域資料存放區](#)
- [修改全域資料存放區](#)
- [將次要叢集升級為主要叢集](#)
- [從全域資料存放區移除區域](#)
- [刪除全域資料存放區](#)

### 使用現有叢集建立全域資料存放區

在此案例中，您使用現有的叢集做為新全域資料存放區的主要叢集。然後您可以在不同的 AWS 區域中建立次要唯讀叢集。這個次要叢集會從主要叢集接收自動的非同步更新。


#### Important

現有叢集必須使用 Valkey 7.2 或更新版本或 Redis 5.0.6 OSS 或更新版本的引擎。

### 使用現有叢集建立全域資料存放區


1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

2. 在導覽窗格中，選擇全域資料存放區，然後選擇建立全域資料存放區。
3. 在主要叢集設定頁面上，執行下列動作：
  - 在全域資料存放區資訊欄位中，輸入新全域資料存放區的名稱。
  - (選用) 輸入 Description (描述) 值。
4. 在區域叢集下，選取使用現有區域叢集。
5. 在現有叢集下，選取您要使用的現有叢集。
6. 保持以下選項不變。它們會預先填入以符合主要叢集組態，您無法變更它們。
  - 引擎版本
  - 節點類型
  - 參數群組

 Note

ElastiCache 從提供的參數群組的值自動產生新的參數群組，並將新的參數群組套用至叢集。使用此新參數群組可修改全域資料存放區上的參數。每個自動產生的參數群組都只與一個叢集相關聯，因此也只與一個全域資料存放區相關聯。

- 碎片數量
- Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

 Note

您可以選擇客戶受管 AWS KMS 金鑰並選擇金鑰，以提供不同的加密金鑰。如需詳細資訊，請參閱[使用客戶受管 AWS KMS 金鑰](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。對於 Valkey 7.2 及更新版本和 Redis OSS 引擎 6.0 版更新版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：
  - No Access Control (無存取控制) - 這是預設的設定。這表示沒有任何限制。
  - User Group Access Control List (使用者群組存取控制清單) - 選擇具有已定義的一組使用者和可用作業許可的使用者群組。如需詳細資訊，請參閱[使用主控台和管理使用者群組 CLI](#)。
  - AUTH 預設使用者 - Valkey 或 Redis OSS 伺服器的身分驗證機制。如需詳細資訊，請參閱[AUTH](#)。



7. (選用) 視需要更新其餘的次要叢集設定。這些值會預先填入與主要叢集相同的值，但您可以更新它們以符合該叢集的特定需求。
  - 連線埠
  - 複本數量
  - 子網路群組
  - 偏好的可用區域
  - 安全群組
  - 客戶受管 (AWS KMS 金鑰)
  - AUTH 權杖
  - 啟用自動備份
  - Backup retention period (備份保留期間)
  - 備份時段
  - Maintenance window (維護時段)
  - SNS 通知的主題
8. 選擇 Create (建立)。這樣做會將全域資料存放區的狀態設定為 Creating (正在建立)。狀態轉變為 Modifying (修改中)之後，主要叢集便與全域資料存放區建立關聯，且次要叢集處於 Associating (建立關聯中)狀態。

在主要叢集和次要叢集皆關聯至全域資料存放區之後，狀態會變更為 Available (可用)。此時，您已具備接受讀取和寫入的主要叢集，以及接受從主要叢集複寫之讀取的次要叢集。

頁面會更新，以指出叢集是否為全域資料存放區的一部分，包括：

- Global Datastore (全域資料存放區) - 叢集所屬全域資料存放區的名稱。
- Global Datastore Role (全域資料存放區角色) - 主要或次要叢集的角色。

您最多可以在不同的 AWS 區域中新增一個額外的次要叢集。如需詳細資訊，請參閱[將區域新增至全域資料存放區](#)。


使用新的主要叢集建立新的全域資料存放區

如果您選擇以新叢集建立全域資料存放區，請依照下列程序操作。

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟



2. 在導覽窗格中，選擇全域資料存放區，然後選擇建立全域資料存放區。
3. 在 Primary cluster settings (主要叢集設定) 中，執行下列動作：
  - a. 針對 Cluster mode (叢集模式)，選擇 Enabled (已啟用) 或 Disabled (已停用)。
  - b. 對於全域資料存放區資訊，輸入名稱 的值。ElastiCache 使用尾碼來產生全域資料存放區的唯一名稱。您可以使用您在此處指定的尾碼來搜尋全域資料存放區。
  - c. (選用) 輸入 Global Datastore Description (全域資料存放區描述) 的值。
4. 在 Regional cluster (區域叢集) 中：
  - a. 針對區域，選擇可用的 AWS 區域。
  - b. 選擇 Create new regional cluster (建立新的區域叢集)，或 Use existing regional cluster (使用現有的區域叢集)。
  - c. 如果選擇 Create new regional cluster (建立新的區域叢集)，請在 Cluster info (叢集資訊) 中，輸入叢集的名稱和選用描述。
  - d. 在 Location (位置) 中，建議您接受 Multi-AZ (多可用區域) 和 Auto-failover (自動容錯移轉) 的預設設定。
5. 在 Cluster settings (叢集設定) 中
  - a. 針對 Engine version (引擎版本)，選擇可用版本，即 5.0.6 或更新版本。
  - b. 針對 Port (連接埠)，使用預設連接埠 6379。如果您基於任何原因需要使用不同的連接埠，請輸入連接埠號碼。
  - c. 對於 Parameter group (參數群組)，請選擇參數群組或新建參數群組。參數群組可控制叢集的執行時間參數。如需參數群組的詳細資訊，請參閱 [Valkey](#) 和 [Redis OSS 參數](#) 和 [建立 ElastiCache 參數群組](#)。

 Note

當您選取參數群組來設定引擎組態值時，該參數群組會套用至全域資料存放區中的所有叢集。在 Parameter Groups (參數群組) 頁面上，yes/no (是/否) Global (全域) 屬性指出參數群組是否為全域資料存放區的一部分。

- d. 在 Node type (節點類型) 中，選擇向下箭頭 (▼) )。  
在 Change node type (變更節點類型) 對話方塊中，為您想要的節點類型選擇 Instance family (執行個體系列) 的值。然後選擇您要用於此叢集的節點類型，然後選擇 Save (儲存)。

如需詳細資訊，請參閱[選擇您的節點大小](#)。

如果選擇 r6gd 節點類型，將自動啟用資料分層。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

e. 如果您要建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集：

針對 Number of replicas ( 複本數 )，選擇您要讓此叢集具備的複本數目。

f. 如果您要建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集：

i. 針對碎片數量，選擇您想要在此 Valkey 或 Redis ( 啟用叢集模式 ) 叢集中取得的碎片數量 OSS ( 分割區/節點群組 )。

對於某些版本的 Valkey 或 Redis OSS ( 啟用叢集模式 )，您可以動態變更叢集中的碎片數量：

- Redis OSS 3.2.10 及更新版本 – 如果您的叢集執行 Redis OSS 3.2.10 或更新版本，您可以動態變更叢集中的碎片數量。如需詳細資訊，請參閱[在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)。
- 其他 Redis OSS 版本 – 如果您的叢集在 3.2.10 版本 OSS 之前執行 Redis 版本，則有其他方法。在這種情況下，若要變更叢集中的碎片數量，請使用新的碎片數量建立新的叢集。如需詳細資訊，請參閱[從備份還原到新的快取](#)。

ii. 針對 Replicas per shard ( 每個碎片的複本 )，選擇您要讓每個碎片具備的僅供讀取複本節點數目。

Valkey 或 Redis 存在下列限制 OSS ( 啟用叢集模式 )。

- 如果您已啟用多個可用區，請確保每個碎片至少有一個複本。
- 使用主控台建立叢集時，每個碎片的複本數都相同。
- 每個碎片的僅供讀取複本數均固定而無法變更。如果您發現每個碎片需要更多或更少的複本 ( API/CLI：節點群組 )，則必須使用新的複本數目建立新的叢集。如需詳細資訊，請參閱[教學課程：使用外部建立的備份來尋找新的自我設計叢集](#)。

6. 針對子網路群組設定，選擇您要套用至此叢集的子網路。ElastiCache 提供預設 IPv4 子網路群組，或者您可以選擇建立新的子網路群組。對於 IPv6，您需要使用 IPv6 CIDR 區塊建立子網路群組。如果您選擇雙堆疊，則您必須選取探索 IP 類型，可以是 IPv6 或 IPv4。

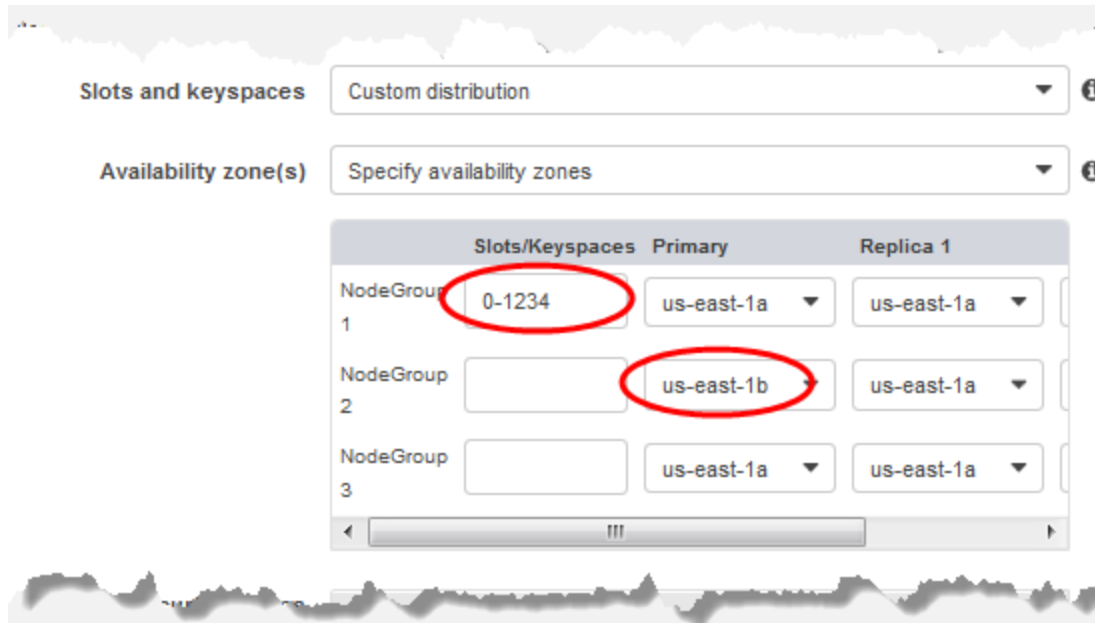
如需詳細資訊，請參閱在您的 [中建立子網路 VPC](#)。

7. 針對 Availability zone placements ( 可用區域置放 )，您有兩種選項：

- 沒有偏好設定 – ElastiCache 選擇可用區域。
- Specify availability zones (指定可用區域) - 您可為每個叢集指定可用區域。

如果您選擇指定可用區域，請針對每個碎片中的每個叢集，從清單中選擇可用區域。

如需詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。



指定 Keyspace 和可用區域

8. 選擇 Next (下一步)
  9. 在進階 Valkey 和 Redis OSS設定下
- 針對 Security (安全) :


i. 若要加密資料，您有下列選項：

- Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

**Note**

您可以選擇 Customer Managed AWS KMS 金鑰並選擇 金鑰，以提供不同的加密金鑰。如需詳細資訊，請參閱[從 AWS 使用客戶受管金鑰KMS](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。針對 Valkey 7.2 及更高版本，以及 Redis OSS引擎 6.0 及更高版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：
- No Access Control (無存取控制) - 這是預設的設定。這表示使用者存取叢集時不受任何限制。
- User Group Access Control List (使用者群組存取控制清單) - 選取的使用者群組需具有可存取叢集的已定義使用者集。如需詳細資訊，請參閱[使用主控台和管理使用者群組 CLI](#)。
- AUTH 預設使用者 – Valkey 或 Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱[AUTH](#)。
- AUTH – Valkey 或 Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱[AUTH](#)。

 Note

對於 3.2.6 版以後的 Redis OSS版本，除了 3.2.10 版之外，AUTH是唯一的選項。

- ii. 在 Security groups (安全群組) 中，選擇要用於此叢集的安全群組。安全群組可做為防火牆來控制叢集的網路存取。您可以使用的預設安全群組，VPC或建立新的安全群組。

如需安全群組的詳細資訊，請參閱 Amazon 使用者指南中的 [的安全群組VPC](#)。 VPC

10. 針對定期排程自動備份，請選取 Enable automatic backups (啟用自動備份)，然後輸入每份自動備份要保存的天數，過了此天數後即會自動將其刪除。如果您不想定期排程自動備份，請清除 Enable automatic backups (啟用自動備份) 核取方塊。在任何一種情況下，您都可以選擇建立手動備份。

如需備份和還原的詳細資訊，請參閱 [快照和還原](#)。

11. (選用) 指定維護時段。維護時段是每週 ElastiCache 排程叢集系統維護的時間，通常為一小時。您可以允許為您的維護時段 ElastiCache 選擇日期和時間（無偏好設定），也可以自行選擇天數、時間和持續時間（指定維護時段）。如果您從清單中選擇 Specify maintenance window (指定維護時段)，請為您的維護時段選擇 Start day (開始日)、Start time (開始時間) 和 Duration (持續時間)。所有時間都是 UCT 次。

如需詳細資訊，請參閱[管理 ElastiCache 叢集維護](#)。

12. (選用) 針對 Logs (日誌)：

- 在日誌格式下，選擇文字或 JSON。
  - 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
  - 在日誌目的地下，選擇建立新 CloudWatch 日誌群組名稱或 Firehose 串流名稱，或選擇選取現有日誌群組名稱或 Firehose 串流名稱，然後選擇您的 CloudWatch 日誌群組名稱或您的 Firehose 串流名稱。
13. 對於標籤，為協助您管理叢集和其他 ElastiCache 資源，您可以標籤形式將自己的中繼資料指派給每個資源。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。
  14. 檢閱所有項目和選項，然後進行任何所需的更正。就緒後，請選擇 Next (下一步)。
  15. 在上述步驟中設定叢集之後，現在設定次要叢集詳細資訊。
  16. 在區域叢集下，選擇第 叢集所在的 AWS 區域。
  17. 在 Cluster info (叢集資訊) 中，輸入叢集的名稱和選用描述。
  18. 下列選項會預先填入以符合主要叢集組態，且無法變更：
    - 位置
    - 引擎版本
    - 執行個體類型
    - 節點類型
    - 碎片數量
    - 參數群組

**Note**


ElastiCache 從提供的參數群組的值自動產生新的參數群組，並將新的參數群組套用至叢集。使用此新參數群組可修改全域資料存放區上的參數。每個自動產生的參數群組都只與一個叢集相關聯，因此也只與一個全域資料存放區相關聯。

- Encryption at rest (靜態加密) - 啟用存放在磁碟上的資料加密功能。如需詳細資訊，請參閱[靜態加密](#)。

**Note**

您可以選擇客戶受管 AWS KMS 金鑰並選擇金鑰，以提供不同的加密金鑰。如需詳細資訊，請參閱[使用客戶受管 AWS KMS 金鑰](#)。

- Encryption in-transit (傳輸中加密) - 啟用傳輸中資料加密功能。如需詳細資訊，請參閱[傳輸中加密](#)。針對 Valkey 7.2 及更高版本，以及 Redis OSS引擎 6.4 及更高版本，如果您啟用傳輸中加密，系統會提示您指定下列其中一個存取控制選項：
  - No Access Control (無存取控制) - 這是預設的設定。這表示使用者存取叢集時不受任何限制。
  - User Group Access Control List (使用者群組存取控制清單) - 選擇的使用者群組需具有可存取叢集的已定義使用者集。如需詳細資訊，請參閱[使用主控台和 管理使用者群組 CLI](#)。
  - AUTH 預設使用者 – Valkey 或 Redis OSS伺服器的身分驗證機制。如需詳細資訊，請參閱[AUTH](#)。

 Note

對於 4.0.2 之間的 Redis OSS版本，當第一次支援傳輸中加密時，和 6.0.4 AUTH是唯一的選項。

其餘的次要叢集設定會預先填入與主要叢集相同的值，但是可以更新下列項目以符合該叢集的特定需求：

- 連線埠
- 複本數量
- 子網路群組
- 偏好的可用區域
- 安全群組
- 客戶受管 ( AWS KMS 金鑰 )
- AUTH 權杖
- 啟用自動備份
- Backup retention period (備份保留期間)
- 備份時段
- Maintenance window (維護時段)
- SNS 通知的主題

19. 選擇 Create (建立)。這樣做會將全域資料存放區的狀態設定為 Creating (正在建立)。在主要叢集和次要叢集皆關聯至全域資料存放區之後，狀態會變更為 Available (可用)。您已具備接受讀取和寫入的主要叢集，以及接受從主要叢集複寫之讀取的次要叢集。

此頁面也會更新，以指出叢集是否為全域資料存放區的一部分，包括下列項目：

- Global Datastore (全域資料存放區) - 叢集所屬全域資料存放區的名稱。
- Global Datastore Role (全域資料存放區角色) - 主要或次要叢集的角色。

您最多可以在不同的 AWS 區域中新增一個額外的次要叢集。如需詳細資訊，請參閱[將區域新增至全域資料存放區](#)。

### 檢視全域資料存放區詳細資訊

您可以在全球資料存放區頁面上檢視現有全球資料存放區的詳細資訊，並加以修改。

### 檢視全域資料存放區詳細資訊

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇全域資料存放區，然後選擇可用的全域資料存放區。

接著，您可以檢查下列全域資料存放區屬性：

- 全域資料存放區名稱：全域資料存放區的名稱
- 描述：全域資料存放區的描述
- 狀態：選項包括：
  - 正在建立
  - Modifying (正在修改)
  - 可用性
  - 正在刪除
  - Primary-Only (僅限主要) - 此狀態表示全域資料存放區僅包含主要叢集。所有次要叢集都已刪除或未成功建立。
- 叢集模式：啟用或停用
- 引擎版本：執行全域資料存放區的 Valkey 或 Redis OSS 引擎版本
- 執行個體節點類型：用於全域資料存放區的節點類型



- 靜態加密：啟用或停用
- 傳輸中加密：啟用或停用
- AUTH：啟用或停用

您可以對全域資料存放區進行下列變更：

- [將區域新增至全域資料存放區](#)
- [從全域資料存放區移除區域](#)
- [將次要叢集升級為主要叢集](#)
- [修改全域資料存放區](#)

全域資料存放區頁面也會列出組成全域資料存放區的個別叢集，以及每個叢集的下列屬性：

- 區域 - 儲存叢集 AWS 的區域
- Role (角色) - 主要或次要
- Cluster name (叢集名稱) - 叢集的名稱
- Status (狀態) - 選項包括：
  - Associating (正在關聯) - 叢集正在與全域資料存放區進行關聯
  - Associated (已關聯) - 叢集已關聯至全域資料存放區關聯
  - Disassociating (正在取消關聯) - 正在進行從使用全域資料存放區名稱的全域資料存放區移除次要叢集的程序。之後，次要叢集將不再收到主要叢集的更新，但會保留為該 AWS 區域中的獨立叢集。
  - Disassociated (已取消關聯) - 次要叢集已從全域資料存放區移除，現在是其 AWS 區域中的獨立叢集。
- 全域資料存放區複本延遲 – 顯示全域資料存放區中每個次要 AWS 區域的一個值。這是次要區域的主節點和主要區域的主節點之間的延遲。對於啟用叢集模式的 Valkey 或 Redis OSS，延遲表示碎片之間的最大延遲秒數。

### 將區域新增至全域資料存放區

您最多可以將一個額外 AWS 區域新增至現有的全域資料存放區。在此案例中，您要在從主要叢集接收自動和非同步更新的個別 AWS 區域中建立唯讀叢集。



## 將 AWS 區域新增至全域資料存放區

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇全域資料存放區，然後選擇現有的全域資料存放區。
3. 選擇新增區域叢集，然後選擇要存放次要叢集 AWS 的區域。
4. 在叢集資訊下，輸入名稱的值，以及選擇性地輸入叢集描述的值。
5. 保持以下選項不變。這些項目會預先填入以符合主要叢集組態，且您無法加以變更。

- 引擎版本
- 執行個體類型
- 節點類型
- 碎片數量
- 參數群組

### Note

ElastiCache 從提供的參數群組的值自動產生新的參數群組，並將新的參數群組套用至叢集。使用此新參數群組可修改全域資料存放區上的參數。每個自動產生的參數群組都只與一個叢集相關聯，因此也只與一個全域資料存放區相關聯。

- 靜態加密

### Note

您可以選擇 Customer Managed AWS KMS 金鑰並選擇 金鑰，以提供不同的加密金鑰。

- 傳輸中加密
  - AUTH
6. (選用) 更新其餘的次要叢集設定。這些值會預先填入與主要叢集相同的值，但您可以更新它們以符合該叢集的特定需求：

- 連線埠
- 複本數量
- 子網路群組

- 偏好的可用區域
- 安全群組
- 客戶受管 AWS KMS 金鑰 )
- AUTH 權杖
- 啟用自動備份
- Backup retention period (備份保留期間)
- 備份時段
- Maintenance window (維護時段)
- SNS 通知的主題

## 7. 選擇新增。

### 修改全域資料存放區

您可以修改區域叢集的屬性。除了將次要叢集升級為主要叢集之外，全域資料存放區上一次只能進行一項修改作業。如需詳細資訊，請參閱[將次要叢集升級為主要叢集](#)。

### 修改全域資料存放區

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇全域資料存放區，然後在全域資料存放區名稱中，選擇全域資料存放區。
3. 選擇 Modify (修改)，然後選擇下列選項：
  - Modify description (修改描述) - 更新全域資料存放區的描述
  - 修改引擎版本 – 只有 Valkey 7.2 及更新版本或 Redis OSS 引擎 5.0.6 及更新版本可用。
  - Modify node type (修改節點類型) - 可利用垂直方式 (擴充規模和縮減規模) 和水平方式 (橫向縮減和擴展) 來擴展區域叢集。選項包括 R5 和 M5 節點系列。如需節點類型的詳細資訊，請參閱[支援的節點類型](#)。
  - Modify Automatic Failover (修改自動容錯移轉) - 啟用或停用自動容錯移轉功能。當您在區域叢集中啟用容錯移轉和主要節點意外關閉時，會 ElastiCache 容錯移轉至其中一個區域複本。如需詳細資訊，請參閱[自動容錯移轉](#)。

對於已啟用 OSS 叢集模式的 Valkey 或 Redis 叢集：

- Add shards (新增碎片) - 輸入要新增的碎片數目，並選擇性指定一或多個可用區域。

- 刪除碎片 – 選擇要在每個 AWS 區域中刪除的碎片。
- Rebalance shards (重新平衡碎片) - 重新平衡位置分佈，確保在叢集中現有的碎片間均勻分佈。

若要修改全域資料存放區的參數，請修改全域資料存放區中任何成員叢集的參數群組。會自動將此變更 ElastiCache 套用至該全域資料存放區中的所有叢集。若要修改該叢集的參數群組，請使用 Valkey 或 Redis OSS 主控台或 [ModifyCacheCluster](#) API 操作。如需詳細資訊，請參閱 [修改 ElastiCache 參數群組](#)。當您修改全域資料存放區中包含之任何叢集的參數群組時，它會套用至該全域資料存放區內的所有叢集。

若要重設整個參數群組或特定參數，請使用 [ResetCacheParameterGroup](#) API 操作。

### 將次要叢集升級為主要叢集

如果主要叢集或 AWS 區域無法使用或遇到效能問題，您可以將次要叢集提升為主要叢集。即使正在進行其他修改，也可隨時進行升級。您也可以同時發佈多個升級，全域資料存放區最終會解析為一個主要叢集。如果您同時提升多個次要叢集，則 ElastiCache 使用 Valkey 或 Redis OSS 並不保證最終解析為主要叢集。

### 將次要叢集升級為主要叢集

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇全域資料存放區。
3. 選擇全域資料存放區名稱以檢視詳細資訊。
4. 選擇 Secondary (次要) 叢集。
5. 選擇 Promote to primary (升級為主要)。

接著系統會提示您確認您的決定，並顯示下列警告：Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the *secondary* cluster to primary?

The current primary cluster in *primary region* will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.

6. 如果您要繼續升級，請選擇 Confirm (確認)，如果不想繼續升級，請選擇 Cancel (取消)。

如果您選擇確認，則全域資料存放區會移至 Modifying (正在修改) 狀態，且在升級完成之前無法使用。

### 從全域資料存放區移除區域

您可以使用下列程序，從全域資料存放區移除 AWS 區域。

#### 從全域資料存放區移除 AWS 區域

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇全域資料存放區。
3. 選擇全域資料存放區。
4. 選擇您要移除的 Region (區域)。
5. 選擇 Remove region (移除區域)。

#### Note

此選項僅適用於次要叢集。

接著系統會提示您確認您的決定，並顯示下列警告： Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?

6. 如果您要繼續升級，請選擇 Confirm (確認)，如果不想繼續升級，請選擇 Cancel (取消)。

如果您選擇確認，則會移除 AWS 區域，次要叢集將不再接收複寫更新。

### 刪除全域資料存放區

若要刪除全域資料存放區，請先移除所有次要叢集。如需詳細資訊，請參閱 [從全域資料存放區移除區域](#)。這樣做會使全域資料存放區處於 primary-only (僅限主要) 狀態。

### 刪除全域資料存放區

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

2. 在導覽窗格中，選擇全域資料存放區。
3. 在 Global Datastore Name (全域資料存放區名稱) 下，選擇您要刪除的全域資料存放區，然後選擇 Delete (刪除)。

接著系統會提示您確認您的決定，並顯示下列警告：Are you sure you want to delete this Global Datastore?

4. 選擇 刪除。

全域資料存放區會轉換為 Deleting (正在刪除) 狀態。

## 使用全域資料存放區 ( CLI )

您可以使用 AWS Command Line Interface ( AWS CLI ) 從命令列控制多個 AWS 服務，並透過指令碼將其自動化。您可以使用 AWS CLI 進行臨時 ( 一次性 ) 操作。

### 下載和設定 AWS CLI

AWS CLI 會在 Windows、macOS 或 Linux 上執行。請使用下列程序來下載並進行設定。

### 若要下載、安裝和設定 CLI

1. 在[AWS 命令列介面](#)網頁上下載 AWS CLI。
2. 請遵循 AWS Command Line Interface 使用者指南 CLI AWS 中的安裝 AWS CLI 和設定的指示。

### 將 AWS CLI與全域資料存放區搭配使用

使用下列CLI操作來使用全域資料存放區：

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \
 --global-replication-group-id-suffix my global datastore \
 --primary-replication-group-id sample-repl-group \
 --global-replication-group-description an optional description of the global
 datastore
```

建立 Amazon 時 ElastiCache，會自動將字首套用至全域資料存放區 ID。每個 AWS 區域都有自己的字首。例如，在美國西部 (加利佛尼亞北部) 區域建立的全域資料儲存 ID 都以 "virxk" 開頭，加上

您提供的尾碼名稱。尾碼與自動產生的字首結合，可保證跨多個區域的全域資料存放區名稱保持唯一性。

下表列出每個 AWS 區域及其全域資料存放區 ID 字首。

| 區域名稱/區域                        | 字首    |
|--------------------------------|-------|
| 美國東部 (俄亥俄) 區域<br>us-east-2     | fpkhr |
| 美國東部 (維吉尼亞北部) 區域<br>us-east-1  | ldgnf |
| 美國西部 (加利佛尼亞北部) 區域<br>us-west-1 | virxk |
| 美國西部 (奧勒岡) 區域<br>us-west-2     | sgau  |
| 加拿大 (中部) 區域<br>ca-central-1    | bxodz |
| 亞太 (孟買) 區域<br>ap-south-1       | erpgt |
| 亞太區域 (東京) 區域<br>ap-northeast-1 | quwsw |
| 亞太區域 (首爾) 區域<br>ap-northeast-2 | lfqnh |

| 區域名稱/區域                         | 字首    |
|---------------------------------|-------|
| 亞太 (大阪) 區域<br>ap-northeast-3    | nlapn |
| 亞太區域 (新加坡) 區域<br>ap-southeast-1 | vlqxn |
| 亞太區域 (雪梨) 區域<br>ap-southeast-2  | vbgxd |
| 歐洲 (法蘭克福) 區域<br>eu-central-1    | iudkw |
| 歐洲 (愛爾蘭) 區域<br>eu-west-1        | gxeiz |
| 歐洲 (倫敦) 區域<br>eu-west-2         | okuqm |
| 歐洲 (巴黎) 區域<br>eu-west-3         | fgjhi |
| 南美洲 (聖保羅) 區域<br>sa-east-1       | juxlw |
| 中國 (北京) 區域<br>cn-north-1        | emvgo |
| 中國 (寧夏) 區域<br>cn-northwest-1    | ckbem |

| 區域名稱/區域                              | 字首    |
|--------------------------------------|-------|
| 亞太區域 (香港) 區域<br>ap-east-1            | knjmp |
| AWS GovCloud (美國西部)<br>us-gov-west-1 | sgwui |

- [create-replication-group](#) – 使用此操作，透過將全域資料存放區的名稱提供給 `--global-replication-group-id` 參數，為全域資料存放區建立次要叢集。

```
aws elasticache create-replication-group \
 --replication-group-id secondary replication group name \
 --replication-group-description "Replication group description" \
 --global-replication-group-id global datastore name
```

呼叫此操作並傳遞 `--global-replication-group-id` 值時，ElastiCache 會從全域複寫群組的主要複寫群組推斷下列參數的值。請勿傳入這些參數的值：

"PrimaryClusterId",  
 "AutomaticFailoverEnabled",  
 "NumNodeGroups",  
 "CacheParameterGroupName",  
 "CacheNodeType",  
 "Engine",  
 "EngineVersion",  
 "CacheSecurityGroupNames",  
 "EnableTransitEncryption",  
 "AtRestEncryptionEnabled",



"SnapshotArns",

"SnapshotName"

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \
 --global-replication-group-id my global datastore \
 --show-member-info an optional parameter that returns a list of the primary and
 secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id my global datastore \
 --automatic-failover-enabled \
 --cache-node-type node type \
 --cache-parameter-group-name parameter group name \
 --engine-version engine version \
 --apply-immediately \
 --global-replication-group-description description
```

## Redis 對 OSS Valkey 跨引擎升級 ElastiCache GlobalDataStore

您可以使用主控台或 將現有的 Redis OSS全域複寫群組升級至 Valkey API 引擎CLI。

如果您有現有的 Redis OSS全域複寫群組，您可以使用 指定新的引擎和引擎版本 `modify-global-replication-group`，升級至 ValkeyAPI。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id myGlobalReplGroup \
 --engine valkey \
 --apply-immediately \
 --engine-version 7.2
```

針對 Windows：

```
aws elasticache modify-global-replication-group ^
```

```
--global-replication-group-id myGlobalReplGroup ^
--engine valkey ^
--apply-immediately ^
--engine-version 7.2
```

如果您將自訂快取參數群組套用至要升級的現有 Redis OSS全域複寫群組，您也需要在請求中傳遞自訂 Valkey 快取參數群組。輸入 Valkey 自訂參數群組必須與現有的 Redis OSS自訂參數群組具有相同的 Redis OSS 靜態參數值。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id myGlobalReplGroup \
 --engine valkey \
 --engine-version 7.2 \
 --apply-immediately \
 --cache-parameter-group-name myParamGroup
```

針對 Windows：

```
aws elasticache modify-global-replication-group ^
 --global-replication-group-id myGlobalReplGroup ^
 --engine valkey ^
 --engine-version 7.2 ^
 --apply-immediately ^
 --cache-parameter-group-name myParamGroup
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \
 --global-replication-group-id my global datastore \
 --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \
 --global-replication-group-id my global datastore \
 --replication-group-id my secondary cluster \
 --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \
 --global-replication-group-id my global datastore \
 --primary-region The AWS Region of the primary cluster \
 --primary-replication-group-id The name of the global datastore, including the
 suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name \
 --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name \
 --node-group-count 3
```

- [rebalance-shards-in-global-複本-群組](#)

```
aws elasticache rebalance-shards-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name
```

使用說明來列出 ElastiCache 使用 Valkey 或 Redis 的所有可用命令OSS。

```
aws elasticache help
```

您還可以使用說明來描述特定命令，並了解更多關於它的用法：

```
aws elasticache create-global-replication-group help
```

## 使用複寫群組的高可用性

單節點 Amazon ElastiCache Valkey 和 Redis OSS叢集是具有有限資料保護服務的記憶體內實體 ( AOF )。若您的叢集因任何原因失敗，您會遺失所有叢集的資料。不過，如果您執行 Valkey 或

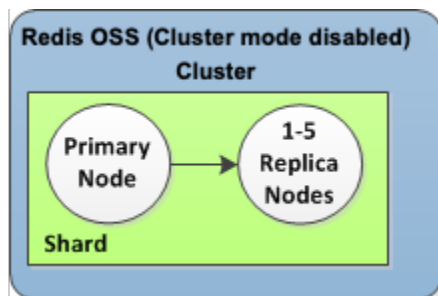
Redis OSS引擎，您可以將 2 到 6 個節點分組到具有複本的叢集中，其中 1 到 5 個唯讀節點包含群組單一讀取/寫入主要節點的複寫資料。在此案例中，若一個節點因任何原因失敗，因為其已在一或多個其他節點中複寫，因此您也不會遺失所有資料。因為複寫延遲，若失敗的是主要讀取/寫入節點，有些資料可能會遺失。

如下圖所示，複寫結構包含於包含於 Valkey 或 Redis OSS叢集的碎片（API/ 中稱為節點群組CLI）內。Valkey 或 Redis OSS（停用叢集模式）叢集一律有一個碎片。Valkey 或 Redis OSS（啟用叢集模式）叢集最多可以有 500 個碎片，其中叢集的資料分割橫跨碎片。您可以建立具有較高數量碎片和較低數量複本的叢集，每一叢集最高總計為 90 個節點。此叢集組態的範圍可以從 90 個碎片和 0 個複本到 15 個碎片和 5 個複本，這是允許的複本最大數量。

節點或碎片限制可以提高到每個叢集最多 500 個，使用 Valkey 和 Redis OSS引擎 5.0.6 版或更新版本。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個（每個碎片一個主要版本和 5 個複本）到 500 個（單一主要版本並且沒有複本）之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路CIDR太小，範圍太小，或者子網路被其他叢集共用且大量使用。如需詳細資訊，請參閱[建立子網路群組](#)。

對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱[AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type（每執行個體類型每叢集的節點數）。



Valkey 或 Redis OSS（停用叢集模式）叢集有一個碎片和 0 到 5 個複本節點

若帶有複本的叢集已啟用多個可用區，並且當主要節點故障時，主要節點會容錯移轉至僅供讀取複本。因為資料會以非同步的方式在複本節點上更新，可能會因更新複本節點時發生延遲而導致某些資料遺失。如需詳細資訊，請參閱[緩解執行 Valkey 或 Redis 時的失敗 OSS](#)。

## 主題

- [了解 Valkey 和 Redis OSS複寫](#)
- [複寫：停用 Valkey 和 Redis OSS 叢集模式與啟用](#)
- [搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS](#)

- [同步與備份的實作方式](#)
- [建立 Valkey 或 Redis OSS 複寫群組](#)
- [檢視複寫群組詳細資訊](#)
- [尋找複寫群組端點](#)
- [修改複寫群組](#)
- [刪除複寫群組](#)
- [變更複本的數量](#)
- [針對 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組，將僅供讀取複本提升為主要複本](#)

## 了解 Valkey 和 Redis OSS 複寫

Redis 以兩種方式 OSS 實作複寫：

- 使用包含每個節點中叢集所有資料的單一碎片：Valkey 或 Redis OSS ( 停用叢集模式 )
- 將資料分割為最多 500 個碎片：Valkey 或 Redis OSS ( 啟用叢集模式 )

複寫群組中的每個碎片都具備單一讀/寫主節點，以及最多 5 個僅供讀取複本節點。您可以建立具有較高數量碎片和較低數量複本的叢集，每一叢集最高總計為 90 個節點。此叢集組態的範圍可以從 90 個碎片和 0 個複本到 15 個碎片和 5 個複本，這是允許的複本最大數量。

如果 Redis OSS 引擎版本為 5.0.6 或更高，節點或碎片限制可以增加到每個叢集最多 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 ( 每個碎片一個主要版本和 5 個複本 ) 到 500 個 ( 單一主要版本並且沒有複本 ) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路 CIDR 太小，範圍太小，或者子網路被其他叢集共用且大量使用。如需詳細資訊，請參閱 [建立子網路群組](#)。

對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

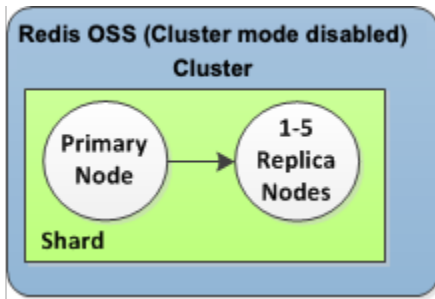
若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

### 主題

- [Valkey 或 Redis OSS \( 停用叢集模式 \)](#)
- [Valkey 或 Redis OSS \( 啟用叢集模式 \)](#)

### Valkey 或 Redis OSS ( 停用叢集模式 )

Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集具有單一碎片，其中包含節點集合；一個主要讀取/寫入節點和最多五個次要唯讀複本節點。每個僅供讀取複本都會保有從叢集主要節點取得的資料複本。非同步複寫機制會用於將僅供讀取複本與主要節點保持同步。應用程式可從叢集內的任何節點進行讀取。應用程式只能寫入主要節點。僅供讀取複本可改善讀取輸送量，並在節點故障時保護資料，避免遺失。



具有單一碎片和複本節點的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集

您可以使用具有複本節點的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集來擴展解決方案，以 ElastiCache 處理讀取密集型應用程式，或支援同時從相同叢集讀取的大量用戶端。

Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集中的所有節點必須位於相同的區域。

當您將僅供讀取複本新增到叢集時，所有來自主要節點的資料都會複製到新節點。之後，每當資料寫入到主要節點時，變更都會以非同步方式散佈到所有僅供讀取複本。

若要改善容錯能力並減少寫入停機時間，請為具有複本的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集啟用具有自動容錯移轉的多可用區。如需詳細資訊，請參閱[搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS](#)。

您可以變更 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集內節點的角色，其中包含交換角色的主要複本和其中一個複本。您可能會為了調校效能而執行此作業。例如，針對需要大量寫入活動的 web 應用程式，您可以選擇網路延遲最低的節點。如需詳細資訊，請參閱[針對 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組，將僅供讀取複本提升為主要複本](#)。

Valkey 或 Redis OSS ( 啟用叢集模式 )

Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集由 1 到 500 個碎片 ( API/CLI : 節點群組 ) 組成。每個碎片都有一個主節點，以及最多 5 個僅供讀取複本節點。組態的範圍可以從 90 個碎片和 0 個複本到 15 個碎片和 5 個複本，這是允許的複本最大數量。

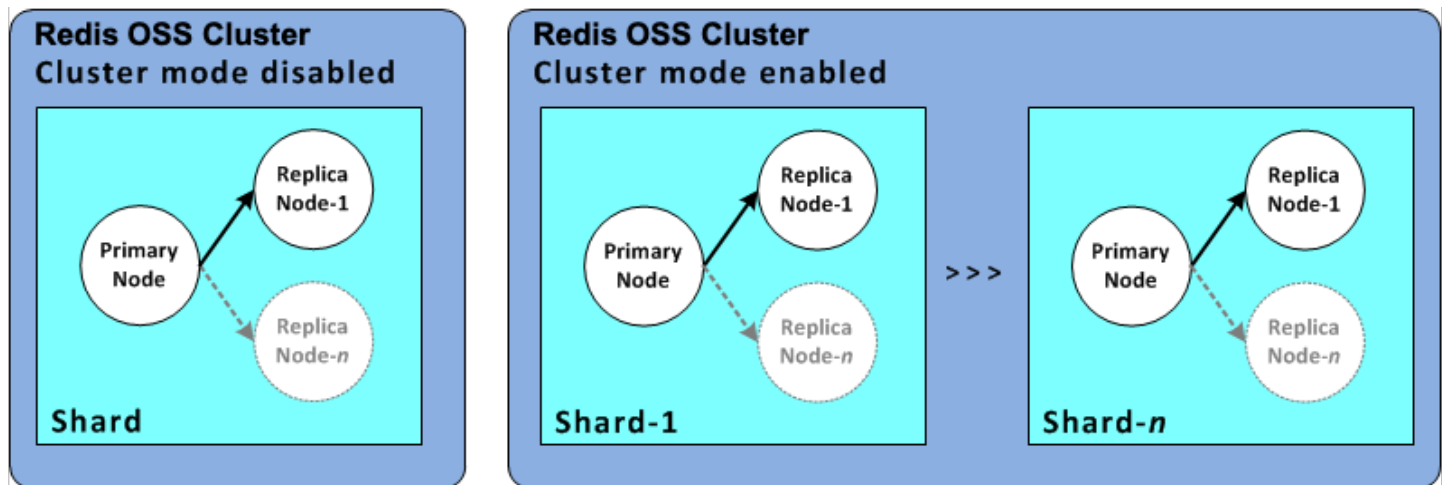
如果引擎版本為 Valkey 7.2 和更高版本，或 Redis 5.0.6 和更高版本，則節點或碎片限制可以增加到每個叢集最多 OSS 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 ( 每個碎片一個主要版本和 5 個複本 ) 到 500 個 ( 單一主要版本並且沒有複本 ) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路 CIDR 太小，範圍太小，或者子網路被其他叢集共用且大量使用。如需詳細資訊，請參閱[建立子網路群組](#)。

對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

每個碎片中的僅供讀取複本都會保有從碎片主要節點取得的資料複本。非同步複寫機制會用於將僅供讀取複本與主要節點保持同步。應用程式可從叢集內的任何節點進行讀取。應用程式只能寫入主要節點。僅供讀取複本可增強讀取擴展性及保護資料遺失。資料會在 Valkey 或 Redis OSS (啟用叢集模式) 叢集的碎片上進行分割。

應用程式會使用 Valkey 或 Redis OSS (啟用叢集模式) 叢集的組態端點來與叢集中的節點連線。如需詳細資訊，請參閱在 [中尋找連線端點 ElastiCache](#)。



Valkey 或 Redis OSS (啟用叢集模式) 叢集，具有多個碎片和複本節點

Valkey 或 Redis OSS (啟用叢集模式) 叢集中的所有節點必須位於相同的區域。若要改善容錯能力，您可以在該區域中的多個可用區域內同時佈建主要節點及僅供讀取複本。

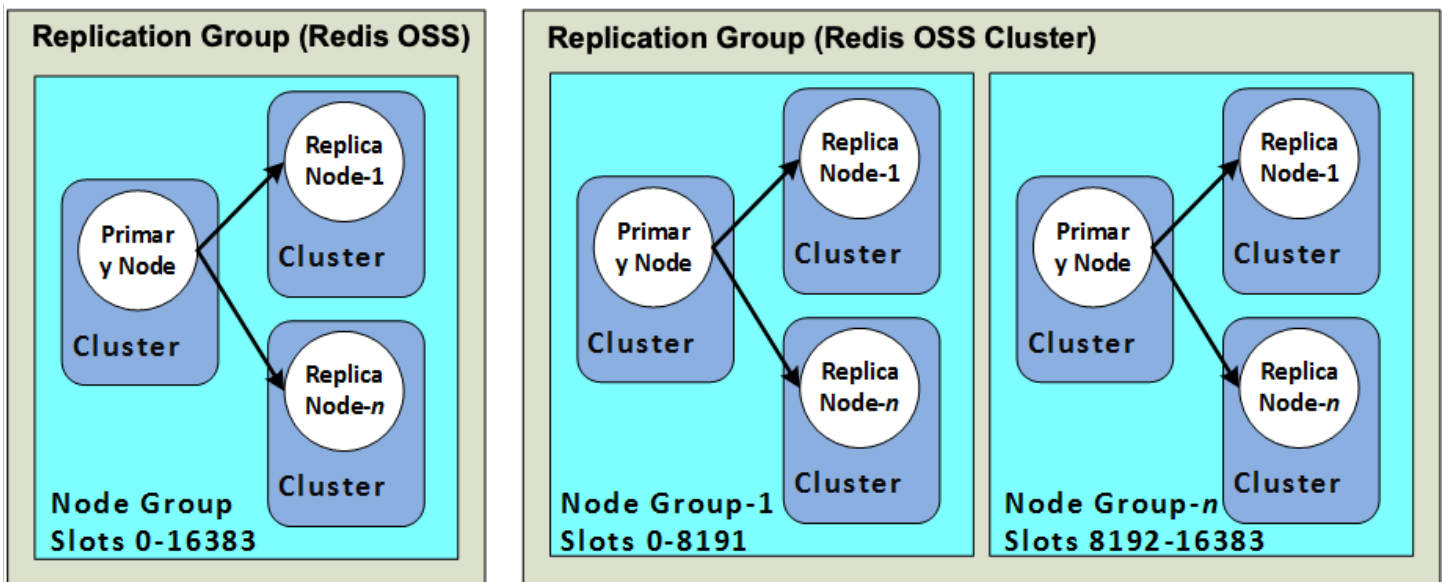
目前，Valkey 或 Redis OSS (啟用叢集模式) 功能有一些限制。

- 您無法將任何複本節點手動提升為主要節點。

## 複寫：停用 Valkey 和 Redis OSS 叢集模式與啟用

從 Valkey 7.2 和 Redis 3.2 OSS版開始，您可以建立兩種不同類型叢集之一 (API/CLI：複寫群組)。Valkey 或 Redis OSS (停用叢集模式) 叢集一律具有單一碎片 (API/CLI：節點群組)，最多具有 5 個僅供讀取複本節點。Valkey 或 Redis OSS (啟用叢集模式) 叢集最多有 500 個碎片，每個碎片都有 1 到 5 個僅供讀取複本節點。





Valkey 或 Redis OSS ( 停用叢集模式 ) ，以及 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集

下表摘要說明 Valkey 或 Redis OSS ( 停用叢集模式 ) 與 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集之間的重要差異。

比較 Valkey 或 Redis OSS ( 停用叢集模式 ) 和 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集

| 功能     | Valkey 或 Redis OSS ( 停用叢集模式 ) | Valkey 或 Redis OSS ( 啟用叢集模式 )                                                                                    |
|--------|-------------------------------|------------------------------------------------------------------------------------------------------------------|
| 可修改    | 是。支援新增和刪除複本節點，以及向上擴展節點類型。     | 有限。如需更多詳細資訊，請參閱「 <a href="#">的版本管理 ElastiCache</a> 」及「 <a href="#">在 Valkey 或 Redis 中擴展叢集 OSS ( 已啟用叢集模式 )</a> 」。 |
| 資料分割   | 否                             | 是                                                                                                                |
| 碎片     | 1                             | 1 到 500                                                                                                          |
| 僅供讀取複本 | 0 到 5                         | 每個碎片 0 到 5 個。                                                                                                    |

| 功能         | Valkey 或 Redis OSS ( 停用叢集模式 )                                                       | Valkey 或 Redis OSS ( 啟用叢集模式 )                                                            |
|------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|            | <p><b>⚠ Important</b></p> <p>若您沒有複本而節點失敗，則可能會發生資料遺失。</p>                            | <p><b>⚠ Important</b></p> <p>若您沒有複本而節點失敗，您可能遺失該碎片中的所有資料。</p>                             |
| Multi-AZ   | <p>是，至少 1 個複本。</p> <p>選用。預設開啟。</p>                                                  | <p>是</p> <p>選用。預設開啟。</p>                                                                 |
| 快照 (備份)    | <p>是，建立單一 .rdb 檔案。</p>                                                              | <p>是，為每個碎片建立唯一 .rdb 檔案。</p>                                                              |
| 還原         | <p>是，使用來自 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的單一 .rdb 檔案。</p>                          | <p>是，使用來自 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的 .rdb 檔案。</p> |
| 支援         | <p>所有 Valkey 和 Redis OSS 版本</p>                                                     | <p>所有 Valkey 版本，以及 Redis OSS 3.2 和更新版本</p>                                               |
| 引擎是否可升級    | <p>是，但包含限制。如需詳細資訊，請參閱<a href="#">的版本管理 ElastiCache</a>。</p>                         | <p>是，但包含限制。如需詳細資訊，請參閱<a href="#">的版本管理 ElastiCache</a>。</p>                              |
| 加密         | <p>3.2.6 版 ( 排程於 EOL，請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。</p> | <p>3.2.6 版 ( 排程於 EOL，請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。</p>      |
| HIPAA 符合資格 | <p>3.2.6 版 ( 排程於 EOL，請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。</p> | <p>3.2.6 版 ( 排程於 EOL，請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。</p>      |

| 功能         | Valkey 或 Redis OSS ( 停用叢集模式 )                                                 | Valkey 或 Redis OSS ( 啟用叢集模式 )                                                 |
|------------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| PCI DSS 合規 | 3.2.6 版 ( 排程於 EOL, 請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。 | 3.2.6 版 ( 排程於 EOL, 請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 和 4.0.10 及更新版本。 |
| 線上重新分片     | N/A                                                                           | 3.2.10 版 ( 排程於 EOL, 請參閱 <a href="#">Redis OSS 版本生命週期結束排程</a> ) 及更新版本。         |

我應該選擇哪一個？

在 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 Redis OSS ( 啟用叢集模式 ) 之間進行選擇時，請考慮下列因素：

- 擴展與分割 - 業務需求變動。您需要為峰值需求佈建，或在需求變更時擴展。Valkey 或 Redis OSS ( 停用叢集模式 ) 支援擴展。您可以透過新增或刪除複本節點來擴展讀取容量，或是透過向上擴展至更大的節點類型來擴展容量。這些操作都需要時間。如需詳細資訊，請參閱 [擴展 Valkey 或 Redis 的複本節點 OSS \( 停用叢集模式 \)](#)。

Valkey 或 Redis OSS ( 啟用叢集模式 ) 支援將資料分割至最多 500 個節點群組。您可以在商業需求變更時動態調整碎片的數量。分割的其中一項優勢，便是您會將負載擴張到更大量的端點，減少尖峰需求期間的存取瓶頸。此外，因為資料可擴張到多個伺服器，您可以容納更大的資料集。如需擴展分割區的資訊，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)。

- 節點大小與節點數量 – 由於 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集只有一個碎片，因此節點類型必須足夠大，以容納叢集的所有資料，並加上必要的額外負荷。另一方面，由於在使用 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集時，您可以將資料分割為數個碎片，因此節點類型可以更小，不過您需要更多類型。如需詳細資訊，請參閱 [選擇您的節點大小](#)。
- 讀取 v. 寫入 – 如果叢集的主要負載是讀取資料的應用程式，您可以透過新增和刪除僅供讀取複本來擴展 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集。不過，僅供讀取複本的數量上限為 5 個。如果

叢集上的負載為寫入密集，您可以從具有多個碎片的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的其他寫入端點中獲益。

無論您選擇實作的叢集類型為何，請務必選擇適合您目前及未來需求的節點類型。

## 搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS

在某些情況下，ElastiCache 使用 Valkey 和 Redis OSS 可能需要取代主要節點；包括特定類型的計劃維護，以及主要節點或可用區域故障的不太可能事件。

此取代會導致叢集出現一些停機情況，但如果啟用異地同步備份，停機時間可降至最低。主節點的角色會自動容錯移轉到其中一個僅供讀取複本。不需要建立新的主節點和佈建，因為 ElastiCache 會以透明方式處理。此容錯移轉及複本提升可確保您能在提升完成時立即繼續寫入新的主要節點。

ElastiCache 也會傳播提升複本的網域名稱服務（DNS）名稱。它會執行這項操作的原因在於，若您的應用程式正在寫入主要端點，您的應用程式中便不需要變更任何端點。如果您是從個別端點讀取，則請確保將提升至主要端點的複本讀取端點變更為新複本的端點。

如果是計劃的節點替換，這些替換因維護更新或自助式更新而啟動，請留意以下事項：

- 對於 ElastiCache Valkey 和 Redis OSS 叢集，計劃的節點替換會在叢集提供傳入寫入請求時完成。
- 對於在 5.0.6 或更新版本引擎上執行且已啟用多可用區域功能的 Valkey 和 Redis OSS 叢集模式停用叢集，計劃節點替換會在叢集提供傳入寫入請求時完成。
- 對於在 4.0.10 或更早的引擎上執行且啟用多可用區域且停用 Valkey 和 Redis OSS 叢集模式的叢集，您可能會注意到與 DNS 更新相關聯的短暫寫入中斷。此中斷最多可能需要幾秒鐘的時間。此程序的速度比重新建立及佈建新的主節點更快（也就是沒有啟用異地同步備份時發生的程序）。

您可以使用 ElastiCache 管理主控台、AWS CLI 或 啟用多可用區 ElastiCache API。

在 Valkey 或 Redis OSS 叢集（在 API 和中 CLI，複寫群組）上啟用 ElastiCache 多可用區可改善容錯能力。當您的叢集的讀取/寫入主要叢集因任何原因變得無法連線或失敗的情況下，這特別有用。僅在每個碎片中具有多個節點的 Valkey 和 Redis OSS 叢集上支援多可用區。

### 主題

- [啟用多個可用區](#)
- [具有異地同步備份回應的故障案例](#)
- [測試自動容錯移轉](#)
- [多可用區域的限制](#)

## 啟用多個可用區

您可以使用 ElastiCache 主控台或 [建立或修改叢集 \( API 或 CLI、複寫群組 \)](#) ElastiCache時 AWS CLI，啟用多可用區域API。

您只能在至少有一個可用僅供讀取複本的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集上啟用多可用區域。沒有僅供讀取複本的叢集無法提供高度可用性或容錯能力。如需建立附帶複寫叢集的資訊，請參閱[建立 Valkey 或 Redis OSS複寫群組](#)。如需將僅供讀取複本新增到附帶複寫叢集的資訊，請參閱[新增 Valkey 或 Redis 的僅供讀取複本 OSS \( 停用叢集模式 \)](#)。

### 主題

- [啟用異地同步備份 \(主控台\)](#)
- [啟用異地同步備份 \(AWS CLI\)](#)
- [啟用異地同步備份 \(ElastiCache API\)](#)

### 啟用異地同步備份 (主控台)

您可以在建立新的 Valkey 或 Redis OSS叢集時，使用 ElastiCache 主控台啟用多可用區域，或修改具有複寫的現有叢集。

根據預設，Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集會啟用多可用區域。

#### Important

ElastiCache 只有當叢集在所有碎片中包含至少一個與主要 不同可用區域中的複本時，才會自動啟用多可用區域。

### 使用 ElastiCache 主控台建立叢集時啟用多可用區域

如需此程序的詳細資訊，請參閱[建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)。請務必確保您有一或多個複本，並啟用多個可用區。

### 在現有叢集上啟用異地同步備份 (主控台)

如需此程序的詳細資訊，請參閱[使用 ElastiCache AWS Management Console](#)修改叢集。

### 啟用異地同步備份 (AWS CLI)

下列程式碼範例使用 AWS CLI 為複寫群組 啟用多可用區redis12。

**⚠ Important**

複寫群組 `redis12` 必須已存在，並且其中必須至少要有一個可用的僅供讀取複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id redis12 \
 --automatic-failover-enabled \
 --multi-az-enabled \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id redis12 ^
 --automatic-failover-enabled ^
 --multi-az-enabled ^
 --apply-immediately
```

此命令的JSON輸出應如下所示。

```
{
 "ReplicationGroup": {
 "Status": "modifying",
 "Description": "One shard, two nodes",
 "NodeGroups": [
 {
 "Status": "modifying",
 "NodeGroupMembers": [
 {
 "CurrentRole": "primary",
 "PreferredAvailabilityZone": "us-west-2b",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis12-001"
 },
 {
 "CurrentRole": "secondary",
 "PreferredAvailabilityZone": "us-west-2c",
 "CacheNodeId": "0002",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis12-001.v5r9dc.0002.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis12-001"
 }
],
 "CacheClusterId": "redis12-001"
 }
],
 "CacheClusterId": "redis12-001"
 }
}
```

```
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2a",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis12-002"
 }
],
 "NodeGroupId": "0001",
 "PrimaryEndpoint": {
 "Port": 6379,
 "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"
 }
}
],
"ReplicationGroupId": "redis12",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabling",
"MultiAZ": "enabled",
"SnapshotWindow": "07:00-08:00",
"SnapshottingClusterId": "redis12-002",
"MemberClusters": [
 "redis12-001",
 "redis12-002"
],
"PendingModifiedValues": {}
}
}
```

如需詳細資訊，請參閱 AWS CLI 命令參考中的下列主題：

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) 在 AWS CLI 命令參考 中。

### 啟用異地同步備份 (ElastiCache API)

下列程式碼範例使用 ElastiCache API 為複寫群組 啟用多可用區 redis12。



**Note**

若要使用此範例，複寫群組 `redis12` 必須已存在，並且其中必須至少要有一個可用的僅供讀取複本。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&AutoFailover=true
&MultiAZEnabled=true
&ReplicationGroupId=redis12
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140401T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱ElastiCache API參考 中的這些主題：

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

## 具有異地同步備份回應的故障案例

在引入多可用區域之前，透過重新建立並重新佈建失敗節點，ElastiCache 偵測到並取代叢集的失敗節點。如果啟用多個可用區，故障的主要節點會容錯移轉至複寫延遲最短的複本。選取的複本會自動提升為主要節點，這比建立並重新佈建新的主要節點更快。此程序通常只需要幾秒鐘，您便能再次寫入叢集。

啟用多可用區時，ElastiCache 會持續監控主要節點的狀態。若主要節點故障，便會根據故障的類型執行以下其中一個動作。

### 主題

- [只有主節點故障的故障案例](#)
- [主節點和某些僅供讀取複本故障的故障案例](#)
- [整個叢集故障的故障案例](#)

### 只有主節點故障的故障案例

若只有主要節點故障，複寫延遲最短的僅供讀取複本便會提升至主要叢集。接著會在與故障的主要節點相同的可用區域中建立並佈建遭取代的僅供讀取複本。

當只有主要節點失敗時，ElastiCache Multi-AZ 會執行下列動作：

1. 失敗的主要節點會離線。
2. 複寫延隔最少的僅供讀取複本會提升為主要節點。

寫入通常可以在提升程序完成時繼續，這通常僅需要數秒鐘。如果您的應用程式正在寫入主要端點，您不需要變更寫入或讀取的端點。ElastiCache 會傳播提升複本DNS的名稱。

3. 啟動及佈建替換用的僅供讀取複本。

替換用的僅供讀取複本會在失敗主要節點所在的可用區域內啟動，維持節點的分佈。

4. 複本會與新的主要節點同步。

新複本可供使用之後，請注意下列效果：

- 主要端點 – 您不需要對應用程式進行任何變更，因為新主要節點DNS的名稱會傳播到主要端點。
- 讀取端點 - 讀取者端點會自動更新，以指向新的複本節點。

如需尋找叢集端點的資訊，請參閱下列主題：

- [尋找 Valkey 或 Redis OSS \( 停用叢集模式 \) 叢集的端點 \( 主控台 \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( AWS CLI \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( ElastiCache API \)](#)

### 主節點和某些僅供讀取複本故障的故障案例

若主要節點及至少一個僅供讀取複本失敗時，複寫延隔最少的可用複本會提升至主要叢集。新的僅供讀取複本也會在失敗節點及提升為主要節點複本的相同可用區域內建立及佈建。

當主要節點和某些僅供讀取複本失敗時，ElastiCache Multi-AZ 會執行下列動作：

1. 失敗的主要節點和失敗的僅供讀取複本會離線。
2. 複寫延隔最少的可用複本會提升為主要節點。

寫入通常可以在提升程序完成時繼續，這通常僅需要數秒鐘。如果您的應用程式正在寫入主要端點，則不需要變更 writes. ElastiCache propagates 的端點DNS名稱。

3. 建立及佈建替換用的複本。

替換用的複本會在失敗節點所在的可用區域內建立，維持節點的分佈。

4. 所有叢集都會和新的主要節點同步。

在新節點可用之後，請對應用程式進行以下變更：

- 主要端點 - 不要對應用程式做任何變更。新主要節點DNS的名稱會傳播到主要端點。
- 讀取端點 - 讀取端點會自動更新，以指向新的複本節點。

如需尋找複寫群組端點的資訊，請參閱下列主題：

- [尋找 Valkey 或 Redis OSS \( 停用叢集模式 \) 叢集的端點 \( 主控台 \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( AWS CLI \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( ElastiCache API \)](#)

## 整個叢集故障的故障案例

若所有項目都失敗，便會在原始節點的相同可用區域內重新建立及佈建所有節點。

在此案例中，叢集內的所有資料都會因為叢集內的每個節點都發生故障而遺失。這種情況相當罕見。

當整個叢集失敗時，ElastiCache Multi-AZ 會執行下列動作：

1. 失敗的主要節點和僅供讀取複本會離線。
2. 建立及佈建替換用的主要節點。
3. 建立及佈建替換用的複本。

替代項目會在失敗節點所在的可用區域內建立，維持節點的分佈。

因為整個叢集失敗，資料會遺失，並且所有新的節點都會從零開始。

因為每個替換用節點都會具備與欲取代節點相同的端點，因此您不需要在應用程式內對任何端點進行變更。

如需尋找複寫群組端點的資訊，請參閱下列主題：

- [尋找 Valkey 或 Redis OSS \( 停用叢集模式 \) 叢集的端點 \( 主控台 \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( AWS CLI \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( ElastiCache API \)](#)

我們建議您在不同可用區域內建立主要節點和僅供讀取複本，以提升您的容錯能力層級。

## 測試自動容錯移轉

啟用自動容錯移轉之後，您可以使用 ElastiCache 主控台、AWS CLI 和 進行測試 ElastiCache API。

在測試時，請注意下列事項：

- 您可以使用此操作，在任何滾動 24 小時期間內，測試最多 15 個碎片（在 ElastiCache API 和中稱為節點群組 AWS CLI）的自動容錯移轉。
- 如果您在不同叢集中的碎片上呼叫此操作（在 API 和中稱為複寫群組 CLI），您可以同時進行呼叫。
- 在某些情況下，您可以在相同 Valkey 或 Redis OSS（啟用叢集模式）複寫群組的不同碎片上多次呼叫此操作。在這種情況下，必須先完成第一個節點取代，才能夠執行後續呼叫。
- 若要判斷節點取代是否已完成，請使用 Amazon ElastiCache 主控台、AWS CLI 或 檢查事件 ElastiCache API。尋找與自動容錯移轉相關的以下事件（根據可能發生的順序列出）：
  1. 複寫群組訊息：Test Failover API called for node group <node-group-id>
  2. 快取叢集訊息：Failover from primary node <primary-node-id> to replica node <node-id> completed
  3. 複寫群組訊息：Failover from primary node <primary-node-id> to replica node <node-id> completed
  4. 快取叢集訊息：Recovering cache nodes <node-id>
  5. 快取叢集訊息：Finished recovery for cache nodes <node-id>

如需詳細資訊，請參閱下列內容：

- [檢視 ElastiCache 事件](#) ElastiCache 使用者指南中的
- [DescribeEvents](#) ElastiCache API 參考中的
- AWS CLI 命令參考中的 [describe-events](#)。
- 這是為了在 ElastiCache 容錯移轉時測試應用程式的行為 API 而設計。並非設計成啟動容錯移轉以解決叢集問題的操作工具。此外，在某些情況下，例如大規模操作事件，AWS 可能會封鎖此 API。

## 主題

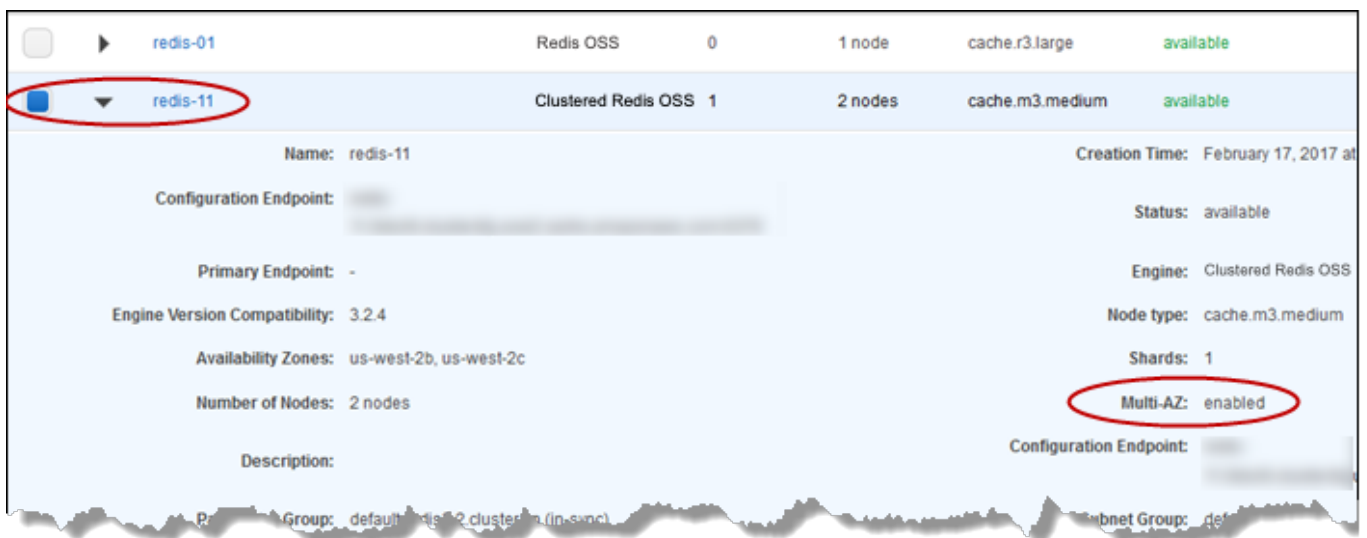
- [使用 測試自動容錯移轉 AWS Management Console](#)
- [使用 測試自動容錯移轉 AWS CLI](#)
- [使用 測試自動容錯移轉 ElastiCache API](#)

## 使用 測試自動容錯移轉 AWS Management Console

使用下列程序，透過主控台測試自動容錯移轉。

### 測試自動容錯移轉

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS。
3. 從叢集清單中，選擇要測試的叢集左側的方塊。此叢集至少需要有一個僅供讀取複本節點。
4. 在 Details (詳細資訊) 區域中，確認此叢集已啟用異地同步備份。若叢集尚未啟用多個可用區，請選擇不同叢集，或是修改此叢集以啟用多個可用區。如需詳細資訊，請參閱 [使用 ElastiCache AWS Management Console](#)。



5. 針對 Valkey 或 Redis OSS ( 停用叢集模式 )，選擇叢集的名稱。

針對 Valkey 或 Redis OSS ( 啟用叢集模式 )，請執行下列動作：

- a. 選擇叢集名稱。
  - b. 在碎片頁面上，針對要在其中測試容錯移轉的碎片 ( API和 中稱為節點群組CLI )，選擇碎片的名稱。
6. 在 Nodes (節點) 頁面上，選擇 Failover Primary (容錯移轉主要節點)。
  7. 選擇 Continue (繼續) 來容錯移轉主要節點，或是 Cancel (取消) 來取消操作而不容錯移轉主要節點。

在容錯移轉程序期間，主控台會繼續將節點的狀態顯示為「可用」。若要追蹤容錯移轉測試的進度，請從主控台導覽窗格選擇 Events (事件)。在 Events (事件) 標籤上，觀察指出您容錯移轉已啟動的事件 (Test Failover API called) 並完成 (Recovery completed)。

## 使用 測試自動容錯移轉 AWS CLI

您可以使用 AWS CLI 操作，在任何啟用多可用區域叢集上測試自動容錯移轉 `test-failover`。

### 參數

- `--replication-group-id` - 必要。要測試的複寫群組 (在主控台上為叢集)。
- `--node-group-id` - 必要項目。您欲測試自動容錯移轉的節點群組名稱。在連續 24 小時期間內，您最多可以測試 15 個節點群組。

下列範例使用在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 `redis00-0003` 中的節點群組上 AWS CLI 測試自動容錯移轉 `redis00`。

### Example 測試自動容錯移轉

若為 Linux、macOS 或 Unix :

```
aws elasticache test-failover \
 --replication-group-id redis00 \
 --node-group-id redis00-0003
```

針對 Windows :

```
aws elasticache test-failover ^
 --replication-group-id redis00 ^
 --node-group-id redis00-0003
```

上述命令的輸出會與以下內容相似。

```
{
 "ReplicationGroup": {
 "Status": "available",
 "Description": "1 shard, 3 nodes (1 + 2 replicas)",
 "NodeGroups": [

```

```
{
 "Status": "available",
 "NodeGroupMembers": [
 {
 "CurrentRole": "primary",
 "PreferredAvailabilityZone": "us-west-2c",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis1x3-001"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2a",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis1x3-002"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2b",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address":
"redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "redis1x3-003"
 }
],
 "NodeGroupId": "0001",
 "PrimaryEndpoint": {
 "Port": 6379,
 "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
 }
},
```



```
 "ClusterEnabled": false,
 "ReplicationGroupId": "redis1x3",
 "SnapshotRetentionLimit": 1,
 "AutomaticFailover": "enabled",
 "MultiAZ": "enabled",
 "SnapshotWindow": "11:30-12:30",
 "SnapshottingClusterId": "redis1x3-002",
 "MemberClusters": [
 "redis1x3-001",
 "redis1x3-002",
 "redis1x3-003"
],
 "CacheNodeType": "cache.m3.medium",
 "DataTiering": "disabled",
 "PendingModifiedValues": {}
 }
}
```

若要追蹤容錯移轉的進度，請使用 AWS CLI `describe-events` 操作。

如需詳細資訊，請參閱下列內容：

- AWS CLI 命令參考中的 [test-failover](#)。
- AWS CLI 命令參考中的 [describe-events](#)。

## 使用 測試自動容錯移轉 ElastiCache API

您可以使用 `TestFailover` 操作，在啟用多可用區域 ElastiCache API 的任何叢集上測試自動容錯移轉 `TestFailover`。

### 參數

- `ReplicationGroupId` - 必要項目。要測試的複寫群組 (在主控台上為叢集)。
- `NodeGroupId` - 必要項目。您欲測試自動容錯移轉的節點群組名稱。在連續 24 小時期間內，您最多可以測試 15 個節點群組。

以下範例會在複寫群組 (主控台上為叢集) `redis00` 中的節點群組 `redis00-0003` 上測試自動容錯移轉。

## Example 測試自動容錯移轉

```
https://elasticache.us-west-2.amazonaws.com/
?Action=TestFailover
&NodeGroupId=redis00-0003
&ReplicationGroupId=redis00
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140401T192317Z
&X-Amz-Credential=<credential>
```

若要追蹤容錯移轉的進度，請使用 ElastiCache DescribeEvents API 操作。

如需詳細資訊，請參閱下列內容：

- [TestFailover](#) ElastiCache API 參考中的
- [DescribeEvents](#) ElastiCache API 參考中的

### 多可用區域的限制

請注意多可用區下列限制：

- Valkey 和 Redis 2.8.6 版及更新 OSS 版本支援多可用區。
- T1 節點類型不支援多可用區。
- Valkey 和 Redis OSS 複寫是非同步的。因此，當主要節點容錯移轉至複本時，可能會因複寫延遲而導致一小部分的資料遺失。

在選擇要提升至主要複本的複本時，會 ElastiCache 選擇具有最小複寫延遲的複本。換句話說，它會選擇最新的複本。這麼做有助於減少遺失的資料量。具有最少複寫延遲的複本可以和失敗的主要節點位於相同可用區域或不同可用區域中。

- 當您在停用叢集模式的 Valkey 或 Redis OSS 叢集上，手動將僅供讀取複本提升為主要複本時，您只能在停用多可用區和自動容錯移轉時執行此操作。若要將僅供讀取複本節點提升為主要節點，請採取下列步驟：

1. 停用叢集上的多個可用區。

2. 停用叢集上的自動容錯移轉。您可以清除複寫群組的自動容錯移轉核取方塊，透過主控台執行此操作。您也可以呼叫 `ModifyReplicationGroup` 操作 `false` 時，將 `AutomaticFailoverEnabled` 屬性設定為 `false`，AWS CLI 藉此使用執行此操作。
  3. 將僅供讀取複本提升為主要節點。
  4. 重新啟用多個可用區。
- ElastiCache (RedisOSS) 多可用區和僅附加檔案 (AOF) 是互斥的。若您啟用其中一項，便無法啟用另外一項。
  - 節點故障很少會是因為整個可用區域失敗所造成。在此情況下，只有在可用區域已備份時，才會建立取代失敗主要節點的複本。例如，假設有一個複寫群組，其中主要節點位於 AZ-a，複本位於 AZ-b 及 AZ-c。若主要節點失敗，複寫延隔最少的複本便會提升至主要叢集。然後，只有在 AZ-a 備份且可用時，才能在 AZ-a 中 ElastiCache 建立新的複本 (失敗的主要位置)。
  - 由客戶初始化的主要節點重新開機不會觸發自動容錯移轉。其他重新開機和故障會觸發自動容錯移轉。
  - 主要節點重新開機時，便會在回到線上時清除資料。當僅供讀取複本看到已清除的主要叢集時，它們便會清除資料複本，造成資料遺失。
  - 在提升僅供讀取複本之後，其他複本便會和新的主要節點同步。初始化同步後，複本的內容會被刪除，而且它們會從新的主要節點同步資料。此同步程序會造成短暫的中斷，在此期間無法存取複本。此同步程序也會在與複本進行同步時，於主要節點上造成暫時性的負載增加。此行為是 Valkey 和 Redis 的原生行為 OSS，對多可用區而言並非唯一 ElastiCache。如需此行為的詳細資訊，請參閱 Valkey 網站上的 [複寫](#)。

#### Important

對於 Valkey 7.2.6 及更新版本或 Redis 2.8.22 及更新 OSS 版本，您無法建立外部複本。對於 2.8.22 之前的 Redis OSS 版本，建議您不要將外部複本連接至已啟用多可用區域功能的 ElastiCache 叢集。此不支援的組態可能會產生問題，ElastiCache 導致無法正確執行容錯移轉和復原。若要將外部複本連接至 ElastiCache 叢集，請先確定未啟用多可用區，再進行連線。

## 同步與備份的實作方式

所有支援的 Valkey 和 Redis 版本都OSS支援主要節點和複本節點之間的備份和同步。不過，備份和同步的實作方式會因版本而異。

### Redis 2.8.22 OSS版及更新版本

Redis OSS複寫，在 2.8.22 版及更新版本中，有兩種方法可供選擇。如需詳細資訊，請參閱 [2.8.22 之前的 Redis OSS版本](#) 和 [快照和還原](#)。

在無分支過程期間，若寫入負載龐大，對叢集進行的寫入會發生延遲，確保您不會累積過多變更並因此防止快照。

### 2.8.22 之前的 Redis OSS版本

2.8.22 之前的 Redis OSS 備份和同步是一個三步驟程序。

1. 分支，並在背景程序中，將叢集資料序列化到磁碟。這會建立 point-in-time快照。
2. 在前景中，將變更日誌累積在用戶端輸出緩衝區中。

#### Important

若變更日誌超過用戶端輸出緩衝區大小，則備份或同步便會失敗。如需詳細資訊，請參閱[確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)。

3. 最後，將傳輸快取資料，以及變更日誌傳送到複本節點。

## 建立 Valkey 或 Redis OSS 複寫群組

您有以下選項，可用來建立包含複本節點的叢集。當您已有可用的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，而該叢集與具有要用作主要節點的複本的任何叢集沒有關聯時，就會套用一個叢集。如果您需要在叢集中建立主要節點及僅供讀取複本，則適用另一個選項。目前，必須從頭開始建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集。

### 選項 1 : [使用現有叢集建立複寫群組](#)

使用此選項可運用現有的單節點 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集。您需將此現有節點指定為新叢集中的主節點，然後將 1 到 5 個僅供讀取複本分別新增到叢集。若現有叢集正在作用中，僅供讀取複本便會在建立時與其同步。請參閱 [使用現有叢集建立複寫群組](#)。

#### Important

您無法使用現有叢集建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集。若要使用 ElastiCache 主控台建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( API/CLI : 複寫群組 )，請參閱 [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)。

### 選項 2 : [從頭開始建立 Valkey 或 Redis OSS 複寫群組](#)

如果您尚未擁有可用的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，以用作叢集的主要節點，或者如果您想要建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集，請使用此選項。請參閱 [從頭開始建立 Valkey 或 Redis OSS 複寫群組](#)。

## 使用現有叢集建立複寫群組

可用的叢集是現有的單節點 Valkey 或 Redis OSS 叢集。目前，Valkey 或 Redis OSS ( 啟用叢集模式 ) 不支援使用可用的單節點叢集建立具有複本的叢集。如果您想要建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集，請參閱 [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)。

只有在您具有 Valkey 或 Redis OSS ( 停用叢集模式 ) 單節點叢集時，才能使用下列程序。此叢集的節點會成為新叢集中的主要節點。如果您沒有可以用作新叢集主要的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，請參閱 [從頭開始建立 Valkey 或 Redis OSS 複寫群組](#)。

使用現有叢集建立複寫群組 ( 主控台 )

請參閱主題：[使用 ElastiCache AWS Management Console](#)。

使用可用的 Valkey 或 Redis OSS 快取叢集建立複寫群組 ( AWS CLI )

在使用時，為主要使用可用的 Valkey 或 Redis OSS 快取叢集時，有兩個步驟可建立具有僅供讀取複本的複寫群組 AWS CLI。

使用時，AWS CLI 您會建立複寫群組，指定可用的獨立節點作為叢集的主要節點，`--primary-cluster-id` 以及使用 CLI 命令在叢集中所需的節點數量：`create-replication-group`。包含以下參數。

`--replication-group-id`

您要建立的複寫群組名稱。此參數的值會用來做為新增節點名稱的基礎，並在 `--replication-group-id` 的結尾加上 3 位數的序號。例如：`sample-repl-group-001`。

Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

`--replication-group-description`

複寫群組的說明。

`--num-node-groups`

此叢集中的您希望的節點數量。此值包含主要節點。此參數的最大值為 6。

## --primary-cluster-id

您要成為此複寫群組中主要節點的可用 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集節點的名稱。

下列命令 `sample-repl-group` 會使用可用的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集 `redis01` 作為複寫群組的主要節點來建立複寫群組。它會建立 2 個新節點 (皆為僅供讀取複本)。 `redis01` 的設定 (即參數群組、安全群組、節點類型、引擎版本等) 會套用到所有複寫群組中的節點。

若為 Linux、macOS 或 Unix :

```
aws elasticache create-replication-group \
 --replication-group-id sample-repl-group \
 --replication-group-description "demo cluster with replicas" \
 --num-cache-clusters 3 \
 --primary-cluster-id redis01
```

針對 Windows :

```
aws elasticache create-replication-group ^
 --replication-group-id sample-repl-group ^
 --replication-group-description "demo cluster with replicas" ^
 --num-cache-clusters 3 ^
 --primary-cluster-id redis01
```

如需您可能想要使用的其他資訊和參數，請參閱 AWS CLI 主題 [create-replication-group](#)。

接下來，將僅供讀取複本新增到複寫群組

在建立複寫群組之後，使用 `create-cache-cluster` 命令將一到五個僅供讀取複本新增到其中，並確認包含以下參數。

## --cache-cluster-id

您欲新增到複寫群組的叢集名稱。

叢集命名限制條件如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。

- 不能連續包含兩個連字號。
- 結尾不能是連字號。

`--replication-group-id`

您要將此快取叢集新增到其中的複寫群組名稱。

針對每個您欲新增到複寫群組的僅供讀取複本重複此命令，並只變更 `--cache-cluster-id` 參數的值。

#### Note

請記住，複寫群組無法擁有超過五個僅供讀取複本。嘗試將僅供讀取複本新增到已有五個僅供讀取複本的複寫群組，會造成操作失敗。

以下程式碼會將僅供讀取複本 `my-replica01` 新增到複寫群組 `sample-repl-group`。主要叢集的設定 (參數群組、安全群組、節點類型等) 會在新增到複寫群組時套用到節點。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \
 --cache-cluster-id my-replica01 \
 --replication-group-id sample-repl-group
```

針對 Windows：

```
aws elasticache create-cache-cluster ^
 --cache-cluster-id my-replica01 ^
 --replication-group-id sample-repl-group
```

此命令的輸出看起來會與以下內容相似。

```
{
 "ReplicationGroup": {
 "Status": "creating",
 "Description": "demo cluster with replicas",
 "ClusterEnabled": false,
 "ReplicationGroupId": "sample-repl-group",
```



```
 "SnapshotRetentionLimit": 1,
 "AutomaticFailover": "disabled",
 "SnapshotWindow": "00:00-01:00",
 "SnapshottingClusterId": "redis01",
 "MemberClusters": [
 "sample-repl-group-001",
 "sample-repl-group-002",
 "redis01"
],
 "CacheNodeType": "cache.m4.large",
 "DataTiering": "disabled",
 "PendingModifiedValues": {}
}
}
```

如需詳細資訊，請參閱 AWS CLI 主題：

- [create-replication-group](#)
- [modify-replication-group](#)

將複本新增至獨立 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集 ( ElastiCache API )

使用時 ElastiCache API，您會建立複寫群組，指定可用的獨立節點作為叢集的主要節點，PrimaryClusterId 以及使用 CLI 命令指定叢集中想要的節點數量 CreateReplicationGroup。包含以下參數。

### ReplicationGroupId

您要建立的複寫群組名稱。此參數的值會用來做為新增節點名稱的基礎，並在 ReplicationGroupId 的結尾加上 3 位數的序號。例如：sample-repl-group-001。

Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

### ReplicationGroupDescription

包含複本的叢集說明。

## NumCacheClusters

此叢集中的您希望的節點數量。此值包含主要節點。此參數的最大值為 6。

## PrimaryClusterId

您要成為此叢集中主要節點的可用 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的名稱。

下列命令 `sample-repl-group` 會使用可用的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集 `redis01` 作為複寫群組的主要節點，建立具有複本的叢集。它會建立 2 個新節點 (皆為僅供讀取複本)。 `redis01` 的設定 (即參數群組、安全群組、節點類型、引擎版本等) 會套用到所有複寫群組中的節點。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateReplicationGroup
&Engine=redis
&EngineVersion=6.0
&ReplicationGroupDescription=Demo%20cluster%20with%20replicas
&ReplicationGroupId=sample-repl-group
&PrimaryClusterId=redis01
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需其他資訊，請參閱 ElastiCache APL 主題：

- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

接下來，將僅供讀取複本新增到複寫群組

在建立複寫群組之後，使用 `CreateCacheCluster` 操作將一到五個僅供讀取複本新增到其中，並確認包含以下參數。

## CacheClusterId

您欲新增到複寫群組的叢集名稱。

叢集命名限制條件如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

## ReplicationGroupId

您要將此快取叢集新增到其中的複寫群組名稱。

針對每個您欲新增到複寫群組的僅供讀取複本重複此操作，並只變更 CacheClusterId 參數的值。

以下程式碼會將僅供讀取複本 (myReplica01) 新增到複寫群組 (myReplGroup)。主要叢集的設定 (參數群組、安全群組、節點類型等) 會在新增到複寫群組時套用到節點。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheCluster
&CacheClusterId=myReplica01
&ReplicationGroupId=myReplGroup
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request
&X-Amz-Date=20150202T170651Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=[signature-value]
```

如需您可能想要使用的其他資訊和參數，請參閱 ElastiCache API 主題 [CreateCacheCluster](#)。

## 從頭開始建立 Valkey 或 Redis OSS 複寫群組

接下來，您可以了解如何建立 Valkey 或 Redis OSS 複寫群組，而不使用現有的 Valkey 或 Redis OSS 叢集作為主要群組。您可以使用 ElastiCache 主控台 AWS CLI、或從頭建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 ElastiCache Redis OSS ( 啟用叢集模式 ) 複寫群組 API。

在繼續之前，請先決定您要建立 Valkey 或 Redis OSS ( 停用叢集模式 )，還是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。如需有關決定的指導，請參閱 [複寫：停用 Valkey 和 Redis OSS 叢集模式與啟用](#)。

### 主題

- [從頭開始建立 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組](#)
- [從頭開始在 Valkey 或 Redis OSS \( 已啟用叢集模式 \) 中建立複寫群組](#)

## 從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組

您可以使用主控台、或，從頭 ElastiCache 開始建立 Valkey AWS CLI 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ElastiCache API。Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組一律有一個節點群組、一個主要叢集，以及最多五個僅供讀取複本。Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組不支援分割您的資料。

### Note

節點/碎片的限制可增加至每一叢集 500 個。若要請求增加配額，請參閱 [AWS 服務配額](#)，並在請求中加入執行個體類型。

若要從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，請採取下列其中一種方法：

從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ( AWS CLI )

下列程序會使用 建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 AWS CLI。

當您從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組時，只需對 AWS CLI `create-replication-group` 命令進行單一呼叫，即可建立複寫群組及其所有節點。包含以下參數。

`--replication-group-id`

您要建立的複寫群組名稱。

Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

`--replication-group-description`

複寫群組的說明。

`--num-cache-clusters`

您希望與此複寫群組一同建立的節點數量，其為主要節點及僅供讀取複本數的總和。

若您啟用異地同步備份 (`--automatic-failover-enabled`)，`--num-cache-clusters` 的值必須至少為 2。

## --cache-node-type

複寫群組中每個節點的節點類型。

ElastiCache 支援下列節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需每個節點類型效能詳細資訊的詳細資訊，請參閱 [Amazon EC2執行個體類型](#)。

## --data-tiering-enabled

如果您使用的是 r6gd 節點類型，應設定此參數。如果您不想要資料分層，應設為 `--no-data-tiering-enabled`。如需詳細資訊，請參閱 [中的資料分層 ElastiCache](#)。

## --cache-parameter-group

指定對應到您引擎版本的參數群組。如果您執行 Redis OSS 3.2.4 或更新版本，請指定 `default.redis3.2` 參數群組或衍生自的參數群組 `default.redis3.2`，以建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組。如需詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#)。

## --網路類型

`ipv4`、`ipv6` 或 `dual-stack`。若您選擇雙堆疊，則必須將 `--IpDiscovery` 參數設定為 `ipv4` 或 `ipv6`。

## --engine

`redis`

## --engine-version

若要擁有最豐富的功能組，請選擇最新的引擎版本。

節點的名稱會透過在複寫群組名稱後方加上 `-00#` 來衍生取得。例如，若使用複寫群組名稱 `myReplGroup`，則主要節點的名稱就會是 `myReplGroup-001`，僅供讀取複本的名稱則為 `myReplGroup-002` 到 `myReplGroup-006`。

若您希望在此複寫群組上啟用傳輸中及待用加密，請新增 `--transit-encryption-enabled` 或 `--at-rest-encryption-enabled` 參數或同時新增兩者，並符合以下條件。

- 您的複寫群組必須執行 Redis 3.2.6 版或 4.0.10 OSS版。
- 複寫群組必須在 Amazon 中建立VPC。
- 您也必須包含 `--cache-subnet-group` 參數。

- 您也必須將參數 `--auth-token` 包含客戶指定的 AUTH 字符串值（密碼），才能在此複寫群組上執行操作。

下列操作會建立 `sample-repl-group` 具有三個節點的 Valkey 或 Redis OSS（停用叢集模式）複寫群組，包括主要和兩個複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id sample-repl-group \
 --replication-group-description "Demo cluster with replicas" \
 --num-cache-clusters 3 \
 --cache-node-type cache.m4.large \
 --engine redis
```

針對 Windows：

```
aws elasticache create-replication-group ^
 --replication-group-id sample-repl-group ^
 --replication-group-description "Demo cluster with replicas" ^
 --num-cache-clusters 3 ^
 --cache-node-type cache.m4.large ^
 --engine redis
```

此命令的輸出看起來會與以下內容相似。

```
{
 "ReplicationGroup": {
 "Status": "creating",
 "Description": "Demo cluster with replicas",
 "ClusterEnabled": false,
 "ReplicationGroupId": "sample-repl-group",
 "SnapshotRetentionLimit": 0,
 "AutomaticFailover": "disabled",
 "SnapshotWindow": "01:30-02:30",
 "MemberClusters": [
 "sample-repl-group-001",
 "sample-repl-group-002",
 "sample-repl-group-003"
],
 "CacheNodeType": "cache.m4.large",
```

```
 "DataTiering": "disabled",
 "PendingModifiedValues": {}
 }
}
```

如需您可能想要使用的其他資訊和參數，請參閱 AWS CLI 主題 [create-replication-group](#)。

從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ( ElastiCache API )

下列程序會使用 建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ElastiCache API。

當您從頭開始建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組時，您會透過對 ElastiCache API `CreateReplicationGroup` 操作的單一呼叫來建立複寫群組及其所有節點。包含以下參數。

#### ReplicationGroupId

您要建立的複寫群組名稱。

Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

#### ReplicationGroupDescription

您複寫群組的說明。

#### NumCacheClusters

您希望與此複寫群組一同建立的節點總數，其為主要節點及僅供讀取複本數的總和。

若您啟用異地同步備份 (`AutomaticFailoverEnabled=true`)，`NumCacheClusters` 的值必須至少為 2。

#### CacheNodeType

複寫群組中每個節點的節點類型。

ElastiCache 支援下列節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需每個節點類型效能詳細資訊的詳細資訊，請參閱 [Amazon EC2 執行個體類型](#)。



## --data-tiering-enabled

如果您使用的是 r6gd 節點類型，應設定此參數。如果您不想要資料分層，應設為 `--no-data-tiering-enabled`。如需詳細資訊，請參閱[中的資料分層 ElastiCache](#)。

## CacheParameterGroup

指定對應到您引擎版本的參數群組。如果您執行 Redis OSS 3.2.4 或更新版本，請指定 `default.redis3.2` 參數群組或衍生自的參數群組 `default.redis3.2`，以建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組。如需詳細資訊，請參閱[Valkey 和 Redis OSS 參數](#)。

## --網路類型

`ipv4`、`ipv6` 或 `dual-stack`。若您選擇雙堆疊，則必須將 `--IpDiscovery` 參數設定為 `ipv4` 或 `ipv6`。

## 引擎

`redis`

## EngineVersion

`6.0`

節點的名稱會透過在複寫群組名稱後方加上 `-00#` 來衍生取得。例如，若使用複寫群組名稱 `myReplGroup`，則主要節點的名稱就會是 `myReplGroup-001`，僅供讀取複本的名稱則為 `myReplGroup-002` 到 `myReplGroup-006`。

若您希望在此複寫群組上啟用傳輸中及待用加密，請新增 `TransitEncryptionEnabled=true` 或 `AtRestEncryptionEnabled=true` 參數或同時新增兩者，並符合以下條件。

- 您的複寫群組必須執行 Redis 3.2.6 版或 4.0.10 OSS版。
- 複寫群組必須在 Amazon 中建立VPC。
- 您也必須包含 `CacheSubnetGroup` 參數。
- 您也必須將參數 `AuthToken` 包含客戶指定的 AUTH 字符串值 ( 密碼 )，才能在此複寫群組上執行操作。

下列操作會建立 `myReplGroup` 具有三個節點的 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，包括主要和兩個複本。

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=CreateReplicationGroup
&CacheNodeType=cache.m4.large
&CacheParameterGroup=default.redis6.x
&Engine=redis
&EngineVersion=6.0
&NumCacheClusters=3
&ReplicationGroupDescription=test%20group
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需您可能想要使用的其他資訊和參數，請參閱 ElastiCache API主題 [CreateReplicationGroup](#).

從頭開始在 Valkey 或 Redis OSS ( 已啟用叢集模式 ) 中建立複寫群組

您可以使用 ElastiCache 主控台、或 建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( API/CLI : 複寫群組 AWS CLI ) ElastiCache API。Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組具有 1 到 500 個碎片 ( API/CLI : 節點群組 )、每個碎片中的主要節點，以及每個碎片中最多 5 個僅供讀取複本。您可以建立具有較高數量碎片和較低數量複本的叢集，每一叢集最高總計為 90 個節點。此叢集組態的範圍可以從 90 個碎片和 0 個複本到 15 個碎片和 5 個複本，這是允許的複本最大數量。

如果 Valkey 或 Redis OSS 引擎版本為 5.0.6 或更新版本，節點或碎片限制可以提高到每個叢集最多 500 個。例如，您可以選擇設定具有 500 個節點的叢集，並容許碎片在 83 個 ( 每個碎片一個主要版本和 5 個複本 ) 到 500 個 ( 單一主要版本並且沒有複本 ) 之間變化。請確保有足夠的可用 IP 地址來容納增加的數量。常見陷阱包括子網路群組中的子網路 CIDR 太小，範圍太小，或者子網路被其他叢集共用且大量使用。如需詳細資訊，請參閱 [建立子網路群組](#)。

對於 5.0.6 以下的版本，每個叢集的限制為 250 個。

若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

在 Valkey 或 Redis 中建立叢集 OSS ( 已啟用叢集模式 )

- [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)
- [從頭開始建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [從頭開始在 Valkey 或 Redis OSS \( 啟用叢集模式 \) 中建立複寫群組 \( ElastiCache API \)](#)

建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( 主控台 )

若要建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集，請參閱 [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)。請務必啟用叢集模式 (Cluster Mode enabled (Scale Out) (啟用叢集模式 (向外擴展)))，並在其中指定至少兩個碎片及一個複本節點。

從頭開始建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組 ( AWS CLI )

下列程序會使用 建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組 AWS CLI。

當您從頭開始建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組時，只需對 AWS CLI `create-replication-group` 命令進行單一呼叫，即可建立複寫群組及其所有節點。包含以下參數。

`--replication-group-id`

您要建立的複寫群組名稱。

Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

--replication-group-description

複寫群組的說明。

--cache-node-type

複寫群組中每個節點的節點類型。

ElastiCache 支援下列節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需每個節點類型效能詳細資訊的詳細資訊，請參閱 [Amazon EC2 執行個體類型](#)。

--data-tiering-enabled

如果您使用的是 r6gd 節點類型，應設定此參數。如果您不想要資料分層，應設為 --no-data-tiering-enabled。如需詳細資訊，請參閱 [中的資料分層 ElastiCache](#)。

--cache-parameter-group

指定 default.redis6.x.cluster.on 參數群組或從 衍生的參數群組 default.redis6.x.cluster.on，以建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。如需詳細資訊，請參閱 [Redis OSS 6.x 參數變更](#)。

--engine

redis

--engine-version

3.2.4

--num-node-groups

此複寫群組中節點群組的數量。有效值為 1 到 500。

**Note**

節點/碎片的限制可增加至每一叢集 500 個。若要請求增加限制，請參閱 [AWS 服務配額](#)，並選取限制類型「每一執行個體類型每一叢集的節點數」。

**--replicas-per-node-group**

每個節點群組中複本節點的數量。有效值為 0 到 5。

**--網路類型**

ipv4、ipv6 或 dual-stack。若您選擇雙堆疊，則必須將 `--IpDiscovery` 參數設定為 ipv4 或 ipv6。

若您希望在此複寫群組上啟用傳輸中及待用加密，請新增 `--transit-encryption-enabled` 或 `--at-rest-encryption-enabled` 參數或同時新增兩者，並符合以下條件。

- 您的複寫群組必須執行 Redis 3.2.6 版或 4.0.10 OSS版。
- 複寫群組必須在 Amazon 中建立VPC。
- 您也必須包含 `--cache-subnet-group` 參數。
- 您也必須將參數 `--auth-token` 包含客戶指定的AUTH字符串值（密碼），才能在此複寫群組上執行操作。

下列操作會建立 `sample-repl-group` 具有三個節點群組/碎片（`--`）的 Valkey 或 Redis OSS（啟用叢集模式）複寫群組 `num-node-groups`，每個節點都有三個節點，一個主要和兩個僅供讀取複本（`--replicas-per-node-group`）。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id sample-repl-group \
 --replication-group-description "Demo cluster with replicas" \
 --num-node-groups 3 \
 --replicas-per-node-group 2 \
 --cache-node-type cache.m4.large \
 --engine redis \
 --security-group-ids SECURITY_GROUP_ID \
 --cache-subnet-group-name SUBNET_GROUP_NAME>
```

針對 Windows :

```
aws elasticache create-replication-group ^
--replication-group-id sample-repl-group ^
--replication-group-description "Demo cluster with replicas" ^
--num-node-groups 3 ^
--replicas-per-node-group 2 ^
--cache-node-type cache.m4.large ^
--engine redis ^
--security-group-ids SECURITY_GROUP_ID ^
--cache-subnet-group-name SUBNET_GROUP_NAME>
```

上述命令會產生以下輸出。

```
{
 "ReplicationGroup": {
 "Status": "creating",
 "Description": "Demo cluster with replicas",
 "ReplicationGroupId": "sample-repl-group",
 "SnapshotRetentionLimit": 0,
 "AutomaticFailover": "enabled",
 "SnapshotWindow": "05:30-06:30",
 "MemberClusters": [
 "sample-repl-group-0001-001",
 "sample-repl-group-0001-002",
 "sample-repl-group-0001-003",
 "sample-repl-group-0002-001",
 "sample-repl-group-0002-002",
 "sample-repl-group-0002-003",
 "sample-repl-group-0003-001",
 "sample-repl-group-0003-002",
 "sample-repl-group-0003-003"
],
 "PendingModifiedValues": {}
 }
}
```

當您從頭開始建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組時，您可以使用 `--node-group-configuration` 參數來設定叢集中的每個碎片，如下列設定兩個節點群組 ( 主控台：碎片 )

的範例所示。第一個碎片有兩個節點：一個主要節點及一個僅供讀取複本。第二個碎片有三個節點：一個主要節點及兩個僅供讀取複本。

### --node-group-configuration

每個節點群組的組態。--node-group-configuration 參數由以下欄位組成。

- PrimaryAvailabilityZone - 此節點群組主節點所在的可用區域。如果省略此參數，ElastiCache 請選擇主要節點的可用區域。

範例：us-west-2a。

- ReplicaAvailabilityZones - 僅供讀取複本所在可用區域的逗點分隔清單。此清單中的可用區域數量必須等於 ReplicaCount 的值。如果省略此參數，ElastiCache 請選擇複本節點的可用區域。

範例："us-west-2a,us-west-2b,us-west-2c"

- ReplicaCount - 此節點群組中複本節點的數量。
- Slots - 指定節點群組 Keyspace 的字串。此字串格式為 startKey-endKey。如果省略此參數，會在節點群組之間平均 ElastiCache 分配金鑰。

範例："0-4999"

下列操作會建立new-group具有兩個節點群組/碎片 OSS ( ) 的 Valkey 或 Redis ( 啟用叢集模式 ) 複寫群組--num-node-groups。與上述範例不同，每個節點群組的設定都會與其他節點群組不同 (--node-group-configuration)。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id new-group \
 --replication-group-description "Sharded replication group" \
 --engine redis \
 --snapshot-retention-limit 8 \
 --cache-node-type cache.m4.medium \
 --num-node-groups 2 \
 --node-group-configuration \
 "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-
east-1c',ReplicaAvailabilityZones='us-east-1b'" \
```

```
"ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

針對 Windows :

```
aws elasticache create-replication-group ^
--replication-group-id new-group ^
--replication-group-description "Sharded replication group" ^
--engine redis ^
--snapshot-retention-limit 8 ^
--cache-node-type cache.m4.medium ^
--num-node-groups 2 ^
--node-group-configuration \
 "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \
 "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

上述操作會產生以下輸出。

```
{
 "ReplicationGroup": {
 "Status": "creating",
 "Description": "Sharded replication group",
 "ReplicationGroupId": "rc-rg",
 "SnapshotRetentionLimit": 8,
 "AutomaticFailover": "enabled",
 "SnapshotWindow": "10:00-11:00",
 "MemberClusters": [
 "rc-rg-0001-001",
 "rc-rg-0001-002",
 "rc-rg-0002-001",
 "rc-rg-0002-002",
 "rc-rg-0002-003"
],
 "PendingModifiedValues": {}
 }
}
```

如需您可能想要使用的其他資訊和參數，請參閱 AWS CLI 主題 [create-replication-group](#)。



從頭開始在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 中建立複寫群組 ( ElastiCache API )

下列程序會使用 建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組 ElastiCache API。

當您從頭開始建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組時，您會透過對 ElastiCache `APICreateReplicationGroup` 操作的單一呼叫來建立複寫群組及其所有節點。包含以下參數。

#### ReplicationGroupId

您要建立的複寫群組名稱。

Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組命名限制如下：

- 必須包含 1-40 個英數字元或連字號。
- 必須以字母開頭。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。

#### ReplicationGroupDescription

複寫群組的說明。

#### NumNodeGroups

您希望和此複寫群組一同建立的節點群組數量。有效值為 1 到 500。

#### ReplicasPerNodeGroup

每個節點群組中複本節點的數量。有效值為 1 到 5。

#### NodeGroupConfiguration

每個節點群組的組態。NodeGroupConfiguration 參數由以下欄位組成。

- PrimaryAvailabilityZone - 此節點群組主節點所在的可用區域。如果省略此參數，ElastiCache 請選擇主要節點的可用區域。

範例：us-west-2a。

- ReplicaAvailabilityZones - 僅供讀取複本所在可用區域的清單。此清單中的可用區域數量必須等於 ReplicaCount 的值。如果省略此參數，ElastiCache 請選擇複本節點的可用區域。
- ReplicaCount - 此節點群組中複本節點的數量。
- Slots - 指定節點群組 Keyspace 的字串。此字串格式為 startKey-endKey。如果省略此參數，會在節點群組之間平均 ElastiCache 分配金鑰。

範例："0-4999"

## CacheNodeType

複寫群組中每個節點的節點類型。

ElastiCache 支援下列節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需每個節點類型效能詳細資訊的詳細資訊，請參閱 [Amazon EC2執行個體類型](#)。

### --data-tiering-enabled

如果您使用的是 r6gd 節點類型，應設定此參數。如果您不想要資料分層，應設為 --no-data-tiering-enabled。如需詳細資訊，請參閱 [中的資料分層 ElastiCache](#)。

## CacheParameterGroup

指定 default.redis6.x.cluster.on 參數群組或從 衍生的參數群組 default.redis6.x.cluster.on，以建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。如需詳細資訊，請參閱 [Redis OSS 6.x 參數變更](#)。

### --網路類型

ipv4、ipv 或 dual-stack。若您選擇雙堆疊，則必須將 --IpDiscovery 參數設定為 ipv4 或 ipv6。

### 引擎

redis

## EngineVersion

6.0

若您希望在此複寫群組上啟用傳輸中及待用加密，請新增 TransitEncryptionEnabled=true 或 AtRestEncryptionEnabled=true 參數或同時新增兩者，並符合以下條件。

- 您的複寫群組必須執行 Redis 3.2.6 版或 4.0.10 OSS版。
- 複寫群組必須在 Amazon 中建立VPC。
- 您也必須包含 CacheSubnetGroup 參數。

- 您也必須將 參數AuthToken包含客戶指定的AUTH字符串值（密碼），才能在此複寫群組上執行操作。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateReplicationGroup
&CacheNodeType=cache.m4.large
&CacheParameterGroup=default.redis6.xcluster.on
&Engine=redis
&EngineVersion=6.0
&NumNodeGroups=3
&ReplicasPerNodeGroup=2
&ReplicationGroupDescription=test%20group
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需您可能想要使用的其他資訊和參數，請參閱 ElastiCache API主題 [CreateReplicationGroup](#)。

## 檢視複寫群組詳細資訊

有時候，您可能會想要檢視複寫群組的詳細資訊。您可以使用 ElastiCache 主控台、AWS CLI 適用於的 ElastiCache或 ElastiCache API。Valkey 或 Redis OSS（停用叢集模式）和 Valkey 或 Redis OSS（啟用叢集模式）的主控制台程序不同。

### 檢視複寫群組詳細資訊

- [使用複本檢視 Valkey 或 Redis OSS（停用叢集模式）](#)
  - [檢視 Valkey 或 Redis OSS（停用叢集模式）複寫群組（主控台）](#)
  - [檢視 Valkey 或 Redis OSS（停用叢集模式）複寫群組（AWS CLI）](#)
  - [檢視 Valkey 或 Redis OSS（停用叢集模式）複寫群組（ElastiCache API）](#)
- [檢視複寫群組：Valkey 或 Redis OSS（已啟用叢集模式）](#)
  - [檢視 Valkey 或 Redis OSS（啟用叢集模式）叢集（主控台）](#)
  - [檢視 Valkey 或 Redis OSS（已啟用叢集模式）叢集（AWS CLI）](#)
  - [檢視 Valkey 或 Redis OSS（啟用叢集模式）叢集（ElastiCache API）](#)

- [檢視複寫群組詳細資訊 \(AWS CLI\)](#)
- [檢視複寫群組詳細資訊 \(ElastiCache API\)](#)

使用複本檢視 Valkey 或 Redis OSS ( 停用叢集模式 )

您可以使用 ElastiCache 主控台、AWS CLI for 或 來檢視具有複本 ( API/CLI : 複寫群組 ) 的 Valkey ElastiCache或 Redis OSS ( 停用叢集模式 ) 叢集的詳細資訊 ElastiCache API。

檢視 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的詳細資訊

- [檢視 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組 \( 主控台 \)](#)
- [檢視 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [檢視 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組 \( ElastiCache API \)](#)

檢視 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ( 主控台 )

若要使用 ElastiCache 主控台檢視具有複本的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的詳細資訊，請參閱主題 [檢視 Valkey 或 Redis OSS \( 停用叢集模式 \) 詳細資訊 \( 主控台 \)](#)。

檢視 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ( AWS CLI )

如需顯示 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組詳細資訊 AWS CLI 的範例，請參閱 [檢視複寫群組詳細資訊 \(AWS CLI\)](#)。

檢視 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 ( ElastiCache API )

如需 ElastiCache API顯示 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組詳細資訊的範例，請參閱 [檢視複寫群組詳細資訊 \(ElastiCache API\)](#)。

檢視複寫群組 : Valkey 或 Redis OSS ( 已啟用叢集模式 )

檢視 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( 主控台 )

若要使用 ElastiCache 主控台檢視 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的詳細資訊，請參閱 [檢視 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \) 的詳細資訊](#)。

檢視 Valkey 或 Redis OSS ( 已啟用叢集模式 ) 叢集 ( AWS CLI )

如需 ElastiCache CLI顯示 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組詳細資訊的範例，請參閱 [檢視複寫群組詳細資訊 \(AWS CLI\)](#)。

## 檢視 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( ElastiCache API )

如需 ElastiCache API 顯示 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組詳細資訊的範例，請參閱 [檢視複寫群組詳細資訊 \(ElastiCache API\)](#)。

### 檢視複寫群組詳細資訊 (AWS CLI)

您可以使用 命令檢視複寫群組 AWS CLI `describe-replication-groups` 的詳細資訊。使用以下選用參數來縮小清單的搜尋範圍。忽略參數會傳回最多 100 個複寫群組的詳細資訊。

#### 選用的參數

- `--replication-group-id` - 使用此參數來列出特定複寫群組的詳細資訊。若指定複寫群組擁有超過一個節點群組，結果會以節點群組做為群組傳回。
- `--max-items` - 使用此參數來限制列出的複寫群組數量。`--max-items` 的值不可小於 20 或大於 100。

#### Example

以下程式碼清單會列出最多 100 個複寫群組的詳細資訊。

```
aws elasticache describe-replication-groups
```

以下程式碼會列出 `sample-repl-group` 的詳細資訊。

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

以下程式碼會列出 `sample-repl-group` 的詳細資訊。

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

以下程式碼清單會列出最多 25 個複寫群組的詳細資訊。

```
aws elasticache describe-replication-groups --max-items 25
```

此操作的輸出應該看起來像這樣 ( JSON 格式 )。

```
{
 "ReplicationGroups": [
 {
```

```
"Status": "available",
"Description": "test",
"NodeGroups": [
 {
 "Status": "available",
 "NodeGroupMembers": [
 {
 "CurrentRole": "primary",
 "PreferredAvailabilityZone": "us-west-2a",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "rg-name-001"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2b",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "rg-name-002"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2c",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "rg-name-003"
 }
],
 "NodeGroupId": "0001",
 "PrimaryEndpoint": {
 "Port": 6379,
 "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
 }
 }
],
```

```
 "ReplicationGroupId": "rg-name",
 "AutomaticFailover": "enabled",
 "SnapshottingClusterId": "rg-name-002",
 "MemberClusters": [
 "rg-name-001",
 "rg-name-002",
 "rg-name-003"
],
 "PendingModifiedValues": {}
 },
 {
 ... some output omitted for brevity
 }
]
```

如需詳細資訊，請參閱 AWS CLI for ElastiCache 主題 [describe-replication-groups](#)。

### 檢視複寫群組詳細資訊 (ElastiCache API)

您可以使用 AWS CLI `DescribeReplicationGroups` 操作檢視複寫的詳細資訊。使用以下選用參數來縮小清單的搜尋範圍。忽略參數會傳回最多 100 個複寫群組的詳細資訊。

#### 選用的參數

- `ReplicationGroupId` - 使用此參數來列出特定複寫群組的詳細資訊。若指定複寫群組擁有超過一個節點群組，結果會以節點群組做為群組傳回。
- `MaxRecords` - 使用此參數來限制列出的複寫群組數量。`MaxRecords` 的值不可小於 20 或大於 100。預設為 100。

#### Example

以下程式碼清單會列出最多 100 個複寫群組的詳細資訊。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

以下程式碼會列出 myReplGroup 的詳細資訊。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

以下程式碼清單會列出最多 25 個叢集的詳細資訊。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&MaxRecords=25
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 ElastiCache API 參考主題 [DescribeReplicationGroups](#)。



## 尋找複寫群組端點

應用程式可以連線到複寫群組中的任何節點，前提是其具有該節點的DNS端點和連接埠號碼。取決於您執行的是 Valkey 或 Redis OSS ( 停用叢集模式 ) 或 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組，您會對不同的端點感興趣。

### Valkey 或 Redis OSS ( 停用叢集模式 )

具有複本的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集有三種端點類型：主要端點、讀取器端點和節點端點。主要端點是一律解析為叢集中主要節點DNS的名稱。主要端點不會受到您叢集變更的影響 (例如將僅供讀取複本提升至主要角色)。針對寫入活動，我們建議您的應用程式連線到主要端點。

讀取器端點將在 ElastiCache 叢集中的所有僅供讀取複本之間平均分割端點的傳入連線。其他因素 (例如應用程式建立連線或應用程式如何 (重新) 使用連線) 將決定流量分佈。隨著複本的新增或移除，讀取器端點會跟著叢集的變更即時保持在最新狀態。您可以將 ElastiCache ( Redis OSS ) 叢集的多個僅供讀取複本放在在不同的 AWS 可用區域 ( AZ )，以確保讀取器端點的高可用性。

#### Note

讀取器端點並非負載平衡器。它是以循環方式解析為其中一個複本節點的 IP 地址DNS的記錄。

針對讀取活動，應用程式也可連線到叢集中的任何節點。與主要端點不同，節點端點會解析至特定端點。若您在叢集中進行變更 (例如新增或刪除複本)，您必須更新您應用程式中的節點端點。

### Valkey 或 Redis OSS ( 啟用叢集模式 )

具有複本的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集，因為它們具有多個碎片 ( API/CLI：節點群組 )，這表示它們也具有多個主要節點，與 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集具有不同的端點結構。Valkey 或 Redis OSS ( 啟用叢集模式 ) 具有組態端點，「知道」叢集中的所有主要和節點端點。您的應用程式會連線到組態端點。每當您的應用程式從叢集的組態端點、Valkey 和 Redis 寫入或讀取OSS時，在場景後面決定金鑰所屬的碎片，以及該碎片中要使用的端點。這對您的應用程式來說都相當透明。

您可以使用 ElastiCache 主控台、AWS CLI或 找到叢集的端點 ElastiCache API。

## 尋找複寫群組端點

若要尋找您複寫群組的端點，請參閱以下其中一個主題：

- [尋找 Valkey 或 Redis OSS \( 停用叢集模式 \) 叢集的端點 \( 主控台 \)](#)
- [尋找 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \) 的端點](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( AWS CLI \)](#)
- [尋找 Valkey 或 Redis OSS 複寫群組的端點 \( ElastiCache API \)](#)

## 修改複寫群組

### ⚠ 重要限制

- 目前，ElastiCache 支援對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組進行有限的修改，例如使用 API 操作 `ModifyReplicationGroup` ( CLI : ) 變更引擎版本 `modify-replication-group`。您可以使用 API 操作 ( CLI : ) 修改 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片 [ModifyReplicationGroupShardConfiguration](#) ( 節點群組 ) 數量 [modify-replication-group-shard-configuration](#)。如需詳細資訊，請參閱 [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)。

對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集進行的其他修改需要您建立包含變更之新叢集的叢集。

- 您可以將 Valkey 或 Redis OSS ( 停用叢集模式 ) 和 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組升級至較新的引擎版本。不過，您無法降級至舊版引擎，除非您刪除現有的叢集或複寫群組，並重新建立叢集。如需詳細資訊，請參閱 [的版本管理 ElastiCache](#)。
- 您可以使用主控台或 [modify-replication-group](#) CLI 命令，升級 ElastiCache 使用停用叢集模式的 Valkey [ModifyReplicationGroup](#) API 或 Redis OSS 叢集現有的，以使用啟用叢集模式，如下範例所示。或者，您也可以依照 [修改叢集模式](#) 中的步驟執行。

您可以使用 ElastiCache 主控台、或 AWS CLI 修改 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的設定 ElastiCache API。目前，ElastiCache 支援對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組進行有限數量的修改。其他修改需要您建立目前複寫群組的備份，然後使用該備份來植入新的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。

### 主題

- [使用 AWS Management Console](#)
- [使用 AWS CLI](#)
- [使用 ElastiCache API](#)

### 使用 AWS Management Console

若要修改 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，請參閱 [修改 ElastiCache 叢集](#)。

## 使用 AWS CLI

以下是 `modify-replication-group` 命令 AWS CLI 的範例。您可以使用相同命令對複寫群組進行其他修改。

在現有的 Valkey 或 Redis OSS 複寫群組上啟用多可用區：

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --multi-az-enabled = true
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --multi-az-enabled
```

將叢集模式從停用修改為啟用：

若要將叢集模式從停用修改為啟用，您必須先將叢集模式設定為相容。相容模式可讓您的 Valkey 或 Redis OSS 用戶端使用啟用叢集模式和停用叢集模式進行連線。遷移所有 Valkey 或 Redis OSS 用戶端以使用已啟用的叢集模式後，您就可以完成叢集模式組態，並將叢集模式設定為已啟用。

若為 Linux、macOS 或 Unix：

將叢集模式設為相容。

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --cache-parameter-group-name myParameterGroupName \
 --cluster-mode compatible
```

將叢集模式設為啟用。

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --cluster-mode enabled
```

針對 Windows：

將叢集模式設為相容。

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --cache-parameter-group-name myParameterGroupName ^
 --cluster-mode compatible
```

將叢集模式設為啟用。

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --cluster-mode enabled
```

如需 命令的詳細資訊 AWS CLI `modify-replication-group`，請參閱 [modify-replication-group](#) 或 [修改 \( Redis \) 使用者指南 中的叢集模式](#)。ElastiCache OSS

使用 ElastiCache API

下列 ElastiCache API操作會在現有的 Valkey 或 Redis OSS複寫群組上啟用多可用區。您可以使用相同操作對複寫群組進行其他修改。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&AutomaticFailoverEnabled=true
&Mutli-AZEnabled=true
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需 `ModifyReplicationGroup` 操作的詳細資訊 ElastiCache API，請參閱 [ModifyReplicationGroup](#).

## 刪除複寫群組

如果您不再需要具有複本的其中一個叢集（在 API/ 中稱為複寫群組 CLI），則可以將其刪除。當您刪除複寫群組時，會 ElastiCache 刪除該群組中的所有節點。

此作業開始後便無法中斷或取消。

### Warning

- 當您刪除 ElastiCache（Redis OSS）叢集時，會保留手動快照。您還可以選擇在刪除叢集之前建立最終快照。自動快照則不會保留。
- CreateSnapshot 建立最終快照需要許可。如果沒有此許可，API 呼叫將會失敗，但有 Access Denied 例外狀況。

### 刪除複寫群組 (主控台)

若要刪除具有複本的叢集，請參閱 [在中刪除叢集 ElastiCache](#)。

### 刪除複寫群組 (AWS CLI)

使用命令 [delete-replication-group](#) 刪除複寫群組。

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

隨即出現提示，請您確認該項決定。輸入 y (是) 來立即啟動操作。在程序啟動後，便無法復原。

```
After you begin deleting this replication group, all of its nodes will be deleted as well.
```

```
Are you sure you want to delete this replication group? [Ny]y
```

```
REPLICATIONGROUP my-repgroup My replication group deleting
```

### 刪除複寫群組 (ElastiCache API)

Call [DeleteReplicationGroup](#) 搭配 ReplicationGroup 參數。

### Example

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DeleteReplicationGroup
&ReplicationGroupId=my-repgroup
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

### Note

若您將 `RetainPrimaryCluster` 參數設為 `true`，便會刪除所有僅供讀取複本，但仍會保留主要叢集。

## 變更複本的數量

您可以使用 AWS Management Console、或動態增加或減少 Valkey 或 Redis AWS CLI OSS 複寫群組中的僅供讀取複本數量 ElastiCache API。如果您的複寫群組是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組，您可以選擇要增加或減少複本數量的碎片 ( 節點群組 )。

若要動態變更複寫群組中的複本數量，請從下表選擇適合您情況的操作。

| 執行此作業 | 針對 Valkey 或 Redis OSS ( 啟用叢集模式 ) | 針對 Valkey 或 Redis OSS ( 停用叢集模式 )                                                           |
|-------|----------------------------------|--------------------------------------------------------------------------------------------|
| 新增複本  | <a href="#">增加碎片中的複本數量</a>       | <a href="#">增加碎片中的複本數量</a><br><br><a href="#">新增 Valkey 或 Redis 的僅供讀取複本 OSS ( 停用叢集模式 )</a> |
| 刪除複本  | <a href="#">減少碎片中的複本數量</a>       | <a href="#">減少碎片中的複本數量</a><br><br><a href="#">刪除 Valkey 或 Redis 的僅供讀取複本 OSS ( 停用叢集模式 )</a> |



## 增加碎片中的複本數量

您可以增加 Valkey 或 Redis OSS ( 啟用叢集模式 ) 碎片或 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組中的複本數量，最多 5 個。您可以使用 AWS Management Console、AWS CLI 或來執行此操作 ElastiCache API。

### 主題

- [使用 AWS Management Console](#)
- [使用 AWS CLI](#)
- [使用 ElastiCache API](#)

### 使用 AWS Management Console

下列程序使用主控台來增加 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組中的複本數量。

#### 若要增加碎片中的複本數量

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS，然後選擇您要新增複本的複寫群組名稱。
3. 選擇每個您欲新增複本碎片的方塊。
4. 選擇 Add replicas (新增複本)。
5. 完成 Add Replicas to Shards (新增複本至碎片) 頁面：
  - 針對 New number of replicas/shard (新的複本/碎片數)，輸入您希望所選取碎片擁有的複本數。此值必須大於等於 Current Number of Replicas per shard (目前每個碎片的複本數)，並小於等於五。我們建議您包含至少兩個複本，做為運作的最低需求。
  - 針對可用區域，選擇無偏好設定以為每個新複本 ElastiCache 選擇可用區域，或指定可用區域以為每個新複本選擇可用區域。

如果您選擇 Specify Availability Zones (指定可用區域)，請使用清單指定每個複本的可用區域。

6. 選擇 Add (新增) 以新增複本，或選擇 Cancel (取消) 以取消操作。

### 使用 AWS CLI

若要增加 Valkey 或 Redis OSS 碎片中的複本數量，請使用 `increase-replica-count` 命令搭配下列參數：

- `--replication-group-id` - 必要項目。識別您希望增加複本數的複寫群組。
- `--apply-immediately` 或 `--no-apply-immediately` - 必要項目。指定是否要立即增加複本計數 (`--apply-immediately`)，或是在下一次維護時間時進行 (`--no-apply-immediately`)。目前不支援 `--no-apply-immediately`。
- `--new-replica-count` - 選用。指定您希望完成時擁有的複本節點數，最多五個。針對只有一個節點群組或 Valkey 或 Redis OSS ( 啟用 OSS 叢集模式 ) 群組，或您希望所有節點群組擁有相同數量複本的複寫群組，使用此參數。若此值不大於目前節點群組中的複本數，則呼叫會失敗並附帶異常。
- `--replica-configuration` - 選用。允許您為每個節點群組分別設定複本數及可用區域。針對您要獨立設定每個節點群組的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 群組，使用此參數。

`--replica-configuration` 有三個選用成員：

- `NodeId` - 待設定節點群組的四位數 ID。對於 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，碎片 ID 一律為 0001。若要尋找 Valkey 或 Redis OSS ( 啟用叢集模式 ) 節點群組 ( 碎片 ) ID，請參閱 [尋找碎片的 ID](#)。
- `NewReplicaCount` - 此作業結束後，您希望此節點群組中擁有的複本數。其值必須大於目前的複本數，最多五個。若此值不大於目前節點群組中的複本數，則呼叫會失敗並附帶異常。
- `PreferredAvailabilityZones` - `PreferredAvailabilityZone` 字串的清單，指定複寫群組節點所在的可用區域。`PreferredAvailabilityZone` 值的數量必須等於 `NewReplicaCount + 1` 的值，以包含主要節點。如果 `--replica-configuration` 省略此成員，ElastiCache ( Redis OSS ) 會為每個新複本選擇可用區域。

#### Important

您必須在呼叫中包含 `--new-replica-count` 或 `--replica-configuration` 參數，但不可同時包含兩者。

## Example

以下範例會將複寫群組 `sample-repl-group` 中的複本數增加到三。範例完成時，每個節點群組中都會有三個複本。此數字適用於具有單一節點群組的 Valkey 或 Redis OSS ( 停用叢集模式 ) 群組，或是具有多個節點群組的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache increase-replica-count \
 --replication-group-id sample-repl-group \
 --new-replica-count 3 \
 --apply-immediately
```

針對 Windows :

```
aws elasticache increase-replica-count ^\
 --replication-group-id sample-repl-group ^\
 --new-replica-count 3 ^\
 --apply-immediately
```

以下範例會將複寫群組 `sample-repl-group` 中的複本數增加到為兩個指定節點群組指定的值。由於有多個節點群組，因此這是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。當指定選用的 PreferredAvailabilityZones 時，所列出的可用區域數量必須等於 NewReplicaCount + 1 的值。此方法可涵蓋由 NodeGroupId 識別群組的主要節點。

若為 Linux、macOS 或 Unix :

```
aws elasticache increase-replica-count \
 --replication-group-id sample-repl-group \
 --replica-configuration \
 NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \
 NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
 --apply-immediately
```

針對 Windows :

```
aws elasticache increase-replica-count ^\
 --replication-group-id sample-repl-group ^\
 --replica-configuration ^\
 NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^\
 NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
 --apply-immediately
```

如需使用 增加複本數量的詳細資訊CLI，請參閱 Amazon Command Line 參考[increase-replica-count](#)中的。ElastiCache

## 使用 ElastiCache API

若要增加 Valkey 或 Redis OSS 碎片中的複本數量，請使用具有下列參數 IncreaseReplicaCount 的動作：

- ReplicationGroupId - 必要項目。識別您希望增加複本數的複寫群組。
- ApplyImmediately - 必要項目。指定是否要立即增加複本計數 (ApplyImmediately=True)，或是在下一次維護時間時進行 (ApplyImmediately=False)。目前不支援 ApplyImmediately=False。
- NewReplicaCount - 選用。指定您希望完成時擁有的複本節點數，最多五個。針對只有一個節點群組的 Valkey 或 Redis OSS (停用叢集模式) 複寫群組，或您希望所有節點群組擁有相同數量複本的 Valkey 或 Redis OSS (啟用叢集模式) 群組，使用此參數。若此值不大於目前節點群組中的複本數，則呼叫會失敗並附帶異常。
- ReplicaConfiguration - 選用。允許您為每個節點群組分別設定複本數及可用區域。針對您要獨立設定每個節點群組的 Valkey 或 Redis OSS (啟用叢集模式) 群組，使用此參數。

ReplicaConfiguraion 有三個選用成員：

- NodeGroupId - 待設定節點群組的四位數 ID。對於 Valkey 或 Redis OSS (停用叢集模式) 複寫群組，節點群組 (碎片) ID 一律為 0001。若要尋找 Valkey 或 Redis OSS (啟用叢集模式) 節點群組 (碎片) ID，請參閱 [尋找碎片的 ID](#)。
- NewReplicaCount - 此作業結束後，您希望此節點群組中擁有的複本數。其值必須大於目前的複本數，並且最多五個。若此值不大於目前節點群組中的複本數，則呼叫會失敗並附帶異常。
- PreferredAvailabilityZones - PreferredAvailabilityZone 字串的清單，指定複寫群組節點所在的可用區域。PreferredAvailabilityZone 值的數量必須等於 NewReplicaCount + 1 的值，以包含主要節點。如果 ReplicaConfiguration 省略此成員，ElastiCache (Redis OSS) 會為每個新複本選擇可用區域。

### Important

您必須在呼叫中包含 NewReplicaCount 或 ReplicaConfiguration 參數，但不可同時包含兩者。

## Example

以下範例會將複寫群組 `sample-repl-group` 中的複本數增加到三。範例完成時，每個節點群組中都會有三個複本。此數字適用於具有單一節點群組的 Valkey 或 Redis OSS ( 停用叢集模式 ) 群組，或是具有多個節點群組的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 群組。

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=IncreaseReplicaCount
 &ApplyImmediately=True
 &NewReplicaCount=3
 &ReplicationGroupId=sample-repl-group
 &Version=2015-02-02
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20150202T192317Z
 &X-Amz-Credential=<credential>
```

以下範例會將複寫群組 `sample-repl-group` 中的複本數增加到為兩個指定節點群組指定的值。由於有多個節點群組，因此這是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。當指定選用的 `PreferredAvailabilityZones` 時，所列出的可用區域數量必須等於 `NewReplicaCount + 1` 的值。此方法可涵蓋由 `NodeGroupId` 識別群組的主要節點。

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=IncreaseReplicaCount
 &ApplyImmediately=True
 &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
 &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1b
 &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
 &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
```

```
 &ReplicationGroupId=sample-repl-group
```

```
 &Version=2015-02-02
```

```
 &SignatureVersion=4
```

```
 &SignatureMethod=HmacSHA256
```

```
 &Timestamp=20150202T192317Z
```

```
 &X-Amz-Credential=<credential>
```

如需使用 增加複本數量的詳細資訊API，請參閱 Amazon 參考[IncreaseReplicaCount](#)中的 ElastiCache API。

## 減少碎片中的複本數量

對於 Valkey 或 Redis OSS ( 啟用叢集模式 ) ，您可以減少碎片中的複本數量，或在 Valkey 或 Redis 的複寫群組中減少複本數量 OSS ( 停用叢集模式 ) ：

- 對於 Valkey 或 Redis OSS ( 停用叢集模式 ) ，如果啟用多可用區，您可以將複本數量減少為 1，如果未啟用，則可以減少為零。
- 對於 Valkey 或 Redis OSS ( 啟用叢集模式 ) ，您可以將複本數量減少到零。但是，若主要節點失敗，您將無法容錯移轉至複本。

您可以使用 AWS Management Console、AWS CLI 或 ElastiCache API 來減少節點群組 ( 碎片 ) 或複寫群組中的複本數量。

### 主題

- [使用 AWS Management Console](#)
- [使用 AWS CLI](#)
- [使用 ElastiCache API](#)

### 使用 AWS Management Console

下列程序使用主控台來減少 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組中的複本數量。

若要減少 Valkey 或 Redis OSS 碎片中的複本數量

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Valkey 或 Redis OSS，然後選擇您要刪除複本的複寫群組名稱。
3. 選擇每個您欲從其中移除複本節點的碎片方塊。
4. 選擇 Delete replicas (刪除複本)。
5. 完成 Delete Replicas from Shards (從碎片刪除複本) 頁面：
  - a. 針對 New number of replicas/shard (新的複本/碎片數)，輸入您希望所選取碎片擁有的複本數。該數字必須等於或大於 1。我們建議您為每個碎片設定至少兩個複本，做為運作的最低需求。
  - b. 選擇 Delete (刪除) 以刪除複本，或選擇 Cancel (取消) 以取消操作。



### ⚠ Important

- 如果您未指定要刪除的複本節點，ElastiCache (Redis OSS) 會自動選取要刪除的複本節點。執行此操作時，ElastiCache (Redis OSS) 會嘗試保留複寫群組的多可用區架構，接著保留具有主要 最低複寫延遲的複本。
- 您無法刪除複寫群組中的主節點。若您指定刪除主要節點，操作會失敗並出現錯誤，指出已選取主要節點進行刪除。

## 使用 AWS CLI

若要減少 Valkey 或 Redis OSS 碎片中的複本數量，請使用 `decrease-replica-count` 命令搭配下列參數：

- `--replication-group-id` - 必要項目。識別您希望減少複本數的複寫群組。
- `--apply-immediately` 或 `--no-apply-immediately` - 必要項目。指定是否要立即減少複本計數 (`--apply-immediately`)，或是在下一次維護時間時進行 (`--no-apply-immediately`)。目前不支援 `--no-apply-immediately`。
- `--new-replica-count` - 選用。指定您希望的複本節點數量。`--new-replica-count` 的值必須為有效值，小於節點群組中目前的數量。如需了解最低允許值，請參閱[減少碎片中的複本數量](#)。若 `--new-replica-count` 的值不符合此需求，呼叫便會失敗。
- `--replicas-to-remove` - 選用。包含 IDs 指定要移除之複本節點的節點清單。
- `--replica-configuration` - 選用。允許您為每個節點群組分別設定複本數及可用區域。針對您要獨立設定每個節點群組的 Valkey 或 Redis OSS (啟用叢集模式) 群組，使用此參數。

`--replica-configuration` 有三個選用成員：

- `NodeGroupId` - 待設定節點群組的四位數 ID。對於 Valkey 或 Redis OSS (停用叢集模式) 複寫群組，碎片 ID 一律為 0001。若要尋找 Valkey 或 Redis OSS (啟用叢集模式) 節點群組 (碎片) ID，請參閱[尋找碎片的 ID](#)。
- `NewReplicaCount` - 選用參數，指定您希望的複本節點數。`NewReplicaCount` 的值必須為有效值，小於節點群組中目前的數量。如需了解最低允許值，請參閱[減少碎片中的複本數量](#)。若 `NewReplicaCount` 的值不符合此需求，呼叫便會失敗。
- `PreferredAvailabilityZones` - `PreferredAvailabilityZone` 字串的清單，指定複寫群組節點所在的可用區域。`PreferredAvailabilityZone` 值的數量必須等於 `NewReplicaCount + 1` 的值，以包含主要節點。如果 `--replica-configuration` 省略此成員，ElastiCache (Redis OSS) 會為每個新複本選擇可用區域。



**⚠ Important**

您必須包含一個 (並且只能一個) `--new-replica-count`、`--replicas-to-remove` 或 `--replica-configuration` 參數。

**Example**

以下範例會使用 `--new-replica-count` 將複寫群組 `sample-repl-group` 中的複本數減少至一。範例完成時，每個節點群組中都會有一個複本。無論是具有單一節點群組的 Valkey 或 Redis OSS (停用叢集模式) 群組，還是具有多個節點群組的 Valkey 或 Redis OSS (啟用叢集模式) 群組，此數字都適用。

若為 Linux、macOS 或 Unix：

```
aws elasticache decrease-replica-count
 --replication-group-id sample-repl-group \
 --new-replica-count 1 \
 --apply-immediately
```

針對 Windows：

```
aws elasticache decrease-replica-count ^
 --replication-group-id sample-repl-group ^
 --new-replica-count 1 ^
 --apply-immediately
```

以下範例透過從節點群組移除兩個指定複本 (0001 與 0003)，來減少複寫群組 `sample-repl-group` 中的複本數。

若為 Linux、macOS 或 Unix：

```
aws elasticache decrease-replica-count \
 --replication-group-id sample-repl-group \
 --replicas-to-remove 0001,0003 \
 --apply-immediately
```

針對 Windows：

```
aws elasticache decrease-replica-count ^
```

```
--replication-group-id sample-repl-group ^
--replicas-to-remove 0001,0003 \
--apply-immediately
```

以下範例會使用 `--replica-configuration` 將複寫群組 `sample-repl-group` 中的複本數減少到為兩個指定節點群組指定的值。由於有多個節點群組，因此這是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。當指定選用的 `PreferredAvailabilityZones` 時，所列出的可用區域數量必須等於 `NewReplicaCount + 1` 的值。此方法可涵蓋由 `NodeGroupId` 識別群組的主要節點。

若為 Linux、macOS 或 Unix：

```
aws elasticache decrease-replica-count \
 --replication-group-id sample-repl-group \
 --replica-configuration \
 NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-east-1c \
 NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
 --apply-immediately
```

針對 Windows：

```
aws elasticache decrease-replica-count ^
 --replication-group-id sample-repl-group ^
 --replica-configuration ^
 NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c ^
 NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
 --apply-immediately
```

如需使用 減少複本數量的詳細資訊CLI，請參閱 Amazon Command Line 參考[decrease-replica-count](#)中的。ElastiCache

### 使用 ElastiCache API

若要減少 Valkey 或 Redis OSS 碎片中的複本數量，請使用具有下列參數 `DecreaseReplicaCount` 的動作：

- `ReplicationGroupId` - 必要項目。識別您希望減少複本數的複寫群組。

- `ApplyImmediately` - 必要項目。指定是否要立即減少複本計數 (`ApplyImmediately=True`)，或是在下一次維護時間時進行 (`ApplyImmediately=False`)。目前不支援 `ApplyImmediately=False`。
- `NewReplicaCount` - 選用。指定您希望的複本節點數量。`NewReplicaCount` 的值必須為有效值，小於節點群組中目前的數量。如需了解最低允許值，請參閱[減少碎片中的複本數量](#)。若 `--new-replica-count` 的值不符合此需求，呼叫便會失敗。
- `ReplicasToRemove` - 選用。包含IDs指定要移除之複本節點的節點清單。
- `ReplicaConfiguration` - 選用。包含節點群組清單，允許您為每個節點群組分別設定複本數及可用區域。針對您要獨立設定每個節點群組的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 群組，使用此參數。

`ReplicaConfiguration` 有三個選用成員：

- `NodeId` - 待設定節點群組的四位數 ID。對於 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，節點群組 ID 一律為 0001。若要尋找 Valkey 或 Redis OSS ( 啟用叢集模式 ) 節點群組 ( 碎片 ) ID，請參閱 [尋找碎片的 ID](#)。
- `NewReplicaCount` - 此作業結束後，您希望此節點群組中擁有的複本數。若已啟用多個可用區，該值必須小於目前的複本數，且其最小值為 1；若未啟用具備自動容錯移轉的多個可用區，則最小值為 0。若此值不小於目前節點群組中的複本數，則呼叫會失敗並附帶異常。
- `PreferredAvailabilityZones` - `PreferredAvailabilityZone` 字串的清單，指定複寫群組節點所在的可用區域。`PreferredAvailabilityZone` 值的數量必須等於 `NewReplicaCount + 1` 的值，以包含主要節點。如果 `ReplicaConfiguration` 省略此成員，ElastiCache ( Redis OSS ) 會為每個新複本選擇可用區域。

#### Important

您必須包含一個 (並且只能一個) `NewReplicaCount`、`ReplicasToRemove` 或 `ReplicaConfiguration` 參數。

## Example

以下範例會使用 `NewReplicaCount` 將複寫群組 `sample-repl-group` 中的複本數減少至一。範例完成時，每個節點群組中都會有一個複本。無論是具有單一節點群組的 Valkey 或 Redis OSS ( 停用叢集模式 ) 群組，還是具有多個節點群組的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 群組，此數字都適用。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&NewReplicaCount=1
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

以下範例透過從節點群組移除兩個指定複本 (0001 與 0003) , 來減少複寫群組 sample-repl-group 中的複本數。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicasToRemove.ReplicaToRemove.1=0001
&ReplicasToRemove.ReplicaToRemove.2=0003
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

以下範例會使用 ReplicaConfiguration 將複寫群組 sample-repl-group 中的複本數減少到為兩個指定節點群組指定的值。由於有多個節點群組，因此這是 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。當指定選用的 PreferredAvailabilityZones 時，所列出的可用區域數量必須等於 NewReplicaCount + 1 的值。此方法可涵蓋由 NodeGroupId 識別群組的主要節點。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
&ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
```

```
&ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
&ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
 &ReplicationGroupId=sample-repl-group
 &Version=2015-02-02
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20150202T192317Z
 &X-Amz-Credential=<credential>
```

如需使用 減少複本數量的詳細資訊API，請參閱 Amazon 參考[DecreaseReplicaCount](#)中的 ElastiCache API

新增 Valkey 或 Redis 的僅供讀取複本 OSS ( 停用叢集模式 )

下列主題中的資訊僅適用於 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組。

隨著您的流量增加，您可能會希望將讀取擴張到更多節點，減少任何一個節點上的讀取壓力。在本主題中，您可以了解如何將僅供讀取複本新增至 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集。

Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組最多可以有五個僅供讀取複本。若您嘗試將僅供讀取複本新增到已有五個僅供讀取複本的複寫群組，操作便會失敗。

如需有關將複本新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組的資訊，請參閱下列內容：

- [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)
- [增加碎片中的複本數量](#)

您可以使用 ElastiCache 主控台、或將僅供讀取複本新增至 Valkey 或 Redis OSS ( 停用叢集模式 ) AWS CLI叢集 ElastiCache API。

相關主題

- [將節點新增至 ElastiCache 叢集](#)

- [將僅供讀取複本新增到複寫群組 \(AWS CLI\)](#)
- [使用 將僅供讀取複本新增至複寫群組 API](#)

### 將僅供讀取複本新增到複寫群組 (AWS CLI)

若要將僅供讀取複本新增至 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，請使用 AWS CLI `create-cache-cluster` 命令，搭配 參數 `--replication-group-id` 指定要新增叢集 ( 節點 ) 的複寫群組。

以下範例會建立 `my-read replica` 叢集，並將它新增到 `my-replication-group` 複寫群組。僅供讀取複本的節點類型、參數群組、安全群組、維護時間及其他設定都與 `my-replication-group` 中的其他節點相同。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \
 --cache-cluster-id my-read-replica \
 --replication-group-id my-replication-group
```

針對 Windows：

```
aws elasticache create-cache-cluster ^
 --cache-cluster-id my-read-replica ^
 --replication-group-id my-replication-group
```

如需使用 新增僅供讀取複本的詳細資訊CLI，請參閱 [create-cache-cluster](#) 在 Amazon ElastiCache Command Line 參考中。

### 使用 將僅供讀取複本新增至複寫群組 API

若要將僅供讀取複本新增至 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，請使用 ElastiCache `CreateCacheCluster` 操作搭配 參數 `ReplicationGroupId` 來指定要新增叢集 ( 節點 ) 的複寫群組。

以下範例會建立 `myReadReplica` 叢集，並將它新增到 `myReplicationGroup` 複寫群組。僅供讀取複本的節點類型、參數群組、安全群組、維護時間及其他設定都與 `myReplicationGroup` 中的其他節點相同。

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=CreateCacheCluster
&CacheClusterId=myReadReplica
&ReplicationGroupId=myReplicationGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需使用 新增僅供讀取複本的詳細資訊API，請參閱 [CreateCacheCluster](#) Amazon ElastiCache API 參考中的。

刪除 Valkey 或 Redis 的僅供讀取複本 OSS ( 停用叢集模式 )

下列主題中的資訊僅適用於 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組。

隨著 Valkey 或 Redis OSS複寫群組上的讀取流量變更，您可能想要新增或移除僅供讀取複本。從 複寫群組移除節點與刪除叢集相同，雖然有以下限制：

- 您無法從複寫群組移除主要節點。若您要刪除主要節點，請執行以下作業：
  1. 將僅供讀取複本提升為主要節點。如需將僅供讀取複本提升為主要節點的詳細資訊，請參閱[針對 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組，將僅供讀取複本提升為主要複本](#)。
  2. 刪除舊的主要節點。如需了解此方法的限制，請參閱下一點。
- 若在複寫群組上啟用異地同步備份，您便無法從複寫群組移除最後一個僅供讀取複本。在此情況下，請執行下列操作：
  1. 透過停用異地同步備份來修改複寫群組 如需詳細資訊，請參閱[修改複寫群組](#)。
  2. 刪除僅供讀取複本。

您可以使用 ElastiCache 主控台、AWS CLI for 或 ElastiCache，從 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組中移除僅供讀取複本 ElastiCache API。

如需從 Valkey 或 Redis OSS複寫群組刪除叢集的指示，請參閱以下內容：

- [使用 AWS Management Console](#)
- [使用 AWS CLI 刪除 ElastiCache 叢集](#)
- [使用 ElastiCache API](#)
- [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)

- [減少碎片中的複本數量](#)



## 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，將僅供讀取複本提升為主要複本

下列主題中的資訊僅適用於 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組。

您可以使用 AWS Management Console、或將 Valkey AWS CLI 或 Redis OSS ( 停用叢集模式 ) 僅供讀取複本提升為主要複本 ElastiCache API。您無法在複寫群組上啟用具備自動容錯移轉的異地同步備份時，將僅供讀取複本提升為主要節點。若要將 Valkey 或 Redis OSS ( 停用叢集模式 ) 複本提升為啟用多可用區域複寫群組的主要複本，請執行下列動作：

1. 修改複寫群組以停用異地同步備份 (您所有的叢集不需要位於相同可用區域，也能執行此作業)。如需詳細資訊，請參閱[修改複寫群組](#)。
2. 將僅供讀取複本提升為主要節點。
3. 修改複寫群組來重新啟用異地同步備份。

執行 Redis 2.6.13 OSS 或更早版本的複寫群組無法使用多可用區。

### 使用 AWS Management Console

以下程序會使用主控台將複本節點提升為主要節點。

將僅供讀取複本提升為主節點 (主控台)

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 如果您要提升的複本是啟用多可用區域之 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組的成員，請先修改複寫群組以停用多可用區域，然後再繼續。如需詳細資訊，請參閱[修改複寫群組](#)。
3. 選擇 Valkey 或 Redis OSS，然後從叢集清單中選擇您要修改的複寫群組。此複寫群組必須執行「Redis」引擎而非「Clustered Redis」引擎，並且必須至少擁有兩個以上的節點。
4. 從節點清單中，選擇您希望提升為主要節點的複本節點，然後針對 Actions (動作)，選擇 Promote (提升)。
5. 在 Promote Read Replica (提升僅供讀取複本) 對話方塊中，執行下列作業：
  - a. 針對立即套用，選擇 Yes (是) 來立即提升僅供讀取複本，或是 No (否) 來在下一次叢集的維修時間內提升。
  - b. 選擇 Promote (提升) 以提升僅供讀取複本，或選擇 Cancel (取消) 以取消操作。

6. 若在您開始提升程序前該叢集已啟用異地同步備份，請等到複寫群組的狀態成為 available (可用) 時再修改叢集以重新啟用異地同步備份。如需詳細資訊，請參閱[修改複寫群組](#)。

## 使用 AWS CLI

若複寫群組已啟用異地同步備份，您便無法將僅供讀取複本提升為主要節點。在某些情況下，您要提升的複本可能是已啟用異地同步備份的複寫群組的成員。在這些情況下，您必須先修改複寫群組來停用異地同步備份，之後才能繼續。您所有的叢集不需要位於相同可用區域，也能執行此作業。如需修改複寫群組的詳細資訊，請參閱[修改複寫群組](#)。

下列 AWS CLI 命令修改複寫群組 `sample-repl-group`，使僅供讀取複本 `my-replica-1` 成為複寫群組的主要複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id sample-repl-group \
 --primary-cluster-id my-replica-1
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id sample-repl-group ^
 --primary-cluster-id my-replica-1
```

如需修改複寫群組的詳細資訊，請參閱 [modify-replication-group](#) 在 Amazon ElastiCache Command Line 參考中。

## 使用 ElastiCache API

若複寫群組已啟用異地同步備份，您便無法將僅供讀取複本提升為主要節點。在某些情況下，您要提升的複本可能是已啟用異地同步備份的複寫群組的成員。在這些情況下，您必須先修改複寫群組來停用異地同步備份，之後才能繼續。您所有的叢集不需要位於相同可用區域，也能執行此作業。如需修改複寫群組的詳細資訊，請參閱[修改複寫群組](#)。

下列 ElastiCache API 動作會修改複寫群組 `myReplGroup`，使僅供讀取複本 `myReplica-1` 成為複寫群組的主要複本。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
```

```
&ReplicationGroupId=myReplGroup
&PrimaryClusterId=myReplica-1
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需修改複寫群組的詳細資訊，請參閱 [ModifyReplicationGroup](#) Amazon ElastiCache API 參考中的。

## 管理 ElastiCache 叢集維護

每個叢集都有每週一次的維護時段，會在此期間套用任何系統變更。使用 Valkey 和 Redis OSS，複寫群組具有相同的每週維護時段。如果您在建立或修改叢集或複寫群組時未指定偏好的維護時段，會在一週中隨機選擇的日期，在您區域的維護時段內 ElastiCache 指派 60 分鐘的維護時段。

60 分鐘的維護時段隨機選自每個區域的 8 小時時段。以下資料表列出每個區域的時段，預設維護時段會從此時段中指派。您可以選擇區域維護時段以外的偏好維護時段。

| 區域代碼           | 區域名稱        | 區域維護時段              |
|----------------|-------------|---------------------|
| ap-northeast-1 | 亞太 (東京) 區域  | 13 : 00–21 : 00 UTC |
| ap-northeast-2 | 亞太 (首爾) 區域  | 12 : 00–20 : 00 UTC |
| ap-northeast-3 | 亞太 (大阪) 區域  | 12 : 00–20 : 00 UTC |
| ap-southeast-3 | 亞太區域 (雅加達)  | 14 : 00–22 : 00 UTC |
| ap-south-1     | 亞太 (孟買) 區域  | 17 : 30–1 : 30 UTC  |
| ap-southeast-1 | 亞太 (新加坡) 區域 | 14 : 00–22 : 00 UTC |
| cn-north-1     | 中國 (北京) 區域  | 14 : 00–22 : 00 UTC |
| cn-northwest-1 | 中國 (寧夏) 區域  | 14 : 00–22 : 00 UTC |

| 區域代碼           | 區域名稱                         | 區域維護時段              |
|----------------|------------------------------|---------------------|
| ap-east-1      | 亞太區域 (香港) 區域                 | 13 : 00–21 : 00 UTC |
| ap-southeast-2 | 亞太 (雪梨) 區域                   | 12 : 00–20 : 00 UTC |
| eu-west-3      | 歐洲 (巴黎) 區域                   | 23 : 59–07 : 29 UTC |
| af-south-1     | 非洲 (開普敦) 區域                  | 13 : 00–21 : 00 UTC |
| eu-central-1   | 歐洲 (法蘭克福) 區域                 | 23 : 00–07 : 00 UTC |
| eu-west-1      | 歐洲 (愛爾蘭) 區域                  | 22 : 00–06 : 00 UTC |
| eu-west-2      | 歐洲 (倫敦) 區域                   | 23 : 00–07 : 00 UTC |
| me-south-1     | Middle East (Bahrain) Region | 13 : 00–21 : 00 UTC |
| me-central-1   | 中東 ( UAE ) 區域                | 13 : 00–21 : 00 UTC |
| eu-south-1     | Europe (Milan) Region        | 21 : 00–05 : 00 UTC |
| sa-east-1      | 南美洲 (聖保羅) 區域                 | 01 : 00–09 : 00 UTC |
| us-east-1      | 美國東部 (維吉尼亞北部) 區域             | 03 : 00–11 : 00 UTC |
| us-east-2      | 美國東部 (俄亥俄) 區域                | 04 : 00–12 : 00 UTC |
| us-gov-west-1  | AWS GovCloud (US) 區域         | 06 : 00–14 : 00 UTC |
| us-west-1      | 美國西部 (加利佛尼亞北部) 區域            | 06 : 00–14 : 00 UTC |
| us-west-2      | 美國西部 (奧勒岡) 區域                | 06 : 00–14 : 00 UTC |

### 變更叢集或複寫群組的維護時段

維護時段應落在使用量最低的時段，因此可能需要不時進行調整。您可以修改叢集或複寫群組來指定時間範圍，最多 24 小時，您已請求的所有維護活動都會在此期間進行。在此期間會進行所有您請求的延遲或待處理叢集修改。

**Note**

如果您想要立即使用 套用節點類型修改和/或引擎升級，AWS Management Console 請選取立即套用方塊。否則，這些修改將在您排程的下一次維護時段套用。若要使用 API，請參閱 [modify-replication-group](#) 或 [modify-cache-cluster](#)。

**其他資訊**

如需維護時段和節點取代的資訊，請參閱下列內容：

- [ElastiCache 維護](#) —FAQ 維護和節點替換
- [更換節點 \( Memcached \)](#) —管理 Memcached 的節點替換
- [修改 ElastiCache 叢集](#) - 變更叢集的維護時段
- [更換節點 \( Valkey 和 RedisOSS \)](#) - 管理節點更換
- [修改複寫群組](#) - 變更複寫群組的維護時段

**使用參數群組設定引擎 ElastiCache 參數**

Amazon ElastiCache 使用 參數來控制節點和叢集的執行期屬性。一般而言，更新的引擎版本會包含額外參數，可支援更新的功能。如需 Memcached 參數的資料表，請參閱 [Memcached 專用參數](#)。如需 Valkey 和 Redis OSS 參數的資料表，請參閱 [Valkey 和 Redis OSS 參數](#)。

如您所預期的，有些參數值 (例如 maxmemory) 會由引擎與節點類型決定。如需依節點類型的 Memcached 參數值資料表，請參閱 [Memcached 節點類型專用參數](#)。如需依節點類型列出這些 Valkey 和 Redis OSS 參數值的資料表，請參閱 [Redis OSS 節點類型特定參數](#)。

**Note**

如需 Memcached 限定參數，請參閱 [Memcached 限定參數](#)。

**主題**

- [中的參數管理 ElastiCache](#)
- [中的快取參數群組層 ElastiCache](#)
- [建立 ElastiCache 參數群組](#)

- [依名稱列出 ElastiCache 參數群組](#)
- [列出 ElastiCache 參數群組的值](#)
- [修改 ElastiCache 參數群組](#)
- [刪除 ElastiCache 參數群組](#)
- [引擎特定參數](#)

## 中的參數管理 ElastiCache

ElastiCache 參數會分組為具名參數群組，以便更輕鬆地管理參數。參數群組代表在啟動期間傳遞給引擎軟體的參數特定值組合。這些值會決定每個節點上的引擎程序在執行時間的行為。特定參數群組上的參數值會套用到所有與群組相關聯的節點，無論節點所屬的叢集為何。

若要調整您叢集的效能，您可以修改一部分的參數值，或是變更叢集的參數群組。

- 您無法修改或刪除預設參數群組。若您需要自訂參數值，您必須建立自訂參數群組。
- 對於 Memcached，參數群組系列和您指派給它的叢集必須相容。例如，若您的叢集執行 Memcached 1.4.8 版，您只能使用來自 Memcached 1.4 系列的參數群組 (預設或自訂)。

對於 Redis，參數群組系列和您指派給它的叢集必須相容。例如，如果您的叢集執行 Redis 3.2.10 OSS 版，您只能使用來自 Redis OSS 3.2 系列的預設或自訂參數群組。

- 若您變更叢集的參數群組，任何可條件式修改的參數值在目前及新的參數群組中都必須相同。
- 對於 Memcached，當您變更叢集的參數時，變更會立即套用至叢集。無論是變更叢集的參數群組本身或是叢集的參數群組內的參數值，均適用此情況。若要判斷特定參數變更何時套用，請參閱 [Memcached 專用參數](#) 資料表中的變更生效。如需重新啟動叢集節點的相關資訊，請參閱 [重新啟動叢集](#)。
- 對於 Redis，當您變更叢集的參數時，變更會立即套用到叢集，或在叢集節點重新啟動後套用到叢集，但有下列例外狀況。無論是變更叢集的參數群組本身或是叢集的參數群組內的參數值，均適用此情況。若要判斷特定參數變更何時套用，請參閱 [Valkey 和 Redis OSS 參數](#) 資料表中的變更生效。

如需重新啟動 Valkey 或 Redis OSS 節點的詳細資訊，請參閱 [重新啟動節點](#)。

### Valkey 或 Redis OSS (已啟用叢集模式) 參數變更

如果您在 Valkey 或 Redis OSS (啟用叢集模式) 叢集上對下列參數進行變更，請遵循後續步驟。

- activerehashing
  - databases
1. 建立叢集的手動備份。請參閱 [取得手動備份](#)。
  2. 刪除叢集。請參閱 [刪除叢集](#)。
  3. 使用更改的參數群組和備份來植入新的叢集，以還原叢集。請參閱 [從備份還原到新的快取](#)。

對其他參數進行變更不需要這樣做。

- 您可以將參數群組與 Valkey 和 Redis OSS 全域資料存放區建立關聯。全域資料存放區是跨越 AWS 區域的一或多個叢集集合。在此情況下，組成全域資料存放區的所有叢集都會共用參數群組。對主要叢集參數群組進行的任何修改都會複寫到全域資料存放區中的所有其餘叢集。如需詳細資訊，請參閱 [使用全域資料存放區跨 AWS 區域複寫](#)。

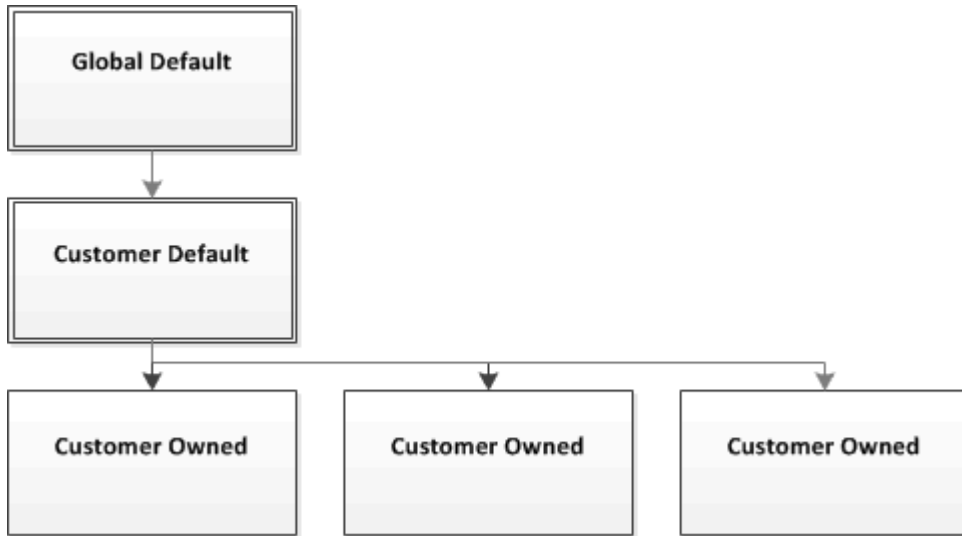
您可以查看下列位置來檢查參數群組是否屬於全域資料存放區的一部分：

- 在參數群組頁面上的 ElastiCache 主控台上，是/否全域屬性
- [CacheParameterGroup](#) API 操作的是/否 IsGlobal 屬性



## 中的快取參數群組層 ElastiCache

Amazon ElastiCache 有三個快取參數群組層，如下所示。



### Amazon ElastiCache 參數群組層

#### 全域預設

區域中所有 Amazon ElastiCache 客戶的頂層根參數群組。

全域預設快取參數群組：

- 已預留給 `default`，ElastiCache 且客戶無法使用。

#### 客戶預設

全域預設快取參數群組的複本，其建立的目的是供客戶使用。

客戶預設快取參數群組：

- 由 `default` 建立和擁有 ElastiCache。
- 可供客戶用來做為任何執行此快取參數群組所支援引擎版本叢集的快取參數群組使用。
- 客戶無法對其進行編輯。

#### 客戶自有

客戶預設快取參數群組的複本。客戶自有快取參數群組會在客戶建立快取參數群組時建立。

客戶自有快取參數群組：

- 由客戶建立及擁有。
- 可指派給任何客戶的相容叢集。
- 客戶可對其進行修改，建立自訂快取參數群組。

並非所有參數值皆可修改。如需 Memcached 值的詳細資訊，請參閱 [Memcached 專用參數](#)。如需 Valkey 和 Redis OSS值的詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#)。

## 建立 ElastiCache 參數群組

若您希望修改不同於預設值的一或多個參數值，便需要建立新的參數群組。您可以使用ElastiCache 主控台、AWS CLI或 建立參數群組 ElastiCache API。

### 建立 ElastiCache 參數群組（主控台）

下列程序說明如何使用 ElastiCache主控台建立參數群組。

#### 使用 ElastiCache 主控台建立參數群組

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 若要建立新的參數群組，請選擇 Create Parameter Group (建立參數群組)。

Create Parameter Group (建立參數群組) 畫面隨即出現。

4. 從 Family (系列) 清單，選擇做為您參數群組範本的參數群組系列。

參數群組系列，例如 memcached1.4 或 redis3.2，會定義參數群組中的實際參數及其初始值。參數群組系列必須與叢集的引擎和版本一致。

5. 在 Name (名稱) 方塊中，輸入此參數群組的唯一名稱。

建立叢集或修改叢集的參數群組時，您便會根據其名稱選擇參數群組。因此，我們建議選擇附帶資訊且能以某種方式識別參數群組系列的名稱。

參數群組命名限制條件如下：

- 必須以ASCII字母開頭。

- 只能包含ASCII字母、數字和連字號。
  - 長度必須介於 1 至 255 個字元之間。
  - 不能連續包含兩個連字號。
  - 結尾不能是連字號。
6. 在 Description (描述) 方塊中，輸入參數群組的描述。
  7. 若要建立參數群組，請選擇 Create (建立)。若要終止程序而不建立參數群組，請選擇 Cancel (取消)。
  8. 建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需詳細資訊，請參閱[修改 ElastiCache 參數群組](#)。

### 建立 ElastiCache 參數群組 ( AWS CLI )

若要使用 建立參數群組 AWS CLI，請使用 命令 `create-cache-parameter-group` 搭配這些參數。

- `--cache-parameter-group-name` - 參數群組的名稱。  
參數群組命名限制條件如下：
  - 必須以ASCII字母開頭。
  - 只能包含ASCII字母、數字和連字號。
  - 長度必須介於 1 至 255 個字元之間。
  - 不能連續包含兩個連字號。
  - 結尾不能是連字號。
- `--cache-parameter-group-family` - 參數群組的引擎和版本系列。
- `--description` - 使用者提供的參數群組說明。

### Example

下列範例使用 `memcachedmyMem1.4` 系列作為範本，建立名為 14 的參數群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --cache-parameter-group-family memcached1.4 \
 --description "My first parameter group"
```

## 針對 Windows :

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --cache-parameter-group-family memcached1.4 ^
 --description "My first parameter group"
```

此命令的輸出看起來會與以下內容相似。

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
}
```

## Example

下列範例會使用 `redismyRed2.8` 系列作為範本，建立名為 28 的參數群組。

若為 Linux、macOS 或 Unix :

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myRed28 \
 --cache-parameter-group-family redis2.8 \
 --description "My first parameter group"
```

## 針對 Windows :

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myRed28 ^
 --cache-parameter-group-family redis2.8 ^
 --description "My first parameter group"
```

此命令的輸出看起來會與以下內容相似。

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myRed28",
 "CacheParameterGroupFamily": "redis2.8",
 }
}
```

```
 "Description": "My first parameter group"
 }
}
```

建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需詳細資訊，請參閱[修改 ElastiCache 參數群組](#)。

如需詳細資訊，請參閱[create-cache-parameter-group](#)。

## 建立 ElastiCache 參數群組 (ElastiCache API)

若要使用 建立參數群組 ElastiCache API，請使用 CreateCacheParameterGroup 動作搭配這些參數。

- ParameterGroupName - 參數群組的名稱。

參數群組命名限制條件如下：

- 必須以ASCII字母開頭。
- 只能包含ASCII字母、數字和連字號。
- 長度必須介於 1 至 255 個字元之間。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。
- CacheParameterGroupFamily - 參數群組的引擎和版本系列。例如：memcached1.4。
- CacheParameterGroupFamily - 參數群組的引擎和版本系列。例如：redis2.8。
- Description - 使用者提供的參數群組說明。

## Example

下列範例使用 memcachedmyMem1.4 系列作為範本，建立名為 14 的參數群組。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheParameterGroup
&CacheParameterGroupFamily=memcached1.4
&CacheParameterGroupName=myMem14
&Description=My%20first%20parameter%20group
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
```

```
&X-Amz-Credential=<credential>
```

此動作的回應看起來會與以下內容相似。

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <CreateCacheParameterGroupResult>
 <CacheParameterGroup>
 <CacheParameterGroupName>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My first parameter group</Description>
 </CacheParameterGroup>
 </CreateCacheParameterGroupResult>
 <ResponseMetadata>
 <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
 </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

## Example

下列範例會使用 redismyRed2.8 系列作為範本，建立名為 28 的參數群組。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheParameterGroup
&CacheParameterGroupFamily=redis2.8
&CacheParameterGroupName=myRed28
&Description=My%20first%20parameter%20group
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

此動作的回應看起來會與以下內容相似。

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <CreateCacheParameterGroupResult>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRed28</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
 <Description>My first parameter group</Description>
```

```
</CacheParameterGroup>
</CreateCacheParameterGroupResult>
<ResponseMetadata>
 <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
</ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需詳細資訊，請參閱[修改 ElastiCache 參數群組](#)。

如需詳細資訊，請參閱[CreateCacheParameterGroup](#)。

## 依名稱列出 ElastiCache 參數群組

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache 列出參數群組 API。

### 依名稱列出參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台檢視參數群組的清單。

### 使用 ElastiCache 主控台列出參數群組

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。

### 依名稱列出 ElastiCache 參數群組 (AWS CLI)

若要使用產生參數群組清單 AWS CLI，請使用命令 `describe-cache-parameter-groups`。若您提供參數群組的名稱，便只會列出該參數群組。若您沒有提供參數群組的名稱，最多會列出 `--max-records` 個參數群組。在任一種情況下，都會列出參數群組的名稱、系列和描述。

### Example

下列範例程式碼會列出參數群組 `myMem14`。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myMem14
```

針對 Windows：

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myMem14
```

此命令的輸出看起來會與以下內容相似，列出參數群組的名稱、系列和描述。

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myMem14",
```



```
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
]
```

## Example

下列範例程式碼會列出參數群組 myRed28。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myRed28
```

針對 Windows：

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myRed28
```

此命令的輸出看起來會與以下內容相似，列出參數群組的名稱、系列和描述。

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myRed28",
 "CacheParameterGroupFamily": "redis2.8",
 "Description": "My first parameter group"
 }
]
}
```

## Example

下列範例程式碼列出在 Redis OSS引擎 myRed5.0.6 版之後執行的參數群組的參數群組 56。如果參數群組是 [使用全域資料存放區跨 AWS 區域複寫](#) 的一部分，則在輸出中傳回的 IsGlobal 屬性值將是 Yes。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myRed56
```

## 針對 Windows :

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myRed56
```

此命令的輸出看起來會像這樣，列出參數群組的名稱、系列 isGlobal 和描述。

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myRed56",
 "CacheParameterGroupFamily": "redis5.0",
 "Description": "My first parameter group",
 "IsGlobal": "yes"
 }
]
}
```

## Example

以下範本程式碼會列出最多 10 個參數群組。

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

此命令的JSON輸出看起來像這樣，會列出名稱、系列、描述，如果 redis5.6，則參數群組是否屬於每個參數群組的全域資料存放區 ( isGlobal )。

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "custom-redis32",
 "CacheParameterGroupFamily": "redis3.2",
 "Description": "custom parameter group with reserved-memory > 0"
 },
 {
 "CacheParameterGroupName": "default.memcached1.4",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "Default parameter group for memcached1.4"
 },
 {
 "CacheParameterGroupName": "default.redis2.6",
 "CacheParameterGroupFamily": "redis2.6",
 "Description": "Default parameter group for redis2.6"
 }
]
}
```

```
 "Description": "Default parameter group for redis2.6"
 },
 {
 "CacheParameterGroupName": "default.redis2.8",
 "CacheParameterGroupFamily": "redis2.8",
 "Description": "Default parameter group for redis2.8"
 },
 {
 "CacheParameterGroupName": "default.redis3.2",
 "CacheParameterGroupFamily": "redis3.2",
 "Description": "Default parameter group for redis3.2"
 },
 {
 "CacheParameterGroupName": "default.redis3.2.cluster.on",
 "CacheParameterGroupFamily": "redis3.2",
 "Description": "Customized default parameter group for redis3.2 with
cluster mode on"
 },
 {
 "CacheParameterGroupName": "default.redis5.6.cluster.on",
 "CacheParameterGroupFamily": "redis5.0",
 "Description": "Customized default parameter group for redis5.6 with
cluster mode on",
 "isGlobal": "yes"
 },
]
}
```

如需詳細資訊，請參閱[describe-cache-parameter-groups](#)。

依名稱列出 ElastiCache 參數群組 (ElastiCache API)

若要使用產生參數群組清單 ElastiCache API，請使用 DescribeCacheParameterGroups 動作。若您提供參數群組的名稱，便只會列出該參數群組。若您沒有提供參數群組的名稱，最多會列出 MaxRecords 個參數群組。在任一種情況下，都會列出參數群組的名稱、系列和描述。

### Example

下列範例程式碼會列出參數群組 myMem14。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myMem14
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

此動作的回應看起來會與以下內容相似，列出每個參數群組的名稱、系列和描述。

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <DescribeCacheParameterGroupsResult>
 <CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My custom Memcached 1.4 parameter group</Description>
 </CacheParameterGroup>
 </CacheParameterGroups>
 </DescribeCacheParameterGroupsResult>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

## Example

以下範本程式碼會列出最多 10 個參數群組。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

此動作的回應看起來會像這樣，列出名稱、系列、描述，如果參數群組屬於全域資料存放區 ( isGlobal )，則會列出每個參數群組的名稱、系列、描述，如果是 redis5.6，則會列出。

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
```

```

<DescribeCacheParameterGroupsResult>
 <CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
 <Description>My custom Redis 2.8 parameter group</Description>
 </CacheParameterGroup>
 <CacheParameterGroup>
 <CacheParameterGroupName>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My custom Memcached 1.4 parameter group</Description>
 </CacheParameterGroup>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
 <Description>My custom redis 5.6 parameter group</Description>
 <isGlobal>yes</isGlobal>
 </CacheParameterGroup>
 </CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>

```

## Example

下列範例程式碼會列出參數群組 myRed28。

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

此動作的回應看起來會與以下內容相似，列出名稱、系列和描述。

```

<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <DescribeCacheParameterGroupsResult>

```

```

<CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRed28</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
 <Description>My custom Redis 2.8 parameter group</Description>
 </CacheParameterGroup>
</CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>

```

## Example

下列範例程式碼會列出參數群組 myRed56。

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed56
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

此動作的回應看起來像這樣，會列出名稱、系列、描述，以及參數群組是否為全域資料存放區 ( ) 的一部分 isGlobal。

```

<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <DescribeCacheParameterGroupsResult>
 <CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRed56</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
 <Description>My custom Redis 5.6 parameter group</Description>
 <isGlobal>yes</isGlobal>
 </CacheParameterGroup>
 </CacheParameterGroups>
 </DescribeCacheParameterGroupsResult>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>

```

```
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

如需詳細資訊，請參閱[DescribeCacheParameterGroups](#)。

## 列出 ElastiCache 參數群組的值

您可以使用主控台、AWS CLI 或 列出參數群組 ElastiCache 的參數及其值 ElastiCache API。

### 列出 ElastiCache 參數群組的值 ( 主控台 )

下列程序說明如何使用 ElastiCache 主控台列出參數群組的參數及其值。

使用 ElastiCache 主控台列出參數群組的參數及其值

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望列出其參數及值的參數群組。

參數及其值會在畫面底部列出。根據參數的數量，您可能需要向上或向下捲動來尋找您想要的參數。

### 列出參數群組的值 (AWS CLI)

若要使用 列出參數群組的參數及其值 AWS CLI，請使用 命令 `describe-cache-parameters`。

#### Example

下列範例程式碼會列出參數群組 `myMem14` 的所有 Memcached 參數及其值。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myMem14
```

針對 Windows：

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myMem14
```

#### Example

下列範例程式碼會列出參數群組 `myRedis28` 的所有參數及其值。



若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myRedis28
```

針對 Windows：

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myRed28
```

如需詳細資訊，請參閱[describe-cache-parameters](#)。

列出參數群組的值 (ElastiCache API)

若要使用 列出參數群組的參數及其值 ElastiCache API，請使用 DescribeCacheParameters 動作。

Example

下列範例程式碼會列出參數群組 myMem14 的所有 Memcached 參數。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameters
&CacheParameterGroupName=myMem14
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

此動作的回應看起來會與以下內容相似。此回應已進行截斷。

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
 <DescribeCacheParametersResult>
 <CacheClusterClassSpecificParameters>
 <CacheNodeTypeSpecificParameter>
 <DataType>integer</DataType>
 <Source>system</Source>
 <IsModifiable>>false</IsModifiable>
 <Description>The maximum configurable amount of memory to use to store items,
in megabytes.</Description>
 <CacheNodeTypeSpecificValues>
```

```

 <CacheNodeTypeSpecificValue>
 <Value>1000</Value>
 <CacheClusterClass>cache.c1.medium</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>6000</Value>
 <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>7100</Value>
 <CacheClusterClass>cache.m1.large</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>1300</Value>
 <CacheClusterClass>cache.m1.small</CacheClusterClass>
 </CacheNodeTypeSpecificValue>

```

...output omitted...

```

 </CacheClusterClassSpecificParameters>
 </DescribeCacheParametersResult>
 <ResponseMetadata>
 <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParametersResponse>

```

## Example

下列範例程式碼會列出參數群組 `myRed28` 的所有參數。

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameters
&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

此動作的回應看起來會與以下內容相似。此回應已進行截斷。

```

<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">

```

```

<DescribeCacheParametersResult>
 <CacheClusterClassSpecificParameters>
 <CacheNodeTypeSpecificParameter>
 <DataType>integer</DataType>
 <Source>system</Source>
 <IsModifiable>>false</IsModifiable>
 <Description>The maximum configurable amount of memory to use to store items,
in megabytes.</Description>
 <CacheNodeTypeSpecificValues>
 <CacheNodeTypeSpecificValue>
 <Value>1000</Value>
 <CacheClusterClass>cache.c1.medium</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>6000</Value>
 <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>7100</Value>
 <CacheClusterClass>cache.m1.large</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 <CacheNodeTypeSpecificValue>
 <Value>1300</Value>
 <CacheClusterClass>cache.m1.small</CacheClusterClass>
 </CacheNodeTypeSpecificValue>
 </CacheNodeTypeSpecificValues>
 </CacheClusterClassSpecificParameters>
 </DescribeCacheParametersResult>
 <ResponseMetadata>
 <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParametersResponse>
...output omitted...

```

如需詳細資訊，請參閱[DescribeCacheParameters](#)。

## 修改 ElastiCache 參數群組

### Important

您無法修改任何預設參數群組。

您可以修改參數群組中的某些參數值。這些參數值都會套用到與參數群組相關聯的叢集。如需參數值變更何時套用至參數群組的詳細資訊，請參閱 [Valkey 和 Redis OSS 參數](#) 和 [Memcached 專用參數](#)。

## 修改參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台變更 cluster-enabled 參數的值。您會使用相同的程序來變更任何參數的值。

### 使用 ElastiCache 主控台變更參數的值

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望修改的參數群組。

參數群組的參數會在畫面底部列出。您可能需要查看各頁才能看到所有參數。

4. 若要修改一或多個參數，請選擇 Edit Parameters (編輯參數)。
5. 在 Edit Parameter Group: (編輯參數群組:) 畫面中，使用向左和向右箭頭捲動，直到找到 binding\_protocol 參數，然後在 Value (數值) 欄中輸入 ascii。
6. 選擇 Save Changes (儲存變更)。
7. 對於 Memcached，若要尋找您變更的參數名稱，請參閱 [Memcached 專用參數](#)。若變更是在「重新啟動之後」才套用到參數，請重新開機每個使用此參數群組的叢集。如需詳細資訊，請參閱 [重新啟動叢集](#)。
8. 使用 Valkey 和 Redis OSS，若要尋找您變更的參數名稱，請參閱 [Valkey 和 Redis OSS 參數](#)。如果您有 Valkey 或 Redis OSS (停用叢集模式) 叢集，並變更下列參數，則必須重新啟動叢集中的節點：

- activerehashing
- databases

如需詳細資訊，請參閱 [重新啟動節點](#)。

#### Valkey 或 Redis OSS (已啟用叢集模式) 參數變更

如果您在 Valkey 或 Redis OSS (啟用叢集模式) 叢集上對下列參數進行變更，請遵循後續步驟。

- activerehashing
- databases

1. 使用 Redis，您可以重新建立叢集的手動備份。請參閱 [取得手動備份](#)。
2. 刪除叢集。請參閱 [刪除叢集](#)。
3. 您可以使用更改的參數群組和備份來植入新的叢集，以還原叢集。請參閱 [從備份還原到新的快取](#)。

對其他參數進行變更不需要這樣做。

### 修改參數群組 (AWS CLI)

若要使用 變更參數的值 AWS CLI，請使用 命令 `modify-cache-parameter-group`。

#### Example

若要使用 Memcached 尋找您要變更之參數的名稱和許可值，請參閱 [Memcached 專用參數](#)

下列範本程式碼會為 myMem14 參數群組上的兩個參數 (`chunk_size` 和 `chunk_size_growth_fact`) 設定值。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --parameter-name-values \
 ParameterName=chunk_size,ParameterValue=96 \
 ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

針對 Windows：

```
aws elasticache modify-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --parameter-name-values ^
 ParameterName=chunk_size,ParameterValue=96 ^
 ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

此命令的輸出看起來會與以下內容相似。

```
{
 "CacheParameterGroupName": "myMem14"
}
```

### Example

使用 Valkey 和 Redis OSS，若要尋找您要變更之參數的名稱和許可值，請參閱 [Valkey 和 Redis OSS 參數](#)

下列範例程式碼會設定兩個參數的值，reserved-memory-percent 並在參數群組 上啟用叢集 myredis32-on-30。我們將 reserved-memory-percent 設為 30 (30%) 並啟用叢集，yes 以便參數群組可以與 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (複寫群組) 搭配使用。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name myredis32-on-30 \
 --parameter-name-values \
 ParameterName=reserved-memory-percent,ParameterValue=30 \
 ParameterName=cluster-enabled,ParameterValue=yes
```

針對 Windows：

```
aws elasticache modify-cache-parameter-group ^
 --cache-parameter-group-name myredis32-on-30 ^
 --parameter-name-values ^
 ParameterName=reserved-memory-percent,ParameterValue=30 ^
 ParameterName=cluster-enabled,ParameterValue=yes
```

此命令的輸出看起來會與以下內容相似。

```
{
 "CacheParameterGroupName": "my-redis32-on-30"
}
```

如需詳細資訊，請參閱 [modify-cache-parameter-group](#)。

若要尋找您要變更的參數名稱，請參閱 [Valkey 和 Redis OSS 參數](#)。

如果您有 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，並變更下列參數，則必須重新啟動叢集中的節點：

- activerehashing
- databases

如需詳細資訊，請參閱[重新啟動節點](#)。

#### Valkey 或 Redis OSS ( 已啟用叢集模式 ) 參數變更

如果您在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集上對下列參數進行變更，請遵循後續步驟。

- activerehashing
- databases

1. 建立叢集的手動備份。請參閱 [取得手動備份](#)。
2. 刪除叢集。請參閱 [刪除叢集](#)。
3. 您可以使用更改的參數群組和備份來植入新的叢集，以還原叢集。請參閱 [從備份還原到新的快取](#)。

對其他參數進行變更不需要這樣做。

#### 修改參數群組 (ElastiCache API)

若要使用 變更參數群組的參數值 ElastiCache API，請使用 ModifyCacheParameterGroup 動作。

#### Example

若要使用 Memcached 尋找您要變更之參數的名稱和許可值，請參閱 [Memcached 專用參數](#)

下列範本程式碼會為 myMem14 參數群組上的兩個參數 (chunk\_size 和 chunk\_size\_growth\_fact) 設定值。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myMem14
```

```
&ParameterNameValues.member.1.ParameterName=chunk_size
&ParameterNameValues.member.1.ParameterValue=96
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
&ParameterNameValues.member.2.ParameterValue=1.5
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

## Example

使用 Valkey 和 Redis OSS，若要尋找您要變更之參數的名稱和許可值，請參閱 [Valkey 和 Redis OSS 參數](#)

下列範例程式碼會設定兩個參數的值，reserved-memory-percent 並在參數群組 上啟用叢集 myredis32-on-30。我們將 reserved-memory-percent 設為 30 (30%) 並啟用叢集，yes 以便參數群組可以與 Valkey 或 Redis OSS (啟用叢集模式) 叢集 (複寫群組) 搭配使用。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myredis32-on-30
&ParameterNameValues.member.1.ParameterName=reserved-memory-percent
&ParameterNameValues.member.1.ParameterValue=30
&ParameterNameValues.member.2.ParameterName=cluster-enabled
&ParameterNameValues.member.2.ParameterValue=yes
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ModifyCacheParameterGroup](#)。

如果您有 Valkey 或 Redis OSS (停用叢集模式) 叢集，並變更下列參數，則必須重新啟動叢集中的節點：

- activerehashing
- databases

如需詳細資訊，請參閱 [重新啟動節點](#)。



### Valkey 或 Redis OSS ( 已啟用叢集模式 ) 參數變更

如果您在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集上對下列參數進行變更，請遵循後續步驟。

- activerehashing
- databases

1. 建立叢集的手動備份。請參閱 [取得手動備份](#)。
2. 刪除叢集。請參閱 [在中刪除叢集 ElastiCache](#)。
3. 您可以使用更改的參數群組和備份來植入新的叢集，以還原叢集。請參閱 [從備份還原到新的快取](#)。

對其他參數進行變更不需要這樣做。

## 刪除 ElastiCache 參數群組

您可以使用 ElastiCache 主控台、AWS CLI 或 刪除自訂參數群組 ElastiCache API。

若參數群組已和任何叢集相關聯，您便無法刪除參數群組。您也無法刪除任何預設參數群組。

### 刪除參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台刪除參數群組。

#### 使用 ElastiCache 主控台刪除參數群組

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望刪除的參數群組。

Delete (刪除) 按鈕即變成作用中。

4. 選擇 刪除。

Delete Parameter Group (刪除參數群組) 確認畫面隨即出現。

5. 若要刪除參數群組，請在 Delete Parameter Groups (刪除參數群組) 確認畫面上，選擇 Delete (刪除)。

若要保留參數群組，請選擇 Cancel (取消)。

### 刪除參數群組 (AWS CLI)

若要使用 刪除參數群組 AWS CLI，請使用 命令 `delete-cache-parameter-group`。針對要刪除的參數群組，以 `--cache-parameter-group-name` 指定的參數群組不能有任何與其相關聯的叢集，也不能是預設參數群組。

下列範例程式碼會刪除 myMem14 參數群組。

#### Example

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-parameter-group \
```

```
--cache-parameter-group-name myRed28
```

針對 Windows :

```
aws elasticache delete-cache-parameter-group ^
--cache-parameter-group-name myRed28
```

如需詳細資訊，請參閱[delete-cache-parameter-group](#)。

刪除參數群組 (ElastiCache API)

若要使用 刪除參數群組 ElastiCache API，請使用 DeleteCacheParameterGroup 動作。針對要刪除的參數群組，以 CacheParameterGroupName 指定的參數群組不能有任何與其相關聯的叢集，也不能是預設參數群組。

Example

使用 Memcached 時，下列範例程式碼會刪除 myMem14 參數群組。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DeleteCacheParameterGroup
&CacheParameterGroupName=myMem14
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Example

下列範例程式碼會刪除 myRed28 參數群組。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DeleteCacheParameterGroup
&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱[DeleteCacheParameterGroup](#)。

## 引擎特定參數

### Valkey 和 Redis OSS

Valkey 7.2 參數與 Redis OSS 7 參數相同。

如果您未指定 Valkey 或 Redis OSS 叢集的參數群組，則會使用適合您引擎版本的預設參數群組。您無法變更預設參數群組中任何參數的值。但是，只要可條件式修改參數的值在兩個參數群組中都是相同的，您便可以建立自訂參數群組並隨時將其指派給您的叢集。如需詳細資訊，請參閱[建立 ElastiCache 參數群組](#)。

### 主題

- [Valkey 和 Redis OSS 參數](#)
- [Memcached 專用參數](#)

## Valkey 和 Redis OSS 參數

### 主題

- [Valkey 7.2 和 Redis OSS 7 參數變更](#)
- [Redis OSS 6.x 參數變更](#)
- [Redis OSS 5.0.3 參數變更](#)
- [Redis OSS 5.0.0 參數變更](#)
- [Redis OSS 4.0.10 參數變更](#)
- [Redis OSS 3.2.10 參數變更](#)
- [Redis OSS 3.2.6 參數變更](#)
- [Redis OSS 3.2.4 參數變更](#)
- [Redis OSS 2.8.24 \( 增強 \) 新增參數](#)
- [Redis OSS 2.8.23 \( 增強 \) 新增的參數](#)
- [Redis OSS 2.8.22 \( 增強 \) 新增的參數](#)
- [Redis OSS 2.8.21 新增參數](#)
- [Redis OSS 2.8.19 新增參數](#)
- [Redis OSS 2.8.6 新增參數](#)
- [Redis OSS 2.6.13 參數](#)
- [Redis OSS 節點類型特定參數](#)

### Valkey 7.2 和 Redis OSS 7 參數變更

參數群組系列 : redis7

Redis OSS 7 預設參數群組如下所示 :

- `default.redis7` – 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集和複寫群組 , 使用此參數群組 , 或從中衍生的參數群組。
- `default.redis7.cluster.on` – 針對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組 , 使用此參數群組 , 或從中衍生的參數群組。

在 Redis 7 OSS 中新增的參數如下所示。

名稱	詳細資訊	描述
cluster-allow-pubsubshard-when-down	<p>允許的值：yes、no</p> <p>預設：yes</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	<p>當預設值為 yes 時，只要叢集認定自身擁有槽，就會允許節點在叢集處於停用狀態時提供 pubsub 碎片流量。</p>
cluster-preferred-endpoint-type	<p>允許的值：ip、tls-dynamic</p> <p>預設：tls-dynamic</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	<p>此值控制針對 MOVED/ASKING 請求傳回的端點，以及針對 CLUSTER SLOTS 和 的端點欄位 CLUSTER SHARDS。當該值設定為 ip 時，節點會公告其 IP 地址。當值設定為 tls-dynamic 時，節點會在啟用時 encryption-in-transit 公告主機名稱，否則會公告 ip 地址。</p>
latency-tracking	<p>允許的值：yes、no</p> <p>預設：no</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	<p>設定為 yes 時，會追蹤每個命令的延遲情況，並可透過 INFO 延遲統計資料命令匯出百分位數分佈，同時透過 LATENCY 命令匯出累積延遲分佈 (長條圖)。</p>
hash-max-listpack-entries	<p>允許的值：0+</p> <p>預設：512</p>	<p>要壓縮資料集的雜湊項目數目上限。</p>

名稱	詳細資訊	描述
	<p>類型：整數</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	
hash-max-listpack-value	<p>允許的值：0+</p> <p>預設：64</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	最大雜湊項目的臨界值，以便壓縮資料集。
zset-max-listpack-entries	<p>允許的值：0+</p> <p>預設：128</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	要壓縮資料集的雜湊有序集項目數目上限。

名稱	詳細資訊	描述
zset-max-listpack-value	允許的值：0+ 預設：64 類型：整數 可修改：是 變更生效：直接套用至叢集中所有節點。	要壓縮資料集的雜湊有序集項目數目上限。

Redis 7 OSS 中變更的參數如下所示。

名稱	詳細資訊	描述
activeresharding	可修改：no。在 Redis OSS 7 中，預設會隱藏並啟用此參數。為了將其停用，您需要建立一個 <a href="#">支援案例</a> 。	「可修改」先前為「是」。

在 Redis 7 OSS 中移除的參數如下所示。

名稱	詳細資訊	描述
hash-max-ziplist-entries	允許的值：0+ 預設：512 類型：整數 可修改：是 變更生效：直接套用至叢集中所有節點。	使用 listpack 而非 ziplist 來表示小雜湊編碼



名稱	詳細資訊	描述
hash-max-ziplist-value	<p>允許的值：0+</p> <p>預設：64</p> <p>類型：整數</p> <p>可修改：是</p> <p>變更生效：直接套用至叢集中所有節點。</p>	使用 listpack 而非 ziplist 來表示小雜湊編碼
zset-max-ziplist-entries	<p>允許的值：0+</p> <p>預設：128</p> <p>類型：整數</p> <p>可修改：是</p> <p>變更生效：直接套用至叢集中所有節點。</p>	使用 listpack 而非 ziplist 來表示小雜湊編碼。
zset-max-ziplist-value	<p>允許的值：0+</p> <p>預設：64</p> <p>類型：整數</p> <p>可修改：是</p> <p>變更生效：直接套用至叢集中所有節點。</p>	使用 listpack 而非 ziplist 來表示小雜湊編碼。

名稱	詳細資訊	描述
list-max-ziplist-size	允許的值： 預設：-2 類型：整數 可修改：是 變更生效：直接套用至叢集中所有節點。	每個內部清單節點允許的項目數量。

## Redis OSS 6.x 參數變更

參數群組系列：redis6.x

Redis OSS 6.x 預設參數群組如下所示：

- default.redis6.x – 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。
- default.redis6.x.cluster.on – 針對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。

### Note

在 Redis OSS 引擎 6.2 版中，當 r6gd 節點系列引入與 [搭配使用時中的資料分層 ElastiCache](#)，r6gd 節點類型僅支援新版本、volatile-lru 和 allkeys-lru max-memory 政策。

如需詳細資訊，請參閱 [ElastiCache \( Redis OSS \) 6.2 版 \( 增強版 \)](#) 和 [ElastiCache \( Redis OSS \) 6.0 版 \( 增強版 \)](#)。

在 Redis 6.x OSS 中新增的參數如下所示。

詳細資訊	描述	
<code>acl-pubsub-default</code> (added in 6.2)	<p>允許的值：resetchannels、allchannels</p> <p>預設：allchannels</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：與叢集相關聯的現有 Redis OSS 使用者將繼續擁有現有的許可。更新使用者或重新啟動叢集以更新現有的 Redis OSS 使用者。</p>	部署到此叢集 ACL 之使用者的預設 pubsub 頻道許可。
<code>cluster-allow-reads-when-down</code> (added in 6.0)	<p>預設：否</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	<p>設定為是時，即使節點無法達到主要數量，Redis OSS ( 啟用叢集模式 ) 複寫群組仍會繼續處理讀取命令。</p> <p>當設定為預設值 no 時，複寫群組會拒絕所有命令。如果您使用的叢集少於三個節點群組，或者您的應用程式可以安全處理過時的讀取作業，建議您將此值設定為 yes。</p>
<code>tracking-table-max-keys</code> (added in 6.0)	<p>預設值：1,000,000</p> <p>類型：數字</p> <p>可修改：是</p> <p>變生效：直接套用至叢集中所有節點。</p>	<p>為了協助用戶端快取，Redis OSS 支援追蹤哪些用戶端已存取哪些金鑰。</p> <p>追蹤的索引鍵有所修改時，會傳送失效訊息給所有用戶端，通知他們快取的值不再有效。此值可讓您指定此資料表的上限。超過此參數值之後，會隨機傳送失效訊息給用戶端。系統應調校此值以限制記憶體使用量，同時仍會追蹤足夠的索引鍵。索引鍵也會在記憶體不足的情況下失效。</p>
<code>acllog-max-len</code>	<p>預設：128</p>	此值對應至 ACL 日誌中的項目數量上限。

詳細資訊	描述	
(added in 6.0)	類型：數字  可修改：是  變生效：直接套用至叢集中所有節點。	
active-expire-effort (added in 6.0)	預設：1  類型：數字  可修改：是  變生效：直接套用至叢集中所有節點。	<p>Redis OSS會刪除超過兩個機制存活時間的金鑰。一種機制是系統會存取一個索引鍵，並發現其過期。另一種機制則是定期任務對索引鍵進行取樣，而導致超過存留時間的索引鍵過期。此參數會定義 Redis OSS 用來使定期任務中的項目過期的努力量。</p> <p>預設值 1 會嘗試避免有超過 10% 的過期索引鍵仍存在於記憶體中。也會嘗試避免佔用總記憶體的 25% 以上以及為系統增加延遲。您最多可以將此值增加 10，以增加使索引鍵過期花費的工作量。權衡更高CPU，也可能更高延遲。除非您看到高記憶體用量，並且可以容忍CPU使用率增加，否則建議值為 1。</p>
lazyfree-lazy-user-del (added in 6.0)	預設：否  類型：字串  可修改：是  變生效：直接套用至叢集中所有節點。	<p>將值設定為 yes 時，DEL 命令的作用與 UNLINK 相同。</p>

在 Redis OSS 6.x 中移除的參數如下所示。

名稱	詳細資訊	描述
lua-replicate-commands	允許值：是/否 預設：是 類型：布林值 可修改：是 變更生效：立即生效	是否一律在 Lua 指令碼中啟用 Lua 效果複寫

### Redis OSS 5.0.3 參數變更

參數群組系列：redis5.0

#### Redis OSS 5.0 預設參數群組

- `default.redis5.0` – 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。
- `default.redis5.0.cluster.on` – 將此參數群組或衍生的參數群組用於 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組。

#### 在 Redis 5.0.3 OSS 中新增的參數

名稱	詳細資訊	描述
rename-commands	預設：none 類型：字串 可修改：是 變更生效：直接套用至叢集中所有節點。	重新命名 Redis OSS 命令的空間分隔清單。下列限制命令清單可供重新命名使用：  APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BR POPLUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS

名稱	詳細資訊	描述
		<p>GEORADIUSBYMEMBER GET GETBIT  GETRANGE GETSET HDEL HEXISTS  HGET HGETALL HINCRBY HINCRBYFL  OAT HKEYS HLEN HMGET HMSET HSET  HSETNX HSTRLEN HVALS INCR INCRBY  INCRBYFLOAT INFO KEYS LASTSAVE  LINDEX LINSERT LLEN LPOP LPU  SH LPUSHX LRANGE LREM LSET LTRIM  MEMORY MGET MONITOR MOVE MSET  MSETNX MULTI OBJECT PERSIST PEXPIRE  PEXPIREAT PFADD PFCOUNT PFMERGE  PING PSETEX PSUBSCRIBE PUBSUB PTTL  PUBLISH PUNSUBSCRIBE RANDOMKEY  READONLY READWRITE RENAME RENAMENX  RESTORE ROLE RPOP RPOPLPUSH  RPUSH RPUSHX SADD SCARD SCRIPT  SDIFF SDIFFSTORE SELECT SET  SETBIT SETEX SETNX SETRANGE  SINTER SINTERSTORE SISMEMBER  SLOWLOG SMEMBERS SMOVE SORT SPOP  SRANDMEMBER SREM STRLEN SUBSCRIBE  UNION UNIONSTORE SWAPDB  TIME TOUCH TTL TYPE UNSUBSCRIBE  UNLINK UNWATCH WAIT WATCH ZADD  ZCARD ZCOUNT ZINCRBY ZINTERSTO  RE ZLEXCOUNT ZPOPMAX ZPOPMIN  ZRANGE ZRANGEBYLEX ZREVRANGE  BYLEX ZRANGEBYSCORE ZRANK ZREM  ZREMRANGEBYLEX ZREMRANGEBYRANK  ZREMRANGEBYSCORE ZREVRANGE  ZREVRANGEBYSCORE ZREVRANK ZSCORE  ZUNIONSTORE SCAN SSCAN HSCAN  ZSCAN XINFO XADD XTRIM XDEL XRA  NGE XREVRANGE XLEN XREAD XGROUP</p>

名稱	詳細資訊	描述
		XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_ RO LOLWUT XSETID SUBSTR

如需詳細資訊，請參閱[ElastiCache \( Redis OSS \) 5.0.6 版 \( 增強版 \)](#)。

## Redis OSS 5.0.0 參數變更

參數群組系列：redis5.0

### Redis OSS 5.0 預設參數群組

- `default.redis5.0` – 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。
- `default.redis5.0.cluster.on` – 將此參數群組或衍生的參數群組用於 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組。

### 在 Redis 5.0 OSS 中新增的參數

名稱	詳細資訊	描述
<code>stream-node-max-bytes</code>	允許的值：0+ 預設：4096 類型：整數 可修改：是 變更生效：立即生效	串流資料結構是節點基數樹狀結構，它會在內部編碼多個項目。使用這個組態可指定基數樹狀結構中單一節點的最大大小 (以位元組為單位)。如果設為 0，則節點的大小沒有限制。
<code>stream-node-max-entries</code>	允許的值：0+ 預設：100 類型：整數	串流資料結構是節點基數樹狀結構，它會在內部編碼多個項目。使用此組態來指定在附加新串流項目時，切換至新節點之前，單一節點可包含的最大項目數。如果設為 0，則樹狀結構節點中的項目數沒有限制

名稱	詳細資訊	描述
	可修改：是 變更生效：立即生效	
active-defrag-max-scan-fields	允許值：1 到 1000000 預設：1000 類型：整數 可修改：是 變更生效：立即生效	將從主字典掃描處理的 set/hash/zset/list 欄位的最大數量
lua-replicate-commands	允許值：是/否 預設：是 類型：布林值 可修改：是 變更生效：立即生效	是否一律在 Lua 指令碼中啟用 Lua 效果複寫
replica-ignore-maxmemory	預設：是 類型：布林值 可修改：否	不從主要複本獨立移出項目，決定複本是否要忽略 maxmemory 設定

Redis OSS已重新命名引擎 5.0 版中的多個參數，以回應社群意見回饋。如需詳細資訊，請參閱 [Redis 5 OSS 中的新功能？](#)。下表列出新名稱以及它們與舊版的對應方式。



## 在 Redis 5.0 OSS 中重新命名的參數

名稱	詳細資訊	描述
replica-lazy-flush	<p>預設：是</p> <p>類型：布林值</p> <p>可修改：否</p> <p>舊名稱：slave-lazy-flush</p>	在複本同步期間執行非同步 flushDB。
client-output-buffer-limit-replica-hard-limit	<p>預設：如需數值，請參閱 <a href="#">Redis OSS 節點類型特定參數</a></p> <p>類型：整數</p> <p>可修改：否</p> <p>舊名稱：client-output-buffer-limit-slave-hard-limit</p>	對於 Redis OSS 僅供讀取複本：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線。
client-output-buffer-limit-replica-soft-limit	<p>預設：如需數值，請參閱 <a href="#">Redis OSS 節點類型特定參數</a></p> <p>類型：整數</p> <p>可修改：否</p> <p>舊名稱：client-output-buffer-limit-slave-soft-limit</p>	對於 Redis 僅供 OSS 讀取複本：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線，但只有在仍存在此情況時，才會中斷連線 client-output-buffer-limit-replica-soft-seconds 。
client-output-buffer-limit-replica-soft-seconds	<p>預設：60</p> <p>類型：整數</p> <p>可修改：否</p> <p>舊名稱：client-output-buffer-limit-slave-soft-seconds</p>	對於 Redis OSS 僅供讀取複本：如果用戶端的輸出緩衝區保留在 client-output-buffer-limit-replica-soft-limit 位元組超過此秒數，則會中斷用戶端的連線。

名稱	詳細資訊	描述
replica-allow-chaining	預設：否 類型：字串 可修改：否 舊名稱：slave-allow-chaining	決定 Redis 中的僅供讀取複本是否可以 OSS 擁有自己的僅供讀取複本。
min-replicas-to-write	預設：0 類型：整數 可修改：是 舊名稱：min-slaves-to-write 變生效：立即	為了讓主要節點接受來自用戶端的寫入，所需要的最低可用僅供讀取複本數量。若可用複本數低於此數字，屆時主要節點便會不再接受寫入請求。 如果此參數或 min-replicas-max-lag 為 0，則即使沒有可用的複本，主要節點仍會接受寫入請求。
min-replicas-max-lag	預設：10 類型：整數 可修改：是 舊名稱：min-slaves-max-lag 變生效：立即	秒數，在此期間主要節點必須從僅供讀取複本接收 ping 請求。若經過此秒數而主要節點仍未接收到 ping，便會將複本視為無法使用。如果可用複本的數量低於 min-replicas-to-write，則主要複本會在該時間點停止接受寫入。 如果此參數或 min-replicas-to-write 為 0，則即使沒有可用的複本，主要節點仍會接受寫入請求。
close-on-replica-write	預設：是 類型：布林值 可修改：是 舊名稱：close-on-slave-write 變生效：立即	若啟用，則會中斷嘗試寫入僅供讀取複本的用戶端連線。

## 在 Redis OSS 5.0 中移除參數

名稱	詳細資訊	描述
repl-timeout	預設：60  可修改：否	此參數在此版本中無法使用。

## Redis OSS 4.0.10 參數變更

參數群組系列：redis4.0

### Redis OSS 4.0.x 預設參數群組

- `default.redis4.0` – 針對 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。
- `default.redis4.0.cluster.on` – 針對 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集和複寫群組，使用此參數群組，或從中衍生的參數群組。

## 在 Redis OSS 4.0.10 中變更的參數

名稱	詳細資訊	描述
maxmemory-policy	允許值：allkeys-lru、volatile-lru、 <b>allkeys-lfu</b> 、 <b>volatile-lfu</b> 、allkeys-random、volatile-random、volatile-ttl 及 noeviction  預設：volatile-lru  類型：字串  可修改：是	2.6.13 版中新增了 maxmemory-policy。在 4.0.10 版中，新增了兩個新的許可值：allkeys-lfu，它將使用近似值 LFU、和來清除任何金鑰 volatile-lfu，並在具有過期集的金鑰 LFU 之間使用近似值來規避。在 6.2 版本中，當導入 r6gd 節點系列以與資料分層搭配使用時，r6gd 節點類型僅支援 noeviction、volatile-lru 和 allkeys-lru 最大記憶體政策。

名稱	詳細資訊	描述
	變更生效：立即生效	

### 在 Redis OSS 4.0.10 中新增的參數

名稱	詳細資訊	描述
非同步刪除參數		
lazyfree-lazy- eviction	允許值：是/否 預設：否 類型：布林值 可修改：是 變更生效：立即生效	在移出時執行非同步刪除。
lazyfree-lazy-expire	允許值：是/否 預設：否 類型：布林值 可修改：是 變更生效：立即生效	在過期鍵上執行非同步刪除。
lazyfree-lazy-server-del	允許值：是/否 預設：否 類型：布林值 可修改：是 變更生效：立即生效	針對更新數值的命令執行非同步刪除。

名稱	詳細資訊	描述
<code>slave-lazy-flush</code>	<p>允許值：N/A</p> <p>預設：否</p> <p>類型：布林值</p> <p>可修改：否</p> <p>變生效：不適用</p>	在從屬同步 (slave sync) 期間執行非同步 flushDB。
LFU 參數		
<code>lfu-log-factor</code>	<p>允許值：任何 &gt; 0 的整數</p> <p>預設：10</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：立即生效</p>	設定日誌因數，決定使鍵計數器飽和的鍵命中次數。
<code>lfu-decay-time</code>	<p>允許值：任何整數</p> <p>預設：1</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：立即生效</p>	用來遞減鍵計數器的時間 (以分鐘計)。
作用中磁碟重組參數		

名稱	詳細資訊	描述
activedefrag	允許值：是/否 預設：否 類型：布林值 可修改：是 變生效：立即生效	啟用的主動磁碟重組。
active-defrag-ignore-bytes	允許值：10485760-104857600 預設：104857600 類型：整數 可修改：是 變生效：立即生效	啟動主動磁碟重組所需要的最低分散廢棄物數量。
active-defrag-threshold-lower	允許值：1 到 100 預設：10 類型：整數 可修改：是 變生效：立即生效	啟動主動磁碟重組所需要的最低分散百分比。

名稱	詳細資訊	描述
active-defrag-threshold-upper	允許值：1 到 100 預設：100 類型：整數 可修改：是 變生效：立即生效	進行最大程度投入量所需要的最高分散百分比。
active-defrag-cycle-min	允許值：1 到 75 預設：25 類型：整數 可修改：是 變生效：立即生效	以CPU百分比為單位進行重組的最小努力。
active-defrag-cycle-max	允許值：1 到 75 預設：75 類型：整數 可修改：是 變生效：立即生效	以CPU百分比為單位進行重組的最大努力。
用戶端輸出緩衝區參數		

名稱	詳細資訊	描述
client-query-buffer-limit	允許值：1048576-1073741824  預設：1073741824  類型：整數  可修改：是  變生效：立即生效	單一用戶端查詢緩衝區的大小上限。
proto-max-bulk-len	允許值：1048576-536870912  預設：536870912  類型：整數  可修改：是  變生效：立即生效	單一元素請求的大小上限。

### Redis OSS 3.2.10 參數變更

參數群組系列：redis3.2

ElastiCache ( Redis OSS ) 3.2.10 不支援其他參數。

### Redis OSS 3.2.6 參數變更

參數群組系列：redis3.2

對於 Redis OSS 3.2.6，不支援其他參數。

### Redis OSS 3.2.4 參數變更

參數群組系列：redis3.2

從 Redis OSS 3.2.4 開始，有兩個預設參數群組。



- `default.redis3.2` – 執行 Redis OSS 3.2.4 時，如果您想要建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組，但仍使用 Redis 3.2.4 OSS 的其他功能，請指定此參數群組或從中衍生的參數群組。
- `default.redis3.2.cluster.on` – 當您想要建立 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組時，請指定此參數群組或從中衍生的參數群組。

## 主題

- [Redis OSS 3.2.4 的新參數](#)
- [Redis OSS 3.2.4 中變更的參數 \( 增強版 \)](#)

## Redis OSS 3.2.4 的新參數

參數群組系列：redis3.2

對於 Redis OSS 3.2.4，支援下列其他參數。

名稱	詳細資訊	描述
<code>list-max-ziplist-size</code>	預設值：-2 類型：整數 可修改：否	<p>清單會以特別的方式進行編碼以節省空間。每個內部清單節點允許的項目數量可透過固定的大小上限或元素數量上限來指定。針對固定的大小上限，請使用 -5 到 -1，表示：</p> <ul style="list-style-type: none"> <li>• -5：大小上限：64 Kb - 不建議用於一般工作負載</li> <li>• -4：大小上限：32 Kb - 不建議</li> <li>• -3：大小上限：16 Kb - 不建議</li> <li>• -2：大小上限：8 Kb - 建議</li> <li>• -1：大小上限：4 Kb - 建議</li> <li>• 正值表示在每個清單節點中存放最多該數量的元素。</li> </ul>

名稱	詳細資訊	描述
list-compress-depth	預設：0 類型：整數 可修改：是 變更生效：立即	<p>清單也可以進行壓縮。壓縮深度是來自清單每一端的快速清單 (quicklist) 壓縮清單 (ziplist) 節點數量，這些節點會從壓縮中排除。清單的前端和尾端一律不會進行壓縮，以進行快速的推送及彈出操作。設定如下：</p> <ul style="list-style-type: none"> <li>• 0：停用所有壓縮。</li> <li>• 1：啟動壓縮，從第一個節點進入前端和尾端。  <pre>[head]-&gt;node-&gt;node-&gt;...-&gt;node-&gt;[tail]</pre>           除了 [head] 和 [tail] 之外的所有節點都會進行壓縮。</li> <li>• 2：啟動壓縮，從第二個節點進入前端和尾端。  <pre>[head]-&gt;[next]-&gt;node-&gt;node-&gt;...-&gt;node-&gt;[prev]-&gt;[tail]</pre>           [head]、[next]、[prev] 和 [tail] 不會進行壓縮。其他所有節點都會壓縮。</li> <li>• 其他等服務...</li> </ul>

名稱	詳細資訊	描述
cluster-enabled	<p>預設：否/是 *</p> <p>類型：字串</p> <p>可修改：否</p>	<p>指出這是叢集模式 OSS ( 是 ) 中的 Valkey 或 Redis ( 啟用叢集模式 ) 複寫群組，還是非叢集模式 OSS ( 否 ) 中的 Valkey 或 Redis ( 啟用叢集模式 ) 複寫群組。叢集模式中的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組可以將資料分割至最多 500 個節點群組。</p> <p>* Redis OSS 3.2.x 有兩個預設參數群組。</p> <ul style="list-style-type: none"> <li>• default.redis3.2 – 預設值 no。</li> <li>• default.redis3.2.cluster.on – 預設值 yes。</li> </ul>
cluster-require-full-coverage	<p>預設：否</p> <p>類型：布林值</p> <p>可修改：是</p> <p>變更生效：立即</p>	<p>將設定為時yes，如果叢集模式中的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 節點偵測到至少有一個未覆蓋的雜湊插槽 ( 沒有可用的節點正在服務它 )，則會停止接受查詢。這會在叢集部分關閉時導致叢集無法使用。它會在所有位置再次受到涵蓋時重新恢復可用狀態。</p> <p>但是，有時候您可能會希望一部分正常運作的叢集針對仍然涵蓋的 keypace 部分繼續接受查詢。若要這麼做，只需將 cluster-require-full-coverage 選項設為 no 即可。</p>

名稱	詳細資訊	描述
hll-spars e-max-byt es	預設：3000 類型：整數 可修改：是 變更生效：立即	<p>HyperLogLog 稀疏表示位元組限制。限制包含 16 位元組的標頭。當 HyperLogLog 使用稀疏表示法的 超過此限制時，它會轉換為密集表示法。</p> <p>不建議使用大於 16000 的值，因為屆時密集表示可以更有效率的使用記憶體。</p> <p>我們建議值約為 3000，以享有節省空間編碼的好處，而不會太慢 PFADD，也就是 <math>O(N)</math> 搭配稀疏編碼。當 CPU 不是問題時，該值可以提高到 ~10000，但空間是，而且資料集由許多 HyperLogLogs 組成，其基數介於 0 - 15000 範圍內。</p>
reserved- memory-pe rcent	預設：25 類型：整數 可修改：是 變更生效：立即	<p>為非資料用途預留的節點記憶體百分比。依預設，Redis OSS 資料佔用空間會成長，直到消耗節點的所有記憶體為止。若發生此情況，則節點效能很可能會因過度進行記憶體分頁而降低。透過保留記憶體，您可以將一些可用的記憶體留給非 Redis OSS 用途，以協助減少分頁數量。</p> <p>此參數是 特有的 ElastiCache，不屬於標準 Redis OSS 分佈。</p> <p>如需詳細資訊，請參閱 <a href="#">reserved-memory</a> 和 <a href="#">管理 Valkey 和 Redis 的預留記憶體 OSS</a>。</p>

## Redis OSS 3.2.4 中變更的參數 ( 增強版 )

參數群組系列：redis3.2

對於 Redis OSS 3.2.4，下列參數已變更。

名稱	詳細資訊	變更
activereshashing	可修改：如果參數群組未與任何快取叢集相關聯，則為「是」。否則為否。	「可修改」先前為「否」。
databases	可修改：如果參數群組未與任何快取叢集相關聯，則為「是」。否則為否。	「可修改」先前為「否」。
appendonly	預設：關閉 可修改：否	如果您想要從較早的 Redis OSS 版本升級，您必須先appendonly 關閉。
appendfsync	預設：關閉 可修改：否	如果您想要從較早的 Redis OSS 版本升級，您必須先appendfsync 關閉。
repl-timeout	預設：60 可修改：否	現已變更為無法修改，且其預設為 60。
tcp-keepalive	預設：300	先前預設為 0。
list-max-ziplist-entries		參數已無法使用。
list-max-ziplist-value		參數已無法使用。

## Redis OSS 2.8.24 (增強) 新增參數

參數群組系列：redis2.8

對於 Redis OSS 2.8.24，不支援其他參數。

## Redis OSS 2.8.23 ( 增強 ) 新增的參數

參數群組系列：redis2.8

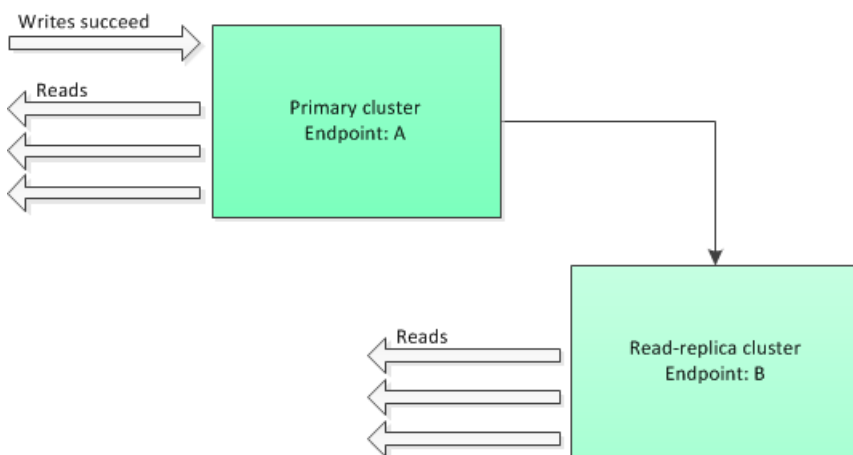
對於 Redis OSS 2.8.23，支援下列其他參數。

名稱	詳細資訊	描述
close-on-slave-write	預設：是 類型：字串 (是/否) 可修改：是 變生效：立即	若啟用，則會中斷嘗試寫入僅供讀取複本的用戶端連線。

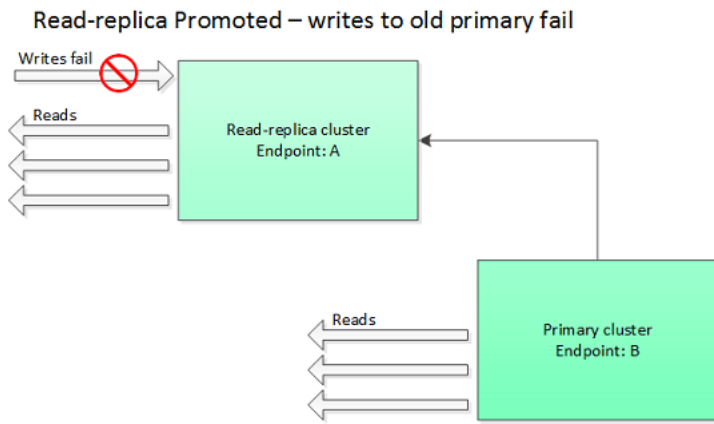
## 運作方式 close-on-slave-write

Amazon 會引入 close-on-slave-write 參數 ElastiCache，讓您更能控制叢集在主要節點和僅供讀取複本節點交換角色將僅供讀取複本提升為主要節點時如何回應。

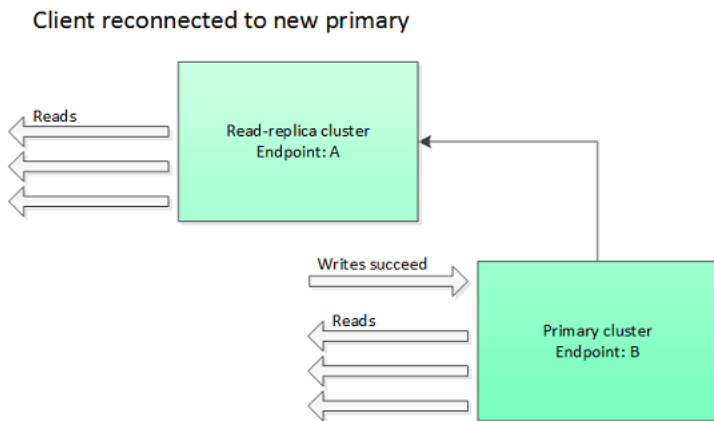
Before read-replica promotion



若僅供讀取複本叢集因啟用異地同步備份的複寫群組容錯移轉之外的任何原因提升為主要節點，則用戶端會繼續嘗試寫入端點 A。因為端點 A 已成為僅供讀取複本的端點，因此寫入會失敗。這是 ElastiCache 引進 OSS 之前 Redis 的行為，close-on-replica-write 以及停用時的行為 close-on-replica-write。



啟用 `close-on-replica-write` 時，每一次當用戶端嘗試寫入僅供讀取複本時，便會關閉用戶端對叢集的連線。您的應用程式邏輯應該會偵測中斷連線、檢查DNS資料表，然後重新連線至主要端點，現在將是端點 B。



### 何時可以停用 `close-on-replica-write`

若停用 `close-on-replica-write` 對導致寫入失敗的叢集，為什麼還要停用 `close-on-replica-write`？

如先前所提及，啟用 `close-on-replica-write` 時，每一次當用戶端嘗試寫入僅供讀取複本時，便會關閉用戶端對叢集的連線。建立與節點的新連線需要時間。因此，寫入複本請求所導致的中斷連線及重新連線也會影響透過相同連線處理的讀取請求延遲。此效果會持續存在，直到建立新連線為止。若您的應用程式特別需要進行大量讀取或是對延遲非常敏感，您可能會偏好將用戶端保持連線，以避免降低讀取效能。

### Redis OSS 2.8.22 (增強) 新增的參數

參數群組系列：redis2.8

對於 Redis OSS 2.8.22，不支援其他參數。

**⚠ Important**

- 從 Redis 2.8.22 OSS版開始，`repl-backlog-size`會套用至主要叢集以及複本叢集。
- 從 Redis 2.8.22 OSS版開始，不支援 `repl-timeout` 參數。如果變更，ElastiCache 會以預設（60 秒）覆寫，就像使用 `appendonly` 一樣。

以下參數已不再受到支援。

- `appendonly`
- `appendfsync`
- `repl-timeout`

**Redis OSS 2.8.21 新增參數**

參數群組系列：redis2.8

對於 Redis OSS 2.8.21，不支援其他參數。

**Redis OSS 2.8.19 新增參數**

參數群組系列：redis2.8

對於 Redis OSS 2.8.19，不支援其他參數。

**Redis OSS 2.8.6 新增參數**

參數群組系列：redis2.8

對於 Redis OSS 2.8.6，支援下列其他參數。

名稱	詳細資訊	描述
<code>min-slaves-max-lag</code>	預設：10 類型：整數 可修改：是 變生效：立即	秒數，在此期間主要節點必須從僅供讀取複本接收 ping 請求。若經過此秒數而主要節點仍未接收到 ping，便會將複本視為無法使用。如果可用複本的數量低於 <code>min-slave</code>



名稱	詳細資訊	描述
		<p>s-to-write，則主要複本會在該時間點停止接受寫入。</p> <p>如果此參數或 min-slaves-to-write 為 0，則即使沒有可用的複本，主要節點仍會接受寫入請求。</p>
min-slaves-to-write	<p>預設：0</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：立即</p>	<p>為了讓主要節點接受來自用戶端的寫入，所需要的最低可用僅供讀取複本數量。若可用複本數低於此數字，屆時主要節點便會不再接受寫入請求。</p> <p>如果此參數或 min-slaves-max-lag 為 0，則即使沒有可用的複本，主要節點仍會接受寫入請求。</p>

名稱	詳細資訊	描述
notify-keyspace-events	預設：(空字串) 類型：字串 可修改：是 變生效：立即	<p>Redis OSS可以通知用戶端的鍵空間事件類型。每個事件類型都是以單一字母表示：</p> <ul style="list-style-type: none"> <li>• K - Keyspace 事件，以 __keyspace@&lt;db&gt;__ 字首發佈</li> <li>• E - 索引鍵事件，以 __keyevent@&lt;db&gt;__ 字首發佈</li> <li>• g — 一般、非特定命令，例如 DEL、EXPIRE、RENAME等。</li> <li>• \$ - 字串命令</li> <li>• l - 列出命令</li> <li>• s - 設定命令</li> <li>• h - 雜湊命令</li> <li>• z - 排序集命令</li> <li>• x - 過期事件 (每次索引鍵過期時產生的事件)</li> <li>• e - 移出事件 (為 maxmemory 移出索引鍵時產生的事件)</li> <li>• A - g\$lshzxe 的別名</li> </ul> <p>您可以任意組合這些事件類型。例如，AKE 表示 Redis OSS可以發佈所有事件類型的通知。</p>

名稱	詳細資訊	描述
		<p>請不要使用任何以上列出字元以外的字元。嘗試執行此作業會導致錯誤訊息。</p> <p>根據預設，此參數會設為空字串，表示停用 keyspace 事件通知。</p>
repl-backlog-size	<p>預設：1048576</p> <p>類型：整數</p> <p>可修改：是</p> <p>變更生效：立即</p>	<p>主要節點待處理項目緩衝區的大小 (位元組)。待處理項目會用於記錄主要節點資料的更新。當僅供讀取複本連線到主要節點時，它會嘗試執行部分同步 (psync)，套用來自待處理項目的資料來更新至主要節點目前的情況。若 psync 失敗，便需要完整同步。</p> <p>此參數的最小值為 16384。</p> <div data-bbox="1008 1003 1507 1276" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>從 Redis OSS 2.8.22 開始，此參數適用於主要叢集和僅供讀取複本。</p> </div>
repl-backlog-ttl	<p>預設：3600</p> <p>類型：整數</p> <p>可修改：是</p> <p>變更生效：立即</p>	<p>主要節點保留待處理項目緩衝區的秒數。從最後一次複本節點中斷連線的時間開始，待處理項目中的資料會保持不變，直到 repl-backlog-ttl 過期。若複本在此期間內並未連線到主要節點，主要節點便會釋放待處理項目緩衝區。當複本最後重新連線時，便需要與主要節點執行完整同步。</p> <p>若將此參數設為 0，便永遠不會釋放待處理項目緩衝區。</p>

名稱	詳細資訊	描述
repl-timeout	預設：60 類型：整數 可修改：是 變更生效：立即	表示以下項目的逾時期間 (秒數)： <ul style="list-style-type: none"> <li>從僅供讀取複本視角，於同步期間的大量資料傳輸</li> <li>從複本視角的主要節點逾時</li> <li>從主要節點視角的複本逾時</li> </ul>

## Redis OSS 2.6.13 參數

參數群組系列：redis2.6

Redis OSS 2.6.13 是 OSS 支援的第一個 Redis 版本 ElastiCache。下表顯示 ElastiCache 支援的 Redis OSS 2.6.13 參數。

名稱	詳細資訊	描述
activeresharding	預設：是 類型：字串 (是/否) 可修改：是 變更生效：建立時	判斷是否啟用 Redis 的主動重新雜湊功能。主要雜湊資料表每秒修復十次；每個堆疊操作耗用 1 毫秒 CPU 的時間。  您可以在建立參數群組時設定此值。將新的參數群組指派給叢集時，此值在舊的及新的參數群組中都必須相同。
appendonly	預設：否 類型：字串 可修改：是 變更生效：立即	啟用或停用 Redis 的僅附加檔案功能 (AOF)。AOF 會擷取變更快取中資料的任何 Redis OSS 命令，並用於從某些節點故障中復原。  預設值為 <code>no</code> ，表示 AOF 已關閉。將此參數設定為 <code>yes</code> 以啟用 AOF。  如需詳細資訊，請參閱 <a href="#">減少故障</a> 。

名稱	詳細資訊	描述
		<p> Note</p> <p>cache.t1.micro 和 cache.t2.* 節點不支援僅附加檔案 ( AOF )。針對此類型的節點，appendonly 參數值會遭到忽略。</p> <p> Note</p> <p>對於多可用區域複寫群組，AOF 是不允許的。</p>
appendfsync	<p>預設：everysec</p> <p>類型：字串</p> <p>可修改：是</p> <p>變生效：立即</p>	<p>當 appendonly 設定為是時，會控制AOF輸出緩衝區寫入磁碟的頻率：</p> <ul style="list-style-type: none"> <li>• no - 視需要將緩衝區排清至磁碟。</li> <li>• everysec - 每秒會排清一次緩衝區。此為預設值。</li> <li>• always - 每次修改叢集中的資料時都排清緩衝區。</li> <li>• 2.8.22 及更新版本不支援 appendfsync。</li> </ul>
client-output-buffer-limit-normal-hard-limit	<p>預設：0</p> <p>類型：整數</p> <p>可修改：是</p> <p>變生效：立即</p>	<p>若用戶端的輸出緩衝區達到指定的位元組數，便會中斷用戶端連線。預設為零 (無硬式限制)。</p>

名稱	詳細資訊	描述
<code>client-output-buffer-limit-normal-soft-limit</code>	預設：0 類型：整數 可修改：是 變更生效：立即	若用戶端的輸出緩衝區達到指定的位元組數，便會中斷用戶端連線，但只有在此條件持續達 <code>client-output-buffer-limit-normal-soft-seconds</code> 時。預設為零 (無軟式限制)。
<code>client-output-buffer-limit-normal-soft-seconds</code>	預設：0 類型：整數 可修改：是 變更生效：立即	若用戶端的輸出緩衝區維持在 <code>client-output-buffer-limit-normal-soft-limit</code> 位元組超過此秒數，便會中斷用戶端連線。預設為零 (無時間限制)。
<code>client-output-buffer-limit-pubsub-hard-limit</code>	預設：33554432 類型：整數 可修改：是 變更生效：立即	對於 Redis OSS 發佈/訂閱用戶端：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線。
<code>client-output-buffer-limit-pubsub-soft-limit</code>	預設：8388608 類型：整數 可修改：是 變更生效：立即	對於 Redis OSS 發佈/訂閱用戶端：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線，但只有在 仍存在此情況時，才會中斷連線 <code>client-output-buffer-limit-pubsub-soft-seconds</code> 。
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	預設：60 類型：整數 可修改：是 變更生效：立即	對於 Redis OSS 發佈/訂閱用戶端：如果用戶端的輸出緩衝區保留在 <code>client-output-buffer-limit-pubsub-soft-limit</code> 位元組超過此秒數，用戶端將會中斷連線。

名稱	詳細資訊	描述
client-output-buffer-limit-slave-hard-limit	預設：如需數值，請參閱 <a href="#">Redis OSS節點類型特定參數</a> 類型：整數 可修改：否	對於 Redis 僅供OSS讀取複本：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線。
client-output-buffer-limit-slave-soft-limit	預設：如需數值，請參閱 <a href="#">Redis OSS節點類型特定參數</a> 類型：整數 可修改：否	對於 Redis 僅供OSS讀取複本：如果用戶端的輸出緩衝區達到指定的位元組數，用戶端將會中斷連線，但只有在持續發生此情況時，才會中斷連線client-output-buffer-limit-slave-soft-seconds。
client-output-buffer-limit-slave-soft-seconds	預設：60 類型：整數 可修改：否	對於 Redis OSS僅供讀取複本：如果用戶端的輸出緩衝區保留在client-output-buffer-limit-slave-soft-limit 位元組超過此秒數，則會中斷用戶端的連線。
databases	預設：16 類型：整數 可修改：否 變生效：建立時	資料庫分割成的邏輯分割區數量。我們建議將此值維持在較低程度。  您可以在建立參數群組時設定此值。將新的參數群組指派給叢集時，此值在舊的及新的參數群組中都必須相同。
hash-max-ziplist-entries	預設：512 類型：整數 可修改：是 變生效：立即	決定用於雜湊的記憶體數量。少於指定項目數的雜湊會使用特別的編碼存放，以節省空間。

名稱	詳細資訊	描述
hash-max-ziplist-value	預設：64 類型：整數 可修改：是 變生效：立即	決定用於雜湊的記憶體數量。項目小於指定位元組數的雜湊會使用特別的編碼存放，以節省空間。
list-max-ziplist-entries	預設：512 類型：整數 可修改：是 變生效：立即	決定用於清單的記憶體數量。少於指定項目數的清單會使用特別的編碼存放，以節省空間。
list-max-ziplist-value	預設：64 類型：整數 可修改：是 變生效：立即	決定用於清單的記憶體數量。項目小於指定位元組數的清單會使用特別的編碼存放，以節省空間。
lua-time-limit	預設：5000 類型：整數 可修改：否	Lua 指令碼在 ElastiCache 採取動作停止指令碼之前的最大執行時間，以毫秒為單位。  lua-time-limit 如果超過，所有 Redis OSS 命令都會傳回格式 ____-BUSY 的錯誤。由於此狀態可能導致許多必要的 Redis OSS 操作受到干擾，ElastiCache 會先發出 SCRIPT KILL 命令。如果失敗，ElastiCache 會強制重新啟動 Redis OSS。
maxclients 此值適用於所有執行個體類型，但明確指定的除外	預設：65000 類型：整數 可修改：否	一次可連線的用戶端數量上限。



名稱	詳細資訊	描述
	<p>t2.medium 預設：20000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t2.small 預設：20000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t2.micro 預設：20000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t4g.micro 預設：20000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t3.medium 預設：46000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t3.small 預設：46000</p> <p>類型：整數</p> <p>可修改：否</p>	
	<p>t3.micro 預設：20000</p> <p>類型：整數</p> <p>可修改：否</p>	

名稱	詳細資訊	描述
maxmemory-policy	預設：volatile-lru 類型：字串 可修改：是 變更生效：立即	到達記憶體用量上限時，針對鍵的移出政策。  有效值為：volatile-lru   allkeys-lru   volatile-random   allkeys-random   volatile-ttl   noeviction  如需詳細資訊，請參閱 <a href="#">使用 Valkey 或 Redis OSS 作為LRU快取</a> 。
maxmemory-samples	預設：3 類型：整數 可修改：是 變更生效：立即	對於 least-recently-used (LRU) 和 time-to-live (TTL) 計算，此參數代表要檢查的金鑰範例大小。依預設，Redis OSS會選擇 3 個金鑰，並使用至少最近使用過的金鑰。
reserved-memory	預設：0 類型：整數 可修改：是 變更生效：立即	為非資料用量預留的總記憶體 (位元組)。依預設，Redis OSS節點會成長，直到消耗節點的 maxmemory (請參閱 <a href="#">Redis OSS節點類型特定參數</a> )。若發生此情況，則節點效能很可能會因過度進行記憶體分頁而降低。透過保留記憶體，您可以為非 Redis OSS目的預留一些可用的記憶體，以協助減少分頁數量。  此參數是特有的 ElastiCache，不屬於標準 Redis OSS分佈。  如需詳細資訊，請參閱 reserved-memory-percent 和 <a href="#">管理 Valkey 和 Redis 的預留記憶體 OSS</a> 。

名稱	詳細資訊	描述
set-max-intset-entries	預設：512 類型：整數 可修改：是 變更生效：立即	決定要用於特定類型組 (基數為 10，介於 64 位元帶正負號整數範圍內整數的字串) 的記憶體數量。這類少於指定項目數的組會使用特別的編碼存放，以節省空間。
slave-allow-chaining	預設：否 類型：字串 可修改：否	決定 Redis 中的僅供讀取複本是否可以 OSS 擁有自己的僅供讀取複本。
slowlog-log-slower-than	預設：10000 類型：整數 可修改：是 變更生效：立即	Redis OSS Slow Log 功能記錄命令的最長執行時間，以微秒為單位。
slowlog-max-len	預設：128 類型：整數 可修改：是 變更生效：立即	Redis OSS Slow Log 的長度上限。
tcp-keepalive	預設：0 類型：整數 可修改：是 變更生效：立即	若將此設為非零值 (N)，節點用戶端便會每 N 秒輪詢一次，確保仍然持續連線。使用預設設定的 0，便步會發生任何輪詢。 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b>              此參數的某些方面在 Redis 3.2.4 OSS 版中已變更。請參閱 <a href="#">Redis OSS 3.2.4 中變更的參數 (增強版)</a>。</p> </div>

名稱	詳細資訊	描述
timeout	預設：0 類型：整數 可修改：是 變更生效：立即	節點在逾時前等待的秒數。數值為： <ul style="list-style-type: none"> <li>• 0 - 永不中斷閒置用戶端的連線。</li> <li>• 1-19 - 無效值。</li> <li>• <math>\geq 20</math> - 節點在中斷與閒置用戶端連線前所等待的秒數。</li> </ul>
zset-max-ziplist-entries	預設：128 類型：整數 可修改：是 變更生效：立即	決定用於排序組的記憶體數量。少於指定元素數的排序組會使用特別的編碼存放，以節省空間。
zset-max-ziplist-value	預設：64 類型：整數 可修改：是 變更生效：立即	決定用於排序組的記憶體數量。項目小於指定位元組數的排序組會使用特別的編碼存放，以節省空間。

### Note

如果您未指定 Redis 2.6.13 OSS 叢集的參數群組，則會使用預設參數群組 ( `default.redis2.6` )。您無法變更預設參數群組中任何參數的值。但是，您永遠可以建立自訂參數群組並隨時將其指派給您的叢集。

## Redis OSS節點類型特定參數

雖然大多數的參數都只有單一值，有些參數則可能會根據所使用的節點類型而有不同的值。下表顯示每個節點類型 `maxmemory`、`client-output-buffer-limit-slave-hard-limit` 及 `client-`

`output-buffer-limit-slave-soft-limit` 參數的預設值。`maxmemory` 的值為您在節點上可以用於資料及其他用途的位元組上限。如需可用指標的詳細資訊，請參閱[可用記憶體](#)。

**Note**

`maxmemory` 參數無法修改。

節點類型	Maxmemory	C lient-output-buffe r-limit-slave-hard- limit	C lient-output-buffe r-limit-slave-soft- limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g.small	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880

節點類型	Maxmemory	C lient-output-buffe r-limit-slave-hard- limit	C lient-output-buffe r-limit-slave-soft- limit
cache.m2.2xlarge	35022438400	3502243840	3502243840
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172

節點類型	Maxmemory	C lient-output-buffe r-limit-slave-hard- limit	C lient-output-buffe r-limit-slave-soft- limit
cache.m6g.2xlarge	27966669210	2796666921	2796666921
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3.2xlarge	62495129600	6081740800	6081740800
cache.r3.4xlarge	126458265600	12268339200	12268339200
cache.r3.8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970

節點類型	Maxmemory	C lient-output-buffe r-limit-slave-hard- limit	C lient-output-buffe r-limit-slave-soft- limit
cache.r5.2xlarge	56711183565	5671118356	5671118356
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6gd.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6gd.12xlarge	341206346547	34120634655	34120634655
cache.r6gd.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970



節點類型	Maxmemory	C lient-output-buffe r-limit-slave-hard- limit	C lient-output-buffe r-limit-slave-soft- limit
cache.r7g.2xlarge	56711183565	5671118356	5671118356
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

**Note**

預設VPC情況下，所有目前世代的執行個體類型都會在 Amazon Virtual Private Cloud 中建立。

T1 執行個體不支援多個可用區。

T1 和 T2 執行個體不支援 Redis OSS AOF。

Redis 2.8.22 版appendonly及更新OSS版本appendfsync不支援 Redis OSS組態變數和。

## Memcached 專用參數

### Memcached

若您沒有為 Memcached 叢集指定參數群組，則會使用適合您引擎版本的預設參數群組。您無法變更預設參數群組中任何參數的值。但是，您可以建立自訂參數群組並隨時將其指派給您的叢集。如需詳細資訊，請參閱[建立 ElastiCache 參數群組](#)。

### 主題

- [Memcached 1.6.17 的變更](#)
- [Memcached 1.6.6 版新增參數](#)
- [Memcached 1.5.10 參數變更](#)
- [Memcached 1.4.34 版新增參數](#)
- [Memcached 1.4.33 版新增參數](#)
- [Memcached 1.4.24 版新增參數](#)
- [Memcached 1.4.14 版新增參數](#)
- [Memcached 1.4.5 版支援參數](#)
- [Memcached 連線額外負荷](#)
- [Memcached 節點類型專用參數](#)

### Memcached 1.6.17 的變更

從 Memcached 1.6.17 開始，我們不再支援下列管理命令：lru\_crawler、lru 和 slabs。這些變更生效後，您就無法在執行期透過命令啟用/停用 lru\_crawler。請透過修改自訂參數群組來啟用/停用 lru\_crawler。

## Memcached 1.6.6 版新增參數

針對 Memcached 1.6.6，沒有支援任何額外參數。

參數群組系列：memcached1.6

## Memcached 1.5.10 參數變更

針對 Memcached 1.5.10 版，支援以下額外參數。

參數群組系列：memcached1.5

名稱	詳細資訊	描述
no_modern	<p>預設：1</p> <p>類型：布林值</p> <p>可修改：是</p> <p>允許的值：0、1</p> <p>變生效：啟動時</p>	<p>用於停用 slab_reassign、lru_maintainer_thread、lru_segmented 和 maxconns_fast 命令的別名。</p> <p>使用 Memcached 1.5 和更新版本時，no_modern 也會將 hash_algorithm 設定為 jenkins。</p> <p>此外，使用 Memcached 1.5.10 時，inline_ascii_reponse 是由參數控制 parallelly。這表示如果 no_modern 已停用，則 inline_ascii_response 會停用。從 Memcached 引擎 1.5.16 起，inline_ascii_response 參數不再適用，因此 no_modern 啟用或停用對沒有影響 inline_ascii_response。</p>

名稱	詳細資訊	描述
		<p>如果 <code>no_modern</code> 已停用，則會 <code>maxconns_fast</code> <code>WILL</code> 啟用 <code>slab_reassign</code>、<code>lru_segmented</code>、<code>lru_maintainer_thread</code> 和。由於 <code>slab_automove</code> 和 <code>hash_algorithm</code> 參數不是 <code>SWITCH</code> 參數，因此其設定是以參數群組中的組態為基礎。</p> <p>如果您想要停用 <code>no_modern</code> 並還原至 <code>modern</code>，您必須設定自訂參數群組以停用此參數，然後重新啟動，這些變更才會生效。</p> <div data-bbox="1008 894 1507 1545" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px;"> <p> <b>Note</b></p> <p>自 2021 年 8 月 20 日起，此參數的預設組態值已從 0 變更為 1。更新後的預設值會在 2021 年 8 月 20 日之後，由每個區域的新 ElastiCache 使用者自動取得。在 2021 年 8 月 20 日之前，區域中的現有 ElastiCache 使用者需要手動修改其自訂參數群組，才能取得此新變更。</p> </div>

名稱	詳細資訊	描述
<code>inline_ascii_resp</code>	預設：0 類型：布林值 可修改：是 允許的值：0、1 變生效：啟動時	在項目內存放來自 VALUE 回應的數字，使用最多 24 個位元組。get、ASCII faster 集的小慢。

針對 Memcached 1.5.10，會移除以下參數。

名稱	詳細資訊	描述
<code>expirezero_does_no_t_evict</code>	預設：0 類型：布林值 可修改：是 允許的值：0、1 變生效：啟動時	此版本中已不再支援。
<code>modern</code>	預設：1 類型：布林值 可修改：是 (如果設為 <code>no_modern</code> ，則需要重新啟動) 允許的值：0、1 變生效：啟動時	此版本中已不再支援。從此版本開始，每次啟動或重新啟動時， <code>no-modern</code> 預設會啟用。

## Memcached 1.4.34 版新增參數

針對 Memcached 1.4.34，沒有支援任何額外參數。

參數群組系列：memcached1.4

## Memcached 1.4.33 版新增參數

針對 Memcached 1.4.33 版，支援以下額外參數。

參數群組系列：memcached1.4

名稱	詳細資訊	描述
modern	預設：啟用 類型：布林值 可修改：是 變生效：啟動時	多種功能的別名。啟用 modern 等同於開啟以下命令，並使用 murmur3 雜湊演算法：slab_reassign、slab_auto_move、lru_crawler、lru_maintainer、maxconns_fast 及 hash_algorithm=murmur3。
watch	預設：啟用 類型：布林值 可修改：是 變生效：立即 若使用者到達其 watcher_logbuf_size 與 worker_logbuf_size 限制，則日誌可能會遭到卸除。	日誌擷取、移出或突變。例如，當使用者開啟 watch 時，他們會在 get、set、delete 或 update 發生時看到日誌。

名稱	詳細資訊	描述
<code>idle_timeout</code>	預設：0 (停用) 類型：整數 可修改：是 變生效：啟動時	在詢問關閉前允許用戶端閒置的最低秒數。數值範圍：0 到 86400。
<code>track_sizes</code>	預設：停用 類型：布林值 可修改：是 變生效：啟動時	顯示每個 Slab 群組使用的大小。 啟用 <code>track_sizes</code> 可讓您執行 <code>stats sizes</code> ，而無須執行 <code>stats sizes_enable</code> 。
<code>watcher_logbuf_size</code>	預設：256 (KB) 類型：整數 可修改：是 變生效：啟動時	<code>watch</code> 命令會開啟 Memcached 的串流日誌。但是， <code>watch</code> 可能會在移出率、突變率或擷取率過高，足以導致日誌緩衝區填滿時卸除日誌。在這種情況下，使用者可增加緩衝區大小來減少遺失日誌的機率。
<code>worker_logbuf_size</code>	預設：64 (KB) 類型：整數 可修改：是 變生效：啟動時	<code>watch</code> 命令會開啟 Memcached 的串流日誌。但是， <code>watch</code> 可能會在移出率、突變率或擷取率過高，足以導致日誌緩衝區填滿時卸除日誌。在這種情況下，使用者可增加緩衝區大小來減少遺失日誌的機率。

名稱	詳細資訊	描述
slab_chunk_max	預設：524288 (位元組) 類型：整數 可修改：是 變生效：啟動時	指定 Slab 的大小上限。設定較小的 Slab 大小以更有效率地使用記憶體。大於 slab_chunk_max 的項目會分割成多個 Slab。
lru_crawler metadump [all 1 2 3]	預設：停用 類型：布林值 可修改：是 變生效：立即	若啟用 lru_crawler，則此命令會傾印所有鍵。  all 1 2 3 - 所有 Slab，或是指定特定的 Slab 編號。

## Memcached 1.4.24 版新增參數


針對 Memcached 1.4.24 版，支援以下額外參數。

參數群組系列：memcached1.4

名稱	詳細資訊	描述
disable_flush_all	預設：0 (停用) 類型：布林值 可修改：是 變生效：啟動時	新增參數 (-F) 來停用 flush_all。在您永遠不想在生產執行個體上執行完整排清時很有用。  數值：0、1 (當此值為 0 時，使用者便可執行 flush_all )。
hash_algorithm	預設：jenkins 類型：字串 可修改：是	要使用的雜湊演算法。允許值：murmur3 及 jenkins。



名稱	詳細資訊	描述
	變更生效：啟動時	
<code>lru_crawler</code>	<p>預設：0 (停用)</p> <p>類型：布林值</p> <p>可修改：是</p> <p>變更生效：重新啟動後</p> <div data-bbox="678 688 922 1117" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>您可以於執行時間從命令列暫時啟用 <code>lru_crawler</code>。如需詳細資訊，請參閱「描述」欄。</p> </div>	<p>清理已過期的項目 Slab 類別。此為在背景執行且影響程度低的程序。目前需要使用手動命令初始化網路爬取。</p> <p>若要暫時啟用，請在命令列執行 <code>lru_crawler enable</code>。</p> <p><code>lru_crawler 1,3,5</code> 會網路爬取 Slab 類別 1、3 及 5，尋找過期的項目，並新增到釋放清單。</p> <p>數值：0、1</p> <div data-bbox="1036 961 1479 1344" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>在命令列啟用 <code>lru_crawler</code> 會啟用網路爬取程式，直到在命令列停用或下一次重新開機為止。若要永久啟用，您必須修改參數值。如需詳細資訊，請參閱<a href="#">修改 ElastiCache 參數群組</a>。</p> </div>
<code>lru_maintainer</code>	<p>預設：0 (停用)</p> <p>類型：布林值</p> <p>可修改：是</p> <p>變更生效：啟動時</p>	<p>已達到LRUs容量時，在之間隨機切換項目的背景執行緒。數值：0、1。</p>

名稱	詳細資訊	描述
expirezero_does_no_t_evict	預設：0 (停用) 類型：布林值 可修改：是 變生效：啟動時	搭配 lru_maintainer 使用時，讓過期時間為 0 的項目無法移出。 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;">  <b>Warning</b>              這可能會填滿其他可移出項目的可用記憶體。           </div> 可設為略過 lru_maintainer 。

### Memcached 1.4.14 版新增參數

針對 Memcached 1.4.14 版，支援以下額外參數。

參數群組系列：memcached1.4

### Memcached 1.4.14 版中新增的參數

名稱	描述
config_max	ElastiCache 組態項目的數量上限。

名稱	描述
<code>config_size_max</code>	組態項目的大小上限，單位為位元組。
<code>hashpower_init</code>	雜 ElastiCache 湊表的初始大小，表示為 2 的功率。預設為 16 ( $2^{16}$ )，或 65536 個鍵。

名稱	描述
maxconns_fast	<p>變更到達連線數量上限時處理新連線請求的方式。若將此參數設為 0 (零)，則新的連線會新增到待處理項目佇列，並等待其他連線關閉。如果參數設定為 1，ElastiCache 則將錯誤傳送至用戶端，並立即關閉連線。</p>

名稱	描述
slab_automove	<p>調整 Slab 自動移動演算法：若將此參數設為 0 (零)，則停用自動移動演算法。如果設定為 1，ElastiCache 會採取緩慢、保守的方法自動移動厚切塊。如果將其設定為 2，則每當有驅逐時，ElastiCache 積極移動厚切塊。(除非用於測試用途，否則不建議此模式。)</p>

名稱	描述
slab_reassign	啟用或停用 Slab 指派。若將此參數設為 1，您可以使用 "slabs reassign" 命令來手動重新指派記憶體。

## Memcached 1.4.5 版支援參數

參數群組系列：memcached1.4

針對 Memcached 1.4.5 版，支援以下參數。

## Memcached 1.4.5 版中新增的參數

名稱	詳細資訊	描述
backlog_queue_limit	預設：1024 類型：整數 可修改：否	待處理佇列限制。
binding_protocol	預設：自動 類型：字串 可修改：是 變生效：重新啟動後	繫結通訊協定。 允許值：ascii 及 auto。 如需修改 binding_protocol 值的指導，請參閱 <a href="#">修改 ElastiCache 參數群組</a> 。
cas_disabled	預設值：0 (false) 類型：布林值 可修改：是 變生效：重新啟動後	如果 1 (true)，檢查並設定 (CAS) 操作將會停用，而且儲存的項目比CAS啟用時耗用較少 8 個位元組。
chunk_size	預設：48 類型：整數 可修改：是 變生效：重新啟動後	配置給最小項目鍵、值及標記的空間數量下限 (位元組)。
chunk_size_growth_factor	預設：1.25 類型：浮動 可修改：是 變生效：重新啟動後	成長因數，控制每個後續 Memcached 區塊的大小。每個區塊都會比前一個區塊大 chunk_size_growth_factor 倍。

名稱	詳細資訊	描述
error_on_memory_exhausted	預設值：0 (false) 類型：布林值 可修改：是 變生效：重新啟動後	若為 1 (True)，則當沒有記憶體可存放項目時，Memcached 會傳回錯誤而非移出項目。
large_memory_pages	預設值：0 (false) 類型：布林值 可修改：否	如果 1 ( true )，ElastiCache 會嘗試使用大型記憶體頁面。
lock_down_paged_memory	預設值：0 (false) 類型：布林值 可修改：否	如果 1 ( true )，ElastiCache 會鎖定所有分頁記憶體。
max_item_size	預設：1048576 類型：整數 可修改：是 變生效：重新啟動後	可存放在叢集中的最大項目大小 (位元組)。
max_simultaneous_connections	預設：65000 類型：整數 可修改：否	同時連線數量上限。
maximize_core_file_limit	預設值：0 (false) 類型：布林值 可修改： 變生效：重新啟動後	如果 1 ( true )，ElastiCache 會最大化核心檔案限制。



名稱	詳細資訊	描述
memcached_connections_overhead	預設：100 類型：整數 可修改：是 變生效：重新啟動後	要為 Memcached 連線及其他雜項額外負荷預留的記憶體數量。如需此參數的相關資訊，請參閱 <a href="#">Memcached 連線額外負荷</a> 。
requests_per_event	預設：20 類型：整數 可修改：否	指定連線每個事件的請求數上限。此限制為避免資源耗盡的必要項目。

## Memcached 連線額外負荷

在每個節點上，可供存放項目的記憶體為該節點可用記憶體的總數 (該值存放在 `max_cache_memory` 參數中)，減去用於連線及其他額外負荷 (該值存放在 `memcached_connections_overhead` 參數中) 所使用的記憶體。例如，`cache.m1.small` 類型的節點具有 1300MB 的 `max_cache_memory`。使用預設 100MB 的 `memcached_connections_overhead` 值，Memcached 程序便會擁有 1200MB，可用於存放項目。

`memcached_connections_overhead` 參數的預設值足以滿足大多數的使用案例。但是，連線額外負荷所需要的配置數量可能會因多項因素而不同，包括請求率、承載大小，以及連線數量。

您可以變更 `memcached_connections_overhead` 的值來更進一步配合您應用程式的需求。例如，增加 `memcached_connections_overhead` 參數的值會減少可用於存放項目的記憶體數量，為連線額外負荷提供更大的緩衝區。減少 `memcached_connections_overhead` 參數的值可讓您擁有更多記憶體來存放項目，但可能會增加切換 (swap) 用量及降低效能。若您察覺切換 (swap) 用量及效能降低，請嘗試增加 `memcached_connections_overhead` 參數的值。

### Important

針對 `cache.t1.micro` 節點類型，`memcached_connections_overhead` 的值會透過以下方式決定：

- 如果您叢集使用預設參數群組，ElastiCache 會將 的值設定為 `memcached_connections_overhead` 13MB。

- 若您的叢集使用您自行建立的參數群組，您可以將 `memcached_connections_overhead` 的值設為您所選擇的值。

## Memcached 節點類型專用參數


雖然大多數的參數都只有單一值，有些參數則可能會根據所使用的節點類型而有不同的值。下表顯示每個節點類型 `max_cache_memory` 及 `num_threads` 參數的預設值。這些參數上的值都無法修改。

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1
cache.t2.medium	3301	2
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

 Note

所有 T2 執行個體都是在 Amazon Virtual Private Cloud ( Amazon VPC ) 中建立。

## 擴展 ElastiCache

您可以擴展 ElastiCache 快取以符合您的需求。無伺服器快取和自行設計的叢集提供多種不同的擴展選項。

### Scaling ElastiCache Serverless

ElastiCache Serverless 會在工作負載流量上升或下降時自動容納工作負載流量。對於每個無 ElastiCache 伺服器快取，ElastiCache 會持續追蹤資源的使用率CPU，例如、記憶體和網路。當任何這些資源受到限制時，ElastiCache Serverless 會透過新增碎片並重新分發資料至新的碎片來橫向擴展，而不會讓應用程式停機。您可以透過監控快取資料儲存的指標和 `ElastiCacheProcessingUnits (ECPU) BytesUsedForCache` 的運算用量，CloudWatch 來監控 中快取所使用的資源。

### 設定擴展限制以管理成本

您可以選擇在快取資料儲存上設定最大用量，並為快取設定ECPU每秒用量，以控制快取成本。這樣做將確保您的快取用量永遠不會超過所設定的上限。

如果您設定擴展上限，則當快取達到上限時，應用程式可能會遇到快取效能降低的情況。當您設定快取資料儲存上限，且快取資料儲存達到上限時，ElastiCache 將會使用 LRU 邏輯，開始逸出快取中已設定 Time-To-Live (TTL) 的資料。如果沒有可以清除的資料，則寫入其他資料的請求將會收到記憶體不足 (OOM) 錯誤訊息。當您設定ECPU每秒上限，且工作負載的運算使用率超過此值時，ElastiCache 將開始限流請求。

如果您在 `BytesUsedForCache`或 `ElastiCacheProcessingUnits` 上設定了上限，強烈建議在低於上限的值設定CloudWatch 警示，以便在快取操作接近這些限制時收到通知。建議設定的警示值為上限的 75%。請參閱有關如何設定 CloudWatch 警示的文件。

### 使用 ElastiCache Serverless 預先擴展

#### ElastiCache 無伺服器預先擴展

透過預先擴展，也稱為預熱，您可以設定 ElastiCache 快取的支援限制下限。您可以為每秒的 ElastiCache 處理單位 (ECPUs) 或資料儲存設定這些最小值。這對於準備預期的擴展事件很有用。例如，如果遊戲公司預期在新遊戲啟動的第一分鐘內登入次數會增加 5 倍，他們可以為快取做好準備，以因應用量大幅增加的情況。

您可以使用 ElastiCache 主控台、CLI或執行預先擴展API。無 ElastiCache 伺服器會在 60 分鐘內更新快取上可用的ECPUs每秒 / 秒，並在完成最低限制更新時傳送事件通知。

## 預先擴展的運作方式

當透過主控台、或 CLI 更新 ECPUs/秒或資料儲存的下限時 API，該新限制可在 1 小時內使用。ElastiCache Serverless 支援空快取的 30KECPUs/秒，以及使用從複本讀取功能時的 90KECPUs/秒。每 10-12 分鐘 ElastiCache 可以加倍 ECPUs/秒。此擴展速度足以應付大多數工作負載。如果您預期即將發生的擴展事件可能超過此速率，則建議您將最小 ECPUs/秒設定為峰值事件至少 60 分鐘之前預期的峰值 ECPUs/秒。否則，應用程式可能會遇到延遲增加和請求限流。

一旦下限更新完成，ElastiCache Serverless 將開始測量您 ECPUs 每秒的新下限或新的最低儲存。即使您的應用程式未在快取上執行請求，或您的資料儲存用量低於下限，也會發生這種情況。當您從目前的設定降低下限時，更新會立即進行，因此 ElastiCache Serverless 將立即開始以新的下限計量。

### Note

- 當您設定最低用量限制時，即使實際用量低於最低用量限制，仍需支付該限制的費用。ECPU 或超過最低用量限制的資料儲存用量會按一般費率收費。例如，如果您將最低用量限制設定為 ECPUs 每秒 100,000，則每小時至少需支付 1.224 美元（使用 us-east-1 ECPU 的價格），即使您的用量低於該最低用量。
- ElastiCache Serverless 支援快取上彙總層級的請求最小規模。ElastiCache Serverless 也支援每個插槽最多 30KECPUs/秒（使用 READONLY 連線從複本讀取時為 90KECPUs/秒）。作為最佳實務，您的應用程式應確保金鑰在 Valkey 或 Redis OSS 插槽和金鑰之間的流量盡可能一致。

## 使用主控台 和 設定擴展限制 AWS CLI

### 使用 AWS 主控台設定擴展限制

- 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
- 在導覽窗格中，選擇您要修改之快取上執行的引擎。
- 接著會出現執行所選引擎的快取清單。
- 選擇快取名稱左側的選項按鈕來選擇要修改的快取。
- 選擇 Actions (動作)，然後選擇 Modify (修改)。
- 在用量限制下，設定適當的記憶體或運算限制。
- 按一下預覽變更，然後儲存變更。

## 使用 設定擴展限制 AWS CLI

若要使用 變更擴展限制CLI，請使用 modify-serverless-cache API。

Linux：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

## 使用 移除擴展限制 CLI

若要使用 移除擴展限制CLI，請將下限和上限參數設定為 0。

Linux：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

## 擴展自我設計的叢集

您的應用程式需要處理的資料量通常是動態的。它會隨著您的業務成長或遇到需求的一般波動而增加或減少。如果您自行管理快取，您必須為需求尖峰佈建足夠的硬體，這可能所費不貲。透過使用 Amazon ElastiCache，您可以擴展以滿足目前需求，只需支付使用的費用。ElastiCache 可讓您擴展快取以符合需求。



**Note**

如果 Valkey 或 Redis OSS 叢集在一或多個區域中複寫，則這些區域會依序擴展。擴展時，次要區域會先擴展，然後擴展主要區域。縮減規模時，主要區域會先是，然後是任何次要區域。更新引擎版本時，順序為次要區域，然後是主要區域。

**主題**

- [Memcached 的擴展叢集](#)
- [擴展 Valkey 或 Redis 的叢集 OSS \( 停用叢集模式 \)](#)
- [擴展 Valkey 或 Redis 的複本節點 OSS \( 停用叢集模式 \)](#)
- [在 Valkey 或 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)

**Memcached 的擴展叢集**

下表可協助您找到要執行擴展動作的正確主題。

**擴展 Memcached 叢集**

Memcached 叢集由 1 到 60 個節點組成。向外和向內擴展 Memcached 叢集，就如同從叢集新增或移除節點般容易。

因為您可以在 Memcached 叢集的所有節點間分割資料，您很少需要向上擴展至具有更大記憶體節點類型。不過，因為 Memcached 引擎不會持續保存資料，如果您確實擴展至不同的節點類型，您的新叢集一開始會是空白的，除非您的應用程式填入資料。

**擴展 Memcached 叢集**

動作	主題
向外擴展	<a href="#">將節點新增至叢集</a>
向內擴展	<a href="#">從叢集刪除節點</a>
變更節點類型	<a href="#">垂直擴展 Memcached</a>

Memcached 叢集由 1 到 60 個節點組成。向外和向內擴展 Memcached 叢集，就如同從叢集新增或移除節點般容易。

因為您可以在 Memcached 叢集的所有節點間分割資料，您很少需要向上擴展至具有更大記憶體節點類型。不過，因為 Memcached 引擎不會持續保存資料，如果您確實擴展至不同的節點類型，您的新叢集一開始會是空白的，除非您的應用程式填入資料。

## 主題

- [水平擴展 Memcached](#)
- [垂直擴展 Memcached](#)

### 水平擴展 Memcached

Memcached 引擎支援在多個節點間分割您的資料。因此，Memcached 叢集可輕鬆水平擴展。Memcached 叢集可以有 1 到 60 個節點。若要水平擴展 Memcached 叢集，只需要新增或移除節點。

下列主題詳細說明如何透過新增或移除節點，向外或向內擴展 Memcached 叢集。

- [將節點新增至叢集](#)
- [從您的叢集刪除節點](#)

每次在 Memcached 叢集中變更節點的數量，您必須至少為您的一些金鑰空間進行重新映射，使得它可映射到正確的節點。如需有關負載平衡 Memcached 叢集的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以進行有效的負載平衡 \( Memcached \)](#)。

如果是在 Memcached 叢集上使用自動搜索，則不需在您新增或移除節點時，變更應用程式中的端點。如需自動搜索的詳細資訊，請參閱 [自動識別叢集中的節點 \( Memcached \)](#)。如果您不使用自動搜索，每次在 Memcached 叢集中變更節點的數量時，即必須更新應用程式中的端點。

### 垂直擴展 Memcached

向上或向下擴展 Memcached 叢集時，您必須建立新的叢集。Memcached 叢集一開始一律是空白的，除非您的應用程式填入資料。

#### Important

如果您要向下擴展至較小的節點類型，請確定該較小的節點類型對您的資料和成本而言已足夠。如需詳細資訊，請參閱[選取快取節點大小](#)。

## 主題

- [垂直擴展 Memcached \(主控台\)](#)
- [垂直擴展 Memcached \(AWS CLI\)](#)
- [垂直擴展 Memcached \(ElastiCache API\)](#)

### 垂直擴展 Memcached (主控台)

下列程序會逐步引導您使用 ElastiCache 主控台垂直擴展叢集。

#### 垂直擴展 Memcached 叢集 (主控台)

1. 使用新的節點類型來建立新叢集。如需詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。
2. 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱[尋找叢集的端點 \(主控台\) \( Memcached \)](#)。
3. 刪除舊叢集。如需詳細資訊，請參閱[刪除 Memcached 中的新節點](#)。

### 垂直擴展 Memcached (AWS CLI)

下列程序會逐步引導您使用 AWS CLI，來垂直擴展 Memcached 快取叢集。

#### 垂直擴展 Memcached 快取叢集 (AWS CLI)

1. 使用新的節點類型來建立新快取叢集。如需詳細資訊，請參閱[使用 建立叢集CLI](#)。
2. 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱[尋找端點 \(AWS CLI\)](#)。
3. 刪除舊快取叢集。如需詳細資訊，請參閱[使用 AWS CLI 刪除 ElastiCache 叢集](#)。

### 垂直擴展 Memcached (ElastiCache API)

下列程序會引導您使用 垂直擴展 Memcached 快取叢集 ElastiCache API。

#### 垂直擴展 Memcached 快取叢集 ( ElastiCache API )

1. 使用新的節點類型來建立新快取叢集。如需詳細資訊，請參閱 [建立 Memcached 的叢集 \( ElastiCache API \)](#)。
2. 在應用程式中，將端點更新為新快取叢集的端點。如需詳細資訊，請參閱[尋找端點 \( ElastiCache API \)](#)。
3. 刪除舊快取叢集。如需詳細資訊，請參閱[使用 ElastiCache API](#)。

## 擴展 Valkey 或 Redis 的叢集 OSS ( 停用叢集模式 )

Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集可以是具有 0 個碎片的單節點叢集，或具有 1 個碎片的多節點叢集。單一節點叢集會使用一個節點來同時進行讀取和寫入。多節點叢集一律會有 1 個節點做為讀/寫主節點，以及 0 到 5 個僅供讀取複本節點。

### 主題

- [擴展 Valkey 或 Redis 的單節點叢集 OSS \( 停用叢集模式 \)](#)

### 擴展 Valkey 或 Redis OSS叢集

動作	Valkey 或 Redis OSS ( 停用叢集模式 )	Valkey 或 Redis OSS ( 啟用叢集模式 )
向內擴展	<a href="#">從 ElastiCache 叢集移除節點</a>	<a href="#">在 Valkey 或 Redis 中擴展叢集 OSS ( 已啟用叢集模式 )</a>
向外擴展	<a href="#">將節點新增至叢集</a>	<a href="#">Valkey 或 Redis 的線上重新共享 OSS ( 啟用叢集模式 )</a>
變更節點類型	<p>為更大的節點類型：</p> <ul style="list-style-type: none"> <li>• <a href="#">擴展單節點 Valkey 或 Redis OSS叢集</a></li> <li>• <a href="#">使用複本擴展 Valkey 或 Redis OSS叢集</a></li> </ul> <p>為更小的節點類型：</p> <ul style="list-style-type: none"> <li>• <a href="#">縮減單節點 Valkey 或 Redis OSS叢集</a></li> <li>• <a href="#">使用複本縮減 Valkey 或 Redis OSS叢集</a></li> </ul>	<a href="#">透過修改節點類型來進行線上垂直擴展</a>

動作	Valkey 或 Redis OSS ( 停用叢集模式 )	Valkey 或 Redis OSS ( 啟用叢集模式 )
變更節點群組的數量	Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集不支援	<a href="#">在 Valkey 或 Redis 中擴展叢集 OSS ( 已啟用叢集模式 )</a>

## 內容

- [擴展 Valkey 或 Redis 的單節點叢集 OSS \( 停用叢集模式 \)](#)
  - [擴展單節點 Valkey 或 Redis OSS叢集](#)
    - [針對 Valkey 或 Redis 擴展單一節點叢集 OSS \( 停用叢集模式 \) \( 主控台 \)](#)
    - [擴展單節點 Valkey 或 Redis OSS快取叢集 \( AWS CLI \)](#)
    - [擴展單節點 Valkey 或 Redis OSS快取叢集 \( ElastiCache API \)](#)
  - [縮減單節點 Valkey 或 Redis OSS叢集](#)
    - [縮減單節點 Valkey 或 Redis OSS叢集 \( 主控台 \)](#)
    - [縮減單節點 Valkey 或 Redis OSS快取叢集 \( AWS CLI \)](#)
    - [縮減單節點 Valkey 或 Redis OSS快取叢集 \( ElastiCache API \)](#)

## 擴展 Valkey 或 Redis 的單節點叢集 OSS ( 停用叢集模式 )

Valkey 或 Redis OSS ( 停用叢集模式 ) 節點必須足夠大，才能包含快取的所有資料，以及 Valkey 或 Redis OSS額外負荷。若要變更 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集的資料容量，您必須垂直擴展；擴展至較大的節點類型以增加資料容量，或擴展至較小的節點類型以降低資料容量。

ElastiCache 擴展程序旨在盡力保留現有資料，並需要成功的 Valkey 或 Redis OSS複寫。對於 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，建議為 Valkey 或 Redis 提供足夠的記憶體OSS。

您無法跨多個 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集分割資料。不過，如果您只需要增加或減少叢集的讀取容量，則可以使用複本節點建立 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，並新增或移除僅供讀取複本。若要使用單節點 Valkey 或 Redis OSS快取叢集作為主要叢集，建立具有複本節點的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集，請參閱 [建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)。

利用複本建立叢集之後，您可以透過新增僅供讀取複本來增加讀取容量。之後，如果需要，您可以透過移除僅供讀取複本來減少讀取容量。如需詳細資訊，請參閱 [增加讀取容量](#) 或 [減少讀取容量](#)。

除了能夠擴展讀取容量之外，具有複本的 Valkey 或 Redis OSS（停用叢集模式）叢集還提供其他業務優勢。如需詳細資訊，請參閱[使用複寫群組的高可用性](#)。

### Important

如果您的參數群組使用 `reserved-memory` 來為 Valkey 或 Redis OSS 額外負荷預留記憶體，在開始擴展之前，請確定您有一個自訂參數群組，可為您的新節點類型保留正確的記憶體數量。或者，您可以修改自訂參數群組，使得它會對您的新叢集使用 `reserved-memory-percent` 並使用該參數群組。

如果您使用 `reserved-memory-percent`，則不需執行此動作。

如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

## 主題

- [擴展單節點 Valkey 或 Redis OSS 叢集](#)
- [縮減單節點 Valkey 或 Redis OSS 叢集](#)

## 擴展單節點 Valkey 或 Redis OSS叢集

當您擴展單一節點 Valkey 或 Redis OSS叢集時，無論您使用 ElastiCache 主控台、AWS CLI或，ElastiCache 都會執行下列程序 ElastiCache API。

1. 具有新節點類型的新快取叢集會在與現有快取叢集相同的可用區域中向上加速運轉。
2. 現有快取叢集中的快取資料會複製到新快取叢集。此程序需要的時間取決於您的節點類型以及快取叢集中的資料量。
3. 系統會使用新快取叢集提供讀取和寫入。新快取叢集的端點與舊快取叢集中的端點相同，因此您不需要更新應用程式中的端點。當DNS項目更新時，您會注意到主要節點的讀取和寫入短暫中斷（幾秒鐘）。
4. ElastiCache 會刪除舊的快取叢集。您會注意到來自舊節點的讀取和寫入發生短暫中斷（幾秒鐘），因為與舊節點的連線將會中斷。

### Note

針對執行 r6gd 節點類型的叢集，您只能擴展到 r6gd 節點系列中的節點大小。

如下表所示，如果您已為下一個維護時段排程引擎升級，則會封鎖您的 Valkey 或 Redis OSS擴展操作。如需維護時段的詳細資訊，請參閱[管理 ElastiCache 叢集維護](#)。

### 封鎖的 Valkey 或 Redis OSS操作

擱置中操作	封鎖的操作
向上擴展	立即升級引擎
引擎升級	立即向上擴展
向上擴展與升級引擎	立即向上擴展
	立即升級引擎

如果您有正封鎖您的擱置中操作，您可以執行下列其中一項。

- 透過清除立即套用核取方塊（CLI使用：`--no-apply-immediately`，API使用：`ApplyImmediately=false`），為下一個維護時段排程您的 Valkey 或 Redis OSS擴展操作。

- 請等到下一個維護時段（或之後）執行您的 Valkey 或 Redis OSS 擴展操作。
- 使用選取的立即套用核取方塊，將 Valkey 或 Redis OSS 引擎升級新增至此快取叢集修改（CLI 使用：`--apply-immediately`，API 使用：`ApplyImmediately=true`）。這樣會透過讓引擎升級立即執行，來解除封鎖您的向上擴展操作。

您可以使用 ElastiCache 主控台、或擴展單一節點 Valkey AWS CLI 或 Redis OSS（停用叢集模式）叢集 ElastiCache API。

#### Important

如果您的參數群組使用 `reserved-memory` 來為 Valkey 或 Redis OSS 額外負荷預留記憶體，在開始擴展之前，請確定您有一個自訂參數群組，可為您的新節點類型保留正確的記憶體數量。或者，您可以修改自訂參數群組，使得它會對您的新叢集使用 `reserved-memory-percent` 並使用該參數群組。

如果您使用 `reserved-memory-percent`，則不需執行此動作。

如需詳細資訊，請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

針對 Valkey 或 Redis 擴展單一節點叢集 OSS（停用叢集模式）（主控台）

下列程序說明如何使用 ElastiCache 管理主控台擴展單一節點 Valkey 或 Redis OSS 叢集。在此過程中，您的 Valkey 或 Redis OSS 叢集將繼續以最短的停機時間提供請求。

擴展單一節點 Valkey 或 Redis OSS 叢集（主控台）

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 或 Redis OSS 叢集。
3. 從叢集清單中，選擇您要擴展的叢集（它必須執行 Valkey 或 Redis OSS 引擎，而非叢集 Valkey 或 Redis OSS 引擎）。
4. 選擇 Modify (修改)。
5. 在 Modify Cluster (修改叢集) 精靈中：
  - a. 從 Node type (節點類型) 清單選擇您要擴展的節點類型。
  - b. 如果您使用 `reserved-memory` 來管理記憶體，請從 Parameter Group (參數群組) 清單中，選擇會為新節點類型保留正確記憶體數量的自訂參數群組。



6. 如果您要立即執行向上擴展程序，請選擇 Apply immediately (立即套用) 方塊。如果未選擇 Apply immediately (立即套用) 方塊，在此叢集的下一個維護時段期間，會執行向上擴展程序。
7. 選擇 Modify (修改)。

如果您在上一步驟中選擇 Apply immediately (立即套用)，叢集的狀態變更為 modifying (修改中)。當狀態變更為 available (可用)，修改即已完成，並且您可以開始使用新叢集。

### 擴展單節點 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

下列程序說明如何使用 擴展單一節點 Valkey 或 Redis OSS 快取叢集 AWS CLI。在此過程中，您的 Valkey 或 Redis OSS 叢集將繼續以最短的停機時間提供請求。

### 若要擴展單一節點 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

1. 使用下列參數執行 list-allowed-node-type-modifications 命令，決定您可以擴展至的 AWS CLI 節點類型。

- --cache-cluster-id

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --cache-cluster-id my-cache-cluster-id
```

針對 Windows：

```
aws elasticache list-allowed-node-type-modifications ^
 --cache-cluster-id my-cache-cluster-id
```

從上述命令輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ScaleUpModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
```

```

 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
]
 "ScaleDownModifications": [
 "cache.t2.micro",
 "cache.t2.small ",
 "cache.t2.medium ",
 "cache.t1.small ",
],
}

```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在 AWS CLI 參考 中。

2. 使用 AWS CLI `modify-cache-cluster` 命令和下列參數，修改您現有的快取叢集，指定要擴展的快取叢集和新的較大節點類型。
  - `--cache-cluster-id` - 您要擴充規模之快取叢集的名稱。
  - `--cache-node-type` - 您要擴展快取叢集的新節點類型。此值必須是步驟 1 中 `list-allowed-node-type-modifications` 命令傳回的其中一個節點類型。
  - `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `--apply-immediately` - 會造成立即套用擴充規模程序。若要將向上擴展程序延期至叢集的下一個維護時段，請使用 `--no-apply-immediately` 參數。

若為 Linux、macOS 或 Unix：

```

aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cache-cluster \
 --cache-node-type cache.m3.xlarge \
 --cache-parameter-group-name redis32-m2-xl \
 --apply-immediately

```

針對 Windows：

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-redis-cache-cluster ^
 --cache-node-type cache.m3.xlarge ^
 --cache-parameter-group-name redis32-m2-xl ^
 --apply-immediately
```

從上述命令輸出看起來像這樣（JSON 格式）。

```
{
 "CacheCluster": {
 "Engine": "redis",
 "CacheParameterGroup": {
 "CacheNodeIdsToReboot": [],
 "CacheParameterGroupName": "default.redis6.x",
 "ParameterApplyStatus": "in-sync"
 },
 "SnapshotRetentionLimit": 1,
 "CacheClusterId": "my-redis-cache-cluster",
 "CacheSecurityGroups": [],
 "NumCacheNodes": 1,
 "SnapshotWindow": "00:00-01:00",
 "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
 "AutoMinorVersionUpgrade": true,
 "CacheClusterStatus": "modifying",
 "PreferredAvailabilityZone": "us-west-2a",
 "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
 "CacheSubnetGroupName": "default",
 "EngineVersion": "6.0",
 "PendingModifiedValues": {
 "CacheNodeType": "cache.m3.2xlarge"
 },
 "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
 "CacheNodeType": "cache.m3.medium",
 "DataTiering": "disabled"
 }
}
```

如需詳細資訊，請參閱 [modify-cache-cluster](#) 在AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately`，請使用具有下列參數的 命令來 AWS CLI `describe-cache-clusters` 檢查新快取叢集的狀態。當狀態變更為 `available` (可用)，您可以開始使用新的、較大的快取叢集。
  - `--cache-cache cluster-id` – 單節點 Valkey 或 Redis OSS快取叢集的名稱。使用此參數來描述特定快取叢集 (而非所有快取叢集)。

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

如需詳細資訊，請參閱 [describe-cache-clusters](#) 在 AWS CLI 參考 中。

### 擴展單節點 Valkey 或 Redis OSS快取叢集 (ElastiCache API)

下列程序說明如何使用 擴展單節點 Valkey 或 Redis OSS快取叢集 ElastiCache API。在此過程中，您的 Valkey 或 Redis OSS叢集將繼續以最短的停機時間提供請求。

### 若要擴展單一節點 Valkey 或 Redis OSS快取叢集 (ElastiCache API)

1. 使用下列參數執行 `ListAllowedNodeTypeModifications` 動作，決定您可以擴展至的 ElastiCache API 節點類型。
  - `CacheClusterId` – 您要擴展的單節點 Valkey 或 Redis OSS快取叢集的名稱。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&CacheClusterId=MyRedisCacheCluster
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考 中。

2. 使用 `ModifyCacheCluster` ElastiCache API 動作和下列參數，修改現有的快取叢集，指定要擴展的快取叢集和新的較大節點類型。
  - `CacheClusterId` - 您要擴充規模之快取叢集的名稱。

- **CacheNodeType** - 您要為快取叢集擴展規模的較大型新節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypeModifications` 動作傳回的節點類型之一。
- **CacheParameterGroupName** - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
- **ApplyImmediately** - 設定為 `true` 會讓擴充規模程序立即執行。若要將向上擴展程序延期至叢集的下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheCluster
&ApplyImmediately=true
&CacheClusterId=MyRedisCacheCluster
&CacheNodeType=cache.m3.xlarge
&CacheParameterGroupName=redis32-m2-x1
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ModifyCacheCluster](#) 在 Amazon ElastiCache API 參考中。

3. 如果您使用 `ApplyImmediately=true`，請使用具有下列參數 `DescribeCacheClusters` 的動作來 ElastiCache API 檢查新快取叢集的狀態。當狀態變更為 `available` (可用)，您可以開始使用新的、較大的快取叢集。
  - **CacheClusterId** – 單節點 Valkey 或 Redis OSS 快取叢集的名稱。使用此參數來描述特定快取叢集 (而非所有快取叢集)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterId=MyRedisCacheCluster
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeCacheClusters](#) 在 Amazon ElastiCache API 參考 中。

## 縮減單節點 Valkey 或 Redis OSS叢集

下列各節將逐步說明如何將單節點 Valkey 或 Redis OSS叢集縮減為較小的節點類型。確保新的較小節點類型足夠大，足以容納所有資料，而 Valkey 或 Redis OSS額外負荷對於新 Valkey 或 Redis OSS叢集的長期成功至關重要。如需詳細資訊，請參閱[確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)。

### Note

針對執行 r6gd 節點類型的叢集，您只能擴展到 r6gd 節點系列中的節點大小。

### 主題

- [縮減單節點 Valkey 或 Redis OSS叢集 \( 主控台 \)](#)
- [縮減單節點 Valkey 或 Redis OSS快照叢集 \( AWS CLI \)](#)
- [縮減單節點 Valkey 或 Redis OSS快照叢集 \( ElastiCache API \)](#)

### 縮減單節點 Valkey 或 Redis OSS叢集 ( 主控台 )

下列程序會逐步引導您使用 ElastiCache 主控台將單節點 Valkey 或 Redis OSS叢集縮減為較小的節點類型。

### Important

如果您的參數群組使用 `reserved-memory` 來為 Valkey 或 Redis OSS額外負荷預留記憶體，在開始擴展之前，請確定您有一個自訂參數群組，可為您的新節點類型保留正確的記憶體數量。或者，您可以修改自訂參數群組，使得它會對您的新叢集使用 `reserved-memory-percent` 並使用該參數群組。

如果您使用 `reserved-memory-percent`，則不需執行此動作。

如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

### 縮減單節點 Valkey 或 Redis OSS叢集 ( 主控台 )

1. 確保該較小的節點類型對您的資料和成本而言已足夠。
2. 如果您的參數群組使用 `reserved-memory` 來為 Valkey 或 Redis OSS額外負荷預留記憶體，請確定您有自訂參數群組來為新節點類型預留正確的記憶體數量。

或者，您可以將自訂參數群組修改為使用 `reserved-memory-percent`。如需詳細資訊，請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

- 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
- 從叢集的清單，選擇您要縮減規模的叢集。此叢集必須執行 Valkey 或 Redis OSS 引擎，而非叢集 Valkey 或 Redis OSS 引擎。
- 選擇 Modify (修改)。
- 在 Modify Cluster (修改叢集) 精靈中：
  - 從 Node type (節點類型) 清單，選擇您要縮減規模的節點類型。
  - 如果您使用 `reserved-memory` 來管理記憶體，請從 Parameter Group (參數群組) 清單中，選擇會為新節點類型保留正確記憶體數量的自訂參數群組。
- 如果您要立即執行縮減規模程序，請選擇 Apply immediately (立即套用) 核取方塊。如果將 Apply immediately (立即套用) 核取方塊保留未選擇，在此叢集的下一個維護時段期間，會執行縮減規模程序。
- 選擇 Modify (修改)。
- 當叢集的狀態從 `modifying` (修改中) 變更為 `available` (可用)，您的叢集即已擴展為新節點類型。不需要在應用程式中更新端點。

### 縮減單節點 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

下列程序說明如何使用 縮減單節點 Valkey 或 Redis OSS 快取叢集 AWS CLI。

#### 若要縮減單節點 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

- 使用下列參數執行 `list-allowed-node-type-modifications` 命令，AWS CLI 決定您可以縮減規模的節點類型。

- `--cache-cluster-id`

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --cache-cluster-id my-cache-cluster-id
```



## 針對 Windows :

```
aws elasticache list-allowed-node-type-modifications ^
 --cache-cluster-id my-cache-cluster-id
```

從上述命令輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ScaleUpModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
],
 "ScaleDownModifications": [
 "cache.t2.micro",
 "cache.t2.small",
 "cache.t2.medium",
 "cache.t1.small"
],
}
```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在AWS CLI 參考 中。

2. 使用 AWS CLI `modify-cache-cluster` 命令和下列參數，修改您現有的快取叢集，指定要縮減規模的快取叢集和新的較小節點類型。
  - `--cache-cluster-id` - 您要縮減規模之快取叢集的名稱。
  - `--cache-node-type` - 您要擴展快取叢集的新節點類型。此值必須是步驟 1 中 `list-allowed-node-type-modifications` 命令傳回的其中一個節點類型。

- `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
- `--apply-immediately` - 會讓縮減規模程序立即套用。若要將向上擴展程序延期至叢集的下一個維護時段，請使用 `--no-apply-immediately` 參數。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cache-cluster \
 --cache-node-type cache.m3.xlarge \
 --cache-parameter-group-name redis32-m2-xl \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-redis-cache-cluster ^
 --cache-node-type cache.m3.xlarge ^
 --cache-parameter-group-name redis32-m2-xl ^
 --apply-immediately
```

從上述命令輸出看起來像這樣（JSON 格式）。

```
{
 "CacheCluster": {
 "Engine": "redis",
 "CacheParameterGroup": {
 "CacheNodeIdsToReboot": [],
 "CacheParameterGroupName": "default.redis6.x",
 "ParameterApplyStatus": "in-sync"
 },
 "SnapshotRetentionLimit": 1,
 "CacheClusterId": "my-redis-cache-cluster",
 "CacheSecurityGroups": [],
 "NumCacheNodes": 1,
 "SnapshotWindow": "00:00-01:00",
 "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
 "AutoMinorVersionUpgrade": true,
 "CacheClusterStatus": "modifying",
 },
}
```

```
"PreferredAvailabilityZone": "us-west-2a",
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
"CacheSubnetGroupName": "default",
"EngineVersion": "6.0",
"PendingModifiedValues": {
 "CacheNodeType": "cache.m3.2xlarge"
},
"PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled"
}
}
```

如需詳細資訊，請參閱 [modify-cache-cluster](#) 在AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately`，請使用具有下列參數的 命令來 AWS CLI `describe-cache-clusters` 檢查新快取叢集的狀態。當狀態變更為 `available` (可用)，您可以開始使用新的、較大的快取叢集。
  - `--cache-cache cluster-id` – 單節點 Valkey 或 Redis OSS快取叢集的名稱。使用此參數來描述特定快取叢集 (而非所有快取叢集)。

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

如需詳細資訊，請參閱 [describe-cache-clusters](#) 在AWS CLI 參考 中。

## 縮減單節點 Valkey 或 Redis OSS快取叢集 (ElastiCache API)

下列程序說明如何使用 擴展單一節點 Valkey 或 Redis OSS快取叢集 ElastiCache API。

### 若要縮減單節點 Valkey 或 Redis OSS快取叢集 (ElastiCache API)

1. 使用下列參數執行 `ListAllowedNodeTypeModifications` 動作，ElastiCache API 決定您可以縮減規模的節點類型。
  - `CacheClusterId` – 您要縮減規模的單節點 Valkey 或 Redis OSS快取叢集的名稱。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
```

```
&CacheClusterId=MyRedisCacheCluster
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考中。

2. 使用 `ModifyCacheCluster` ElastiCache API 動作和下列參數，修改現有的快取叢集，指定要擴展的快取叢集和新的較大節點類型。
  - `CacheClusterId` - 您要縮減規模之快取叢集的名稱。
  - `CacheNodeType` - 您要將快取叢集縮減規模的新節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypeModifications` 動作傳回的節點類型之一。
  - `CacheParameterGroupName` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `ApplyImmediately` - 設定為 `true` 會造成縮減規模程序立即執行。若要將向上擴展程序延期至叢集的下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheCluster
&ApplyImmediately=true
&CacheClusterId=MyRedisCacheCluster
&CacheNodeType=cache.m3.xlarge
&CacheParameterGroupName redis32-m2-xl
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ModifyCacheCluster](#) 在 Amazon ElastiCache API 參考中。

3. 如果您使用 `ApplyImmediately=true`，請使用具有下列參數 `DescribeCacheClusters` 的動作來 ElastiCache API 檢查新快取叢集的狀態。當狀態變更為 `available` (可用)，您就可以開始使用新的、較小的快取叢集。

- `CacheClusterId` – 單節點 Valkey 或 Redis OSS 快取叢集的名稱。使用此參數來描述特定快取叢集 (而非所有快取叢集)。

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=DescribeCacheClusters
 &CacheClusterId=MyRedisCacheCluster
 &Version=2015-02-02
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20150202T192317Z
 &X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeCacheClusters](#) 在 Amazon ElastiCache API 參考中。

## 擴展 Valkey 或 Redis 的複本節點 OSS ( 停用叢集模式 )

具有複本節點的 Valkey 或 Redis OSS叢集 ( 在 API/ 中稱為複寫群組CLI ) 透過具有啟用自動容錯移轉的多可用區複寫提供高可用性。具有複本節點的叢集是最多六個 Valkey 或 Redis OSS節點的邏輯集合，其中一個節點主要節點能夠同時提供讀取和寫入請求。叢集中的所有其他節點為主要節點的唯一讀複本。寫入主要節點的資料會非同步複寫到叢集中的所有僅供讀取複本。由於 Valkey 或 Redis OSS ( 停用叢集模式 ) 不支援跨多個叢集分割資料，因此 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組中的每個節點都會包含整個快取資料集。Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集支援將資料分割至最多 500 個碎片。

若要變更您叢集的資料容量，您必須將它向上擴展至較大的節點類型，或向下擴展至較小的節點類型。

若要變更您叢集的讀取容量，請新增更多僅供讀取複本 ( 最多 5 個 )，或移除僅供讀取複本。

ElastiCache 擴展程序旨在盡力保留現有資料，並需要成功的 Valkey 或 Redis OSS複寫。對於具有複本的 Valkey 或 Redis OSS叢集，我們建議為 Valkey 或 Redis 提供足夠的記憶體OSS。

### 主題

- [使用複本擴展 Valkey 或 Redis OSS叢集](#)
- [使用複本縮減 Valkey 或 Redis OSS叢集](#)
- [增加讀取容量](#)
- [減少讀取容量](#)

### 相關主題

- [使用複寫群組的高可用性](#)
- [複寫：停用 Valkey 和 Redis OSS 叢集模式與啟用](#)
- [搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS](#)
- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS快照](#)

### 主題

- [使用複本擴展 Valkey 或 Redis OSS叢集](#)
- [使用複本縮減 Valkey 或 Redis OSS叢集](#)
- [增加讀取容量](#)
- [減少讀取容量](#)



## 使用複本擴展 Valkey 或 Redis OSS 叢集

Amazon ElastiCache 提供主控台、CLI 和 API 支援，將您的 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組擴展。

啟動擴展程序時，會 ElastiCache 執行下列動作：

1. 使用新節點類型啟動複寫群組。
2. 將所有資料從目前的主要節點複製到新的主要節點。
3. 將新的僅供讀取複本與新主要節點同步。
4. 更新 DNS 項目，使其指向新節點。因此，您不需在應用程式中更新端點。對於 Valkey 7.2 及更高版本或 Redis OSS 5.0.5 及更高版本，您可以在叢集繼續保持連線並提供傳入請求時擴展自動容錯移轉啟用的叢集。在 Redis 4OSS.0.10 版及更新版本中，您可能注意到 DNS 項目更新時，先前節點的讀取和寫入短暫中斷。
5. 刪除舊節點 ( CLI/API：複寫群組 )。您會注意到來自舊節點的讀取和寫入發生短暫中斷 (幾秒鐘)，因為與舊節點的連線將會中斷。

此程序需要的時間與您的節點類型以及叢集中的資料量有關。

如下表所示，如果您已為叢集的下一個維護時段排程引擎升級，則會封鎖您的 Valkey 或 Redis OSS 擴展操作。

### 封鎖的 Valkey 或 Redis OSS 操作

擱置中操作	封鎖的操作
向上擴展	立即升級引擎
引擎升級	立即向上擴展
向上擴展與升級引擎	立即向上擴展
	立即升級引擎

如果您有正封鎖您的擱置中操作，您可以執行下列其中一項。

- 透過清除立即套用核取方塊 ( CLI 使用：`--no-apply-immediately`，API 使用：`ApplyImmediately=false` )，為下一個維護時段排程您的 Valkey 或 Redis OSS 擴展操作。



- 請等到下一個維護時段（或之後）執行您的 Valkey 或 Redis OSS 擴展操作。
- 使用選取的立即套用核取方塊，將 Valkey 或 Redis OSS 引擎升級新增至此快取叢集修改（CLI 使用：`--apply-immediately`，API 使用：`ApplyImmediately=true`）。這樣會透過讓引擎升級立即執行，來解除封鎖您的向上擴展操作。

下列各節說明如何使用 ElastiCache 主控台、AWS CLI 和 `awscli`，使用複本向上擴展您的 Valkey 或 Redis OSS 叢集 ElastiCache API。

### Important

如果您的參數群組使用 `reserved-memory` 來為 Valkey 或 Redis OSS 額外負荷預留記憶體，在開始擴展之前，請確定您有一個自訂參數群組，可為您的新節點類型保留正確的記憶體數量。或者，您可以修改自訂參數群組，使得它會對您的新叢集使用 `reserved-memory-percent` 並使用該參數群組。

如果您使用 `reserved-memory-percent`，則不需執行此動作。

如需詳細資訊，請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

## 使用複本擴展 Valkey 或 Redis OSS 叢集（主控台）

向上擴展至較大的節點類型所需的時間可能不同，取決於節點類型和您目前叢集中的資料量。

下列程序會使用 ElastiCache 主控台，將叢集的複本從目前的節點類型擴展到新的、較大的節點類型。在此過程中，DNS 項目更新時，主要節點中其他版本的讀取和寫入可能會短暫中斷。對於在 5.0.6 版及更高版本上執行的節點，您可能會看到不到 1 秒的停機時間，而較舊版本則有幾秒的停機時間。

## 使用複本擴展 Valkey 或 Redis OSS 叢集（主控台）

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集
3. 從叢集的清單，選擇您要向上擴展的叢集。此叢集必須執行 Valkey 或 Redis OSS 引擎，而非叢集 Valkey 或 Redis OSS 引擎。
4. 選擇 Modify (修改)。
5. 在 Modify Cluster (修改叢集) 精靈中：
  - a. 從 Node type (節點類型) 清單選擇您要擴展的節點類型。請注意，並非所有節點類型都可縮減規模。

- b. 如果您使用 `reserved-memory` 來管理記憶體，請從 Parameter Group (參數群組) 清單中，選擇會為新節點類型保留正確記憶體數量的自訂參數群組。
6. 如果您要立即執行向上擴展程序，請選擇 `Apply immediately` (立即套用) 核取方塊。如果將 `Apply immediately` (立即套用) 核取方塊保留未選擇，在此叢集的下一個維護時段期間，會執行向上擴展程序。
7. 選擇 `Modify` (修改)。
8. 當叢集的狀態從 `modifying` (修改中) 變更為 `available` (可用)，您的叢集即已擴展為新節點類型。不需要在應用程式中更新端點。

### 擴展 Valkey 或 Redis OSS 複寫群組 ( AWS CLI )

下列程序會使用 AWS CLI，將複寫群組從其目前的節點類型擴展至新的、較大的節點類型。在此過程中，會 ElastiCache 更新 DNS 項目，以便它們指向新的節點。因此，您不需在應用程式中更新端點。對於 Valkey 7.2 及更高版本或 Redis OSS 5.0.5 及更高版本，您可以在叢集繼續保持連線並提供傳入請求時擴展自動容錯移轉啟用的叢集。在 4.0.10 版及更新版本中，您可能會注意到 DNS 項目更新時，從主要節點讀取和寫入先前版本的短暫中斷。

向上擴展至較大的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

### 擴展 Valkey 或 Redis OSS 複寫群組 ( AWS CLI )

1. 使用下列參數執行 `list-allowed-node-type-modifications` 命令，AWS CLI 決定您可以擴展到哪些節點類型。
  - `--replication-group-id` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-repl-group
```

針對 Windows：

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-repl-group
```

此操作的輸出看起來像這樣（JSON 格式）。

```
{
 "ScaleUpModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
]
}
```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在AWS CLI 參考 中。

2. 使用具有下列參數的 AWS CLI `modify-replication-group` 命令，將您目前的複寫群組擴展到新的節點類型。
  - `--replication-group-id` - 複寫群組的名稱。
  - `--cache-node-type` - 此複寫群組中較大的新快取叢集節點類型。此值必須是上一個步驟中 `list-allowed-node-type-modifications` 命令傳回的執行個體類型之一。
  - `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `--apply-immediately` - 會造成立即套用擴充規模程序。若要將向上擴展操作延期至下一個維護時段，請使用 `--no-apply-immediately`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id my-repl-group \
 --cache-node-type cache.m3.xlarge \
```

```
--cache-parameter-group-name redis32-m3-2x1 \
--apply-immediately
```

針對 Windows :

```
aws elasticache modify-replication-group ^
 --replication-group-id my-repl-group ^
 --cache-node-type cache.m3.xlarge ^
 --cache-parameter-group-name redis32-m3-2x1 \
 --apply-immediately
```

此命令的輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ReplicationGroup": {
 "Status": "available",
 "Description": "Some description",
 "NodeGroups": [{
 "Status": "available",
 "NodeGroupMembers": [{
 "CurrentRole": "primary",
 "PreferredAvailabilityZone": "us-west-2b",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "my-repl-group-001"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2c",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "my-repl-group-002"
 }
],
 "NodeGroupId": "0001",
```

```
"PrimaryEndpoint": {
 "Port": 6379,
 "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
},
}],
"ReplicationGroupId": "my-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "disabled",
"SnapshotWindow": "12:00-13:00",
"SnapshottingClusterId": "my-repl-group-002",
"MemberClusters": [
 "my-repl-group-001",
 "my-repl-group-002"
],
"PendingModifiedValues": {}
}
}
```

如需詳細資訊，請參閱 [modify-replication-group](#) 在 AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately` 參數，請使用具有下列參數的 `describe-replication-group` 命令來 AWS CLI 監控複寫群組的狀態。當狀態仍在修改時，您可能會看到在 5.0.6 版及更高版本上執行的節點有不到 1 秒的停機時間，以及在 DNS 項目更新時，從主要節點短暫中斷讀取和寫入較舊的版本。
  - `--replication-group-id` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-replication-groups \
 --replication-group-id my-replication-group
```

針對 Windows：

```
aws elasticache describe-replication-groups ^
 --replication-group-id my-replication-group
```

如需詳細資訊，請參閱 參考 [describe-replication-groups](#) 中的 。 AWS CLI

## 擴展 Valkey 或 Redis OSS 複寫群組 (ElastiCache API)

下列程序會使用 將您的複寫群組從目前的節點類型擴展到新的較大節點類型 ElastiCache API。對於 Valkey 7.2 及更高版本或 Redis OSS 5.0.5 及更高版本，您可以在叢集繼續保持連線並提供傳入請求時擴展啟用自動容錯移轉的叢集。在 Redis OSS 4.0.10 及更低版本上，您可能會注意到 DNS 項目更新時，從主要節點讀取和寫入先前版本的短暫中斷。

向上擴展至較大的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

## 擴展 Valkey 或 Redis OSS 複寫群組 (ElastiCache API)

1. 使用 `ListAllowedNodeTypeModifications` 動作搭配下列參數，ElastiCache API 決定您可以擴展到哪些節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考中。

2. 使用 `ModifyRedplicationGroup` ElastiCache API 動作和下列參數，將您目前的複寫群組擴展到新的節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。
  - `CacheNodeType` - 此複寫群組中較大的新快取叢集節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypeModifications` 動作傳回的執行個體類型之一。
  - `CacheParameterGroupName` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。

- `ApplyImmediately` - 設定為 `true` 會立即套用擴充規模程序。若要將向上擴展程序延期至下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&CacheNodeType=cache.m3.2xlarge
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyReplicationGroup](#) 在 Amazon ElastiCache API 參考 中。

3. 如果您使用 `ApplyImmediately=true`，請使用具有下列參數 `DescribeReplicationGroups` 的動作來 ElastiCache API 監控複寫群組的狀態。當狀態從 `modifying` (修改中) 變更為 `available` (可用)，您即可以開始寫入新的、已向上擴展的複寫群組。
  - `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeReplicationGroups](#) 在 Amazon ElastiCache API 參考 中。





## 使用複本縮減 Valkey 或 Redis OSS 叢集

下列各節將逐步說明如何使用複本節點將 Valkey 或 Redis OSS ( 停用叢集模式 ) 快取叢集縮減為較小的節點類型。請務必確保新的，較小的節點類型足夠大而可以容納所有資料和成本。如需詳細資訊，請參閱[確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)。

### Note

針對執行 r6gd 節點類型的叢集，您只能擴展到 r6gd 節點系列中的節點大小。

### Important

如果您的參數群組使用 reserved-memory 來為 Valkey 或 Redis OSS 額外負荷預留記憶體，在開始擴展之前，請確定您有一個自訂參數群組，可為您的新節點類型保留正確的記憶體數量。或者，您可以修改自訂參數群組，使得它會對您的新叢集使用 reserved-memory-percent 並使用該參數群組。

如果您使用 reserved-memory-percent，則不需執行此動作。

如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

## 使用複本縮減 Valkey 或 Redis OSS 叢集規模 ( 主控台 )

下列程序會使用 ElastiCache 主控台，使用複本節點將 Valkey 或 Redis OSS 叢集擴展至較小的節點類型。

### 使用複本節點縮減 Valkey 或 Redis OSS 叢集規模 ( 主控台 )

1. 確保該較小的節點類型對您的資料和成本而言已足夠。
2. 如果您的參數群組使用 reserved-memory 來為 Valkey 或 Redis OSS 額外負荷預留記憶體，請確定您有自訂參數群組來為新節點類型預留正確的記憶體數量。

或者，您可以將自訂參數群組修改為使用 reserved-memory-percent。如需詳細資訊，請參閱[管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

3. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
4. 從叢集的清單，選擇您要縮減規模的叢集。此叢集必須執行 Valkey 或 Redis OSS 引擎，而非叢集 Valkey 或 Redis OSS 引擎。

5. 選擇 Modify (修改)。
6. 在 Modify Cluster (修改叢集) 精靈中：
  - a. 從 Node type (節點類型) 清單，選擇您要縮減規模的節點類型。
  - b. 如果您使用 reserved-memory 來管理記憶體，請從 Parameter Group (參數群組) 清單中，選擇會為新節點類型保留正確記憶體數量的自訂參數群組。
7. 如果您要立即執行縮減規模程序，請選擇 Apply immediately (立即套用) 核取方塊。如果將 Apply immediately (立即套用) 核取方塊保留未選擇，在此叢集的下一個維護時段期間，會執行縮減規模程序。
8. 選擇 Modify (修改)。
9. 當叢集的狀態從 modifying (修改中) 變更為 available (可用)，您的叢集即已擴展為新節點類型。不需要在應用程式中更新端點。

### 縮減 Valkey 或 Redis OSS 複寫群組 ( AWS CLI )

下列程序會使用 AWS CLI，將複寫群組從其目前的節點類型擴展至新的、較小的節點類型。在此過程中，會 ElastiCache 更新 DNS 項目，以便它們指向新的節點。因此，您不需在應用程式中更新端點。對於上述 Valkey 7.2 或 Redis OSS 5.0.5 及更高版本，您可以在叢集繼續保持連線並提供傳入請求時擴展啟用自動容錯移轉的叢集。在 4.0.10 版及更新版本中，您可能會注意到 DNS 項目更新時，從主要節點讀取和寫入先前版本的短暫中斷。

不過，來自僅供讀取複本快取叢集的讀取會繼續不中斷。

向下擴展至較小的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

### 縮減 Valkey 或 Redis OSS 複寫群組 ( AWS CLI )

1. 使用下列參數執行 list-allowed-node-type-modifications 命令，AWS CLI 決定您可以縮減規模的節點類型。
  - `--replication-group-id` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-repl-group
```

針對 Windows :

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-repl-group
```

此操作的輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ScaleDownModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
]
}
```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在AWS CLI 參考 中。

2. 使用具有下列參數的 AWS CLI `modify-replication-group` 命令，將您目前的複寫群組擴展到新的節點類型。
  - `--replication-group-id` - 複寫群組的名稱。
  - `--cache-node-type` - 此複寫群組中較小的新快取叢集節點類型。此值必須是上一個步驟中 `list-allowed-node-type-modifications` 命令傳回的執行個體類型之一。
  - `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `--apply-immediately` - 會造成立即套用擴充規模程序。若要將向上擴展操作延期至下一個維護時段，請使用 `--no-apply-immediately`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id my-repl-group \
 --cache-node-type cache.t2.small \
 --cache-parameter-group-name redis32-m3-2x1 \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id my-repl-group ^
 --cache-node-type cache.t2.small ^
 --cache-parameter-group-name redis32-m3-2x1 \
 --apply-immediately
```

此命令的輸出看起來像這樣（JSON 格式）。

```
{"ReplicationGroup": {
 "Status": "available",
 "Description": "Some description",
 "NodeGroups": [
 {
 "Status": "available",
 "NodeGroupMembers": [
 {
 "CurrentRole": "primary",
 "PreferredAvailabilityZone": "us-west-2b",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "my-repl-group-001"
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2c",
```

```
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
 },
 "CacheClusterId": "my-repl-group-002"
 }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
 "Port": 6379,
 "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
}
}
],
"ReplicationGroupId": "my-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "disabled",
"SnapshotWindow": "12:00-13:00",
"SnapshottingClusterId": "my-repl-group-002",
"MemberClusters": [
 "my-repl-group-001",
 "my-repl-group-002",
],
"PendingModifiedValues": {}
}
}
```

如需詳細資訊，請參閱 [modify-replication-group](#) 在 AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately` 參數，請使用具有下列參數的 `describe-replication-group` 命令來 AWS CLI 監控複寫群組的狀態。當狀態從 `modifying` (修改中) 變更為 `available` (可用)，您即可以開始寫入新的、已向下擴展的複寫群組。
  - `--replication-group-id` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-replication-group \
```

```
--replication-group-id my-replication-group
```

針對 Windows :

```
aws elasticache describe-replication-groups ^
 --replication-group-id my-replication-group
```

如需詳細資訊，請參閱 參考 [describe-replication-groups](#) 中的 。 AWS CLI

## 縮減 Valkey 或 Redis OSS 複寫群組 (ElastiCache API)

下列程序會使用 將複寫群組從目前的節點類型擴展到新的較小節點類型 ElastiCache API。在此過程中，會 ElastiCache 更新DNS項目，以便它們指向新的節點。因此，您不需在應用程式中更新端點。對於 Valkey 7.2 及更高版本或 Redis OSS 5.0.5 及更高版本，您可以在叢集繼續保持連線並提供傳入請求時擴展自動容錯移轉啟用的叢集。在 Redis 4OSS.0.10 版及更新版本中，您可能會注意到DNS項目更新時，先前節點的讀取和寫入短暫中斷。不過，來自僅供讀取複本快取叢集的讀取會繼續不中斷。

向下擴展至較小的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

## 縮減 Valkey 或 Redis OSS 複寫群組 (ElastiCache API)

1. 使用 ListAllowedNodeTypeModifications 動作搭配下列參數，ElastiCache API 決定您可以縮減為哪些節點類型。
  - ReplicationGroupId - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考中。

2. 使用 `ModifyReplicationGroup` ElastiCache API 動作和下列參數，將您目前的複寫群組擴展到新的節點類型。

- `ReplicationGroupId` - 複寫群組的名稱。
- `CacheNodeType` - 此複寫群組中較小的新快取叢集節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypesModifications` 動作傳回的執行個體類型之一。
- `CacheParameterGroupName` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
- `ApplyImmediately` - 設定為 `true` 會立即套用擴充規模程序。若要將向下擴展程序延期至下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&CacheNodeType=cache.m3.2xlarge
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyReplicationGroup](#) 在 Amazon ElastiCache API 參考 中。

3. 如果您使用 `ApplyImmediately=true`，請使用具有下列參數 `DescribeReplicationGroups` 的動作來 ElastiCache API 監控複寫群組的狀態。當狀態從 `modifying` (修改中) 變更為 `available` (可用)，您即可以開始寫入新的、已向下擴展的複寫群組。

- `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeReplicationGroups](#) 在 Amazon ElastiCache API 參考 中。



## 增加讀取容量

若要增加讀取容量，請將僅供讀取複本（最多五個）新增至您的 Valkey 或 Redis OSS 複寫群組。

您可以使用 ElastiCache 主控台、或來擴展 Valkey AWS CLI 或 Redis OSS 叢集的讀取容量 ElastiCache API。如需詳細資訊，請參閱 [新增 Valkey 或 Redis 的僅供讀取複本 OSS \(停用叢集模式\)](#)。

## 減少讀取容量

若要降低讀取容量，請從具有複本的 Valkey 或 Redis OSS 叢集中刪除一或多個僅供讀取複本（在 API/ 中稱為複寫群組 CLI）。如果叢集為已啟用自動容錯移轉的多個可用區，若未先停用多個可用區，則無法刪除最後一個僅供讀取複本。如需詳細資訊，請參閱[修改複寫群組](#)。

如需詳細資訊，請參閱[刪除 Valkey 或 Redis 的僅供讀取複本 OSS（停用叢集模式）](#)。

## 在 Valkey 或 Redis 中擴展叢集 OSS ( 已啟用叢集模式 )

隨著叢集的需求變更，您可以變更 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片數量，以決定改善效能或降低成本。我們建議使用線上水平擴展來執行此動作，因為它可允許叢集在擴展程序期間繼續提供請求的服務。

您用來決定重新擴展叢集的可能條件包括下列：

- 記憶體壓力：

如果叢集中的節點遭受記憶體壓力，您可以決定向外擴展，使得您有更多資源能更妥善地存放資料和提供請求的服務。

您可以監控下列指標來判斷節點是否受到記憶體壓力：FreeableMemory、SwapUsage和 BytesUseForCache。

- CPU 或網路瓶頸：

如果延遲/傳輸量問題正困擾著您的叢集，您可能需要向外擴展來解決問題。

您可以透過監控下列指標來監控延遲和輸送量層

級：CPUUtilization、NetworkBytesInNetworkBytesOut、CurrConnections和 NewConnections。

- 您的叢集過度擴展：

對叢集的目前需求使得向內擴展不會傷害效能和減少成本。

您可以使用下列指標來監控叢集的使用，以判斷您是否可以安全地擴

展：FreeableMemory、SwapUsage、BytesUseForCache、CPUUtilization、NetworkBytesIn、NetworkBytesOut CurrConnections和 NewConnections。

### 擴展的效能影響

使用離線程序擴展時，您的叢集將有一大部分程序會離線，因此無法提供請求的服務。使用線上方法擴展時，因為擴展是運算密集的操作，效能會有一些下降，然後，您的叢集會繼續在整個擴展操作中提供請求的服務。您的降級程度取決於正常CPU使用率和資料。

有兩種方法可以擴展您的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集；水平和垂直擴展。

- 水平擴展可讓您透過新增或移除節點群組 (碎片)，來變更複寫群組中節點群組 (碎片) 的數量。線上重新分片程序允許向內/向外擴展，同時間叢集仍可繼續服務傳入請求。

在您的新叢集中以與在舊叢集中不同的方式設定槽。僅限離線方法。

- 垂直擴展 - 變更節點類型以調整叢集大小。線上垂直擴展允許向上/向下擴展，同時間叢集仍可繼續服務傳入請求。

如果您要透過縮減或縮減叢集的大小和記憶體容量，請確保新組態有足夠的記憶體來容納資料和 Valkey 或 Redis OSS 額外負荷。

如需詳細資訊，請參閱[選取快取節點大小](#)。

## 內容

- [Valkey 或 Redis 的離線重新共享 OSS \( 啟用叢集模式 \)](#)
- [Valkey 或 Redis 的線上重新共享 OSS \( 啟用叢集模式 \)](#)
  - [使用線上重新分片功能新增碎片](#)
  - [使用線上重新分片移除碎片](#)
    - [移除碎片 \(主控台\)](#)
    - [移除碎片 \(AWS CLI\)](#)
    - [移除碎片 \(ElastiCache API\)](#)
  - [線上碎片重新平衡](#)
    - [線上碎片重新平衡 \(主控台\)](#)
    - [線上碎片重新平衡 \(AWS CLI\)](#)
    - [線上碎片重新平衡 \(ElastiCache API\)](#)
- [透過修改節點類型來進行線上垂直擴展](#)
  - [線上擴充規模](#)
    - [擴展 Valkey 或 Redis OSS 快取叢集 \( 主控台 \)](#)
    - [擴展 Valkey 或 Redis OSS 快取叢集 \( AWS CLI \)](#)
    - [擴展 Valkey 或 Redis OSS 快取叢集 \( ElastiCache API \)](#)
  - [線上縮減規模](#)
    - [縮減 Valkey 或 Redis OSS 快取叢集 \( 主控台 \)](#)
    - [縮減 Valkey 或 Redis OSS 快取叢集 \( AWS CLI \)](#)
    - [縮減 Valkey 或 Redis OSS 快取叢集 \( ElastiCache API \)](#)

## Valkey 或 Redis 的離線重新共享 OSS ( 啟用叢集模式 )

您從離線碎片重新組態可獲得的主要優點是，您可以做得更多而不只是從複寫群組新增或移除碎片。當您在離線時重新碎片和重新平衡時，除了變更複寫群組中的碎片數量之外，您還可以執行下列動作：

### Note

啟用資料分層的 Valkey 或 Redis OSS 叢集不支援離線重新編譯。如需詳細資訊，請參閱 [中的資料分層 ElastiCache](#)。

- 變更您複寫群組的節點類型。
- 指定複寫群組中每個節點的可用區域。
- 升級至較新的引擎版本。
- 獨立指定每個碎片中複本節點的數量。
- 指定每個碎片的金鑰空間。

離線碎片重新組態的主要缺點是，您的叢集會離線開始進程序的還原部分，並繼續直到您在應用程式中更新端點為止。叢集離線的時間長度會因叢集中資料量而不同。

若要離線重新設定碎片 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集

1. 建立現有 Valkey 或 Redis OSS 叢集的手動備份。如需詳細資訊，請參閱 [取得手動備份](#)。
2. 透過從備份還原來建立新叢集。如需詳細資訊，請參閱 [從備份還原到新的快取](#)。
3. 在應用程式中，將端點更新為新叢集的二端點。如需詳細資訊，請參閱 [在中尋找連線端點 ElastiCache](#)。

## Valkey 或 Redis 的線上重新共享 OSS ( 啟用叢集模式 )

透過使用線上重新分割和碎片重新平衡 ElastiCache Valkey 7.2 或更新版本，或 Redis 3.2.10 OSS 或更新版本，您可以動態擴展您的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集，而不需要停機。此方法表示叢集可以繼續提供請求的服務 (甚至是在擴展或重新平衡進行中時)。

您可以執行下列作業：

- 橫向擴展 – 將碎片 ( 節點群組 ) 新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ( 複寫群組 )，以增加讀取和寫入容量。

如果您將一或多個碎片新增至複寫群組，每個新碎片中的節點數量與現有碎片最小節點的數量相同。

- 縮減規模 – 將碎片從 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中移除，以降低讀取和寫入容量，進而降低成本。
- 重新平衡 – 在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片之間移動鍵空間，以便盡可能在碎片之間平均分佈。

您無法執行下列：

- 獨立設定碎片：

您不可以獨立為碎片指定金鑰空間。若要執行此動作，您必須使用離線程序。

目前，下列限制適用於 ElastiCache 線上重新共用和重新平衡：

- 這些程序需要 Valkey 7.2 及更新版本或 Redis OSS引擎 3.2.10 或更新版本。如需升級引擎版本的詳細資訊，請參閱[的版本管理 ElastiCache](#)。
- 槽或金鑰空間和大型項目的限制為：

如果碎片中的任何金鑰包含大型項目，在向外擴展或重新平衡時，不會將該金鑰遷移至新碎片。此功能可能造成不平衡的碎片。

如果碎片中的任何金鑰包含大型項目 (序列化後項目大於 256 MB)，在向內擴展時，不會刪除該碎片。此功能可能造成一些碎片不會遭到刪除。

- 向外擴展時，任何新碎片中節點的數量會等於現有碎片中最小節點的數量。
- 向外擴展時，所有現有碎片共有的任何標籤都會複製到新碎片。
- 擴展全域資料存放區叢集時，ElastiCache 不會自動將函數從其中一個現有節點複寫至新的節點 ( s )。我們建議在橫向擴展叢集之後，將函數載入新的碎片，如此一來每個碎片都會具有相同函數。

#### Note

在中 ElastiCache，使用 Valkey 7.2 和更新版本，以及使用 Redis 7 和更新OSS版本：在擴展叢集時，ElastiCache 會自動將載入其中一個現有節點 ( 隨機選取 ) 的函數複寫至新的節點。如果您的應用程式使用[函數](#)，建議您先將所有函數載入所有碎片，然後再向外擴展，以免叢集最終在不同的碎片上出現不同的函數定義。

如需詳細資訊，請參閱[線上叢集大小調整](#)。

您可以使用 AWS Management Console、AWS CLI 和 水平擴展或重新平衡您的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 ElastiCache API。

### 使用線上重新分片功能新增碎片

您可以使用 AWS Management Console AWS CLI、ElastiCache 或 將碎片新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 API。當您將碎片新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集時，現有碎片上的任何標籤都會複製到新的碎片。

### 新增碎片 (主控台)

您可以使用 AWS Management Console 將一或多個碎片新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集。下列程序描述該程序。

將碎片新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集

1. 在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集。
3. 找到並選擇您要新增碎片的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的名稱，而不是叢集名稱左側的方塊。

#### Tip

Valkey 或 Redis OSS ( 啟用叢集模式 ) 在模式欄中顯示叢集 Valkey 或叢集 Redis OSS

4. 選擇 Add shard (新增碎片)。
  - a. 針對 Number of shards to be added (要新增的碎片數)，選擇您要新增至此叢集的碎片數量。
  - b. 針對 Availability zone(s) (可用區域)，選擇 No preference (無偏好設定) 或 Specify availability zones (指定可用區域) 中的一個。
  - c. 如果您選擇 Specify availability zones (指定可用區域)，針對每個碎片中的每個節點，從可用區域的清單選取節點的可用區域。
  - d. 選擇新增。

## 新增碎片 (AWS CLI)

下列程序說明如何使用 新增碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 AWS CLI。

使用下列參數搭配 `modify-replication-group-shard-configuration`。

### 參數

- `--apply-immediately` - 必要項目。指定碎片重新組態操作要立即開始。
- `--replication-group-id` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `--node-group-count` - 必要項目。指定操作完成時要存在的碎片數量 (節點群組)。新增碎片時，`--node-group-count` 的值必須大於目前的碎片數量。

您可以選擇性地使用 `--resharding-configuration` 為複寫群組中的每個節點指定可用區域。

- `--resharding-configuration` - 選用。複寫群組的每個碎片中每個節點偏好的可用區域清單。只有在 `--node-group-count` 的值大於目前的碎片數量時，才使用此參數。如果在新增碎片時省略此參數，Amazon ElastiCache 會為新節點選取可用區域。

下列範例會重新設定名稱為 `my-cluster` 的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中四個碎片上的鍵空間。此範例也會指定每個碎片中每個節點的可用區域。該操作會立即開始。

### Example - 新增碎片

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group-shard-configuration \
 --replication-group-id my-cluster \
 --node-group-count 4 \
 --resharding-configuration \
 "PreferredAvailabilityZones=us-east-2a,us-east-2c" \
 "PreferredAvailabilityZones=us-east-2b,us-east-2a" \
 "PreferredAvailabilityZones=us-east-2c,us-east-2d" \
 "PreferredAvailabilityZones=us-east-2d,us-east-2c" \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group-shard-configuration ^
```



```
--replication-group-id my-cluster ^
--node-group-count 4 ^
--resharding-configuration ^
 "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^
 "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^
 "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^
 "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^
--apply-immediately
```

如需詳細資訊，請參閱 AWS CLI 文件中的 [modify-replication-group-shard-configuration](#)。

## 新增碎片 (ElastiCache API)

您可以使用 ElastiCache API 來重新設定 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片，方法是使用 `ModifyReplicationGroupShardConfiguration` 操作。

使用下列參數搭配 `ModifyReplicationGroupShardConfiguration`。

### 參數

- `ApplyImmediately=true` - 必要項目。指定碎片重新組態操作要立即開始。
- `ReplicationGroupId` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `NodeGroupCount` - 必要項目。指定操作完成時要存在的碎片數量 (節點群組)。新增碎片時，`NodeGroupCount` 的值必須大於目前的碎片數量。

您可以選擇性地使用 `ReshardingConfiguration` 為複寫群組中的每個節點指定可用區域。

- `ReshardingConfiguration` - 選用。複寫群組的每個碎片中每個節點偏好的可用區域清單。只有在 `NodeGroupCount` 的值大於目前的碎片數量時，才使用此參數。如果在新增碎片時省略此參數，Amazon 會為新節點 ElastiCache 選取可用區域。

下列程序說明如何使用 新增碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 ElastiCache API。

### Example - 新增碎片

下列範例會將節點群組新增至 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集 `my-cluster`，因此操作完成時總共有四個節點群組。此範例也會指定每個碎片中每個節點的可用區域。該操作會立即開始。

```
https://elasticache.us-east-2.amazonaws.com/
 ?Action=ModifyReplicationGroupShardConfiguration
```

```
&ApplyImmediately=true
&NodeGroupCount=4
&ReplicationGroupId=my-cluster

&ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2a

&ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2c

&ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2b

&ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2a

&ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2c

&ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2d

&ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2d

&ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2c

&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20171002T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 參考 [ModifyReplicationGroupShardConfiguration](#) 中的 ElastiCache API。

## 使用線上重新分片移除碎片

您可以使用 AWS Management Console AWS CLI、或 [AWS CLI](#)，從 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集移除碎片ElastiCache API。

## 主題

- [移除碎片 \(主控台\)](#)
- [移除碎片 \(AWS CLI\)](#)

- [移除碎片 \(ElastiCache API\)](#)

### 移除碎片 (主控台)

下列程序說明如何使用 移除碎片，重新設定 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片 AWS Management Console。

從複寫群組中移除節點群組 ( 碎片 ) 之前，ElastiCache 請確定所有資料都符合其餘碎片。如果資料可納入，則會按請求從複寫群組刪除指定的碎片。如果資料無法納入其他的節點群組，流程會中止，複寫群組的節點群組設定會與請求之前相同。

您可以使用 AWS Management Console 從您的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中移除一或多個碎片。您無法移除複寫群組中的所有碎片。而是必須刪除複寫群組。如需詳細資訊，請參閱[刪除複寫群組](#)。下列程序描述用於刪除一或多個碎片的程序。

從 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中移除碎片

1. 在 開啟 ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。
3. 找到並選擇要從中移除碎片的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的名稱，而不是叢集名稱左側的方塊。


 Tip

在 Shards 欄中，Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的值為 1 或更高。

4. 從碎片的清單，選擇您要刪除的每個碎片名稱左側的方塊。
5. 選擇 Delete shard (刪除碎片)。

### 移除碎片 (AWS CLI)

下列程序說明如何使用 移除碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 AWS CLI。

 Important

從複寫群組中移除節點群組 ( 碎片 ) 之前，ElastiCache 請確定所有資料都符合其餘碎片。如果資料可納入，則會按請求從複寫群組刪除指定的碎片 (`--node-groups-to-remove`)，而

其金鑰空間會映射至其餘的碎片。如果資料無法納入其他的節點群組，流程會中止，複寫群組的節點群組設定會與請求之前相同。

您可以使用 AWS CLI 從您的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中移除一或多個碎片。您無法移除複寫群組中的所有碎片。而是必須刪除複寫群組。如需詳細資訊，請參閱[刪除複寫群組](#)。

使用下列參數搭配 `modify-replication-group-shard-configuration`。

### 參數

- `--apply-immediately` - 必要項目。指定碎片重新組態操作要立即開始。
- `--replication-group-id` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `--node-group-count` - 必要項目。指定操作完成時要存在的碎片數量 (節點群組)。移除碎片時，`--node-group-count` 的值必須小於目前的碎片數量。
- `--node-groups-to-remove` - 當 `--node-group-count` 小於目前的數量節點群組 (碎片) 時為必要項目。IDs 要從複寫群組中移除的碎片 (節點群組) 清單。

下列程序描述用於刪除一或多個碎片的程序。

### Example - 移除碎片

下列範例會從 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中移除兩個節點群組 `my-cluster`，因此操作完成時總共有兩個節點群組。來自移除碎片的金鑰空間會平均分散在其餘碎片間。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group-shard-configuration \
 --replication-group-id my-cluster \
 --node-group-count 2 \
 --node-groups-to-remove "0002" "0003" \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group-shard-configuration ^\
 --replication-group-id my-cluster ^\
 --node-group-count 2 ^\
 --node-groups-to-remove "0002" "0003" ^
```

```
--apply-immediately
```

## 移除碎片 (ElastiCache API)

您可以使用 ElastiCache API 來重新設定 Valkey 或 Redis OSS (啟用叢集模式) 叢集中的碎片，方法是使用 `ModifyReplicationGroupShardConfiguration` 操作。

下列程序說明如何使用 移除碎片，在 Valkey 或 Redis OSS (啟用叢集模式) 叢集中重新設定碎片 ElastiCache API。

### Important

從複寫群組中移除節點群組 (碎片) 之前，ElastiCache 請確定所有資料都符合其餘碎片。如果資料可納入，則會按請求從複寫群組刪除指定的碎片 (`NodeGroupsToRemove`)，而其金鑰空間會映射至其餘的碎片。如果資料無法納入其他的節點群組，流程會中止，複寫群組的節點群組設定會與請求之前相同。

您可以使用 ElastiCache API 從您的 Valkey 或 Redis OSS (啟用叢集模式) 叢集中移除一或多個碎片。您無法移除複寫群組中的所有碎片。而是必須刪除複寫群組。如需詳細資訊，請參閱 [刪除複寫群組](#)。

使用下列參數搭配 `ModifyReplicationGroupShardConfiguration`。

### 參數

- `ApplyImmediately=true` - 必要項目。指定碎片重新組態操作要立即開始。
- `ReplicationGroupId` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `NodeGroupCount` - 必要項目。指定操作完成時要存在的碎片數量 (節點群組)。移除碎片時，`NodeGroupCount` 的值必須小於目前的碎片數量。
- `NodeGroupsToRemove` - 當 `--node-group-count` 小於目前的數量節點群組 (碎片) 時為必要項目。IDs 要從複寫群組中移除的碎片 (節點群組) 清單。

下列程序描述用於刪除一或多個碎片的程序。

### Example - 移除碎片

下列範例會從 Valkey 或 Redis OSS (啟用叢集模式) 叢集中移除兩個節點群組 `my-cluster`，因此操作完成時總共有兩個節點群組。來自移除碎片的金鑰空間會平均分散在其餘碎片間。

```
https://elasticache.us-east-2.amazonaws.com/
?Action=ModifyReplicationGroupShardConfiguration
&ApplyImmediately=true
&NodeGroupCount=2
&ReplicationGroupId=my-cluster
&NodeGroupsToRemove.member.1=0002
&NodeGroupsToRemove.member.2=0003
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20171002T192317Z
&X-Amz-Credential=<credential>
```

## 線上碎片重新平衡

您可以使用 AWS Management Console AWS CLI、或重新平衡 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片ElastiCache API。

### 主題

- [線上碎片重新平衡 \(主控台\)](#)
- [線上碎片重新平衡 \(AWS CLI\)](#)
- [線上碎片重新平衡 \(ElastiCache API\)](#)

## 線上碎片重新平衡 (主控台)

下列程序說明如何使用重新平衡碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 AWS Management Console。

### 重新平衡 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集上碎片之間的鍵空間

1. 在開啟 ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。
3. 選擇您要重新平衡的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的名稱，而不是名稱左側的方塊。

#### Tip

在 Shards 欄中，Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集的值為 1 或更高。

4. 選擇 Rebalance (重新平衡)。
5. 提示時，選擇 Rebalance (重新平衡)。您可能會看到類似這樣的訊息：*Slots in the replication group are uniformly distributed. Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*。如果您這樣做，請選擇取消。

### 線上碎片重新平衡 (AWS CLI)

使用下列參數搭配 `modify-replication-group-shard-configuration`。

#### 參數

- `-apply-immediately` - 必要項目。指定碎片重新組態操作要立即開始。
- `--replication-group-id` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `--node-group-count` - 必要項目。若要重新平衡叢集中所有碎片間的金鑰空間，此值必須與目前的碎片數量相同。

下列程序說明如何使用 重新平衡碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 AWS CLI。

#### Example - 重新平衡叢集中的碎片

下列範例會重新平衡 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的插槽，`my-cluster` 以便盡可能平均地分配插槽。`--node-group-count (4)` 的值為目前在叢集中碎片的數量。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group-shard-configuration \
 --replication-group-id my-cluster \
 --node-group-count 4 \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group-shard-configuration ^
 --replication-group-id my-cluster ^
 --node-group-count 4 ^
```

```
--apply-immediately
```

## 線上碎片重新平衡 (ElastiCache API)

您可以使用 ElastiCache API 來重新設定 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的碎片，方法是使用 `ModifyReplicationGroupShardConfiguration` 操作。

使用下列參數搭配 `ModifyReplicationGroupShardConfiguration`。

### 參數

- `ApplyImmediately=true` - 必要項目。指定碎片重新組態操作要立即開始。
- `ReplicationGroupId` - 必要項目。指定碎片重新組態操作要執行所在的複寫群組 (叢集)。
- `NodeGroupCount` - 必要項目。若要重新平衡叢集中所有碎片間的金鑰空間，此值必須與目前的碎片數量相同。

下列程序說明如何使用 重新平衡碎片，在 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中重新設定碎片 ElastiCache API。

### Example - 重新平衡叢集

下列範例會重新平衡 Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集中的插槽，`my-cluster` 以便盡可能平均地分配插槽。`NodeGroupCount (4)` 的值為目前在叢集中碎片的數量。

```
https://elasticache.us-east-2.amazonaws.com/
 ?Action=ModifyReplicationGroupShardConfiguration
 &ApplyImmediately=true
 &NodeGroupCount=4
 &ReplicationGroupId=my-cluster
 &Version=2015-02-02
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20171002T192317Z
 &X-Amz-Credential=<credential>
```

### 透過修改節點類型來進行線上垂直擴展

透過搭配 Valkey 7.2 版或更新版本，或 Redis 3.2.10 OSS 版或更新版本使用線上垂直擴展，您可以在最短的停機時間下動態擴展您的 Valkey 或 Redis OSS 叢集。這可讓 Valkey 或 Redis OSS 叢集在擴展時提供請求。



**Note**

不支援在資料分層叢集 (例如, 使用 r6gd 節點類型的叢集) 與未使用資料分層叢集 (例如, 使用 r6g 節點類型的叢集) 之間的擴展。如需詳細資訊, 請參閱 [中的資料分層 ElastiCache](#)。

您可以執行下列作業：

- 擴展 – 透過調整 Valkey 或 Redis OSS叢集的節點類型以使用較大的節點類型來增加讀取和寫入容量。

ElastiCache 動態調整叢集的大小, 同時保持連線和服務請求。

- 縮減規模 - 將節點類型向下調整為使用較小的節點, 減少讀取和寫入容量。再次強調, ElastiCache 動態調整叢集的大小, 同時保持線上狀態並提供請求。在這種情況下, 您透過縮減節點來降低成本。

**Note**

向上擴展和向下擴展程序牽涉到使用新選取的節點類型來建立叢集, 並將新節點與先前的節點進行同步。若要確保順暢的向上/向下擴展流程, 請執行以下操作：

- 確保您有足夠的 ENI (彈性網路介面) 容量。若是向下擴展, 請確定較小的節點有足夠的記憶體, 可以吸收預期的流量。

如需記憶體管理的最佳實務, 請參閱 [管理 Valkey 和 Redis 的預留記憶體 OSS](#)。

- 雖然垂直擴展程序的目標是保持全面上線, 但此程序也需要在舊節點和新節點之間同步資料。建議您在預期資料流量最小的時間內啟動向上/向下擴展。
- 盡可能在預備環境中測試您的應用程式行為。

**內容**

- [線上擴充規模](#)
  - [擴展 Valkey 或 Redis OSS快取叢集 \(主控台\)](#)
  - [擴展 Valkey 或 Redis OSS快取叢集 \(AWS CLI\)](#)
  - [擴展 Valkey 或 Redis OSS快取叢集 \(ElastiCache API\)](#)
- [線上縮減規模](#)
  - [縮減 Valkey 或 Redis OSS快取叢集 \(主控台\)](#)

- [縮減 Valkey 或 Redis OSS 快取叢集 \( AWS CLI \)](#)
- [縮減 Valkey 或 Redis OSS 快取叢集 \( ElastiCache API \)](#)

## 線上擴充規模

### 主題

- [擴展 Valkey 或 Redis OSS 快取叢集 \( 主控台 \)](#)
- [擴展 Valkey 或 Redis OSS 快取叢集 \( AWS CLI \)](#)
- [擴展 Valkey 或 Redis OSS 快取叢集 \( ElastiCache API \)](#)

### 擴展 Valkey 或 Redis OSS 快取叢集 ( 主控台 )

下列程序說明如何使用 ElastiCache 管理主控台擴展 Valkey 或 Redis OSS 叢集。在此過程中，您的叢集將繼續以最短的停機時間提供請求。

### 擴展 Valkey 或 Redis OSS 叢集 ( 主控台 )

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集。
3. 從叢集的清單中，選擇叢集。
4. 選擇 Modify (修改)。
5. 在 Modify Cluster (修改叢集) 精靈中：
  - 從 Node type (節點類型) 清單選擇您要擴展的節點類型。若要向上擴展，請選取大於現有節點的節點類型。
6. 如果您要立即執行向上擴展程序，請選擇 Apply immediately (立即套用) 方塊。如果未選擇 Apply immediately (立即套用) 方塊，在此叢集的下一個維護時段期間，會執行向上擴展程序。
7. 選擇 Modify (修改)。

如果您在上一步驟中選擇 Apply immediately (立即套用)，叢集的狀態變更為 modifying (修改中)。當狀態變更為 available (可用)，修改即已完成，並且您可以開始使用新叢集。

## 擴展 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

下列程序說明如何使用 擴展 Valkey 或 Redis OSS 快取叢集 AWS CLI。在此過程中，您的叢集將繼續以最短的停機時間提供請求。

## 擴展 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

1. 使用下列參數執行 `list-allowed-node-type-modifications` 命令，決定您可以擴展至的 AWS CLI 節點類型。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-replication-group-id
```

針對 Windows：

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-replication-group-id
```

從上述命令輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ScaleUpModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
]
 "ScaleDownModifications": [
 "cache.t2.micro",
 "cache.t2.small",
 "cache.t2.medium",
]
}
```

```
 "cache.t1.small "
],
}
```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在AWS CLI 參考 中。

2. 使用 AWS CLI `modify-replication-group` 命令和下列參數，修改複寫群組以擴展至新的較大節點類型。
  - `--replication-group-id` - 您要擴充規模的複寫群組名稱。
  - `--cache-node-type` - 您要擴展快取叢集的新節點類型。此值必須是步驟 1 中 `list-allowed-node-type-modifications` 命令傳回的其中一個節點類型。
  - `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `--apply-immediately` - 會造成立即套用擴充規模程序。若要將向上擴展程序延期至叢集的下一個維護時段，請使用 `--no-apply-immediately` 參數。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-cluster \
 --cache-node-type cache.m3.xlarge \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id my-redis-cluster ^
 --cache-node-type cache.m3.xlarge ^
 --apply-immediately
```

從上述命令輸出看起來像這樣（JSON 格式）。

```
{
 "ReplicationGroup": {
 "Status": "modifying",
 "Description": "my-redis-cluster",
```

```
"NodeGroups": [
 {
 "Status": "modifying",
 "Slots": "0-16383",
 "NodeGroupId": "0001",
 "NodeGroupMembers": [
 {
 "PreferredAvailabilityZone": "us-east-1f",
 "CacheNodeId": "0001",
 "CacheClusterId": "my-redis-cluster-0001-001"
 },
 {
 "PreferredAvailabilityZone": "us-east-1d",
 "CacheNodeId": "0001",
 "CacheClusterId": "my-redis-cluster-0001-002"
 }
]
 }
],
"ConfigurationEndpoint": {
 "Port": 6379,
 "Address": "my-redis-
cluster.r7gdfi.clustercfg.us1.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "my-redis-cluster",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "07:30-08:30",
"MemberClusters": [
 "my-redis-cluster-0001-001",
 "my-redis-cluster-0001-002"
],
"CacheNodeType": "cache.m3.xlarge",
"DataTiering": "disabled"
"PendingModifiedValues": {}
}
}
```

如需詳細資訊，請參閱 [modify-replication-group](#) 在AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately`，請使用具有下列參數的 `命令來 AWS CLI describe-cache-clusters` 檢查快取叢集的狀態。當狀態變更為可用，您就可以開始使用新的、較大的快取叢集節點。

### 擴展 Valkey 或 Redis OSS 快取叢集 (ElastiCache API)

下列程序會使用 `awscli`，將快取叢集從目前的節點類型擴展到新的較大節點類型 ElastiCache API。在此過程中，`awscli` 會 ElastiCache 更新 DNS 項目，以便它們指向新的節點。因此，您不需在應用程式中更新端點。對於 Valkey 7.2 及更新版本 Redis OSS 5.0.5 及更新版本，您可以在叢集持續上線並提供傳入請求時擴展自動容錯移轉啟用的叢集。在 Redis OSS 4.0.10 及更低版本上，您可能會注意到 DNS 項目更新時，先前節點的讀取和寫入短暫中斷。

向上擴展至較大的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

### 擴展 Valkey 或 Redis OSS 快取叢集 (ElastiCache API)

1. 使用 `ListAllowedNodeTypeModifications` 動作搭配下列參數，ElastiCache API 決定您可以擴展到哪些節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考中。

2. 使用 `ModifyReplicationGroup` ElastiCache API 動作和下列參數，將您目前的複寫群組擴展到新的節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。

- `CacheNodeType` - 此複寫群組中較大的新快取叢集節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypeModifications` 動作傳回的執行個體類型之一。
- `CacheParameterGroupName` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
- `ApplyImmediately` - 設定為 `true` 會立即套用擴充規模程序。若要將向上擴展程序延期至下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&CacheNodeType=cache.m3.2xlarge
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyReplicationGroup](#) 在 Amazon ElastiCache API 參考 中。

3. 如果您使用 `ApplyImmediately=true`，請使用具有下列參數 `DescribeReplicationGroups` 的動作來 ElastiCache API 監控複寫群組的狀態。當狀態從 `modifying` (修改中) 變更為 `available` (可用)，您即可以開始寫入新的、已向上擴展的複寫群組。
  - `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [DescribeReplicationGroups](#) 在 Amazon ElastiCache API 參考 中。

## 線上縮減規模

### 主題

- [縮減 Valkey 或 Redis OSS 快取叢集 \( 主控台 \)](#)
- [縮減 Valkey 或 Redis OSS 快取叢集 \( AWS CLI \)](#)
- [縮減 Valkey 或 Redis OSS 快取叢集 \( ElastiCache API \)](#)

### 縮減 Valkey 或 Redis OSS 快取叢集 ( 主控台 )

下列程序說明如何使用 ElastiCache 管理主控台縮減 Valkey 或 Redis OSS 叢集規模。在此過程中，您的 Valkey 或 Redis OSS 叢集將繼續以最短的停機時間提供請求。

### 若要縮減 Valkey 或 Redis OSS 叢集 ( 主控台 )

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS 叢集。
3. 從叢集的清單中，選擇您偏好的叢集。
4. 選擇 Modify (修改)。
5. 在 Modify Cluster (修改叢集) 精靈中：
  - 從 Node type (節點類型) 清單選擇您要擴展的節點類型。若要向下擴展，請選取小於現有節點的節點類型。請注意，並非所有節點類型都可縮減規模。
6. 如果您要立即執行向下擴展程序，請選擇 Apply immediately (立即套用) 方塊。如果未選擇 Apply immediately (立即套用) 方塊，則會在此叢集的下一個維護時段期間，執行向下擴展程序。
7. 選擇 Modify (修改)。

如果您在上一個步驟中選擇 Apply immediately (立即套用)，叢集的狀態變更為 modifying (修改中)。當狀態變更為 available (可用)，修改即已完成，並且您可以開始使用新叢集。



## 縮減 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

下列程序說明如何使用 縮減 Valkey 或 Redis OSS 快取叢集的規模 AWS CLI。在此過程中，您的叢集將繼續以最短的停機時間提供請求。

### 若要縮減 Valkey 或 Redis OSS 快取叢集 ( AWS CLI )

1. 使用下列參數執行 `list-allowed-node-type-modifications` 命令，AWS CLI 決定您可以縮減規模的節點類型。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-replication-group-id
```

針對 Windows：

```
aws elasticache list-allowed-node-type-modifications ^\
 --replication-group-id my-replication-group-id
```

從上述命令輸出看起來像這樣 ( JSON 格式 )。

```
{
 "ScaleUpModifications": [
 "cache.m3.2xlarge",
 "cache.m3.large",
 "cache.m3.xlarge",
 "cache.m4.10xlarge",
 "cache.m4.2xlarge",
 "cache.m4.4xlarge",
 "cache.m4.large",
 "cache.m4.xlarge",
 "cache.r3.2xlarge",
 "cache.r3.4xlarge",
 "cache.r3.8xlarge",
 "cache.r3.large",
 "cache.r3.xlarge"
]

 "ScaleDownModifications": [
 "cache.t2.micro",
 "cache.t2.small",
]
}
```

```

 "cache.t2.medium ",
 "cache.t1.small"
]
}

```

如需詳細資訊，請參閱 [list-allowed-node-type-修改](#) 在 AWS CLI 參考 中。

2. 使用 AWS CLI `modify-replication-group` 命令和下列參數，修改複寫群組以縮減至新的較小節點類型。
  - `--replication-group-id` - 您要縮減規模的複寫群組名稱。
  - `--cache-node-type` - 您要擴展快取叢集的新節點類型。此值必須是步驟 1 中 `list-allowed-node-type-modifications` 命令傳回的其中一個節點類型。
  - `--cache-parameter-group-name` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
  - `--apply-immediately` - 會造成立即套用擴充規模程序。若要將向下擴展程序延期至叢集的下一個維護時段，請使用 `--no-apply-immediately` 參數。

若為 Linux、macOS 或 Unix：

```

aws elasticache modify-replication-group \
 --replication-group-id my-redis-cluster \
 --cache-node-type cache.t2.micro \
 --apply-immediately

```

針對 Windows：

```

aws elasticache modify-replication-group ^
 --replication-group-id my-redis-cluster ^
 --cache-node-type cache.t2.micro ^
 --apply-immediately

```

從上述命令輸出看起來像這樣（JSON 格式）。

```

{
 "ReplicationGroup": {
 "Status": "modifying",

```

```

 "Description": "my-redis-cluster",
 "NodeGroups": [
 {
 "Status": "modifying",
 "Slots": "0-16383",
 "NodeGroupId": "0001",
 "NodeGroupMembers": [
 {
 "PreferredAvailabilityZone": "us-east-1f",
 "CacheNodeId": "0001",
 "CacheClusterId": "my-redis-cluster-0001-001"
 },
 {
 "PreferredAvailabilityZone": "us-east-1d",
 "CacheNodeId": "0001",
 "CacheClusterId": "my-redis-cluster-0001-002"
 }
]
 }
],
 "ConfigurationEndpoint": {
 "Port": 6379,
 "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
 },
 "ClusterEnabled": true,
 "ReplicationGroupId": "my-redis-cluster",
 "SnapshotRetentionLimit": 1,
 "AutomaticFailover": "enabled",
 "SnapshotWindow": "07:30-08:30",
 "MemberClusters": [
 "my-redis-cluster-0001-001",
 "my-redis-cluster-0001-002"
],
 "CacheNodeType": "cache.t2.micro",
 "DataTiering": "disabled"
 "PendingModifiedValues": {}
 }
}

```

如需詳細資訊，請參閱 [modify-replication-group](#) 在AWS CLI 參考 中。

3. 如果您使用 `--apply-immediately`，請使用具有下列參數的 命令來 AWS CLI `describe-cache-clusters` 檢查快取叢集的狀態。當狀態變更為可用，您就可以開始使用新的、較小的快取叢集節點。

### 縮減 Valkey 或 Redis OSS 快取叢集 (ElastiCache API)

下列程序會使用 將複寫群組從目前的節點類型擴展到新的較小節點類型 ElastiCache API。在此過程中，您的 Valkey 或 Redis OSS 叢集將繼續以最短的停機時間提供請求。

向下擴展至較小的節點類型所需的時間可能不同，取決於節點類型和您目前快取叢集中的資料量。

### 縮減規模 (ElastiCache API)

1. 使用 `ListAllowedNodeTypeModifications` 動作搭配下列參數，ElastiCache API 決定您可以縮減為哪些節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。使用此參數來描述特定複寫群組 (而非所有複寫群組)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [ListAllowedNodeTypeModifications](#) 在 Amazon ElastiCache API 參考中。

2. 使用 `ModifyReplicationGroup` ElastiCache API 動作和下列參數，將您目前的複寫群組縮減為新的節點類型。
  - `ReplicationGroupId` - 複寫群組的名稱。
  - `CacheNodeType` - 此複寫群組中較小的新快取叢集節點類型。此值必須是上一個步驟中 `ListAllowedNodeTypeModifications` 動作傳回的執行個體類型之一。

- `CacheParameterGroupName` - [選用] 如果您使用 `reserved-memory` 來管理叢集的保留記憶體，請使用此參數。指定自訂快取參數群組，其會為您的新節點類型保留正確的記憶體數量。如果您要使用 `reserved-memory-percent`，則可以忽略此參數。
- `ApplyImmediately` - 設定為 `true` 會立即套用縮減規模程序。若要將向下擴展程序延期至下一個維護時段，請使用 `ApplyImmediately=false`。

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=ModifyReplicationGroup
 &ApplyImmediately=true
 &CacheNodeType=cache.t2.micro
 &CacheParameterGroupName=redis32-m3-2x1
 &ReplicationGroupId=myReplGroup
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20141201T220302Z
 &Version=2014-12-01
 &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
 &X-Amz-Date=20141201T220302Z
 &X-Amz-SignedHeaders=Host
 &X-Amz-Expires=20141201T220302Z
 &X-Amz-Credential=<credential>
 &X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyReplicationGroup](#) 在 Amazon ElastiCache API 參考 中。

## 適用於 Valkey 和 Redis JSON 的 入門 OSS

ElastiCache 支援原生 JavaScript Object Notation (JSON) 格式，這是一種簡單、無結構描述的方式，可編碼 Valkey 和 Redis OSS叢集內的複雜資料集。您可以使用叢集內的 JavaScript 物件表示法 (JSON) 格式原生存放和存取資料，並更新存放在這些叢集中的JSON資料，而不需要管理自訂程式碼來序列化和還原序列化。

除了將 Valkey 和 Redis OSSAPI操作用於透過 操作的應用程式之外JSON，您現在可以有效率地擷取和更新JSON文件的特定部分，而無需操作整個物件。這可以提高效能並降低成本。您也可以使用 [Goessner 樣式JSONPath查詢](#)來搜尋JSON文件內容。

使用支援的引擎版本建立叢集後，JSON資料類型和相關聯的命令會自動可用。API RDB與JSON 模組第2版相容，因此您可以輕鬆將現有的JSON型Valkey和Redis OSS應用程式遷移至ElastiCache。如需支援命令的詳細資訊，請參閱[支援的Valkey和Redis OSS命令](#)。

JSON與相關的指標JsonBasedCmds和JsonBasedCmdsLatency會併入，CloudWatch以監控此資料類型的使用情況。如需詳細資訊，請參閱[Valkey和Redis的指標OSS](#)。

#### Note

若要使用JSON，您必須執行Valkey 7.2及更新版本，或Redis OSS引擎6.2.6版或更新版本。

## 主題

- [JSON 資料類型概觀](#)
- [支援的Valkey和Redis OSS命令](#)

## JSON 資料類型概觀

ElastiCache支援許多用於JSON資料類型的Valkey和Redis OSS命令。以下是JSON資料類型的概觀，以及支援命令的詳細清單。

## 術語

術語	描述
JSON 文件	表示JSON金鑰的值。
JSON 值	指JSON文件的子集，包括代表整個文件的根。值可以是容器或容器內的項目。
JSON 元素	等同於JSON值。

## 支援JSON的標準

JSON格式符合[RFC 7159](#)和[ECMA-404](#) JSON資料交換標準。UTF支援JSON文字中的-8 [Unicode](#)。

## 根元素

根元素可以是任何JSON資料類型。請注意，在舊版 RFC 4627 中，僅允許物件或陣列作為根值。自更新至 RFC 7159 以來，JSON文件的根可以是任何JSON資料類型。

## 文件大小限制

JSON 文件會以針對快速存取和修改而最佳化的格式儲存在內部。比起相同文件的等同序列化版本，這種格式一般會導致使用更多記憶體。

單一JSON文件的記憶體使用量限制為 64 MB，即記憶體內資料結構的大小，而非JSON字串。您可以使用 `JSON.DEBUG MEMORY` 命令來檢查JSON文件耗用的記憶體量。

## JSON ACLs

- 與現有的每個資料類型類別（`@string`、`@hash` 等）類似，會新增一個類別 `@json`，以簡化對JSON命令和資料的管理存取。沒有其他現有的 Valkey 或 Redis OSS命令是 `@json` 類別的成員。所有JSON命令都會強制執行任何鍵空間或命令限制和許可。
- 有五個現有的 Valkey 和 Redis OSSACL類別會更新，以包含新的JSON命令：`@read`、`@write`、`@fast`、`@slow` 和 `@admin`。下表指示JSON命令對應至適當的類別。

### ACL

JSON 命令	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		

JSON 命令	@read	@write	@fast	@slow	@admin
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		
JSON.OBJECTLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		



JSON 命令	@read	@write	@fast	@slow	@admin
JSON.STRL EN	y		y		
JSON.STRL EN	y		y		
JSON.TOGG LE		y	y		
JSON.TYPE	y		y		
JSON.NUMI NCRBY		y	y		

## 巢狀深度限制

當JSON物件或陣列的元素本身是另一個JSON物件或陣列時，表示內部物件或陣列在外部物件或陣列中「巢狀」。巢狀深度上限為 128。任何建立巢狀深度大於 128 文件的嘗試，都會遭到拒絕，並顯示錯誤。

## 命令語法

大多數命令需要金鑰名稱作為第一個引數。部分命令也有路徑引數。如果路徑引數為選用且未提供，則預設為根。

標記法：

- 必要引數以角括弧括住。例如：`<key>`
- 選用引數以方括弧括住。例如：`[path]`
- 其他選用引數會以省略號 ("...") 表示。例如：`[json ...]`

## 路徑語法

Redis JSON支援兩種路徑語法：

- 增強語法 – JSONPath 遵循 [Goessner](#) 所描述的語法，如下表所示。為清楚說明，我們重新排序並修改表格中的描述。

- 受限語法 – 查詢功能有限。

### Note

部分命令的結果對於使用哪種類型的路徑語法很敏感。

如果查詢路徑以 '\$' 開頭，它會使用增強型語法。否則，將使用受限語法。

### 增強型語法

符號/表達式	描述
\$	根元素。
. 或 []	子運算子。
..	遞迴下降。
*	萬用字元。物件或陣列中的所有元素。
[]	陣列下標運算子。索引以 0 為基礎。
[,]	聯集運算子。
[start:end:step]	陣列分割運算子。
?()	將篩選條件 (指令碼) 表達式套用至目前陣列或物件。
()	篩選條件表達式。
@	用於指正在處理的目前節點的篩選條件表達式。
==	等於，用於篩選條件表達式。
!=	不等於，用於篩選條件表達式。
>	大於，用於篩選條件表達式。
>=	大於等於，用於篩選條件表達式。

符號/表達式	描述
<	小於，用於篩選條件表達式。
<=	小於等於，用於篩選條件表達式。
&&	邏輯 AND，用於合併多個篩選條件表達式。
	邏輯 OR，用於結合多個篩選條件表達式。

## 範例

下列範例建立在 [Goessner](#) 的範例XML資料上，我們已透過新增其他欄位修改這些資料。

```
{ "store": {
 "book": [
 { "category": "reference",
 "author": "Nigel Rees",
 "title": "Sayings of the Century",
 "price": 8.95,
 "in-stock": true,
 "sold": true
 },
 { "category": "fiction",
 "author": "Evelyn Waugh",
 "title": "Sword of Honour",
 "price": 12.99,
 "in-stock": false,
 "sold": true
 },
 { "category": "fiction",
 "author": "Herman Melville",
 "title": "Moby Dick",
 "isbn": "0-553-21311-3",
 "price": 8.99,
 "in-stock": true,
 "sold": false
 },
 { "category": "fiction",
 "author": "J. R. R. Tolkien",
 "title": "The Lord of the Rings",
 "isbn": "0-395-19395-8",
```

```

 "price": 22.99,
 "in-stock": false,
 "sold": false
 }
],
"bicycle": {
 "color": "red",
 "price": 19.95,
 "in-stock": true,
 "sold": false
}
}
}

```

路徑	描述
<code>\$.store.book[*].author</code>	商店中所有書籍的作者。
<code>\$.author</code>	所有作者。
<code>\$.store.*</code>	商店的所有成員。
<code>\$["store"].*</code>	商店的所有成員。
<code>\$.store..price</code>	商店裡所有商品的價格。
<code>\$.*</code>	JSON 結構的所有遞迴成員。
<code>\$.book[*]</code>	所有書籍。
<code>\$.book[0]</code>	第一本書。
<code>\$.book[-1]</code>	最後一本書。
<code>\$.book[0:2]</code>	前兩本書。
<code>\$.book[0,1]</code>	前兩本書。
<code>\$.book[0:4]</code>	從索引 0 到 3 (不包括結尾索引) 的書籍。
<code>\$.book[0:4:2]</code>	索引 0、2 的書籍。

路徑	描述
<code>\$.book[?(@.isbn)]</code>	所有具有 ISBN 號碼的書籍。
<code>\$.book[?(@.price&lt;10)]</code>	不到 10 美元的所有書籍。
<code>'\$.book[?(@.price &lt; 10)]'</code>	不到 10 美元的所有書籍。(如果路徑包含空格，必須以英文引號括住。)
<code>'\$.book[?(@["price"] &lt; 10)]'</code>	不到 10 美元的所有書籍。
<code>'\$.book[?(@.["price"] &lt; 10)]'</code>	不到 10 美元的所有書籍。
<code>\$.book[?(@.price&gt;=10&amp;&amp;@.price&lt;=100)]</code>	價格範圍介於 10 至 100 美元 (含) 的所有書籍。
<code>'\$.book[?(@.price&gt;=10 &amp;&amp; @.price&lt;=100)]'</code>	價格範圍介於 10 至 100 美元 (含) 的所有書籍。(如果路徑包含空格，必須以英文引號括住。)
<code>\$.book[?(@.sold==true  @.in-stock==false)]</code>	售完或缺貨的所有書籍。
<code>'\$.book[?(@.sold == true    @.in-stock == false)]'</code>	售完或缺貨的所有書籍。(如果路徑包含空格，必須以英文引號括住。)
<code>'\$.store.book[?(@.["category"] == "fiction")]</code>	小說類別中的所有書籍。
<code>'\$.store.book[?(@.["category"] != "fiction")]</code>	非小說類別中的所有書籍。

其他篩選條件表達式範例：

```
127.0.0.1:6379> JSON.SET k1 . '{"books": [{"price":5,"sold":true,"in-stock":true,"title":"foo"}, {"price":15,"sold":false,"title":"abc"}]}'
OK
127.0.0.1:6379> JSON.GET k1 $.books[?(@.price>1&&@.price<20&&@.in-stock)]
"[{"price":5,"sold":true,"in-stock":true,"title":"foo"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.price>1 && @.price<20 && @.in-stock)]'
"[{"price":5,"sold":true,"in-stock":true,"title":"foo"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?((@.price>1 && @.price<20) && (@.sold==false))]'
"[{"price":15,"sold":false,"title":"abc"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.title == "abc")]'
[{"price":15,"sold":false,"title":"abc"}]
```

```

127.0.0.1:6379> JSON.SET k2 . '[1,2,3,4,5]'
127.0.0.1:6379> JSON.GET k2 $.*.[?(@>2)]
"[3,4,5]"
127.0.0.1:6379> JSON.GET k2 '$.*.[?(@ > 2)]'
"[3,4,5]"

127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
OK
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
"[true,true]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
"[true,true]"
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
"[2,3,4]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
"[2,3,4]"

```

## 受限語法

符號/表達式	描述
. 或 []	子運算子。
[]	陣列下標運算子。索引以 0 為基礎。

## 範例

路徑	描述
.store.book[0].author	第一本書的作者。
.store.book[-1].author	最後一本書的作者。
.address.city	城市名稱。
["store"]["book"][0]["title"]	第一本書的標題。
["store"]["book"][-1]["title"]	最後一本書的標題。

**Note**

本文件中引用的所有 [Goessner](#) 內容均受 [創用 CC 授權](#) 規範。

## 常見錯誤字首

每個錯誤訊息都有一個字首。以下是常見錯誤字首清單。

字首	描述
ERR	一般錯誤。
LIMIT	超過大小上限時發生的錯誤。例如，超過文件大小上限或巢狀深度限制。
NONEXISTENT	沒有索引鍵或路徑。
OUTOFBOUNDARIES	超出範圍的陣列索引。
SYNTAXERR	語法錯誤。
WRONGTYPE	錯誤值類型。

## JSON相關指標

提供下列JSON資訊指標：

Info	描述
json_total_memory_bytes	分配給JSON物件的記憶體總數。
json_num_documents	Valkey 或 Redis 中的文件總數OSS。

若要查詢核心指標，請執行下列命令：

```
info json_core_metrics
```

## ElastiCache 如何使用 Valkey 和 Redis 與 OSS 互動 JSON

下一節說明如何 ElastiCache 使用 Valkey 和 Redis 與 JSON 資料類型 OSS 互動。

### 運算子優先順序

評估用於篩選的條件表達式時，`&&` 優先，然後評估 `||`，就像大多數語言一樣。系統會先執行括弧內的操作。

### 路徑巢狀上限行為

ElastiCache (Redis OSS) 中的路徑巢狀限制上限為 128。`$.a.b.c.d...` 等值只能達到 128 個等級。

### 處理數值

JSON 沒有整數和浮點數的個別資料類型。均稱為數字。

### 數值表示法：

輸入時收到 JSON 數字時，會轉換為兩個內部二進位表示式之一：64 位元帶正負號的整數或 64 位元 IEEE 雙精度浮點。不會保留原始字串和全部格式化。因此，當數字作為 JSON 回應的一部分輸出時，它會從內部二進位表示法轉換為使用一般格式規則的可列印字串。這些規則可能導致產生的字串與接收的字串不同。

### 算術命令 NUMINCRBY 和 NUMMULTBY：

- 如果兩個數字都是整數，且結果超出的範圍 `int64`，則會自動變成 64 位元 IEEE 雙精度浮點數。
- 如果至少一個數字是浮點數，則結果為 64 位元 IEEE 雙精度浮點數。
- 如果結果超過 64 位元 IEEE 的雙重範圍，命令會傳回 `OVERFLOW` 錯誤。

如需可用命令的詳細清單，請參閱 [支援的 Valkey 和 Redis OSS 命令](#)。

### 直接陣列篩選

ElastiCache 透過 Valkey 或 Redis 直接 OSS 篩選陣列物件。

對於類似的資料 `[0,1,2,3,4,5,6]` 和類似的路徑查詢 `[$[?(@<4)]]`，或類似的資料 `{"my_key": [0,1,2,3,4,5,6]}` 和類似的路徑查詢 `$.my_key[?(@<4)]`，在這兩種情況下，ElastiCache Valkey 或 Redis OSS 都會傳回 `【1, 2, 3】`。



## 陣列索引行為

ElastiCache 搭配 Valkey 或 Redis OSS可允許陣列的正索引和負索引。針對長度為五的陣列，0 會查詢第一個元素，1 會查詢第二個，以此類推。負數從陣列結尾開始，所以 -1 會查詢第五個元素，-2 會查詢第四個元素，以此類推。

為了確保客戶的可預測行為，ElastiCache 使用 Valkey 或 Redis OSS不會將陣列索引向下或向上四捨五入，因此如果您的陣列長度為 5，呼叫索引 5 或更高，或 -6 或更低，則不會產生結果。

## 嚴格語法評估

即使JSON路徑子集包含有效的路徑，MemoryDB 不允許具有無效語法的路徑。這是為了我們的客戶保持正確行為。

## 支援的 Valkey 和 Redis OSS命令

ElastiCache 支援下列 Valkey 和 Redis OSSJSON命令：

### 主題

- [JSON.ARRAPPEND](#)
- [JSON.ARRINDEX](#)
- [JSON.ARRINSERT](#)
- [JSON.ARRLEN](#)
- [JSON.ARRPOP](#)
- [JSON.ARRTRIM](#)
- [JSON.CLEAR](#)
- [JSON.DEBUG](#)
- [JSON.DEL](#)
- [JSON.FORGET](#)
- [JSON.GET](#)
- [JSON.MGET](#)
- [JSON.NUMINCRBY](#)
- [JSON.NUMMULTBY](#)
- [JSON.OBJLEN](#)

- [JSON.OBJKEYS](#)
- [JSON.RESP](#)
- [JSON.SET](#)
- [JSON.STRAPPEND](#)
- [JSON.STRLEN](#)
- [JSON.TOGGLE](#)
- [JSON.TYPE](#)

## JSON.ARRAPPEND

將一或多個值附加到路徑中的陣列值。

### 語法

```
JSON.ARRAPPEND <key> <path> <json> [json ...]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（必要） – JSON 路徑。
- json（必要） – 要附加至陣列JSON的值。

### 傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑的新陣列長度。
- 如果值不是陣列，其相應的傳回值為 null。
- SYNTAXERR 如果其中一個輸入 json 引數不是有效的JSON字串，則為錯誤。
- 如果沒有路徑，會發生 NONEXISTENT 錯誤。

如果路徑是受限語法：

- 整數，新陣列長度。
- 如果選取多個陣列值，命令會傳回上次更新陣列的新長度。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。

- SYNTAXERR 如果其中一個輸入 json 引數不是有效的JSON字串，則為錯誤。
- 如果沒有路徑，會發生 NONEXISTENT 錯誤。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND k1 $[*] '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[["c"],["a"],["a","b"],["c"]]"
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND k1 [-1] '"c"'
(integer) 3
127.0.0.1:6379> JSON.GET k1
"[[],["a"],["a"],["b"],["c"]]"
```

## JSON.ARRINDEX

搜尋路徑上陣列中第一次出現的純量JSON值。

- 將索引四捨五入到陣列的開頭和結尾，處理超出範圍的錯誤。
- 如果開頭 > 結尾，傳回 -1 (找不到)。

## 語法

```
JSON.ARRINDEX <key> <path> <json-scalar> [start [end]]
```

- 金鑰 (必要) – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。

- 路徑 (必要) – JSON 路徑。
- json-scalar (必要) – 要搜尋的純量值。JSON 純量是指非物件或陣列的值。也就是說，字串、數字、布林值和 null 是純量值。
- start (選用) – 開頭索引，包含。如果未提供，預設為 0。
- end (選用) – 結束索引，排除。如果未提供，預設為 0，表示包含最後一個元素。0 或 -1 表示包含最後一個元素。

## 傳回

如果路徑是增強型語法：

- 整數陣列。每個值都是路徑的陣列中相符元素的索引。如果找不到，則值為 -1。
- 如果值不是陣列，其相應的傳回值為 null。

如果路徑是受限語法：

- 如果找不到，則為整數、相符元素的索引或 -1。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 $[*] '"b"'
1) (integer) -1
2) (integer) -1
3) (integer) 1
4) (integer) 1
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 .children '"Tom"'
(integer) 2
```

## JSON.ARRINSERT

將一或多個值插入索引前路徑的陣列值。

### 語法

```
JSON.ARRINSERT <key> <path> <index> <json> [json ...]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（必要） – JSON 路徑。
- index（必要） – 插入值之前的陣列索引。
- json（必要） – 要附加至陣列JSON的值。

### 傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑的新陣列長度。
- 如果值是空陣列，其相應的傳回值為 null。
- 如果值不是陣列，其相應的傳回值為 null。
- 如果索引引數超出範圍，會發生 OUTFBOUNDARIES 錯誤。

如果路徑是受限語法：

- 整數，新陣列長度。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。
- 如果索引引數超出範圍，會發生 OUTFBOUNDARIES 錯誤。

### 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 $[*] 0 '"c"'
```

```
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"c\",\"a\"],[\"c\",\"a\",\"b\"]]"
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 . 0 "c"
(integer) 4
127.0.0.1:6379> JSON.GET k1
"[[\"c\", [], [\"a\"], [\"a\", \"b\"]]"
```

## JSON.ARRLEN

取得路徑上陣列值的長度。

語法

```
JSON.ARRLEN <key> [path]
```

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用）– JSON 路徑。如果未提供，則預設為根。

傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑的陣列長度。
- 如果值不是陣列，其相應的傳回值為 null。
- 如果沒有文件索引鍵，則為 null。

如果路徑是受限語法：

- 大量字串陣列。每個元素都是物件中的索引鍵名稱。

- 整數，陣列長度。
- 如果選取多個物件，命令會傳回第一個陣列的長度。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。
- 如果沒有路徑，會發生 WRONGTYPE 錯誤。
- 如果沒有文件索引鍵，則為 null。

## 範例

### 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], [\"a\"], [\"a\", \"b\"], [\"a\", \"b\", \"c\"]]'
(error) SYNTAXERR Failed to parse JSON string due to syntax error
127.0.0.1:6379> JSON.SET k1 . '[[[], [\"a\"], [\"a\", \"b\"], [\"a\", \"b\", \"c\"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[[], \"a\", [\"a\", \"b\"], [\"a\", \"b\", \"c\"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
1) (integer) 0
2) (nil)
3) (integer) 2
4) (integer) 3
5) (nil)
```

### 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], [\"a\"], [\"a\", \"b\"], [\"a\", \"b\", \"c\"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k1 $[3]
1) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[[], \"a\", [\"a\", \"b\"], [\"a\", \"b\", \"c\"], 4]'
```

```
OK
127.0.0.1:6379> JSON.ARRLEN k2 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 $[1]
1) (nil)
127.0.0.1:6379> JSON.ARRLEN k2 $[2]
1) (integer) 2
```

## JSON.ARRPOP

移除並傳回陣列索引的元素。彈出空陣列會傳回 null。

### 語法

```
JSON.ARRPOP <key> [path [index]]
```

- 金鑰 ( 必要 ) – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑 ( 選用 ) – JSON 路徑。如果未提供，則預設為根。
- index (選用) – 開始彈出的陣列位置。
  - 如果未提供，預設為 -1，表示最後一個元素。
  - 負值表示從最後一個元素數起的位置。
  - 超出範圍的索引會四捨五入到各自的陣列範圍。

### 傳回

如果路徑是增強型語法：

- 大量字串陣列代表每個路徑的彈出值。
- 如果值是空陣列，其相應的傳回值為 null。
- 如果值不是陣列，其相應的傳回值為 null。

如果路徑是受限語法：

- 大量字串，代表彈出JSON的值。
- 如果陣列是空的，則為 null。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。



## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1 $[*]
1) (nil)
2) "\"a\""
3) "\"b\""
127.0.0.1:6379> JSON.GET k1
"[[[], [], [\"a\"]]"
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1
"[\\"a\\", \\"b\"]"
127.0.0.1:6379> JSON.GET k1
"[[[], [\"a\"]]"

127.0.0.1:6379> JSON.SET k2 . '[[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k2 . 0
"[]"
127.0.0.1:6379> JSON.GET k2
"[[\"a\"], [\"a\\", \\"b\"]]"
```

## JSON.ARRTRIM

修剪路徑上的陣列，使其成為子陣列 [start, end]，都包含在內。

- 如果陣列是空的，不必做任何事，會傳回 0。
- 如果開頭 < 0，則將其視為 0。
- 如果結尾 >= 大小 (陣列的大小)，則將其視為 size-1。
- 如果開頭 >= 大小或開頭 > 結尾，清空陣列並傳回 0。

## 語法

```
JSON.ARRINSERT <key> <path> <start> <end>
```

- 金鑰 (必要) – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑 (必要) – JSON 路徑。
- start (必要) – 開頭索引，包含。
- end (必要) – 結束索引，包含。

## 傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑的新陣列長度。
- 如果值是空陣列，其相應的傳回值為 null。
- 如果值不是陣列，其相應的傳回值為 null。
- 如果索引引數超出範圍，會發生 OUTFOUBOUNDARIES 錯誤。

如果路徑是受限語法：

- 整數，新陣列長度。
- 如果陣列是空的，則為 null。
- 如果路徑上的值不是陣列，會發生 WRONGTYPE 錯誤。
- 如果索引引數超出範圍，會發生 OUTFOUBOUNDARIES 錯誤。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
127.0.0.1:6379> JSON.GET k1
"[[],[\\"a\"],[\\"a\\","\\"b\"],[\\"a\\","\\"b\\"]]"
```

## 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 .children 0 1
(integer) 2
127.0.0.1:6379> JSON.GET k1 .children
"[\\"John\\",\\"Jack\\""]"
```

## JSON.CLEAR

清除陣列或路徑上的物件。

### 語法

```
JSON.CLEAR <key> [path]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用） – JSON 路徑。如果未提供，則預設為根。

### 傳回

- 整數，已清除的容器數目。
- 清除一個空陣列或物件計為清除 1 個容器。
- 清除非容器值會傳回 0。

### 範例

```
127.0.0.1:6379> JSON.SET k1 . '[[[], [0], [0,1], [0,1,2], 1, true, null, "d"]]'
OK
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 7
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 4
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
```

```
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

## JSON.DEBUG

報告資訊。支援的子命令如下：

- MEMORY <key> **【path】** – 以JSON值的位元組為單位報告記憶體用量。如果未提供，路徑預設為根。
- FIELDS <key> **【path】** – 報告指定文件路徑中的欄位數目。如果未提供，路徑預設為根。每個非容器JSON值都會計為一個欄位。物件和陣列會遞迴計算每個包含JSON值的一個欄位。除根容器外，每個容器值都計為一個附加欄位。
- HELP – 列印 命令的說明訊息。

### 語法

```
JSON.DEBUG <subcommand & arguments>
```

取決於子命令：

### MEMORY

- 如果路徑是增強型語法：
  - 傳回整數陣列，代表每個路徑上JSON值的記憶體大小（以位元組為單位）。
  - 如果 Valkey 或 Redis OSS金鑰不存在，則傳回空陣列。
- 如果路徑是受限語法：
  - 傳回整數、記憶體大小和以位元組為單位JSON的值。
  - 如果 Valkey 或 Redis OSS金鑰不存在，則傳回 null。

### FIELDS

- 如果路徑是增強型語法：
  - 傳回整數陣列，其代表每個路徑JSON的值欄位數。
  - 如果 Valkey 或 Redis OSS金鑰不存在，則傳回空陣列。

- 如果路徑是受限語法：
  - 傳回整數，JSON值的欄位數目。
  - 如果 Valkey 或 Redis OSS金鑰不存在，則傳回 null。

HELP – 傳回一系列說明訊息。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2},
[1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
5) (integer) 16
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
6) (integer) 0
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
```

```
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}], "children": [], "spouse": null}'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1
(integer) 632
127.0.0.1:6379> JSON.DEBUG MEMORY k1 .phoneNumbers
(integer) 166

127.0.0.1:6379> JSON.DEBUG FIELDS k1
(integer) 19
127.0.0.1:6379> JSON.DEBUG FIELDS k1 .address
(integer) 4

127.0.0.1:6379> JSON.DEBUG HELP
1) JSON.DEBUG MEMORY <key> [path] - report memory size (bytes) of the JSON element.
 Path defaults to root if not provided.
2) JSON.DEBUG FIELDS <key> [path] - report number of fields in the JSON element. Path
 defaults to root if not provided.
3) JSON.DEBUG HELP - print help message.
```

## JSON.DEL

刪除文件金鑰中路徑JSON的值。如果路徑是根，則相當於從 Valkey 或 Redis 刪除金鑰OSS。

### 語法

```
JSON.DEL <key> [path]
```

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS金鑰。
- 路徑（選用）– JSON 路徑。如果未提供，則預設為根。

### 傳回

- 刪除的元素數目。
- 如果 Valkey 或 Redis OSS金鑰不存在，則為 0。
- 如果JSON路徑無效或不存在，則為 0。

### 範例

## 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 $.d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 $.e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

## 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 .d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 .e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

## JSON.FORGET

[JSON.DEL](#) 的別名。

## JSON.GET

傳回JSON以一或多個路徑序列化的。

## 語法

```
JSON.GET <key>
```

```
[INDENT indentation-string]
[NEWLINE newline-string]
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- INDENT/NEWLINE/SPACE（選用） – 控制傳回 JSON 字串的格式，即「Pretty Print」。個別預設值為空字串。可在任何組合中覆寫。可以按任何順序指定。
- NOESCAPE - 選用，允許存在於舊版相容性，且沒有其他效果。
- 路徑（選用） – 零個或多個 JSON 路徑，如果沒有指定，則預設為根。路徑引數必須放在最後。

## 傳回

### 增強型路徑語法：

如果提供一個路徑：

- 傳回值陣列的序列化字串。
- 如果未選取任何值，此命令會傳回空陣列。

如果提供多個路徑：

- 傳回字串 JSON 物件，其中每個路徑都是金鑰。
- 如果混合增強型和受限路徑語法，結果會按照增強型語法。
- 如果沒有路徑，則其對應的值會是空陣列。

## 範例

### 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK
```



```

127.0.0.1:6379> JSON.GET k1 $.address.*
"[\"21 2nd Street\", \"New York\", \"NY\", \"10021-3100\"]"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
"[\"\\t\"21 2nd Street\", \"\\t\"New York\", \"\\t\"NY\", \"\\t\"10021-3100\"\\n]"
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
"{\"$.firstName\": [\"John\"], \"$.lastName\": [\"Smith\"], \"$.age\": [27]}"
127.0.0.1:6379> JSON.SET k2 . '{"a": {}, "b": {"a": 1}, "c": {"a": 1, "b": 2}}'
OK
127.0.0.1:6379> json.get k2 $.*
"[{}, {\"a\": 1}, {\"a\": 1, \"b\": 2}, 1, 1, 2]"

```

### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 .
'{"firstName": "John", "lastName": "Smith", "age": 27, "weight": 135.25, "isAlive": true, "address":
{"street": "21 2nd Street", "city": "New
York", "state": "NY", "zipcode": "10021-3100"}, "phoneNumbers":
[{"type": "home", "number": "212 555-1234"}, {"type": "office", "number": "646
555-4567"}], "children": [], "spouse": null}'
OK
127.0.0.1:6379> JSON.GET k1 .address
"{\"street\": \"21 2nd Street\", \"city\": \"New York\", \"state\": \"NY\", \"zipcode\":
\"10021-3100\"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
"{\"\\t\"street\": \"21 2nd Street\", \"\\t\"city\": \"New York\", \"\\t\"state\": \"NY\", \"\\t\"zipcode\": \"10021-3100\"\\n}"
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\": \"John\", \".lastName\": \"Smith\", \".age\": 27}"

```

## JSON.MGET

從多個文件金鑰JSONs的路徑取得序列化。它會傳回不存在金鑰或JSON路徑的 null。

### 語法

```
JSON.MGET <key> [key ...] <path>
```

- 金鑰（必要） – 文件類型的一或多個 Valkey 或 Redis OSS金鑰。
- 路徑（必要） – JSON 路徑。

## 傳回

- 大量字串陣列。陣列的大小等於命令中的索引鍵數量。陣列的每個元素都會填入 ( a ) 路徑JSON定位的序列化，或 ( b ) 如果金鑰不存在、文件中不存在路徑或路徑無效 ( 語法錯誤 ) ，則為 null。
- 如果任何指定的金鑰存在且不是JSON金鑰，則命令會傳回WRONGTYPE錯誤。

## 範例

### 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK
127.0.0.1:6379> JSON.MGET k1 k2 k3 $.address.city
1) "[\ "New York\ "]"
2) "[\ "Boston\ "]"
3) "[\ "Seattle\ "]"
```

### 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK

127.0.0.1:6379> JSON.MGET k1 k2 k3 .address.city
1) "\"New York\""
2) "\"Seattle\""
3) "\"Seattle\""
```

## JSON.NUMINCRBY

以指定的數字增加路徑上的數值。

### 語法

```
JSON.NUMINCRBY <key> <path> <number>
```

- 金鑰 (必要) – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑 (必要) – JSON 路徑。
- number (必要) – 數字。

### 傳回

如果路徑是增強型語法：

- 大量字串陣列代表每個路徑上產生的值。
- 如果值不是數字，則其對應的傳回值為 null。
- 如果無法剖析數字，會發生 WRONGTYPE 錯誤。
- OVERFLOW 如果結果超出 64 位元 IEEE 雙位元的範圍，則發生錯誤。
- 如果沒有文件索引鍵，則為 NONEXISTENT。

如果路徑是受限語法：

- 大量字串代表結果值。
- 如果選取多個值，命令會傳回上次所更新值的結果。
- 如果路徑上的值不是數字，會發生 WRONGTYPE 錯誤。
- 如果無法剖析數字，會發生 WRONGTYPE 錯誤。
- OVERFLOW 如果結果超出 64 位元 IEEE 雙位元的範圍，則發生錯誤。
- 如果沒有文件索引鍵，則為 NONEXISTENT。

### 範例

## 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 10
"[11,12,13]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[11,12,13]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.a[*] 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.b[*] 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.c[*] 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 $.a.* 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.b.* 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.c.* 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.d.* 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":2,\"b\":3},\"c\":{\"\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 $.a.* 1
"[null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.b.* 1
"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
```

```

"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\", \"b\":2},\"c\":{\"a\":\"a\", \"b\":\"b\"},\"d\":{ \"a\":2, \"b\":\"b\", \"c\":4}}"

```

### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[1] 10
"12"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,12,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .a[*] 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k1 .b[*] 1
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .c[*] 1
"3"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[*] 1
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 .a.* 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k2 .b.* 1
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":1},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"

```

```
127.0.0.1:6379> JSON.NUMINCRBY k2 .c.* 1
"3"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .d.* 1
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

## JSON.NUMMULTBY

以指定數字乘於路徑上的數值。

### 語法

```
JSON.NUMMULTBY <key> <path> <number>
```

- 金鑰 (必要) – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑 (必要) – JSON 路徑。
- number (必要) – 數字。

### 傳回

如果路徑是增強型語法：

- 大量字串陣列代表每個路徑上產生的值。
- 如果值不是數字，則其對應的傳回值為 null。
- 如果無法剖析數字，會發生 WRONGTYPE 錯誤。

- OVERFLOW 如果結果超出 64 位元IEEE雙精度浮點數的範圍，則為錯誤。
- 如果沒有文件索引鍵，則為 NONEXISTENT。

如果路徑是受限語法：

- 大量字串代表結果值。
- 如果選取多個值，命令會傳回上次所更新值的結果。
- 如果路徑上的值不是數字，會發生 WRONGTYPE 錯誤。
- 如果無法剖析數字，會發生 WRONGTYPE 錯誤。
- OVERFLOW 如果結果超出 64 位元IEEE雙位元的範圍，則發生錯誤。
- 如果沒有文件索引鍵，則為 NONEXISTENT。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.a[*] 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.b[*] 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.c[*] 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 $.a.* 2
"[]"
```

```

127.0.0.1:6379> JSON.NUMMULTBY k2 $.b.* 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.c.* 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.d.* 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 $.a.* 2
"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
"[2,null,6]"

```

#### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[1] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,4,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .a[*] 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k1 .b[*] 2
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .c[*] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[*] 2
"6"

```



```
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":1,\"\"b\":2},\"c\":{\"\"a\":1,\"\"b\":2,\"\"c\":3}}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":2,\"\"b\":4},\"c\":{\"\"a\":1,\"\"b\":2,\"\"c\":3}}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":2,\"\"b\":4},\"c\":{\"\"a\":2,\"\"b\":4,\"\"c\":6}}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 .a.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"\"a\":1,\"\"b\":2},\"c\":{\"\"a\":1,\"\"b\":2},\"d\":{\"\"a\":1,\"\"b\":2,\"\"c\":3}}}"
127.0.0.1:6379> JSON.NUMMULTBY k3 .c.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"\"a\":2,\"\"b\":2},\"c\":{\"\"a\":2,\"\"b\":2},\"d\":{\"\"a\":2,\"\"b\":2,\"\"c\":6}}}"
```

## JSON.OBJLEN

取得路徑上物件值中的索引鍵數目。

## 語法

```
JSON.OBJLEN <key> [path]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用） – JSON 路徑。如果未提供，則預設為根。

## 傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑的物件長度。
- 如果值不是物件，其相應的傳回值為 null。
- 如果沒有文件索引鍵，則為 null。

如果路徑是受限語法：

- 整數，物件中的索引鍵數目。
- 如果選取多個物件，命令會傳回第一個物件的長度。
- 如果路徑上的值不是物件，會發生 WRONGTYPE 錯誤。
- 如果沒有路徑，會發生 WRONGTYPE 錯誤。
- 如果沒有文件索引鍵，則為 null。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
```

```

127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)

```

#### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 .a
(integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 .a.*
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.OBJLEN k1 .b
(integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 .b.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .c
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .c.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .d
(integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 .d.*
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .*
(integer) 0

```

## JSON.OBJKEYS

取得路徑上物件值中的索引鍵名稱。

### 語法

```
JSON.OBJKEYS <key> [path]
```

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用）– JSON 路徑。如果未提供，則預設為根。

### 傳回

如果路徑是增強型語法：

- 大量字串陣列。每個元素都是相符物件中的索引鍵陣列。
- 如果值不是物件，其相應的傳回值是空白值。
- 如果沒有文件索引鍵，則為 null。

如果路徑是受限語法：

- 大量字串陣列。每個元素都是物件中的索引鍵名稱。
- 如果選取多個物件，命令會傳回第一個物件的索引鍵。
- 如果路徑上的值不是物件，會發生 WRONGTYPE 錯誤。
- 如果沒有路徑，會發生 WRONGTYPE 錯誤。
- 如果沒有文件索引鍵，則為 null。

### 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3, "b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 $.*
1) (empty array)
2) 1) "a"
3) 1) "a"
```

```

 2) "b"
4) 1) "a"
 2) "b"
 3) "c"
5) (empty array)
127.0.0.1:6379> JSON.OBJKEYS k1 $.d
1) 1) "a"
 2) "b"
 3) "c"

```

### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 .*
1) "a"
127.0.0.1:6379> JSON.OBJKEYS k1 .d
1) "a"
2) "b"
3) "c"

```

## JSON.RESP

傳回 Valkey 或 Redis OSS 序列化通訊協定 ( ) 中指定路徑JSON的值RESP。如果值為容器，則回應為RESP陣列或巢狀陣列。

- JSON null 會映射至 RESP Null 大量字串。
- JSON 布林值會對應至個別的 RESP Simple Strings。
- 整數編號會映射至RESP整數。
- 64 位元IEEE雙浮點數會映射至RESP大量字串。
- JSON 字串會映射到RESP大量字串。
- JSON 陣列會以RESP陣列表示，其中第一個元素是簡單的字串 **【**，後面接著陣列的元素。
- JSON 物件以RESP陣列表示，其中第一個元素是簡單字串 **{**，後面接著鍵值對，每個都是RESP大量字串。

### 語法

```
JSON.RESP <key> [path]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用） – JSON 路徑。如果未提供，則預設為根。

## 傳回

如果路徑是增強型語法：

- 陣列的陣列。每個陣列元素代表一個路徑上值RESP的形式。
- 如果沒有文件索引鍵，則為空陣列。

如果路徑是受限語法：

- 陣列，代表路徑中值RESP的形式。
- 如果沒有文件索引鍵，則為 null。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 $.address
1) 1) {
 2) 1) "street"
 2) "21 2nd Street"
 3) 1) "city"
 2) "New York"
 4) 1) "state"
 2) "NY"
 5) 1) "zipcode"
 2) "10021-3100"
```

```
127.0.0.1:6379> JSON.RESP k1 $.address.*
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
1) 1) [
 2) 1) {
 2) 1) "type"
 2) "home"
 3) 1) "number"
 2) "555 555-1234"
 3) 1) {
 2) 1) "type"
 2) "office"
 3) 1) "number"
 2) "555 555-4567"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
1) 1) {
 2) 1) "type"
 2) "home"
 3) 1) "number"
 2) "212 555-1234"
2) 1) {
 2) 1) "type"
 2) "office"
 3) 1) "number"
 2) "555 555-4567"
```

### 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}], "children":[], "spouse":null}'
OK
```

```
127.0.0.1:6379> JSON.RESP k1 .address
```

```
1) {
2) 1) "street"
 2) "21 2nd Street"
3) 1) "city"
 2) "New York"
4) 1) "state"
 2) "NY"
5) 1) "zipcode"
 2) "10021-3100"
```

```
127.0.0.1:6379> JSON.RESP k1
```

```
1) {
2) 1) "firstName"
 2) "John"
3) 1) "lastName"
 2) "Smith"
4) 1) "age"
 2) (integer) 27
5) 1) "weight"
 2) "135.25"
6) 1) "isAlive"
 2) true
7) 1) "address"
 2) 1) {
 2) 1) "street"
 2) "21 2nd Street"
 3) 1) "city"
 2) "New York"
 4) 1) "state"
 2) "NY"
 5) 1) "zipcode"
 2) "10021-3100"
8) 1) "phoneNumbers"
 2) 1) [
 2) 1) {
 2) 1) "type"
 2) "home"
 3) 1) "number"
 2) "212 555-1234"
 3) 1) {
 2) 1) "type"
 2) "office"
 3) 1) "number"
```



```
2) "555 555-4567"
9) 1) "children"
 2) 1) [
10) 1) "spouse"
 2) (nil)
```

## JSON.SET

在路徑上設定JSON值。

如果路徑呼叫物件成員：

- 如果父元素不存在，命令會傳回NONEXISTENT錯誤。
- 如果父元素存在，但不是物件，則命令會傳回 ERROR。
- 如果有父元素且為物件：
  - 如果沒有成員，只會在父物件是路徑中的最後一個子系時，將新成員附加至父物件。否則，命令會傳回NONEXISTENT錯誤。
  - 如果成員存在，其值將被JSON值取代。

如果路徑呼叫陣列索引：

- 如果父元素不存在，命令會傳回NONEXISTENT錯誤。
- 如果父元素存在，但不是陣列，則命令會傳回 ERROR。
- 如果父元素存在，但索引超出範圍，則命令會傳回OUTOFBOUNDARIES錯誤。
- 如果父元素存在且索引有效，則該元素將被新JSON值取代。

如果路徑呼叫物件或陣列，則該值（物件或陣列）將被新JSON值取代。

### 語法

```
JSON.SET <key> <path> <json> [NX | XX]
```

[NX | XX] 您可以在其中有 0 或 1 個 [NX | XX] 識別碼。

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS金鑰。
- 路徑（必要）– JSON 路徑。對於新金鑰，JSON路徑必須是根 "."。

- NX (選用) – 如果路徑是根，則只有在金鑰不存在時才設定值。也就是插入新文件。如果路徑不是根，請只在沒有路徑時才設定值。也就是將值插入文件中。
- XX (選用) – 如果路徑是根，則只有在金鑰存在時才設定值。也就是取代現有文件。如果路徑不是根，請只在有路徑時才設定值。也就是更新現有值。

## 傳回

- 成功時有簡單字串 'OK'。
- 如果不符合 NX 或 XX 條件，即為 null。

## 範例

### 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.SET k1 $.a.* '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"

127.0.0.1:6379> JSON.SET k2 . '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k2 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":[0,0,0,0,0]}"
```

### 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"c":{"a":1, "b":2}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k1 .c.a '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.SET k1 .e[-1] '0'
OK
127.0.0.1:6379> JSON.GET k1
```

```
"{\c\":{\a\":0,\b\":2},\e\":[1,2,3,4,0]}"
127.0.0.1:6379> JSON.SET k1 .e[5] '0'
(error) OUTOFBOUNDARIES Array index is out of bounds
```

## JSON.STRAPPEND

將字串附加至路徑的JSON字串。

### 語法

```
JSON.STRAPPEND <key> [path] <json_string>
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用） – JSON 路徑。如果未提供，則預設為根。
- json\_string（必要） – 字串的JSON表示法。請注意，必須引用JSON字串。例如："string example"。

### 傳回

如果路徑是增強型語法：

- 整數陣列代表字串在每個路徑的新長度。
- 如果路徑上的值不是字符串，則其對應的傳回值為 null。
- SYNTAXERR 如果輸入 json 引數不是有效的JSON字串，則會出現錯誤。
- 如果沒有路徑，會發生 NONEXISTENT 錯誤。

如果路徑是受限語法：

- 整數，字串的新長度。
- 如果選取多個字串值，該命令會傳回上次所更新字串的新長度。
- 如果路徑上的值不是字串，會發生 WRONGTYPE 錯誤。
- WRONGTYPE 如果輸入 json 引數不是有效的JSON字串，則會出現錯誤。
- 如果沒有路徑，會發生 NONEXISTENT 錯誤。

### 範例

## 增強型路徑語法：

```

127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
"b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a 'a'
1) (integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* 'a'
1) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* 'a'
1) (integer) 2
2) (nil)
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* 'a'
1) (integer) 2
2) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b 'a'
1) (integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* 'a'
1) (nil)
2) (integer) 2
3) (nil)

```

## 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
"b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a 'a'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* 'a'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* 'a'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* 'a'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b 'a'
(integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* 'a'
(integer) 2

```

## JSON.STRLEN

取得路徑中JSON字串值的長度。

### 語法

```
JSON.STRLEN <key> [path]
```

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用）– JSON 路徑。如果未提供，則預設為根。

### 傳回

如果路徑是增強型語法：

- 整數陣列代表每個路徑上的字串值長度。
- 如果值不是字串，其對應的傳回值為 null。
- 如果沒有文件索引鍵，則為 null。

如果路徑是受限語法：

- 整數，字串的長度。
- 如果選取多個字串值，該命令會傳回第一個字串的長度。
- 如果路徑上的值不是字串，會發生 WRONGTYPE 錯誤。
- 如果沒有路徑，會發生 NONEXISTENT 錯誤。
- 如果沒有文件索引鍵，則為 null。

### 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
```

```

1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
1) (integer) 1
2) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
1) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
1) (nil)
2) (integer) 1
3) (nil)

```

### 受限路徑語法：

```

127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
"b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 .a.a
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .a.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.b
(integer) 2
127.0.0.1:6379> JSON.STRLEN k1 .d.*
(integer) 1

```

## JSON.TOGGLE

在路徑上將布林值切換為 true 或 false。

### 語法

```
JSON.TOGGLE <key> [path]
```

- 金鑰（必要）– JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用）– JSON 路徑。如果未提供，則預設為根。

### 傳回

如果路徑是增強型語法：

- 整數陣列 (0 - false , 1 - true) , 代表每個路徑上產生的布林值。
- 如果值不是布林值 , 其對應的傳回值為 null。
- 如果沒有文件索引鍵 , 則為 NONEXISTENT。

如果路徑是受限語法：

- 字串 ("true" / "false") , 代表產生的布林值。
- 如果沒有文件索引鍵 , 則為 NONEXISTENT。
- 如果路徑上的值不是布林值 , 會發生 WRONGTYPE 錯誤。

## 範例

增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '{"a":true, "b":false, "c":1, "d":null, "e":"foo", "f":
[], "g":{}}'
OK
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 0
2) (integer) 1
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 1
2) (integer) 0
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
```

受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . true
```

```
OK
127.0.0.1:6379> JSON.TOGGLE k1
"false"
127.0.0.1:6379> JSON.TOGGLE k1
"true"

127.0.0.1:6379> JSON.SET k2 . '{"isAvailable": false}'
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

## JSON.TYPE

報告指定路徑上值的類型。

### 語法

```
JSON.TYPE <key> [path]
```

- 金鑰（必要） – JSON 文件類型的 Valkey 或 Redis OSS 金鑰。
- 路徑（選用） – JSON 路徑。如果未提供，則預設為根。

### 傳回

如果路徑是增強型語法：

- 字串陣列代表每個路徑上值的類型。該類型是 {"null", "boolean", "string", "number", "integer", "object" and "array"} 之一。
- 如果沒有路徑，其對應的傳回值為 null。
- 如果沒有文件索引鍵，則為空陣列。

如果路徑是受限語法：

- 字串，值的類型
- 如果沒有文件索引鍵，則為 null。



- 如果JSON路徑無效或不存在，則為 Null。

## 範例

### 增強型路徑語法：

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

### 受限路徑語法：

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"}, {"type":"office","number":"646
555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.TYPE k1
object
127.0.0.1:6379> JSON.TYPE k1 .children
array
127.0.0.1:6379> JSON.TYPE k1 .firstName
string
127.0.0.1:6379> JSON.TYPE k1 .age
integer
127.0.0.1:6379> JSON.TYPE k1 .weight
number
127.0.0.1:6379> JSON.TYPE k1 .isAlive
boolean
127.0.0.1:6379> JSON.TYPE k1 .spouse
null
```

# 標記您的 ElastiCache 資源

為了協助您管理叢集和其他 ElastiCache 資源，您可以以標籤的形式將自己的中繼資料指派給每個資源。標籤可讓您以不同方式分類 AWS 資源，例如依用途、擁有者或環境分類。當您有許多相同類型的資源時，這將會很有用，因為—您可以依據先前指派的標籤，快速識別特定的資源。本主題說明標籤並示範如何建立它們。

## Warning

根據最佳實務，建議您不要在標籤中包含敏感資料。

## 標籤基本概念

標籤是您指派給 AWS 資源的標籤。每個標籤皆包含由您定義的一個金鑰與一個選用值。標籤可讓您以不同方式分類 AWS 資源，例如依用途或擁有者分類。例如，您可以為帳戶的 ElastiCache 叢集定義一組標籤，協助您追蹤每個執行個體的擁有者和使用者群組。

我們建議您為每種資源類型建立符合您需求的標籤金鑰。使用一致的標籤金鑰組可讓您更輕鬆管理您的資源。您可以根據您新增的標籤搜尋和篩選資源。如需如何實作有效資源標記策略的詳細資訊，請參閱 [AWS 白皮書標記最佳實務](#)。

標籤對 沒有任何語義意義，ElastiCache 並嚴格解譯為一串字元。此外，標籤不會自動指派給您的資源。您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為 null。若您將與現有標籤具有相同鍵的標籤新增到該資源，則新值會覆寫舊值。如果您刪除資源，也會刪除任何該資源的標籤。此外，如果您為複寫群組新增或刪除標籤，也會為該複寫群組中的所有節點新增或移除其標籤。

您可以使用 AWS Management Console、AWS CLI 和 來使用標籤 ElastiCache API。

如果您使用的是 IAM，您可以控制 AWS 帳戶中哪些使用者具有建立、編輯或刪除標籤的許可。如需詳細資訊，請參閱 [資源層級許可](#)。

## 您可以標記的資源

您可以標記帳戶中已存在的大多數 ElastiCache 資源。下表列出支援標籤建立的資源。如果您使用的是 AWS Management Console，您可以使用標籤 [編輯器](#) 將標籤套用至資源。有些資源畫面可讓您在建立資源時指定資源的標籤；例如，具有 Name 索引鍵和您指定值的標籤。在大多數的案例中，主控台會立即在建立資源後套用標籤 (而非在資源建立過程時)。主控台可能會根據名稱標籤組織資源，但此標籤對 ElastiCache 服務沒有任何語義意義。

此外，有些資源建立動作可讓您在建立資源時指定資源的標籤。若標籤無法在資源建立時套用，我們會轉返資源建立程序。這可確保資源不是具有標籤建立，就是不會建立，因此無論何時都不會有不具有標籤的資源。藉由在建立時為資源建立標籤，您可以消除在資源建立後執行自訂標籤指令碼的必要。

如果您使用的是 Amazon ElastiCache API、AWS CLI、或 AWS SDK，您可以在相關 ElastiCache API 動作上使用 Tags 參數來套用標籤。這些類別為：


- CreateServerlessCache
- CreateCacheCluster
- CreateReplicationGroup
- CopyServerlessCacheSnapshot
- CopySnapshot
- CreateCacheParameterGroup
- CreateCacheSecurityGroup
- CreateCacheSubnetGroup
- CreateServerlessCacheSnapshot
- CreateSnapshot
- CreateUserGroup
- CreateUser
- PurchaseReservedCacheNodesOffering

下表說明可標記 ElastiCache 的資源，以及可使用 AWS CLI、ElastiCache API 或 建立時可標記的資源 AWS SDK。

#### 資源的標記支援 ElastiCache

支援標籤	支援在建立時標記
是	是
是	是

支援標籤	支援在建立時標記
是	是
是	是
是	是
是	是
是	是
是	是
是	是
是	是
是	是

 Note

您無法為全域資料存放區加上標籤。

您可以在IAM政策中將 ElastiCache API標籤型資源層級許可套用至支援建立時標記的動作，以對可在建立時標記資源的使用者和群組實作精細控制。您的資源從建立時便已獲得妥善的保護，標籤會立即套

用到您的資源。因此，控制資源使用情況的任何標籤式資源層級許可都會立即生效。您可以更準確的追蹤和報告您的資源。您可以強制新資源使用標籤，並控制哪些標籤金鑰和值會在您的資源上設定。

如需詳細資訊，請參閱[為資源加上標籤的範例](#)。

如需為資源加上標籤以便計費的詳細資訊，請參閱「[使用成本配置標籤監控成本](#)」。

## 標記快取和快照

以下規則適用於屬於請求作業一部分的標記程序：

- **CreateReplicationGroup:**
  - 如果 `--primary-cluster-id` 和 `--tags` 參數包含在請求中，就會將請求標籤新增至複寫群組，並傳播至複寫群組中的所有快取叢集。如果主快取叢集有現有的標籤，請求標籤會覆寫這些標籤，以便讓所有節點上的標籤保持一致。

如果沒有請求標籤，就會將主快取叢集標籤新增至複寫群組，並傳播至所有快取叢集。
  - 如果提供 `--snapshot-name` 或 `--serverless-cache-snapshot-name`：

如果請求中包含標籤，複寫群組只會使用這些標籤進行標記。如果請求中不含任何標籤，就會將快照標籤新增至複寫群組。
  - 如果提供 `--global-replication-group-id`：

如果請求中包含標籤，就會將這些請求標籤新增至複寫群組，並傳播至所有快取叢集。
- **CreateCacheCluster :**
  - 如果提供 `--replication-group-id`：

如果請求中包含標籤，快取叢集只會使用這些標籤進行標記。如果請求中不含任何標籤，快取叢集會繼承複寫群組標籤，而不是主快取叢集的標籤。
  - 如果提供 `--snapshot-name`：

如果請求中包含標籤，快取叢集只會使用這些標籤進行標記。如果請求中不含任何標籤，就會將快照標籤新增至快取叢集。
- **CreateServerlessCache :**
  - 如果請求中包含標籤，就只會將這些請求標籤新增至無伺服器快取。
- **CreateSnapshot :**
  - 如果提供 `--replication-group-id`：

如果請求中包含標籤，就只會將這些請求標籤新增至快照。如果請求中不含任何標籤，就會將複寫群組標籤新增至快照。

- 如果提供 `--cache-cluster-id` :

如果請求中包含標籤，就只會將這些請求標籤新增至快照。如果請求中不含任何標籤，就會將快取叢集標籤新增至快照。

- 針對自動快照：

標籤會從複寫群組標籤傳播。

- `CreateServerlessCacheSnapshot` :

- 如果請求中包含標籤，就只會將這些請求標籤新增至無伺服器快取快照。

- `CopySnapshot` :

- 如果請求中包含標籤，就只會將這些請求標籤新增至快照。如果請求中不含任何標籤，就會將來源快照標籤新增至複製的快照。

- `CopyServerlessCacheSnapshot` :

- 如果請求中包含標籤，就只會將這些請求標籤新增至無伺服器快取快照。

- `AddTagsToResource` 和 `RemoveTagsFromResource` :

- 將從複寫群組新增/移除標籤，並將動作傳播到複寫群組中的所有叢集。

#### Note

`AddTagsToResource` 和 `RemoveTagsFromResource`無法用於預設參數和安全群組。

- `IncreaseReplicaCount` 和 `ModifyReplicationGroupShardConfiguration` :

- 新增至複寫群組的所有新叢集都會套用與複寫群組相同的標籤。

## 標籤限制

以下基本限制適用於標籤：

- 每一資源最多標籤數 – 50
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。
- 金鑰長度上限 – UTF-8 中的 128 個 Unicode 字元。
- 最大值長度 – UTF-8 中的 256 Unicode 字元。

- 雖然 ElastiCache 允許其標籤中的任何字元，但其他服務可能具有限制性。跨服務允許的字元為：字母、數字和空格，以 UTF-8 表示，以及下列字元：+ - = . \_ : / @
- 標籤金鑰與值皆區分大小寫。
- 字aws:首會保留以供 AWS 使用。如果標籤具有此字首的標籤金鑰，則您無法編輯或刪除標籤的金鑰或值。具 aws: 字首的標籤，不算在受資源限制的標籤計數內。

您無法僅根據標籤終止、停止或刪除資源。您必須指定資源識別符。例如，若要刪除您套用稱為 DeleteMe 標籤金鑰的快照，您必須搭配快照的資源識別符 (例如 DeleteSnapshot) 使用 snap-1234567890abcdef0 動作。

如需可標記 ElastiCache 資源的詳細資訊，請參閱 [您可以標記的資源](#)。

## 為資源加上標籤的範例

- 使用標籤建立無伺服器快取。此範例使用 Memcached 作為引擎。

```
aws elasticache create-serverless-cache \
 --serverless-cache-name CacheName \
 --engine memcached \
 --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- 新增標籤至無伺服器快取

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \
 --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 將標籤新增到複寫群組。

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \
 --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 使用標籤建立快取叢集。

```
aws elasticache create-cache-cluster \
 --cluster-id testing-tags \
 --cluster-description cluster-test \
 --cache-subnet-group-name test \
 --cache-node-type cache.t2.micro \
 --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

```
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 使用標籤建立快取叢集。此範例使用 Redis 作為引擎。

```
aws elasticache create-cache-cluster \
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 建立包含標籤的無伺服器快照。此範例使用 Memcached 作為引擎。

```
aws elasticache create-serverless-cache-snapshot \
--serverless-cache-name testing-tags \
--serverless-cache-snapshot-name bkp-testing-tags-scs \
--tags Key="work",Value="foo"
```

- 使用標籤建立快照。

快照目前僅適用於 Redis。在此情況下，如果您對請求新增了標籤，即使複寫群組包含標籤，快照也只會接收請求標籤。

```
aws elasticache create-snapshot \
--replication-group-id testing-tags \
--snapshot-name bkp-testing-tags-rg \
--tags Key="work",Value="foo"
```

## 標籤型存取控制政策範例

1. 僅當叢集具有標籤 Project= 時，才允許對叢集AddTagsToResource執行動作XYZ。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
```



```

 "Action": "elasticache:AddTagsToResource",
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Project": "XYZ"
 }
 }
 }
]
}

```

2. 如果複寫群組包含 Project 和 Service 標籤，且索引鍵與 Project 和 Service 不同，便允許該複寫群組的 RemoveTagsFromResource 動作。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "elasticache:RemoveTagsFromResource",
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Service": "Elasticache",
 "aws:ResourceTag/Project": "XYZ"
 },
 "ForAnyValue:StringNotEqualsIgnoreCase": {
 "aws:TagKeys": [
 "Project",
 "Service"
]
 }
 }
 }
]
}

```

3. 只有在標籤不是 Project 和 Service 時，才允許對任何資源執行 AddTagsToResource。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "elasticache:AddTagsToResource",
 "Resource": [
 "arn:aws:elasticache:*:*:*:*"
],
 "Condition": {
 "ForAnyValue:StringNotEqualsIgnoreCase": {
 "aws:TagKeys": [
 "Service",
 "Project"
]
 }
 }
 }
]
}
```

4. 如果請求具有 Tag Project=Foo，則拒絕 CreateReplicationGroup 動作。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": "elasticache:CreateReplicationGroup",
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Project": "Foo"
 }
 }
 }
]
}
```

5. 如果來源快照具有 Project=XYZ 標籤，且請求標籤為 Service=Elasticache，則會拒絕 CopySnapshot 動作。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": "elasticache:CopySnapshot",
 "Resource": [
 "arn:aws:elasticache:*:*:snapshot:*"
],
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Project": "XYZ",
 "aws:RequestTag/Service": "Elasticache"
 }
 }
 }
]
}
```

6. 如果請求標籤 CreateCacheCluster 遺失或不等於 Project、Dev 或 QA，則拒絕 Prod 動作。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*",
 "arn:aws:elasticache:*:*:securitygroup:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "elasticache:CreateCacheCluster"
]
 }
]
}
```

```
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "Null": {
 "aws:RequestTag/Project": "true"
 }
 }
},
{
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:AddTagsToResource"
],
 "Resource": "arn:aws:elasticache:*:*:cluster:*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Project": [
 "Dev",
 "Prod",
 "QA"
]
 }
 }
}
]
```

如需條件索引鍵的相關資訊，請參閱 [使用條件索引鍵](#)。

## 使用成本配置標籤監控成本

當您將成本分配標籤新增至 Amazon 中的資源時 ElastiCache，您可以透過將發票上的費用依資源標籤值分組來追蹤成本。

ElastiCache 成本分配標籤是您定義 ElastiCache 資源並與之建立關聯的鍵值對。索引鍵與值皆會區分大小寫。您可以使用標籤索引鍵來定義類別，而標籤值可為該類別中的某個項目。例如，您可以定義標籤索引鍵 `CostCenter`，且標籤值為 `10010`，指出資源是指派給 `10010` 成本中心。您也可利用 `Environment` 之類的索引鍵，和 `test` 或 `production` 之類的值，以使用標籤來指定用於測試或生產的資源。我們建議您使用一組一致的標籤索引鍵，讓您更輕鬆地追蹤與資源關聯的成本。

使用成本分配標籤來整理 AWS 帳單，以反映您自己的成本結構。若要執行此操作，請註冊以取得包含標籤索引鍵值 AWS 的帳戶帳單。接著，若要查看合併資源的成本，請根據具有相同標籤鍵值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。

您也可以結合標籤來以更高的細節層次追蹤成本。例如，若要按區域追蹤您的服務成本，您可以使用標籤索引鍵 Service 和 Region。在某個資源上，您會有值 ElastiCache 和 Asia Pacific (Singapore)，而在另一個資源上，則有值 ElastiCache 和 Europe (Frankfurt)。然後，您可以看到依區域劃分的總 ElastiCache 成本。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的[使用成本分配標籤](#)。

您可以將 ElastiCache 成本分配標籤新增至 ElastiCache 自行設計的叢集。新增、列出、修改、複製或移除標籤時，該操作只會套用至指定的叢集。

### ElastiCache 成本分配標籤的特徵

- 成本分配標籤會套用至在 CLI 和 API 操作中指定為 ElastiCache 的資源 ARN。資源類型會是「叢集」。

範例 ARN：`arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

範例 ARN：`arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- 標籤金鑰是標籤必要的名稱。索引鍵的字串值的長度可以是 1 到 128 個 Unicode 字元，不可在前面加上 `aws:`。此字串只能包含一組 Unicode 字母、數字、空格、底線 (`_`)、句點 (`.`)、冒號 (`:`)、反斜線 (`\`)、等號 (`=`)、加號 (`+`)、連字號 (`-`) 或 @ 符號 (`@`)。
- 標籤值為標籤的選用值。值的字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上 `aws:`。此字串只能包含一組 Unicode 字母、數字、空格、底線 (`_`)、句點 (`.`)、冒號 (`:`)、反斜線 (`\`)、等號 (`=`)、加號 (`+`)、連字號 (`-`) 或 @ 符號 (`@`)。
- ElastiCache 資源最多可以有 50 個標籤。
- 標籤組中的值不必是唯一的。例如，您可以有一個標籤組，其中的索引鍵 Service 和 Application 都有值 ElastiCache。

AWS 不會將任何語義意義套用至您的標籤。標籤會嚴格解譯為字元字串。AWS 不會自動在任何 ElastiCache 資源上設定任何標籤。

## 使用 管理您的成本分配標籤 AWS CLI

您可以使用 AWS CLI 來新增、修改或移除成本分配標籤。

成本分配標籤會套用至 ElastiCache 叢集。要使用 ARN ( Amazon Resource Name ) 指定要標記的叢集。

範例 ARN : `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### 主題

- [使用 列出標籤 AWS CLI](#)
- [使用 新增標籤 AWS CLI](#)
- [使用 修改標籤 AWS CLI](#)
- [使用 移除標籤 AWS CLI](#)

## 使用 列出標籤 AWS CLI

您可以使用 AWS CLI 來列出現有 ElastiCache 資源上的標籤，方法是使用 [list-tags-for-resource](#) 操作。

下列程式碼使用 AWS CLI 列出 us-west-2 my-cluster 區域中 Memcached 叢集上的標籤。

若為 Linux、macOS 或 Unix :

```
aws elasticache list-tags-for-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

針對 Windows :

```
aws elasticache list-tags-for-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

下列程式碼使用 AWS CLI 列出 us-west-2 區域中 my-cluster 叢集中 Valkey 或 Redis OSS 節點 my-cluster-001 上的標籤。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-tags-for-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

針對 Windows：

```
aws elasticache list-tags-for-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

此操作的輸出看起來應該類似以下，這是資源上所有標籤的清單。

```
{
 "TagList": [
 {
 "Value": "10110",
 "Key": "CostCenter"
 },
 {
 "Value": "EC2",
 "Key": "Service"
 }
]
}
```

如果資源上沒有標籤，則輸出將為空的 TagList。

```
{
 "TagList": []
}
```

如需詳細資訊，請參閱 AWS CLI 的 ElastiCache [list-tags-for-resource](#)。

## 使用 新增標籤 AWS CLI

您可以使用 AWS CLI 將標籤新增至現有 ElastiCache 資源，方法是使用 [add-tags-to-resource](#) CLI 操作。如果標籤索引鍵不存在於資源上，則索引鍵和值會新增至資源。如果索引鍵已存在於資源上，則與該索引鍵相關聯的值會更新為新的值。

下列程式碼使用 AWS CLI 將索引鍵 Service 和 Region 與值 elasticache 和 us-west-2 分別新增至 us-west-2 區域中 my-cluster-001 叢集中的節點。my-cluster

## Memcached

若為 Linux、macOS 或 Unix :

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tags Key=Service,Value=elasticache \
 Key=Region,Value=us-west-2
```

針對 Windows :

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tags Key=Service,Value=elasticache ^
 Key=Region,Value=us-west-2
```

## Redis

若為 Linux、macOS 或 Unix :

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tags Key=Service,Value=elasticache \
 Key=Region,Value=us-west-2
```

針對 Windows :

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tags Key=Service,Value=elasticache ^
 Key=Region,Value=us-west-2
```

此操作的輸出看起來應該類似以下，這是在操作後資源上所有標籤的清單。

```
{
 "TagList": [
 {
 "Value": "elasticache",
 "Key": "Service"
 },
 {
```



```
 "Value": "us-west-2",
 "Key": "Region"
 }
]
}
```

如需詳細資訊，請參閱 AWS CLI 的 ElastiCache [add-tags-to-resource](#)。

當您使用 操作建立新叢集時 AWS CLI ，您也可以使用 將標籤新增至叢集 [create-cache-cluster](#)。使用 ElastiCache 管理主控台建立叢集時，您無法新增標籤。建立叢集之後，您可以接著使用主控台來將標籤新增至叢集。

## 使用 修改標籤 AWS CLI

您可以使用 AWS CLI 來修改 ElastiCache 叢集上的標籤。

若要修改標籤：

- 使用 [add-tags-to-resource](#) 新增標籤和值，或變更與現有標籤相關聯的值。
- 使用 [remove-tags-from-resource](#) 從資源中移除指定的標籤。

這兩項操作的輸出會是指定叢集上標籤和其值的清單。

## 使用 移除標籤 AWS CLI

您可以使用 `aws elasticache remove-tags-from-resource`，從現有 ElastiCache ( Memcached ) 叢集 AWS CLI 中移除標籤，方法是使用 [remove-tags-from-resource](#) 操作。

對於 Memcached，下列程式碼使用 AWS CLI 來Region從 us-west-2 區域中叢集my-cluster-001中的節點移除具有金鑰 Service 和 my-cluster 的標籤。

若為 Linux、macOS 或 Unix：

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tag-keys PM Service
```

針對 Windows：

```
aws elasticache remove-tags-from-resource ^
```

```
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
--tag-keys PM Service
```

對於 Redis，下列程式碼使用 AWS CLI 來 Region 從 us-west-2 區域中叢集 my-cluster-001 中的節點移除具有金鑰 Service 和 my-cluster 的標籤。

若為 Linux、macOS 或 Unix：

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tag-keys PM Service
```

針對 Windows：

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tag-keys PM Service
```

此操作的輸出看起來應該類似以下，這是在操作後資源上所有標籤的清單。

```
{
 "TagList": []
}
```

如需詳細資訊，請參閱 AWS CLI 的 ElastiCache [remove-tags-from-resource](#)。

## 使用 管理您的成本分配標籤 ElastiCache API

您可以使用 新增 ElastiCache API、修改或移除成本分配標籤。

成本分配標籤會 ElastiCache 套用至 Memcached 叢集。要使用 ARN ( Amazon Resource Name ) 指定要標記的叢集。

範例 ARN : arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster

### 主題

- [使用 列出標籤 ElastiCache API](#)
- [使用 新增標籤 ElastiCache API](#)

- [使用 修改標籤 ElastiCache API](#)
- [使用 移除標籤 ElastiCache API](#)

## 使用 列出標籤 ElastiCache API

您可以使用 ElastiCache API來列出現有資源上的標籤，方法是使用 [ListTagsForResource](#) 操作。

對於 Memcached，下列程式碼使用 ElastiCache API來列出 us-west-2 my-cluster 區域中資源上的標籤。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListTagsForResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

對於 Redis，下列程式碼使用 ElastiCache API來列出 us-west-2 my-cluster-001 區域中資源上的標籤。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListTagsForResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

## 使用 新增標籤 ElastiCache API

您可以使用 ElastiCache API將標籤新增至現有 ElastiCache 叢集，方法是使用 [AddTagsToResource](#) 操作。如果標籤索引鍵不存在於資源上，則索引鍵和值會新增至資源。如果索引鍵已存在於資源上，則與該索引鍵相關聯的值會更新為新的值。

下列程式碼使用 ElastiCache API 分別新增 金鑰Service和 Region 與 值elasticacheus-west-2。對於 Memcached，這會套用至資源 my-cluster。對於 Redis，這會套用至 us-west-2 my-cluster-001 區域中的資源。

## Memcached

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

## Redis

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

如需詳細資訊，請參閱 [AddTagsToResource](#) 在 Amazon ElastiCache API 參考 中。

## 使用 修改標籤 ElastiCache API

您可以使用 ElastiCache API 來修改 ElastiCache 叢集上的標籤。

若要修改標籤的值：

- 使用 [AddTagsToResource](#) 操作，以新增標籤和值，或變更現有標籤的值。
- 使用 [RemoveTagsFromResource](#) 從資源中移除標籤。

這兩項操作的輸出會是指定資源上標籤和其值的清單。

使用 [RemoveTagsFromResource](#) 從資源中移除標籤。

## 使用 移除標籤 ElastiCache API

您可以使用 ElastiCache API 來從現有 ElastiCache ( Memcached ) 叢集中移除標籤，方法是使用 [RemoveTagsFromResource](#) 操作。

下列程式碼使用 ElastiCache API，Region 從 us-west-2 my-cluster 區域中叢集 my-cluster-001 中的節點移除具有金鑰 Service 和 的標籤。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=RemoveTagsFromResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&TagKeys.member.1=Service
&TagKeys.member.2=Region
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

## 使用 Amazon ElastiCache Well-Architected Lens

本節說明 Amazon ElastiCache Well-Architected Lens，這是一系列設計原則和指南，用於設計架構良好的 ElastiCache 工作負載。

- ElastiCache Lens 是 [AWS Well-Architected Framework](#) 的附加元件。
- 每個支柱都有一組問題，可協助開始有關 ElastiCache 架構的討論。
  - 每個問題都有一些引導練習，且配有分數以便製作報告。
    - 必要 - 進入 prod 之前的必要前提 (缺少代表高風險)
    - 最佳 - 客戶能達到的最佳狀態
    - 良好 - 我們建議客戶達到的狀態 (缺少代表中等風險)
- Well-Architected 的術語
  - [元件](#) – 一起根據要求交付的程式碼、組態 AWS 和資源。元件會與其他元件互動，而且經常等同於微型服務架構中的服務。
  - [工作負載](#) - 一組共同提供商業價值的元件。工作負載的範例包括行銷網站、電子商務網站、行動應用程式後端系統、分析平台等。

**Note**

本指南尚未更新，以包含無 ElastiCache 伺服器快取和新 Valkey 引擎的相關資訊。

**主題**

- [Amazon ElastiCache Well-Architected Lens 卓越營運支柱](#)
- [Amazon ElastiCache Well-Architected Lens 安全支柱](#)
- [Amazon ElastiCache Well-Architected Lens 可靠性支柱](#)
- [Amazon ElastiCache Well-Architected Lens 效能效率支柱](#)
- [Amazon ElastiCache Well-Architected Lens 成本最佳化支柱](#)

## Amazon ElastiCache Well-Architected Lens 卓越營運支柱

卓越營運支柱著重於執行和監控系統，以提供商業價值並持續改善流程和程序。重要主題包括自動化變更、回應事件，以及定義管理日常作業的標準。

**主題**

- [OE 1：您如何了解和回應 ElastiCache 叢集觸發的警示和事件？](#)
- [OE 2：您何時以及如何擴展現有 ElastiCache 叢集？](#)
- [OE 3：如何管理 ElastiCache 叢集資源和維護叢集 up-to-date？](#)
- [OE 4：如何管理用戶端與 ElastiCache 叢集的連線？](#)
- [OE 5：如何部署工作負載的 ElastiCache 元件？](#)
- [OE 6：如何規劃故障因應措施及減少故障？](#)
- [OE 7：如何對 Valkey 或 Redis OSS引擎事件進行疑難排解？](#)

### OE 1：您如何了解和回應 ElastiCache 叢集觸發的警示和事件？

**問題層級簡介：**當您操作ElastiCache 叢集時，您可以選擇在特定事件發生時接收通知和提醒。依預設 ElastiCache，會記錄與資源相關的**事件**，例如容錯移轉、節點取代、擴展操作、排程維護等。每個事件都包含日期與時間、來源名稱與來源類型，以及說明。

**問題難易度優點：**只要能夠了解並管理觸發叢集產生警示的事件背後的基本原因，就能更有效地操作並適當回應事件。

- **【必要】** 在 ElastiCache 主控台ElastiCache 上檢閱 產生的事件（選取您的區域之後），或使用 [Amazon Command Line Interface](#)（AWS CLI）[describe-events](#) 命令和 [ElastiCache API](#)。設定 ElastiCache 使用 Amazon Simple Notification Service（Amazon）傳送重要叢集事件的通知 SNS。將 Amazon SNS與叢集搭配使用，可讓您以程式設計方式對 ElastiCache 事件採取動作。
- 事件分成兩大類：目前事件和排定事件。目前事件的清單包括：資源建立和刪除、擴展操作、容錯移轉、節點重新啟動、快照建立、叢集參數修改、CA 憑證更新、失敗事件（叢集佈建失敗 - VPC 或 ENI-、擴展失敗 - ENI- 和快照失敗）。排定事件清單包括：排定在維護時段進行更換的節點，以及重新排定的節點更換作業。
- 雖然您不需要立即回應其中一些事件，但務必先查看所有失敗事件：
  - ElastiCache:AddCacheNodeFailed
  - ElastiCache:CacheClusterProvisioningFailed
  - ElastiCache:CacheClusterScalingFailed
  - ElastiCache:CacheNodesRebooted
  - ElastiCache : SnapshotFailed（OSS僅限 Valkey 或 Redis）
- [資源]：
  - [管理 ElastiCache Amazon SNS通知](#)
  - [事件通知和 Amazon SNS](#)
- **【最佳】** 若要自動回應事件，請利用 AWS 和 SNS Lambda Functions 等產品和服務功能。遵循最佳實務進行小量、頻繁、可恢復的變更，以程式碼形式隨著時間讓您的操作演進。您應該使用 Amazon CloudWatch 指標來監控叢集。

**【資源】**：[使用 AWS Lambda、Amazon Route 53 和 Amazon 監控 ElastiCache（Redis OSS）（停用叢集模式）](#) 僅供讀取複本端點SNS使用 Lambda 和 的使用案例SNS。

## OE 2：您何時以及如何擴展現有 ElastiCache 叢集？

**問題層級簡介：**正確調整ElastiCache 叢集大小是一種平衡行為，每次基礎工作負載類型發生變更時都需要評估。您的目標是讓您的工作負載在適當規模的環境下運作。

**問題難易度優點：**資源過度利用可能會導致延遲增加且整體效能降低。另一方面來說，使用不足可能會導致資源過度佈建，而以非最佳成本最佳化。只要適當調整環境的規模，您就能在效能效率與成本最佳化之間取得平衡。若要修復資源的過度或不足使用，ElastiCache 可以兩維擴展。您可以增加或減少節點容量來垂直擴展。您也可以新增和移除節點來水平擴展。

- 主要節點上的 **【必要】** CPU和網路過度使用應透過將讀取操作卸載和重新導向至複本節點來解決。使用複本節點進行讀取操作，可降低主節點的使用率。這可以透過連線至已停用叢集模式的ElastiCache 讀取器端點，或使用已啟用叢集模式的READONLY命令，在您的 Valkey 或 Redis OSS 用戶端程式庫中設定。

[資源]：

- [在中尋找連線端點 ElastiCache](#)
- [適當調整叢集規模](#)
- [READONLY 命令](#)
- **【必要】** 監控關鍵叢集資源的使用率CPU，例如、記憶體和網路。需追蹤這些特定叢集資源的使用率，才能在擴展和擴展操作類型方面做出明智的決策。對於停用的 ElastiCache ( Redis OSS ) 叢集模式，主要節點和複本節點可以垂直擴展。複本節點也可以從 0 到 5 個節點水平擴展。若叢集模式已啟用，叢集的每個碎片內情況也相同。此外，您可以增加或減少碎片的數量。

[資源]：

- [使用 Amazon 使用 ElastiCache \( RedisOSS \) 監控最佳實務 CloudWatch](#)
- [擴展 ElastiCache \( RedisOSS \) 叢集](#)
- [Memcached 叢集的 ElastiCache擴展](#)
- [最佳] 隨著時間監控趨勢可幫助您偵測到在特定時間點監控時，可能無法察覺的工作負載變更。若要偵測更長期的趨勢，請使用 CloudWatch指標來掃描更長的時間範圍。觀察長期 CloudWatch 指標所學到的知識，應該會通知您有關叢集資源使用率的預測。CloudWatch 資料點和指標最多可使用 455 天。

[資源]：

- [使用 CloudWatch指標監控 ElastiCache \( RedisOSS \)](#)
- [使用 CloudWatch 指標監控 Memcached](#)
- [使用 Amazon 使用 ElastiCache \( Redis OSS \) 監控最佳實務 CloudWatch](#)
- **【最佳】** 如果使用 範本建立 ElastiCache 資源CloudFormation ，最佳實務是使用 CloudFormation 範本執行變更，以保持操作一致性，並避免未受管理的組態變更和堆疊偏離。

[資源]：

- [ElastiCache 資源類型參考 CloudFormation](#)
- **【最佳】** 使用叢集操作資料自動化擴展操作，並在 中定義閾值 CloudWatch 以設定警示。使用 CloudWatch Events and Simple Notification Service ( SNS ) 來觸發 Lambda 函數，並執行 ElastiCache API以自動擴展叢集。例如，在 EngineCPUUtilization 指標經過長時間之後達到



80% 時，將碎片新增至叢集。另一個選項是使用 DatabaseMemoryUsedPercentages 作為記憶體型閾值。

[資源]：

- [使用 Amazon CloudWatch 警示](#)
- [什麼是 Amazon CloudWatch 事件？](#)
- [AWS Lambda 搭配 Amazon Simple Notification Service 使用](#)
- [ElastiCache API 參考](#)

### OE 3：如何管理 ElastiCache 叢集資源和維護叢集 up-to-date？

問題層級簡介：大規模操作時，您必須能夠精確找出所有 ElastiCache 資源。推出新的應用程式功能時，您需要在所有 ElastiCache 環境類型之間建立叢集版本對稱性：開發、測試和生產。資源屬性可讓您針對不同的操作目標分隔環境，例如在推出新功能和啟用新的安全機制時。

問題難易度優點：將開發、測試和生產環境加以分隔，是最佳操作實務。在整個環境中利用充分了解並妥善記載的程序對叢集和節點套用最新的軟體修補程式，同樣也是最佳實務。利用原生 ElastiCache 功能，您的工程團隊可以專注於實現業務目標，而不是 ElastiCache 維護。

- **【最佳】** 在可用的最新引擎版本上執行，並在自助服務更新可用時立即套用。在您指定的叢集維護時段期間 ElastiCache 自動更新其基礎基礎設施。不過，叢集中執行的節點則會透過自助式更新進行更新。這些更新有兩種類型：安全修補程式或次要軟體更新。您務必了解修補程式類型的差異，以及套用的時機。

[資源]：

- [Amazon 中的自助式更新 ElastiCache](#)
- [Amazon ElastiCache Managed Maintenance and Service Update 說明頁面](#)
- **【最佳】** 使用標籤組織 ElastiCache 資源。在複寫群組上使用標籤，而非在個別節點上使用。您可以設定在查詢資源時顯示的標籤，也可以使用標籤來執行搜尋和套用篩選器。您可使用資源群組輕鬆建立和維護擁有共同標籤集的資源集合。

[資源]：

- [標記最佳實務](#)
- [ElastiCache 資源類型參考 CloudFormation](#)
- [參數群組](#)

## OE 4：如何管理用戶端與 ElastiCache 叢集的連線？

**問題層級簡介：**大規模操作時，您需要了解用戶端如何與 ElastiCache 叢集連線，以管理您的應用程式操作方面（例如回應時間）。

**問題難易度優點：**選擇最合適的連線機制，可確保應用程式不會因為連線錯誤（例如逾時）而中斷連線。

- [必要] 將讀取與寫入操作分開，並連線至複本節點來執行讀取操作。但是，請注意，當您將寫入與讀取區分開時，由於 Valkey 和 Redis OSS 複寫的非同步性質，在寫入金鑰之後，您將失去立即讀取金鑰的能力。該WAIT命令可以利用來改善真實世界的資料安全，並強制複本在回應用戶端之前確認寫入，且整體效能成本為。使用複本節點進行讀取操作時，可以使用停用叢集模式的 ElastiCache 讀取器端點，在 ElastiCache（Redis OSS）用戶端程式庫中設定複本節點。對於啟用的叢集模式，請使用 ElastiCache（Redis OSS）READONLY命令。對於許多 ElastiCache（Redis OSS）用戶端程式庫，ElastiCache（Redis OSS）READONLY 預設會實作或透過組態設定實作。

[資源]：

- [在中尋找連線端點 ElastiCache](#)
- [READONLY](#)
- [必要] 使用連線集區。建立TCP連線在用戶端和伺服器端都有CPU時間成本，而集區可讓您重複使用TCP連線。

為了減少連線額外負荷，建議您使用連線集區。有了連線集區，您的應用程式就可以「隨意」重複使用和釋出連線，而不會因建立連線而產生成本。您可以透過 ElastiCache（Redis OSS）用戶端程式庫（如果支援）實作連線集區，並搭配適用於應用程式環境的架構，或從頭開始建置。

- [最佳] 確實將用戶端的通訊端逾時設定為至少 1 秒（相較於數個用戶端中的一般預設值「無」）。
  - 若設定的逾時值太低，可能導致伺服器負載較高時發生逾時。若設定的值太高，則可能導致您的應用程式花費很長的時間來偵測連線問題。
  - 藉由在用戶端應用程式中實作連線集區來控制新連線的數量。這可減少開啟和關閉連線所需的延遲和CPU使用率，並在叢集上TLS啟用時執行TLS交握。

**【資源】：**[設定 ElastiCache（Redis OSS）以獲得更高的可用性](#)

- [良好] 使用管道傳輸（您的使用案例允許的話）可大幅提高效能。
  - 透過管道，您可以減少應用程式用戶端與叢集之間的往返時間（RTT），即使用戶端尚未讀取先前的回應，也可以處理新的請求。
  - 使用管道傳輸可一次將多個命令傳送至伺服器，而不需等待回覆/確認。管道傳輸的缺點在於，當您最後大量截取所有回應時，可能已有錯誤發生，但您直到最後才察覺到。

- 當傳回錯誤而忽略錯誤請求時，實作方法來重試請求。

[資源]：[管道傳輸](#)

## OE 5：如何部署工作負載的 ElastiCache 元件？

問題層級簡介：ElastiCache 環境可以透過 AWS 主控台手動部署，或透過 APIs、CLI、工具組等以程式設計方式部署。卓越運作最佳實務建議您，盡可能透過程式碼自動化部署。此外，ElastiCache 叢集可以依工作負載隔離，也可以為了成本最佳化目的而合併。

問題層級優點：選擇最適合您 ElastiCache 環境的部署機制，可隨著時間改善卓越營運。建議您盡可能以程式碼形式執行操作，以便盡量減少人為錯誤並增加事件的重複能力、彈性和回應時間。

透過了解工作負載隔離需求，您可以選擇每個工作負載都有專用 ElastiCache 環境，或將多個工作負載合併為單一叢集，或其組合。了解當中的權衡，有助於在卓越運作和成本最佳化之間取得平衡

- **【必要】** 了解 可用的部署選項 ElastiCache，並盡可能將這些程序自動化。自動化的可能途徑包括 CloudFormation、AWS CLI/ SDK和 APIs。

[資源]：

- [Amazon ElastiCache 資源類型參考](#)
- [elasticache](#)
- [Amazon ElastiCache API 參考](#)
- [必要] 為所有工作負載決定所需的叢集隔離層級。
  - [最佳]：高度隔離 - 工作負載對叢集的對應為 1:1。允許對每個工作負載 ElastiCache 的資源存取、調整大小、擴展和管理進行最精細的控制。
  - [較佳]：中度隔離 - 依用途採 M:1 隔離，但可能跨多個工作負載共用 (例如，一個叢集專門用來快取工作負載，而另一個叢集專門用來傳訊)。
  - [良好]：低度隔離 - 所有用途均採 M:1 隔離，完全共用。建議用於可接受共用存取的工作負載。

## OE 6：如何規劃故障因應措施及減少故障？

問題層級簡介：卓越營運包括透過執行定期的 "mortem 前期" 練習來預測失敗，以識別潛在的失敗來源，以便移除或緩解失敗。ElastiCache 提供容錯移轉，API 允許模擬節點失敗事件用於測試目的。

問題難易度優點：透過提前測試故障情況，您就可以了解它們如何影響您的工作負載。這樣就能安全測試回應程序及其有效性，並且讓您的團隊熟悉其執行過程。

**【必要】** 在開發/測試帳戶中定期執行容錯移轉測試。 [TestFailover](#)

## OE 7：如何對 Valkey 或 Redis OSS引擎事件進行疑難排解？

**問題層級簡介：**卓越營運需要能夠調查服務層級和引擎層級資訊，以分析叢集的運作狀態和狀態。ElastiCache 可以將 Valkey 或 Redis OSS引擎日誌傳送至 Amazon CloudWatch 和 Amazon Kinesis Data Firehose

**問題層級優點：**在 ElastiCache 叢集上啟用 Valkey 或 Redis OSS引擎日誌，可讓您深入了解影響叢集運作狀態和效能的事件。Valkey 或 Redis OSS引擎日誌會直接提供無法透過 ElastiCache 事件機制取得的引擎資料。透過仔細觀察 ElastiCache 事件（請參閱先前的 OE-1）和引擎日誌，從ElastiCache 服務角度和引擎角度進行故障診斷時，可以判斷事件順序。

- **【必要】** 確保已啟用 Redis OSS引擎記錄功能，該功能自 ElastiCache（Redis OSS）6.2 及更新版本起提供。此功能可在建立叢集期間啟用，或是在建立後，藉由修改叢集來啟用。
- 判斷 Amazon CloudWatch Logs 或 Amazon Kinesis Data Firehose 是否為 Redis OSS引擎日誌的適當目標。
- 在 CloudWatch 或 Kinesis Data Firehose 中選取適當的目標日誌，以保留日誌。如果您有多個叢集，請考慮讓每個叢集使用不同的目標日誌，因為這樣做有助於在故障診斷時隔離資料。

[資源]：

- 日誌傳送：[日誌傳送](#)
- 記錄目的地：[Amazon CloudWatch Logs](#)
- Amazon CloudWatch Logs 簡介：[什麼是 Amazon CloudWatch Logs？](#)
- Amazon Kinesis Data Firehose 簡介：[什麼是 Amazon Kinesis Data Firehose？](#)
- **【最佳】** 如果使用 Amazon CloudWatch Logs，請考慮利用 Amazon CloudWatch Logs Insights 查詢 Valkey 或 Redis OSS引擎日誌以取得重要資訊。

例如，針對包含 Valkey 或 Redis OSS引擎日誌的 CloudWatch 日誌群組建立查詢，這些日誌將傳回具有「WARNING」LogLevel 的事件，例如：

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

**【資源】**：[使用 Logs Insights 分析 CloudWatch 日誌資料](#)

## Amazon ElastiCache Well-Architected Lens 安全支柱

安全支柱著重於保護資訊和系統。重要主題包括資料的機密性和完整性、識別和管理誰可以透過權限為主的管理做什麼、保護系統，以及建立控制項來偵測安全事件。

### 主題

- [SEC 1：您在控制授權存取ElastiCache 資料方面採取了哪些步驟？](#)
- [SEC 2：您的應用程式是否需要對 ElastiCache超過 和超過 聯網型控制項的額外授權？](#)
- [SEC 3：命令是否會不慎執行，導致資料遺失或失敗的風險？](#)
- [SEC 4：如何透過 確保靜態資料加密 ElastiCache](#)
- [SEC 5：如何使用 加密傳輸中資料 ElastiCache？](#)
- [SEC 6：如何限制控制平面資源的存取？](#)
- [SEC 7：如何偵測和回應安全事件？](#)

### SEC 1：您在控制授權存取ElastiCache 資料方面採取了哪些步驟？

問題層級簡介：所有 ElastiCache叢集的設計都是從、無VPC伺服器函數（AWS Lambda）或容器（Amazon Elastic Container Service）中的 Amazon Elastic Compute Cloud 執行個體存取。最常見的案例是從相同 Amazon Virtual Private Cloud（Amazon Virtual Private Cloud 內的 Amazon Elastic Compute Cloud 執行個體存取 ElastiCache 叢集。您必須先授權 Amazon EC2執行個體存取叢集，才能從 Amazon EC2執行個體連線至叢集。若要存取在 中執行的 ElastiCache 叢集VPC，則必須將網路輸入授予叢集。

問題層級優點：網路傳入叢集是透過VPC安全群組控制。安全群組可做為 Amazon EC2執行個體的虛擬防火牆，以控制傳入和傳出流量。傳入規則會控制傳入至您的執行個體的流量，以及傳出規則會控制從您的執行個體傳出的流量。如果是 ElastiCache，在啟動叢集時，需要關聯安全群組。這樣可確保組成叢集的所有節點都已設有傳入和傳出流量規則。此外，ElastiCache 設定為僅在私有子網路上部署，以便只能透過 VPC的私有網路從 存取。

- [必要] 與叢集相關聯的安全群組會控制叢集的網路輸入和存取。根據預設，安全群組不會定義任何傳入規則，因此不會傳入 的路徑 ElastiCache。若要啟用此功能，請在安全群組上設定傳入規則，指定來源 IP 地址/範圍、TCP輸入流量和 ElastiCache 叢集的連接埠（例如 ElastiCache（Redis OSS）的預設連接埠 6379）。雖然可以允許非常廣泛的輸入來源，就像 VPC（0.0.0.0/0）中的所有資源一樣，但建議盡可能精細地定義傳入規則，例如僅授權對在特定安全群組關聯之 Amazon EC2執行個體上執行的 Valkey 或 Redis OSS用戶端的傳入存取。

[資源]：



- [子網路和子網路群組](#)
- [存取您的叢集或複寫群組](#)
- [使用安全群組控制前往資源的流量](#)
- [適用於 Linux 執行個體的 Amazon Elastic Compute Cloud 安全群組](#)
- **【必要】** AWS Identity and Access Management 政策可指派給允許其存取 ElastiCache 資料的 AWS Lambda 函數。若要啟用此功能，請使用 `AWSLambdaVPCLambdaAccessExecutionRole` 許可建立 IAM 執行角色，然後將角色指派給 AWS Lambda 函數。

**【資源】**：在 Amazon ElastiCache 中設定 Lambda 函數以存取 Amazon VPC：[教學課程：設定 Lambda ElastiCache 函數以存取 Amazon VPC](#)

## SEC 2：您的應用程式是否需要對 ElastiCache 超過 和超過 聯網型控制項的額外授權？

**問題層級簡介**：在需要限制或控制個別用戶端層級對 ElastiCache (Redis OSS) 叢集的存取的情況下，建議透過 ElastiCache (Redis OSS) AUTH 命令進行身分驗證。ElastiCache (Redis OSS) 身分驗證權杖，搭配選用的使用者和使用者群組管理，可讓 ElastiCache (Redis OSS) 在允許用戶端執行命令和存取金鑰之前要求密碼，藉此改善資料平面安全性。

**問題層級優點**：為了協助保護資料安全，ElastiCache (Redis OSS) 提供機制來防止未經授權存取您的資料。這包括強制執行角色型存取控制 (RBAC) AUTH 或 AUTH 權杖 (密碼)，供用戶端在執行授權命令 ElastiCache 之前用來連線。

- **【最佳】** 對於 ElastiCache (Redis OSS) 6.x 和更新版本，透過定義使用者群組、使用者和存取字串來定義身分驗證和授權控制。將使用者指派至使用者群組，然後將使用者群組指派至叢集。若要使用 RBAC，必須在建立叢集時選取它，且必須啟用傳輸中加密。確保您使用的是支援的 Valkey 或 Redis OSS 用戶端，TLS 以利用 RBAC。

[資源]：

- [套用至 RBAC ElastiCache \(Redis OSS\) 的複寫群組](#)
- [使用存取字串指定許可](#)
- [ACL](#)
- [支援的 ElastiCache \(RedisOSS\) 版本](#)
- **【最佳】** 對於 6.x 之前的 ElastiCache (Redis OSS) 版本，除了為 ElastiCache (Redis OSS) 設定強式字符/密碼和維護嚴格的密碼政策之外 AUTH，最佳實務是輪換密碼/字符。ElastiCache 可以在任何指定時間管理最多兩 (2) 個身分驗證字符。您也可以修改叢集，以明確要求使用身分驗證字符。

**【資源】**：[修改現有 ElastiCache \( RedisOSS \) 叢集上的AUTH權杖](#)

### SEC 3：命令是否會不慎執行，導致資料遺失或失敗的風險？

**問題層級簡介**：有許多 Valkey 或 Redis OSS 命令，如果錯誤執行或由惡意執行者執行，可能會對操作造成負面影響。從效能和資料安全的角度來看，這些命令可能會產生預料之外的後果。例如，開發人員可能會定期在開發環境中呼叫 FLUSHALL 命令，由於發生錯誤，可能會無意中嘗試在生產系統上呼叫此命令，從而導致意外的資料遺失。

**問題層級優點**：從 ElastiCache ( Redis OSS ) 5.0.3 開始，您可以重新命名可能對工作負載造成干擾的特定命令。重新命名命令有助於防止它們在叢集上意外執行。

• [必要]

[資源]：

- [ElastiCache \( Redis OSS \) 5.0.3 版 \( 已棄用，請使用 5.0.6 版 \)](#)
- [Redis OSS 5.0.3 參數變更](#)
- [Redis OSS 安全性](#)

### SEC 4：如何透過 確保靜態資料加密 ElastiCache

**問題層級簡介**：雖然 ElastiCache ( Redis OSS ) 是記憶體內資料存放區，但可以加密任何可能保留 ( 在儲存體上 ) 作為叢集標準操作一部分的資料。這包括寫入 Amazon S3 的排程和手動備份，以及因同步和交換操作而儲存到磁碟儲存空間的資料。M6g 和 R6g 系列中的執行個體類型也具有永遠開啟記憶體內加密功能。

**問題層級優點**：ElastiCache ( Redis OSS ) 提供可選的靜態加密，以提高資料安全性。

• **【必要】** 只有在 ElastiCache 叢集 ( 複寫群組 ) 上建立時，才能啟用靜態加密。無法藉由修改現有叢集來開始加密靜態資料。根據預設，ElastiCache 會提供和管理靜態加密中使用的金鑰。

[資源]：

- [靜態加密限制](#)
- [啟用靜態加密](#)
- **【最佳】** 利用在記憶體中加密資料的 Amazon EC2 執行個體類型 ( 例如 M6g 或 R6g )。盡可能考慮管理自己的靜態加密金鑰。對於更嚴格的資料安全環境，AWS Key Management Service ( KMS ) 可用於自我管理客戶主金鑰 ( CMK )。透過與的 ElastiCache 整合 AWS Key

Management Service，您可以建立、擁有和管理用於 ElastiCache（Redis OSS）叢集靜態資料加密的金鑰。

[資源]：

- [使用來自的客戶受管金鑰 AWS Key Management Service](#)
- [AWS 金鑰管理服務](#)
- [AWS KMS 概念](#)

## SEC 5：如何使用 加密傳輸中資料 ElastiCache？

問題難易度簡介：一般常會要求在傳輸過程中減少資料遭到入侵的情況。這表示分散式系統元件內的資料，以及應用程式用戶端和叢集節點之間的資料。ElastiCache（Redis OSS）允許加密用戶端和叢集之間傳輸中的資料，以及叢集節點本身之間的資料，藉此支援此要求。M6g 和 R6g 系列中的執行個體類型也具有永遠開啟記憶體內加密功能。

問題層級優點：Amazon 傳輸 ElastiCache 中加密是一項選用功能，可讓您在資料從一個位置傳輸到另一個位置時，提高其最脆弱點的資料安全性。

- **【必要】** 傳輸中加密只能在建立 ElastiCache（Redis OSS）叢集（複寫群組）時啟用。請注意，由於加密/解密資料需要進行額外處理，因此實作傳輸中加密會對效能造成一些影響。若要了解影響，建議在啟用之前和之後對您的工作負載進行基準測試 encryption-in-transit。

[資源]：

- [傳輸中加密概觀](#)

## SEC 6：如何限制控制平面資源的存取？

問題層級簡介：IAM 針對 ElastiCache（Redis OSS）政策和 ARN 啟用精細存取控制，允許更嚴格的控制來管理 ElastiCache（Redis OSS）叢集的建立、修改和刪除。

問題層級優點：Amazon ElastiCache 資源的管理，例如複寫群組、節點等，可能會受限於根據 IAM 政策具有特定許可 AWS 的帳戶，以提高資源的安全性和可靠性。

- **【必要】** 透過將特定 AWS Identity and Access Management 政策指派給使用者來 AWS 管理對 Amazon ElastiCache 資源的存取，從而更精細地控制哪些帳戶可以在叢集上執行哪些動作。

[資源]：

- [管理 資源 ElastiCache 的存取許可概觀](#)



- [針對 Amazon 使用身分型政策 \( IAM 政策 \) ElastiCache](#)

## SEC 7：如何偵測和回應安全事件？

問題層級簡介：ElastiCache部署啟用時RBAC，會匯出 CloudWatch 指標以通知使用者安全事件。這些指標有助於識別RBAC未獲授權的身分驗證、存取金鑰或執行命令的失敗嘗試。

此外，AWS 產品和服務資源可透過自動化部署和記錄所有動作和修改，以供日後檢閱/稽核，協助保護整體工作負載。

問題難易度優點：藉由監控事件可讓您的組織根據您的需求、政策和程序做出回應。自動監控和回應這些安全事件，可強化您的整體安全態勢。

- **【必要】** 熟悉發佈的與RBAC身分驗證和授權失敗相關的CloudWatch 指標。
  - AuthenticationFailures = 嘗試驗證 Valkey 或 Redis 失敗 OSS
  - KeyAuthorizationFailures = 使用者在未經許可的情況下嘗試存取金鑰失敗
  - CommandAuthorizationFailures = 使用者在未經許可的情況下嘗試執行命令失敗

[資源]：

- [Valkey 或 Redis 的指標 OSS](#)
- [最佳] 建議在這些指標上設定警示和通知，並視需要做出回應。

[資源]：

- [使用 Amazon CloudWatch 警示](#)
- **【最佳】** 使用 Valkey 或 Redis OSSACLLOG命令來收集更多詳細資訊

[資源]：

- [ACL LOG](#)
- **【最佳】** 熟悉 AWS 產品和服務功能，因為它與監控、記錄和分析 ElastiCache 部署和事件有關

[資源]：

- [使用 記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-備份-檢查](#)
- [監控與 CloudWatch 指標搭配使用](#)

## Amazon ElastiCache Well-Architected Lens 可靠性支柱

可靠性支柱著重於執行其預期函數的工作負載，以及如何從失敗中快速復原以滿足需求。關鍵主題包括分散式系統設計、復原規劃和適應不斷變化的需求。

### 主題

- [REL 1：您如何支援高可用性（HA）架構部署？](#)
- [REL 2：您如何透過達成復原點目標（RPOs）ElastiCache？](#)
- [REL 3：您如何支援災難復原（DR）要求？](#)
- [REL 4：您如何有效地規劃容錯移轉？](#)
- [REL 5：您的 ElastiCache 元件是否設計用於擴展？](#)

### REL 1：您如何支援高可用性（HA）架構部署？

問題層級簡介：了解 Amazon 的高可用性架構 ElastiCache 可讓您在可用性事件期間以彈性狀態運作。

問題層級優點：建構 ElastiCache 叢集以適應故障，可確保 ElastiCache 部署的可用性更高。

- **【必要】** 判斷叢集所需的可靠性層級 ElastiCache。不同的工作負載具有不同的彈性標準，從完全暫時性工作負載到關鍵任務工作負載都有。為您操作所在的每一種環境（例如開發、測試和生產）定義需求。

快取引擎：ElastiCache（Memcached）vs ElastiCache（Redis OSS）

1. ElastiCache（Memcached）不提供任何複寫機制，主要用於暫時性工作負載。
  2. ElastiCache（RedisOSS）提供下列討論的 HA 功能
- **【最佳】** 對於需要 HA 的工作負載，請在叢集模式下使用 ElastiCache（Redis OSS），每個碎片至少有兩個複本，即使是只需要一個碎片的小型輸送量需求工作負載也是如此。
    1. 若叢集模式已啟用，則會自動啟用多可用區。

多可用區可在進行任何規劃或意外的維護工作以及減少可用區域故障時。藉由自動從主節點容錯移轉至複本的方式，將停機時間降至最低。
    2. 對於碎片工作負載，由於 Valkey 或 Redis OSS 叢集通訊協定需要大多數主要節點才能達到數量，因此至少有三個碎片可在容錯移轉事件期間提供更快的復原。
    3. 在整體可用性中設定兩個或多個複本。

若有兩個複本，就能在其中一個複本進行維護的情況下，改善讀取可擴展性以及讀取可用性。

#### 4. 使用以 Graviton2 為基礎的節點類型 (大多數區域中的預設節點)。

ElastiCache ( Redis OSS ) 已在這些節點上新增最佳化效能。因此，您可以獲得更好的複寫和同步處理效能，進而改善整體可用性。

#### 5. 監控和調整大小以處理預期的流量峰值：在繁重負載下，ElastiCache ( Redis OSS ) 引擎可能會變得無回應，這會影響可用性。DatabaseMemoryUsagePercentage BytesUsedForCache是記憶體用量的良好指標，而ReplicationLag是根據您的寫入速率的複寫運作狀態指標。您可以使用這些指標來觸發叢集擴展。

#### 6. 在[生產容錯移轉事件 API](#)之前，使用容錯移轉進行測試，以確保用戶端恢復能力。

[資源]：

- [設定 ElastiCache \( Redis OSS \) 以獲得更高的可用性](#)
- [使用複寫群組的高可用性](#)

## REL 2：您如何透過 達成復原點目標 ( RPOs ) ElastiCache ？

問題層級簡介：了解工作負載RPO，為 ElastiCache 備份和復原策略的決策提供資訊。

問題層級優點：擁有就地RPO策略可以改善災難復原案例的業務連續性。設計備份和還原政策可協助您滿足 ElastiCache 資料的復原點目標 ( RPO )。ElastiCache ( Redis OSS ) 提供儲存在 Amazon S3 中的快照功能，以及可設定的保留政策。這些快照會在定義的備份時段拍攝，並由服務自動處理。如果您的工作負載需要更精細程度的備份，您可以選擇每天最多建立 20 個手動備份。手動建立的備份不受服務保留政策的約束，可以無限期保留。

- **【必要】** 了解並記錄 ElastiCache 部署RPO的。
  - 請注意，Memcached 不提供任何備份程序。
  - 檢閱 ElastiCache Backup and Restore 功能的功能。
- [最佳] 備妥通訊良好的程序來備份叢集。
  - 視需要啟動手動備份。
  - 檢閱自動備份的保留政策。
  - 請注意，手動備份將無限期保留。
  - 將自動備份排程在低使用量的期間進行。
  - 針對讀取複本執行備份操作，以確保對叢集效能的影響降到最低。
- **【良好】** 利用 的排程備份功能 ElastiCache ，在定義的時段期間定期備份您的資料。
  - 定期測試從備份還原的程序。

- [資源]：
  - [Redis OSS](#)
  - [ElastiCache \( Redis OSS \) 的備份和還原](#)
  - [建立手動備份](#)
  - [排程自動備份](#)
  - [備份和還原 ElastiCache \( RedisOSS \) 叢集](#)

## REL 3：您如何支援災難復原（DR）要求？

**問題層級簡介：**災難復原是任何工作負載規劃的重要層面。ElastiCache（Redis OSS）提供多種選項，可依據工作負載彈性需求實作災難復原。使用 Amazon ElastiCache Global Datastore，您可以寫入一個區域中的 ElastiCache（RedisOSS）叢集，並可從其他兩個跨區域複本叢集讀取資料，從而實現跨區域的低延遲讀取和災難復原。

**問題難易度優點：**了解各種災難情境並規劃因應措施，就可確保業務連續性。災難復原策略必須在成本、效能影響和可能的資料遺失之間取得平衡。

- **【必要】** 根據工作負載需求開發並記錄所有 ElastiCache 元件的 DR 策略。ElastiCache 在部分使用案例中，完全短暫且不需要任何 DR 策略，而其他則位於頻譜的另一端，且需要非常強大的 DR 策略。所有選項都必須針對成本最佳化加以權衡，也就是說，彈性越大，所需的基礎設施數量也越多。

了解區域層級和多區域層級可用的災難復原選項。

- 建議採用多可用區部署來防範可用區域故障。請務必在多可用區域架構中啟用叢集模式進行部署，其中至少 3 個 AZs 可用。
- 建議使用全域資料存放區來防範區域故障。
- [最佳] 針對需要區域層級彈性的工作負載啟用全域資料存放區。
  - 制定計劃，以在主要區域降級時容錯移轉至次要區域。
  - 在生產環境中進行容錯移轉之前，先測試多區域容錯移轉程序。
  - 監控 ReplicationLag 指標，以了解容錯移轉事件期間資料遺失可能造成的影響。
- [資源]：
  - [減少故障](#)
  - [使用全域資料存放區跨 AWS 區域複寫](#)
  - [從備份還原並選擇性地調整叢集規模](#)

## REL 4：您如何有效地規劃容錯移轉？

**問題層級簡介：**使用自動容錯移轉啟用多可用區域是 ElastiCache 最佳實務。在某些情況下，ElastiCache ( Redis OSS ) 會取代主要節點，作為服務操作的一部分。範例情況包括規劃的維護事件，以及少見的節點故障或可用區域問題。成功的容錯移轉取決於 ElastiCache 和用戶端程式庫組態。

**問題層級優點：**搭配特定 ElastiCache ( Redis OSS ) 用戶端程式庫，遵循 ElastiCache 容錯移轉的最佳實務，有助於減少容錯移轉事件期間的潛在停機時間。

- [必要] 若叢集模式已停用，請使用逾時，如此用戶端就能偵測出是否需要與舊的主節點中斷連線，並使用更新的主要端點 IP 地址重新連線至新的主節點。若叢集模式已啟用，則用戶端程式庫會負責偵測基礎叢集拓撲中的變更。這最常透過 ElastiCache ( Redis OSS ) 用戶端程式庫中的組態設定來完成，這也可讓您設定頻率和重新整理方法。每個用戶端程式庫都提供自己的設定，如需詳細資訊，可參閱各自對應的文件。

[資源]：

- [使用多可用區將 ElastiCache \( Redis OSS \) 中的停機時間降到最低](#)
- 檢閱 ElastiCache ( Redis OSS ) 用戶端程式庫的最佳實務。
- [必要] 容錯移轉成功與否，取決於主節點和複本節點之間是否有運作狀態良好的複寫環境。檢閱並了解 Valkey 和 Redis OSS 複寫的非同步性質，以及主要節點和複本節點之間複寫延遲要報告的可用 CloudWatch 指標。對於需要更大資料安全的使用案例，請利用 WAIT 命令強制複本在回應連線的用戶端之前確認寫入。

[資源]：

- [Valkey 或 Redis 的指標 OSS](#)
- [使用 Amazon 使用 ElastiCache \( Redis OSS \) 監控最佳實務 CloudWatch](#)
- **【最佳】** 使用 ElastiCache 測試容錯移轉 定期驗證容錯移轉期間應用程式的回應能力 API。

[資源]：

- [測試自動容錯移轉至 Amazon 上的僅供讀取複本 ElastiCache \( Redis OSS \)](#)
- [測試自動容錯移轉](#)

## REL 5：您的 ElastiCache 元件是否設計用於擴展？

**問題層級簡介：**透過了解擴展功能和可用的部署拓撲，您的 ElastiCache 元件可以隨著時間調整，以滿足不斷變化的工作負載需求。ElastiCache 提供 4 向擴展：進/出（水平）以及上/下（垂直）。

問題層級優點：遵循 ElastiCache 部署的最佳實務可提供最大的擴展彈性，並符合水平擴展的 Well Architected 原則，以將失敗的影響降至最低。

- [必要] 了解叢集模式已啟用與叢集模式已停用的拓撲之間的差異。幾乎所有情況下都建議您在叢集模式已啟用時進行部署，因為這樣就能隨著時間提供更高的可擴展性。叢集模式已停用的元件會在藉由新增讀取複本進行水平擴展的能力上受到限制。
- [必要] 了解擴展的時機和方式。
  - 如需更多 READIOPS：新增複本
  - 如需更多 WRITEOPS：新增碎片（橫向擴展）
  - 如需更多網路 IO：使用網路最佳化的執行個體，縱向擴展
- **【最佳】** 在啟用叢集模式的情況下部署 ElastiCache 元件，偏向更多、更小的節點，而不是更少、更大型的節點。這樣做可有效地限制節點故障的影響範圍。
- [最佳] 在叢集中包含複本，以增強擴展事件期間的回應能力
- **【良好】** 對於停用的叢集模式，請利用僅供讀取複本來提高整體僅供讀取容量。ElastiCache 已支援停用叢集模式最多 5 個僅供讀取複本，以及垂直擴展。
- [資源]：
  - [擴展 ElastiCache \( RedisOSS \) 叢集](#)
  - [線上縱向擴展](#)
  - [Memcached 叢集ElastiCache 的擴展](#)

## Amazon ElastiCache Well-Architected Lens 效能效率支柱

效能效率支柱著重於有效率地使用 IT 和運算資源。重要主題包括根據工作負載需求選取適當的資源類型和大小、監控效能，以及做出明智的決策，以便隨著業務需求的發展維持效率。

### 主題

- [PE 1：如何監控 Amazon ElastiCache 叢集的效能？](#)
- [PE 2：您如何在 ElastiCache 叢集節點間分發工作？](#)
- [PE 3：針對快取工作負載，如何追蹤和報告快取的有效性和效能？](#)
- [PE 4：如何利用您的工作負載讓網路資源和連線得到最佳運用？](#)
- [PE 5：如何管理刪除和/或移出索引鍵？](#)
- [PE 6：如何建立模型並與 中的資料互動 ElastiCache？](#)
- [PE 7：如何在 Amazon ElastiCache 叢集中記錄慢速執行的命令？](#)



- [PE8 : Auto Scaling 如何協助提高ElastiCache 叢集的效能？](#)

## PE 1 : 如何監控 Amazon ElastiCache 叢集的效能？

**問題難易度簡介：**藉由了解現有的監控指標，您就能識別目前的使用率。適當監控有助於識別影響叢集效能的潛在瓶頸。

**問題難易度優點：**了解與叢集相關聯的指標有助於引導最佳化技術，進而降低延遲並增加輸送量。

- [必要] 使用工作負載的子集進行基準效能測試。
  - 您應該使用像是負載測試這類機制來監控實際工作負載的效能。
  - 在執行這些測試時監控 CloudWatch 指標，以了解可用的指標，並建立效能基準。
- **【最佳】** 對於 ElastiCache ( Redis OSS ) 工作負載，重新命名運算上昂貴的命令，例如 KEYS，以限制使用者在生產叢集上執行封鎖命令的能力。
  - ElastiCache ( Redis OSS ) 執行引擎 6.x 的工作負載，可以利用角色型存取控制來限制特定命令。使用 AWS 主控台或 建立使用者和使用者群組CLI，並將使用者群組與 ElastiCache ( Redis OSS ) 叢集建立關聯，即可控制對命令的存取。在 Redis OSS 6 中，RBAC當 啟用時，我們可以使用 "-@dangerous"，而且會不允許該使用者使用昂貴的命令MONITOR，例如 KEYS、SORT、等。
  - 對於引擎 5.x 版，請使用 ElastiCache ( Redis OSS ) 叢集參數群組上的 rename-commands 參數重新命名命令。
- [較佳] 分析緩慢的查詢並尋找最佳化技術。
  - 對於 ElastiCache ( Redis OSS ) 工作負載，請分析慢速日誌，進一步了解您的查詢。例如，您可以使用下列命令 valkey-cli slowlog get 10 來顯示超出延遲閾值 (預設為 10 秒) 的最後 10 個命令。
  - 某些查詢可以使用複雜的 ElastiCache ( RedisOSS ) 資料結構更有效率地執行。舉例來說，針對數值樣式範圍的查詢，應用程式可以使用排序集來實作簡單的數值索引。管理這些索引可以減少對資料集執行掃描的次數，並且以更高的效能效率傳回資料。
  - 對於 ElastiCache ( Redis OSS ) 工作負載，redis-benchmark提供簡單的介面，用於使用使用者定義的輸入來測試不同命令的效能，例如用戶端數量和資料大小。
  - 由於 Memcached 僅支援簡單的索引鍵層級命令，因此請考慮建置其他索引鍵作為索引，以避免反覆查看索引鍵空間來處理用戶端查詢。
- [資源] :
  - [監控與 CloudWatch 指標搭配使用](#)
  - [使用 Amazon CloudWatch 警示](#)

- [Valkey 和 Redis OSS 特定參數](#)
- [SLOWLOG](#)
- [基準](#)

## PE 2：您如何在 ElastiCache 叢集節點間分發工作？

問題層級簡介：應用程式連線至 Amazon ElastiCache 節點的方式可能會影響叢集的效能和可擴展性。

問題難易度優點：正確使用叢集中的可用節點，可確保將工作分配到可用的資源。以下技術也有助於避免資源閒置。

- **【必要】** 讓用戶端連線到適當的 ElastiCache 端點。
  - ElastiCache ( Redis OSS ) 會根據使用的叢集模式實作不同的端點。對於啟用叢集模式，ElastiCache 將提供組態端點。對於停用的叢集模式，ElastiCache 提供主要端點，通常用於寫入，以及用於平衡複本讀取的讀取器端點。正確實作這些端點可提升效能並且更輕鬆地擴展操作。除非有特定需求，否則避免連線到個別節點端點。
  - 對於多節點 Memcached 叢集，ElastiCache 提供啟用 Auto Discovery 的組態端點。建議使用雜湊演算法將工作平均分配到各個快取節點。許多 Memcached 用戶端程式庫會實作一致的雜湊。檢查您使用的程式庫文件，查看其是否支援一致性雜湊以及其實作方式。您可以在[這裡](#)找到有關實做這些功能的詳細資訊。
- **【更好的】** 利用啟用的 ElastiCache ( Redis OSS ) 叢集模式來改善可擴展性。
  - ElastiCache ( Redis OSS ) ( 啟用叢集模式 ) 叢集支援[線上擴展操作](#) ( 輸出/輸入和上/下 )，以協助在碎片之間動態分發資料。使用組態端點可確保您的叢集感知用戶端能夠因應叢集拓撲中的變更進行調整。
  - 您也可以可以在 ElastiCache ( Redis OSS ) ( 啟用叢集模式 ) 叢集中的可用碎片之間移動雜湊區，以重新平衡叢集。這樣做有助於更有效率地在可用碎片中分配工作。
- **[較佳]** 實施策略來識別和修復工作負載中的快速鍵。
  - 考慮多維度 Valkey 或 Redis OSS 資料結構的影響，例如清單、串流、集等。這些資料結構儲存在單一金鑰中，這些金鑰位於單一節點上。相當大型的多維度索引鍵可能比其他資料類型利用更多的網路容量和記憶體，並且可能造成該節點的使用率不成比例。您的工作負載設計應盡可能將資料存取分散到多個獨立的索引鍵。
  - 工作負載中的快速鍵可能會影響使用中節點的效能。對於 ElastiCache ( Redis OSS ) 工作負載，`valkey-cli --hotkeys` 如果已制定 LFU 最大記憶體政策，您可以使用偵測快速鍵。



- 請考慮將快速鍵複寫到多個節點，讓分配到各節點的存取權更平均。此方法需要用戶端寫入多個主要節點（Valkey 或 Redis OSS 節點本身不提供此功能），並維護除了原始金鑰名稱之外要讀取的金鑰名稱清單。
- ElastiCache 搭配 Valkey 7.2 及更高版本，以及 Redis 第 6 版及更高 OSS 版本，支援伺服器輔助用戶端快取。這可讓應用程式等待變更金鑰，然後再將網路呼叫回 ElastiCache。
- [資源]：
  - [ElastiCache 使用 Valkey 和 Redis 設定 OSS 以獲得更高的可用性](#)
  - [在中尋找連線端點 ElastiCache](#)
  - [負載平衡最佳實務](#)
  - [Valkey 或 Redis 的線上重新共享 OSS \( 啟用叢集模式 \)](#)
  - [Valkey 和 Redis 中的用戶端快取 OSS](#)

### PE 3：針對快取工作負載，如何追蹤和報告快取的有效性和效能？

問題層級簡介：快取是上常見的工作負載 ElastiCache，您必須了解如何管理快取的有效性和效能。

問題難易度優點：您的應用程式可能會出現效能遲緩的跡象。假如能夠使用快取專用指標做出明智的決策來提高應用程式效能，這點對於快取工作負載而言至關重要。

- [必要] 測量並追蹤經過一段時間的快取命中率。快取的效率是由其「快取命中率」所決定。快取命中率的計算定義是索引鍵命中總數除以命中加未命中總數。命中率越接近 1，表示快取越有效。快取未命中數量是造成快取命中率低的原因。快取中找不到請求的索引鍵時，就會產生快取未命中數。快取中沒有某個索引鍵是因為該索引鍵已移出或刪除、已過期或不曾存在。了解索引鍵為什麼不在快取中，並制定適當的策略將其納入快取中。

[資源]：

- [Valkey 和 Redis 的指標 OSS](#)
- **【必要】** 測量並收集應用程式快取效能，以及延遲和 CPU 使用率值，以了解您是否需要調整您的 time-to-live 或其他應用程式元件。為每個資料結構 ElastiCache 提供彙總延遲的一組 CloudWatch 指標。這些延遲指標是使用來自 ElastiCache ( Redis OSS ) INFO 命令的命令統計資料計算，不包含網路和 I/O 時間。這只是 ElastiCache ( Redis OSS ) 處理操作所耗用的時間。

[資源]：

- [Valkey 和 Redis 的指標 OSS](#)
- [使用 Amazon 使用 ElastiCache \( Redis OSS \) 監控最佳實務 CloudWatch](#)

- [最佳] 選擇最合乎您需要的快取策略。快取未命中數量是造成快取命中率低的原因。如果您的工作負載設計是維持低快取未命中數量 (例如即時通訊)，那麼最好檢閱您的快取策略，並對工作負載套用最適當的解決方案 (例如查詢檢測) 來測量記憶體和效能。您實際用來填入和維護快取的策略，取決於您的用戶端需要快取的資料，以及該資料的存取模式。例如，您不太可能對串流應用程式上的個人化推薦和熱門新聞報導採用相同的策略。

[資源]：

- [Memcached 的快取策略](#)
- [快取最佳實務](#)
- [Amazon ElastiCache 白皮書的大規模效能](#)

## PE 4：如何利用您的工作負載讓網路資源和連線得到最佳運用？

問題層級簡介：ElastiCache 許多應用程式用戶端支援 (Redis OSS) 和 ElastiCache (Memcached)，且實作可能會有所不同。您需要了解現有的網路和連線管理，以分析潛在的效能影響。

問題難易度優點：有效運用網路資源可改善叢集的效能效率。下列建議可減少網路需求，並改善叢集延遲和輸送量。

- **【必要】** 主動管理 ElastiCache 叢集的連線。
  - 在應用程式中使用連線集區，可減少因開啟和關閉連線而對叢集造成的額外負荷量。CloudWatch 使用 `CurrConnections` 和 監控 Amazon 中的連線行為 `NewConnections`。
  - 適時確實地關閉用戶端連線，以避免洩漏連線。連線管理策略包括確實地關閉未使用的連線，以及設定連線逾時。
  - 針對 Memcached 工作負載保留了用於處理連線的可設定記憶體數量，稱為 `memcached_connections_overhead`。
- [較佳] 壓縮大型物件以減少記憶體並改善網路輸送量。
  - 資料壓縮可減少所需的網路輸送量 (Gbps)，但會增加應用程式壓縮和解壓縮資料的工作量。
  - 壓縮還可以減少索引鍵耗用的記憶體數量
  - 請根據您應用程式的需求，考慮壓縮比和壓縮速度之間的權衡。
- [資源]：
  - [ElastiCache \(Redis OSS\) - 全球資料存放區](#)
  - [Memcached 專用參數](#)
  - [ElastiCache \(Redis OSS\) 5.0.3 增強了 I/O 處理，以提高效能](#)

- [Valkey 和 Redis 的指標 OSS](#)
- [設定 ElastiCache \( Redis OSS \) 以獲得更高的可用性](#)

## PE 5：如何管理刪除和/或移出索引鍵？

問題層級簡介：工作負載有不同的需求，以及當叢集節點接近記憶體耗用限制時的預期行為。ElastiCache ( Redis OSS ) 有不同的政策來處理這些情況。

問題難易度優點：適當管理可用記憶體並且了解移出政策，將有助於確保在超過執行個體記憶體限制時，能夠察覺到叢集行為。

- [必要] 檢測資料存取權以評估要套用的政策。找出適當的最大記憶體政策，以控制是否要在叢集上執行移出，以及移出的方式。
  - 當達到叢集上的最大記憶體耗用量，而且已設有允許移出的政策時，就會進行移除。在此情況下，叢集的行為會取決於指定的移出政策。此政策可以使用 ElastiCache ( Redis OSS ) 叢集參數群組 `maxmemory-policy` 上的 進行管理。
  - 預設政策會逸出具有設定過期時間 ( TTL 值 ) 的金鑰來 `volatile-lru` 釋放記憶體。最少常用 ( LFU ) 和最少最近使用 ( LRU ) 的政策會根據用量移除金鑰。
  - 對於 Memcached 工作負載，會有一個預設 LRU 政策來控制每個節點上的驅逐。您可以使用 Amazon 上的 Evictions 指標來監控 Amazon ElastiCache 叢集上的驅逐次數 CloudWatch。
- [較佳] 將刪除行為標準化，就可控制對叢集的效能影響，進而避免非預期的效能瓶頸發生。
  - 對於 ElastiCache ( Redis OSS ) 工作負載，從叢集明確移除金鑰時，UNLINK 就像 DEL：它會移除指定的金鑰。然而，該命令會在不同的執行緒中執行實際的記憶體回收，因此不會封鎖，但 DEL 會封鎖。實際的移除操作會在之後以非同步方式進行。
  - 對於 ElastiCache ( Redis OSS ) 6.x 工作負載，可以使用 `lazyfree-lazy-user-del` 參數在參數群組中修改 DEL 命令的行為。
- [資源]：
  - [使用參數群組設定引擎 ElastiCache 參數](#)
  - [UNLINK](#)
  - [使用的雲端財務管理 AWS](#)

## PE 6：如何建立模型並與 中的資料互動 ElastiCache ？

**問題層級簡介：** ElastiCache 高度依賴於資料結構和使用的資料模型的應用程式，但還需要考慮基礎資料存放區（如果有）。了解可用的 ElastiCache（Redis OSS）資料結構，並確保您使用最適合您需求的資料結構。

**問題層級優點：** 中的資料建模ElastiCache 有數個層，包括應用程式使用案例、資料類型和資料元素之間的關係。此外，每個 ElastiCache（Redis OSS）資料類型和命令都有自己的妥善記錄效能簽章。

- [最佳] 最佳實務是減少意外覆寫資料的情況。使用盡可能減少索引鍵名稱重疊的命名慣例。資料結構的慣用命名是使用像是 APPNAME:CONTEXT:ID 這類階層方法，例如 ORDER-APP:CUSTOMER:123。

[資源]：

- [索引鍵命名](#)
- **【最佳】** ElastiCache（Redis OSS）命令具有 Big O 記號定義的時間複雜性。命令的這種時間複雜度是其影響的演算/數學表示法。當您在應用程式中引入新的資料類型時，需仔細查看相關命令的時間複雜度。時間複雜度為  $O(1)$  的命令在時間上是恆定的，並不取決於輸入大小，但是時間複雜度為  $O(N)$  的命令在時間上是線性的，因此會受輸入大小的影響。由於 ElastiCache（Redis OSS）的單一螺紋設計，大量的高時間複雜性操作將導致效能降低和潛在的操作逾時。

[資源]：

- [命令](#)
- **【最佳】** 使用 APIs 來取得叢集中資料模型的 GUI 可見性。

[資源]：

- [Redis OSS Commander](#)
- [Redis OSS 瀏覽器](#)
- [Redsmin](#)

## PE 7：如何在 Amazon ElastiCache 叢集中記錄慢速執行的命令？

**問題難易度簡介：** 透過擷取、彙總和通知長時間執行的命令，達到效能調整效益。透過了解執行命令所需的時間，您可以判斷哪些命令會導致效能不佳，以及封鎖引擎以最佳方式執行的命令。ElastiCache（Redis OSS）也能夠將此資訊轉送至 Amazon CloudWatch 或 Amazon Kinesis Data Firehose。

**問題難易度優點：** 針對慢速命令記錄到專用的永久位置並提供通知事件，有助於進行詳細的效能分析，並可用於觸發自動化事件。

- **【必要】** 執行引擎 6.0 版或更新版本的 Amazon ElastiCache ( Redis OSS )，在叢集上啟用正確設定的參數群組和SLOWLOG記錄。
  - 只有在引擎版本相容性設定為 Valkey 7.2 及更高版本，或 Redis 6.0 版或更高OSS版本時，才能使用所需的參數。
  - SLOWLOG 當命令的伺服器執行時間超過指定的值時，就會發生記錄。叢集的行為取決於相關聯的參數群組參數，也就是 `slowlog-log-slower-than` 和 `slowlog-max-len`。
  - 變更會立即生效。
- **【最佳】** 利用 CloudWatch 或 Kinesis Data Firehose 功能。
  - 使用 CloudWatch、CloudWatchLogs Insights 和 Amazon Simple Notification Services 的篩選和警示功能來實現效能監控和事件通知。
  - 使用 Kinesis Data Firehose 的串流功能將SLOWLOG日誌封存至永久儲存體，或觸發自動叢集參數調校。
  - 判斷 JSON或 純TEXT格式是否最符合您的需求。
  - 提供發佈至 CloudWatch 或 Kinesis Data Firehose 的IAM許可。
- [較佳] 將 `slowlog-log-slower-than` 設定為預設值以外的值。
  - 此參數決定命令在記錄為慢速執行命令之前，可在 Valkey 或 Redis OSS引擎內執行多久。預設值為 10,000 微秒 (10 毫秒)。對於某些工作負載而言，預設值可能太高。
  - 根據應用程式需求和測試結果，決定更合乎您工作負載的值；不過，太低的值可能會產生過多資料。
- [較佳] 保留 `slowlog-max-len` 的預設值。
  - 此參數會決定在任何指定時間，Valkey 或 Redis OSS記憶體中擷取多少慢速執行命令的上限。值為 0 會有效停用擷取。值越高，記憶體中儲存的項目就越多，因而減少重要資訊在檢閱之前就遭到移出的情形。預設值為 128。
  - 預設值適用於大多數的工作負載。如果需要透過 SLOWLOG命令從 `valkey-cli` 擴充的時段中分析資料，請考慮增加此值。這允許更多命令保留在 Valkey 或 Redis OSS記憶體中。

如果您要將SLOWLOG資料發射至 CloudWatch Logs 或 Kinesis Data Firehose，資料會保留，而且可以在 ElastiCache 系統外部進行分析，減少在 Valkey 或 Redis OSS記憶體中存放大量慢速執行命令的需求。

- [資源]：
  - [如何開啟 ElastiCache \( Redis OSS \) 快取叢集中的慢速登入？](#)
  - [日誌傳送](#)

- <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
- [Amazon Kinesis Data Firehose](#)

## PE8 : Auto Scaling 如何協助提高ElastiCache 叢集的效能？

問題層級簡介：透過實作 Valkey 或 Redis OSS自動擴展功能，您的 ElastiCache 元件可以隨著時間調整，以自動增加或減少所需的碎片或複本。藉由實作目標追蹤或排程的擴展政策就能達到此目的。

問題層級優點：了解和規劃工作負載中的峰值可確保增強的快取效能和業務連續性。ElastiCache ( Redis OSS ) Auto Scaling 會持續監控您的 CPU/Memory 使用率，以確保您的叢集以所需的效能層級運作。

- **【必要】** 啟動 ElastiCache ( Redis ) 的叢集時OSS：
  1. 確定已啟用叢集模式
  2. 確定執行個體屬於支援自動擴展的特定類型和大小系列
  3. 確保叢集未在全域資料存放區、Outposts 或 Local Zones 中執行

[資源]：

- [在 Valkey 和 Redis 中擴展叢集 OSS \( 已啟用叢集模式 \)](#)
- [搭配碎片使用自動擴展](#)
- [搭配複本使用自動擴展](#)
- [最佳] 確認您的工作負載為大量讀取或大量寫入，以定義擴展政策。為了獲得最佳效能，請使用單獨一個追蹤指標。建議您避免針對每個維度實施多項政策，因為自動擴展政策會在命中目標時橫向擴展，但只有在所有目標追蹤政策都準備好縮減時才會縮減。

[資源]：

- [自動擴展政策](#)
- [定義擴展政策](#)
- [最佳] 隨著時間監控效能可幫助您偵測到在特定時間點監控時，可能無法察覺的工作負載變更。您可以分析叢集使用率在四週期間內的對應CloudWatch 指標，以判斷目標值閾值。如果您仍不確定要選擇哪個值，建議從支援的最小預先定義指標值開始。

[資源]：

- [監控與 CloudWatch 指標搭配使用](#)



- [較佳] 我們建議您使用預期的最小和最大工作負載來測試應用程式，藉此找出叢集制定擴展政策和減輕可用性問題所需的準確碎片/複本數量。

[資源]：

- [註冊可擴展的目標](#)
- [使用註冊可擴展目標 AWS CLI](#)

## Amazon ElastiCache Well-Architected Lens 成本最佳化支柱

成本最佳化支柱著重於避免不必要的成本。重要主題包括了解和控制錢花費在哪裡、選取最合適的節點類型 (使用支援根據工作負載需求進行資料分層的執行個體)、適切的資源類型數量 (有多少個讀取複本)、分析經過一段時間後的花費，以及因應業務需求進行擴展而不超支。

主題

- [COST 1：如何識別和追蹤與 ElastiCache 資源相關的成本？如何制定出各種機制，讓使用者能夠建立、管理及處置建立的資源？](#)
- [COST 2：如何使用持續監控工具來協助您最佳化與 ElastiCache 資源相關聯的成本？](#)
- [COST 3：您是否應使用支援資料分層的執行個體類型？資料分層有何優點？何時不適合使用資料分層執行個體？](#)

**COST 1：如何識別和追蹤與 ElastiCache 資源相關的成本？如何制定出各種機制，讓使用者能夠建立、管理及處置建立的資源？**

問題難易度簡介：若要了解成本指標，就需要有多個團隊參與並且跨團隊協作：軟體工程、資料管理、產品負責人、財務及領導階層。若要找出關鍵的成本驅動因素，就需要參與的各方都了解服務用量控制槓桿與成本管理權衡，這時常成為投入成本最佳化的努力獲得成功與不太成功的關鍵差異。確保您擁有適當的程序和工具，以追蹤從開發到生產和淘汰期間建立的資源，可協助您管理與相關的 ElastiCache 成本。

問題層級優點：持續追蹤與工作負載相關的所有成本需要深入了解包含 ElastiCache 作為其元件之一的架構。此外，您應制定成本管理計劃來收集用量，並與您的預算進行比較。

- **【必要】** 建立 Cloud Center of Excellence ( CCoE ) 及其創始章程之一，以擁有定義、追蹤和針對組織 ElastiCache 用量的指標採取行動。如果 CCoE 存在 和 函數，請確保它知道如何讀取和追蹤與相關聯的成本 ElastiCache。建立資源時，請使用 IAM 角色和政策來驗證只有特定團隊和群組可以實現資源。這樣可確保成本與業務成果相關聯，並從成本的角度建立明確的責任脈絡。

1. CCoE 應識別、定義和發佈每月定期更新的成本指標，這些指標圍繞類別資料中的金鑰 ElastiCache 用量，例如：
  - a. 使用的節點類型及其屬性：標準與記憶體最佳化、隨需與預留執行個體、區域和可用區域
  - b. 環境類型：免費、開發、測試和生產
  - c. 備份儲存與保留策略
  - d. 區域內與跨區域的資料傳輸
  - e. 在 Amazon Outposts 上執行的執行個體
2. CCoE 由跨職能團隊組成，其中包含組織中軟體工程、資料管理、產品團隊、財務和領導團隊的非專屬代表。

[資源]：

- [打造雲端卓越中心](#)
  - [Amazon ElastiCache 定價](#)
- [必要] 使用成本分配標籤以較低精細程度來追蹤成本。使用 AWS 成本管理來視覺化、了解和管理一段時間內的 AWS 成本和用量。
    1. 使用標籤來整理您的資源，並使用成本分配標籤來詳細追蹤您的 AWS 成本。啟用成本分配標籤後，AWS 會使用成本分配標籤來整理成本分配報告中的資源成本，讓您更輕鬆地分類和追蹤 AWS 成本。AWS 提供兩種類型的成本分配標籤、AWS 產生的標籤和使用者定義標籤。為您 AWS 定義、建立和套用 AWS 產生的標籤，以及定義、建立和套用使用者定義標籤。您必須分別啟用這兩種標籤，它們才會顯示在 Cost Management 或成本分配報告中。
    2. 使用成本分配標籤來組織 AWS 帳單，以反映您自己的成本結構。當您將成本分配標籤新增至 Amazon 中的資源時 ElastiCache，您將能夠透過將發票上的費用依資源標籤值分組來追蹤成本。您可考慮結合標籤，以便更深入追蹤成本的細節。

[資源]：

- [使用 AWS 成本分配標籤](#)
  - [使用成本分配標籤監控成本](#)
  - [AWS Cost Explorer](#)
- **【最佳】** 將 ElastiCache 成本連接到跨組織達到的指標。
    1. 考慮業務指標以及像是延遲等操作指標 - 您的業務模型中有哪些概念是可以跨角色理解的？這些指標需要讓組織中越多角色理解越好。
    2. 範例 - 同時提供服務的使用者、每項操作和每個使用者的最大和平均延遲、使用者參與度分數、使用者回流率/週、工作階段長度/使用者、放棄率、快取命中率，以及追蹤的索引鍵



[資源]：

- [監控與 CloudWatch 指標搭配使用](#)
- **【良好】** 使用的整個工作負載，保持 up-to-date 指標和成本的架構和操作可見性 ElastiCache。
  1. 了解您的整個解決方案生態系統，ElastiCache 傾向成為 AWS 其技術集中服務的完整生態系統的一部分，從用戶端到 API Gateway、Redshift 和 QuickSight 報告工具（例如）。
  2. 在您的架構圖上，對應來自用戶端、連線、安全性、記憶體內操作、儲存、資源自動化、資料存取和管理的解決方案元件。每一層都連接到整個解決方案，並且有自己的需求和功能，能夠增添和/或有助您管理整體成本的能力。
  3. 您的圖表應包括使用運算、聯網、儲存、生命週期政策、指標收集，以及應用程式的操作和功能 ElastiCache 元素
  4. 工作負載的需求可能會隨著時間而演進，因此您務必繼續維護並記錄對基礎元件以及主要功能目標的理解程度，以便在工作負載成本管理中保持主動。
  5. 高階主管對可見性、責任、優先順序和資源的支援，對於您擁有有效的 成本管理策略至關重要 ElastiCache。

## COST 2：如何使用持續監控工具來協助您最佳化與 ElastiCache 資源相關聯的成本？

**問題層級簡介：**您需要在 ElastiCache 成本和應用程式效能指標之間取得適當的平衡。Amazon CloudWatch 提供關鍵操作指標的可見性，可協助您評估 ElastiCache 資源是否過度使用或不足，以滿足您的需求。從成本最佳化的角度來看，您需要了解何時過度佈建，並能夠開發適當的機制來調整資源大小 ElastiCache，同時維持營運、可用性、彈性和效能需求。

**問題難易度優點：**在理想狀態下，您將佈建足夠的資源來滿足工作負載運作需求，並且沒有資源使用率不足的情況，而導致處於非最佳成本狀態。您需要能夠識別和避免長時間操作過大 ElastiCache 的資源。

- **【必要】** 使用 CloudWatch 來監控 ElastiCache 叢集，並分析這些指標與您的 AWS Cost Explorer 儀表板有何關聯。
  1. ElastiCache 同時提供主機層級指標（例如，CPU 用量）和快取引擎軟體特有的指標（例如，快取取得和快取遺失）。每隔 60 秒會針對每個快取節點測量及發佈這些指標。
  2. ElastiCache 效能指標（CPUUtilization EngineUtilization、SwapUsage CurrConnections、和 Evictions）可能表示您需要向上/向下擴展（使用較大/較小快取節點類型）或輸入/輸出（新增更多/較少碎片）。藉由建立教戰手冊對照表來預估額外成本，以及達到應用程式效能閾值所需的最短和最長時間，從而了解擴展決策的成本影響。

[資源]：

- [監控與 CloudWatch 指標搭配使用](#)
  - [應監控哪些指標？](#)
  - [Amazon ElastiCache 定價](#)
- [必要] 了解並記錄您的備份策略和成本影響。
1. 使用 ElastiCache 時，備份會儲存在 Amazon S3 中，可提供持久的儲存。您需要了解與您的故障復原能力相關的成本影響。
  2. 啟用自動備份，這樣將會刪除保留期限已過的備份檔案。

[資源]：

- [排程自動備份](#)
  - [Amazon Simple Storage Service 定價](#)
- [最佳] 針對執行個體使用預留節點，這是為了管理已充分了解並記錄的工作負載成本刻意而為的策略。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度（一或三年）。此費用遠低於您使用隨需節點時需支付的每小時使用費。
1. 您可能需要使用隨需節點操作 ElastiCache 叢集，直到您已收集足夠的資料來估計預留執行個體需求為止。規劃並記錄滿足您的需求所需的資源，並比較各執行個體類型（隨需與預留）的預期成本
  2. 定期評估可用的新快取節點類型，並從成本和操作指標的角度評估是否合理，以便將您的執行個體機群移轉到新的快取節點類型

### COST 3：您是否應使用支援資料分層的執行個體類型？資料分層有何優點？何時不適合使用資料分層執行個體？

問題難易度簡介：選取適當的執行個體類型不僅會影響效能和服務層面，還會影響財務層面。執行個體類型有各種不同的相關成本。您可能自然而然會選取一個或少數幾個可滿足記憶體中所有儲存需求的大型執行個體類型。但是隨著專案逐漸成熟，這可能會產生重大的成本影響。確保選取正確的執行個體類型需要定期檢查 ElastiCache 物件閒置時間。

問題難易度優點：您應該清楚了解各種不同的執行個體類型對您目前和未來的成本有何影響。邊際或定期工作負載變更不應造成不成比例的成本變更。在工作負載允許的情況下，選擇支援資料分層的執行個體類型就能提供單一儲存價格更實惠的可用儲存。由於每個執行個體可用的 SSD 儲存資料分層執行個體支援更高的每個執行個體總資料量。

- [必要] 了解資料分層執行個體的限制

1. 僅適用於 ElastiCache ( Redis OSS ) 叢集。
2. 僅限幾種執行個體類型可支援資料分層。
3. 僅支援 ElastiCache ( Redis OSS ) 6.2 版及更新版本
4. 大型項目不會切換為 SSD。超過 128 MiB 的物件會保存在記憶體中。

[資源]：

- [資料分層](#)
- [Amazon ElastiCache 定價](#)
- [必要] 了解工作負載定期存取的資料庫百分比。
  1. 資料分層執行個體非常適合經常存取整個資料集的一小部分，但仍需要快速存取其餘資料的工作負載。換句話說，熱資料對暖資料的比例約為 20:80。
  2. 發展叢集層級的物件閒置時間追蹤。
  3. 超過 500 Gb 資料量的大型實作就是很好的選擇。
- [必要] 了解資料分層執行個體並非某些工作負載的可選項目。
  1. 存取較不常用的物件需要很小的效能成本，因為這些物件會換成本機 SSD。如果您的應用程式對回應時間較為敏感，請測試對工作負載的影響。
  2. 不適用大多儲存大小超過 128 MiB 之大型物件的快取。

[資源]：

- [限制](#)
- [最佳] 預留執行個體類型可支援資料分層。這樣可確保每個執行個體的資料儲存量擁有最低成本。
  1. 您可能需要使用非資料分層執行個體操作 ElastiCache 叢集，直到您更了解您的需求為止。
  2. 分析 ElastiCache 叢集的資料用量模式。
  3. 建立自動化的工作來定期收集物件閒置時間。
  4. 如果您發現有很大百分比 (約 80%) 的物件閒置了一段時間，則視為適合讓您的工作負載記錄調查結果，並建議將叢集遷移到支援資料分層的執行個體。
  5. 定期評估可用的新快取節點類型，並從成本和操作指標的角度評估是否合理，以便將您的執行個體機群移轉到新的快取節點類型。

[資源]：

- [OBJECT IDLETIME](#)
- [Amazon ElastiCache 定價](#)

# 使用的常見疑難排解步驟和最佳實務 ElastiCache

下列主題針對您在使用時可能遇到的錯誤和問題提供疑難排解建議 ElastiCache。如果您發現此處未列出的問題，您可以使用此頁面上的意見回饋按鈕進行報告。

如需常見支援問題的更多疑難排解建議和答案，請造訪 [AWS 知識中心](#)

## 主題

- [連線問題](#)
- [Valkey 或 Redis OSS用戶端錯誤](#)
- [對 ElastiCache Serverless 中的高延遲進行故障診斷](#)
- [對 ElastiCache Serverless 中的限流問題進行故障診斷](#)
- [持續性連線問題](#)
- [相關主題](#)

## 連線問題

如果您無法連線至 ElastiCache 快取，請考慮下列其中一項：

1. 使用 TLS：如果您在嘗試連線至 ElastiCache 端點時遇到掛載連線，則可能不會 TLS 在用戶端中使用。如果您使用 ElastiCache Serverless，傳輸中的加密一律會啟用。請確定您的用戶端正在使用 TLS 連線到快取。[進一步了解如何連線至 TLS 已啟用的快取](#)。
2. VPC：ElastiCache 快取只能在內存取 VPC。確定您存取快取和 ElastiCache 快取的 EC2 執行個體是在相同的 VPC 中建立 VPC。或者，您必須啟用 VPCEC2 執行個體所在的 VPC 與您要建立快取 VPC 的 VPC 之間的 [VPC 對等](#)。
3. 安全群組：ElastiCache 使用安全群組來控制對快取的存取。考慮下列各項：
  - a. 請確定 ElastiCache 快取所使用的安全群組允許從 EC2 執行個體傳入存取它。請參閱 [此處](#)，了解如何在安全群組中正確設定傳入規則。
  - b. 請確定 ElastiCache 快取使用的安全群組允許存取快取的連接埠（無伺服器 6379 和 6380，自行設計 6379）。ElastiCache 使用這些連接埠接受 Valkey 或 Redis OSS 命令。在此處了解如何設定連接埠存取權的詳細資訊。

如果連線仍然困難，請參閱 [持續性連線問題](#) 以取得其他步驟。

## Valkey 或 Redis OSS用戶端錯誤

ElastiCache 無伺服器只能使用支援 Valkey 或 Redis OSS叢集模式通訊協定的用戶端存取。可根據叢集組態，從任一模式下的用戶端存取自行設計的叢集。

如果您的用戶端發生錯誤，請考慮下列事項：

1. 叢集模式：如果您使用 [SELECT](#) 命令遇到 CROSSLOT 錯誤或錯誤，則可能嘗試使用不支援叢集通訊協定的 Valkey 或 Redis OSS用戶端存取已啟用叢集模式的快取。ElastiCache Serverless 僅支援支援 Valkey 或 Redis OSS叢集通訊協定的用戶端。如果您想要在「停用叢集模式」（CMD）OSS 中使用 Valkey 或 Redis，則必須設計自己的叢集。
2. CROSSLOT 錯誤：如果您遇到 ERR CROSSLOT Keys in request don't hash to the same slot 錯誤，您可能嘗試存取不屬於叢集模式快取中相同插槽的金鑰。提醒您，ElastiCache Serverless 一律以叢集模式運作。只有在涉及的所有金鑰都位於相同的雜湊槽中時，才允許涉及多個金鑰的多金鑰操作、交易或 Lua 指令碼。

如需有關設定 Valkey 或 Redis OSS用戶端的其他最佳實務，請檢閱此 [部落格文章](#)。

## 對 ElastiCache Serverless 中的高延遲進行故障診斷

如果您的工作負載似乎發生高延遲，您可以分析 CloudWatch SuccessfulReadRequestLatency 和 SuccessfulWriteRequestLatency 指標，以檢查延遲是否與無 ElastiCache 伺服器有關。這些指標會測量無 ElastiCache 伺服器內部的延遲 - 不包括用戶端與無 ElastiCache 伺服器端點之間的用戶端延遲和網路跳脫時間。

### 對用戶端延遲進行故障診斷

如果您在用戶端注意到延遲增加，但沒有對應的增加，CloudWatch SuccessfulReadRequestLatency 以及測量伺服器端延遲的 SuccessfulWriteRequestLatency 指標，請考慮下列事項：

- 確保安全群組允許存取連接埠 6379 和 6380：ElastiCache Serverless 使用主要端點的 6379 連接埠，以及讀取器端點的 6380 連接埠。有些用戶端會為每個新連線建立與兩個連接埠的連線，即使您的應用程式未使用從複本讀取功能。如果您的安全群組不允許傳入存取這兩個連接埠，則建立連線可能需要更長的時間。在此處了解如何設定連接埠存取權的詳細資訊。

### 對伺服器端延遲進行故障診斷



某些變化性和偶爾尖峰不應是造成疑慮的原因。不過，如果Average統計資料顯示急劇增加並持續，您應該檢查 AWS Health Dashboard 和個人運作狀態儀表板以取得詳細資訊。如有必要，請考慮使用開啟支援案例 AWS Support。

請考慮下列最佳實務和策略，以減少延遲：

- 啟用從複本讀取：如果您的應用程式允許，建議您在 Valkey 或 Redis OSS用戶端中啟用「從複本讀取」功能，以擴展讀取並實現更低的延遲。啟用時，ElastiCache Serverless 會嘗試將您的讀取請求路由至與用戶端位於相同可用區域（AZ）的複本快取節點，以避免跨可用區域網路延遲。請注意，在用戶端中啟用從複本讀取功能表示您的應用程式接受最終的資料一致性。如果您在寫入金鑰後嘗試讀取資料，應用程式可能會收到較舊的資料一段時間。
- 確保您的應用程式部署在AZs與快取相同的 中：如果您的應用程式未部署在AZs與快取相同的 中，您可能會觀察到更高的用戶端延遲。建立無伺服器快取時，您可以提供應用程式存取快取的子網路，而無 ElastiCache 伺服器則在這些子網路中建立VPC端點。確保您的應用程式部署在相同的 中AZs。否則，您的應用程式在存取快取時可能會產生跨可用區跳轉，導致更高的用戶端延遲。
- 重複使用連線：ElastiCache 無伺服器請求是透過TLS已啟用的TCP連線使用RESP通訊協定提出。啟動連線（包括驗證連線，如果已設定）需要時間，因此第一個請求的延遲高於一般。透過已初始化連線的請求可提供 ElastiCache一致的低延遲。因此，您應該考慮使用連線集區或重複使用現有的 Valkey 或 Redis OSS連線。
- 擴展速度：ElastiCache 無伺服器隨著請求率的增加自動擴展。請求速率突然大幅增加，速度比無 ElastiCache 伺服器擴展的速度快，可能會導致延遲增加一段時間。ElastiCache Serverless 通常可以快速提高其支援的請求速率，最多需要 10-12 分鐘才能使請求速率加倍。
- 檢查長時間執行的命令：某些 Valkey 或 Redis OSS命令，包括大型資料結構上的 Lua 指令碼或命令，可能會長時間執行。若要識別這些命令，會 ElastiCache 發佈命令層級指標。透過 [ElastiCache Serverless](#)，您可以使用 BasedECPUs指標。
- 限流請求：在無 ElastiCache 伺服器中限流請求時，您可能會遇到應用程式中用戶端延遲增加的情況。當請求在 ElastiCache Serverless 中限流時，您應該會看到 ThrottledRequests [ElastiCache Serverless](#) 指標增加。檢閱以下章節，以針對限流請求進行疑難排解。
- 金鑰和請求的均勻分佈：ElastiCache 在搭配 Valkey 和 Redis 的情況下OSS，每個插槽的金鑰或請求分佈不平均可能會導致一個熱槽，這可能會導致延遲增加。ElastiCache Serverless 在執行簡單 SET/GET 命令的工作負載中，支援單一插槽上高達 30,000 ECPUs/秒（90,000 ECPUs/秒，當使用複本讀取時）。我們建議您評估金鑰並請求在插槽之間的分佈，並確保在請求率超過此限制時分佈一致。

## 對 ElastiCache Serverless 中的限流問題進行故障診斷

在服務導向的架構和分散式系統中，限制各種服務元件處理API呼叫的速率稱為限流。這可平滑尖峰、控制元件輸送量不相符，並在發生非預期的操作事件時允許更可預測的復原。ElastiCache Serverless 專為這些類型的架構而設計，且大多數 Valkey 或 Redis OSS用戶端已針對限流請求內建重試。某種程度的限流對應用程式來說不一定是問題，但資料工作流程中對延遲敏感的部分如果持續遭到限流，可能會對使用者體驗造成負面影響，並降低系統的整體效率。

當請求在 ElastiCache Serverless 中限流時，您應該會看到 `ThrottledRequests` [ElastiCache Serverless](#) 指標增加。如果您注意到有大量限流請求，請考慮下列事項：

- **擴展速度：**ElastiCache 當您擷取更多資料或提高請求率時，無伺服器會自動擴展。如果您的應用程式擴展速度比無 ElastiCache 伺服器擴展的速度快，那麼您的請求可能會受到限制，而無 ElastiCache 伺服器擴展可能會適應您的工作負載。無 ElastiCache 伺服器通常可以快速增加儲存體大小，最多需要 10-12 分鐘的時間來將快取中的儲存體大小加倍。
- **金鑰和請求的均勻分佈：**ElastiCache 在搭配 Valkey 或 Redis 的情況下OSS，每個插槽的金鑰或請求分佈不平均可能會導致熱插槽。如果單一插槽的請求率超過每秒 30,000 ECPUs個，在執行簡單 SET/GET 命令的工作負載中，熱插槽可能會導致請求限流。
- **從複本讀取：**如果您的應用程式允許，請考慮使用「從複本讀取」功能。大多數 Valkey 或 Redis OSS用戶端可以設定為「規模讀取」，將讀取導向複本節點。此功能可讓您擴展讀取流量。此外，無 ElastiCache 伺服器會自動將從複本請求讀取路由至與您應用程式相同可用區域中的節點，進而降低延遲。啟用從複本讀取時，對於使用簡單 SET/GET 命令的工作負載，您可以在單一插槽ECPUs上達到每秒高達 90,000 個。

## 持續性連線問題

使用 疑難排解持續性連線問題時，必須驗證下列項目 ElastiCache：

### 主題

- [安全群組](#)
- [網路 ACLs](#)
- [路由表](#)
- [DNS 解析度](#)
- [透過伺服器端診斷識別問題](#)
- [網路連線能力驗證](#)
- [網路相關限制](#)

- [CPU 用量](#)
- [從伺服器端終止連線](#)
- [Amazon EC2執行個體的用戶端疑難排解](#)
- [解析完成單個請求所花費的時間](#)

## 安全群組

安全群組是保護 ElastiCache 用戶端（EC2執行個體、AWS Lambda 函數、Amazon ECS容器等）和 ElastiCache 快取的虛擬防火牆。安全群組具有狀態，這表示允許傳入或傳出流量之後，該流量的回應將在該特定安全群組的內容中自動授權。

具狀態功能需要安全群組追蹤所有授權的連線，而且追蹤的連線數有限額。如果達到限額，新的連線將會失敗。請參閱疑難排解一節，以取得如何識別用戶端或 ElastiCache 端是否達到限制的說明。

您可以同時將單一安全群組指派給用戶端和 ElastiCache 叢集，或為每個群組指派個別安全群組。

對於這兩種情況，您需要允許來自來源的 ElastiCache 連接埠上的TCP傳出流量，以及相同連接埠上的傳入流量到 ElastiCache。Memcached 的預設連接埠為 11211，Valkey 或 Redis 為 6379OSS。根據預設，安全群組允許所有對外流量。在此情況下，只需要目標安全群組中的傳入規則。

如需詳細資訊，請參閱[存取 Amazon 中 ElastiCache 叢集的存取模式VPC](#)。

## 網路 ACLs

網路存取控制清單（ACLs）是無狀態規則。必須允許雙向（傳入和傳出）的流量才能成功。網路ACLs 會指派給子網路，而非特定資源。可以將相同的 ACL 指派給 ElastiCache 和 用戶端資源，特別是當它們位於相同的子網路中時。

根據預設，網路ACLs允許所有交易。不過，您可以自訂為拒絕或允許流量。此外，ACL規則的評估是依序的，這表示符合流量的最低數字的規則將允許或拒絕它。允許 Valkey 或 Redis OSS流量的最小組態為：

用戶端網路ACL：

- 傳入規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- 類型：自訂TCP規則；
- 通訊協定：TCP
- Port Range (連接埠範圍)：1024-65535



- 來源：0.0.0.0/0 (或建立 ElastiCache 叢集子網路的個別規則)
- Allow/Deny (允許/拒絕)：Allow (允許)
  
- 傳出規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- 類型：自訂TCP規則；
- 通訊協定：TCP
- Port Range (連接埠範圍)：6379
- 來源：0.0.0.0/0 (或 ElastiCache 叢集子網路。請記住，使用特定 IPs 可能會在容錯移轉或擴展叢集時產生問題)
- Allow/Deny (允許/拒絕)：Allow (允許)

#### ElastiCache 網路ACL：

- 傳入規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- 類型：自訂TCP規則；
- 通訊協定：TCP
- Port Range (連接埠範圍)：6379
- 來源：0.0.0.0/0 (或建立 ElastiCache 叢集子網路的個別規則)
- Allow/Deny (允許/拒絕)：Allow (允許)
  
- 傳出規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- 類型：自訂TCP規則；
- 通訊協定：TCP
- Port Range (連接埠範圍)：1024-65535
- 來源：0.0.0.0/0 (或 ElastiCache 叢集子網路。請記住，使用特定 IPs 可能會在容錯移轉或擴展叢集時產生問題)
- Allow/Deny (允許/拒絕)：Allow (允許)

如需詳細資訊，請參閱[網路 ACLs](#)。

## 路由表

與 Network 類似ACLs，每個子網路可以有不同的路由表。如果用戶端和 ElastiCache 叢集位於不同的子網路中，請確保其路由表允許它們彼此互相連接。

涉及多個 VPCs、動態路由或網路防火牆的更複雜環境可能難以進行故障診斷。請參閱 [網路連線能力驗證](#)，確認您的網路設定是否適當。

## DNS 解析度

ElastiCache 會根據DNS名稱提供服務端點。可用的端點為 Configuration、Primary、Reader 及 Node 端點。如需詳細資訊，請參閱 [尋找連線端點](#)。

在修改容錯移轉或叢集的情況下，與端點名稱相關聯的地址可能會變更且會自動更新。

自訂DNS設定（即不使用 VPCDNS服務）可能不知道 ElastiCache提供DNS的名稱。請確定您的系統可以使用系統工具 dig（如下所示）或成功解析 ElastiCache 端點nslookup。

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
1.2.3.4
```

您也可以透過 VPCDNS服務強制名稱解析：

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

## 透過伺服器端診斷識別問題

CloudWatch 來自 ElastiCache 引擎的指標和執行期資訊是常見的來源或資訊，用於識別連線問題的潛在來源。優質的分析資料通常從下列項目開始：

- CPU 用量：Valkey 和 Redis OSS是多執行緒應用程式。但是每個命令的執行發生在單個（主）執行序中。因此，ElastiCache 會提供指標 CPUUtilization和 EngineCPUUtilization。EngineCPUUtilization提供專用於 Valkey 或 Redis 程序的使用CPU率OSS，以及所有 CPUUtilization的使用率vCPUs。具有多個 vCPU 的節點通常具有不同的 CPUUtilization和值EngineCPUUtilization，第二個值通常較高。高的原因EngineCPUUtilization可能是請求數量增加，或需要大量CPU時間才能完成的複雜操作。您可以透過以下項目來識別兩者：
  - 請求數增加：檢查其他符合 EngineCPUUtilization 模式的指標是否增加。有用的指標包括：

- CacheHits 和 CacheMisses：成功請求或沒有在快取中找到有效項目的請求之數目。如果未命中與命中的比率很高，表示應用程式正在對無效果的請求浪費時間和資源。
- SetTypeCmds 和 GetTypeCmds：這些指標與 EngineCPUUtilization 相關，有助於了解寫入請求 (由 SetTypeCmds 測量) 或讀取請求 (由 GetTypeCmds 測量) 的負載是否明顯高。如果負載主要是讀取作業，使用多個僅供讀取複本可以在多個節點之間平衡請求，並撥出主節點用於寫入作業。在停用叢集模式的叢集中，可以使用 ElastiCache 讀取器端點在應用程式中建立額外的連線組態，以使用僅供讀取複本。如需詳細資訊，請參閱 [尋找連線端點](#)。讀取作業必須提交至這個額外的連線。寫入作業將透過一般主要端點完成。在啟用叢集模式的情況下，建議您使用原生支援僅供讀取複本的程式庫。使用正確的旗標，程式庫將能夠自動探索叢集拓撲、複本節點、透過 [READONLY](#) Valkey 或 Redis OSS 命令啟用讀取操作，並將讀取請求提交至複本。
- 連線數增加：
  - CurrConnections 和 NewConnections：CurrConnection 是收集資料點時的已建立連線數，而 NewConnections 會顯示這段期間內建立的連線數。

建立和處理連線意味著高額CPU的額外負荷。此外，建立新連線所需的TCP三向交握會對整體回應時間產生負面影響。

NewConnections 每分鐘數千個 ElastiCache 節點表示只有幾個命令建立和使用連線，這並非最佳狀態。保持連線的成立狀態，並重複使用這些連線來進行新作業是最佳實務。當用戶端應用程式支援並妥善實作連線集區或持續連線時，此情況有可能實現。使用連線集區時，currConnections 數不會有很大的變化，而 NewConnections 應盡可能壓低。Valkey 和 Redis 使用少量 OSS 提供最佳效能 currConnections。currConnection 保持數十或數百的順序，可將資源的使用降至最低，以支援用戶端緩衝區和CPU循環等個別連線，以提供連線。

- 網路輸送量：
  - 判斷頻寬：ElastiCache 節點具有與節點大小成比例的網路頻寬。由於應用程式具有不同的特性，結果可能會根據工作負載而有所不同。例如，具有高速率小請求的應用程式往往影響比網路輸送量更多的CPU用量，而較大的金鑰會導致更高的網路使用率。因此，建議您使用實際工作負載來測試節點，以便更深入了解限制。

模擬來自應用程式的負載會提供更準確的結果。但是，基準化分析工具可讓您更妥善了解限制。

- 針對請求主要都是讀取的情況，使用複本進行讀取作業將能減輕主節點上的負載。如果使用案例主要是寫入，則使用許多複本將會提高網路使用量。針對寫入主節點的每個位元組，系統會將 N 個位元組傳送到複本 (N 即複本的數量)。寫入密集型工作負載的最佳實務是搭配啟用叢集模式的 ElastiCache (Redis OSS) 使用，因此可以平衡多個碎片的寫入，或擴展到具有更多網路功能的節點類型。

- 和 CloudWatchmetrics NetworkBytesIn分別NetworkBytesOut提供進出節點的資料量。ReplicationBytes是專用於資料複寫的流量。

如需詳細資訊，請參閱[網路相關限制](#)。

- 複雜命令：Redis OSS命令會在單一執行緒上提供，這表示請求會依序提供。單一緩慢命令可能會影響其他請求和連線，最終導致逾時。使用作用於多個值、索引鍵或資料類型的命令時必須小心。根據參數的數量或其輸入或輸出值的大小，可能會封鎖或終止連線。

一個惡名昭彰的例子是 KEYS 命令。它會掃描整個 Keyspace，搜尋指定的模式，並在其執行過程中阻止其他命令的執行。Redis OSS使用「Big O」符號來描述其命令複雜性。

索引鍵命令帶有  $O(N)$  時間複雜度， $N$  是資料庫中索引鍵的數量。因此，索引鍵數越多，命令執行速度就會越慢。KEYS 可能會以不同的方式造成問題：如果沒有使用搜尋模式，該命令將會傳回所有可用的索引鍵名稱。在具有數千個或數百萬個項目的資料庫中，系統會建立巨大的輸出並淹沒網路緩衝區。

如果使用搜尋模式，則只會將符合該模式的索引鍵傳回用戶端。但是，引擎仍然會掃描整個 Keyspace 搜尋它，且完成命令的時間相同。

KEYS 的替代方案是 SCAN 命令。它會反覆運算 Keyspace，限制在特定數量的項目中執行反覆運算，避免延長引擎上區塊的執行時間。

掃描具有 COUNT 參數，用來設定反覆運算區塊的大小。預設值為 10 (每個反覆運算 10 個項目)。

根據資料庫中項目數的不同，COUNT 值小的區塊需執行更多的反覆運算才能完成完整掃描，而較大的值每次反覆運算時會讓引擎維持忙碌狀態更久。小計數值會使 SCAN 執行速度較慢，而較大的值可能會對 KEYS 造成上述相同問題。

範例是以計數值 10 執行 SCAN 命令，將需要在具有 1 百萬個索引鍵的資料庫上進行 100,000 次重複作業。如果平均網路封包來回時間為 0.5 毫秒，則傳輸請求會花費大約 50,000 毫秒 (50 秒)。

另一方面，如果計數值為 100,000，則需要執行單次反覆運算，且傳輸只會花 0.5 毫秒。不過，引擎會完全封鎖其他作業，直到該命令完成掃描所有的 Keyspace。

除了 KEYS 以外，如果未正確使用，其他幾個命令可能有害。若要查看所有命令及其個別時間複雜性的清單，請前往 [Valkey 和 Redis OSS命令](#)。

潛在問題範例：

- Lua 指令碼：Valkey 和 Redis OSS提供內嵌 Lua 解譯器，允許在伺服器端執行指令碼。Valkey 和 Redis 上的 Lua 指令碼OSS是在引擎層級上執行，且依定義為原子，這表示指令碼正在執行時不允許執行其他命令或指令碼。Lua 指令碼提供直接在引擎上執行多個命令、決策演算法、資料剖析等的可能性。雖然這類指令碼的不可部分完成特性和卸載應用程式的機會很誘人，但將指令碼用於小型作業時必須小心。在上 ElastiCache，Lea 指令碼的執行時間限制為 5 秒。未寫入 Keyspace 的指令碼將在 5 秒過後自動終止。為了避免資料損毀和不一致，如果指令碼執行沒有在 5 秒內完成，且執行期間有任何寫入，節點便會容錯移轉。[交易](#)是確保 Redis 中多個相關金鑰修改一致性的替代方案OSS。交易允許執行一個命令區塊，觀察現有的索引鍵修改項目。如果觀察的任何索引鍵在交易完成之前有所變更，系統會捨棄所有修改。
- 大量刪除項目：DEL 命令接受多個參數，這些參數是要刪除的索引鍵名稱。刪除操作是同步的，如果參數清單很大，或包含大清單、集、排序集或雜湊（包含數個子項目的資料結構），則需要很長CPU的時間。換句話說，如果單一索引鍵有許多元素，即使只是刪除單一索引鍵，也可能需要花相當長的時間。的替代方案DEL是 UNLINK，這是可用的非同步命令，因為 Redis OSS 4。UNLINK 必須DEL盡可能優先於。從 ElastiCache（Redis OSS）6x 開始，lazyfree-lazy-user-del 參數會讓DEL命令的行為與啟用UNLINK時類似。如需詳細資訊，請參閱 [Redis OSS 6.0 參數變更](#)。
- 對多個索引鍵上產生作用的命令：先前提到的 DEL 是接受多個參數的命令，且其執行時間將直接與此成正比。不過，Redis OSS提供更多類似運作的命令。範例為 MSET 和 MGET 允許一次插入或檢索多個字串索引鍵。使用它們可能有助於減少許多個別 SET 或 GET 命令的固有網路延遲。不過，廣泛的參數清單會影響CPU用量。

雖然僅CPU利用率不是造成連線問題的原因，但花太多時間處理多個金鑰上的單一或少量命令，可能會導致其他請求失敗並提高整體CPU使用率。

索引鍵的數量和大小會影響命令的複雜性，進而影響完成時間。

可以作用於多個索引鍵的其他命令範例

有：HMGET、HMSET、MSETNX、PFCOUNT、PFMERGE、SDIFF、SDIFFSTORE、SINTER、SINTERSTORE 或 ZINTERSTORE。

- 對多種資料類型執行的命令：Redis OSS也提供對一或多個金鑰執行動作的命令，無論其資料類型為何。ElastiCache（Redis OSS）提供指標KeyBasedCmds來監控這類命令。此指標會加總在所選期間內執行下列命令的次數：
  - O(N) 複雜性：
    - KEYS
  - O(1)
    - EXISTS

- OBJECT
  - PTTL
  - RANDOMKEY
  - TTL
  - TYPE
  - EXPIRE
  - EXPIREAT
  - MOVE
  - PERSIST
  - PEXPIRE
  - PEXPIREAT
  - UNLINK (O(N) 用於回收記憶體。但是，記憶體回收任務發生在單獨一個執行序中，且不會阻止引擎
- 根據資料類型而不同的複雜性時間：
    - DEL
    - DUMP
    - RENAME 視為具有 O(1) 複雜性的命令，但在 DEL 內部執行。執行時間會依重新命名索引鍵的大小而變動。
    - RENAMENX
    - RESTORE
    - SORT
  - 大雜湊：雜湊是一種資料類型，允許單個索引鍵具有多個鍵值子項目。每個雜湊都可以存放 4.294.967.295 個項目，且大雜湊上的作業可能會變得非常昂貴。與 KEYS 類似，雜湊具有帶有 O(N) 時間複雜度的 HKEYS 命令，N 是雜湊中的項目數。HSCAN 必須優先於 HKEYS 以避免長時間執行命令。HDEL、HGETALL、HMGET、HMSET 和 HVALS 命令應在大雜湊上謹慎使用。
  - 其他大數據結構：除了雜湊之外，其他資料結構也可能是密集CPU的。集合、清單、排序集合和 Hyperloglog 也可能需要花大量時間來操作，取決於其大小和使用的命令。如需這些命令的詳細資訊，請參閱 [Valkey 和 Redis OSS命令](#)。



## 網路連線能力驗證

檢閱與DNS解析度、安全群組ACLs、網路和路由表相關的網路組態後，可以使用可VPC連線性分析器和系統工具驗證連線。

Reachability Analyzer 會測試網路連線能力，並確認是否滿足所有的需求和許可。針對下列測試，您將需要中其中一個可用 ElastiCache 節點的 ENI ID（彈性網路介面識別）VPC。若要取得此資料，請執行下列操作：

1. 前往 <https://console.aws.amazon.com/ec2/v2/home?#NIC>：
2. 根據 ElastiCache 叢集名稱或先前DNS驗證的 IP 地址來篩選介面清單。
3. 寫下或以其他方式儲存 ENI ID。如果顯示多個介面，請檢閱描述以確認它們屬於正確的 ElastiCache 叢集，然後選擇其中一個。
4. 繼續下一個步驟。
5. 在家中 <https://console.aws.amazon.com/vpc/建立分析路徑?#ReachabilityAnalyzer> 並選擇下列選項：
  - 來源類型：如果您的 ElastiCache 用戶端在 Amazon EC2執行個體上執行，或者如果用戶端使用其他服務，例如 AWS Fargate Amazon ECS搭配 awsvpc 網路等），以及各自的資源 ID（EC2執行個體或 ENI ID）AWS Lambda，請選擇執行個體；
  - 目的地類型：選擇網路介面，然後從清單中選擇 ElasticacheENI。
  - 目的地連接埠：為 ElastiCache（Redis OSS）指定 6379，或為 ElastiCache（Memcached）指定 11211。這些是使用預設組態定義的連接埠，此範例假設未加以變更。
  - 通訊協定：TCP

建立分析路徑，並等待一段時間產生結果。如果狀態為無法連線，請開啟分析詳細資料並檢閱分析總管，取得遭封鎖請求的詳細資訊。

如果通過可連線性測試，請繼續執行作業系統層級的驗證。

若要驗證 ElastiCache 服務連接埠上的TCP連線：在 Amazon Linux 上，Nping 可在套件中使用，nmap並可測試 ElastiCache 連接埠上的TCP連線，並提供建立連線的網路往返時間。使用此選項來驗證網路連線和 ElastiCache 叢集目前的延遲，如下所示：

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com

Starting Nping 0.6.40 (http://nmap.org/nping) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
```

```
(Output suppressed)
```

```
Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

根據預設，nping 每傳送 5 次探查會延遲 1 秒。您可以使用「-c」選項來增加探查數，並使用「--delay」來變更傳送新測試的時間。

如果測試nping失敗且通過VPC了連線能力分析工具測試，請您的系統管理員檢閱可能的主機型防火牆規則、非對稱路由規則，或作業系統層級的任何其他可能限制。

在 ElastiCache 主控台上，檢查 ElastiCache 叢集詳細資訊中是否已啟用傳輸中加密。如果啟用傳輸中加密，請確認是否可以使用下列命令建立 TLS 工作階段：

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

如果連線和 TLS 交涉成功，預期會產生大量輸出。檢查最後一行中可用的傳回碼，該值必須是 0 (ok)。如果 openssl 傳回不同內容，請在 <https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS> 檢查錯誤的原因。

如果所有基礎設施和作業系統測試都通過，但您的應用程式仍無法連線至 ElastiCache，請檢查應用程式組態是否符合 ElastiCache 設定。常見的錯誤包括：

- 您的應用程式不支援 ElastiCache 叢集模式，且 ElastiCache 已啟用叢集模式；
- 您的應用程式不支援 TLS/SSL，且 ElastiCache 已啟用傳輸中加密；
- 應用程式支援 TLS/SSL，但沒有正確的組態旗標或受信任的憑證授權機構；

## 網路相關限制

- 連線數上限：同時連線數具有硬性限制。每個 ElastiCache 節點允許所有用戶端之間最多 65,000 個同時連線。此限制可以透過上的 CurrConnections 指標進行監控 CloudWatch。不過，用戶端也有自己的傳出連線限制。在 Linux 上，請使用下列命令檢查允許的暫時性連接埠範圍：

```
sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

在之前的範例中，將允許 28231 連線從相同的來源到相同的目的地 IP (ElastiCache 節點) 和連接埠。下列命令顯示特定 ElastiCache 節點的連線數量 (IP 1.2.3.4)：



```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

如果數字太高，您的系統可能會因為嘗試處理連線請求而過載。建議考慮實作適用技術，例如連線集區或持續連線，以便更妥善處理連線。盡可能設定連線集區，將連線數上限限制為幾百個。此外，用於處理逾時或其他連線例外狀況的退避邏輯是明智的做法，可避免發生問題的情況下連線流失。

- 網路流量限制：檢查下列 [CloudWatch Redis 指標OSS](#)，以識別節點上可能命中的網路限制 ElastiCache：
  - NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded：因為輸送量超過彙總頻寬限制而形成的網路封包。

請務必注意，寫入主節點的每個位元組都會複寫到 N 個複本 (N 是複本數)。具有小型節點類型、多個複本以及密集寫入型請求的叢集，可能無法處理複寫待處理項目。對於這種情況，最佳實務是擴充規模 (更改節點類型)、水平擴展 (在啟用叢集模式的叢集中增加碎片)，減少複本數或最大限度減少寫入次數。

- NetworkContrackAllowanceExceeded：因為超過指派給節點的連線數 (在所有安全群組之間追蹤) 上限而形成的封包。這段期間內，新的連線可能會失敗。
- NetworkPackets PerSecondAllowanceExceeded：超過每秒的封包數上限。以高極小型請求率為基礎的工作負載，可能會在達到頻寬上限之前達到此限制。

上述指標是確認節點達到網路限制的理想方式。但是，限制也可以藉由網路指標維持平穩來識別。

如果長時間觀察到高原期，則可能會隨後出現複寫延遲、用於快取的位元組增加、釋放可用記憶體減少、高交換和CPU用量。Amazon EC2執行個體也有網路限制，可透過[ENA驅動程式指標](#)追蹤。具有增強型聯網支援和ENA驅動程式 2.2.10 或更新版本的 Linux 執行個體可以使用命令檢閱限制計數器：

```
ethtool -S eth0 | grep "allowance_exceeded"
```

## CPU 用量

CPU 使用量指標是調查的起點，下列項目有助於縮小 ElastiCache 可能的問題範圍：

- Redis OSS SlowLogs：ElastiCache 預設組態會保留最後 128 個命令，這些命令需要超過 10 毫秒的時間才能完成。慢速命令的歷史記錄會在引擎執行時間內保留，且會在故障或重新啟動的情況下遺失。如果清單達到 128 個項目，將會刪除舊活動，為新的活動騰出空間。慢速事件清單的大小和視

為緩慢的執行時間，可以透過[自訂參數群組](#)中的 `slowlog-max-len` 和 `slowlog-log-slower-than` 參數加以修改。慢速日誌清單可以透過在引擎上執行 `SLOWLOG GET 128` 來擷取，其中 128 是最後回報的 128 個慢速命令。每個項目都有下列欄位：

```

1) 1) (integer) 1 -----> Sequential ID
 2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
 3) (integer) 4823378 -----> Time in microseconds to complete the command.
 4) 1) "keys" -----> Command
 2) "*" -----> Arguments
 5) "1.2.3.4:57004"-> Source

```

上述事件發生於 12 月 26 日 19:26:07UTC，需要 4.8 秒（4.823 毫秒）才能完成，並且是由用戶端 1.2.3.4 請求的 KEYS 命令所造成。

在 Linux 上，時間戳記可以使用命令日期轉換：

```

$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020

```

使用 Python：

```

>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)

```

或使用的 Windows 上 PowerShell：

```

PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
DateTime : 12/26/2020 7:26:07 PM
UtcDateTime :
 : 12/26/2020 7:26:07 PM
LocalDateTime : 12/26/2020 2:26:07 PM
Date : 12/26/2020 12:00:00 AM
Day : 26
DayOfWeek : Saturday
DayOfYear : 361
Hour : 19
Millisecond : 0
Minute : 26
Month :

```

```

 : 12
Offset : 00:00:00Ticks : 637446075670000000
UtcTicks
 : 637446075670000000
TimeOfDay : 19:26:07
Year : 2020

```

在短時間內 (一分鐘以內) 出現許多緩慢命令是值得關注的原因。檢閱命令的性質及如何最佳化 (請參閱前面的範例)。如果經常報告具有 O ( 1 ) 時間複雜性的命令，請檢查上述高CPU用量的其他因素。

- 延遲指標：ElastiCache ( RedisOSS ) 提供 CloudWatch 指標來監控不同類別命令的平均延遲。資料點的計算方式是將類別中命令的執行總次數除以期間的總執行時間。請務必了解，延遲指標結果是多個命令的彙總結果。單一命令可能會導致非預期的結果 (例如逾時)，而不會對指標造成重大影響。針對這種情況，慢速事件會是更準確的資訊來源。下列清單包含可用的延遲指標，以及影響這些指標的各別命令。
  - EvalBasedCmdsLatency：與 Lua Script 命令、eval、相關evalsha；
  - GeoSpatialBasedCmdsLatency: geodist, geohash, geopos, georadius, georadiusbymember, geoadd;
  - GetTypeCmdsLatency：讀取命令，無論資料類型為何；
  - HashBasedCmdsLatency: hexists, hget, hgetall, hkeys, hlen, hmget, hvals, hstrlen, hdel, hincrby, hincrbyfloat, hmset, hset, hsetnx;
  - HyperLogLogBasedCmdsLatency: pfselftest, pfcount, pfdebug, pfadd, pfmerge;
  - KeyBasedCmdsLatency：可對不同資料類型採取動作的命令：dump、exists、keys、object、pttl、randomkey、ttl、typedel、expire、expireat、
  - ListBasedCmdsLatency：
    - lindex、llen、lrange、lpop、brpop、brpoplpush、linsert、lpop、lpush、lpushx、lrem、lset、ltrim、rpop
  - PubSubBasedCmdsLatency：
    - psubscribe、publish、pubsub、punsubscribe、subscribe、subunsubscribe；
  - SetBasedCmdsLatency: scard, sdiff, sinter, sismember, smembers, srandmember, sunion, sadd, sdiffstore, sinterstore, smove, spop, srem, sunionstore;
  - SetTypeCmdsLatency：寫入命令，無論資料類型為何；
  - SortedSetBasedCmdsLatency: zcard, zcount, zrange, zrangebyscore, zrank, zrevrange, zrevrangebyscore, zrevrank, zscore, zrangebylex, zrevrangebylex, zlexcount, zadd, zincrby, zinterstore, zrem, zremrangebyrank,

zremrangebyscore, zunionstore, zremrangebylex, zpopmax, zpopmin, bzpopmin, bzpopmax;

- StringBasedCmdsLatency: bitcount, get, getbit, getrange, mget, strlen, substr, bitpos, append, bitop, bitfield, decr, decrby, getset, incr, incrby, incrbyfloat, mset, msetnx, psetex, set, setbit, setex, setnx, setrange;
- StreamBasedCmdsLatency: xrange, xrevrange, xlen, xread, xpending, xinfo, xadd, xgroup, readgroup, xack, xclaim, xdel, xtrim, xsetid;
- Redis OSS執行期命令：
  - info commandstats：提供自 Redis OSS引擎啟動以來執行的命令清單、累積執行次數、總執行時間和每個命令的平均執行時間；
  - client list：提供目前連線的用戶端和相關資訊的清單，如緩衝區使用率、上次執行的命令等；
- 備份和複寫：2.8.22 之前的 ElastiCache ( Redis OSS ) 版本使用叉式程序來建立備份，並處理與複本的完整同步。這種方法可能會在寫入密集型使用案例中產生顯著的記憶體額外負荷。

從 ElastiCache Redis OSS 2.8.22 開始，AWS 推出了無叉備份和複寫方法。新方法可能會延遲寫入以防止失敗。這兩種方法都可能導致CPU使用率提高，導致回應時間提高，進而在其執行期間導致用戶端逾時。請一律檢查是否在備份期間內發生用戶端故障或 SaveInProgress 指標為 1 的情況。建議您將備份時段排定為使用率低的時段，盡量減少用戶端發生問題或備份失敗的可能性。

## 從伺服器端終止連線

預設 ElastiCache ( Redis OSS ) 組態會無限期建立用戶端連線。不過在某些情況下可能需終止連線。例如：

- 用戶端應用程式中的錯誤可能會造成連線遭遺忘，並以閒置狀態保持成立。這就是所謂的「連線流失」，結果會發現 CurrConnections 指標中已成立的連線數穩定增加。此行為可能會導致用戶端或 ElastiCache 端的飽和。當無法從用戶端立即修正時，有些管理員會在其 ElastiCache 參數群組中設定「逾時」值。逾時是讓閒置連線持續存在的時間 (以秒為單位)。如果用戶端在此期間未提交任何請求，則 Redis OSS引擎會在連線達到逾時值時立即終止連線。較小的逾時值可能會導致不必要的連線中斷，屆時用戶端需要妥善處理並重新連線，進而導致延遲。
- 用於存放索引鍵的記憶體與用戶端緩衝區共用。具有大型請求或回應的慢速用戶端，可能需要大量的記憶體來處理其緩衝區。預設 ElastiCache ( Redis OSS ) 組態不會限制一般用戶端輸出緩衝區的大小。如果達到 maxmemory 上限，引擎將會嘗試移出項目以滿足緩衝區使用量。在極端低記憶體情況下，ElastiCache ( Redis OSS ) 可能會選擇中斷使用大型用戶端輸出緩衝區的用戶端連線，以釋放記憶體並保留叢集的運作狀態。

可以使用自訂組態來限制用戶端緩衝區的大小，且用戶端達到限制時將會中斷連線。不過用戶端應該能處理未預期的中斷連線。處理一般用戶端的緩衝區大小的參數如下：

- `client-query-buffer-limit`：單一輸入請求的大小上限；
- `client-output-buffer-limit-normal-soft-limit`：用戶端連線的軟限制。如果超過軟限制的時間超過在上定義的秒數 `client-output-buffer-limit-normal-soft-seconds` 或者達到硬限制，則連線將會終止；
- `client-output-buffer-limit-normal-soft-seconds`：超過 `client-output-buffer-limit-normal-soft-limit` 的連線允許的時間 `normal-soft-limit`；
- `client-output-buffer-limit-normal-hard-limit`：達到此限制的連線會立即終止。

除了一般用戶端緩衝區，下列選項也可控制複本節點和 Pub/Sub (發佈/訂閱) 用戶端的緩衝區：

- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-replica-soft-seconds`;
- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-pubsub-soft-limit`;
- `client-output-buffer-limit-pubsub-soft-seconds`;
- `client-output-buffer-limit-pubsub-hard-limit`;

## Amazon EC2執行個體的用戶端疑難排解

用戶端的負載和回應能力也會影響對的請求 ElastiCache。EC2 執行個體和作業系統限制需要仔細檢閱，同時疑難排解間歇性連線或逾時問題。幾項需觀察的要點：

- CPU:
  - EC2 執行個體CPU用量：確定 CPU 尚未飽和或接近 100%。歷史分析可以透過完成 CloudWatch，但請記住，資料點精細度為 1 分鐘（啟用詳細監控）或 5 分鐘；
  - 如果使用[爆量EC2執行個體](#)，請確定其CPU點數餘額尚未耗盡。此資訊可在 CPUcreditBalance CloudWatch 指標上取得。
  - 短期高CPU用量可能會導致逾時，而不會反映上的 100% 使用率 CloudWatch。這種情況需使用作業系統工具 (如 `top`、`ps` 和 `mpstat`) 進行即時監控。
- 網路
  - 根據執行個體容量檢查網路輸送量是否低於可接受的值。如需詳細資訊，請參閱 [Amazon EC2執行個體類型](#)

- 在具有 ena 增強型網路驅動程式的執行個體上，查看 [ENA 統計數字](#) 以取得逾時或超出的限制。下列統計數字有助於確認網路限制飽和情況：
  - `bw_in_allowance_exceeded/bw_out_allowance_exceeded`：因過量的傳入或傳出輸送量而造成的封包數；
  - `contrack_allowance_exceeded`：由於安全群組 [連線追蹤限制](#) 而捨棄的封包數。當此限制飽和時，新的連線將會失敗；
  - `linklocal_allowance_exceeded`：透過 對執行個體中繼資料提出過多請求而捨棄 NTPVPC 的封包數量 DNS。所有服務的限制都是每秒 1024 個封包；
  - `pps_allowance_exceeded`：由於每秒封包比率過高而捨棄的封包數。當網路流量每秒包含在數千或數百萬個非常小的請求上時，就會達到此 PPS 限制。ElastiCache 流量可透過管道或命令來最佳化，以便能夠像 一樣同時執行多個操作，MGET 以更好地使用網路封包 GET。

## 解析完成單個請求所花費的時間

- 在網路上：Tcpdump 和 Wireshark ( 命令列上的 tshark ) 是便利的工具，可了解請求需要多少時間才能移動網路、命中 ElastiCache 引擎並取得回報。下列範例說明如何使用下列命令建立單一請求：

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

tcpdump 與上面的命令同時執行並傳回了：

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
 > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
 8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
 53962565, ack 177032945, win
 28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
 length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
 options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
```

```

1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
 1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
 options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
 211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
 7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
 [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel

```

從上面的輸出，我們可以確認TCP三向交握已在 222 微秒（918091 - 917869）內完成，並在 173 微秒（918295 - 918122）內提交並傳回 ping 命令。

從請求到關閉連線花了 438 微秒（918307 - 917869）。這些結果可以確認網路和引擎的回應時間良好，而且可以專注於其他元件進行調查。

- 在作業系統上：Strace 有助於識別作業系統層級的時間落差。實際應用程式的分析會更廣泛，建議使用專門的應用程式剖析工具或除錯工具。下列範例只會顯示基礎作業系統元件是否正常運作，若不正常則可能需要進一步調查。使用相同的 Redis OSSPING 命令，strace 我們可獲得：

```

$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)
 6379
1609430221.697712 (+ 0.000000) execve("/usr/bin/nc", ["nc",
 "example.xxxxxx.ng.0001.use...", "6379"], 0x7ffffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
 (+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
 (+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3

```







# Amazon 的安全性 ElastiCache

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您受益於為滿足最安全敏感組織的需求而建置的資料中心和網路架構。

安全性是 AWS 和 之間的共同責任。[共同責任模型](#) 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。在 [AWS 合規計畫](#) 中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon 的合規計劃 ElastiCache，請參閱[AWS 依合規計劃範圍中的服務](#)。
- 雲端安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 Amazon 時套用共同責任模型 ElastiCache。下列主題說明如何設定 Amazon ElastiCache 以符合您的安全和合規目標。您也會了解如何使用 AWS 其他服務來協助您監控和保護 Amazon ElastiCache 資源。

## 主題

- [Amazon 的資料保護 ElastiCache](#)
- [網際網路流量隱私權](#)
- [Amazon 的身分和存取管理 ElastiCache](#)
- [Amazon 的合規驗證 ElastiCache](#)
- [Amazon 中的復原能力 ElastiCache](#)
- [中的基礎設施安全 AWS ElastiCache](#)
- [中的服務更新 ElastiCache](#)
- [常見漏洞與暴露 \( CVE \) : 中解決的安全漏洞 ElastiCache](#)

## Amazon 的資料保護 ElastiCache

AWS [共同責任模型](#) 適用於 AWS ElastiCache ( ) 中的資料保護 ElastiCache。如本模型所述，AWS 負責保護執行所有 AWS Cloud 的全域基礎設施。您負責維護在此基礎設施上託管內容的控制權。此內容包含您使用之 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權 FAQ](#)。

為了資料保護目的，我們建議您保護 AWS 帳戶憑證，並使用 AWS Identity and Access Management (IAM) 設定個別帳戶 IAM。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 對每個帳戶使用多重要素驗證 (MFA)。
- 使用 TLS 與 AWS 資源通訊。
- 使用設定 API 和使用者活動日誌 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及服務中的所有 AWS 預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Simple Storage Service (Amazon Simple Storage Service (Amazon S3)) 的個人資料。

我們強烈建議您絕對不要將客戶帳戶號碼等敏感的識別資訊，放在自由格式的欄位中，例如 Name (名稱) 欄位。這包括當您使用主控台、ElastiCache 或使用 AWS 或其他 API AWS CLI 服務時 AWS SDKs。您輸入的任何資料 ElastiCache 或其他服務都可能被擷取以包含在診斷日誌中。當您將 URL 提供給外部伺服器時，請勿在 URL 中包含憑證資訊，URL 以驗證您對該伺服器的請求。

## 主題

- [Amazon 中的資料安全 ElastiCache](#)

## Amazon 中的資料安全 ElastiCache

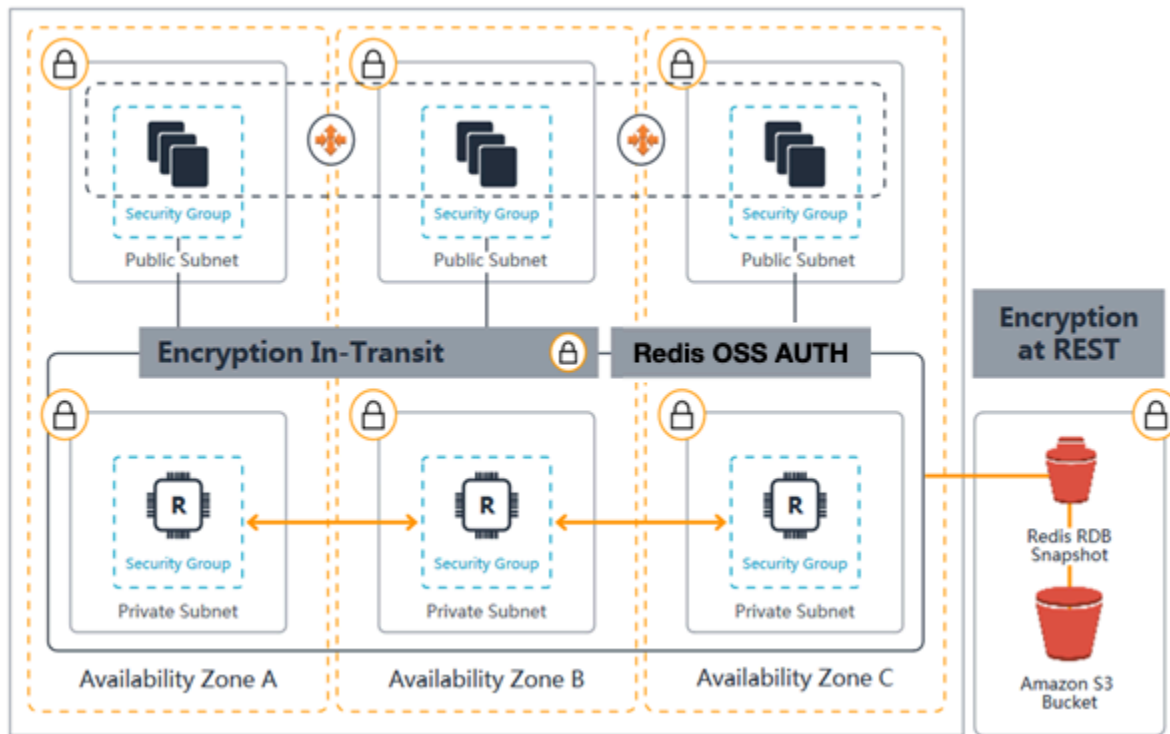
為了協助保護資料安全，Amazon ElastiCache 和 Amazon EC2 提供機制來防止未經授權存取伺服器上的資料。

Amazon ElastiCache (Memcached) 為執行 Memcached 版 1.6.12 或更新版本的快取上的資料提供加密功能。

Amazon ElastiCache with Valkey and Redis 為執行 Valkey 7.2 或更新版本的快取以及 Redis 3.2.6 OSS 版 (排程為 EOL，請參閱 [Redis OSS 版本生命週期結束排程](#))、4.0.10 或更新版本的資料 OSS 提供加密功能。Amazon ElastiCache 也支援使用 IAM 或 Valkey 和 Redis OSS 來驗證使用者 AUTH，並使用角色型存取控制 (IAM) 授權使用者操作 RBAC。

- 傳輸中加密會在您的資料從某一處移到另一處時進行加密，例如在您叢集內的節點之間移動，或是在您的快取與應用程式之間移動。
- 待用加密可在同步和備份操作期間加密您磁碟上的資料。

ElastiCache 支援使用 IAM 和 Valkey 和 Redis OSSAUTH 命令驗證使用者，以及使用角色型存取控制 ( ) 授權使用者操作 RBAC。



ElastiCache 使用 Valkey 和 Redis OSS安全圖表

## 主題

- [ElastiCache 傳輸中加密 \( TLS \)](#)
- [中的靜態加密 ElastiCache](#)
- [身分驗證和授權](#)

## ElastiCache 傳輸中加密 ( TLS )

為了協助保護資料安全，Amazon ElastiCache 和 Amazon EC2 提供機制來防止未經授權存取伺服器上的資料。透過提供傳輸中加密功能，ElastiCache 您可以提供工具，以便在資料從一個位置移至另一個位置時協助保護資料。

所有 Valkey 或 Redis 無OSS伺服器快取都已啟用傳輸中加密。對於自行設計的叢集，您可以在建立複寫群組時將參數設定為 `TransitEncryptionEnabledtrue` ( CLI : `--transit-encryption-enabled` )，以啟用複寫群組的傳輸中加密。無論您是使用 AWS Management Console、或建立複寫群組 AWS CLI，都可以執行此操作 ElastiCache API。

所有無伺服器快取都會啟用傳輸中加密。對於自行設計的叢集，您可以在使用 `TransitEncryptionEnabledtrue` ( CLI : `--transit-encryption-enabled` ) 操作建立快取叢集時，將參數設定為 `CreateCacheCluster` ( CLI : `create-cache-cluster` ) ，以啟用快取叢集的傳輸中加密。

## 主題

- [傳輸中加密概觀](#)
- [傳輸中加密條件 \( Valkey 和 RedisOSS \)](#)
- [傳輸中加密條件 \( Memcached \)](#)
- [傳輸中加密最佳實務](#)
- [其他 Valkey 和 Redis OSS選項](#)
- [啟用 Memcached 的傳輸中加密](#)
- [啟用傳輸中加密](#)
- [使用 valkey-cli 使用傳輸中加密連線至 ElastiCache \( Valkey \) 或 Amazon ElastiCache \( RedisOSS \)](#)
- [使用 Python 在自我設計的 Redis OSS叢集上啟用傳輸中加密](#)
- [啟用傳輸中加密時的最佳實務](#)
- [使用 Openssl \( Memcached \) 連線至已啟用傳輸中加密的節點](#)
- [使用 Java 建立 TLS Memcached 用戶端](#)
- [使用 建立 TLS Memcached 用戶端 PHP](#)

## 傳輸中加密概觀

Amazon 傳輸 ElastiCache 中加密是一項功能，可讓您在資料從一個位置傳輸到另一個位置時，提高其最脆弱點的資料安全性。因為要加密和解密端點的資料需要一些處理，啟用傳輸中加密可能會有一些效能影響。您應該對您具有和不具有傳輸中加密的資料進行基準分析，以判斷對您的使用案例的影響。

ElastiCache 傳輸中加密實作下列功能：

- 加密的用戶端連線 — 快取節點的用戶端連線會 TLS 加密。
- 加密的伺服器連線：在叢集中的節點之間移動的資料會經過加密。
- 伺服器身分驗證 - 用戶端可以驗證是否已連線至正確的伺服器。
- 用戶端身分驗證：使用 Valkey 和 Redis OSS AUTH 功能，伺服器可以驗證用戶端。

## 傳輸中加密條件 ( Valkey 和 RedisOSS )

當您規劃自行設計的叢集實作時，應謹記 Amazon 傳輸 ElastiCache 中加密的下列限制：

- 執行 Valkey 7.2 及更新版本以及 Redis 3.2.6、4.0.10 及更新OSS版本的複寫群組支援傳輸中加密。
- 執行 Valkey 7.2 和更新版本以及 Redis 7 和更新OSS版本的複寫群組支援修改現有叢集的傳輸中加密設定。
- 只有 Amazon 中執行的複寫群組才支援傳輸中加密VPC。
- 執行下列節點類型的複寫群組不支援傳輸中加密：M1, M2。

如需詳細資訊，請參閱[支援的節點類型](#)。

- 透過將參數 TransitEncryptionEnabled 明確設定為 true，可啟用傳輸中加密。
- 確保您的快取用戶端支援TLS連線，且您已在用戶端組態中啟用連線。
- 第 TLS 6 TLS 版及更高 ElastiCache 版本的所有 AWS 區域皆已棄用舊 1.0 和 1.1 的使用。ElastiCache 將繼續支援 TLS 1.0 和 1.1，直到 2025 年 5 月 8 日為止。客戶必須在該日期之前更新其用戶端軟體。

## 傳輸中加密條件 ( Memcached )

當您規劃自行設計的叢集實作時，應謹記 Amazon 傳輸 ElastiCache 中加密的下列限制：

- 執行 Memcached 1.6.12 和更新版本的叢集上支援傳輸中加密。
- 傳輸中加密支援 Transport Layer Security ( TLS ) 1.2 和 1.3 版。
- 傳輸中加密僅支援在 Amazon 中執行的叢集VPC。
- 執行下列節點類型的複寫群組不支援傳輸中加密：M1, M2, M3, R3, T2。

如需詳細資訊，請參閱[支援的節點類型](#)。

- 透過將參數 TransitEncryptionEnabled 明確設定為 true，可啟用傳輸中加密。
- 您只能在建立叢集時，在叢集上啟用傳輸中加密。您無法透過修改叢集來開啟和關閉傳輸中加密。
- 確保您的快取用戶端支援TLS連線，且您已在用戶端組態中啟用連線。

## 傳輸中加密最佳實務

- 因為要加密和解密端點的資料需要一些處理，實作傳輸中加密可能會降低效能。對您自己的資料進行傳輸中加密與無加密基準分析的比較，來判斷對您實作的影響。
- 由於建立新連線可能很昂貴，因此您可以透過保留TLS連線來降低傳輸中加密的效能影響。

## 其他 Valkey 和 Redis OSS 選項

如需適用於 Valkey 和 Redis 選項的詳細資訊OSS，請參閱下列連結。

- [中的靜態加密 ElastiCache](#)
- [使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)
- [角色型存取控制 \(RBAC\)](#)
- [Amazon VPCs和安全性 ElastiCache](#)
- [Amazon 的身分和存取管理 ElastiCache](#)

## 啟用 Memcached 的傳輸中加密

若要在使用 AWS 管理主控台建立 Memcached 叢集時，啟用傳輸中加密，請選取以下項目：

- 選擇 Memcached 當作您的引擎。
- 選擇引擎 1.6.12 或更新版本。
- 在 Encryption in transit (傳輸中加密) 中，選擇 Enable (啟用)。

如需 程序 step-by-step，請參閱 [為 Valkey 或 Redis 建立叢集 OSS](#)。

## 啟用傳輸中加密

所有無伺服器快取都會啟用傳輸中加密。在自我設計叢集上，您可以使用 AWS Management Console、AWS CLI或 ElastiCache 啟用傳輸中加密API。

使用 啟用傳輸中加密 AWS Management Console

使用 為新的自我設計叢集啟用傳輸中加密 AWS Management Console

自行設計叢集時，採用「輕鬆建立」方法的「開發/測試」和「生產」組態會使用傳輸中加密。當您自行選擇組態時，請進行下列選擇：

- 選擇引擎版本 3.2.6、4.0.10 或更新版本。
- 按一下傳輸中加密選項的啟用旁的核取方塊。

如需此程序 step-by-step，請參閱下列內容：

- [建立 Valkey \(停用叢集模式\) 叢集 \(主控台\)](#)



- [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)

使用 啟用現有自行設計叢集的傳輸中加密 AWS Management Console

啟用傳輸中加密有兩個步驟，您必須先將傳輸加密模式設為 preferred。此模式可讓您的 Valkey 或 Redis OSS用戶端使用加密和未加密的連線進行連線。遷移所有 Valkey 或 Redis OSS用戶端以使用加密連線後，您可以修改叢集組態，將傳輸加密模式設定為 required。將傳輸加密模式設為 required 會捨棄所有未加密連線，且僅允許加密連線。

將傳輸加密模式設定為偏好

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從 ElastiCache左側導覽窗格中列出的資源中選擇 Valkey 快取或 Redis OSS快取。
3. 選擇您要更新的快取。
4. 選擇 Actions (動作) 下拉式清單，然後選擇 Modify (修改)。
5. 在 Security (安全性) 區段的 Encryption in transit (傳輸中加密) 下選擇 Enable (啟用)。
6. 選擇 Preferred (偏好) 做為 Transit encryption mode (傳輸加密模式)。
7. 選擇 Preview changes (預覽變更)，然後儲存變更。

在您遷移所有 Valkey 或 Redis OSS用戶端以使用加密連線之後：

將傳輸加密模式設定為必要

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從 ElastiCache左側導覽窗格中列出的資源中選擇 Valkey 快取或 Redis OSS快取。
3. 選擇您要更新的快取。
4. 選擇 Actions (動作) 下拉式清單，然後選擇 Modify (修改)。
5. 在 Security (安全性) 區段中，選擇 Required (必要) 做為 Transit encryption mode (傳輸加密模式)。
6. 選擇 Preview changes (預覽變更)，然後儲存變更。

## 使用 啟用傳輸中加密 AWS CLI

若要在使用 建立 Valkey 或 Redis OSS 複寫群組時啟用傳輸中加密 AWS CLI，請使用 參數 `transit-encryption-enabled`。

針對 Valkey 或 Redis OSS ( 停用叢集模式 )，在新的自我設計叢集上啟用傳輸中加密 ( CLI )

使用 AWS CLI 操作 `create-replication-group` 和下列參數來建立具有啟用傳輸中加密的複本的 Valkey 或 Redis OSS 複寫群組：

重要參數：

- **--engine**—必須為 `valkey` 或 `redis`。
- **--engine-version**- 如果引擎是 Redis OSS，則必須為 3.2.6、4.0.10 或更新版本。
- **--transit-encryption-enabled** - 必要項目。如果您啟用傳輸中加密，您也必須為 `--cache-subnet-group` 參數提供一個值。
- **--num-cache-clusters** - 必須至少為 1。此參數的最大值為 6。

如需詳細資訊，請參閱下列內容：

- [從頭開始建立 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [create-replication-group](#)

針對 Valkey 或 Redis ( 已啟用 OSS 叢集模式 )，在新的自我設計叢集上啟用傳輸中加密 ( CLI )

使用 AWS CLI 操作 `create-replication-group` 和下列參數來建立啟用傳輸中加密的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組：

重要參數：

- **--engine**—必須為 `valkey` 或 `redis`。
- **--engine-version**- 如果引擎是 Redis OSS，則必須為 3.2.6、4.0.10 或更新版本。
- **--transit-encryption-enabled** - 必要項目。如果您啟用傳輸中加密，您也必須為 `--cache-subnet-group` 參數提供一個值。
- 使用下列其中一個參數集來指定複寫群組之節點群組的組態：
  - **--num-node-groups** - 指定此複寫群組中碎片 (節點群組) 的數量。此參數的最大值為 500。



**--replicas-per-node-group** - 指定每個節點群組中複本節點的數量。此處指定的值會套用到此複寫群組中的所有碎片。此參數的最大值為 5。

- **--node-group-configuration** - 獨立指定每個碎片的組態。

如需詳細資訊，請參閱下列內容：

- [從頭開始建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [create-replication-group](#)

在使用 AWS CLI 的現有叢集上啟用傳輸中加密

啟用傳輸中加密有兩個步驟，您必須先將傳輸加密模式設為 preferred。此模式可讓您的 Valkey 或 Redis OSS 用戶端使用加密和未加密的連線進行連線。遷移所有 Valkey 或 Redis OSS 用戶端以使用加密連線後，您可以修改叢集組態，將傳輸加密模式設定為 required。將傳輸加密模式設為 required 會捨棄所有未加密連線，且僅允許加密連線。

使用 AWS CLI 操作 `modify-replication-group` 和下列參數來更新停用傳輸中加密的 Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組。

啟用傳輸中加密

1. 使用下列參數 `preferred`，將 `transit-encryption-mode` 設定為
  - **--transit-encryption-enabled** - 必要項目。
  - **--transit-encryption-mode**—必須設定為 `preferred`。
2. 使用下列參數 `required` 將 `transit-encryption-mode` 設定為：
  - **--transit-encryption-enabled** - 必要項目。
  - **--transit-encryption-mode**—必須設定為 `required`。

使用 `valkey-cli` 使用傳輸中加密連線至 ElastiCache ( Valkey ) 或 Amazon ElastiCache ( RedisOSS )

若要從使用傳輸中加密啟用的 ElastiCache ( Redis OSS ) 快取存取資料，您可以使用使用 Secure Socket Layer ( ) 的用戶端 SSL。您也可以在 Amazon Linux 和 Amazon Linux 2 上使用 `valkey-cli` 搭配 TLS/SSL。如果您的用戶端不支援 TLS，您可以使用用戶端主機上的 `stunnel` 命令來建立 Redis OSS 節點的 SSL 通道。

## 與 Linux 的加密連線

若要使用 valkey-cli 連線到啟用傳輸中加密的 Valkey 或 Redis OSS 叢集，請遵循下列步驟：Amazon Linux 2023、Amazon Linux 2 或 Amazon Linux。

1. 下載並編譯 valkey-cli 公用程式。此公用程式包含在 Valkey 軟體分發中。
2. 在 EC2 執行個體的命令提示中，針對您正在使用的 Linux 版本輸入適當的命令。

### Amazon Linux 2023

如果使用 Amazon Linux 2023，請輸入以下內容：

```
sudo yum install redis6 -y
```

然後輸入下列命令，將叢集和連接埠的端點取代為此範例中顯示的內容。

```
valkey-cli -h Primary or Configuration Endpoint --tls -p 6379
```

如需尋找端點的詳細資訊，請參閱[尋找您的節點端點](#)。

### Amazon Linux 2

如果使用 Amazon Linux 2，請輸入以下內容：

```
sudo yum -y install openssl-devel gcc
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean
make valkey-cli BUILD_TLS=yes
sudo install -m 755 src/valkey-cli/usr/local/bin/
```

### Amazon Linux

如果使用 Amazon Linux，請輸入以下內容：

```
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make valkey-cli CC=clang BUILD_TLS=yes
```

```
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

在 Amazon Linux 上，您可能還需要執行下列額外步驟：

```
sudo yum install clang
CC=clang make
sudo make install
```

3. 下載並安裝 valkey-cli 公用程式後，建議您執行選用make-test命令。
4. 若要連線至已啟用加密和身分驗證的叢集，請輸入此命令：

```
valkey-cli -h Primary or Configuration Endpoint --tls -a 'your-password' -p 6379
```

#### Note

如果您在 Amazon Linux 2023 上安裝 redis6，您現在可以使用命令redis6-cli，而不是 valkey-cli：

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

## 使用 stunnel 的加密連線

若要使用 valkey-cli 連線至使用 stunnel 啟用傳輸中加密的 Redis OSS 叢集，請遵循下列步驟。

1. 使用 SSH 連線至您的用戶端並安裝 stunnel。

```
sudo yum install stunnel
```

2. 執行下列命令，以 '/etc/stunnel/valkey-cli.conf' 同時建立和編輯檔案，將 ElastiCache (Redis OSS) 叢集端點新增至一或多個連線參數，並使用下方提供的輸出作為範本。

```
vi /etc/stunnel/valkey-cli.conf

fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
```

```

debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[valkey-cli]
 client = yes
 accept = 127.0.0.1:6379
 connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-cli-replica]
 client = yes
 accept = 127.0.0.1:6380
 connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379

```

在下一個範例中，組態檔案有兩個連線：valkey-cli 和 valkey-cli-replica。參數的設定方式如下：

- client 設為 yes，以指定此 stunnel 執行個體為用戶端。
- accept 設為用戶端 IP。在此範例中，主要設定為連接埠 6379 上的 Redis OSS 預設 127.0.0.1。複本主機必須呼叫不同的連接埠，並設為 6380。您可以使用的暫時性連接埠為 1024–65535。如需詳細資訊，請參閱 Amazon 使用者指南中的[列舉連接埠](#)。VPC
- 連線已設定為 Redis OSS 伺服器端點。如需詳細資訊，請參閱[在中尋找連線端點 ElastiCache](#)。

### 3. 啟動 stunnel。

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

使用 netstat 命令來確認通道已啟動。

```

sudo netstat -tulnp | grep -i stunnel

tcp 0 0 127.0.0.1:6379 0.0.0.0:* LISTEN
3189/stunnel
tcp 0 0 127.0.0.1:6380 0.0.0.0:* LISTEN
3189/stunnel

```

### 4. 使用通道的本機端點連線至加密的 Redis OSS 節點。

- 如果在 ElastiCache (Redis OSS) 叢集建立期間未使用 AUTH 密碼，則此範例會使用 valkey-cli，在 Amazon Linux 上使用 valkey-cli 的完整路徑連線至 ElastiCache (Redis OSS) 伺服器：

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

如果在建立 Redis OSS叢集期間使用AUTH密碼，則此範例會使用 valkey-cli，在 Amazon Linux 上使用 valkey-cli 的完整路徑連線至 Redis OSS 伺服器：

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

或

- 將目錄變更為 redis-7.2.5 並執行下列動作：

如果在 ElastiCache ( Redis OSS ) 叢集建立期間未使用AUTH密碼，則此範例會使用 valkey-cli，在 Amazon Linux 上使用 valkey-cli 的完整路徑連線至 ElastiCache ( Redis OSS ) 伺服器：

```
src/valkey-cli -h localhost -p 6379
```

如果在建立 Redis OSS叢集期間使用AUTH密碼，則此範例會使用 valkey-cli，在 Amazon Linux 上使用 valkey-cli 的完整路徑連線至 Valkey 或 Redis OSS 伺服器：

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

此範例使用 Telnet 連線至 Valkey Redis OSS 伺服器。

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

- 若要停止和關閉SSL通道，pkill則通道程序。

```
sudo pkill stunnel
```

使用 Python 在自我設計的 Redis OSS叢集上啟用傳輸中加密

下列指南將示範如何在最初在停用傳輸中加密的情況下建立的 Redis OSS 7.0 叢集上啟用傳輸中加密。TCP 和 TLS 用戶端將繼續在此過程中與叢集通訊，而不會停機。

Boto3 將從環境變量中取得所需的憑證 (aws\_access\_key\_id、aws\_secret\_access\_key 和 aws\_session\_token)。這些憑證將事先貼到同一個 bash 終端中，我們將在其中執行 python3 以處理本指南中顯示的 Python 程式碼。以下範例中的程式碼是從在相同執行個體中啟動的執行個體處理 VPC，該 EC2 執行個體將用於在執行個體中建立 ElastiCache Redis OSS 叢集。

### Note

- 下列範例使用 boto3 SDK 進行 ElastiCache 管理操作（叢集或使用者建立）和重新清除/進行 redis-py-cluster 資料處理。
- 您必須使用至少 boto3 版本（=）1.26.39，才能將線上 TLS 遷移與叢集修改 搭配使用 API。
- ElastiCache 僅支援具有 Valkey 7.2 版及更新版本或 Redis 7.0 版或更新 OSS 版本的叢集進行線上 TLS 遷移。因此，如果您的叢集執行的 Redis OSS 版本早於 7.0，則需要升級叢集的 Redis OSS 版本。如需這些版本差異的詳細資訊，請參閱 [與 Redis 的主要版本行為和相容性差異 OSS](#)。

### 主題

- [定義將啟動 ElastiCache Valkey 或 Redis OSS 叢集的字串常數](#)
- [定義叢集組態類別](#)
- [定義一個將代表叢集本身的類別](#)
- [（選用）建立包裝程式類別，以示範用戶端連線至 Valkey 或 Redis OSS 叢集](#)
- [建立主函數，以示範變更傳輸中加密組態的程序](#)

## 定義將啟動 ElastiCache Valkey 或 Redis OSS 叢集的字串常數

首先，讓我們定義一些簡單的 Python 字串常數，這些常數將包含建立 ElastiCache 叢集所需的 AWS 實體名稱 Cache Subnet group，例如 security-group、和 default parameter group。所有這些 AWS 實體都必須在您願意使用的區域中 AWS 的帳戶中預先建立。

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
migration user guide"
EC_SUBNET_GROUP = "client-testing"
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

## 定義叢集組態類別

現在，我們來定義一些簡單的 Python 類別，其將代表叢集的組態，其中將包含有關叢集的中繼資料，例如 Valkey 或 Redis OSS 版本、執行個體類型，以及是否啟用或停用傳輸中加密（TLS）。

```
#Class definitions

class Config:
 def __init__(
 self,
 instance_type: str = "cache.t4g.small",
 version: str = "7.0",
 multi_az: bool = True,
 TLS: bool = True,
 name: str = None,
):
 self.instance_type = instance_type
 self.version = version
 self.multi_az = multi_az
 self.TLS = TLS
 self.name = name or f"tls-test"

 def create_base_launch_request(self):
 return {
 "ReplicationGroupId": self.name,
 "TransitEncryptionEnabled": self.TLS,
 "MultiAZEnabled": self.multi_az,
 "CacheNodeType": self.instance_type,
 "Engine": "redis",
 "EngineVersion": self.version,
```

```

 "CacheSubnetGroupName": EC_SUBNET_GROUP ,
 "CacheParameterGroupName":
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
 "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
 "SecurityGroupIds": [SECURITY_GROUP],
 }

class ConfigCME(Config):
 def __init__(
 self,
 instance_type: str = "cache.t4g.small",
 version: str = "7.0",
 multi_az: bool = True,
 TLS: bool = True,
 name: str = None,
 num_shards: int = 2,
 num_replicas_per_shard: int = 1,
):
 super().__init__(instance_type, version, multi_az, TLS, name)
 self.num_shards = num_shards
 self.num_replicas_per_shard = num_replicas_per_shard

 def create_launch_request(self) -> dict:
 launch_request = self.create_base_launch_request()
 launch_request["NumNodeGroups"] = self.num_shards
 launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
 return launch_request

```

定義一個將代表叢集本身的類別

現在，讓我們定義一些簡單的 Python 類別，這些類別將代表 ElastiCache Valkey 或 Redis OSS 叢集本身。此類別將有一個用戶端欄位，該欄位將保存 boto3 用戶端以進行 ElastiCache 管理操作，例如建立叢集和查詢 ElastiCache API。

```

import botocore.config
import boto3

Create boto3 client
def init_client(region: str = "us-east-1"):
 config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
 init_request = dict()
 init_request["config"] = config
 init_request["service_name"] = "elasticache"

```



```
init_request["region_name"] = region
return boto3.client(**init_request)

class ElastiCacheClusterBase:
 def __init__(self, name: str):
 self.name = name
 self.elasticache_client = init_client()

 def get_first_replication_group(self):
 return self.elasticache_client.describe_replication_groups(
 ReplicationGroupId=self.name
)["ReplicationGroups"][0]

 def get_status(self) -> str:
 return self.get_first_replication_group()["Status"]

 def get_transit_encryption_enabled(self) -> bool:
 return self.get_first_replication_group()["TransitEncryptionEnabled"]

 def is_available(self) -> bool:
 return self.get_status() == "available"

 def is_modifying(self) -> bool:
 return self.get_status() == "modifying"

 def wait_for_available(self):
 while True:
 if self.is_available():
 break
 else:
 time.sleep(5)

 def wait_for_modifying(self):
 while True:
 if self.is_modifying():
 break
 else:
 time.sleep(5)

 def delete_cluster(self) -> bool:
 self.elasticache_client.delete_replication_group(
 ReplicationGroupId=self.name, RetainPrimaryCluster=False
)
```

```
def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
 # generate api call to migrate the cluster to TLS preferred or to TLS required
 self.elasticache_client.modify_replication_group(
 ReplicationGroupId=self.name,
 TransitEncryptionMode=new_transit_encryption_mode,
 TransitEncryptionEnabled=True,
 ApplyImmediately=True,
)
 self.wait_for_modifying()

class ElastiCacheClusterCME(ElastiCacheClusterBase):
 def __init__(self, name: str):
 super().__init__(name)

 @classmethod
 def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
 config = config or ConfigCME()
 print(config)
 new_cluster = ElastiCacheClusterCME(config.name)
 launch_request = config.create_launch_request()
 new_cluster.elasticache_client.create_replication_group(**launch_request)
 new_cluster.wait_for_available()
 return new_cluster

 def get_configuration_endpoint(self) -> str:
 return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
#readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
 def __init__(self, config: ConfigCME = None):
 self.config = config or ConfigCME()

 def __enter__(self) -> ElastiCacheClusterCME:
 self.cluster = ElastiCacheClusterCME.launch(self.config)
 return self.cluster

 def __exit__(self, exc_type, exc_val, exc_tb):
 self.cluster.delete_cluster()
```

(選用) 建立包裝程式類別，以示範用戶端連線至 Valkey 或 Redis OSS 叢集

現在，讓我們為 `redis-py-cluster` 用戶端建立一個包裝類別。這個包裝類別將支援使用一些鍵預先填充叢集，然後執行隨機重複的 `get` 命令。

#### Note

這是一個選擇性步驟，但它簡化了在後面步驟中出現的主函數。

```
import redis
import random
from time import perf_counter_ns, time

class DowntimeTestClient:
 def __init__(self, client):
 self.client = client

 # num of keys prefilled
 self.prefilled = 0
 # percent of get above prefilled
 self.percent_get_above_prefilled = 10 # nil result expected when get hit above
prefilled
 # total downtime in nano seconds
 self.downtime_ns = 0
 # num of success and fail operations
 self.success_ops = 0
 self.fail_ops = 0
 self.connection_errors = 0
 self.timeout_errors = 0

 def replace_client(self, client):
 self.client = client

 def prefill_data(self, timelimit_sec=60):
 end_time = time() + timelimit_sec
 while time() < end_time:
 self.client.set(self.prefilled, self.prefilled)
 self.prefilled += 1

unsuccessful operations throw exceptions
```

```
def _exec(self, func):
 try:
 start_ns = perf_counter_ns()
 func()
 self.success_ops += 1
 elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
 # upon succesful execution of func
 # reset random_key to None so that the next command
 # will use a new random key
 self.random_key = None

 except Exception as e:
 elapsed_ns = perf_counter_ns() - start_ns
 self.downtime_ns += elapsed_ns
 # in case of failure- increment the relevant counters so that we will keep
track
 # of how many connection issues we had while trying to communicate with
 # the cluster.
 self.fail_ops += 1
 if e.__class__ is redis.exceptions.ConnectionError:
 self.connection_errors += 1
 if e.__class__ is redis.exceptions.TimeoutError:
 self.timeout_errors += 1

def _repeat_exec(self, func, seconds):
 end_time = time() + seconds
 while time() < end_time:
 self._exec(func)

def _new_random_key_if_needed(self, percent_above_prefilled):
 if self.random_key is None:
 max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
 return random.randint(0, max)
 return self.random_key

def _random_get(self):
 key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
 result = self.client.get(key)
 # we know the key was set for sure only in the case key < self.prefilled
 if key < self.prefilled:
 assert result.decode("UTF-8") == str(key)

def repeat_get(self, seconds=60):
```

```
self._repeat_exec(self._random_get, seconds)

def get_downtime_ms(self) -> int:
 return self.downtime_ns // 10 ** 6

def do_get_until(self, cond_check):
 while not cond_check():
 self.repeat_get()
 # do one more get cycle once condition is met
 self.repeat_get()
```

建立主函數，以示範變更傳輸中加密組態的程序

現在，讓我們定義主函數，它將執行以下操作：

1. 使用 boto3 ElastiCache client 建立叢集。
2. 初始化要以不含的明確TCP連線連線至叢集的redis-py-cluster用戶端TLS。
3. redis-py-cluster 用戶端會用一些資料預先填充叢集。
4. boto3 用戶端將觸發從無TLS遷移到TLS偏好的TLS遷移。
5. 當叢集遷移至 TLS 時Preferred，redis-py-clusterTCP用戶端會將重複get操作傳送至叢集，直到遷移完成為止。
6. 遷移至 TLS Preferred 完成後，我們將宣告叢集支援傳輸中加密。之後，我們將建立用戶端redis-py-cluster，以使用 連線至叢集TLS。
7. 我們將使用新TLS用戶端和舊TCP用戶端傳送一些get命令。
8. boto3 用戶端將觸發從 TLS 遷移TLSPreferred至TLS必要的。
9. 當叢集正在遷移至TLS必要項目時，redis-py-clusterTLS用戶端會將重複get操作傳送至叢集，直到遷移完成為止。

```
import redis

def init_cluster_client(
 cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
DowntimeTestClient:
 # we must use for the host name the cluster configuration endpoint.
 redis_client = redis.RedisCluster(
 host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
 socket_connect_timeout=0.1
```

```
)
test_client = DowntimeTestClient(redis_client)
if prefill_data:
 test_client.prefill_data()
return test_client

if __name__ == '__main__':
 config = ConfigCME(TLS=False, instance_type="cache.m5.large")

 with ElastiCacheCMEManager(config) as cluster:
 # create a client that will connect to the cluster with clear tcp connection
 test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

 # migrate the cluster to TLS Preferred
 cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

 # do repeated get commands until the cluster finishes the migration to TLS
 Preferred
 test_client_tcp.do_get_until(cluster.is_available)

 # verify that in transit encryption is enabled so that clients will be able to
 connect to the cluster with TLS
 assert cluster.get_transit_encryption_enabled() == True

 # create a client that will connect to the cluster with TLS connection.
 # we must first make sure that the cluster indeed supports TLS
 test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

 # by doing get commands with the tcp client for 60 more seconds
 # we can verify that the existing tcp connection to the cluster still works
 test_client_tcp.repeat_get(seconds=60)

 # do get commands with the new TLS client for 60 more seconds
 test_client_tcp.repeat_get(seconds=60)

 # migrate the cluster to TLS required
 cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

 # from this point the tcp clients will be disconnected and we must not use them
 anymore.
 # do get commands with the TLS client until the cluster finishes migration to
 TLS required mode.
 test_client_tls.do_get_until(cluster.is_available)
```

## 啟用傳輸中加密時的最佳實務

啟用傳輸中加密之前：請確定您有適當的DNS記錄處理

### Note

在此過程中，我們正在變更和刪除舊端點。不正確使用端點可能會導致 Valkey 或 Redis OSS 用戶端使用舊的和已刪除的端點，以防止其連接至叢集。

當叢集從無TLS 遷移至TLS偏好 時，會保留舊的每個節點DNS記錄，並以不同的格式產生新的每個節點DNS記錄。TLS啟用的叢集使用與 non-TLS-enabled叢集不同的DNS記錄格式。當叢集設定為加密模式時，ElastiCache 會保留兩個DNS記錄：偏好，以便應用程式和其他 Valkey 或 Redis OSS用戶端可以在它們之間切換。DNS 記錄中的下列變更會在TLS遷移程序期間發生：

說明啟用傳輸中加密時DNS記錄的變更

對於CME叢集

當叢集設定為「傳輸加密模式：偏好」時：

- 未TLS啟用叢集的原始叢集端點將保持作用中狀態。當叢集重新設定表單TLS加密模式「無」為「偏好」時，不會有停機時間。
- 當叢集設定為 TLS- 偏好模式時，將產生新的 TLS Valkey 或 Redis OSS端點。這些新端點將解析為IPs與舊端點相同的端點（非TLS）。
- 新的 TLS Valkey 或 Redis OSS組態端點將在ElastiCache 主控台中公開，並回應 describe-replication-group API。

當叢集設定為「傳輸加密模式：需要」時：

- 舊的TLS未啟用端點將被刪除。TLS 叢集端點不會停機。
- 您可以從cluster-configuration-endpoint ElastiCache 主控台或從 describe-replication-group 擷取新的 API。

對於啟用自動容錯移轉或停用自動容錯移轉的CMD叢集

當複寫群組設為「傳輸加密模式：偏好」時：

- 未TLS啟用叢集的原始主要端點和讀取器端點將保持作用中。

- 當叢集設定為 TLS Preferred 模式時，將產生新的TLS主要和讀取器端點。此新端點將解析為與舊端點（非）相同的 IPTLS。
- 新的主要端點和讀取器端點將在 ElastiCache 主控台中公開，並回應 describe-replication-group API。

當複寫群組設為「傳輸加密模式：需要」時：

- 將刪除舊的非TLS主要和讀取器端點。TLS 叢集端點不會停機。
- 您可以從主控台或 ElastiCache擷取新的主要和讀取器端點describe-replication-groupAPI。

## DNS 記錄的建議用量

### 對於CME叢集

- 使用叢集組態端點，而不是應用程式程式碼中的每個節點DNS記錄。不建議直接使用每個節點DNS名稱，因為它們可能會在新增或移除碎片時變更。
- 不要在應用程式中對叢集組態端點進行硬式編碼，因為它會在此過程中發生變更。
- 在應用程式中對叢集組態端點進行硬式編碼是一種不好的做法，因為它可能在此過程中變更。傳輸中加密完成後，請使用 查詢叢集組態端點 describe-replication-group API（如上所示（粗體）），並使用DNS您在此時間點收到回應的。

### 對於已啟用自動容錯移轉的CMD叢集

- 使用主要端點和讀取器端點，而不是應用程式程式碼中的每個節點DNS名稱，因為刪除舊的每個節點名稱，並在將叢集從不TLS偏好遷移到TLS偏好時產生新的節點DNS名稱。不建議直接使用每個節點DNS名稱，因為您可能會在未來將複本新增至叢集。此外，啟用自動容錯移轉時，ElastiCache 服務會自動變更主要叢集和複本的角色，建議使用主要端點和讀取器端點來協助您追蹤這些變更。最後，使用讀取器端點可協助您在叢集中的複本之間平均分配複本的讀取。
- 在應用程式中硬式編碼主要端點和讀取器端點是不良做法，因為它可以在TLS遷移過程中變更。遷移變更為 TLS- 偏好完成後，使用 describe-replication-group查詢主要端點和讀取器端點，API並使用DNS您從此時間點開始收到的 作為回應。如此，您將能夠以動態方式追蹤端點中的變更。

### 對於停用自動容錯移轉的CMD叢集

- 使用主要端點和讀取器端點，而不是應用程式程式碼中的每個節點DNS名稱。停用自動容錯移轉時，您改為完成擴展、修補、容錯移轉，以及啟用自動容錯移轉時由 ElastiCache 服務自動管理的其



他程序。這可讓您更輕易地手動追蹤不同端點。由於刪除舊的每個節點DNS名稱，並在將叢集從不TLS偏好遷移至不TLS偏好時產生新的節點名稱，因此請勿直接使用每個節點DNS名稱。這是強制性的，以使用戶端可以在 TLS遷移期間連線到叢集。此外，使用讀取器端點時，您將受益於在複本之間平均分散讀取，並在從叢集新增或刪除複本時追蹤 DNS記錄。

- 在應用程式中硬式編碼叢集組態端點是不良做法，因為它可以在TLS遷移過程中變更。

在傳輸過程中加密：注意遷移過程何時完成

傳輸加密模式的變更不是立即的，可能需要一些時間。大型叢集尤其如此。只有當叢集完成遷移至 TLS- 偏好時，它才能接受並同時提供 TCP 和 TLS連線。因此，在傳輸中加密完成之前，您不應建立會嘗試建立叢集TLS連線的用戶端。

在傳輸中加密成功完成或失敗時，有幾種方法可以得到通知：(未顯示於上面的程式碼範例中)：

- 加密完成後，使用 SNS服務取得通知
- 使用 describe-eventsAPI會在加密完成時發出事件
- 在 ElastiCache 主控台中看到已完成加密的訊息

您也可以再應用程式中實作邏輯，以瞭解加密是否完成。在上面的範例中，我們看到了幾種確保叢集完成遷移的方法：

- 等待遷移程序開始 (叢集狀態變更為「修改」)，並等待修改完成 (叢集狀態會變回「可用」)
- 透過查詢，確認叢集已transit\_encryption\_enabled設定為 Truedescribe-replication-groupAPI。

啟用傳輸中加密之後：確定您使用的用戶端設定正確

當叢集處於 TLS偏好模式時，您的應用程式應開啟叢集的TLS連線，並僅使用這些連線。如此一來，您的應用程式在啟用傳輸中加密時就不會遇到停機時間。您可以使用 SSL區段下的資訊命令，確保沒有更明確的 Valkey 或 Redis OSS引擎TCP連線。

```
SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0 (should be zero)
```

```
tls_mode_connected_tls_clients:100
```

使用 OpenSSL ( Memcached ) 連線至已啟用傳輸中加密的節點

若要從啟用傳輸中加密的 ElastiCache ( Memcached ) 節點存取資料，您需要使用使用 Secure Socket Layer ( ) 的用戶端SSL。您也可以使用 Amazo Linux 和 Amazo Linux 2 上使用 OpenSSL s\_client。

在 Amazo Linux 或 Amazo Linux 2 上，使用 OpenSSL s\_client 來連線到啟用傳輸中加密的 Memcached 叢集：

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

使用 Java 建立 TLS Memcached 用戶端

若要在 TLS 模式下建立用戶端，請執行下列動作，以使用適當的初始化用戶端SSLContext：

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
 public static void main(String[] args) throws Exception {
 ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
 // Build SSLContext
 TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
 tmf.init((KeyStore) null);
 SSLContext sslContext = SSLContext.getInstance("TLS");
 sslContext.init(null, tmf.getTrustManagers(), null);
 // Create the client in TLS mode
 connectionFactoryBuilder.setSSLContext(sslContext);
 MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

 // Store a data item for an hour.
 client.set("theKey", 3600, "This is the data value");
 }
}
```

## 使用 建立 TLS Memcached 用戶端 PHP

若要在 TLS 模式下建立用戶端，請執行下列動作，以使用適當的 初始化用戶端SSLContext：

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
 echo $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
```

```
/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
* Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

/* store the data for 60 seconds in the cluster.
* The client will decide which cache host will store this item.
*/
if($tls_client->set('key', 'value', 60)) {
 print "Successfully stored key\n";
} else {
 echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
 print "Successfully retrieved key\n";
} else {
 echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}
```

如需使用PHP用戶端的詳細資訊，請參閱 [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)。

## 中的靜態加密 ElastiCache

為了協助保護資料安全，Amazon ElastiCache 和 Amazon S3 提供不同的方法，以限制對快取中資料的存取。如需詳細資訊，請參閱 [Amazon VPCs和安全性 ElastiCache](#) 和 [Amazon 的身分和存取管理 ElastiCache](#)。

ElastiCache 靜態加密是一項透過加密磁碟上資料來提高資料安全性的功能。此功能在無伺服器快取上 一律為啟用狀態。啟用時，此功能會加密下列層面：

- 同步、備份和交換操作期間的磁碟
- 存放在 Amazon S3 中的備份

在啟用資料分層的叢集中存放於 SSDs ( 固態磁碟機 ) 的資料一律會加密。

ElastiCache 提供靜態的預設 ( 服務受管 ) 加密，以及能夠在 [AWS Key Management Service \( KMS \)](#) 中使用您自己的對稱客戶受管 AWS KMS金鑰。備份快取後，在加密選項中，選擇要使用預設加密金鑰或客戶受管金鑰。如需詳細資訊，請參閱[啟用靜態加密](#)。

### Note

預設 ( 服務受管 ) 加密是 GovCloud ( 美國 ) 區域中唯一可用的選項。

### Important

在現有的自我設計 Valkey 或 Redis OSS叢集上啟用靜態加密涉及在複寫群組上執行備份和還原後刪除現有的複寫群組。

只有在快取上建立複寫群組後，才能啟用靜態加密。因為要加密和解密資料需要經過一些處理，因此在這些操作期間啟用靜態加密可能會影響效能。您應該針對使用靜態加密和未使用靜態加密的資料進行基準分析，以判斷對使用案例的效能影響。

### 主題

- [靜態加密條件](#)
- [使用來自的客戶受管金鑰 AWS KMS](#)
- [啟用靜態加密](#)
- [另請參閱](#)

## 靜態加密條件

當您規劃 ElastiCache 靜態加密實作時，應謹記下列靜態 ElastiCache 加密限制：

- 執行 Valkey 7.2 和更新版本以及 Redis OSS 版本（排程為的 3.2.6EOL，請參閱 [Redis OSS 版本生命週期結束排程](#)）、4.0.10 或更新版本的複寫群組支援靜態加密。
- 只有 Amazon 中執行的複寫群組才支援靜態加密 VPC。
- 靜態加密僅支援執行下列節點類型的複寫群組。
  - R6gd、R6g、R5、R4、R3
  - M6、M5、M4、M3
  - T4g、T3、T2

如需詳細資訊，請參閱 [支援的節點類型](#)

- 透過將參數 `AtRestEncryptionEnabled` 明確設定為 `true`，可啟用靜態加密。
- 您只能在建立複寫群組時，於複寫群組上啟用靜態加密。您無法透過修改複寫群組來將靜態加密切換為開啟和關閉。如需如何在現有複寫群組上實作靜態加密的資訊，請參閱 [啟用靜態加密](#)。
- 如果叢集使用來自 r6gd 系列的節點類型，無論靜態 SSD 加密是否已啟用，儲存在上的資料都會加密。
- 在 AWS GovCloud（us-gov-east-1 和 us-gov-west-1）區域中，無法使用客戶受管金鑰進行靜態加密的選項。
- 如果叢集使用來自 r6gd 系列的節點類型，儲存在上的資料 SSD 會使用選取的客戶受管 AWS KMS 金鑰（或 AWS GovCloud 區域中的服務受管加密）加密。
- 使用 Memcached 時，僅無伺服器快取支援靜態加密。
- 使用 Memcached 時，在 GovCloud（us-gov-east-1 和 us-gov-west-1）區域中無法使用 AWS 靜態加密的客戶受管金鑰選項。

實作靜態加密可能會在備份和節點同步操作期間降低效能。針對您自己的資料進行靜態加密與無加密的基準分析比較，以判斷對實作的效能影響。

## 使用來自的客戶受管金鑰 AWS KMS

ElastiCache 支援靜態加密的對稱客戶受管 AWS KMS 金鑰（KMS 金鑰）。客戶受管 KMS 金鑰是您在 AWS 帳戶中建立、擁有和管理的加密金鑰。如需詳細資訊，請參閱 [AWS KMS Key](#) AWS Management Service 開發人員指南 中的金鑰。金鑰必須先在中 AWS KMS 建立，才能與搭配使用 ElastiCache。

若要了解如何建立 AWS KMS 根金鑰，請參閱[金鑰管理服務開發人員指南中的建立 AWS 金鑰](#)。

ElastiCache 可讓您與 整合 AWS KMS。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的[使用授權](#)。啟用 Amazon 與 ElastiCache 整合不需要客戶動作 AWS KMS。

kms:ViaService 條件金鑰會將金鑰 AWS KMS ( KMS 金鑰 ) 的使用限制為來自指定 AWS 服務的請求。若要 kms:ViaService 搭配使用 ElastiCache，請在條件索引鍵值中包含這兩個 ViaService 名稱：elasticache.AWS\_region.amazonaws.com 和 dax.AWS\_region.amazonaws.com。如需詳細資訊，請參閱 [kms : ViaService](#)。

您可以使用 [AWS CloudTrail](#) 來追蹤 Amazon 代表您 ElastiCache 傳送至 AWS Key Management Service 的請求。所有與客戶受管金鑰 AWS Key Management Service 相關的 API 呼叫都有對應的 CloudTrail 日誌。您也可以透過呼叫來查看 ElastiCache 建立的授予 [ListGrants](#) KMS API。

使用客戶受管金鑰對複寫群組進行加密後，複寫群組的所有備份都會依以下方式進行加密：

- 自動每日備份會使用與叢集關聯的客戶受管金鑰來進行加密。
- 複寫群組被刪除時所建立的最終備份，也使用與複寫群組關聯的客戶受管金鑰進行加密。
- 根據預設，手動建立的備份會加密，以使用與複寫群組相關聯的 KMS 金鑰。您可以透過選擇其他客戶受管金鑰來覆寫此選項。
- 複製備份預設為使用與來源備份關聯的客戶受管金鑰。您可以透過選擇其他客戶受管金鑰來覆寫此選項。

#### Note

- 將備份匯出到所選 Amazon S3 儲存貯體時，無法使用客戶受管金鑰。不過，匯出至 Amazon S3 的所有備份都使用 [伺服器端加密](#) 來加密。您可以選擇將備份檔案複製到新的 S3 物件並使用客戶受管 KMS 金鑰加密、將檔案複製到使用 KMS 金鑰設定預設加密的另一個 S3 儲存貯體，或變更檔案本身中的加密選項。
- 您也可以使用客戶受管金鑰，為不是使用客戶受管金鑰進行加密的複寫群組，加密手動建立的備份。使用此選項，即使原始複寫群組上的資料未加密，Amazon S3 中儲存的備份檔案仍會使用 KMS 金鑰加密。

從備份還原可讓您從可用的加密選項中進行選擇，類似於建立新複寫群組時可用的加密選項。

- 如果刪除金鑰或停用金鑰，並針對您用來加密快取的金鑰撤銷授權，則快取將變成無法恢復。換句話說，它無法在硬體故障後修改或復原。AWS KMS 只會在至少七天的等待期後刪除根金鑰。金鑰刪除後，您可以使用其他客戶受管金鑰建立備份以用於封存。
- 自動金鑰輪換會保留根金鑰的 AWS KMS 屬性，因此輪換不會影響您存取 ElastiCache 資料的能力。加密的 Amazon ElastiCache 快取不支援手動金鑰輪換，這涉及建立新的根金鑰，以及更新對舊金鑰的任何參考。若要進一步了解，請參閱 AWS Key Management Service 開發人員指南中的[輪換 AWS KMS 金鑰](#)。
- 使用 KMS 金鑰加密 ElastiCache 快取需要每個快取授予一次。此授權將在快取的整個生命週期中使用。此外，在備份建立期間，每個備份會使用一個授權。建立備份後，此授權就會淘汰。
- 如需授予和限制的詳細資訊 AWS KMS，請參閱 AWS Key Management Service 開發人員指南中的[限制](#)。

## 啟用靜態加密

所有無伺服器快取都會啟用靜態加密。

建立自行設計的叢集時，您可以藉由將參數 `AtRestEncryptionEnabled` 設定為 `true` 來啟用靜態加密。您無法在現有複寫群組上啟用靜態加密。

您可以在建立 ElastiCache 快取時啟用靜態加密。您可以使用 AWS Management Console、AWS CLI 或來執行此操作 ElastiCache API。

建立快取時，您可以選擇下列其中一個選項：

- 預設 – 此選項使用服務管理的靜態加密。
- 客戶受管金鑰 – 此選項可讓您提供金鑰 ID/ARN 來自 AWS KMS，以進行靜態加密。

若要了解如何建立 AWS KMS 根金鑰，請參閱 AWS Key Management Service 開發人員指南中的[建立金鑰](#)

## 內容

- [使用 啟用靜態加密 AWS Management Console](#)
- [使用 啟用靜態加密 AWS CLI](#)



## 在現有的自我設計 Valkey 或 Redis OSS 叢集上啟用靜態加密

您只能在建立 Valkey 或 Redis OSS 複寫群組時啟用靜態加密。如果您想要啟用現有複寫群組的靜態加密，請執行下列動作。

### 啟用現有複寫群組的靜態加密

1. 建立您現有複寫群組的手動備份。如需詳細資訊，請參閱「[取得手動備份](#)」。
2. 透過從備份還原來建立新的複寫群組。在新的複寫群組上，啟用靜態加密。如需詳細資訊，請參閱[從備份還原到新的快取](#)。
3. 更新應用程式中的端點以指向新的複寫群組。
4. 刪除舊的複寫群組。如需詳細資訊，請參閱 [在中刪除叢集 ElastiCache](#) 或 [刪除複寫群組](#)。

### 使用 啟用靜態加密 AWS Management Console

#### 在無伺服器快取上啟用靜態加密 (主控台)

所有無伺服器快取都會啟用靜態加密。根據預設，AWS 擁有的 KMS 金鑰會用來加密資料。若要選擇您自己的 AWS KMS 金鑰，請進行下列選擇：

- 展開預設設定區段。
- 在預設設定區段下選擇自訂預設設定。
- 在安全性區段下選擇自訂安全性設定。
- 在加密金鑰設定下選擇客戶受管 CMK。
- 在 AWS KMS 金鑰設定下選擇一個金鑰。

#### 在自行設計的叢集上啟用靜態加密 (主控台)

自行設計快取時，採用「輕鬆建立」方法的「開發/測試」和「生產」組態會使用預設金鑰啟用靜態加密。當您自行選擇組態時，請進行下列選擇：

- 選擇 3.2.6、4.0.10 或更新版本做為引擎版本。
- 按一下靜態加密選項的啟用旁的核取方塊。
- 選擇預設金鑰或客戶受管 CMK。

如需此程序 step-by-step，請參閱下列內容：

- [建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)
- [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)

使用 啟用靜態加密 AWS CLI

若要在使用 建立 Valkey 或 Redis OSS叢集時啟用靜態加密 AWS CLI，請在建立複寫群組時使用 `--at-rest-encryption-enabled` 參數。

在 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集上啟用靜態加密 ( CLI )

下列操作會建立 `my-classic-rg` 具有三個節點 OSS ( `--num-cache-clusters` )、一個主要複本和兩個僅供讀取複本的 Valkey 或 Redis ( 停用叢集模式 ) 複寫群組。此複寫群組 ( `--at-rest-encryption-enabled` ) 已啟用靜態加密。

若要啟用此複寫群組的加密，您必須使用下列參數和值：

重要參數

- `--engine`—必須為 `valkey` 或 `redis`。
- `--engine-version`—如果引擎是 Redis OSS，則必須為 3.2.6、4.0.10 或更新版本。
- `--at-rest-encryption-enabled` - 啟用靜態加密的必要項目。

Example 1：具有複本的 Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id my-classic-rg \
 --replication-group-description "3 node replication group" \
 --cache-node-type cache.m4.large \
 --engine redis \
 --at-rest-encryption-enabled \
 --num-cache-clusters 3
```

針對 Windows：

```
aws elasticache create-replication-group ^
 --replication-group-id my-classic-rg ^
```

```
--replication-group-description "3 node replication group" ^
--cache-node-type cache.m4.large ^
--engine redis ^
--at-rest-encryption-enabled ^
--num-cache-clusters 3 ^
```

如需其他資訊，請參閱以下內容：

- [從頭開始建立 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [create-replication-group](#)

在叢集上為 Valkey 或 Redis 啟用靜態加密 OSS ( 啟用叢集模式 ) ( CLI )

下列操作會建立my-clustered-rg具有三個節點群組或碎片 OSS ( --num-node-groups ) 的 Valkey 或 Redis ( 啟用叢集模式 ) 複寫群組。每個 都有三個節點，一個主要和兩個僅供讀取複本 ( --replicas-per-node-group )。此複寫群組 ( --at-rest-encryption-enabled ) 已啟用靜態加密。

若要啟用此複寫群組的加密，您必須使用下列參數和值：

#### 重要參數

- **--engine**—必須為 valkey或 redis。
- **--engine-version**- 如果引擎是 Redis OSS，則必須為 4.0.10 或更新版本。
- **--at-rest-encryption-enabled** - 啟用靜態加密的必要項目。
- **--cache-parameter-group** - 必須為 default-redis4.0.cluster.on 或其衍生項目，才能成為啟用複寫群組的叢集模式。

Example 2 : Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id my-clustered-rg \
 --replication-group-description "redis clustered cluster" \
 --cache-node-type cache.m3.large \
 --num-node-groups 3 \
 --replicas-per-node-group 2 \
 --at-rest-encryption-enabled
```

```
--engine redis \
--engine-version 6.2 \
--at-rest-encryption-enabled \
--cache-parameter-group default.redis6.x.cluster.on
```

針對 Windows :

```
aws elasticache create-replication-group ^
 --replication-group-id my-clustered-rg ^
 --replication-group-description "redis clustered cluster" ^
 --cache-node-type cache.m3.large ^
 --num-node-groups 3 ^
 --replicas-per-node-group 2 ^
 --engine redis ^
 --engine-version 6.2 ^
 --at-rest-encryption-enabled ^
 --cache-parameter-group default.redis6.x.cluster.on
```

如需其他資訊，請參閱以下內容：

- [從頭開始建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 複寫群組 \( AWS CLI \)](#)
- [create-replication-group](#)

另請參閱

- [Amazon VPCs和安全性 ElastiCache](#)
- [Amazon 的身分和存取管理 ElastiCache](#)

## 身分驗證和授權

AWS Identity and Access Management ( IAM ) 是一種 Web 服務，可協助您安全地控制對 AWS 資源的存取。ElastiCache 支援使用 IAM 和 Valkey 和 Redis OSSAUTH命令驗證使用者，以及使用角色型存取控制 ( ) 授權使用者操作RBAC。

主題

- [角色型存取控制 \( RBAC \)](#)
- [使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)
- [在 Valkey 或 Redis ElastiCacheOSS快取上停用存取控制](#)

## 角色型存取控制 ( RBAC )

您可以透過 Valkey 7.2 和更新[使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)版本以及 Redis 6.0 OSS 之後的 中所描述的 Valkey 和 Redis OSSAUTH命令來驗證使用者，而不需要使用 ( RBAC ) 來驗證使用者。RBAC 也是控制無伺服器快取存取的唯一方法。

與 Valkey 和 Redis OSS 不同AUTH，在驗證所有已驗證的用戶端的權杖時，其具有完整快取存取權，RBAC可讓您透過使用者群組控制快取存取權。這些使用者群組的設計目的是整理快取的存取權。

透過 RBAC，您可以使用存取字串建立使用者並為其指派特定許可，如下所述。您可以將使用者指派給與特定角色 ( 管理員、人力資源 ) 一致的使用者群組，然後部署到一或多個 ElastiCache 快取。透過這樣做，您可以使用相同的 Valkey 或 Redis OSS快取，在用戶端之間建立安全界限，並防止用戶端存取彼此的資料。

### Note

RBAC 搭配 Valkey 叢集使用時，您仍然需要指派使用者和使用者群組引擎「redis」。

RBAC 旨在支援在 Redis 6 OSS [ACL](#)中引入。當您RBAC搭配 ElastiCache Valkey 或 Redis OSS快取使用時，有一些限制：

- 您無法在存取字串中指定密碼。您可以使用 [CreateUser](#)或 [ModifyUser](#)呼叫設定密碼。
- 針對使用者權限，您需傳遞 on 和 off 作為存取字串的一部分。如果存取字串中沒有指定這兩項，則會為使用者指派 off，且使用者沒有快取的存取權限。
- 您無法使用禁止和重新命名的命令。如果您指定禁止或重新命名的命令，則會擲回例外情況。如果您想要將存取控制清單 ( ACLs ) 用於重新命名的命令，請指定命令的原始名稱，換句話說，在重新命名命令之前，請指定命令的名稱。
- 您無法使用 reset 命令作為存取字串的一部分。您可以使用API參數指定密碼，ElastiCache ( Redis OSS ) 會管理密碼。因此您無法使用 reset，因為它會刪除使用者的所有密碼。
- Redis OSS 6 引入 [ACLLIST](#)命令。此命令會傳回使用者清單，以及套用至每個使用者的ACL規則。ElastiCache ( Redis OSS ) 支援 ACL LIST命令，但不像 Redis 一樣支援密碼雜湊OSS。透過 ElastiCache ( Redis OSS )，您可以使用 [describe-users](#) 操作來取得類似資訊，包括存取字串中包含的規則。但是 [describe-users](#) 不會擷取使用者密碼。

ElastiCache 搭配 Valkey 和 Redis 支援的其他唯讀命令OSS包括 [ACL WHOAMI](#)、[ACL USERS](#)和 [ACL CAT](#)。ElastiCache 搭配 Valkey 和 Redis OSS不支援任何其他以寫入為基礎的ACL命令。

- 以下為目前的限制：

資源	允許的上限
每使用者群組使用者數	100
使用者數	1000
使用者群組數	100

RBAC 搭配 ElastiCache ( Redis OSS ) 使用 會更詳細地描述如下。

### 主題

- [使用存取字串指定許可](#)
- [ElastiCache 使用 Valkey 或 Redis RBAC套用至 的快取 OSS](#)
- [從 遷移AUTH至 RBAC](#)
- [從 遷移RBAC至 AUTH](#)
- [自動輪換使用者密碼](#)
- [使用 驗證 IAM](#)

### 使用存取字串指定許可

若要指定 ElastiCache ( Redis OSS ) 快取的許可，您可以使用 AWS CLI 或 建立存取字串，並將其指派給使用者 AWS Management Console。

存取字串的定義是套用至使用者的空格分隔規則清單。用於定義使用者可以執行哪些命令，以及使用者可以操作哪些索引鍵。為了執行命令，使用者必須能存取執行中的命令以及該命令存取的所有索引鍵。規則會從左到右累積套用，如果提供的字串中存在冗餘項目，可以改用更簡單的字串而非提供的字串。

如需ACL規則語法的相關資訊，請參閱 [ACL](#)。

在下列範例中，存取字串代表有權存取所有可用索引鍵和命令的活躍使用者。

```
on ~* +@all
```

存取字串語法可細分以下各項：

- on - 使用者是活躍使用者。

- ~\* - 存取權限提供給所有可用的索引鍵。
- +@all - 存取權限提供給所有可用的命令。

先前的設定受到最低限度的限制。您可以修改這些設定，提高安全性。

在下面的範例中，存取字串代表對於以「app::」keyspace 開頭的索引鍵，存取權受限於讀取存取的使用者

```
on ~app::* -@all +@read
```

您可以列出使用者可存取的命令，進一步精簡這些許可：

+*command1* - 使用者的命令存取權僅限於 **## 1**。

+@category - 使用者的存取權限受限於某個命令類別。

如需將存取字串指派給使用者的相關資訊，請參閱「[使用主控台和 建立使用者和使用者群組 CLI](#)」。

如果您要將現有工作負載遷移至 ElastiCache，您可以呼叫 `aws-logs` 來擷取存取字串ACL LIST，但不包括使用者和任何密碼雜湊。

對於 Redis 6.2 版及更高OSS版本，也支援下列存取字串語法：

- &\* - 存取權限提供給所有可用的索引鍵。

對於 Redis 7.0 版及更高OSS版本，也支援下列存取字串語法：

- | - 可用於封鎖子命令 (例如「-config|set」)。
- %R~<pattern> - 新增指定的讀入金鑰模式。這與常態金鑰模式類似，但僅授予符合特定模式的金鑰讀取許可。如需詳細資訊，請參閱[金鑰許可](#)。
- %W~<pattern> - 新增指定的寫入金鑰模式。這與常態金鑰模式類似，但僅授予符合特定模式的金鑰寫入許可。如需詳細資訊，請參閱[ACL金鑰許可](#)。
- %RW~<pattern> - ~<pattern> 的別名。
- (<rule list>) - 建立新的選擇器以配對規則。選擇器會在使用者許可之後受到評估，並根據其定義的順序進行評估。如果命令符合使用者許可或任何選擇器，則允許該命令。請參閱[ACL選取器](#)的詳細資訊。
- clearselectors - 刪除所有連接到使用者的選擇器。

## ElastiCache 使用 Valkey 或 Redis RBAC套用至 的快取 OSS

若要 ElastiCache 搭配 Valkey 或 Redis OSS 使用 RBAC，請執行下列步驟：

1. 建立一或多位使用者。
2. 建立使用者群組，並為該群組新增使用者。
3. 將使用者群組指派給啟用傳輸中加密的快取。

以下詳細說明這些步驟。

### 主題

- [使用主控台和 建立使用者和使用者群組 CLI](#)
- [使用主控台和 管理使用者群組 CLI](#)
- [將使用者群組指派至無伺服器快取](#)
- [將使用者群組指派給複寫群組](#)

### 使用主控台和 建立使用者和使用者群組 CLI

RBAC 使用者的使用者資訊是使用者 ID、使用者名稱，以及選用的密碼和存取字串。存取字串提供索引鍵和命令的許可層級。使用者 ID 對使用者而言是唯一的，而使用者名稱是傳遞給引擎的內容。

請確定您提供的使用者許可對使用者群組的預期用途來說有意義。例如，如果您建立名為 Administrators 的使用者群組，則您新增至該群組的任何使用者都應該將其存取字串設為完整存取索引鍵和命令。針對 e-commerce 使用者群組中的，您可以將其存取字串設為唯讀存取。

ElastiCache 會自動設定具有使用者 ID 和使用者名稱的預設使用者，"default"並將其新增至所有使用者群組。您無法刪除或修改此使用者。此使用者旨在與先前 Redis OSS 版本的預設行為相容，並具有允許其呼叫所有命令和存取所有金鑰的存取字串。

若要將適當的存取控制新增至快取，請將此預設使用者取代為未啟用或未使用高強度密碼的新使用者。若要變更預設使用者，請將使用者名稱設定為 default。然後您可以將其與原始預設使用者交換。

下列程序說明如何交換原始 default 使用者與具有經修改存取字串的另一個 default 使用者。

### 在主控台上修改預設使用者

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。



2. 從導覽窗格選擇使用者群組管理。
3. 針對使用者群組 ID，選擇您要修改的 ID。確定您選擇的是連結而非核取方塊。
4. 選擇 Modify (修改)。
5. 在修改視窗中，選擇管理，並針對使用者名稱選擇要作為預設使用者的使用者。
6. 選擇 Choose (選擇)。
7. 選擇 Modify (修改)。執行這項操作時，原始預設使用者所擁有的任何現有快取連線都會終止。

## 使用 修改預設使用者 AWS CLI

1. 使用以下命令以使用者名稱 default 建立新使用者。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-user \
 --user-id "new-default-user" \
 --user-name "default" \
 --engine "REDIS" \
 --passwords "a-strong-password" \
 --access-string "off +get ~keys*"
```

針對 Windows：

```
aws elasticache create-user ^
 --user-id "new-default-user" ^
 --user-name "default" ^
 --engine "REDIS" ^
 --passwords "a-strong-password" ^
 --access-string "off +get ~keys*"
```

2. 建立使用者群組並新增先前建立的使用者。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-user-group \
 --user-group-id "new-group-2" \
 --engine "REDIS" \
 --user-ids "new-default-user"
```

針對 Windows：

```
aws elasticache create-user-group ^
 --user-group-id "new-group-2" ^
 --engine "REDIS" ^
 --user-ids "new-default-user"
```

### 3. 交換新的 default 使用者與原始 default 使用者。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-user-group \
 --user-group-id test-group \
 --user-ids-to-add "new-default-user" \
 --user-ids-to-remove "default"
```

針對 Windows：

```
aws elasticache modify-user-group ^
 --user-group-id test-group ^
 --user-ids-to-add "new-default-user" ^
 --user-ids-to-remove "default"
```

呼叫此修改操作時，原始預設使用者所擁有的任何現有快取連線都會終止。

建立使用者時，最多可以設定兩個密碼。修改密碼時，任何現有的快取連線都會保留。

特別是，在使用 RBAC for ElastiCache ( Redis OSS ) 時，請注意這些使用者密碼限制：

- 密碼必須為 16 - 128 個可列印字元。
- 不允許使用以下非英數字元：, " " / @。

#### 使用主控台和 管理使用者 CLI

透過下列程序，在主控台上管理使用者。

#### 在主控台上管理使用者

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在 Amazon ElastiCache 儀表板上，選擇使用者管理。以下是可用的選項：

- 建立使用者 - 建立使用者時，您需輸入使用者 ID、使用者名稱、身分驗證模式和存取字串。存取字串會為允許使用者存取的索引鍵和命令設定許可層級。

建立使用者時，最多可以設定兩個密碼。修改密碼時，任何現有的快取連線都會保留。

- 修改使用者 - 可讓您更新使用者的身分驗證設定或變更其存取字串。
- 刪除使用者 - 將帳戶從其所屬的任何使用者群組中移除。

透過下列程序，使用 AWS CLI 來管理使用者。

#### 使用 修改使用者 CLI

- 使用 `modify-user` 命令來更新使用者的密碼或密碼，或變更使用者的存取許可。

修改使用者時，會更新與使用者相關聯的使用者群組，以及與使用者群組相關聯的任何快取。所有現有的連線都會保留。範例如下。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-user \
 --user-id user-id-1 \
 --access-string "~objects:* ~items:* ~public:*" \
 --no-password-required
```

針對 Windows：

```
aws elasticache modify-user ^
 --user-id user-id-1 ^
 --access-string "~objects:* ~items:* ~public:*" ^
 --no-password-required
```

#### Note

不建議您使用 `nopass` 選項。如果您使用此選項，建議將使用者的許可設定為唯讀，並提供一組有限索引鍵的存取權。

## 使用 刪除使用者 CLI

- 使用 `delete-user` 命令來刪除使用者。系統會刪除帳戶，並將其從所屬的任何使用者群組中移除。以下是範例。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-user \
 --user-id user-id-2
```

針對 Windows：

```
aws elasticache delete-user ^
 --user-id user-id-2
```

若要查看使用者清單，請呼叫 [describe-users](#) 作業。

```
aws elasticache describe-users
```

## 使用主控台和 管理使用者群組 CLI

您可以建立使用者群組來整理和控制使用者對一或多個快取的存取權，如下所示。

透過下列程序，使用主控台來管理使用者群組。

### 使用主控台來管理使用者群組

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。
2. 在 Amazon ElastiCache 儀表板上，選擇使用者群組管理。

可透過以下作業來建立新的使用者群組：

- 建立 - 建立使用者群組時，您需新增使用者，然後將使用者群組指派給快取。例如，您可以為具有快取管理角色的使用者建立 Admin 使用者群組。


#### Important

建立使用者群組時，您需加入預設使用者。

- 新增使用者 - 將使用者新增至使用者群組。
- 移除使用者 - 從使用者群組中刪除使用者。將使用者從使用者群組中移除時，他們所擁有的任何現有快取連線都會終止。
- 刪除 - 用來來刪除使用者群組。請注意，系統會刪除使用者群組本身，而不是屬於該群組的使用者。

針對現有的使用者群組，您可以執行以下操作：

- 新增使用者 - 將現有使用者新增到使用者群組中。
- 刪除使用者 - 從使用者群組中移除現有使用者。

 Note

使用者會從使用者群組中移除，但不會從系統中刪除。

使用下列程序來管理使用的使用者群組CLI。

使用 建立新的使用者群組並新增使用者 CLI

- 請使用 `create-user-group` 命令，如下列所示。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-user-group \
 --user-group-id "new-group-1" \
 --engine "REDIS" \
 --user-ids user-id-1, user-id-2
```

針對 Windows：

```
aws elasticache create-user-group ^
 --user-group-id "new-group-1" ^
 --engine "REDIS" ^
 --user-ids user-id-1, user-id-2
```

## 使用 新增使用者或移除目前成員，以修改使用者群組 CLI

- 請使用 `modify-user-group` 命令，如下列所示。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-user-group --user-group-id new-group-1 \
--user-ids-to-add user-id-3 \
--user-ids-to-remove user-id-2
```

針對 Windows：

```
aws elasticache modify-user-group --user-group-id new-group-1 ^
--user-ids-to-add userid-3 ^
--user-ids-to-remove user-id-2
```

### Note

屬於從使用者群組中移除之使用者的任何開啟連線，都會以此命令結束。

## 使用 刪除使用者群組 CLI

- 請使用 `delete-user-group` 命令，如下列所示。系統會刪除使用者群組本身，而不是屬於該群組的使用者。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-user-group /
--user-group-id
```

針對 Windows：

```
aws elasticache delete-user-group ^
--user-group-id
```

若要查看使用者群組清單，您可以呼叫 [describe-user-groups](#) 操作。

```
aws elasticache describe-user-groups \
 --user-group-id test-group
```

將使用者群組指派至無伺服器快取

在您建立使用者群組並新增使用者之後，實作的最後一步RBAC是將使用者群組指派給無伺服器快取。

使用主控台將使用者群組指派至無伺服器快取

若要使用 將使用者群組新增至無伺服器快取 AWS Management Console，請執行下列動作：

- 如需停用叢集模式，請參閱「[建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)」
- 如需啟用叢集模式，請參閱「[建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)」

使用 將使用者群組指派給無伺服器快取 AWS CLI

下列 AWS CLI 操作會使用具有 值的 `user-group-id` 參數來建立無伺服器快取`my-user-group-id`。將子網路群組 `sng-test` 取代為已存在的子網路群組。

重要參數

- **--engine** – 必須為 `valkey` 或 `redis`。
- **--user-group-id** - 此值提供由具有指定的快取存取許可之使用者所組成的使用者群組 ID。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-serverless-cache \
 --serverless-cache-name "new-serverless-cache" \
 --description "new-serverless-cache" \
 --engine "redis" \
 --user-group-id "new-group-1"
```

針對 Windows：

```
aws elasticache create-serverless-cache ^
 --serverless-cache-name "new-serverless-cache" ^
 --description "new-serverless-cache" ^
 --engine "redis" ^
 --user-group-id "new-group-1"
```

下列 AWS CLI 操作會使用 值的 `user-group-id` 參數修改無伺服器快取 `my-user-group-id`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-serverless-cache \
 --serverless-cache-name serverless-cache-1 \
 --user-group-id "new-group-2"
```

針對 Windows：

```
aws elasticache modify-serverless-cache ^
 --serverless-cache-name serverless-cache-1 ^
 --user-group-id "new-group-2"
```

請注意，對快取所做的任何修改都會以非同步方式更新。您可以透過檢視事件來監控進度。如需詳細資訊，請參閱[檢視 ElastiCache 事件](#)。

將使用者群組指派給複寫群組

在您建立使用者群組並新增使用者之後，實作的最後一步 RBAC 是將使用者群組指派給複寫群組。

使用主控台將使用者群組指派給複寫群組

若要使用 將使用者群組新增至複寫 AWS Management Console，請執行下列動作：

- 如需停用叢集模式，請參閱「[建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)」
- 如需啟用叢集模式，請參閱「[建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \( 主控台 \)](#)」

使用 將使用者群組指派給複寫群組 AWS CLI

下列 AWS CLI 操作會建立啟用傳輸中加密 ( TLS ) 的複寫群組，以及值為 的 `user-group-ids` 參數 `my-user-group-id`。將子網路群組 `sng-test` 取代為已存在的子網路群組。

重要參數

- **--engine** – 必須為 `valkey` 或 `redis`。
- **--engine-version** – 必須是 6.0 或更新版本。
- **--transit-encryption-enabled** - 進行身分驗證和為使用者群組建立關聯所需的項目。
- **--user-group-ids** - 此值提供由具有指定的快取存取許可之使用者所組成的使用者群組 ID。



- **--cache-subnet-group** - 為使用者群組建立關聯所需的項目。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id "new-replication-group" \
 --replication-group-description "new-replication-group" \
 --engine "redis" \
 --cache-node-type cache.m5.large \
 --transit-encryption-enabled \
 --user-group-ids "new-group-1" \
 --cache-subnet-group "cache-subnet-group"
```

針對 Windows：

```
aws elasticache create-replication-group ^
 --replication-group-id "new-replication-group" ^
 --replication-group-description "new-replication-group" ^
 --engine "redis" ^
 --cache-node-type cache.m5.large ^
 --transit-encryption-enabled ^
 --user-group-ids "new-group-1" ^
 --cache-subnet-group "cache-subnet-group"
```

下列 AWS CLI 操作會在傳輸中啟用加密 ( TLS ) 並修改 `user-group-ids` 參數值為 的複寫群組 `my-user-group-id`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id replication-group-1 \
 --user-group-ids-to-remove "new-group-1" \
 --user-group-ids-to-add "new-group-2"
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id replication-group-1 ^
 --user-group-ids-to-remove "new-group-1" ^
 --user-group-ids-to-add "new-group-2"
```

記下回應中的 PendingChanges。對快取所做的任何修改都會以非同步方式更新。您可以透過檢視事件來監控進度。如需詳細資訊，請參閱[檢視 ElastiCache 事件](#)。

## 從 遷移AUTH至 RBAC

如果您AUTH依 所述使用，[使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)並想要使用 遷移至 RBAC，請使用下列程序。

使用下列程序，RBAC使用主控台從 遷移AUTH至。

RBAC 使用主控台從 Valkey 或 Redis 遷移OSSAUTH至

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/>開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇要修改快取所在的 AWS 區域。
3. 在導覽窗格中，選擇您要修改之快取上執行的引擎。

接著會出現所選引擎的快取清單。

4. 在快取清單中，選擇您要修改之快取的名稱。
5. 在 Actions (動作) 中，選擇 Modify (修改)。

修改視窗隨即出現。

6. 針對存取控制，選擇使用者群組存取控制清單。
7. 針對使用者群組存取控制清單，選擇一個使用者群組。
8. 選擇預覽變更，然後在下一個畫面上選擇修改。

使用下列程序，從 Valkey 或 Redis 遷移OSSAUTH至RBAC使用 CLI。

RBAC 使用 從 遷移AUTH至 CLI

- 請使用 modify-replication-group 命令，如下列所示。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group --replication-group-id test \
--auth-token-update-strategy DELETE \
--user-group-ids-to-add user-group-1
```

針對 Windows :

```
aws elasticache modify-replication-group --replication-group-id test ^
--auth-token-update-strategy DELETE ^
--user-group-ids-to-add user-group-1
```

## 從 遷移RBAC至 AUTH

如果您使用 RBAC並想要遷移至 Redis OSS AUTH ，請參閱 [從 遷移RBAC至 AUTH](#)。

### Note

如果您需要停用 ElastiCache 快取的存取控制，則需要透過 執行 AWS CLI。如需詳細資訊，請參閱[the section called “在 Valkey 或 Redis ElastiCacheOSS快取上停用存取控制”](#)。

## 自動輪換使用者密碼

使用 AWS Secrets Manager時，您可以將程式碼中的硬式編碼憑證（包括密碼）取代為API呼叫 Secrets Manager，以程式設計方式擷取秘密。這有助於確保不讓某人研究您的程式碼而盜用秘密，因為秘密根本不在程式碼中。此外，您可以設定 Secrets Manager，根據您指定的排程自動輪換秘密。這可讓您以短期秘密取代長期秘密，有助於大幅降低洩漏風險。

使用 Secrets Manager，您可以使用 Secrets Manager 提供的 AWS Lambda 函數自動輪換 ElastiCache（RedisOSS）密碼（即秘密）。

如需的詳細資訊 AWS Secrets Manager，請參閱[什麼是 AWS Secrets Manager？](#)

## ElastiCache 如何使用秘密

Valkey 7.2 具有與 Redis 7.0 OSS 相同的功能集。在 Redis OSS 6 中，ElastiCache 介紹[角色型存取控制（RBAC）](#)如何保護 Valkey 或 Redis OSS叢集。此功能允許以可執行命令和可存取金鑰限制部分連線。使用時RBAC，當客戶使用密碼建立使用者時，需要以純文字手動輸入密碼值，且運算子可以看到。

使用 Secret Manager 時，應用程式會從 Secrets Manager 擷取密碼，而非手動輸入，接著再將密碼儲存在應用程式組態。如需如何執行此作業的資訊，請參閱 [ElastiCache 使用者與秘密的關聯](#)。

使用秘密會產生費用。如需定價資訊，請參閱 [AWS Secrets Manager 定價](#)。

## ElastiCache 使用者與秘密的關聯

Secrets Manager 會在密碼的 SecretString 欄位保留關聯使用者的參考。不會參考 ElastiCache 側邊的秘密。

```
{
 "password": "strongpassword",
 "username": "user1",
 "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
 bond between the secret and the user
}
```

## Lambda 輪換函數

若要啟用 Secrets Manager 自動密碼輪換，您將建立 Lambda 函數，該函數將與 [修改使用者](#) 互動，API 以更新使用者的密碼。

如需其運作方式的詳細資訊，請參閱 [輪換的運作方式](#)。

### Note

對於某些 AWS 服務，為避免混淆代理案例，AWS 建議您同時使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件金鑰。不過，如果您在輪換函數政策中包含 `aws:SourceArn` 條件，則輪換函數只能用來輪換該指定的秘密 ARN。建議您僅包含內容金鑰 `aws:SourceAccount`，以便可以將輪換函數用於多個秘密。

對於您可能遇到的任何問題，請參閱 [對 AWS Secrets Manager 輪換進行故障診斷](#)。

## 如何建立 ElastiCache 使用者並將其與 Secrets Manager 建立關聯

下列步驟說明如何建立使用者，並將其與 Secrets Manager 建立關聯：

### 1. 建立非作用中的使用者

若為 Linux、macOS 或 Unix：

```
aws elasticache create-user \
 --user-id user1 \
 --user-arn arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1
```

```
--user-name user1 \
--engine "REDIS" \
--no-password \ // no authentication is required
--access-string "*off* +get ~keys*" // this disables the user
```

針對 Windows :

```
aws elasticache create-user ^
--user-id user1 ^
--user-name user1 ^
--engine "REDIS" ^
--no-password ^ // no authentication is required
--access-string "*off* +get ~keys*" // this disables the user
```

您將看到類似以下的回應 :

```
{
 "UserId": "user1",
 "UserName": "user1",
 "Status": "active",
 "Engine": "redis",
 "AccessString": "off ~keys* -@all +get",
 "UserGroupIds": [],
 "Authentication": {
 "Type": "no_password"
 },
 "ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
}
```

## 2. 建立秘密

若為 Linux、macOS 或 Unix :

```
aws secretsmanager create-secret \
--name production/ec/user1 \
--secret-string \
'{
 "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
 "username": "user1"
}'
```

針對 Windows :

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
 "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
 "username": "user1"
}'
```

您將看到類似以下的回應：

```
{
 "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
 "Name": "production/ec/user1",
 "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

### 3. 設定 Lambda 函數以輪換您的密碼

- a. 登入 AWS Management Console ，並在 開啟 Lambda 主控台 <https://console.aws.amazon.com/lambda/>
- b. 在導覽面板上，選擇 Functions (函數)，然後選擇您所建立的函數。請選擇函數名稱，而非其左側的核取方塊。
- c. 選擇 Configuration (組態) 索引標籤。
- d. 在 General configuration (一般組態) 中，選擇 Edit (編輯)，然後將 Timeout (逾時) 設定為至少 12 分鐘。
- e. 選擇 Save (儲存)。
- f. 選擇 Environment variables (環境變數)，然後設定下列項目：
  - i. SECRETS\_MANAGER\_ENDPOINT – <https://secretsmanager.REGION.amazonaws.com>
  - ii. SECRET\_ARN – 您在步驟 2 中建立的秘密的 Amazon Resource Name ( ARN )。
  - iii. USER\_NAME – ElastiCache 使用者的使用者名稱，
  - iv. 選擇 Save (儲存)。
- g. 選擇 Permissions (許可)
- h. 在執行角色 下，選擇要在 IAM 主控台上檢視的 Lambda 函數角色名稱。
- i. Lambda 函數需要下列許可，才能修改使用者和設定密碼：

## ElastiCache

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:DescribeUsers",
 "elasticache:ModifyUser"
],
 "Resource": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
 }
]
}
```

## Secrets Manager

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue",
 "secretsmanager:DescribeSecret",
 "secretsmanager:PutSecretValue",
 "secretsmanager:UpdateSecretVersionStage"
],
 "Resource": "arn:aws:secretsmanager:us-east-1:xxxxxxxxxx:secret:XXXX"
 },
 {
 "Effect": "Allow",
 "Action": "secretsmanager:GetRandomPassword",
 "Resource": "*"
 }
]
}
```

### 4. 設定 Secrets Manager 秘密輪換

- a. 使用 AWS Management Console，請參閱[使用主控台設定 AWS Secrets Manager 秘密的自動輪換](#)  
如需輪換排程的詳細資訊，請參閱 [Secrets Manager 輪換中的排程表達式](#)。
- b. 使用 AWS CLI，請參閱[設定 AWS Secrets Manager 使用的自動輪換 AWS Command Line Interface](#)

## 使用 驗證 IAM

### 主題

- [概觀](#)
- [限制](#)
- [設定](#)
- [連接](#)

### 概觀

使用IAM身分驗證時，當您的快取設定為使用 Valkey 或 Redis AWS IAM 第 OSS 7 OSS版或更新版本時，您可以使用身分驗證 ElastiCache 與 Valkey 或 Redis 的連線。這可讓您強化安全模型，並簡化許多管理安全任務。您也可以使用IAM身分驗證來設定每個個別 ElastiCache 快取和 ElastiCache 使用者的精細存取控制，並遵循最低權限許可原則。IAM ElastiCache 使用 Valkey 或 Redis 進行身分驗證 OSS的運作方式是提供短期IAM身分驗證權杖，而不是在 Valkey 或 Redis OSSAUTH或 HELLO命令中提供長期 ElastiCache 使用者密碼。如需有關IAM身分驗證權杖的詳細資訊，請參閱 AWS 一般參考指南中的[簽章第 4 版簽署程序](#)，以及下列程式碼範例。

您可以使用身分及其相關政策來進一步限制 Valkey IAM 或 Redis OSS的存取。您也可以將使用者從聯合身分提供者直接授予 Valkey 或 Redis OSS快取的存取權。

若要搭配 使用 AWS IAM ElastiCache，您必須先建立身分驗證模式設定為 ElastiCache 的使用者IAM。然後，您可以建立或重複使用IAM身分。IAM 身分需要相關聯的政策，才能將elasticache:Connect動作授予 ElastiCache 快取和 ElastiCache 使用者。設定完成後，您可以使用IAM使用者或角色的 AWS 憑證建立IAM身分驗證權杖。最後，您需要在連線至快取時，在 Valkey 或 Redis OSS用戶端中提供短期IAM身分驗證字符作為密碼。支援憑證提供者的 Valkey 或 Redis OSS用戶端可以為每個新連線自動產生臨時憑證。ElastiCache 將對IAM已啟用 ElastiCache 之使用者的連線請求執行IAM身分驗證，並將使用 驗證連線請求IAM。



## 限制

使用IAM身分驗證時，適用下列限制：

- IAM ElastiCache 與 Valkey 7.2 或更新版本以及 Redis 7.0.0 或更新版本搭配使用時，可以使用身分驗證。
- 對於IAM啟用 ElastiCache 的使用者，使用者名稱和使用者 ID 屬性必須相同。
- IAM 驗證權杖的有效期為 15 分鐘。對於長期連線，我們建議您使用支援憑證提供者介面的 Valkey 或 Redis OSS用戶端。
- ElastiCache 與 Valkey 或 Redis 的身分IAM驗證連線OSS將在 12 小時後自動中斷連線。使用新的 IAM身分驗證字符傳送 AUTH或 HELLO命令，可以延長連線 12 小時。
- IAM MULTI EXEC命令不支援身分驗證。
- 目前，IAM身分驗證支援下列全域條件內容金鑰：
  - 將IAM身分驗證與無伺服器快取搭配使用時，支援 `aws:VpcSourceIp`、`aws:SourceVpc`、`aws:CurrentTime`、`aws:EpochTime`、和 `aws:SourceVpc:aws:ResourceTag/%s` (來自相關聯的無伺服器快取和使用者)。
  - 搭配複寫群組使用IAM身分驗證時，支援 `aws:SourceIp`和 `aws:ResourceTag/%s` (來自相關聯的複寫群組和使用者)。

如需全域條件內容索引鍵的詳細資訊，請參閱 IAM 使用者指南中的[AWS 全域條件內容索引鍵](#)。

## 設定

若要設定IAM身分驗證：

### 1. 建立快取

```
aws elasticache create-serverless-cache \
 --serverless-cache-name cache-01 \
 --description "ElastiCache IAM auth application" \
 --engine redis
```

2. 為您的角色建立IAM信任政策文件，允許您的帳戶擔任新角色，如下所示。將政策儲存到名為 `trust-policy.json` 的檔案。

```
{
 "Version": "2012-10-17",
 "Statement": {
```

```
 "Effect": "Allow",
 "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
 "Action": "sts:AssumeRole"
 }
}
```

3. 建立IAM政策文件，如下所示。將政策儲存到名為 `policy.json` 的檔案。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect" : "Allow",
 "Action" : [
 "elasticache:Connect"
],
 "Resource" : [
 "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",
 "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
]
 }
]
}
```

4. 建立 IAM 角色。

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

5. 建立IAM政策。

```
aws iam create-policy \
--policy-name "elasticache-allow-all" \
--policy-document file://policy.json
```

6. 將IAM政策連接至角色。

```
aws iam attach-role-policy \
--role-name "elasticache-iam-auth-app" \
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

7. 建立新的IAM啟用的使用者。

```
aws elasticache create-user \
 --user-name iam-user-01 \
 --user-id iam-user-01 \
 --authentication-mode Type=iam \
 --engine redis \
 --access-string "on ~* +@all"
```

## 8. 建立使用者群組並連接使用者。

```
aws elasticache create-user-group \
 --user-group-id iam-user-group-01 \
 --engine redis \
 --user-ids default iam-user-01

aws elasticache modify-serverless-cache \
 --serverless-cache-name cache-01 \
 --user-group-id iam-user-group-01
```

## 連接

以字符做為密碼進行連線

首先，您需要使用 [AWS SigV4 預先簽署的請求](#) 產生短期IAM身分驗證權杖。在此之後，您在連線至 Valkey 或 Redis OSS快取時，以密碼形式提供IAM身分驗證權杖，如以下範例所示。

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for ElastiCache
// (Redis OSS).
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);
```

```
String iamAuthToken =
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
 .withHost(host)
 .withPort(port)
 .withSsl(ssl)
 .withAuthentication(userId, iamAuthToken)
 .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

以下是 IAMAuthTokenRequest 的定義。

```
public class IAMAuthTokenRequest {
 private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
 private static final String REQUEST_PROTOCOL = "http://";
 private static final String PARAM_ACTION = "Action";
 private static final String PARAM_USER = "User";
 private static final String PARAM_RESOURCE_TYPE = "ResourceType";
 private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
 private static final String ACTION_NAME = "connect";
 private static final String SERVICE_NAME = "elasticache";
 private static final long TOKEN_EXPIRY_SECONDS = 900;

 private final String userId;
 private final String cacheName;
 private final String region;
 private final boolean isServerless;

 public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
isServerless) {
 this.userId = userId;
 this.cacheName = cacheName;
 this.region = region;
 this.isServerless = isServerless;
 }

 public String toSignedRequestUri(AWSCredentials credentials) throws
URISyntaxException {
```

```
Request<Void> request = getSignableRequest();
sign(request, credentials);
return new URIBuilder(request.getEndpoint())
 .addParameters(toNamedValuePair(request.getParameters()))
 .build()
 .toString()
 .replace(REQUEST_PROTOCOL, "");
}

private <T> Request<T> getSignableRequest() {
 Request<T> request = new DefaultRequest<>(SERVICE_NAME);
 request.setHttpMethod(REQUEST_METHOD);
 request.setEndpoint(getRequestUri());
 request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
 request.addParameters(PARAM_USER, Collections.singletonList(userId));
 if (isServerless) {
 request.addParameters(PARAM_RESOURCE_TYPE,
Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
 }
 return request;
}

private URI getRequestUri() {
 return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
}

private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
 AWS4Signer signer = new AWS4Signer();
 signer.setRegionName(region);
 signer.setServiceName(SERVICE_NAME);

 DateTime dateTime = DateTime.now();
 dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));

 signer.presignRequest(request, credentials, dateTime.toDate());
}

private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
 return in.entrySet().stream()
 .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
 .collect(Collectors.toList());
}
}
```

## 使用憑證提供者進行連線

下列程式碼說明如何 ElastiCache 使用身分驗證憑證提供者進行IAM身分驗證。

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
// used as an
// IAM authentication token for ElastiCache (Redis OSS).
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);

// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
 userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
 .withHost(host)
 .withPort(port)
 .withSsl(ssl)
 .withAuthentication(redisCredentialsProvider)
 .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

以下是 Lettuce Redis OSS用戶端的範例，該用戶端會在憑證提供者IAMAuthTokenRequest中包裝，以在需要時自動產生臨時憑證。

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
 private static final long TOKEN_EXPIRY_SECONDS = 900;
```

```
private final AWSCredentialsProvider awsCredentialsProvider;
private final String userId;
private final IAMAuthTokenRequest iamAuthTokenRequest;
private final Supplier<String> iamAuthTokenSupplier;

public RedisIAMAuthCredentialsProvider(String userId,
 IAMAuthTokenRequest iamAuthTokenRequest,
 AWSCredentialsProvider awsCredentialsProvider) {
 this.userName = userName;
 this.awsCredentialsProvider = awsCredentialsProvider;
 this.iamAuthTokenRequest = iamAuthTokenRequest;
 this.iamAuthTokenSupplier =
Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
 TimeUnit.SECONDS);
}

@Override
public Mono<RedisCredentials> resolveCredentials() {
 return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
}

private String getIamAuthToken() {
 return
iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
}
}
```

## 使用 Valkey 和 Redis OSSAUTH命令進行驗證

### Note

AUTH 已由 [取代the section called “角色型存取控制 \( RBAC \)”](#)。所有無伺服器快取都必須使用 RBAC進行身分驗證。

Valkey 和 Redis OSS身分驗證權杖或密碼可讓 Valkey 和 Redis 在允許用戶端執行命令之前OSS要求密碼，藉此改善資料安全性。僅適用於AUTH自行設計的叢集。

### 主題

- [ElastiCache 使用 Valkey 和 Redis 在 AUTH中的 概觀 OSS](#)
- [ElastiCache 使用 Valkey 或 Redis OSS叢集將身分驗證套用至](#)

- [在現有叢集上修改AUTH權杖](#)
- [從 遷移RBAC至 AUTH](#)

ElastiCache 使用 Valkey 和 Redis 在 AUTH 中的 概觀 OSS

當您 AUTH 搭配 Valkey 或 Redis OSS 叢集使用 ElastiCache 時，有一些改進。

特別是，在使用時，請注意這些 AUTH 字符或密碼限制 AUTH：

- 字符 (或密碼) 必須為 16 - 128 個可列印字元。
- 非英數字元僅限使用 (!、&、#、\$、^、<、>、-)。
- AUTH 只能針對 ElastiCache 使用 Valkey 或 Redis OSS 叢集啟用的傳輸中加密啟用。

若要設定高強度的字符，建議您遵循嚴格的密碼政策，例如需符合以下要求：

- 權杖或密碼必須包含至少三種下列字元類型：
  - 大寫字元
  - 小寫字元
  - 數字
  - 非英數字元 (!、&、#、\$、^、<、>、-)
- 權杖或密碼不得包含字典單字或略微修改的字典單字。
- 權杖或密碼不得與最近使用的權杖相同或相似。

ElastiCache 使用 Valkey 或 Redis OSS 叢集將身分驗證套用至

您可以要求使用者在受權杖保護的 Valkey 或 Redis OSS 伺服器上輸入權杖 (密碼)。若要執行此操作，請在建立複寫群組或叢集時包含具有正確字符的參數 `--auth-token (API: AuthToken)`。而且也要在複寫群組或叢集的所有後續命令中加入。

下列 AWS CLI 操作會建立啟用傳輸中加密 (TLS) 和 AUTH 字符的複寫群組 *This-is-a-sample-token*。將子網路群組 `sng-test` 取代為已存在的子網路群組。

重要參數

- `--engine` – 必須為 `valkey` 或 `redis`。
- `--engine-version` – 如果引擎是 Redis OSS，則必須為 3.2.6、4.0.10 或更新版本。



- **--transit-encryption-enabled** – 身分驗證和HIPAA資格所需。
- **--auth-token** – HIPAA資格所需。此值必須是此受權杖保護的 Valkey 或 Redis OSS伺服器的正確權杖。
- **--cache-subnet-group** – HIPAA資格所需。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id authtestgroup \
 --replication-group-description authtest \
 --engine redis \
 --cache-node-type cache.m4.large \
 --num-node-groups 1 \
 --replicas-per-node-group 2 \
 --transit-encryption-enabled \
 --auth-token This-is-a-sample-token \
 --cache-subnet-group sng-test
```

針對 Windows：

```
aws elasticache create-replication-group ^
 --replication-group-id authtestgroup ^
 --replication-group-description authtest ^
 --engine redis ^
 --cache-node-type cache.m4.large ^
 --num-node-groups 1 ^
 --replicas-per-node-group 2 ^
 --transit-encryption-enabled ^
 --auth-token This-is-a-sample-token ^
 --cache-subnet-group sng-test
```

在現有叢集上修改AUTH權杖

若要更輕鬆地更新身分驗證，您可以修改叢集上使用的AUTH權杖。如果引擎版本為 Valkey 7.2 或更新版本，或 Redis 5.0.6 或更新版本，您可以進行此修改。ElastiCache 也必須啟用傳輸中的加密。

修改身分驗證權杖支援兩種策略：ROTATE和SET。ROTATE 策略會將額外的AUTH權杖新增至伺服器，同時保留先前的權杖。SET 策略會更新伺服器，僅支援單一AUTH權杖。使用 `--apply-immediately` 參數進行這些修改呼叫以立即套用變更。

## 輪換AUTH權杖

若要使用新的AUTH權杖更新 Valkey 或 Redis OSS 伺服器，請使用 `ModifyReplicationGroup` API `--auth-token` 參數呼叫做為新的AUTH權杖，並使用值 `--auth-token-update-strategy` 呼叫 `ROTATE`。ROTATE 修改完成後，除了 `auth-token` 參數中指定的權杖之外，叢集還支援先前的AUTH權杖。如果在權AUTH杖輪換之前未在複寫群組上設定AUTH權杖，則叢集除了支援無需身分驗證的連線之外，還支援 `--auth-token` 參數中指定的AUTH權杖。請參閱 [設定AUTH權杖](#) 以使用更新策略將AUTH權杖更新為必要SET。

### Note

如果您之前未設定AUTH權杖，則修改完成後，除了驗證權杖參數中指定的AUTH權杖之外，叢集將不支援權杖。

如果在已支援兩個AUTH權杖的伺服器上執行此修改，則在此操作期間也會移除最舊的AUTH權杖。這可讓伺服器在指定時間支援最多兩個最新的AUTH權杖。

此時，您可以透過更新用戶端以使用最新的AUTH權杖繼續。更新用戶端後，您可以使用AUTH權杖輪換SET策略（在下一節中說明）來專門開始使用新的權杖。

下列 AWS CLI 操作會修改複寫群組以輪換AUTH權杖 *This-is-the-rotated-token*。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-rotated-token \
--auth-token-update-strategy ROTATE \
--apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-rotated-token ^
--auth-token-update-strategy ROTATE ^
--apply-immediately
```

## 設定AUTH權杖

若要更新 Valkey 或 Redis OSS 伺服器以支援單一必要的AUTH權杖，請使用與最後一個權AUTH杖具有相同值的 `--auth-token` 參數和具有值的 `--auth-token-update-strategy` 參數來呼叫 `ModifyReplicationGroupAPI`操作SET。此SET策略只能與具有 2 個AUTH權杖或 1 個先前使用 ROTATE策略之選用AUTH權杖的叢集搭配使用。修改完成後，伺服器僅支援在 `auth-token` 參數中指定的AUTH權杖。

下列 AWS CLI 操作會修改複寫群組，將AUTH權杖設定為 *This-is-the-set-token*。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-set-token \
--auth-token-update-strategy SET \
--apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^\
--replication-group-id authtestgroup ^\
--auth-token This-is-the-set-token ^\
--auth-token-update-strategy SET ^\
--apply-immediately
```

## 在現有叢集上啟用身分驗證

若要在現有的 Valkey 或 Redis OSS 伺服器上啟用身分驗證，請呼叫 `ModifyReplicationGroupAPI`操作。 `ModifyReplicationGroup` 使用 `--auth-token` 參數作為新權杖呼叫，並使用值 `--auth-token-update-strategy`呼叫 ROTATE。

ROTATE 修改完成後，叢集除了支援無需身分驗證的連線之外，還支援 `--auth-token` 參數中指定的AUTH權杖。更新所有用戶端應用程式以OSS使用AUTH權杖對 Valkey 或 Redis 進行身分驗證後，請使用 SET策略將AUTH權杖標記為必要。只有啟用傳輸中加密（TLS）的 Valkey 和 Redis OSS 伺服器才支援啟用身分驗證。

## 從 遷移RBAC至 AUTH

如果您使用 Valkey 或 Redis OSS角色型存取控制（RBAC）驗證使用者[角色型存取控制（RBAC）](#)，如所述，而且您想要遷移至 AUTH，請使用下列程序。您可以使用主控台或進行遷移 CLI。

## AUTH 使用主控台從 遷移RBAC至

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中，選擇要修改的叢集所在的 AWS 區域。
3. 在導覽窗格中，選擇您要修改之叢集所執行的引擎。

接著會出現所選引擎的叢集清單。

4. 在叢集清單中，針對您要修改的叢集，選擇其名稱。
5. 在 Actions (動作) 中，選擇 Modify (修改)。

修改視窗隨即出現。

6. 針對存取控制，選擇 Valkey AUTH 預設使用者存取權或 Redis OSS AUTH 預設使用者存取權。
7. 在 Valkey AUTH權杖 或 Redis OSSAUTH權杖 下，設定新的權杖。
8. 選擇預覽變更，然後在下一個畫面上選擇修改。

## AUTH使用 從 遷移RBAC至 AWS CLI

使用下列其中一個命令，為您的 Valkey 或 Redis OSS複寫群組設定新的選用AUTH權杖。請注意，選用的身分驗證權杖將允許未經驗證的複寫群組存取權，直到驗證權杖標示為必要為止，方法是在下列步驟SET中使用更新策略。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id test \
 --remove-user-groups \
 --auth-token This-is-a-sample-token \
 --auth-token-update-strategy ROTATE \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id test ^
 --remove-user-groups ^
 --auth-token This-is-a-sample-token ^
```

```
--auth-token-update-strategy ROTATE ^
--apply-immediately
```

執行上述命令後，您可以使用新設定的選用AUTH權杖，更新您的 Valkey 或 Redis OSS 應用程式以驗證 ElastiCache 複寫群組。若要完成驗證權杖輪換，請在下列後續命令SET中使用更新策略。這將視需要標記為選用AUTH權杖。驗證權杖更新完成時，複寫群組狀態會顯示為 ACTIVE，且此複寫群組的所有連線都需要身分驗證。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id test \
 --auth-token This-is-a-sample-token \
 --auth-token-update-strategy SET \
 --apply-immediately
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id test ^
 --remove-user-groups ^
 --auth-token This-is-a-sample-token ^
 --auth-token-update-strategy SET ^
 --apply-immediately
```

如需詳細資訊，請參閱[使用 Valkey 和 Redis OSSAUTH命令進行驗證](#)。

#### Note

如果您需要停用 ElastiCache 叢集上的存取控制，請參閱 [the section called “在 Valkey 或 Redis ElastiCacheOSS快取上停用存取控制”](#)。

在 Valkey 或 Redis ElastiCacheOSS快取上停用存取控制

請依照下列指示，停用啟用 Valkey 或 Redis OSS TLS之快取的存取控制。您的快取將有兩種不同類型的組態之一：AUTH預設使用者存取權或使用者群組存取控制清單（RBAC）。如果您的快取是使用 AUTH組態建立的，您必須先將其變更為RBAC組態，才能透過移除使用者群組來停用快取。如果您的快取是使用 RBAC 組態建立的，您可以直接停用它。

## 若要停用使用 設定的 Valkey 或 Redis 無OSS伺服器快取 RBAC

1. 移除使用者群組以停用存取控制。

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
--remove-user-group
```

2. (選用) 確認沒有與無伺服器快取相關聯的使用者群組。

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
cache>
{
 "...",
 "UserGroupId": ""
 "...",
}
```

## 若要使用以AUTH權杖設定的 停用 Valkey 或 Redis OSS快取

1. 將AUTH權杖變更為 , RBAC並指定要新增的使用者群組。

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-
value>
```

2. 確認AUTH字符已停用，且已新增使用者群組。

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-id-value>
{
 "...",
 "AuthTokenEnabled": false,
 "UserGroupIds": [
 "<user-group-value>"
]
 "...",
}
```

3. 移除使用者群組以停用存取控制。

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
 "...
 "PendingModifiedValues": {
 "UserGroups": {
 "UserGroupIdsToAdd": [],
 "UserGroupIdsToRemove": [
 "<user-group-value>"
]
 }
 }
 "...
}
```

4. (選用) 確認沒有與叢集相關聯的使用者群組。AuthTokenEnabled 欄位也應顯示 false。

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
"AuthTokenEnabled": false
```

若要停用使用設定的 Valkey 或 Redis OSS叢集 RBAC

1. 移除使用者群組以停用存取控制。

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
 "...
 "PendingModifiedValues": {
 "UserGroups": {
 "UserGroupIdsToAdd": [],
 "UserGroupIdsToRemove": [
 "<user-group-value>"
]
 }
 }
 "...
}
```

2. (選用) 確認沒有與叢集相關聯的使用者群組。AuthTokenEnabled 欄位也應顯示 false。

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

## 網際網路流量隱私權

Amazon ElastiCache 使用下列技術來保護快取資料，並防止未經授權的存取：

- [Amazon VPCs和安全性 ElastiCache](#) 說明安裝所需的安全群組類型。
- [Amazon 的身分和存取管理 ElastiCache](#) 用於授予和限制使用者、群組和角色的動作。

### 主題

- [Amazon VPCs和安全性 ElastiCache](#)
- [ElastiCache API 和介面VPC端點 \( AWS PrivateLink \)](#)
- [子網路和子網路群組](#)

## Amazon VPCs和安全性 ElastiCache

由於資料安全很重要，ElastiCache 為您提供方法來控制誰可以存取您的資料。控制資料存取的方式取決於您是否在 Amazon Virtual Private Cloud ( Amazon VPC ) 或 Amazon EC2-Classic 中啟動叢集。

### Important

我們已取代使用 Amazon EC2-Classic 啟動 ElastiCache 叢集。所有目前世代的節點只會在 Amazon Virtual Private Cloud 中啟動。

Amazon Virtual Private Cloud ( AmazonVPC ) 服務定義了與傳統資料中心非常相似的虛擬網路。設定 Amazon 時VPC，您可以選取其 IP 地址範圍、建立子網路，以及設定路由表、網路閘道和安全設定。您也可以將快取叢集新增至虛擬網路，並使用 Amazon VPC安全群組控制對快取叢集的存取。

本節說明如何在 Amazon 中手動設定 ElastiCache 叢集VPC。此資訊適用於想要深入了解 ElastiCache 和 Amazon 如何一起VPC運作的使用者。

### 主題



- [了解 ElastiCache 和 Amazon VPCs](#)
- [在 Amazon 中存取 ElastiCache 快取的存取模式 VPC](#)
- [建立虛擬私有雲端 \( VPC \)](#)
- [連線至在 Amazon 中執行的快取 VPC](#)

## 了解 ElastiCache 和 Amazon VPCs

ElastiCache 已與 Amazon Virtual Private Cloud ( Amazon ) 完全整合VPC。對於 ElastiCache 使用者，這表示下列事項：

- 如果 AWS 您的帳戶僅支援 EC2-VPC 平台，ElastiCache 一律會在 Amazon 中啟動叢集VPC。
- 如果您是的新手 AWS，您的叢集將部署到 Amazon VPC。系統VPC會自動為您建立預設值。
- 如果您有預設值VPC，而且在啟動叢集時未指定子網路，則叢集會啟動至預設的 Amazon VPC。

如需詳細資訊，請參閱[偵測支援的平台，以及您是否有預設 VPC](#)。

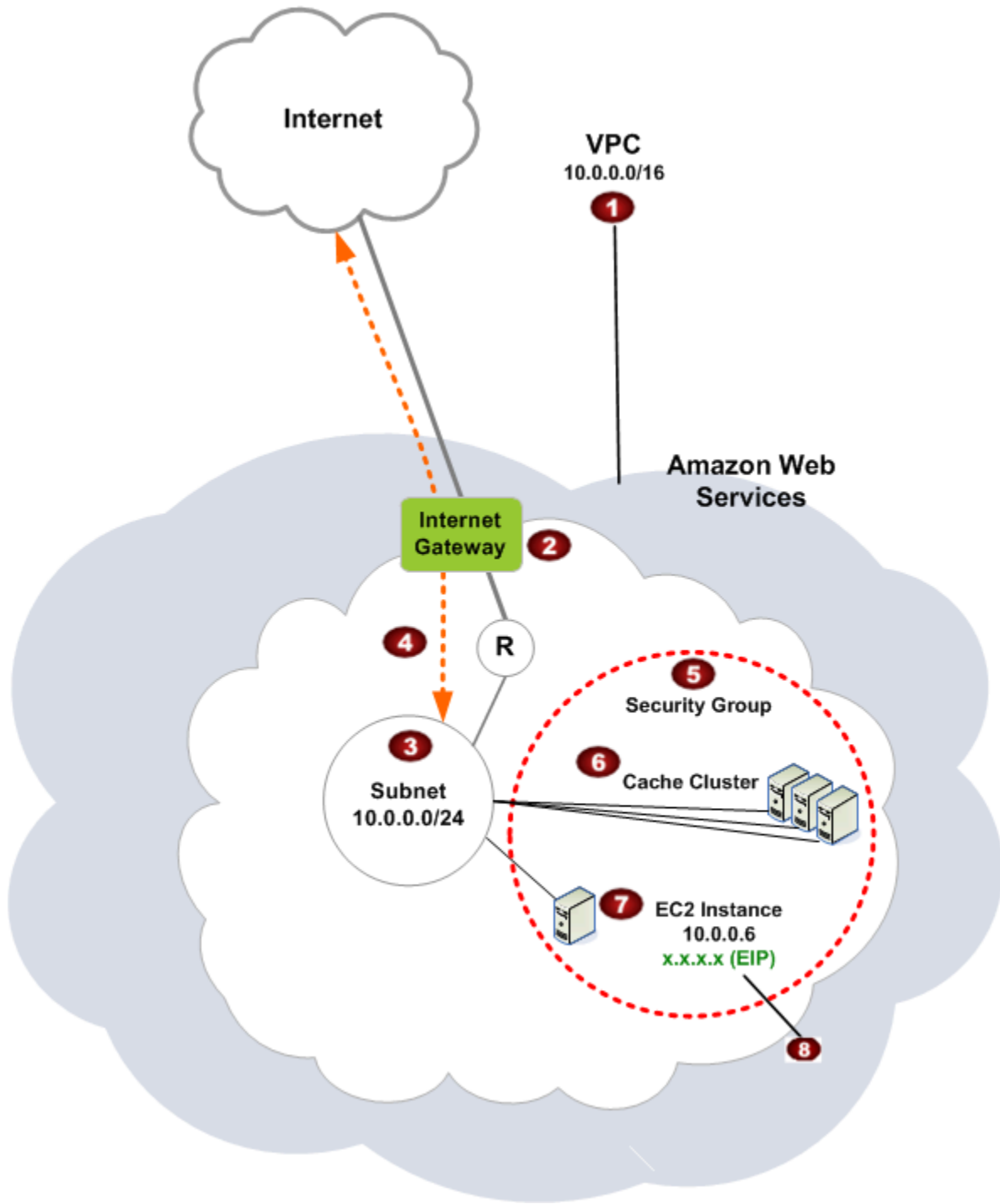
透過 Amazon Virtual Private Cloud，您可以在 AWS 雲端中建立與傳統資料中心非常相似的虛擬網路。您可以設定 Amazon VPC，包括選取其 IP 地址範圍、建立子網路，以及設定路由表、網路閘道和安全設定。

的基本功能 ElastiCache 在虛擬私有雲端中相同；無論您的叢集部署在 Amazon 內外，都 ElastiCache 管理軟體升級、修補、故障偵測和復原VPC。

ElastiCache 在 Amazon 外部部署的快取節點VPC會指派端點/DNS名稱解析的 IP 地址。這提供來自 Amazon Elastic Compute Cloud ( Amazon EC2 ) 執行個體的連線。當您在 Amazon VPC私有子網路中啟動 ElastiCache 叢集時，每個快取節點都會在該子網路內指派私有 IP 地址。

### Amazon ElastiCache 中的 概觀 VPC

下圖和表格描述了 Amazon VPC環境，以及在 Amazon 中啟動的 ElastiCache 叢集和 Amazon EC2執行個體VPC。



1

Amazon VPC是 AWS Cloud 的隔離部分，會指派自己的 IP 地址區塊。

2

網際網路閘道會將您的 Amazon VPC直接連線至網際網路，並提供其他 AWS 資源的存取權，例如在 Amazon 外部執行的 Amazon Simple Storage Service ( Amazon S3 ) VPC。

3

Amazon VPC子網路是 Amazon IP 地址範圍的區段VPC，您可以根據安全和操作需求隔離 AWS 資源。

4

Amazon 中的路由表會VPC引導子網路與網際網路之間的網路流量。Amazon VPC具有隱含路由器，此圖中由 R 的圓圈表示。

5

Amazon VPC安全群組會控制 ElastiCache 叢集和 Amazon EC2執行個體的傳入和傳出流量。

6

您可以在子網路中啟動 ElastiCache 叢集。快取節點具有來自子網路地址範圍的私有 IP 地址。

7

您也可以在此子網路中啟動 Amazon EC2執行個體。每個 Amazon EC2執行個體都有一個來自子網路位址範圍的私有 IP 地址。Amazon EC2執行個體可以連線到相同子網路中的任何快取節點。

8

若要VPC讓 Amazon 中的 Amazon EC2執行個體能夠從網際網路連線，您需要將稱為彈性 IP 地址的靜態公有地址指派給執行個體。

## 必要條件

若要在 Amazon 內建立 ElastiCache 叢集VPC，您的 Amazon VPC必須符合下列要求：

- Amazon 必須VPC允許非專用 Amazon EC2執行個體。您無法 ElastiCache 在為專用執行個體租用 VPC設定的 Amazon 中使用。
- 必須為您的 Amazon VPC. ElastiCache uses 定義快取子網路群組，以選取該子網路內的子網路和 IP 地址，以與您的VPC端點或快取節點建立關聯。
- CIDR 每個子網路的 區塊必須足夠大，以提供備用 IP 地址 ElastiCache 供維護活動期間使用。

## 路由和安全性

您可以在 Amazon 中設定路由，VPC 以控制流量流向（例如，至網際網路閘道或虛擬私有閘道）。透過網際網路閘道，您的 Amazon VPC 可以直接存取其他未在 Amazon 中執行 AWS 的資源 VPC。如果您選擇只擁有與組織本機網路連線的虛擬私有閘道，您可以透過路由網際網路繫結流量 VPN，並使用本機安全政策和防火牆來控制輸出。在這種情況下，當您透過網際網路存取 AWS 資源時，會產生額外的頻寬費用。

您可以使用 Amazon VPC 安全群組來協助保護 Amazon 中的 ElastiCache 叢集和 Amazon EC2 執行個體 VPC。安全群組會在執行個體層級（而非子網路層級）以防火牆形式運作。

### Note

我們強烈建議您使用 DNS 名稱來連線至快取節點，因為基礎 IP 地址可能會變更。

## Amazon VPC 文件

Amazon VPC 有自己的文件集，說明如何建立和使用您的 Amazon VPC。下表提供 Amazon VPC 指南的連結。

描述	文件
如何開始使用 Amazon VPC	<a href="#">Amazon 入門 VPC</a>
如何 VPC 透過使用 Amazon AWS Management Console	<a href="#">Amazon VPC 使用者指南</a>
所有 Amazon VPC 命令的完整描述	<a href="#">Amazon EC2 Command Line 參考</a> （Amazon VPC 命令可在 Amazon EC2 參考中找到）
Amazon VPC API 操作、資料類型和錯誤的完整描述	<a href="#">Amazon EC2 API 參考</a> （Amazon VPC API 操作可在 Amazon EC2 參考中找到）
適用於需要在選擇性 IPsec VPN 連線結束時設定閘道的網路管理員的資訊	<a href="#">什麼是 AWS Site-to-Site VPN？</a>

如需更多 Amazon Virtual Private Cloud 的詳細資訊，請參閱 [Amazon Virtual Private Cloud](#)。

## 在 Amazon 中存取 ElastiCache 快取的存取模式 VPC

Amazon ElastiCache 支援下列存取 Amazon 快取的案例VPC：

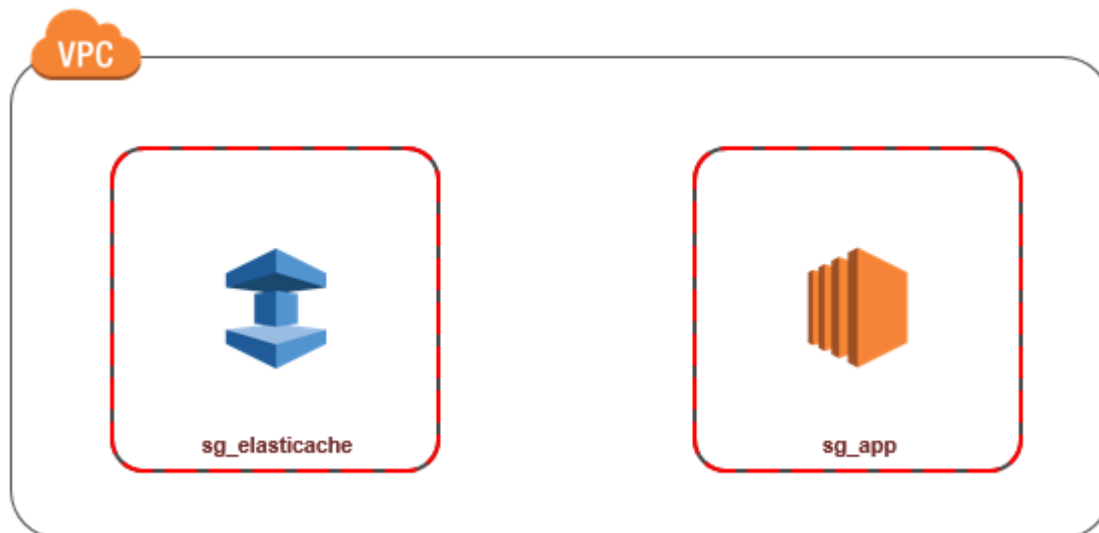
### 內容

- [當 ElastiCache 快取和 Amazon EC2執行個體位於相同 Amazon 時，存取快取 VPC](#)
- [當 ElastiCache 快取和 Amazon EC2執行個體位於不同的 Amazon 時，存取快取 VPCs](#)
  - [當快取和 Amazon EC2執行個體位於相同區域中的不同 Amazon VPCs時，存取 ElastiCache 快取](#)
    - [使用 Transit Gateway](#)
  - [當 ElastiCache 快取和 Amazon EC2執行個體位於不同區域的不同 Amazon VPCs時，存取快取](#)
    - [使用 Transit VPC](#)
- [從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
  - [使用VPN連線從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
  - [使用 Direct Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)

當 ElastiCache 快取和 Amazon EC2執行個體位於相同 Amazon 時，存取快取 VPC

最常見的使用案例是部署在EC2執行個體上的應用程式需要連線到相同 中的快取VPC。

下圖說明此案例。



管理相同EC2執行個體和快取之間存取的最簡單方法是VPC執行下列動作：

1. 為您的快取建立VPC安全群組。此安全群組可用來限制快取的存取權。例如，您可以為此安全群組建立自訂規則，以允許TCP使用您在建立快取時指派給快取的連接埠和用於存取快取的 IP 地址進行存取。

Memcached 快取的預設連接埠是 11211。

Valkey 和 Redis OSS快取的預設連接埠為 6379。

2. 為您的EC2執行個體（網路和應用程式伺服器）建立VPC安全群組。如有需要，此安全群組可以透過的VPC路由表允許從網際網路存取EC2執行個體。例如，您可以在此安全群組上設定規則，以允許透過連接埠 22 TCP存取EC2執行個體。
3. 在快取的安全群組中建立自訂規則，以允許來自您為EC2執行個體建立的安全群組的連線。這樣做會允許安全群組的所有成員存取快取。

#### Note

如果您打算使用[本地區域](#)，請確認本地區域已啟用。當您在該本機區域建立子網路群組時，您的 VPC 會延伸到該本機區域，而您的 VPC會將子網路視為任何其他可用區域中的任何子網路。所有相關閘道和路由表都將自動調整。

在允許從其他VPC安全群組連線的安全群組中建立規則

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/vpc> 開啟 Amazon VPC主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選取或建立您將用於快取的安全群組。在 Inbound Rules (傳入規則) 下方，選取 Edit Inbound Rules (編輯傳入規則)，然後選取 Add Rule (新增規則)。此安全群組將允許其他安全群組成員存取。
4. 從類型選擇自訂TCP規則。

- a. 針對連接埠範圍，指定您在建立快取時所使用的連接埠。

Memcached 快取的預設連接埠是 11211。

Valkey 和 Redis OSS快取和複寫群組的預設連接埠為 6379。

- b. 在 Source (來源) 方塊中輸入安全群組的 ID。從清單中選取您要用於 Amazon EC2執行個體的安全群組。
5. 完成後，請選擇 Save (儲存)。

Inbound rules (3)									
Filter security group rules									
	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sgr-	IPv4	SSH	TCP	22	0.0.0.0/0	-	
<input type="checkbox"/>	-	sgr-	-	Custom TCP	TCP	6379	sg- / default	-	

當 ElastiCache 快取和 Amazon EC2 執行個體位於不同的 Amazon 時，存取快取 VPCs

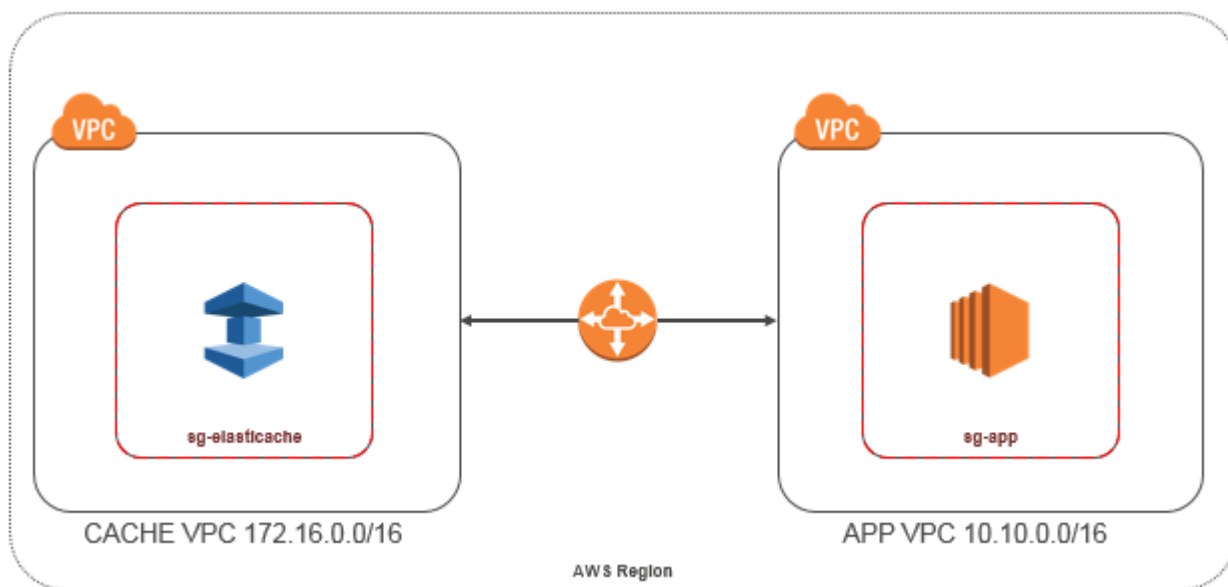
當您的快取與您用來存取快取的 EC2 執行個體 VPC 不同時，有幾種方式可存取快取。如果快取和 EC2 執行個體位於不同的區域，VPCs 但位於相同的區域，您可以使用 VPC 對等。如果快取和 EC2 執行個體位於不同區域，您可以在區域之間建立 VPN 連線。

## 主題

- [當快取和 Amazon EC2 執行個體位於相同區域中的不同 Amazon VPCs 時，存取 ElastiCache 快取](#)
- [當 ElastiCache 快取和 Amazon EC2 執行個體位於不同區域的不同 Amazon VPCs 時，存取快取](#)

當快取和 Amazon EC2 執行個體位於相同區域中的不同 Amazon VPCs 時，存取 ElastiCache 快取

下圖說明使用 Amazon VPC 對等連線，由相同區域中不同 Amazon VPC 中的 Amazon EC2 執行個體存取快取。



Amazon EC2 執行個體在相同區域內不同 Amazon VPC 存取的快取 - VPC 對等連線



VPC 對等連線是兩個之間的網路連線VPCs，可讓您使用私有 IP 地址路由它們之間的流量。任一中的執行個體VPC都可以彼此通訊，就像在相同的網路中一樣。您可以在自己的 Amazon 之間建立VPC 對等連線VPCs，也可以VPC在單一區域中的另一個 AWS 帳戶中建立 Amazon。若要進一步了解 Amazon VPC互連，請參閱 [VPC 文件](#)。

### Note

DNS 對等的名稱解析可能會失敗VPCs，取決於套用至的組態 ElastiCache VPC。若要解決此問題，VPCs必須為DNS主機名稱和DNS解析度啟用兩者。如需詳細資訊，請參閱[啟用對VPC 等連線的DNS解析](#)。

透過VPC對等存取不同 Amazon 中的快取

1. 請確定兩者VPCs沒有重疊的 IP 範圍，否則您將無法對它們進行對等。
2. 對兩個進行對等VPCs。如需詳細資訊，請參閱[建立和接受 Amazon VPC Peering Connection](#)。
3. 更新您的路由表。如需詳細資訊，請參閱[更新VPC對等連線的路由表](#)

下列是上述圖表中用於示範的路由表。請注意，pcx-a894f1c1 是互連連線。

Destination	Target	Destination	Target
172.16.0.0/16	local	10.10.0.0/16	local
10.10.0.0/16	pcx-a894f1c1	0.0.0.0/0	igw-bfdcccd8
		172.16.0.0/16	pcx-a894f1c1

### VPC 路由表

4. 修改 ElastiCache 快取的安全群組，以允許來自對等中應用程式安全群組的傳入連線VPC。如需詳細資訊，請參閱[參考對等VPC安全群組](#)。

透過對等互連連線存取快取，將產生額外的資料傳輸費用。

### 使用 Transit Gateway

傳輸閘道可讓您在相同 AWS 區域中連接VPCs和VPN連線，並在它們之間路由流量。傳輸閘道可跨 AWS 帳戶運作，您可以使用 AWS Resource Access Manager 與其他帳戶共用傳輸閘道。與其他 AWS 帳戶共用傳輸閘道後，帳戶擁有者可以將它們連接至VPCs您的傳輸閘道。這些帳戶的使用者均可隨時刪除連接。

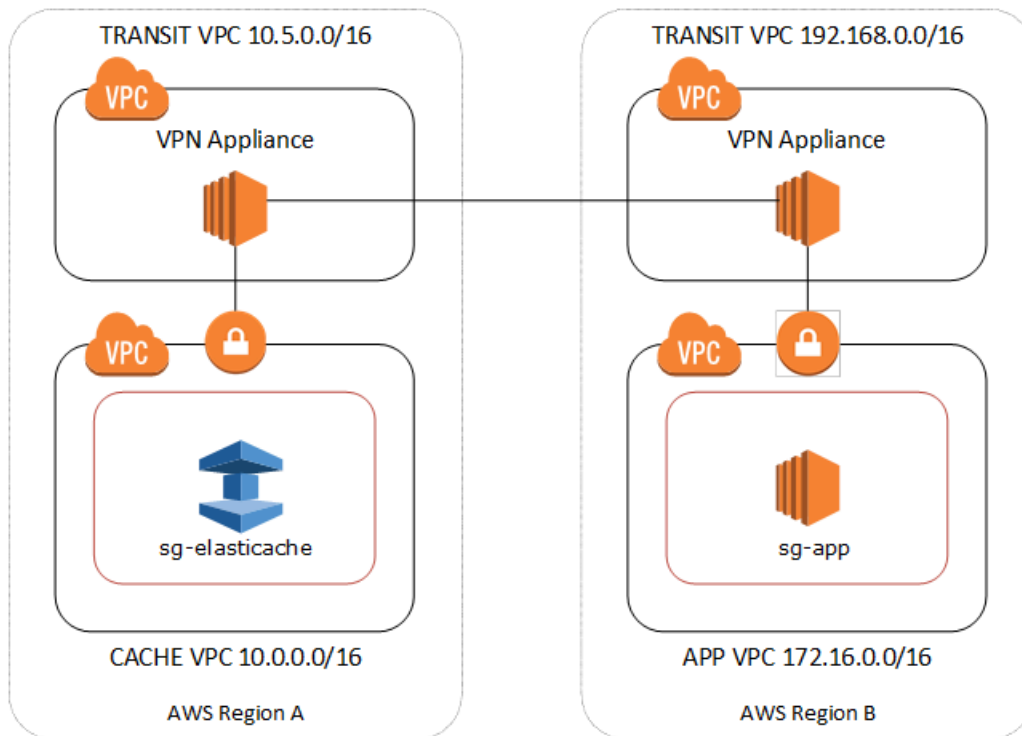
您可以在傳輸閘道上啟用多點傳送，然後建立傳輸閘道多點傳送網域，允許多點傳送流量從多點傳送來源傳送至多點傳送群組成員，透過與網域相關聯的VPC附件。

您也可以在不同 AWS 區域中的傳輸閘道之間建立對等連線連接。這可讓您跨不同區域在傳輸閘道附件之間路由流量。

如需詳細資訊，請參閱[傳輸閘道](#)。

當 ElastiCache 快取和 Amazon EC2執行個體位於不同區域的不同 Amazon VPCs時，存取快取使用 Transit VPC

除了使用VPC互連之外，另一個連接多個分散地理VPCs和遠端網路的常見策略是建立VPC充當全球網路傳輸中心的傳輸。傳輸可VPC簡化網路管理，並將連接多個VPCs和遠端網路所需的連線數量降至最低。此設計可以節省時間和精力並降低費用，因為實際上不具有在託管傳輸中樞建立實體存在或部署實體網路設備的傳統支出。



在不同區域中跨不同VPCs區域連線

建立 Transit Amazon VPC 後，部署在一個區域中的「輪換」VPC中的應用程式可以連接到VPC另一個區域中的「輪換」中的 ElastiCache 快取。

在不同 AWS 區域中存取不同 VPC中的快取

1. 部署 Transit VPC 解決方案。如需詳細資訊，請參閱[AWS Transit Gateway](#)。

2. 更新應用程式和快取中的VPC路由表VPCs，以透過 VGW ( 虛擬私有閘道 ) 和 VPN 裝置路由流量。如果使用邊界閘道通訊協定 ( BGP ) 動態路由，您的路由可能會自動傳播。
3. 修改 ElastiCache 快取的安全群組，以允許從應用程式執行個體 IP 範圍傳入連線。請注意，在此情況下，您將無法參考應用程式伺服器安全群組。

跨區域存取快取將造成網路延遲，並產生額外的跨區域資料傳輸費用。

從客戶資料中心執行的應用程式存取 ElastiCache 快取

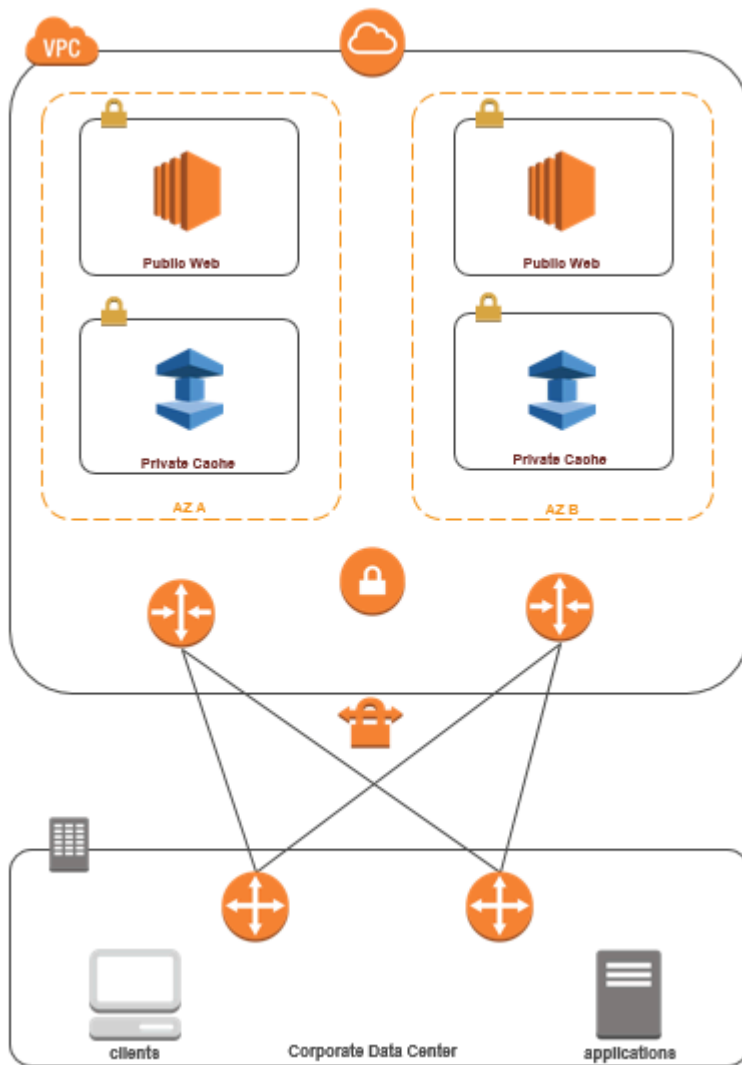
另一個可能的情況是混合架構，其中客戶資料中心中的用戶端或應用程式可能需要存取 中的 ElastiCache 快取VPC。如果客戶的 VPC與資料中心之間透過 VPN或 Direct Connect 存在連線，則也支援此案例。

主題

- [使用VPN連線從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
- [使用 Direct Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)

使用VPN連線從客戶資料中心執行的應用程式存取 ElastiCache 快取

下圖說明使用 VPN 連線從公司網路中執行的應用程式存取 ElastiCache 快取。



透過 ElastiCache 從您的資料中心連線至 VPN

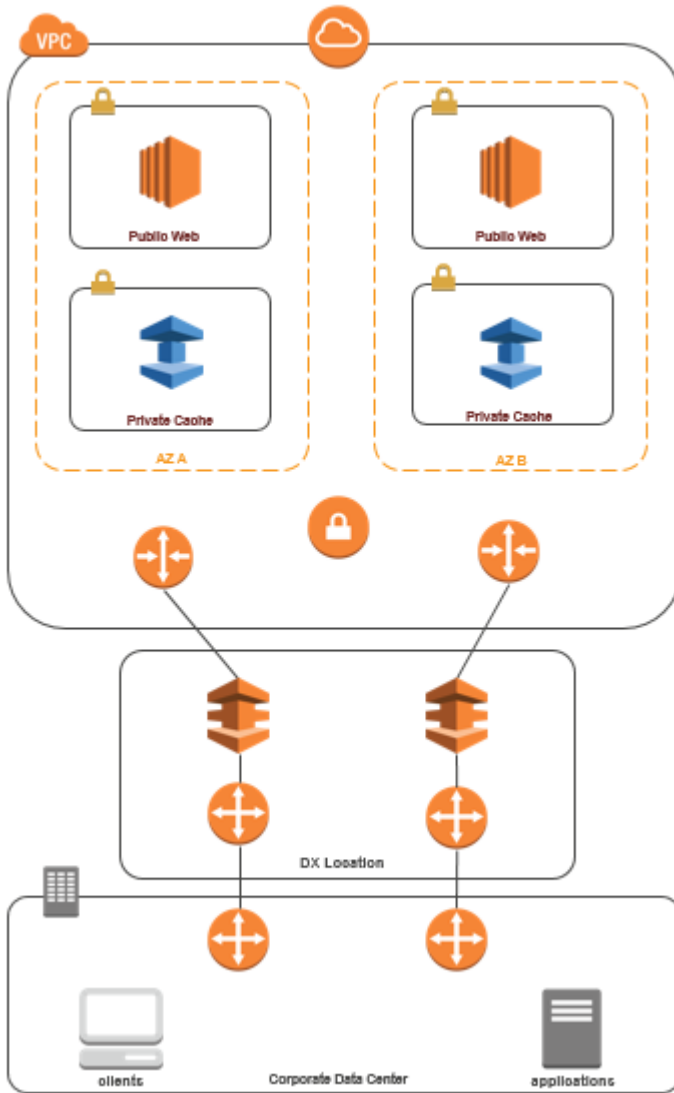
透過VPN連線VPC從內部部署應用程式存取 中的快取

1. 透過將硬體虛擬私有閘道新增至您的 來建立VPN連線VPC。如需詳細資訊，請參閱[將硬體虛擬私有閘道新增至您的 VPC](#)。
2. 更新部署 ElastiCache 快取的子網路VPC路由表，以允許來自內部部署應用程式伺服器的流量。如果動態路由搭配BGP您的路由，則可能會自動傳播。
3. 修改 ElastiCache 快取的安全群組，以允許從內部部署應用程式伺服器傳入連線。

透過VPN連線存取快取將帶來聯網延遲和其他資料傳輸成本。

## 使用 Direct Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取

下圖說明使用 Direct Connect 從公司網路上執行的應用程式存取 ElastiCache 快取。



### 透過 Direct Connect ElastiCache 從資料中心連線至

#### 使用 Direct Connect 從網路中執行的應用程式存取 ElastiCache 快取

1. 建立 Direct Connect 連線。如需詳細資訊，請參閱 [AWS Direct Connect 入門](#)。
2. 修改 ElastiCache 快取的安全群組，以允許從內部部署應用程式伺服器傳入連線。

透過 DX 連線存取快取可能造成網路延遲，並產生額外的跨區域資料傳輸費用。

## 建立虛擬私有雲端 ( VPC )

在此範例中，您會為每個可用區域建立VPC具有私有子網路的 Amazon。

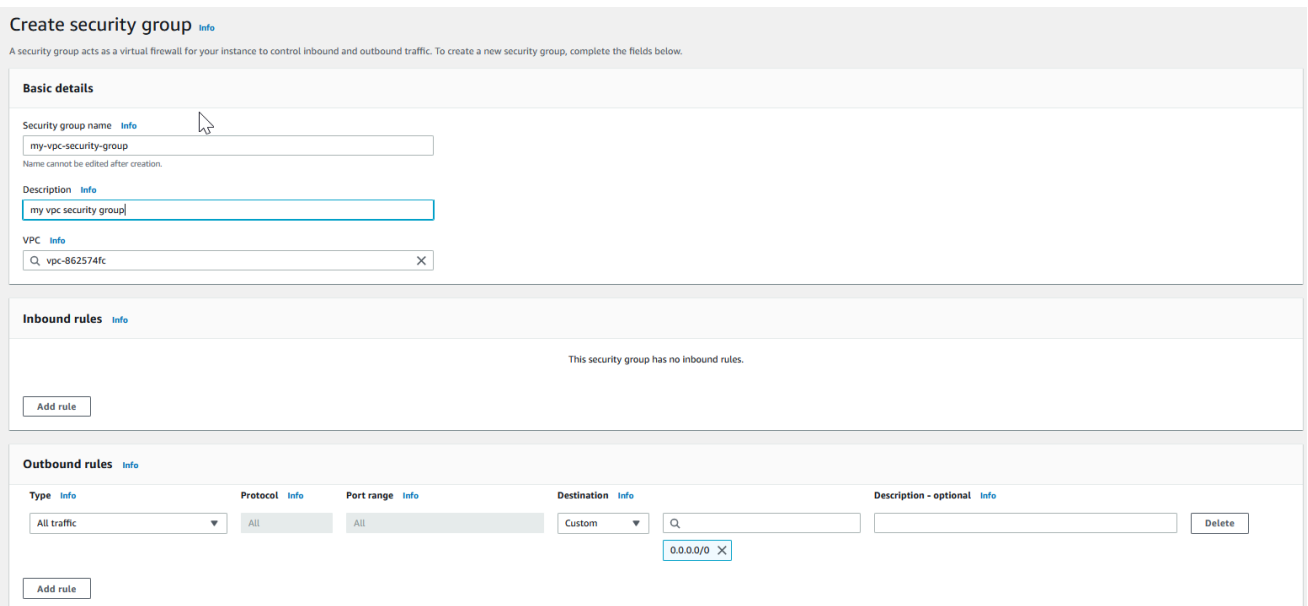
### 建立 Amazon VPC ( 主控台 )

1. 登入 AWS 管理主控台，並在 開啟 Amazon VPC主控台<https://console.aws.amazon.com/vpc/>。
2. 在VPC儀表中，選擇建立 VPC。
3. 在要建立的資源下，選擇 VPC 和更多。
4. 在可用區域數目 ( AZs ) 下，選擇要在其中啟動子網路的可用區域數目。
5. 在公有子網路數量 下，選擇您要新增至 的公有子網路數量VPC。
6. 在私有子網路數量 下，選擇您要新增至 的私有子網路數量VPC。

#### Tip

記下您的子網路識別符，分別為公有和私有。稍後當您啟動叢集並將 Amazon EC2執行個體新增至 Amazon 時，您將需要此資訊VPC。

7. 建立 Amazon VPC安全群組。您將使用此群組作為快取叢集和 Amazon EC2執行個體。
  - a. 在 Amazon VPC Management 主控台的導覽窗格中，選擇安全群組。
  - b. 選擇建立安全群組。
  - c. 在對應的方塊中，為安全群組輸入名稱和描述。在 VPC方塊中，選擇 Amazon 的識別碼 VPC。



Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)  
my-vpc-security-group  
Name cannot be edited after creation.

Description [Info](#)  
my vpc security group

VPC [Info](#)  
vpc-862574fc

**Inbound rules** [Info](#)

This security group has no inbound rules.

[Add rule](#)

**Outbound rules** [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
All traffic	All	All	Custom	<input type="text" value="0.0.0.0/0"/>

[Delete](#)

[Add rule](#)

- d. 當您滿意設定後，請選擇 Yes, Create (是，建立)。
8. 為您的安全群組定義網路傳入規則。此規則可讓您使用 Secure Shell ( ) 連線至 Amazon EC2執行個體SSH。
- a. 在導覽清單中，選擇 Security Groups (安全群組)。
  - b. 在清單中找到您的安全群組，然後選擇它。
  - c. 在 Security Group (安全群組) 下，選擇 Inbound (入站) 標籤。在建立新規則方塊中，選擇 SSH，然後選擇新增規則。
  - d. 為新的傳入規則設定下列值，以允許HTTP存取：
    - 類型：HTTP
    - 資源來源：0.0.0.0/0

選擇 Apply Rule Changes (套用規則變更)。

現在，您可以建立快取子網路群組，並在 Amazon 中啟動快取叢集VPC。

- [建立子網路群組](#)
- [建立 Memcached 叢集 \(主控台\)](#).
- [建立 Valkey \(停用叢集模式\) 叢集 \(主控台\)](#).

## 連線至在 Amazon 中執行的快取 VPC

此範例示範如何在 Amazon 中啟動 Amazon EC2執行個體VPC。然後，您可以登入此執行個體，並存取在 Amazon 中執行的 ElastiCache 快取VPC。

### 連線至在 Amazon 中執行的快取 VPC ( 主控台 )

在此範例中，您可以在 Amazon 中建立 Amazon EC2執行個體VPC。您可以使用此 Amazon EC2執行個體連線到 Amazon 中執行的快取節點VPC。

#### Note

如需有關使用 Amazon 的資訊EC2，請參閱 [Amazon 文件 中的 Amazon EC2入門指南](#)。 [EC2](#)

VPC 使用 Amazon EC2主控台在 Amazon 中建立 Amazon EC2執行個體

1. 登入 AWS Management Console 並在 開啟 Amazon EC2主控台<https://console.aws.amazon.com/ec2/>。
2. 在主控台中，選擇 Launch Instance (啟動執行個體)，然後依照下列步驟執行：
3. 在選擇 Amazon Machine Image (AMI) 頁面上，選擇 64 位元 Amazon Linux AMI，然後選擇選擇。
4. 在 Choose an Instance Type (選擇執行個體類型) 頁面上選擇 3. Configure Instance (3. 設定執行個體)。
5. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面上，選取下列選項：
  - a. 在網路清單中，選擇您的 Amazon VPC。
  - b. 在 Subnet (子網路) 清單中，選擇您的公有子網路。

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1

Purchasing option ⓘ  Request Spot Instances

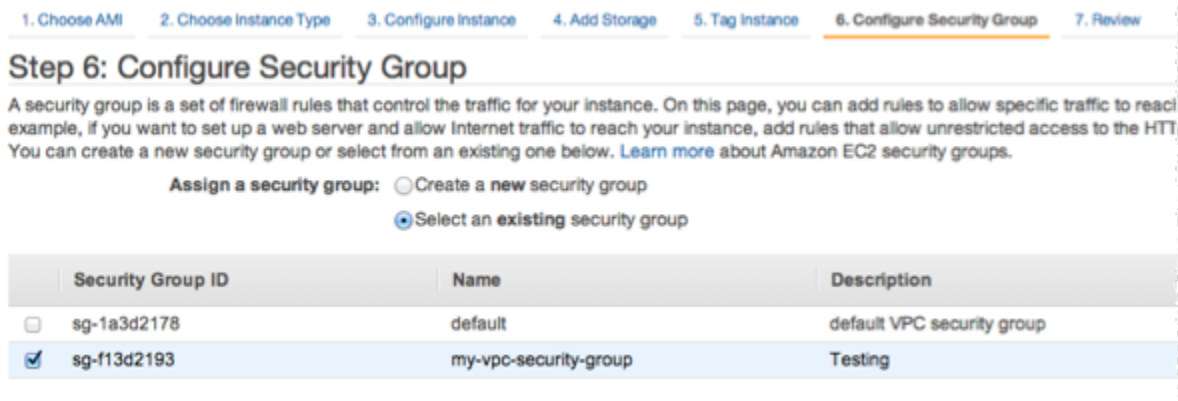
Network ⓘ vpc-d3a77cb6 (10.0.0.0/16) :

Subnet ⓘ subnet-58f5e63a(10.0.0.0/24) | sa-east-1a :   
250 IP Addresses available

Public IP ⓘ  Automatically assign a public IP address to your instances



- 當您滿意設定後，選擇 4. Add Storage (4. 新增儲存體)。
- 在 Add Storage (新增儲存體) 頁面上，選擇 5. Tag Instance (5. 標籤執行個體)。
  - 在標籤執行個體頁面上，輸入 Amazon EC2 執行個體的名稱，然後選擇 6. Configure Security Group (6. 設定安全群組)。
  - 在 Configure Security Group (設定安全群組) 頁面上，選擇 Select an existing security group (選取現有的安全群組)。如需安全群組的詳細資訊，請參閱 [Linux 執行個體的 Amazon EC2 安全群組](#)。



選擇 Amazon VPC 安全群組的名稱，然後選擇檢閱並啟動。

- 在 Review Instance and Launch (檢閱執行個體和啟動) 頁面上，選擇 Launch (啟動)。
- 在 Select an existing key pair or create a new key pair (選取現有的金鑰配對或建立新的金鑰配對) 視窗中，指定您要用於此執行個體的金鑰對。

#### Note

如需有關管理金鑰對的資訊，請參閱 [Amazon EC2 入門指南](#)。

- 當您準備好啟動 Amazon EC2 執行個體時，請選擇啟動。

您現在可以為您剛建立的 Amazon EC2 執行個體指派彈性 IP 地址。您需要使用此 IP 地址來連線至 Amazon EC2 執行個體。

指派彈性 IP 地址 (主控台)

- 在開啟 Amazon VPC 主控台 <https://console.aws.amazon.com/vpc/>。
- 在導覽清單中，選擇彈性 IPs。

3. 選擇 Allocate Elastic IP address (配置彈性 IP 地址)。
4. 在 Allocate Elastic IP address (配置彈性 IP 地址) 對話方塊中，接受預設的 Network Border Group (網路邊界群組)，然後選擇 Allocate (配置)。
5. 從清單中選擇您剛才配置的彈性 IP 地址，然後選擇 Associate Address (關聯地址)。
6. 在關聯地址對話方塊中的執行個體方塊中，選擇您啟動的 Amazon EC2 執行個體 ID。

在 Private IP address (私有 IP 地址) 方塊中，選取該方塊以取得私有 IP 地址，然後選擇 Associate (關聯)。

您現在可以使用您建立的彈性 IP 地址 SSH 來連線至 Amazon EC2 執行個體。

### 連線至您的 Amazon EC2 執行個體

- 開啟命令視窗。在命令提示中，發出以下命令，以您的金鑰對檔案名稱取代 `mykeypair.pem`，以您的彈性 IP 地址取代 `54.207.55.251`。

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

#### Important

尚未登出您的 Amazon EC2 執行個體。

您現在可以與 ElastiCache 叢集互動。在您可以這樣做之前 (如果您還沒有這樣做)，您必須安裝 telnet 公用程式。

### 安裝 Telnet 並與您的快取叢集互動 (AWS CLI)

- 開啟命令視窗。在命令提示中，發出下列命令。在確認提示中，輸入 `y`。

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...
```

```
Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm | 63 kB 00:00

...(output omitted)...

Complete!
```

您現在可以VPC使用 Memcached 或 Redis 連線到。

### VPC 使用 Memcached 連線至

1. 前往的 ElastiCache 主控台，<https://console.aws.amazon.com/elasticache/>並取得快取叢集中其中一個節點的端點。如需詳細資訊，請參閱[尋找連線端點](#)。
2. 使用 Telnet 透過連接埠 11211 連線到快取節點端點。以您的快取節點主機名稱取代以下顯示的主機名稱。

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

您現在已連接到快取引擎，並可以發出命令。在此範例中，您新增一個資料項目到快取，然後立即取得它。最後，您將中斷與快取節點的連線。

若要儲存金鑰和值，請輸入下列兩行：

```
add mykey 0 3600 28
This is the value for mykey
```

快取引擎會以下列方式回應：

```
OK
```

若要擷取 mykey 的值，請輸入下列命令：

```
get mykey
```

快取引擎會以下列方式回應：

```
VALUE mykey 0 28
This is the value for my key
END
```

若要中斷與快取引擎的連線，請輸入下列命令：

```
quit
```

## VPC 使用 Redis 連線至

1. 前往的 ElastiCache 主控台，<https://console.aws.amazon.com/elasticache/>並取得快取叢集中其中一個節點的端點。如需詳細資訊，請參閱[尋找 Redis 的連線端點](#)。
2. 使用 Telnet 透過連接埠 6379 連線到快取節點端點。以您的快取節點主機名稱取代以下顯示的主機名稱。

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

您現在已連接到快取引擎，並可以發出命令。在此範例中，您新增一個資料項目到快取，然後立即取得它。最後，您將中斷與快取節點的連線。

若要儲存金鑰和值，請輸入下列兩行：

```
set mykey myvalue
```

快取引擎會以下列方式回應：

```
OK
```

若要擷取 mykey 的值，請輸入下列命令：

```
get mykey
```

若要中斷與快取引擎的連線，請輸入下列命令：

```
quit
```

3. 前往的 ElastiCache 主控台，<https://console.aws.amazon.com/elasticache/>並取得快取叢集中其中一個節點的端點。如需詳細資訊，[請尋找 Redis 的連線端點](#)OSS。
4. 使用 Telnet 透過連接埠 6379 連線到快取節點端點。以您的快取節點主機名稱取代以下顯示的主機名稱。

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

您現在已連接到快取引擎，並可以發出命令。在此範例中，您新增一個資料項目到快取，然後立即取得它。最後，您將中斷與快取節點的連線。

若要存放索引鍵和值，請輸入下列內容：

```
set mykey myvalue
```

快取引擎會以下列方式回應：

```
OK
```

若要擷取 mykey 的值，請輸入下列命令：

```
get mykey
```

快取引擎會以下列方式回應：

```
get mykey
myvalue
```

若要中斷與快取引擎的連線，請輸入下列命令：

```
quit
```

### Important

為了避免您的帳戶產生額外費用 AWS，請務必在嘗試這些範例後刪除您不再需要的任何 AWS 資源。

## ElastiCache API 和介面VPC端點 ( AWS PrivateLink )

您可以建立介面端點，在 VPC 和 Amazon ElastiCache API 端點之間建立私有連線。VPC 介面端點由提供支援[AWS PrivateLink](#)。AWS PrivateLink 可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線的情況下，私下存取 Amazon ElastiCache API 操作。

您中的執行個體 VPC 不需要公有 IP 地址即可與 Amazon ElastiCache API 端點通訊。您的執行個體也不需要公有 IP 地址，即可使用任何可用的 ElastiCache API 操作。您與 Amazon VPC 之間的流量 ElastiCache 不會離開 Amazon 網路。每個介面端點都由子網路中的一個或多個彈性網路介面表示。如需彈性網路介面的詳細資訊，請參閱 Amazon EC2 使用者指南中的[彈性網路介面](#)。

- 如需 VPC 端點的詳細資訊，請參閱 Amazon VPC 使用者指南中的[介面 VPC 端點 \( AWS PrivateLink \)](#)。
- 如需操作的詳細資訊 ElastiCache API，請參閱[ElastiCache API 操作](#)。

建立介面 VPC 端點之後，如果您為端點啟用私有 DNS 主機名稱，則預設 ElastiCache 端點 ( `https://elasticache.Region.amazonaws.com` ) 會解析至您的 VPC 端點。如果您未啟用私有 DNS 主機名稱，Amazon VPC 會提供 DNS 端點名稱，您可以使用下列格式：

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[介面 VPC 端點 \( AWS PrivateLink \)](#)。ElastiCache 支援對內的所有[API 動作](#)進行呼叫 VPC。

### Note

只能為其中的一個 VPC 端點啟用私有 DNS 主機名稱 VPC。如果您想要建立其他 VPC 端點，則應停用私有 DNS 主機名稱。

## VPC 端點的考量事項

設定 Amazon ElastiCache API 端點的介面 VPC 端點之前，請務必檢閱 Amazon VPC 使用者指南中的[介面端點屬性和限制](#)。您可以使用 ElastiCache API VPC 來管理 Amazon ElastiCache 資源的所有相關操作 AWS PrivateLink。

VPC 端點支援端點 ElastiCache API 政策。根據預設，允許透過端點完整存取 ElastiCache API 操作。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[使用 VPC 端點控制對服務的存取](#)。

## 為 建立介面VPC端點 ElastiCache API

您可以使用 Amazon ElastiCache API VPC主控台或 建立 Amazon 的VPC端點 AWS CLI。如需詳細資訊，請參閱 Amazon VPC使用者指南 中的[建立介面端點](#)。

建立介面VPC端點之後，您可以為端點啟用私有DNS主機名稱。執行此操作時，預設 Amazon ElastiCache 端點（<https://elasticache.Region.amazonaws.com>）會解析至您的VPC端點。對於中國（北京）和中國（寧夏）AWS 區域，您可以使用 [elasticache.cn-north-1.amazonaws.com.cn](https://elasticache.cn-north-1.amazonaws.com.cn) 對北京和[elasticache.cn-northwest-1.amazonaws.com.cn](https://elasticache.cn-northwest-1.amazonaws.com.cn)寧夏的VPC端點提出API請求。如需詳細資訊，請參閱 Amazon VPC使用者指南 中的[透過介面端點存取服務](#)。

## 為 Amazon 建立VPC端點政策 ElastiCache API

您可以將端點政策連接至控制 存取的VPC端點 ElastiCache API。此政策會指定以下項目：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱 Amazon VPC使用者指南 中的[使用VPC端點控制對 服務的存取](#)。

Example VPC 使用 Valkey 或 Redis 執行動作的 ElastiCache API端點政策 OSS

以下是 端點政策的範例 ElastiCache API。連接至端點時，此政策會授予所有資源上所有主體的所列 ElastiCache API動作的存取權。

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster",
 "elasticache:CreateSnapshot"
],
 "Resource": "*"
 }]
}
```

## Example VPC ElastiCache ( Memcached ) API動作的端點政策

以下是 端點政策的範例 ElastiCache API。連接至端點時，此政策會授予所有資源上所有主體的所列 ElastiCache API動作的存取權。

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster"
],
 "Resource": "*"
 }]
}
```

## Example VPC 拒絕來自指定 AWS 帳戶的所有存取的端點政策

下列VPC端點政策拒絕 AWS 帳戶 **123456789012** 使用端點對資源的所有存取。此政策允許來自其他帳戶的所有動作。

```
{
 "Statement": [{
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*",
 "Principal": "*"
 },
 {
 "Action": "*",
 "Effect": "Deny",
 "Resource": "*",
 "Principal": {
 "AWS": [
 "123456789012"
]
 }
 }
]
```



## 子網路和子網路群組

子網路群組是子網路（通常是私有的）集合，您可以為在 Amazon Virtual Private Cloud（VPC）環境中執行的自我設計叢集指定。

如果您在 Amazon 中建立自行設計的叢集 VPC，則必須使用子網路群組。ElastiCache 使用該子網路群組來選擇該子網路內的子網路和 IP 地址，以與您的節點建立關聯。

ElastiCache 提供預設 IPv4 子網路群組，或者您可以選擇建立新的子網路群組。對於 IPv6，您需要使用 IPv6 CIDR 區塊建立子網路群組。如果您選擇雙堆疊，則您必須選取探索 IP 類型，可以是 IPv6 或 IPv4。

ElastiCache Serverless 不會使用子網路群組資源，而是直接在建立期間取得子網路清單。

本節說明如何建立和利用子網路和子網路群組來管理對 ElastiCache 資源的存取。

如需 Amazon VPC 環境中子網路群組用量的詳細資訊，請參閱 [存取 ElastiCache 叢集或複寫群組](#)。

### 主題

- [建立子網路群組](#)
- [將子網路群組指派至快取](#)
- [修改子網路群組](#)
- [刪除子網路群組](#)

## 建立子網路群組

快取子網路群組是您可能想要為 中的快取指定的子網路集合VPC。在 中啟動快取時VPC，您需要選取快取子網路群組。然後使用 ElastiCache 該快取子網路群組，將該子網路內的 IP 地址指派給快取中的每個快取節點。

建立新子網路群組時，請記下可用 IP 地址的數量。如果子網路擁有的可用 IP 地址很少，您還可以新增至叢集的節點數量可能會受到限制。若要解決此問題，您可以對子網路群組指定一或多個子網路，使得您在叢集的可用區域中有足夠數量的 IP 地址。在那之後，您便可以將更多節點新增至您的叢集。

如果您選擇 IPV4作為網路類型，預設子網路群組將可供使用，或者您可以選擇建立新的子網路群組。ElastiCache 使用該子網路群組來選擇該子網路內的子網路和 IP 地址，以與您的節點建立關聯。如果您選擇雙堆疊或 IPV6，系統會引導您建立雙堆疊或IPV6子網路。如需網路類型的詳細資訊，請參閱[網路類型](#)。如需詳細資訊，請參閱[在中建立子網路VPC](#)。

下列程序說明如何建立名為 mysubnetgroup (主控台) AWS CLI、和 的子網路群組ElastiCache API。

### 建立子網路群組 (主控台)

下列程序顯示如何建立子網路群組 (主控台)。

### 建立子網路群組 (主控台)

1. 登入 AWS 管理主控台，然後在 開啟 ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 在導覽清單中，選擇子網路群組。
3. 選擇 Create subnet group (建立子網路群組)。
4. 在建立子網路群組精靈中，執行下列操作。依您需要完成所有設定後，選擇建立。
  - a. 在 Name (名稱) 方塊中，輸入子網路群組的名稱。
  - b. 在 Description (描述) 方塊中，輸入子網路群組的描述。
  - c. 在 VPC ID 方塊中，選擇您的 Amazon VPC。
  - d. 所有子網路都是預設選擇的狀態。在選取子網路面板中，按一下管理，然後選取私有子網路 IDs的可用區域或[本機區域](#) 和 ，然後選擇選擇。
5. 在出現的確認訊息中，選擇 Close (關閉)。

您的新子網路群組會出現在 ElastiCache 主控台的子網路群組清單中。您可以在視窗底部選擇要查看詳細資訊的子網路群組，例如與此群組相關聯的所有子網路。

### 建立子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `create-cache-subnet-group` 來建立子網路群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup \
 --cache-subnet-group-description "Testing" \
 --subnet-ids subnet-53df9c3a
```

針對 Windows：

```
aws elasticache create-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup ^
 --cache-subnet-group-description "Testing" ^
 --subnet-ids subnet-53df9c3a
```

此命令應該產生類似下列的輸出：

```
{
 "CacheSubnetGroup": {
 "VpcId": "vpc-37c3cd17",
 "CacheSubnetGroupDescription": "Testing",
 "Subnets": [
 {
 "SubnetIdentifier": "subnet-53df9c3a",
 "SubnetAvailabilityZone": {
 "Name": "us-west-2a"
 }
 }
],
 "CacheSubnetGroupName": "mysubnetgroup"
 }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [create-cache-subnet-group](#).

## 將子網路群組指派至快取

建立子網路群組後，您可以在 Amazon 中啟動快取VPC。如需更多資訊，請參閱下列內容。

- Memcached 叢集 - 若要啟動 Memcached 叢集，請參閱「[建立 Memcached 叢集 \(主控台\)](#)」。在步驟 7.a ( 進階 Memcached Settings ) 中，選擇VPC子網路群組。
- 獨立 Valkey 或 Redis OSS叢集 – 若要啟動單節點 Valkey 或 Redis OSS叢集，請參閱 [建立 Valkey \(停用叢集模式\) 叢集 \(主控台\)](#)。在步驟 7.a ( 進階 Redis OSS設定 ) 中，選擇VPC子網路群組。
- Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組 – 若要在 中啟動 Valkey 或 Redis OSS ( 停用叢集模式 ) 複寫群組VPC，請參閱 [從頭開始建立 Valkey 或 Redis OSS \( 停用叢集模式 \) 複寫群組](#)。在步驟 7.b ( 進階 Redis OSS設定 ) 中，選擇VPC子網路群組。
- Valkey 或 Redis OSS ( 啟用叢集模式 ) 複寫群組 – [建立 Valkey 或 Redis OSS \( 啟用叢集模式 \) 叢集 \(主控台\)](#)。在步驟 6.i ( 進階 Redis OSS設定 ) 中，選擇VPC子網路群組。

## 修改子網路群組

您可以修改子網路群組的描述，或修改與子網路群組IDs相關聯的子網路清單。如果快取目前正在使用子網路，則無法從該子網路群組刪除該子網路 ID。

下列程序顯示如何修改子網路群組。

### 修改子網路群組 (主控台)

#### 修改子網路群組

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。
3. 在子網路群組清單中，選擇您要修改的子網路群組的選項按鈕，然後選擇修改。
4. 在選取的子網路面板中，選擇管理。
5. 對選取的子網路進行任何變更，然後按一下選擇。
6. 按一下儲存變更，儲存您的變更。

### 修改子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `modify-cache-subnet-group` 來修改子網路群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup \
 --cache-subnet-group-description "New description" \
 --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

針對 Windows：

```
aws elasticache modify-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup ^
 --cache-subnet-group-description "New description" ^
 --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

此命令應該產生類似下列的輸出：

```
{
 "CacheSubnetGroup": {
 "VpcId": "vpc-73cd3c17",
 "CacheSubnetGroupDescription": "New description",
 "Subnets": [
 {
 "SubnetIdentifier": "subnet-42dcf93a",
 "SubnetAvailabilityZone": {
 "Name": "us-west-2a"
 }
 },
 {
 "SubnetIdentifier": "subnet-48fc12a9",
 "SubnetAvailabilityZone": {
 "Name": "us-west-2a"
 }
 }
],
 "CacheSubnetGroupName": "mysubnetgroup"
 }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [modify-cache-subnet-group](#).

## 刪除子網路群組

如果您決定不再需要使用您的子網路群組，您可以將它刪除。如果有某個快取正在使用子網路群組，則無法將它刪除。

下列程序顯示如何刪除子網路群組。

### 刪除子網路群組 (主控台)

#### 刪除子網路群組

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。
3. 在子網路群組的清單中，選擇您要刪除的子網路群組，然後選擇 Delete (刪除)。
4. 當系統要求您確認此操作時，請在文字輸入欄位中輸入子網路群組的名稱，然後選擇刪除。

### 刪除子網路群組 (AWS CLI)

使用 AWS CLI，delete-cache-subnet-group 使用下列參數呼叫 命令：

- `--cache-subnet-group-name mysubnetgroup`

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup
```

針對 Windows：

```
aws elasticache delete-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CLI 主題 [delete-cache-subnet-group](#)。

## Amazon 的身分和存取管理 ElastiCache

AWS Identity and Access Management ( IAM ) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員會控制誰可以驗證 ( 登入 ) 和授權 ( 具有許可 ) 使用 ElastiCache 資源。IAM 是 AWS 服務 您可以免費使用的 。

## 主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [Amazon 如何使用 ElastiCache IAM](#)
- [Amazon 的身分型政策範例 ElastiCache](#)
- [對 Amazon ElastiCache 身分和存取進行疑難排解](#)
- [存取控制](#)
- [管理 ElastiCache 資源的存取許可概觀](#)

## 物件

使用 AWS Identity and Access Management ( IAM ) 的方式會有所不同，具體取決於您在 中執行的工作 ElastiCache。

**服務使用者** – 如果您使用 ElastiCache 服務來執行您的任務，則管理員會為您提供所需的憑證和許可。當您使用更多 ElastiCache 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 中的功能 ElastiCache，請參閱 [對 Amazon ElastiCache 身分和存取進行疑難排解](#)。

**服務管理員** – 如果您負責公司 ElastiCache 的資源，您可能擁有的完整存取權 ElastiCache。您的任務是判斷您的服務使用者應該存取哪些 ElastiCache 功能和資源。然後，您必須向IAM管理員提交請求，以變更服務使用者的許可。請檢閱此頁面上的資訊，以了解的基本概念IAM。若要進一步了解貴公司如何IAM搭配 使用 ElastiCache，請參閱 [Amazon 如何使用 ElastiCache IAM](#)。

**IAM 管理員** – 如果您是IAM管理員，您可能想要了解撰寫政策以管理 存取的詳細資訊 ElastiCache。若要檢視您可以在 中使用的以 ElastiCache 身分為基礎的政策範例IAM，請參閱 [Amazon 的身分型政策範例 ElastiCache](#)。

## 使用身分驗證

驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM使用者身分或擔任 IAM角色來驗證 ( 登入 AWS )。



您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分登入。AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 憑證，都是聯合身分的範例。當您以聯合身分登入時，您的管理員先前會使用 IAM 角色設定身分聯合。當您 AWS 使用聯合來存取時，您會間接擔任角色。

您可以登入 AWS Management Console 或 AWS 存取入口網站，視您的是的使用者類型而定。如需登入的詳細資訊 AWS，請參閱 [使用者指南](#) 中的 [如何登入 AWS 帳戶](#) 您的。AWS 登入

如果您以 AWS 程式設計方式存取，AWS 會提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南 中的 [簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素身分驗證 (MFA) 來提高帳戶的安全性。若要進一步了解，請參閱 AWS IAM Identity Center 使用者指南中的 [多重要素驗證](#)，以及 IAM 使用者指南 [中的使用多重要素驗證 \(MFA\) AWS](#)。

## AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可以完全存取帳戶中的所有 AWS 服務和資源。此身分稱為 AWS 帳戶 根使用者，透過您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需需要您以根使用者身分登入的任務完整清單，請參閱 IAM 使用者指南 中的 [需要根使用者憑證的任務](#)。

## 聯合身分

作為最佳實務，會要求人類使用者，包括需要管理員存取權的使用者，使用 AWS 服務 臨時憑證來與身分提供者使用聯合來存取。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、AWS Directory Service、身分中心目錄，或使用透過身分來源提供的 AWS 服務 憑證存取的任何使用者。當聯合身分存取時 AWS 帳戶，它們會擔任角色，而角色會提供臨時憑證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，或者您可以連線並同步到您身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 AWS IAM Identity Center 使用者指南 中的 [什麼是 IAM Identity Center ?](#)。

## IAM 使用者和群組

[IAM 使用者](#)是 中具有單一個人或應用程式特定許可 AWS 帳戶 的身分。在可能的情況下，我們建議您依賴臨時憑證，而不是建立具有密碼和存取金鑰等長期憑證IAM的使用者。不過，如果您有特定的使用案例需要IAM使用者長期憑證，建議您輪換存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南 中的[定期輪換需要長期憑證的使用案例存取金鑰](#)。

[IAM 群組](#)是指定IAM使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一名為 的群組IAMAdmins，並授予該群組管理IAM資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。若要進一步了解，請參閱 IAM 使用者指南 中的[何時建立IAM使用者（而非角色）](#)。

## IAM 角色

[IAM 角色](#)是 中具有特定許可 AWS 帳戶 的身分。它類似於IAM使用者，但與特定人員無關。您可以透過 AWS Management Console 切換IAM角色 暫時在 中擔任角色。 [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_use\\_switch-role-console.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html)您可以呼叫 AWS CLI 或 AWS API 操作，或使用自訂 來擔任角色URL。如需使用角色方法的詳細資訊，請參閱 IAM 使用者指南 中的[擔任角色的方法](#)。

IAM 具有臨時憑證的角色在下列情況下很有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需聯合角色的相關資訊，請參閱 IAM 使用者指南 中的[為第三方身分提供者建立角色](#)。如果您使用 IAM Identity Center，您可以設定許可集。若要控制身分在身分驗證後可以存取的內容，IAM Identity Center 會將許可集與 中的角色相關聯IAM。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的[許可集](#)。
- 臨時IAM使用者許可 – IAM使用者或角色可以擔任IAM角色，暫時接受特定任務的不同許可。
- 跨帳戶存取 – 您可以使用 IAM角色，允許不同帳戶中的某人（受信任的委託人）存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。不過，使用某些 AWS 服務，您可以將政策直接連接至資源（而不是使用角色作為代理）。若要了解跨帳戶存取的角色和資源型政策之間的差異，請參閱 IAM 使用者指南 [中的跨帳戶資源存取IAM](#)。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在 服務中撥打電話時，該服務通常會在 Amazon 中執行應用程式EC2或在 Amazon S3 中儲存物件。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。

- 轉送存取工作階段 ( FAS ) – 當您使用IAM使用者或角色在 中執行動作時 AWS ，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫的委託人許可 AWS 服務，並結合 AWS 服務 請求向下游服務提出請求的。FAS 只有在服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會發出請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出FAS請求的政策詳細資訊，請參閱[轉送存取工作階段](#)。
- 服務角色 – 服務角色是服務代表您執行動作時擔任[IAM的角色](#)。IAM 管理員可以從 內部建立、修改和刪除服務角色IAM。如需詳細資訊，請參閱 使用者指南 中的[建立角色以將許可委派給 AWS 服務](#)。IAM
- 服務連結角色 – 服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作角色。服務連結角色會顯示在您的 中 AWS 帳戶 ，並由 服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon 上執行的應用程式 EC2 – 您可以使用 IAM角色來管理在EC2執行個體上執行之應用程式的臨時憑證，以及提出 AWS CLI 或 AWS API請求。最好將存取金鑰存放在EC2執行個體中。若要将 AWS 角色指派給EC2執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體設定檔。執行個體設定檔包含 角色，並啟用執行個體上執行的程式EC2，以取得臨時憑證。如需詳細資訊，請參閱 IAM 使用者指南 中的[使用 IAM角色將許可授予在 Amazon EC2執行個體上執行的應用程式](#)。

若要了解如何使用IAM角色或IAM使用者，請參閱 IAM 使用者指南 中的[建立IAM角色（而非使用者）的時機](#)。

## 使用政策管理存取權

您可以透過建立政策並將其連接至 AWS 身分或資源 AWS 來控制 中的存取。政策是 AWS 其中的物件，當與身分或資源相關聯時，會定義其許可。當主體（使用者、根使用者或角色工作階段）發出請求時，會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策都以JSON文件 AWS 形式儲存在 中。如需JSON政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南 中的[JSON 政策概觀](#)。

管理員可以使用 AWS JSON政策來指定誰可以存取什麼。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對所需資源執行動作的許可，IAM管理員可以建立IAM政策。然後，管理員可以將IAM政策新增至角色，使用者可以擔任角色。

IAM 無論您用來執行操作的方法為何，政策都會定義動作的許可。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console、AWS CLI 或 AWS 取得角色資訊 API。

## 身分型政策

身分型政策是 JSON 許可政策文件，您可以附加到身分，例如 IAM 使用者、使用者群組或角色。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分型政策，請參閱 IAM 使用者指南 中的 [建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。若要了解如何在受管政策或內嵌政策之間進行選擇，請參閱 IAM 使用者指南 中的在 [受管政策與內嵌政策之間進行選擇](#)。

## 資源型政策

資源型政策是您連接至資源 JSON 的政策文件。資源型政策的範例包括 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主體可以包括帳戶、使用者、角色、聯合使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策 IAM 中使用來自的 AWS 受管政策。

## 存取控制清單 (ACLs)

存取控制清單 (ACLs) 控制哪些主體 (帳戶成員、使用者或角色) 具有存取資源的許可。ACLs 類似於資源型政策，雖然它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF 和 Amazon VPC 是支援的服務範例 ACLs。若要進一步了解 ACLs，請參閱 Amazon Simple Storage Service 開發人員指南 中的 [存取控制清單 \(ACL\) 概觀](#)。

## 其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可界限是一項進階功能，您可以在其中設定身分型政策可授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南 中的 [IAM 實體許可界限](#)。

- 服務控制政策 ( SCPs ) – SCPs是在 中指定組織或組織單位 ( OU ) 最大許可JSON的政策 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 之多個的服務。如果您啟用組織中的所有功能，則可以將服務控制政策 ( SCPs ) 套用至任何或所有帳戶。SCP 限制成員帳戶中實體的許可，包括每個 AWS 帳戶根使用者。如需 Organizations 和 的詳細資訊 SCPs，請參閱 AWS Organizations 使用者指南 中的[服務控制政策](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南 中的[工作階段政策](#)。

## 多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱 IAM 使用者指南 中的[政策評估邏輯](#)。

## Amazon 如何使用 ElastiCache IAM

在您使用 IAM 管理對 的存取之前 ElastiCache，請先了解哪些IAM功能可與 搭配使用 ElastiCache。

IAM 您可以與 Amazon 搭配使用的功能 ElastiCache

IAM 功能	ElastiCache 支援
<a href="#">身分型政策</a>	是
<a href="#">資源型政策</a>	否
<a href="#">政策動作</a>	是
<a href="#">政策資源</a>	是
<a href="#">政策條件索引鍵</a>	是
<a href="#">ACLs</a>	是
<a href="#">ABAC ( 政策中的標籤 )</a>	是
<a href="#">暫時性憑證</a>	是



IAM 功能	ElastiCache 支援
<a href="#">主體許可</a>	是
<a href="#">服務角色</a>	是
<a href="#">服務連結角色</a>	是

若要取得 ElastiCache 和其他 AWS 服務如何與大多數 IAM 功能搭配使用的高階檢視，請參閱 IAM 使用者指南 中的 [AWS 服務IAM](#)。

## 的身分型政策 ElastiCache

支援身分型政策：是

身分型政策是JSON許可政策文件，您可以附加到身分，例如IAM使用者、使用者群組或角色。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分型政策，請參閱 IAM 使用者指南 中的 [建立IAM政策](#)。

透過身分IAM型政策，您可以指定允許或拒絕的動作和資源，以及允許或拒絕動作的條件。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。若要了解您可以在JSON政策中使用的所有元素，請參閱 IAM 使用者指南 中的 [IAMJSON政策元素參考](#)。

### ElastiCache 的身分型政策範例

若要檢視 ElastiCache 身分型政策的範例，請參閱 [Amazon 的身分型政策範例 ElastiCache](#)。

## ElastiCache 內的資源型政策

支援資源型政策：否

資源型政策是您連接至資源JSON的政策文件。資源型政策的範例包括IAM角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主體可以包括帳戶、使用者、角色、聯合使用者或 AWS 服務。

若要啟用跨帳戶存取，您可以將另一個帳戶中的整個帳戶或IAM實體指定為資源型政策中的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同的時 AWS 帳戶，受信

任帳戶中的IAM管理員也必須授予主體實體（使用者或角色）存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南 [中的跨帳戶資源存取權IAM](#)。

## 的政策動作 ElastiCache

支援政策動作：是

管理員可以使用 AWS JSON政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action元素說明您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API操作相同的名稱。有一些例外狀況，例如沒有相符API操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 ElastiCache 動作清單，請參閱服務授權參考 中的 [Amazon 定義的動作 ElastiCache](#)。

中的政策動作在動作之前 ElastiCache 使用下列字首：

```
elasticache
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
 "elasticache:action1",
 "elasticache:action2"
]
```

您也可以使用萬用字元 (\*) 來指定多個動作。例如，若要指定開頭是 Describe 文字的所有動作，請包含以下動作：

```
"Action": "elasticache:Describe*"
```

若要檢視 ElastiCache 身分型政策的範例，請參閱 [Amazon 的身分型政策範例 ElastiCache](#)。

## 的政策資源 ElastiCache

支援政策資源：是

管理員可以使用 AWS JSON政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素會指定動作套用的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \( ARN \) 指定資源](#)。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (\*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 ElastiCache 資源類型及其的清單ARNs，請參閱服務授權參考中的 [Amazon 定義的資源 ElastiCache](#)。若要了解您可以使用哪些動作指定每個資源ARN的，請參閱 [Amazon 定義的動作 ElastiCache](#)。

若要檢視 ElastiCache 身分型政策的範例，請參閱 [Amazon 的身分型政策範例 ElastiCache](#)。

## ElastiCache 的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用 [條件運算子](#) 的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，只有在使用者使用IAM其使用者名稱加上標籤時，您才能授予IAM使用者存取資源的許可。如需詳細資訊，請參閱 IAM 使用者指南 中的 [IAM政策元素：變數和標籤](#)。



AWS 支援全域條件索引鍵和服務特定條件索引鍵。若要查看所有 AWS 全域條件索引鍵，請參閱 IAM 使用者指南 中的 [AWS 全域條件內容索引鍵](#)。

若要查看 ElastiCache 條件金鑰清單，請參閱服務授權參考 中的 [Amazon 條件金鑰 ElastiCache](#)。若要了解您可以使用條件金鑰的動作和資源，請參閱 [Amazon 定義的動作。ElastiCache](#)

若要檢視 ElastiCache 身分型政策的範例，請參閱 [Amazon 的身分型政策範例 ElastiCache](#)。

## 中的存取控制清單 ( ACLs ) ElastiCache

支援ACLs：是

存取控制清單 ( ACLs ) 控制哪些主體 ( 帳戶成員、使用者或角色 ) 具有存取 資源的許可。ACLs 類似於資源型政策，雖然它們不使用JSON政策文件格式。

## 屬性型存取控制 ( ABAC ) 搭配 ElastiCache

支援 ABAC ( 政策中的標籤 )：是

屬性型存取控制 ( ABAC ) 是一種根據屬性定義許可的授權策略。在 AWS 中，這些屬性稱為標籤。您可以將標籤連接至IAM實體 ( 使用者或角色 ) 和許多 AWS 資源。標記實體和資源是 ABAC 的第一步。然後，您可以設計ABAC政策，以便在主體的標籤與其嘗試存取的資源上的標籤相符時允許操作。

ABAC 有助於快速成長的環境，並有助於處理政策管理變得繁瑣的情況。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需的詳細資訊ABAC，請參閱 使用者指南 中的 [什麼是 ABAC ?](#)。IAM 若要檢視包含設定 之步驟的教學課程ABAC，請參閱 IAM 使用者指南 中的 [使用屬性型存取控制 \( ABAC \)](#)。

## 搭配 使用暫時憑證 ElastiCache

支援臨時憑證：是

當您使用臨時憑證登入時，有些 AWS 服務 無法使用。如需詳細資訊，包括 AWS 服務 使用哪些臨時憑證，請參閱 IAM 使用者指南 中的 [AWS 服務 與 搭配使用IAM](#)。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入，則表示您正在使用臨時憑證。例如，當您 AWS 使用公司的單一登入（SSO）連結存取時，該程序會自動建立臨時憑證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南 中的[切換到角色（主控台）](#)。

您可以使用 AWS CLI 或 手動建立臨時憑證 AWS API。然後，您可以使用這些臨時憑證來存取 AWS。AWS recommends，讓您動態產生臨時憑證，而不是使用長期存取金鑰。如需詳細資訊，請參閱 [中的臨時安全憑證IAM](#)。

## ElastiCache 的跨服務主體權限

支援轉送存取工作階段（FAS）：是

當您使用IAM使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫的委託人許可 AWS 服務，並結合請求向下游服務 AWS 服務 提出請求的。FAS 只有在服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會發出請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出FAS請求的政策詳細資訊，請參閱[轉送存取工作階段](#)。

## 的服務角色 ElastiCache

支援服務角色：是

服務角色是IAM服務擔任的角色，以代表您執行動作。IAM 管理員可以從 內部建立、修改和刪除服務角色IAM。如需詳細資訊，請參閱 使用者指南 中的[建立角色以將許可委派給 AWS 服務](#)。IAM

### Warning

變更服務角色的許可可能會中斷 ElastiCache 功能。只有在 ElastiCache 提供指引時，才能編輯服務角色。

## 的服務連結角色 ElastiCache

支援服務連結角色：是

服務連結角色是連結至的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 中 AWS 帳戶，並由 服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱[AWS 使用的服務IAM](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

## Amazon 的身分型政策範例 ElastiCache

根據預設，使用者和角色沒有建立或修改 ElastiCache 資源的許可。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或來執行任務 AWS API。若要授予使用者對所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者可以擔任角色。

若要了解如何使用這些範例政策文件來建立 IAM 身分型 JSON 政策，請參閱 IAM 使用者指南 中的 [建立 IAM 政策](#)。

如需定義的動作和資源類型的詳細資訊 ElastiCache，包括 ARNs 每種資源類型的格式，請參閱服務授權參考 中的 [Amazon 的動作、資源和條件金鑰 ElastiCache](#)。

### 主題

- [政策最佳實務](#)
- [使用 ElastiCache 主控台](#)
- [允許使用者檢視他們自己的許可](#)

## 政策最佳實務

身分型政策會決定某人是否可以建立、存取或刪除您帳戶中 ElastiCache 的資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用針對許多常見使用案例授予許可的 AWS 受管政策。它們可在您的 AWS 帳戶中使用。建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需詳細資訊，請參閱 IAM 使用者指南 中的 [AWS 受管政策](#) 或 [AWS 任務功能的受管政策](#)。
- 套用最低權限許可 – 當您使用 IAM 政策設定許可時，只會授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的詳細資訊，請參閱 IAM 使用者指南 [中的政策和許可 IAM](#)。
- 使用 IAM 政策中的條件來進一步限制存取：您可以將條件新增至政策，以限制對動作和資源的存取。例如，您可以撰寫政策條件來指定所有請求都必須使用 傳送 SSL。如果透過特定 使用服務動作，例如 AWS 服務，您也可以使用 條件來授予其存取權 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南 中的 [IAM JSON 政策元素：條件](#)。

- 使用 IAM Access Analyzer 驗證您的IAM政策，以確保安全且功能許可 – IAM Access Analyzer 會驗證新的和現有的政策，讓政策遵循IAM政策語言（JSON）和IAM最佳實務。IAM Access Analyzer 提供超過 100 個政策檢查和可操作的建議，協助您撰寫安全且實用的政策。如需詳細資訊，請參閱 IAM 使用者指南 中的[IAM存取分析器政策驗證](#)。
- 需要多因素身分驗證（MFA） – 如果您有需要IAM使用者或根使用者的案例 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫API操作MFA時要求，請將MFA條件新增至您的政策。如需詳細資訊，請參閱 IAM 使用者指南 中的[設定 MFA受保護的API存取](#)。

如需 中最佳實務的詳細資訊IAM，請參閱 IAM 使用者指南 [中的安全最佳實務IAM](#)。

## 使用 ElastiCache 主控台

若要存取 Amazon ElastiCache 主控台，您必須具有一組最低許可。這些許可必須允許您列出和檢視中 ElastiCache 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體（使用者或角色）而言，主控台就無法如預期運作。

對於僅對 AWS CLI 或 進行呼叫的使用者，您不需要允許最低主控台許可 AWS API。相反地，僅允許存取與其API嘗試執行的操作相符的動作。

為了確保使用者和角色仍然可以使用 ElastiCache 主控台，也請將 ElastiCacheConsoleAccess 或 ReadOnly AWS 受管政策連接至實體。如需詳細資訊，請參閱 IAM 使用者指南 中的[新增許可給使用者](#)。

## 允許使用者檢視他們自己的許可

此範例示範如何建立政策，允許使用者檢視連接至其IAM使用者身分的內嵌和受管政策。此政策包含在主控台上完成此動作或使用 AWS CLI 或 以程式設計方式完成此動作的許可 AWS API。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupsForUser",
 "iam:ListAttachedUserPolicies",
 "iam:ListUserPolicies",
 "iam:GetUser"
]
 }
]
}
```

```
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
 },
 {
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam:GetPolicyVersion",
 "iam:GetPolicy",
 "iam:ListAttachedGroupPolicies",
 "iam:ListGroupPolicies",
 "iam:ListPolicyVersions",
 "iam:ListPolicies",
 "iam:ListUsers"
],
 "Resource": "*"
 }
]
```

## 對 Amazon ElastiCache 身分和存取進行疑難排解

使用下列資訊來協助您診斷和修正使用 ElastiCache 時可能遇到的常見問題IAM。

### 主題

- [我無權在 中執行動作 ElastiCache](#)
- [我無權執行 iam : PassRole](#)
- [我想要允許 AWS 帳戶外的人員存取我的 ElastiCache 資源](#)

### 我無權在 中執行動作 ElastiCache

如果 AWS Management Console 告訴您未獲授權執行動作，則必須聯絡管理員尋求協助。您的管理員是提供您使用者名稱和密碼的人員。

下列範例錯誤會在 mateojackson 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 *elasticache:GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-widget* 動作存取 *elasticache:GetWidget* 資源。

## 我無權執行 iam : PassRole

如果您收到錯誤，表示您無權執行 `iam:PassRole` 動作，則必須更新您的政策，才能將角色傳遞給 ElastiCache。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 marymajor IAM 的使用者嘗試使用主控台在 中執行動作時，會發生下列錯誤範例 ElastiCache。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

## 我想要允許 AWS 帳戶外的人員存取我的 ElastiCache 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。對於支援資源型政策或存取控制清單 (ACLs) 的服務，您可以使用這些政策來授予人員對資源的存取權。

如需進一步了解，請參閱以下內容：

- 若要了解 是否 ElastiCache 支援這些功能，請參閱 [Amazon 如何使用 ElastiCache IAM](#)。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱 IAM 使用者指南 中的 [在您 AWS 帳戶 擁有的另一個資源中為 IAM 使用者提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱 使用者指南 中的 [提供存取權給第三方 AWS 帳戶 擁有](#)。IAM
- 若要了解如何透過身分聯合提供存取權，請參閱 IAM 使用者指南 中的 [為外部驗證的使用者提供存取權 \(身分聯合\)](#)。

- 若要了解跨帳戶存取使用角色和資源型政策之間的差異，請參閱 IAM 使用者指南 [中的跨帳戶資源存取IAM](#)。

## 存取控制

您可以擁有有效的憑證來驗證您的請求，但除非您具有許可，否則無法建立或存取 ElastiCache 資源。例如，您必須具有建立 ElastiCache 叢集的許可。

下列各節說明如何管理 的許可 ElastiCache。我們建議您先閱讀概觀。

- [管理 ElastiCache 資源的存取許可概觀](#)
- [針對 Amazon 使用身分型政策（IAM 政策）ElastiCache](#)



## 管理 ElastiCache 資源的存取許可概觀

每個 AWS 資源都由 AWS 帳戶擁有，而建立或存取資源的許可受許可政策的約束。帳戶管理員可以將許可政策連接到身分 IAM（即使用者、群組和角色）。此外，Amazon ElastiCache 也支援將許可政策連接至資源。

### Note

帳戶管理員 (或管理員使用者) 是具有管理員權限的使用者。如需詳細資訊，請參閱 IAM 使用者指南 中的 [IAM 最佳實務](#)。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 [建立權限合集](#) 說明進行操作。

- IAM 透過身分提供者在 中管理的使用者：

建立聯合身分的角色。請遵循 IAM 使用者指南 中 [為第三方身分提供者（聯合）建立角色](#) 的指示。

- IAM 使用者：

- 建立您的使用者可擔任的角色。請遵循 IAM 使用者指南 中 [為 IAM 使用者建立角色](#) 的指示。

- (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南 中 [將許可新增至使用者（主控台）](#) 的指示。

### 主題

- [Amazon ElastiCache 資源和操作](#)
- [了解資源所有權](#)
- [管理資源存取](#)
- [AWS Amazon 的 受管政策 ElastiCache](#)
- [針對 Amazon 使用身分型政策（IAM 政策）ElastiCache](#)
- [資源層級許可](#)
- [使用條件索引鍵](#)
- [使用 Amazon 的服務連結角色 ElastiCache](#)



- [ElastiCache API 許可：動作、資源和條件參考](#)

## Amazon ElastiCache 資源和操作

若要查看 ElastiCache 資源類型及其的清單ARNs，請參閱服務授權參考中的 [Amazon 定義的資源 ElastiCache](#)。若要了解您可以使用哪些動作指定每個資源ARN的，請參閱 [Amazon 定義的動作 ElastiCache](#)。

### 了解資源所有權

資源擁有者是建立資源 AWS 的帳戶。也就是說，資源擁有者是驗證建立資源之請求的主要實體 AWS 帳戶。主體實體可以是根帳戶、IAM使用者或IAM角色)。下列範例說明其如何運作：

- 假設您使用 AWS 帳戶的根帳戶憑證來建立快取叢集。在這種情況下，AWS 您的帳戶是資源的擁有者。在中 ElastiCache，資源是快取叢集。
- 假設您在 AWS 帳戶中建立IAM使用者，並將建立快取叢集的許可授予該使用者。在此例中，該使用者可以建立快取叢集。不過，使用者所屬 AWS 的帳戶擁有快取叢集資源。
- 假設您在 AWS 帳戶中建立具有建立快取叢集許可IAM的角色。在此例中，任何可以擔任該角色的人都能建立快取叢集。角色所屬 AWS 的帳戶擁有快取叢集資源。

### 管理資源存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

#### Note

本節討論在 Amazon 的內容IAM中使用 ElastiCache。它不會提供有關 IAM服務的詳細資訊。如需完整IAM文件，請參閱 使用者指南 中的 [什麼是 IAM?](#)。IAM 如需IAM政策語法和描述的相關資訊，請參閱 IAM 使用者指南 中的 [AWS IAM政策參考](#)。

連接到IAM身分的政策稱為身分型政策 ( IAM 政策 )。連接到資源的政策稱為資源型政策。

#### 主題

- [身分型政策 \( IAM 政策 \)](#)
- [指定政策元素：動作、效果、資源和主體](#)

- [在政策中指定條件](#)

## 身分型政策 ( IAM 政策 )

您可以將政策連接至IAM身分。例如，您可以執行下列動作：

- 將許可政策連接至帳戶中的使用者或群組 - 帳戶管理員能夠透過與特定使用者相關聯的許可政策來授予許可。在這種情況下，許可可讓該使用者建立 ElastiCache 資源，例如快取叢集、參數群組或安全群組。
- 將許可政策連接至角色 ( 授予跨帳戶許可 ) - 您可以將身分型許可政策連接至IAM角色，以授予跨帳戶許可。例如，帳戶 A 中的管理員可以建立角色，將跨帳戶許可授予另一個 AWS 帳戶 ( 例如帳戶 B ) 或服務，AWS 如下所示：
  1. 帳戶 A 管理員會建立IAM角色，並將許可政策附加至授予帳戶 A 中資源許可的角色。
  2. 帳戶 A 管理員將信任政策連接至該角色，識別帳戶 B 做為可擔任該角的委託人。
  3. 然後，帳戶 B 管理員可以將擔任角色的許可委派給帳戶 B 中的任何使用者。這樣做可讓帳戶 B 中的使用者在帳戶 A 中建立或存取資源。在某些情況下，您可能想要授予 AWS 服務許可以擔任角色。為了支援此方法，信任政策中的委託人也可以是 AWS 服務委託人。

如需使用 IAM 委派許可的詳細資訊，請參閱 IAM 使用者指南 中的[存取管理](#)。

以下是允許使用者執行 AWS 帳戶 DescribeCacheClusters 動作的範例政策。ElastiCache 也支援使用 ARNs API 動作的資源來識別特定資源。(此方法也稱為資源層級許可)。

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "DescribeCacheClusters",
 "Effect": "Allow",
 "Action": [
 "elasticache:DescribeCacheClusters"],
 "Resource": resource-arn
]
}
```

如需搭配 使用身分型政策的詳細資訊 ElastiCache，請參閱 [針對 Amazon 使用身分型政策 \( IAM 政策 \) ElastiCache](#)。如需使用者、群組、角色和許可的詳細資訊，請參閱 IAM 使用者指南 中的 [身分 \( 使用者、群組和角色 \)](#)。

## 指定政策元素：動作、效果、資源和主體

對於每個 Amazon ElastiCache 資源（請參閱 [Amazon ElastiCache 資源和操作](#)），服務會定義一組API操作（請參閱[動作](#)）。若要授予這些API操作的許可，請 ElastiCache 定義一組您可以在政策中指定的動作。例如，針對 ElastiCache叢集資源，會定義下列動作：CreateCacheCluster、DeleteCacheCluster和 DescribeCacheCluster。執行API操作可能需要多個動作的許可。

以下是最基本的政策元素：

- 資源 – 在政策中，您可以使用 Amazon Resource Name（ARN）來識別政策適用的資源。如需詳細資訊，請參閱[Amazon ElastiCache 資源和操作](#)。
- 動作：使用動作關鍵字識別您要允許或拒絕的資源操作。例如，根據指定的 Effect，elasticache:CreateCacheCluster許可允許或拒絕使用者執行 Amazon ElastiCache CreateCacheCluster操作的許可。
- 效果 - 您可以指定使用者要求特定動作時會有什麼效果；可為允許或拒絕。如果您未明確授予存取（允許）資源，則隱含地拒絕存取。您也可以明確拒絕存取資源。例如，您可以這樣做以確保使用者無法存取資源，即使不同的政策授與存取。
- 主體 – 在身分型政策（IAM 政策）中，附加政策的使用者是隱含主體。對於資源型政策，您可以指定想要收到許可的使用者、帳戶、服務或其他實體（僅適用於資源型政策）。

若要進一步了解IAM政策語法和描述，請參閱 IAM 使用者指南 中的[AWS IAM政策參考](#)。

如需顯示所有 Amazon ElastiCache API 動作的資料表，請參閱 [ElastiCache API 許可：動作、資源和條件參考](#)。

### 在政策中指定條件

當您授予許可時，您可以使用IAM政策語言來指定政策何時生效的條件。例如，建議只在特定日期之後套用政策。如需以政策語言指定條件的詳細資訊，請參閱 IAM 使用者指南 中的[條件](#)。

欲表示條件，您可以使用預先定義的條件金鑰。若要使用 ElastiCache特定條件金鑰，請參閱 [使用條件索引鍵](#)。您可以視需要使用 AWS全局條件金鑰。如需 AWS全局金鑰的完整清單，請參閱 IAM 使用者指南 中的[可用金鑰條件](#)。

## AWS Amazon 的 受管政策 ElastiCache

AWS 受管政策是由 AWS `AWSManaged` 政策建立和管理的獨立政策旨在為許多常見使用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住，AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的[客戶管理政策](#)，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受管政策中 AWS 定義的許可，則更新會影響政策連接的所有主體身分（使用者、群組和角色）。當新的 AWS 服務 啟動或新的API操作可用於現有服務時，AWS 很有可能會更新受 AWS 管政策。

如需詳細資訊，請參閱 IAM 使用者指南 中的 [AWS 受管政策](#)。

### AWS 受管政策：ElastiCacheServiceRolePolicy

您無法 ElastiCacheServiceRolePolicy 連接至IAM實體。此政策會連接至服務連結角色，ElastiCache 允許 代表您執行動作。

此政策允許 根據需要代表您 ElastiCache 管理 AWS 資源，以管理您的快取：

- `ec2` – 管理要連接至快取節點EC2的網路資源，包括VPC端點（適用於無伺服器快取）、彈性網路介面（ENIs）（適用於自行設計的叢集）和安全群組。
- `cloudwatch` – 將指標資料從服務發出至 CloudWatch。
- `outposts` – 允許在 AWS Outpost 上建立快取節點。

您可以在 IAM 主控台和 受管[ElastiCacheServiceRolePolicy](#)政策參考指南 [ElastiCacheServiceRolePolicy](#) 中找到政策。AWS

### AWS 受管政策：AmazonElastiCacheFullAccess

您可以將AmazonElastiCacheFullAccess政策連接至身分IAM。

此政策允許主體 ElastiCache 使用 AWS 管理主控台完整存取：

- `elasticache` — 存取所有 APIs。

- iam：建立服務運作所需的服務連結角色。
- ec2 — 描述建立快取（VPC、子網路、安全群組）所需的相依EC2資源，並允許建立VPC端點（無伺服器快取）。
- kms — 允許使用 CMKs 的客戶受管 encryption-at-rest。
- cloudwatch — 允許存取指標，以在主控台中顯示 ElastiCache 指標。
- application-autoscaling：允許存取以說明快取的自動擴展政策。
- logs：用於在主控台中填入日誌傳遞功能的日誌串流。
- firehose：用於在主控台中填入日誌傳遞功能的傳遞串流。
- s3：用於在主控台中填入快照還原功能的 S3 儲存貯體。
- outposts — 用於填入 AWS Outposts 以在主控台中建立快取。
- sns — 用於填入主控台中通知功能SNS的主題。

您可以在 IAM 主控台和 受管[AmazonElastiCacheFullAccess](#)政策參考指南 [AmazonElastiCacheFullAccess](#) 中找到政策。AWS

AWS 受管政策：AmazonElastiCacheReadOnlyAccess

您可以將AmazonElastiCacheReadOnlyAccess政策連接至身分IAM。

此政策允許主體 ElastiCache 使用 AWS 管理主控台進行唯讀存取：

- elasticache — 存取唯讀 Describe APIs。

您可以在 IAM 主控台和 受管[AmazonElastiCacheReadOnlyAccess](#)政策參考指南 [AmazonElastiCacheReadOnlyAccess](#) 中找到政策。AWS

ElastiCache 受 AWS 管政策的更新

檢視自此服務開始追蹤這些變更 ElastiCache 以來，受 AWS 管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 ElastiCache 文件歷史記錄頁面上的RSS摘要。

變更	描述	日期
<a href="#">AmazonElastiCacheFullAccess</a> – 更新現有政策	ElastiCache 新增允許管理無伺服器快取，以及透過主控台啟用所有服務功能的許可。	2023 年 11 月 27 日

變更	描述	日期
<a href="#">ElastiCacheServiceRolePolicy</a> – 更新現有政策	ElastiCache 新增允許管理無伺服器快取資源VPC端點的許可。	2023 年 11 月 27 日
ElastiCache 已開始追蹤變更	ElastiCache 已開始追蹤其 AWS 受管政策的變更。	2020 年 2 月 07 日

## 針對 Amazon 使用身分型政策 ( IAM 政策 ) ElastiCache

本主題提供身分型政策的範例，其中帳戶管理員可以將許可政策連接至身分 IAM ( 即使用者、群組和角色 )。

### Important

建議您先閱讀說明基本概念的主題，以及管理 Amazon ElastiCache 資源存取的選項。如需詳細資訊，請參閱[管理 ElastiCache 資源的存取許可概觀](#)。

本主題中的各節涵蓋下列內容：

- [AWS Amazon 的 受管政策 ElastiCache](#)
- [客戶受管政策範例](#)

以下顯示使用 Redis 時許可政策的範例OSS。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowClusterPermissions",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache",
 "elasticache:CreateCacheCluster",
 "elasticache:DescribeServerlessCaches",
 "elasticache:DescribeReplicationGroups",
 "elasticache:DescribeCacheClusters",
```

```

 "elasticache:ModifyServerlessCache",
 "elasticache:ModifyReplicationGroup",
 "elasticache:ModifyCacheCluster"
],
 "Resource": "*"
},
{
 "Sid": "AllowUserToPassRole",
 "Effect": "Allow",
 "Action": ["iam:PassRole"],
 "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
}
]
}

```

以下顯示使用 Memcached 時許可政策的範例。

```

{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "AllowClusterPermissions",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache",
 "elasticache:CreateCacheCluster",
 "elasticache:DescribeServerlessCaches",
 "elasticache:DescribeCacheClusters",
 "elasticache:ModifyServerlessCache",
 "elasticache:ModifyCacheCluster"
],
 "Resource": "*"
 },
 {
 "Sid": "AllowUserToPassRole",
 "Effect": "Allow",
 "Action": ["iam:PassRole"],
 "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
 }
]
}

```

此政策具有兩個陳述式：

- 第一個陳述式會授予 Amazon ElastiCache 動作 ( `elasticache:Create*`、`elasticache:Describe*`、`elasticache:Modify*` ) 的許可
- 第二個陳述式會針對Resource值結尾所指定的IAM角色名稱授予IAM動作 ( `iam:PassRole` ) 的許可。

此政策不指定 Principal 元素，因為您不會在以身分為基礎的政策中，指定取得許可的主體。當您將政策連接至使用者時，這名使用者即為隱含主體。當您將許可政策連接至IAM角色時，角色的信任政策中識別的主體會取得許可。

如需顯示所有 Amazon ElastiCache API 動作及其適用的資源的資料表，請參閱 [ElastiCache API 許可：動作、資源和條件參考](#)。

### 客戶受管政策範例

如果您未使用預設政策並選擇使用自訂受管政策，請確保下列兩件事的其中一項。您應具有呼叫 `iam:createServiceLinkedRole` 的許可 (如需詳細資訊，請參閱[範例 4：允許使用者呼叫 IAM CreateServiceLinkedRole API](#))。或者，您應該已建立 ElastiCache 服務連結角色。

當與使用 Amazon ElastiCache 主控台所需的最低許可結合時，本節中的範例政策會授予其他許可。這些範例也與 AWS SDKs和 相關 AWS CLI。

如需設定IAM使用者和群組的指示，請參閱 IAM 使用者指南 中的[建立您的第一個IAM使用者和管理員群組](#)。

#### Important

在生產中使用IAM政策之前，請務必徹底測試政策。當您使用 ElastiCache 主控台時，某些看似簡單的 ElastiCache 動作可能需要其他動作來支援這些動作。例如，會 `elasticache:CreateCacheCluster` 授予建立 ElastiCache 快取叢集的許可。不過，為了執行此操作，ElastiCache 主控台會使用許多 Describe和 List動作來填入主控台清單。

### 範例

- [範例 1：允許使用者對 ElastiCache 資源的唯讀存取](#)
- [範例 2：允許使用者執行常見的 ElastiCache 系統管理員任務](#)
- [範例 3：允許使用者存取所有 ElastiCache API動作](#)
- [範例 4：允許使用者呼叫 IAM CreateServiceLinkedRole API](#)
- [範例 5：允許使用者使用IAM身分驗證連線到無伺服器快取](#)



### 範例 1：允許使用者對 ElastiCache 資源的唯讀存取

下列政策會授予許可 ElastiCache 動作，允許使用者列出資源。您通常會將此類型的許可政策連接到管理員群組。

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECReadOnly",
 "Effect": "Allow",
 "Action": [
 "elasticache:Describe*",
 "elasticache:List*"
],
 "Resource": "*"
 }
]
```

### 範例 2：允許使用者執行常見的 ElastiCache 系統管理員任務

常見的系統管理員任務包括修改資源。系統管理員也可能想要取得事件的相關資訊 ElastiCache。下列政策會授予使用者許可，以針對這些常見的系統管理員任務執行 ElastiCache 動作。您通常會將此類型的許可政策連接到系統管理員群組。

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowMutations",
 "Effect": "Allow",
 "Action": [
 "elasticache:Modify*",
 "elasticache:Describe*",
 "elasticache:ResetCacheParameterGroup"
],
 "Resource": "*"
 }
]
```

### 範例 3：允許使用者存取所有 ElastiCache API 動作

下列政策允許使用者存取所有 ElastiCache 動作。建議您只將此類型的許可政策授予管理員使用者。

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAllowAll",
 "Effect": "Allow",
 "Action": [
 "elasticache:*"
],
 "Resource": "*"
 }
]
```

#### 範例 4：允許使用者呼叫 IAM CreateServiceLinkedRole API

下列政策可讓使用者呼叫 IAM CreateServiceLinkedRole API。建議您將此類型的許可政策授予叫用突變 ElastiCache 操作的使用者。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateSLRAllows",
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "iam:AWS ServiceName": "elasticache.amazonaws.com"
 }
 }
 }
]
}
```

#### 範例 5：允許使用者使用 IAM 身分驗證連線到無伺服器快取

下列政策允許任何使用者使用 2023-04-01 和 2023-06-30 之間的 IAM 身分驗證連線到任何無伺服器快取。

```
{
 "Version" : "2012-10-17",
 "Statement" :
 [
 {
 "Effect" : "Allow",
 "Action" : ["elasticache:Connect"],
 "Resource" : [
 "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
],
 "Condition": {
 "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
 "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
 }
 },
 {
 "Effect" : "Allow",
 "Action" : ["elasticache:Connect"],
 "Resource" : [
 "arn:aws:elasticache:us-east-1:123456789012:user:*"
]
 }
]
}
```

## 資源層級許可

您可以在IAM政策中指定資源來限制許可範圍。許多 ElastiCache API動作支援的資源類型會因動作的行為而有所不同。每個IAM政策陳述式都會授予對資源執行動作的許可。當動作沒有作用於具名資源，或是當您授予對所有資源執行動作的許可，政策中資源的值是萬用字元 (\*)。對於許多API動作，您可以透過指定資源的 Amazon Resource Name ( ARN ) 或符合多個資源的ARN模式來限制使用者可以修改的資源。若要依資源限制許可，請透過 指定資源ARN。

若要查看 ElastiCache 資源類型及其的清單ARNs，請參閱服務授權參考中的 [Amazon 定義的資源 ElastiCache](#)。若要了解您可以使用哪些動作指定每個資源ARN的，請參閱 [Amazon 定義的動作 ElastiCache](#)。

### 範例

- [範例 1：允許使用者完整存取特定 ElastiCache 資源類型](#)
- [範例 2：拒絕使用者存取無伺服器快取。](#)

## 範例 1：允許使用者完整存取特定 ElastiCache 資源類型

下列政策明確允許無伺服器快取類型的所有資源。

```
{
 "Sid": "Example1",
 "Effect": "Allow",
 "Action": "elasticache:*",
 "Resource": [
 "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
]
}
```

範例 2：拒絕使用者存取無伺服器快取。

下列範例會明確拒絕存取特定無伺服器快取。

```
{
 "Sid": "Example2",
 "Effect": "Deny",
 "Action": "elasticache:*",
 "Resource": [
 "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
]
}
```

## 使用條件索引鍵

您可以指定決定IAM政策如何生效的條件。在 `aws:elasticache` 中 ElastiCache，您可以使用JSON政策的 `Condition` 元素，將請求內容中的金鑰與您在政策中指定的金鑰值進行比較。如需詳細資訊，請參閱 [IAMJSON政策元素：條件](#)。

若要查看 ElastiCache 條件金鑰清單，請參閱服務授權參考中的 [Amazon 條件金鑰 ElastiCache](#)。

如需全域條件索引鍵的清單，請參閱 [AWS 全域條件內容索引鍵](#)。

### 指定條件：使用條件金鑰

若要實作精細控制，您會撰寫IAM許可政策，指定條件來控制特定請求上的一組個別參數。然後，您將政策套用到使用IAM主控台建立IAM的使用者、群組或角色。

若要套用條件，請將條件資訊新增至IAM政策陳述式。在下列範例中，您會指定建立的任何自行設計快取叢集都將屬於節點類型 `cache.r5.large` 的條件。

以下顯示使用 Valkey 或 Redis 時此許可政策的範例OSS。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.r5.large"
]
 }
 }
 }
]
}
```

以下顯示使用 Memcached 時此許可政策的範例。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```

 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.r5.large"
]
 }
 }
 }
]
}

```

如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。

如需使用政策條件運算子的詳細資訊，請參閱「[ElastiCache API 許可：動作、資源和條件參考](#)」。

**範例政策：使用條件進行精細參數控制**

本節顯示對先前列出的 ElastiCache 參數實作精細存取控制的範例政策。

1. `elasticache:MaximumDataStorage`：指定無伺服器快取的資料儲存上限。使用提供的條件，客戶就無法建立可儲存超過特定資料量的快取。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {

```

```

 "Sid": "AllowDependentResources",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],
 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
 "arn:aws:elasticache:*:*:snapshot:*",
 "arn:aws:elasticache:*:*:usergroup:*"
]
},
{
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],
 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscache:*"
],
 "Condition": {
 "NumericLessThanEquals": {
 "elasticache:MaximumDataStorage": "30"
 },
 "StringEquals": {
 "elasticache:DataStorageUnit": "GB"
 }
 }
}
]
}

```

2. `elasticache:MaximumECPUPerSecond` : 指定無伺服器快取ECPUs的每秒最大值。使用提供的條件，客戶無法建立每秒可執行超過特定數目的快取ECPUs。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowDependentResources",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],

```

```

 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
 "arn:aws:elasticache:*:*:snapshot:*",
 "arn:aws:elasticache:*:*:usergroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],
 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscache:*"
],
 "Condition": {
 "NumericLessThanEquals": {
 "elasticache:MaximumECPUPerSecond": "100000"
 }
 }
 }
]
}

```

3. `elasticache:CacheNodeType` : 指定使用者可以建立哪些 `NodeType` (s)。使用提供的條件，客戶可以指定節點類型的單一值或範圍值。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",

```



```

 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.t2.micro",
 "cache.t2.medium"
]
 }
 }
 }
}
]
}

```

4. `elasticache:CacheNodeType`：使用 Memcached，指定使用者可以建立哪些 Node Type (s)。使用提供的條件，客戶可以指定節點類型的單一值或範圍值。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [

```

```

 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.t2.micro",
 "cache.t2.medium"
]
 }
 }
}
]
}

```

##### 5. elasticache : NumNodeGroups : 建立少於 20 個節點群組的複寫群組。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "NumericLessThanEquals": {
 "elasticache:NumNodeGroups": "20"
 }
 }
 }
]
}

```

```
]
}
```

6. `elasticache : ReplicasPerNodeGroup` : 指定每個節點介於 5 到 10 之間的複本。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "NumericGreaterThanEquals": {
 "elasticache:ReplicasPerNodeGroup": "5"
 },
 "NumericLessThanEquals": {
 "elasticache:ReplicasPerNodeGroup": "10"
 }
 }
 }
]
}
```

7. `elasticache : EngineVersion` : 指定引擎 5.0.6 版的用量。

```
{
 "Version": "2012-10-17",
```

```

"Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:EngineVersion": "5.0.6"
 }
 }
 }
]
}

```

## 8. elasticache : EngineVersion : 指定 Memcached 引擎 1.6.6 版的使用

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",

```

```

 "arn:aws:elasticache:*:*:subnetgroup:*"
]
},
{
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:EngineVersion": "1.6.6"
 }
 }
}
]
}

```

9. elasticache : EngineType : 僅使用 Valkey 或 Redis OSS引擎指定。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 }
]
}

```

```

 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:EngineType": "redis"
 }
 }
 }
]
}

```

10elasticache : AtRestEncryptionEnabled : 指定複寫群組只會在啟用加密的情況下建立。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:AtRestEncryptionEnabled": "true"
 }
 }
 }
]
}

```

```
]
 }
}
```

## 11 elasticache : TransitEncryptionEnabled

- a. 將 `false` [CreateReplicationGroup](#) 動作 `elasticache:TransitEncryptionEnabled` 的條件索引鍵設定為 `false`，以指定複寫群組只能在 TLS 未使用時建立：

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:TransitEncryptionEnabled": "false"
 }
 }
 }
]
}
```

在 [CreateReplicationGroup](#) 動作的政策 `false` 中

將 `elasticache:TransitEncryptionEnabled` 條件索引鍵設定為 `false` 時，只有在 TLS 未使用（亦即，如果 `CreateReplicationGroup` 請求不包含參數 `TransitEncryptionEnabled` 設定為 `true` 或 `TransitEncryptionMode` 參數設定為 `required` 時，才會允許請求。

- b. 將 [CreateReplicationGroup](#) 動作true的 elasticache:TransitEncryptionEnabled condition 金鑰設定為 `elasticache:TransitEncryptionEnabled`，以指定複寫群組只能在使用TLS時建立：

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:TransitEncryptionEnabled": "true"
 }
 }
 }
]
}
```

在 [CreateReplicationGroup](#) 動作的政策true中

將elasticache:TransitEncryptionEnabled條件索引鍵設定為 `elasticache:TransitEncryptionEnabled` 時，只有在CreateReplicationGroup請求包含設定為 `elasticache:TransitEncryptionEnabled` 的TransitEncryptionEnabled參數true和設定為 `elasticache:TransitEncryptionMode` 的TransitEncryptionMode參數時，才會允許請求required。

- c. 將 elasticache:TransitEncryptionEnabled設定為 `elasticache:TransitEncryptionEnabled`，true讓 ModifyReplicationGroup動作指定複寫群組只能在使用TLS時修改：



```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:ModifyReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "BoolIfExists": {
 "elasticache:TransitEncryptionEnabled": "true"
 }
 }
 }
]
}
```

在 [ModifyReplicationGroup](#) 動作的政策true中

將elasticache:TransitEncryptionEnabled條件索引鍵設定為 true 時，只有在ModifyReplicationGroup請求包含設定為 true 的TransitEncryptionMode參數時，才會允許請求required。true 也可以選擇性地包含 true 設定為 true 的TransitEncryptionEnabled參數，但在此情況下不需要 true 來啟用 TLS。

12elasticache : AutomaticFailoverEnabled : 指定複寫群組只會在啟用自動容錯移轉的情況下建立。

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 }
],
}
```

```

 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:AutomaticFailoverEnabled": "true"
 }
 }
 }
]
}

```

13elasticache : MultiAZEnabled : 指定複寫群組無法在停用多可用區的情況下建立。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {

```

```

 "Bool": {
 "elasticache:MultiAZEnabled": "false"
 }
 }
}
]
}

```

14.elasticache : ClusterModeEnabled : 指定複寫群組只能在啟用叢集模式的情況下建立。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:ClusterModeEnabled": "true"
 }
 }
 }
]
}

```

15.elasticache : AuthTokenEnabled : 指定複寫群組只能在啟用AUTH字符的情況下建立。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "Bool": {
 "elasticache:AuthTokenEnabled": "true"
 }
 }
 }
]
}

```

16 `elasticache:SnapshotRetentionLimit` : 指定保留快照的天數 ( 或最小值/最大值 )。以下政策強制存放備份至少 30 天。

```

{
 "Version": "2012-10-17",
 "Statement": [

```

```

 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup",
 "elasticache:CreateServerlessCache"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*",
 "arn:aws:elasticache:*:*:serverlesscache:*"
],
 "Condition": {
 "NumericGreaterThanEquals": {
 "elasticache:SnapshotRetentionLimit": "30"
 }
 }
 }
]
}

```

17elasticache : KmsKeyId : 指定客戶受管 AWS KMS金鑰的使用。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowDependentResources",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],
 },
],
}

```

```

 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
 "arn:aws:elasticache:*:*:snapshot:*",
 "arn:aws:elasticache:*:*:usergroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateServerlessCache"
],
 "Resource": [
 "arn:aws:elasticache:*:*:serverlesscache:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:KmsKeyId": "my-key"
 }
 }
 }
]
}

```

18 `elasticache:CacheParameterGroupName`：指定非預設參數群組，其中包含叢集上組織的特定參數。您也可以為參數群組指定命名模式，或阻止刪除特定參數群組名稱。以下是限制只使用 "my-org-param-group" 的範例。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 }
],
}

```

```

 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheParameterGroupName": "my-org-param-group"
 }
 }
 }
]
}

```

19 `elasticache:CacheParameterGroupName`：使用 Memcached，指定具有叢集上組織特定參數的非預設參數群組。您也可以為參數群組指定命名模式，或阻止刪除特定參數群組名稱。以下是限制只使用 "my-org-param-group" 的範例。

```

{
 "Version": "2012-10-17",
 "Statement": [

 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },

 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 }
]
}

```

```

 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheParameterGroupName": "my-org-param-group"
 }
 }
 }
]
}

```

20elasticache : CreateCacheCluster : 如果請求標籤Project遺失或不等於 Dev、或 QA ，則會拒絕CreateCacheCluster動作Prod。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*",
 "arn:aws:elasticache:*:*:securitygroup:*",
 "arn:aws:elasticache:*:*:replicationgroup:*"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "Null": {
 "aws:RequestTag/Project": "true"
 }
 }
 }
]
}

```



```

 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:AddTagsToResource"
],
 "Resource": "arn:aws:elasticache:*:*:cluster:*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Project": [
 "Dev",
 "Prod",
 "QA"
]
 }
 }
 }
]
}

```

21 elasticache : CacheNodeType : 允許 CreateCacheCluster 使用 cacheNodeType cache.r5.large 或 cache.r6g.4xlarge 並標記 Project=XYZ。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 }
]
}

```

```

"Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
"Condition": {
 "StringEqualsIfExists": {
 "elasticache:CacheNodeType": [
 "cache.r5.large",
 "cache.r6g.4xlarge"
]
 },
 "StringEquals": {
 "aws:RequestTag/Project": "XYZ"
 }
}
]
}

```

22.elasticache : CacheNodeType : 允許CreateCacheCluster使用 cacheNodeType cache.r5.large 或 cache.r6g.4xlarge 並標記 Project=XYZ。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {

```

```
"StringEqualsIfExists": {
 "elasticache:CacheNodeType": [
 "cache.r5.large",
 "cache.r6g.4xlarge"
]
},
"StringEquals": {
 "aws:RequestTag/Project": "XYZ"
}
}
]
}
```

### Note

建立同時強制執行標籤和其他條件索引鍵的政策時，因為具有 `--tags` 參數的建立請求需要額外的 `elasticache:AddTagsToResource` 政策，`IfExists` 條件可能需要條件索引鍵元素。

## 使用 Amazon 的服務連結角色 ElastiCache

Amazon ElastiCache 使用 AWS Identity and Access Management ( IAM ) [服務連結角色](#)。服務連結角色是直接連結至 AWS 服務的唯一 IAM 角色類型，例如 Amazon ElastiCache。Amazon 會預先定義 Amazon ElastiCache 服務連結角色 ElastiCache。此角色包含服務需要的所有許可，以代您來呼叫其他的 AWS 服務。

服務連結角色可讓您 ElastiCache 更輕鬆地設定 Amazon，因為您不必手動新增必要的許可。這些角色已存在於 AWS 您的帳戶中，但會連結至 Amazon ElastiCache 使用案例，並具有預先定義的許可。只有 Amazon ElastiCache 可以擔任這些角色，只有這些角色可以使用預先定義的許可政策。您必須先刪除角色的相關資源，才能刪除角色。這可保護您的 Amazon ElastiCache 資源，因為您不會不小心移除存取資源的必要許可。

如需有關支援服務連結角色的其他服務的資訊，請參閱服務連結角色欄中 [AWS 使用的服務IAM](#)，並尋找具有 Yes 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

### 內容

- [Amazon 的服務連結角色許可 ElastiCache](#)

- [建立服務連結角色的許可](#)
- [建立服務連結角色 \( IAM \)](#)
  - [建立服務連結角色 \( IAM 主控台 \)](#)
  - [建立服務連結角色 \( IAM CLI \)](#)
  - [建立服務連結角色 \( IAM API \)](#)
- [編輯 Amazon 的服務連結角色描述 ElastiCache](#)
  - [編輯服務連結角色描述 \( IAM 主控台 \)](#)
  - [編輯服務連結角色描述 \( IAM CLI \)](#)
  - [編輯服務連結角色描述 \( IAM API \)](#)
- [刪除 Amazon 的服務連結角色 ElastiCache](#)
  - [清除服務連結角色](#)
  - [刪除服務連結角色 \( IAM 主控台 \)](#)
  - [刪除服務連結角色 \( IAM CLI \)](#)
  - [刪除服務連結角色 \( IAM API \)](#)

## Amazon 的服務連結角色許可 ElastiCache

### 建立服務連結角色的許可

允許IAM實體建立 AWS ServiceRoleForElastiCache 服務連結角色

將下列政策陳述式新增至該IAM實體的許可：

```
{
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole",
 "iam:PutRolePolicy"
],
 "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
 "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

允許IAM實體刪除 AWS ServiceRoleForElastiCache 服務連結角色

將下列政策陳述式新增至該IAM實體的許可：

```
{
 "Effect": "Allow",
 "Action": [
 "iam:DeleteServiceLinkedRole",
 "iam:GetServiceLinkedRoleDeletionStatus"
],
 "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
 "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

或者，您可以使用 AWS 受管政策來提供 Amazon 的完整存取權 ElastiCache。

### 建立服務連結角色 ( IAM )

您可以使用 IAM 主控台、CLI 或 建立服務連結角色 API。

#### 建立服務連結角色 ( IAM 主控台 )

您可以使用 IAM 主控台來建立服務連結角色。

#### 建立服務連結角色 (主控台)

1. 登入 AWS Management Console 並在 開啟 IAM 主控台 <https://console.aws.amazon.com/iam/>。
2. 在 IAM 主控台的導覽窗格中，選擇角色。然後選擇 Create new role (建立新角色)。
3. 在 Select type of trusted entity (選擇可信任的實體類型) 下，選擇 AWS Service (AWS 服務)。
4. 在 或選取要檢視其使用案例的服務下，選擇 ElastiCache。
5. 選擇下一步：許可。
6. 在 Policy name (政策名稱)，請注意 ElastiCacheServiceRolePolicy 是此角色的必要項目。選擇 Next: Add Tags (下一步：新增標籤)。
7. 請注意，服務連結的角色不支援標籤。選擇 Next: Review (下一步：檢閱)。
8. (選擇性) 針對 Role description (角色描述)，編輯新服務連結角色的描述。
9. 檢閱角色，然後選擇 Create role (建立角色)。

#### 建立服務連結角色 ( IAM CLI )

您可以從 使用 IAM 操作 AWS Command Line Interface 來建立服務連結角色。此角色可包含服務擔任該角色所需的信任政策與內嵌政策。

## 建立服務連結角色 ( CLI )

使用以下操作：

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

## 建立服務連結角色 ( IAM API )

您可以使用 IAM API 來建立服務連結角色。此角色可包含服務擔任該角色所需的信任政策與內嵌政策。

## 建立服務連結角色 ( API )

使用 [CreateServiceLinkedRole](#) API 呼叫。在請求中指定 `elasticache.amazonaws.com` 的服務名稱。

## 編輯 Amazon 的服務連結角色描述 ElastiCache

Amazon ElastiCache 不允許您編輯 AWS ServiceRoleForElastiCache 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。不過，您可以使用編輯角色的描述 IAM。

## 編輯服務連結角色描述 ( IAM 主控台 )

您可以使用 IAM 主控台編輯服務連結角色描述。

## 編輯服務連結角色的說明 (主控台)

1. 在 IAM 主控台的導覽窗格中，選擇角色。
2. 選擇要修改之角色的名稱。
3. 在 Role description (角色說明) 的最右邊，選擇 Edit (編輯)。
4. 在方塊中輸入新的描述，然後選擇 Save (儲存)。

## 編輯服務連結角色描述 ( IAM CLI )

您可以使用中的 IAM 操作 AWS Command Line Interface 來編輯服務連結角色描述。

## 若要變更服務連結角色的描述 ( CLI )

1. (選用) 若要檢視角色的目前描述，請將 AWS CLI 用於 IAM 操作 [get-role](#)。

## Example

```
$ aws iam get-role --role-name AWS ServiceRoleForElastiCache
```

使用角色名稱，而非 ARN，以參考具有 CLI 操作的角色。例如，如果角色具有下列 ARN：arn:aws:iam::123456789012:role/myrole，請將角色稱為 **myrole**。

- 若要更新服務連結角色的描述，請將 AWS CLI 用於 IAM 操作 [update-role-description](#)。

若為 Linux、macOS 或 Unix：

```
$ aws iam update-role-description \
 --role-name AWS ServiceRoleForElastiCache \
 --description "new description"
```

針對 Windows：

```
$ aws iam update-role-description ^
 --role-name AWS ServiceRoleForElastiCache ^
 --description "new description"
```

## 編輯服務連結角色描述 (IAM API)

您可以使用 IAM API 編輯服務連結角色描述。

若要變更服務連結角色的描述 (API)

- (選用) 若要檢視角色的目前描述，請使用 IAM API 操作 [GetRole](#)。

## Example

```
https://iam.amazonaws.com/
?Action=GetRole
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&AUTHPARAMS
```

- 若要更新角色的描述，請使用 IAM API 操作 [UpdateRoleDescription](#)。

## Example

```
https://iam.amazonaws.com/
?Action=UpdateRoleDescription
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&Description="New description"
```

## 刪除 Amazon 的服務連結角色 ElastiCache

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，務必清除您的服務連結角色，之後才能將其刪除。

Amazon ElastiCache 不會為您刪除服務連結角色。

## 清除服務連結角色

在您使用 IAM 刪除服務連結角色之前，請先確認角色沒有與其相關聯的資源（叢集或複寫群組）。

檢查服務連結角色在IAM主控台中是否具有作用中工作階段

1. 登入 AWS Management Console 並在 開啟IAM主控台<https://console.aws.amazon.com/iam/>。
2. 在IAM主控台的導覽窗格中，選擇角色。然後選擇 AWS ServiceRoleForElastiCache 角色的名稱（而非核取方塊）。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 分頁中，檢閱服務連結角色的近期活動。

## 刪除需要的 Amazon ElastiCache 資源 AWS ServiceRoleForElastiCache

- 若要刪除叢集，請參閱下列指示：
  - [使用 AWS Management Console](#)
  - [使用 AWS CLI 刪除 ElastiCache 叢集](#)
  - [使用 ElastiCache API](#)
- 若要刪除複寫群組，請參閱下列指示：
  - [刪除複寫群組 \(主控台\)](#)
  - [刪除複寫群組 \(AWS CLI\)](#)



- [刪除複寫群組 \(ElastiCache API\)](#)

## 刪除服務連結角色 (IAM 主控台)

您可以使用 IAM 主控台來刪除服務連結角色。

### 刪除服務連結角色 (主控台)

1. 登入 AWS Management Console 並在 開啟IAM主控台<https://console.aws.amazon.com/iam/>。
2. 在IAM主控台的導覽窗格中，選擇角色。然後，選擇您要刪除的角色名稱旁的核取方塊，而非名稱或資料列本身。
3. 在頁面頂端的 Role (角色) 動作中選擇 Delete (刪除) 角色。
4. 在確認對話方塊中，檢閱服務上次存取的資料，以顯示每個所選角色上次存取 AWS 服務的時間。這可協助您確認角色目前是否作用中。如果您想要繼續進行，請選擇 Yes, Delete (是，刪除) 來提交服務連結角色以進行刪除。
5. 觀看IAM主控台通知，以監控服務連結角色刪除的進度。由於IAM服務連結角色刪除是非同步的，因此在您提交角色以供刪除之後，刪除任務可能會成功或失敗。如果任務失敗，您可以從通知中選擇 View details (檢視詳細資訊) 或 View Resources (檢視資源)，以了解刪除失敗的原因。

## 刪除服務連結角色 (IAM CLI)

您可以從 使用 IAM 操作 AWS Command Line Interface 來刪除服務連結角色。

### 若要刪除服務連結角色 (CLI)

1. 如果您不知道想要刪除的服務連結角色名稱，請輸入以下命令。此命令會列出您帳戶中的角色及其 Amazon Resource Names (ARNs)。

```
$ aws iam get-role --role-name role-name
```

使用角色名稱，而非 ARN，以參考具有 CLI 操作的角色。例如，如果角色具有 ARN `arn:aws:iam::123456789012:role/myrole`，您會將角色稱為 **myrole**。

2. 因為無法刪除正在使用或具有相關聯資源的服務連結角色，所以您必須提交刪除要求。如果不符合這些條件，則可以拒絕該請求。您必須從回應中擷取 `deletion-task-id`，以檢查刪除任務的狀態。輸入下列內容，提交服務連結角色刪除請求。

```
$ aws iam delete-service-linked-role --role-name role-name
```

### 3. 輸入下列內容來檢查刪除任務的狀態。

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

刪除任務的狀態可以是 NOT\_STARTED、IN\_PROGRESS、SUCCEEDED 或 FAILED。如果刪除失敗，則呼叫會傳回失敗原因，以進行疑難排解。

#### 刪除服務連結角色 ( IAM API )

您可以使用 IAM API 刪除服務連結角色。

#### 刪除服務連結角色 ( API )

1. 若要提交服務連結卷的刪除請求，請呼叫 [DeleteServiceLinkedRole](#)。在請求中，指定角色名稱。

因為無法刪除正在使用或具有相關聯資源的服務連結角色，所以您必須提交刪除要求。如果不符合這些條件，則可以拒絕該請求。您必須從回應中擷取 DeletionTaskId，以檢查刪除任務的狀態。

2. 若要檢查刪除的狀態，請呼叫 [GetServiceLinkedRoleDeletionStatus](#)。在請求中，指定 DeletionTaskId。

刪除任務的狀態可以是 NOT\_STARTED、IN\_PROGRESS、SUCCEEDED 或 FAILED。如果刪除失敗，則呼叫會傳回失敗原因，以進行疑難排解。

## ElastiCache API 許可：動作、資源和條件參考

當您設定[存取控制](#)並寫入要附加至IAM政策（以身分為基礎或資源為基礎）的許可政策時，請使用下表做為參考。資料表列出每個 Amazon ElastiCache API 操作，以及您可以授予執行動作許可的對應動作。您需在政策的 Action 欄位中指定動作，然後在政策的 Resource 欄位中指定資源值。除非另有說明，否則資源為必要項目。某些欄位同時包含必要資源和選用資源。如果沒有資源 ARN，則政策中的資源是萬用字元（\*）。

您可以在 ElastiCache 政策中使用條件索引鍵來表達條件。若要查看 ElastiCache 特定條件索引鍵的清單，以及套用的動作和資源類型，請參閱[使用條件索引鍵](#)。如需 AWS 全金鑰的完整清單，請參閱 IAM 使用者指南 中的[AWS 全域條件內容金鑰](#)。

### Note

若要指定動作，請使用字元 `elasticache:` 首，後面加上 API 操作名稱（例如 `elasticache:DescribeCacheClusters`）。

若要查看 ElastiCache 動作清單，請參閱服務授權參考 中的 [Amazon 定義的動作 ElastiCache](#)。

## Amazon 的合規驗證 ElastiCache

第三方稽核人員會在多個合規計畫中評估 AWS 服務的安全性和 AWS 合規性，例如 SOC、PCI、Fed RAMP 和 HIPAA。

若要了解是否 AWS 服務在特定合規計畫的範圍內，請參閱[AWS 服務 依合規計畫範圍](#)然後選擇您感興趣的合規計畫。如需一般資訊，請參閱 [AWS Compliance Programs](#)。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 [中下載報告 AWS Artifact](#)。

您在使用時的合規責任 AWS 服務 取決於資料的敏感度、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源以協助合規：

- [安全性與合規快速入門指南](#) – 這些部署指南討論架構考量，並提供以 AWS 安全性和合規為重點的基準環境部署步驟。
- [Amazon Web Services 上 HIPAA 安全與合規架構](#) – 本白皮書說明公司如何使用 AWS 來建立 HIPAA 符合資格的應用程式。

**Note**

並非所有 AWS 服務 都HIPAA符合資格。如需詳細資訊，請參閱[HIPAA合格服務參考](#)。

- [AWS 合規資源](#) – 此工作手冊和指南集可能適用於您的產業和位置。
- [AWS 客戶合規指南](#) – 透過合規的角度了解共同的責任模型。本指南摘要說明保護 AWS 服務 指南並映射到跨多個架構的安全控制項的最佳實務（包括國家標準和技術研究所（）NIST、支付卡產業安全標準協會（PCI）和國際標準化組織（ISO））。
- AWS Config 開發人員指南中的[使用規則評估資源](#) – AWS Config 服務會評估資源組態是否符合內部實務、產業準則和法規。
- [AWS Security Hub](#) – 這 AWS 服務 提供 內安全狀態的全面檢視 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [Amazon GuardDuty](#) – 這透過監控環境是否有可疑和惡意活動來 AWS 服務 偵測 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可以透過滿足某些合規架構強制要求的入侵偵測需求，協助您解決各種合規要求DSS，例如 PCI。
- [AWS Audit Manager](#) – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和產業標準的方式。

## 其他資訊

如需 AWS Cloud 合規的一般資訊，請參閱下列內容：

- [FIPS 依服務區分的端點](#)
- [中的服務更新 ElastiCache](#)
- [AWS 雲端合規](#)
- [共同的責任模型](#)
- [AWS PCI DSS 合規計劃](#)

## Amazon 中的復原能力 ElastiCache

AWS 全域基礎設施是以 AWS 區域和可用區域為基礎。AWS 區域提供多個實體隔離和隔離的可用區域，這些區域與低延遲、高輸送量和高冗餘聯網連接。透過可用區域，您所設計與操作的應用程式和資

料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱[AWS 全域基礎設施](#)。

除了 AWS 全球基礎設施之外，Amazon ElastiCache 還提供多種功能，以協助支援您的資料彈性和備份需求。

## 主題

- [減少故障](#)

## 減少故障

規劃 Amazon ElastiCache 實作時，您應該規劃，讓失敗對您的應用程式和資料產生最小的影響。本節中的主題涵蓋您可以採取用來保護您的應用程式和資料不受故障的方法。

## 主題

- [緩解執行 Memcached 時的故障](#)
- [緩解執行 Valkey 或 Redis 時的失敗 OSS](#)
- [建議](#)

## 緩解執行 Memcached 時的故障

執行 Memcached 引擎時，您有下列選項可用於將故障的影響最小化。在您的故障緩解規劃中要解決兩個類型的故障：節點故障和可用區域故障。

### 緩解節點故障

無伺服器快取會利用複寫的 Multi-AZ 架構自動緩解節點故障，讓應用程式清楚了解節點故障的情況。若要緩解自行設計叢集中節點故障的影響，請將您的快取資料分散到多個節點。因為自行設計的叢集不支援複寫，因此節點故障將一律造成您的叢集遺失一些資料。

當您建立 Memcached 叢集時，您可以建立 1 到 60 個節點的叢集，或依特殊要求建立更多節點。將您的資料分割在更大量的節點間表示若節點故障，您遺失的資料較少。例如，如果將您的資料分割在 10 個節點間，任何單一節點大約會存放 10% 的快取資料。在此情況下，節點故障會遺失大約 10% 的快取，這個部分需要在建立和佈建替代節點時加以替代。如果在 3 個較大節點中快取相同的資料，節點的故障會遺失大約 33% 的快取資料。

如需指定 Memcached 叢集中節點數量的詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

## 緩解可用區域故障

無伺服器快取會利用複寫的 Multi-AZ 架構自動緩解可用區域故障，讓應用程式清楚了解 AZ 故障的情況。

若要緩解自行設計叢集中可用區域故障的影響，請盡可能將您的節點分散放到多個可用區域中。萬一發生 AZ 故障，您將遺失在該 AZ 中快取的資料，而不是在其他中快取的資料 AZs。

為什麼要有這麼多節點？

如果我的區域只有 3 個可用區域，既然若一個 AZ 故障，我會遺失大約 1/3 的資料，為什麼我需要超過 3 個節點？

這是個好問題。請記得我們正嘗試緩解兩個獨特類型的故障，節點和可用區域。您是對的，如果您的資料分散在可用區域間，當其中一個區域故障時，您只會遺失在該 AZ 中快取的資料，而不論您擁有的節點數量為何。不過，如果某個節點故障，擁有的節點數量越多可減少資料遺失的比例。

沒有「神奇公式」可決定您叢集中的節點數量。您必須衡量資料遺失的影響、故障可能性與成本，並從中得出自己的結論。

如需指定 Memcached 叢集中節點數量的詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

如需區域和可用區域的詳細資訊，請參閱[區域和可用區域](#)。

## 緩解執行 Valkey 或 Redis 時的失敗 OSS

執行 Valkey 或 Redis OSS 引擎時，您有下列選項，可將節點或可用區域故障的影響降至最低。

### 緩解節點故障

無伺服器快取會利用 Multi-AZ 架構自動緩解節點故障，讓應用程式清楚了解節點故障的情況。自行設計的叢集必須經過適當設定，才能緩解個別節點的故障。

若要減輕 Valkey 或 Redis OSS 節點故障對自行設計叢集的影響，您有下列選項：

#### 主題

- [緩解失敗：Valkey 或 Redis OSS 複寫群組](#)

### 緩解失敗：Valkey 或 Redis OSS 複寫群組

Valkey 或 Redis OSS 複寫群組由單一主節點組成，您的應用程式可以讀取和寫入 1 到 5 個唯讀複本節點。每當資料寫入主要節點，它也會在僅供讀取複本節點上非同步更新。

## 當僅供讀取複本故障時

1. ElastiCache 會偵測失敗的僅供讀取複本。
2. ElastiCache 會取得失敗的節點離線。
3. ElastiCache 在相同的 AZ 中啟動和佈建取代節點。
4. 新節點會與主要節點同步。

在這段時間，您的應用程式可以繼續使用其他節點來讀取和寫入。

## Valkey 或 Redis OSS Multi-AZ

您可以在 Valkey 或 Redis OSS 複寫群組上啟用多可用區。無論您是否啟用多個可用區，系統將偵測並自動取代故障的主要節點。它的發生方式可能因多可用區域是否已啟用而不同。

### 當多個可用區啟用時

1. ElastiCache 會偵測主要節點失敗。
2. ElastiCache 會將複寫延遲最低的僅供讀取複本節點提升為主要節點。
3. 另一個複本與新主要節點同步。
4. ElastiCache 會在失敗的 主要的 AZ 中啟動僅供讀取複本。
5. 新節點會與新提升的主要節點同步。

容錯移轉至複本節點的速度一般會較建立和佈建新主要節點來得快。這表示您的應用程式可以較未啟用多個可用區時更早繼續寫入至您的主要節點。

如需詳細資訊，請參閱[搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將 中的停機時間降到最低 OSS](#)。

### 當多個可用區停用時

1. ElastiCache 會偵測主要失敗。
2. ElastiCache 會離線使用主要。
3. ElastiCache 會建立新的主要節點並佈建，以取代失敗的主要節點。
4. ElastiCache 會將新的主要 與其中一個現有複本同步。
5. 同步完成時，新節點會如同叢集的主要節點般運作。



在此程序期間的步驟 1 到 4 期間，您的應用程式無法寫入主要節點。不過，應用程式可以繼續從複本節點讀取。

為了獲得更多保護，建議您在不同可用區域 ( ) 中啟動複寫群組中的節點AZs。如果您執行此動作，可用區域的故障只會影響該可用區域 (而非其他區域) 中的節點。

如需詳細資訊，請參閱[使用複寫群組的高可用性](#)。

## 緩解可用區域故障

無伺服器快取會利用複寫的 Multi-AZ 架構自動緩解可用區域故障，讓應用程式清楚了解 AZ 故障的情況。

若要緩解自行設計叢集中可用區域故障的影響，請盡量將每個碎片的節點分散放到多個可用區域中。

不論碎片中有多少個節點，如果它們全都放在相同可用區域中，只要該可用區域發生災難性故障，就會造成您遺失所有碎片的資料。不過，如果您在多個 中尋找節點AZs，則任何 AZ 的失敗都會導致您僅失去該 AZ 中的節點。

無論您在何時遺失節點，都會遇到效能降級，因為讀取操作現在由較少的節點共用。此效能降級將繼續，直到節點遭到取代為止。

如需指定 Valkey 或 Redis OSS節點可用區域的資訊，請參閱 [建立 Valkey \( 停用叢集模式 \) 叢集 \( 主控台 \)](#)。

如需區域及可用區域的詳細資訊，請參閱[選擇的區域和可用區域 ElastiCache](#)。

## 建議

我們建議您在自行設計的叢集上建立無伺服器快取，如此您無需額外的組態設定就能自動獲得更優異的容錯能力。不過，建立自行設計的叢集時，您需要針對兩個類型的故障進行規劃：個別節點故障和廣泛的可用區域故障。最佳的故障緩解規劃可解決這兩種類型的故障。

### 將節點故障的影響降至最低

若要在使用 Valkey 或 Redis 時將節點故障的影響降至最低OSS，建議您實作在每個碎片中使用多個節點，並將節點分佈到多個可用區域。無伺服器快取會自動進行此操作。

對於 Valkey 或 Redis 上的自行設計叢集OSS，我們建議您在複寫群組上啟用多可用區，以便在主要節點失敗時 ElastiCache 自動容錯移轉至複本。



執行 Memcached 並在節點間分割您的資料時，使用的節點愈多，任何節點故障時遺失的資料愈少。

將可用區域故障的影響降到最低

若要將可用區域故障的影響降到最低，建議您在盡可能多的不同可用區域中啟動節點。均勻分散節點 AZs，將最大限度地減少 AZ 故障的不太可能事件的影響。無伺服器快取會自動進行此操作。

其他預防工作

如果您正在執行 Valkey 或 Redis OSS，除了上述項目之外，建議您排程叢集的定期備份。備份 (快照) 會建立一個 .rdb 檔案，供您在故障或損毀的情況下用來還原您的快取。如需詳細資訊，請參閱[快照和還原](#)。

## 中的基礎設施安全 AWS ElastiCache

作為受管服務，AWS ElastiCache 受到 AWS 全球網路安全程序的保護，如 [AWS Architecture Center](#) 的安全與合規一節中所述。

您可以使用 AWS 已發佈的 API 呼叫，ElastiCache 透過網路存取。用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。我們建議使用 TLS 1.3 或更新版本。用戶端還必須支援具有完美正向保密性 (PFS) 的密碼套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman ( ) ECDHE。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，必須使用與 IAM 委託人相關聯的存取金鑰 ID 和秘密存取金鑰來簽署請求。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 以產生暫時安全憑證以簽署請求。

## 中的服務更新 ElastiCache

ElastiCache 會自動監控您的快取、叢集和節點機群，以在服務更新可用時套用更新。無伺服器快取的服務更新會自動且公開套用。對於自行設計的叢集，您可以設定預先定義的維護時段，以便 ElastiCache 可以套用這些更新。不過，在某些情況下，您可能會發現此方法過於嚴格，而且可能會限制您的業務流程。

您可以使用服務更新來控制要對您的自行設計叢集套用哪些更新，以及套用的時間。您也可以即時監控這些更新對所選 ElastiCache 叢集的進度。

主題

- [管理自我設計叢集的服務更新](#)

## 管理自我設計叢集的服務更新

ElastiCache 自我設計叢集的服務更新會定期發行。如果您有一或多個符合這些服務更新的自訂叢集，您會在更新發佈時透過電子郵件、SNS、個人運作狀態儀表板 (PHD) 和 Amazon CloudWatch 事件收到通知。更新也會顯示在 ElastiCache 主控台的服務更新頁面上。透過使用此儀表板，您可以檢視機 ElastiCache 群的所有服務更新及其狀態。無伺服器快取的服務更新會公開套用，且無法透過服務更新管理。

您可以在自動更新開始前，控制套用更新的時間。我們強烈建議您盡快套用任何類型的安全更新，以確保您的 ElastiCache 叢集始終 up-to-date 使用目前的安全修補程式。

以下區段更會詳細探討這些選項。

### 套用服務更新

您可以在更新的狀態變成 available (可用) 時，開始將服務更新套用到機群。服務更新為累積更新。換句話說，您未套用的任何更新都會包含在最新的更新中。

如果服務更新已啟用自動更新，您可以選擇在可用時不採取任何動作。ElastiCache 將排程在自動更新開始日期之後的其中一個叢集即將到來的維護時段期間套用更新。您將收到更新每個階段的相關通知。

#### Note

您只能套用狀態為 available (可用) 或 scheduled (已排程) 的服務更新。

如需檢閱和套用任何服務特定更新至適用 ElastiCache 叢集的詳細資訊，請參閱 [使用主控台套用服務更新](#)。

當一個或多個 ElastiCache 叢集有新的服務更新可用時，您可以使用 ElastiCache 主控台、API 或 AWS CLI 套用更新。以下區段會說明您可以用來套用更新的選項。

### 使用主控台套用服務更新

若要檢視可用服務更新的清單與其他資訊，請前往主控台的 Service Updates (服務更新) 頁面。

1. 登入 AWS Management Console 並在 開啟 Amazon ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>。

2. 在導覽窗格上，選擇 Service Updates (服務更新)。
3. 在 Service updates (服務更新) 中，您可以檢視下列項目：
  - Service update name (服務更新名稱)：服務更新的唯一名稱
  - Update type (更新類型)：服務更新的類型，可為 security-update (安全性更新) 或 engine-update (引擎更新)
  - Update severity (更新嚴重性)：套用更新的優先順序：
    - critical (重大)：建議您立即套用此更新 (在 14 天內)。
    - important (重要)：建議您盡快在業務流程允許的情況下套用此更新 (在 30 天內)。
    - medium (中等)：建議您盡快套用此更新 (在 60 天內)。
    - low (低度)：建議您盡快套用此更新程式 (在 90 天內)。
  - Engine version (引擎版本)：如果更新類型為 engine-update (引擎更新)，表示正在更新的引擎版本。
  - 發佈日期：更新發佈且可在叢集上套用的時間。
  - 建議套用截止日期：ElastiCache 套用更新的指引日期。
  - 狀態：更新的狀態，可以是下列狀態之一：
    - 可用：更新可用於必要的叢集。
    - complete (完成)：已套用更新。
    - cancelled (已取消)：更新已取消且不再需要。
    - 過期：更新已不可套用。
4. 選擇個別更新 (而非其左側的按鈕)，檢視服務更新的詳細資訊。

在 Cluster update status (叢集更新狀態) 區段中，您可以檢視叢集清單，這些叢集尚未套用或最近才套用服務更新。針對每個叢集，您可以檢視下列項目：

- Cluster name (叢集名稱)：叢集的名稱
- Nodes updated (已更新的節點)：特定叢集中的個別節點比率，這些節點已更新或仍可用於特定服務更新。
- Update Type (更新類型)：服務更新的類型，可為 security-update (安全性更新) 或 engine-update (引擎更新)
- Status (狀態)：叢集上服務更新的狀態，可以是下列狀態之一：
  - available (可用)：更新可用於必要的叢集。

- `scheduled` (已排程)：已排程更新日期。
- `complete` (完成)：已成功套用更新。狀態為 `complete` (完成) 的叢集會在完成後顯示 7 天。

如果您已選擇任何或所有狀態為 `available` (可用) 或 `scheduled` (已排程) 的叢集，然後選擇 `Apply now` (立即套用)，系統會開始將這些更新套用至這些叢集。

## 使用 AWS CLI 套用服務更新

在收到服務更新可供使用的通知後，您就可以使用 AWS CLI 來檢查和套用這些更新：

- 若要擷取可用服務更新的說明，請執行下列命令：

```
aws elasticache describe-service-updates --service-update-status
available
```

如需詳細資訊，請參閱 [describe-service-updates](#)。

- 若要在叢集清單上套用服務更新，請執行下列命令：

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

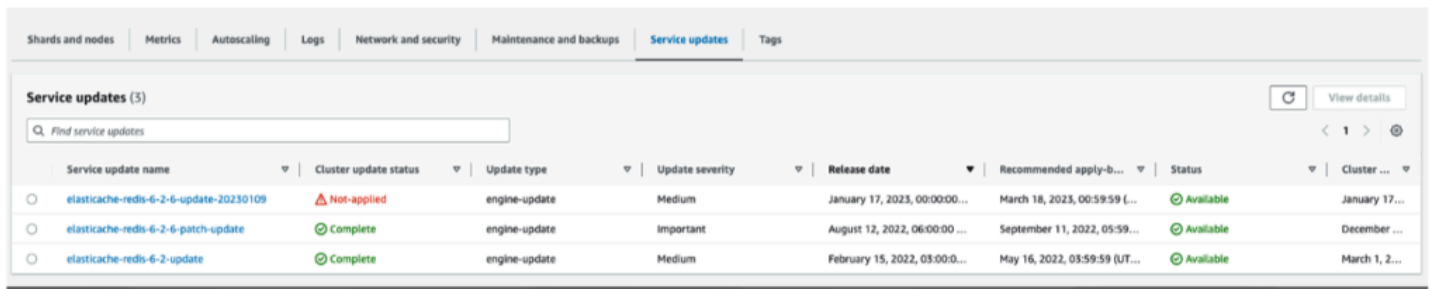
如需詳細資訊，請參閱 [batch-apply-update-action](#)。

## 使用 AWS 主控台驗證您已套用最新的 Service Update

您可以依照下列步驟，驗證 ElastiCache (Redis OSS) 叢集是否正在執行最新的服務更新：

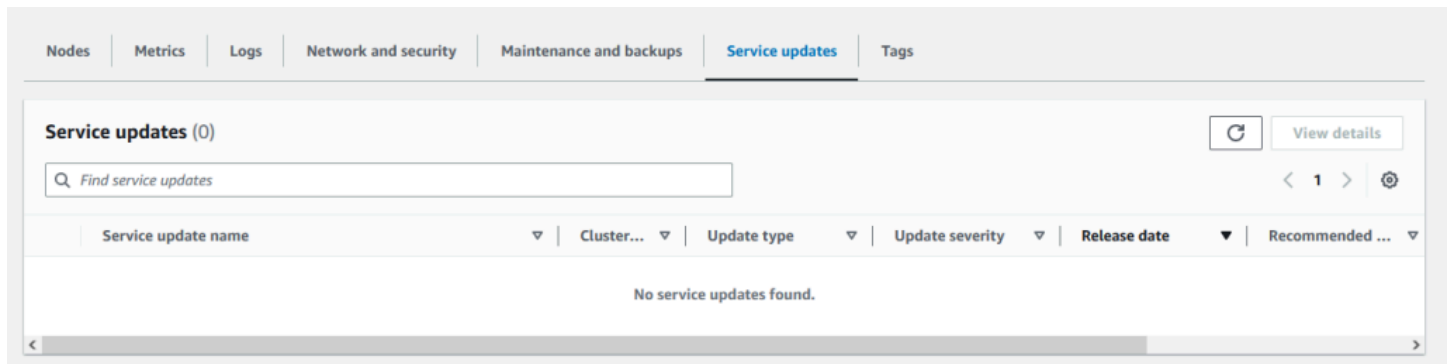
1. 在 Redis 叢集頁面上選擇適用的 OSS 叢集
2. 在導覽窗格中選擇服務更新，以查看該叢集適用的服務更新，如果有的話。

如果主控台顯示服務更新清單，您可以選擇服務更新，然後選擇立即套用。



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-202310109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 05:59:59 (UT...	Available	March 1, 2...

如果主控台顯示「找不到服務更新」，表示 ElastiCache (Redis OSS) 叢集已套用最新的服務更新。



Service update name	Cluster...	Update type	Update severity	Release date	Recommended ...
No service updates found.					

## 停止服務更新

如有需要，您可以停止叢集更新。例如，如果正在進行更新的叢集發生非預期的激增，建議您停止更新。或者，如果更新花費太長時間，而且在尖峰時間中斷業務流程，您可能會想要停止更新。

[停止中](#)操作會立即中斷對那些叢集以及還未更新之任何節點的所有更新。此操作會繼續完成狀態為 in progress (處理中) 的任何節點。然而，它會停止對相同叢集中狀態為 update available (可更新) 的其他節點的更新，並將這些節點還原成 Stopping (停止中) 狀態。

Stopping (停止中) 工作流程已完成，狀態為 Stopping (停止中) 的節點會變更為 Stopped (已停止) 狀態。根據更新的工作流程，某些叢集不會更新任何節點。其他叢集可能會包含一些已更新的節點以及狀態仍是 update available (可更新) 的節點。

您可以稍後返回，在業務流程允許的時候再完成更新進度。在這種情況下，請選擇您想要完成更新的適用叢集，接著選擇 Apply Now (立即套用)。如需詳細資訊，請參閱[套用服務更新](#)。

## 使用主控台

您可以使用 ElastiCache 主控台中斷服務更新。以下內容會示範其做法：

- 在所選叢集上進行服務更新之後，ElastiCache 主控台會在 ElastiCache 儀表板頂端顯示檢視/停止更新索引標籤。

- 若要中斷更新，請選擇 Stop Update (停止更新)。
- 停止更新時，選擇叢集並檢查狀態。叢集會還原成 Stopping (停止中) 狀態，最後會變成 Stopped (已停止) 狀態。

## 使用 AWS CLI

您可以使用 AWS CLI 來中斷服務更新。下列程式碼範例示範其做法：

若是複寫群組，請執行下列操作：

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

若是快取叢集，請執行下列操作：

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

如需詳細資訊，請參閱 [BatchStopUpdateAction](#)。

## 常見漏洞與暴露 ( CVE )：中解決的安全漏洞 ElastiCache

常見漏洞與暴露 ( CVE ) 是公開已知網路安全漏洞的項目清單。每個項目都是一個連結，其中包含識別號碼、描述和至少一個公有參考。您可以在此頁面中找到已在 中解決的安全漏洞清單 ElastiCache。

我們建議您一律升級至最新的 ElastiCache Valkey、Redis OSS 或 ElastiCache Memcached 版本，以防範已知的漏洞。操作 ElastiCache Serverless 快取時，CVE 修正會自動套用至快取。使用 Valkey 或 Redis 操作自行設計的叢集時 OSS，會 ElastiCache 曝光 PATCH 元件。例如，使用 ElastiCache ( Redis OSS ) 6.2.6 版時，主要版本為 6，次要版本為 2，修補程式版本為 6。PATCH 版本適用於向下相容的錯誤修正、安全性修正和功能失效變更。

您可以使用下表來驗證特定版本的 ElastiCache Valkey 和 Redis OSS 是否針對特定安全漏洞進行修正。如果您的 ElastiCache Valkey 或 Redis OSS 叢集執行的版本沒有安全性修正，請參閱下表並採取行動。您可以升級至包含修正的較新 ElastiCache Valkey 或 Redis OSS 版本，或者如果您使用的版本包含修正，請務必參閱 [以套用最新的服務更新管理自我設計叢集的服務更新](#)。如需支援的 ElastiCache 引擎版本以及如何升級的詳細資訊，請參閱 [在 中的引擎版本和升級 ElastiCache](#)。

### Note

- 如果在 ElastiCache 版本中處理 CVE ，則表示它也在較新的版本中處理。因此，例如，如果在 ElastiCache ( Redis OSS ) 6.0.5 版中解決了漏洞，則此弱點會繼續轉送至 6.2.6、7.0.7 和 7.1 版。
- 下表中的星號 ( \*OSS ) 表示您必須為執行 ElastiCache ( Redis ) 版本之 ( ElastiCache RedisOSS ) 叢集套用最新的服務更新，才能解決安全漏洞。如需如何驗證叢集正在執行的 ElastiCache ( Redis OSS ) 版本是否套用最新服務更新的詳細資訊，請參閱 [管理自我設計叢集的服務更新](#)。

ElastiCache ( Redis OSS ) 版本	CVEs 已定址
Redis OSS 6.0.5	<a href="#">CVE-2022-24735*</a> 、 <a href="#">CVE-2022-24736*</a>
Redis OSS 6.2.6	<a href="#">CVE-2022-24834*</a> 、 <a href="#">CVE-2022-35977*</a> 、 <a href="#">CVE-2022-36021*</a> 、 <a href="#">CVE-2022-24735</a> 、 <a href="#">CVE-2022-24736</a>
Redis OSS 7.0.7	<a href="#">CVE-2023-41056*</a> 、 <a href="#">CVE-2022-24834*</a> 、 <a href="#">CVE-2022-35977</a> 、 <a href="#">CVE-2022-36021</a> 、 <a href="#">CVE-2022-24735</a> 、 <a href="#">CVE-2022-24736</a>
Redis OSS 7.1.0	<a href="#">CVE-2023-41056</a> 、 <a href="#">CVE-2022-24834</a> 、 <a href="#">CVE-2022-35977</a> 、 <a href="#">CVE-2022-36021</a> 、 <a href="#">CVE-2022-24735</a> 、 <a href="#">CVE-2022-24736</a>



# 在 Amazon 中記錄和監控 ElastiCache

若要管理快取，您務必要了解快取的執行方式。ElastiCache 會產生發佈至 Amazon CloudWatch Logs 的指標，以監控快取效能。此外，當快取資源發生重大變更（例如建立新的快取或刪除快取）時，ElastiCache 會產生事件。

## 主題

- [Valkey 和 Redis 的無伺服器指標和事件 OSS](#)
- [Memcached 的無伺服器指標和事件](#)
- [使用記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail](#)
- [Amazon ElastiCache 事件 SNS 監控](#)
- [日誌傳送](#)
- [監控與 CloudWatch 指標搭配使用](#)
  
- [使用記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail](#)

## Valkey 和 Redis 的無伺服器指標和事件 OSS

ElastiCache 提供廣泛的指標和事件，以便在使用無伺服器快取時進行監控。這包括 CloudWatch 指標、命令層級指標和事件日誌，可透過 Amazon 擷取 EventBridge。

## 主題

- [無伺服器快取指標](#)
- [無伺服器快取事件](#)
- [針對 Valkey 和 Redis 自行設計的叢集指標和事件 OSS](#)

## 無伺服器快取指標

AWS/ElastiCache 命名空間包含 Valkey 或 Redis 無 OSS 伺服器快取的下列 CloudWatch 指標。

Valkey 或 Redis 的指標代碼 OSS



指標	描述	單位
BytesUsedForCache	儲存在快取中的資料所使用的位元組總數。	位元組
ElastiCacheProcessingUnits	在快取上執行的請求所耗用的 ElastiCacheProcessingUnits ( ECPUs ) 總數	計數
SuccessfulReadRequestLatency	成功讀取請求的延遲。	微秒
SuccessfulWriteRequestLatency	成功寫入請求的延遲	微秒
TotalCmdsCount	快取上執行的所有命令總數	計數
CacheHitRate	表示快取的命中率。這是使用 <code>cache_hits</code> 和 <code>cache_misses</code> 統計資料以下列方式計算的： $cache\_hits / (cache\_hits + cache\_misses)$ 。	百分比
CacheHits	快取中成功的唯讀索引鍵查詢次數。	計數
CurrConnections	快取的用戶端連線數。	計數
ThrottledCmds	由於工作負載擴展速度比 ElastiCache 擴展速度快，ElastiCache 因此調節的請求數量。	計數
NewConnections	在此期間內，伺服器已接受的連線總數。	計數
CurrItems	快取中的項目數。	計數

指標	描述	單位
CurrVolatileItems	快取中具有 的項目數量TTL。	計數
NetworkBytesIn	傳入快取的位元組總數	位元組
NetworkBytesOut	從快取傳出的位元組總數	位元組
移出	由快取移出的索引鍵計數	計數
IamAuthenticationExpirations	過期IAM且經過驗證的 Valkey 或 Redis OSS連線總數。您可以在使用者指南中找到有關 <a href="#">使用 驗證 IAM</a> 的詳細資訊。	計數
IamAuthenticationThrottling	限流 IAM- 驗證的 Valkey 或 Redis OSSAUTH或HELLO請求總數。您可以在使用者指南中找到有關 <a href="#">使用 驗證 IAM</a> 的詳細資訊。	計數
KeyAuthorizationFailures	使用者嘗試存取他們沒有存取許可的金鑰失敗總次數。建議對此設定警示，以偵測未經授權的存取嘗試。	計數
AuthenticationFailures	OSS 使用 AUTH命令向 Valkey 或 Redis 進行身分驗證的失敗嘗試總數。建議對此設定警示，以偵測未經授權的存取嘗試。	計數
CommandAuthorizationFailures	使用者嘗試執行他們沒有呼叫許可的命令失敗總次數。建議對此設定警示，以偵測未經授權的存取嘗試。	計數

## 命令層級指標

ElastiCache 也會發出下列命令層級指標。對於每個命令類型，會 ElastiCache 發出命令的總計數，以及該命令類型ECPUs消耗的數目。

指標	描述	單位
EvalBasedCmds	快取已接收到的 get 命令的數量。	計數
EvalBasedCmdsECPUs	ECPUs 以 eval 為基礎的命令所消耗的。	計數
GeoSpatialBasedCmds	以 geospatial- 為基礎的之命令的命令總數。這是衍生自 Valkey 或 Redis OSS命令統計資料。這是藉由加總所有 geo 類型的命令所衍生：geoadd、geodist、geo hash、geopos、georadius 和 georadiusbymember。	計數
GeoSpatialBasedCmdsECPUs	ECPUs 以地理空間為基礎的命令所消耗的。	計數
GetTypeCmds	唯讀類型命令的總數。這是透過加總所有唯讀類型命令（get、hget、scard、lrange 等）來衍生自 Valkey 或 Redis OSS命令統計資料。	計數
GetTypeCmdsECPUs	ECPUs 讀取命令所耗用。	計數
HashBasedCmds	雜湊類型命令總數。這是從 Valkey 或 Redis OSS命令統計資料衍生而來，方法是加總對一或多個雜湊（hget、hkey、hvals、hdel 等）採取行動的所有命令。	計數

指標	描述	單位
HashBasedCmdsECPUs	ECPUs 由雜湊型命令耗用。	計數
HyperLogLogBasedCmds	HyperLogLog 以為基礎的命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總所有命令的 pf 類型 ( pfadd、pfcount、pfmerge 等 )。	計數
HyperLogLogBasedCmdsECPUs	ECPUs 以 HyperLogLog 為基礎的命令所耗用。	計數
JsonBasedCmds	JSON 命令的總數，包括讀取和寫入命令。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對 JSON 金鑰採取行動的所有 JSON 命令。	計數
JsonBasedCmdsECPUs	ECPUs 由所有 JSON 命令耗用，包括讀取和寫入命令。	計數
JsonBasedGetCmds	JSON 唯讀命令的總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對 JSON 金鑰採取行動的所有 JSON 讀取命令。	計數
JsonBasedGetCmdsECPUs	ECPUs 唯讀 JSON 命令所耗用。	計數
JsonBasedSetCmds	JSON 寫入命令的總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對 JSON 金鑰採取行動的所有 JSON 寫入命令。	計數
JsonBasedSetCmdsECPUs	ECPUs JSON 寫入命令消耗。	計數

指標	描述	單位
KeyBasedCmds	金鑰類型命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來的，方法是加總在多個資料結構（del、過期、重新命名等）中對一或多個金鑰執行動作的所有命令。	計數
KeyBasedCmdsECPUs	ECPUs 金鑰型命令所耗用。	計數
ListBasedCmds	清單類型命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對一或多個清單（lindex、lrange、lpush、ltrim 等）採取行動的所有命令。	計數
ListBasedCmdsECPUs	ECPUs 清單型命令所耗用。	計數
NonKeyTypeCmds	非金鑰型命令的總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總所有不對金鑰採取行動的命令，例如 acl、dbsize 或資訊。	計數
NonKeyTypeCmdsECPUs	ECPUs 由 non-key-based 命令耗用。	計數

指標	描述	單位
PubSubBasedCmds	pub/sub 功能的命令總數。這是從 Valkey 或 Redis OSS命令統計資料衍生而來的，方法是加總用於 pub/sub 功能的所有命令：psubscribe、publish、pubsub、punsubscribe、ssubscribe、sunsubscribe、spublish、subscribe 和 unsubscribe。	計數
PubSubBasedCmdsECPUs	ECPUs pub/sub 型命令所耗用。	計數
SetBasedCmds	集合類型命令總數。這是從 Valkey 或 Redis OSS命令統計資料衍生而來的，方法是加總對一或多個集（卡片、sdiff、sadd、Sunion 等）採取行動的所有命令。	計數
SetBasedCmdsECPUs	ECPUs 由以集為基礎的命令耗用。	計數
SetTypeCmds	寫入類型命令的總數。這是從 Valkey 或 Redis OSS命令統計資料衍生而來的，方法是加總在資料（集合、資料集、sadd、lpop 等）上操作的所有命令類型。	計數
SetTypeCmdsECPUs	ECPUs 寫入命令所耗用。	計數

指標	描述	單位
SortedSetBasedCmds	有序集合類型命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對一或多個排序集（zcount、zrange、zrank、zadd 等）採取行動的所有命令。	計數
SortedSetBasedCmdsECPUs	ECPUs 已排序的命令所耗用。	計數
StringBasedCmds	字串類型命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對一或多個字串（字串、setex、setrange 等）採取行動的所有命令。	計數
StringBasedCmdsECPUs	ECPUs 字串型命令所耗用。	計數
StreamBasedCmds	串流類型命令總數。這是從 Valkey 或 Redis OSS 命令統計資料衍生而來，方法是加總對一或多個串流資料類型（xrange、xlen、xadd、xdel 等）採取行動的所有命令。	計數
StreamBasedCmdsECPUs	ECPUs 串流型命令所耗用。	計數

## 無伺服器快取事件

ElastiCache 會記錄與無伺服器快取相關的事件。此資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的描述。您可以使用 ElastiCache 主控台、AWS CLI `describe-events` 命令或 ElastiCache API 動作 `DescribeEvents` 輕鬆從日誌擷取事件。

您可以選擇使用 Amazon 來監控、擷取、轉換 ElastiCache 事件並對其採取行動 EventBridge。如需詳細資訊，請參閱 Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>。

## 檢視 ElastiCache 事件（主控台）

若要使用 ElastiCache 主控台檢視事件：

1. 登入 AWS Management Console 並開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。
3. 在事件畫面上，清單的每一列代表一個事件，並顯示事件來源、事件類型、事件 GMT 時間，以及事件的描述。您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

## 檢視 ElastiCache 事件（AWS CLI）

若要使用產生 ElastiCache 事件清單 AWS CLI，請使用命令 `describe-events`。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個無伺服器快取事件。

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內無伺服器快取的所有事件。

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

## 無伺服器事件

本節記錄您可能收到的不同類型無伺服器快取事件。

### 無伺服器快取建立事件

Detail-Type	描述	單位	來源	訊息
快取已建立	Cache arn	建立	serverless-cache	快取 <cache-name> 已建立且可供使用。



Detail-Type	描述	單位	來源	訊息
快取已建立	Cache arn Snapshot arn	建立	serverless-cache	快取 <cache-name> 已建立，並已從快照還原資料。您的快取可供使用。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。可用 IP 地址不足，無法建立 VPC 端點。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。請求中提供的子網路無效。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。達到建立 VPC 端點的配額限制。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。您沒有建立 VPC 端點的許可。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。使用者群組 <user-group-name> 中存在具有不相容 Valkey 或 Redis OSS 版本的使用者。

Detail-Type	描述	單位	來源	訊息
快取建立失敗	Cache arn Cache snapshot arn	失敗	serverless-cache	無法建立快取 <cache-name>。 提供的使用者群組 <user-group-name> 不存在。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。 從快照還原資料失敗，原因：<reason>。  失敗原因： <ul style="list-style-type: none"> <li>無法從 S3 擷取檔案。</li> <li>預期的 md5 與實際的 md5 不相符。</li> <li>提供的RDB檔案具有不支援的版本。</li> </ul>

### 無伺服器快取更新事件 ( Valkey 或 RedisOSS )

Detail-Type	資源清單	類別	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	SecurityGroups 已更新快取 <cache-name>。
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的標籤已更新。

Detail-Type	資源清單	類別	來源	訊息
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 更新失敗。使用者群組 <user-group-name> 中存在具有不相容 Valkey 或 Redis OSS 版本的使用者。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的更新失敗。SecurityGroups 更新失敗。
快取更新失敗	Cache arn	組態變更	serverless-cache	由於許可不足，快取 <cache-name> 的更新失敗。SecurityGroups 更新失敗。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 更新失敗。SecurityGroups 更新失敗，因為 SecurityGroups 無效。

### 無伺服器快取刪除事件 ( Valkey 或 RedisOSS )

Detail-Type	資源清單	類別	來源	訊息
快取已刪除	Cache arn	刪除	serverless-cache	快取 <cache-name> 已刪除。

### 無伺服器快取用量限制事件 ( Valkey 或 RedisOSS )

Detail-Type	描述	單位	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的限制已更新。
即將達到快取限制	Cache arn	notification	serverless-cache	位置 <X> 使用超過 <Y>% 的每個位置限制 32 GB。例如，位置 10 使用超過 90% 的每個位置限制 32 GB。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為快取已刪除。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為組態無效。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為目前快取的資料超過新的限制。請先

Detail-Type	描述	單位	來源	訊息
				清空部分資料，再套用限制。

無伺服器快取快照事件 ( Valkey 或 RedisOSS )

Detail-Type	資源清單	類別	來源	訊息
快照已建立	Cache arn Snapshot arn	建立	serverless-cache-snapshot	已建立快取 <cache-name> 的快照 <snapshot-name>。
快照建立失敗	Cache arn Snapshot arn	失敗	serverless-cache-snapshot	無法建立快取 <cache-name> 的快照。快照 <snapshot-name> 建立失敗，客戶受管金鑰 <key-id>，<reason>。  失敗原因訊息： <ul style="list-style-type: none"> <li>• 因為客戶受管金鑰已停用</li> <li>• 因為找不到客戶受管金鑰</li> <li>• 因為請求逾時</li> </ul>
快照建立失敗	Cache arn Snapshot arn	失敗	serverless-cache-snapshot	無法建立快取 <cache-name> 的快照。快照 <snapshot-name> 建立

Detail-Type	資源清單	類別	來源	訊息
				<p>失敗，原因：&lt;reason&gt;。</p> <p>預設原因：</p> <ul style="list-style-type: none"> <li>• 因為內部錯誤</li> </ul>
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	<p>無法匯出快取 &lt;cache-name&gt; 的快照。無法將快照匯出至儲存貯體 %s，因為 ElastiCache 沒有儲存貯體的許可。</p>
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	<p>無法匯出快取 &lt;cache-name&gt; 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體中已有同名的物件。</p>
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	<p>無法匯出快取 &lt;cache-name&gt; 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體擁有者帳戶 ID 已變更。</p>

Detail-Type	資源清單	類別	來源	訊息
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為無法存取 S3 儲存貯體。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為無法存取儲存貯體。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體不存在。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至具有來源快照客戶受管金鑰 %s 的儲存貯體 '%s'，<reason>。

Detail-Type	資源清單	類別	來源	訊息
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快照 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'。
快照複製失敗	Snapshot arn-1 Snapshot arn-2	失敗	serverless-cache-snapshot	無法複製快照 <snapshot-name>。無法將快照 '%s' 複製到快照 '%s'，因為來源快照客戶受管金鑰為 <key-id>，<reason-name>。
快照複製失敗	Snapshot arn-1 Snapshot arn-2	失敗	serverless-cache-snapshot	無法複製快照 <snapshot-name>。無法將快照 '%s' 複製到快照 '%s'，因為目標快照客戶受管金鑰為 '%s' '%s'。

## 針對 Valkey 和 Redis 自行設計的叢集指標和事件 OSS

ElastiCache 提供各式各樣的指標和事件，可在使用 Valkey 和 Redis OSS 時監控自行設計的叢集。這包括透過 [Amazon Simple Notification Service \(SNS\)](#) 提供的主機層級指標、命令層級指標 AWS CLI 和事件日誌 SNS。

### 主題

- [自行設計叢集的指標](#)
- [自行設計叢集的事件 \( Valkey 和 RedisOSS \)](#)



## 自行設計叢集的指標

當您自行設計叢集時，會在每個節點層級 ElastiCache 發出指標，包括主機層級指標和快取指標。

如需主機層級指標的詳細資訊，請參閱 [主機層級指標](#)。

如需節點層級指標的詳細資訊，請參閱 [Valkey 和 Redis 的指標 OSS](#)。

## 自行設計叢集的事件 ( Valkey 和 RedisOSS )

ElastiCache 會記錄與您自行設計快取相關的事件。使用自行設計的叢集時，您可以使用 或使用 Amazon Simple Notification Service AWS CLI ( ) 在 ElastiCache 主控台中檢視叢集事件 SNS。自行設計的叢集事件不會發佈至 Amazon EventBridge。

自行設計的叢集事件資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的說明。您可以使用 ElastiCache 主控台、AWS CLI describe-events 命令或 ElastiCache API 動作 輕鬆從日誌中擷取事件 DescribeEvents。

### 檢視 ElastiCache 事件 ( 主控台 )

下列程序使用 ElastiCache 主控台顯示事件。

使用 ElastiCache 主控台檢視事件

1. 登入 AWS Management Console 並在 開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。
3. 在事件畫面上，清單的每一列代表一個事件，並顯示事件來源、事件類型、事件 GMT 時間，以及事件的描述。您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

### 檢視 ElastiCache 事件 ( AWS CLI )

若要使用 產生 ElastiCache 事件清單 AWS CLI，請使用 命令 describe-events。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個自行設計叢集事件。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內自行設計快取的所有事件。

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

## 自行設計的叢集事件

本節包含您預期會收到的自行設計叢集事件清單。

下列 ElastiCache 事件會觸發 Amazon SNS通知。如需事件的詳細資訊，請參閱 [檢視 ElastiCache 事件](#)。

事件名稱	訊息	描述
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	快取節點已新增至快取叢集，且已就緒可供使用。
ElastiCache : AddCacheNodeFailed 由於可用 IP 地址不足	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	由於可用的 IP 地址不足，而無法新增快取節點。
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	一或多個快取叢集參數已變更。
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	快取叢集的佈建已完成，且快取叢集中的快取節點已就緒並可供使用。
ElastiCache : CacheClusterProvisioningFailed 由於網路狀態不相容	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	已嘗試將新的快取叢集啟動到不存在的虛擬私有雲端 ( VPC )。
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	快取叢集的擴展已成功完成。
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	快取叢集上的擴展操作失敗。

事件名稱	訊息	描述
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>已發生下列其中一項事件：</p> <ul style="list-style-type: none"> <li>• 授權給快取叢集的快取安全群組清單已修改。</li> <li>• 與快取EC2叢集相關聯的任何快取安全群組上已授權一或多個新的安全群組。</li> <li>• 一或多個EC2安全群組已從與快取叢集相關聯的任何快取安全群組中撤銷。</li> </ul>
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機降級或無法連線，並已開始取代快取節點。</p> <div data-bbox="1068 1010 1507 1226" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>取代的快取節點DNS項目不會變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。不過，即使 ElastiCache 已取代快取節點，某些快取用戶端程式庫仍可能停止使用快取節點；在此情況下，應用程式應在發生此事件時重新整理伺服器清單。</p>

事件名稱	訊息	描述
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機降級或無法連線，且已完成取代快取節點。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b> 取代的快取節點DNS項目不會變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。不過，即使 ElastiCache 已取代快取節點，某些快取用戶端程式庫仍可能停止使用快取節點；在此情況下，應用程式應在發生此事件時重新整理伺服器清單。</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>一或多個快取節點已重新開機。</p> <p>訊息 (Memcached) : "Cache node %s shutdown" 然後，第二個訊息 : "Cache node %s restarted"</p>
ElastiCache : CertificateRenewalComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:CertificateRenewalComplete	Amazon CA 憑證已成功續約。
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	已成功建立複寫群組。

事件名稱	訊息	描述
ElastiCache:DeleteCacheClusterComplete	ElastiCache:DeleteCacheClusterComplete : <i>cluster-name</i>	已完成刪除快取叢集和所有與其相關聯的快取節點。
ElastiCache : FailoverComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:FailoverComplete : <i>mycluster</i>	已成功容錯移轉至複本節點。
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	叢集中的複本數量已增加。
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	將複本新增至叢集的程序已經開始。
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	叢集中原先已排程替換的節點，已不再排程替換。
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>叢集中原先已排程替換的節點，已重新排程在通知中所述的新視窗期間進行替換。</p> <p>如需您可以採取哪些動作的資訊，請參閱<a href="#">更換節點 ( Valkey 和 RedisOSS )</a>。</p>

事件名稱	訊息	描述
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	您叢集中的節點，已排程在通知中所述的視窗期間進行替換。  如需您可以採取哪些動作的資訊，請參閱 <a href="#">更換節點 ( Valkey 和 RedisOSS )</a> 。
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	快取節點已從快取叢集移除。
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	已順利完成複寫群組上的向上擴展操作。
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	複寫群組上的擴展操作失敗。
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	節點有可用的自助更新。
ElastiCache : SnapshotComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:SnapshotComplete : <i>cluster-name</i>	快取快照已順利完成。
ElastiCache : SnapshotFailed ( OSS僅限 Valkey 或 Redis )	SnapshotFailed : <i>cluster-name</i>	快取快照已失敗。請參閱叢集的快取事件，以取得失敗原因的詳細資訊。  如果您描述快照，請查看 <a href="#">DescribeSnapshots</a> ，其狀態將是 failed。



## Memcached 的無伺服器指標和事件

本節說明使用無伺服器快取時可監控的指標和事件。

### 主題

- [無伺服器快取指標](#)
- [無伺服器快取事件](#)

## 無伺服器快取指標

本節說明使用 Memcached 無伺服器快取時可監控的指標和事件。

AWS/ElastiCache 命名空間包含 Memcached 無伺服器快取的下列 CloudWatch 指標。

指標	描述	單位
BytesUsedForCache	儲存在快取中的資料所使用的位元組總數。	位元組
ElastiCacheProcessingUnits	在快取上執行的請求所耗用的 ElastiCacheProcessingUnits ( ECPUs ) 總數	計數
SuccessfulReadRequestLatency	成功讀取請求的延遲。	微秒
SuccessfulWriteRequestLatency	成功寫入請求的延遲	微秒
TotalCmdsCount	快取上執行的所有命令總數	計數
CurrConnections	快取的用戶端連線數。	計數
ThrottledCmds	由於工作負載擴展速度比 ElastiCache 擴展速度快，ElastiCache 因此調節的請求數量。	計數



指標	描述	單位
NewConnections	在此期間內，伺服器已接受的連線總數。	計數
CurrItems	快取中的項目數。	計數
NetworkBytesIn	傳入快取的位元組總數	位元組
NetworkBytesOut	從快取傳出的位元組總數	位元組
移出	由快取移出的索引鍵計數	計數
已回收	快取使其到期的金鑰數目	計數

### 命令層級指標

ElastiCache 也會發出下列 Memcached 命令層級指標

指標	描述	單位
CmdGet	快取已接收到的 get 命令的數量。	計數
CmdSet	快取接收到的 set 命令數量。	計數
CmdTouch	快取已接收到的 touch 命令數量。	計數
GetHits	在請求的金鑰已找到的情況下，快取接收到的取得請求的數量。	計數
GetMisses	快取已收到其請求的金鑰未找到的取得請求的數量。	計數
IncrHits	在請求的金鑰已找到的情況下，快取接收到的增加請求的數量。	計數

指標	描述	單位
IncrMisses	在請求的金鑰未找到的情況下，快取接收到的增加請求的數量。	計數
DecrHits	在請求的金鑰已找到的情況下，快取接收到的減少請求的數量。	計數
DecrMisses	在請求的金鑰未找到的情況下，快取接收到的減少請求的數量。	計數
DeleteHits	在請求的金鑰已找到的情況下，快取接收到的刪除請求的數量。	計數
DeleteMisses	在請求的金鑰未找到的情況下，快取接收到的刪除請求的數量。	計數
TouchHits	已被碰觸並賦予新的過期時間的金鑰的數量。	計數
TouchMisses	已執行 touch 但未找到的金鑰數量。	計數
CasHits	在請求的金鑰已找到且 cas 值相符的情況下，快取已接收到的 cas 請求數量。	計數
CasMisses	在請求的金鑰未找到的情況下，快取已接收到 cas 請求數量。	計數
CasBadval	在 cas 值與儲存的 cas 值不相符的情況下，快取已接收到的 cas 請求數量。	計數

指標	描述	單位
CmdFlush	快取已接收到的 flush 命令的數量。	計數

## 無伺服器快取事件

ElastiCache 會記錄與無伺服器快取相關的事件。此資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的描述。您可以使用 ElastiCache 主控台、AWS CLI `describe-events` 命令或 ElastiCache API 動作 `DescribeEvents` 輕鬆從日誌擷取事件。

您可以選擇使用 Amazon 來監控、擷取、轉換 ElastiCache 事件並對其採取行動 EventBridge。如需詳細資訊，請參閱 Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>。

### 檢視 ElastiCache 事件（主控台）

若要使用 ElastiCache 主控台檢視事件：

1. 登入 AWS Management Console，並在開啟 ElastiCache 主控台 <https://console.aws.amazon.com/elasticache/>
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。
3. 在事件畫面上，清單的每一列代表一個事件，並顯示事件來源、事件類型、事件 GMT 時間，以及事件的描述。您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

### 檢視 ElastiCache 事件（AWS CLI）

若要使用產生 ElastiCache 事件清單 AWS CLI，請使用命令 `describe-events`。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個無伺服器快取事件。

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內無伺服器快取的所有事件。

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

## 無伺服器事件

本節記錄您可能收到的不同類型無伺服器快取事件。

### 無伺服器快取建立事件

Detail-Type	描述	單位	來源	訊息
快取已建立	Cache arn	建立	serverless-cache	快取 <cache-name> 已建立且可供使用。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。可用 IP 地址不足，無法建立 VPC 端點。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。請求中提供的子網路無效。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。達到建立 VPC 端點的配額限制。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。您沒有建立 VPC 端點的許可。

### 無伺服器快取更新事件 ( Memcached )

Detail-Type	資源清單	類別	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	SecurityGroups 已更新快取 <cache-name>。
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的標籤已更新。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的更新失敗。SecurityGroups 更新失敗。
快取更新失敗	Cache arn	組態變更	serverless-cache	由於許可不足，快取 <cache-name> 的更新失敗。SecurityGroups 更新失敗。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的更新失敗。SecurityGroups 更新失敗，因為 SecurityGroups 無效。

#### 無伺服器快取刪除事件 ( Memcached )

Detail-Type	資源清單	類別	來源	訊息
快取已刪除	Cache arn	刪除	serverless-cache	快取 <cache-name> 已刪除。

## 無伺服器快取用量限制事件 ( Memcached )

Detail-Type	描述	單位	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的限制已更新。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為快取已刪除。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為組態無效。

## 無伺服器快取快照事件 ( Memcached )

Detail-Type	資源清單	類別	來源	訊息
快照已建立	Cache arn Snapshot arn	建立	serverless-cache-snapshot	已建立快取 <cache-name> 的快照 <snapshot-name>。
快照建立失敗	Cache arn Snapshot arn	失敗	serverless-cache-snapshot	無法建立快取 <cache-name> 的快照。快照 <snapshot-name> 建立失敗，客戶受管金鑰 <key-id>，<reason>。

Detail-Type	資源清單	類別	來源	訊息
				失敗原因訊息： <ul style="list-style-type: none"> <li>• 因為客戶受管金鑰已停用</li> <li>• 因為找不到客戶受管金鑰</li> <li>• 因為請求逾時</li> </ul>
快照建立失敗	Cache arn Snapshot arn	失敗	serverless-cache-snapshot	無法建立快取 <cache-name> 的快照。快照 <snapshot-name> 建立失敗，原因：<reason>。  預設原因： <ul style="list-style-type: none"> <li>• 因為內部錯誤</li> </ul>
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 %s，因為 ElastiCache 沒有儲存貯體的許可。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體中已有同名的物件。

Detail-Type	資源清單	類別	來源	訊息
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體擁有者帳戶 ID 已變更。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為無法存取 S3 儲存貯體。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為無法存取儲存貯體。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'，因為儲存貯體不存在。



Detail-Type	資源清單	類別	來源	訊息
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至具有來源快照客戶受管金鑰 %s 的儲存貯體 '%s'，<reason>。
快照匯出失敗	Snapshot arn	失敗	serverless-cache-snapshot	無法匯出快取 <cache-name> 的快照。無法將快照匯出至儲存貯體 '%s'。
快照複製失敗	Snapshot arn-1 Snapshot arn-2	失敗	serverless-cache-snapshot	無法複製快照 <snapshot-name>。無法將快照 '%s' 複製到快照 '%s'，因為來源快照客戶受管金鑰為 <key-id>，<reason-name>。
快照複製失敗	Snapshot arn-1 Snapshot arn-2	失敗	serverless-cache-snapshot	無法複製快照 <snapshot-name>。無法將快照 '%s' 複製到快照 '%s'，因為目標快照客戶受管金鑰為 '%s' '%s'。

# 使用記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail

Amazon 與 ElastiCache 整合 AWS CloudTrail，此服務提供 Amazon AWS 中使用者、角色或服務所採取動作的記錄 ElastiCache。CloudTrail 會將 Amazon 的所有 API 呼叫擷取 ElastiCache 為事件，包括從 Amazon ElastiCache 主控台呼叫，以及從程式碼呼叫到 Amazon ElastiCache API 操作。如果您建立追蹤，則可以啟用 CloudTrail 事件連續交付至 Amazon S3 儲存貯體，包括 Amazon 的事件 ElastiCache。如果您未設定追蹤，仍然可以在事件歷史記錄中檢視 CloudTrail 主控台最新的最新事件。使用所收集的資訊 CloudTrail，您可以判斷向 Amazon 提出的請求 ElastiCache、提出請求的 IP 地址、提出請求的人員、提出時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [AWS CloudTrail 使用者指南](#)。

## 中的 Amazon ElastiCache 資訊 CloudTrail

CloudTrail 當您建立 AWS 帳戶時，會在您的帳戶上啟用。當活動在 Amazon 中發生時 ElastiCache，該活動會與 CloudTrail 事件歷史記錄中的其他 AWS 服務事件一起記錄在事件中。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱 [檢視具有事件歷史記錄 CloudTrail 的事件](#)。

若要持續記錄您 AWS 帳戶中的事件，包括 Amazon 的事件 ElastiCache，請建立追蹤。追蹤可讓 CloudTrail 將日誌檔案傳遞至 Amazon S3 儲存貯體。根據預設，當您在主控台建立權杖時，權杖會套用到所有區域。追蹤會記錄 AWS 分割區中所有區域的事件，並將日誌檔案傳遞至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定的 Amazon SNS Notifications CloudTrail](#)
- [從多個區域接收 CloudTrail 日誌檔案](#)，並從 [多個帳戶接收 CloudTrail 日誌檔案](#)

所有 Amazon ElastiCache 動作都會由記錄，CloudTrail 並記錄在 [ElastiCache API 參考](#) 中。例如，呼叫 CreateCacheCluster，DescribeCacheCluster 而 ModifyCacheCluster 動作會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是使用根憑證還是 IAM 使用者憑證提出的。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。

- 該請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱[CloudTrail userIdentity](#)元素。

## 了解 Amazon ElastiCache 日誌檔案項目

追蹤是一種組態，可讓您將事件作為日誌檔案交付至您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌項目。事件代表來自任何來源的單一請求，並包含所請求動作、動作日期和時間、請求參數等資訊。CloudTrail log 檔案不是公開API呼叫的有序堆疊追蹤，因此它們不會以任何特定順序顯示。

下列範例顯示示範 CreateCacheCluster動作的 CloudTrail 日誌項目。

```
{
 "eventVersion": "1.01",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "EXAMPLEEXAMPLEEXAMPLE",
 "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "userName": "elasticache-allow"
 },
 "eventTime": "2014-12-01T22:00:35Z",
 "eventSource": "elasticache.amazonaws.com",
 "eventName": "CreateCacheCluster",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.0.2.01",
 "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
 "requestParameters": {
 "numCacheNodes": 2,
 "cacheClusterId": "test-memcached",
 "engine": "memcached",
 "aZMode": "cross-az",
 "cacheNodeType": "cache.m1.small",
 },
 "responseElements": {
 "engine": "memcached",
 "clientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
 "cacheParameterGroup": {
```

```

 "cacheParameterGroupName":"default.memcached1.4",
 "cacheNodeIdsToReboot":{
 },
 "parameterApplyStatus":"in-sync"
 },
 "preferredAvailabilityZone":"Multiple",
 "numCacheNodes":2,
 "cacheNodeType":"cache.m1.small",

 "cacheClusterStatus":"creating",
 "autoMinorVersionUpgrade":true,
 "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
 "cacheClusterId":"test-memcached",
 "engineVersion":"1.4.14",
 "cacheSecurityGroups":[
 {
 "status":"active",
 "cacheSecurityGroupName":"default"
 }
],
 "pendingModifiedValues":{
 }
},
"requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
"eventID":"92762127-7a68-42ce-8787-927d2174cde1"
}

```

下列範例顯示示範 DescribeCacheCluster 動作的 CloudTrail 日誌項目。請注意，對於所有 Amazon ElastiCache Describe 呼叫 ( Describe\* )，ResponseElements 區段會移除並顯示為 null。

```

{
 "eventVersion":"1.01",
 "userIdentity":{
 "type":"IAMUser",
 "principalId":"EXAMPLEEXAMPLEEXAMPLE",
 "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
 "accountId":"123456789012",
 "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
 "userName":"elasticache-allow"
 },
 "eventTime":"2014-12-01T22:01:00Z",
 "eventSource":"elasticache.amazonaws.com",

```

```

 "eventName": "DescribeCacheClusters",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.0.2.01",
 "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
 "requestParameters": {
 "showCacheNodeInfo": false,
 "maxRecords": 100
 },
 "responseElements": null,
 "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
 "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
 }
}

```

下列範例顯示記錄ModifyCacheCluster動作的 CloudTrail 日誌項目。

```

{
 "eventVersion": "1.01",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "EXAMPLEEXAMPLEEXAMPLE",
 "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "userName": "elasticache-allow"
 },
 "eventTime": "2014-12-01T22:32:21Z",
 "eventSource": "elasticache.amazonaws.com",
 "eventName": "ModifyCacheCluster",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.0.2.01",
 "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
 "requestParameters": {
 "applyImmediately": true,
 "numCacheNodes": 3,
 "cacheClusterId": "test-memcached"
 },
 "responseElements": {
 "engine": "memcached",
 "clientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
 "cacheParameterGroup": {
 "cacheParameterGroupName": "default.memcached1.4",
 "cacheNodeIdsToReboot": {

```

```
 },
 "parameterApplyStatus":"in-sync"
 },
 "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
 "preferredAvailabilityZone":"Multiple",
 "numCacheNodes":2,
 "cacheNodeType":"cache.m1.small",
 "cacheClusterStatus":"modifying",
 "autoMinorVersionUpgrade":true,
 "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
 "cacheClusterId":"test-memcached",
 "engineVersion":"1.4.14",
 "cacheSecurityGroups":[
 {
 "status":"active",
 "cacheSecurityGroupName":"default"
 }
],
 "configurationEndpoint":{
 "address":"test-memcached.example.cfg.use1prod.cache.amazonaws.com",
 "port":11211
 },
 "pendingModifiedValues":{
 "numCacheNodes":3
 }
},
"requestID":"807f4bc3-354c-11e4-9376-c1d979ba565a",
"eventID":"e9163565-376f-4223-96e9-9f50528da645"
}
```

## Amazon ElastiCache 事件SNS監控

當叢集發生重大事件時，ElastiCache 會將通知傳送至特定 Amazon SNS主題。範例包含新增節點失敗、新增節點成功、安全群組修改和其他事件。藉由監控重要事件，您可以了解叢集目前的狀態，並根據事件採取正確的動作。

### 主題

- [管理 ElastiCache Amazon SNS通知](#)
- [檢視 ElastiCache 事件](#)
- [事件通知和 Amazon SNS](#)

## 管理 ElastiCache Amazon SNS通知

您可以設定 ElastiCache 使用 Amazon Simple Notification Service ( Amazon ) 傳送重要叢集事件的通知SNS。在這些範例中，您將設定具有 Amazon SNS主題 Amazon Resource Name ( ARN ) 的叢集，以接收通知。

### Note

- 本主題假設您已註冊 Amazon SNS 並已設定和訂閱 Amazon SNS主題。如需操作方式的相關資訊，請參閱 [Amazon Simple Notification Service 開發人員指南](#)。
- 根據預設，API `modify-replication-group` 會影響區域中的所有群組，而不僅僅是目前指定的群組。如果您想要在區域中設定與其他群組不同的特定群組，您可以使用 `--notification-topic-arn` 選項來為該群組建立個別主題。

## 新增 Amazon SNS主題

下列各節說明如何使用 AWS 主控台、AWS CLI 或新增 Amazon SNS主題 ElastiCache API。

### 新增 Amazon SNS主題 ( 主控台 )

下列程序說明如何為叢集新增 Amazon SNS主題。當使用 Valkey 或 Redis OSS 在步驟 2 中為複寫群組新增 Amazon SNS主題時，而不是選擇叢集，請選擇複寫群組。然後遵循相同的剩餘步驟。

### Note

此程序也可用於修改 Amazon SNS主題。

### 為叢集新增或修改 Amazon SNS主題 ( 主控台 )

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在叢集中，選擇要為其新增或修改 Amazon SNS主題 的叢集ARN。
3. 選擇 Modify (修改)。
4. 在通知主題 下的修改叢集中，選擇您要新增SNS的主題，或選擇手動輸入ARN並輸入 Amazon SNS主題ARN的。 SNS

## 5. 選擇 Modify (修改)。

### 新增 Amazon SNS主題 ( AWS CLI )

若要新增或修改叢集的 Amazon SNS主題，請使用 AWS CLI 命令 `modify-cache-cluster`。

下列程式碼範例會將 Amazon SNS主題加入 `my-cluster`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-cluster \
 --notification-topic-arn arn:aws:sns:us-west-2:123456789xxx:ElastiCacheNotifications
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-cluster ^
 --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

如需詳細資訊，請參閱 [modify-cache-cluster](#)。

### 新增 Amazon SNS主題 ( ElastiCache API )

若要新增或修改叢集的 Amazon SNS主題，請使用下列參數呼叫 `ModifyCacheCluster` 動作：

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

### Example

```
https://elasticache.amazon.com/
 ?Action=ModifyCacheCluster
 &ApplyImmediately=false
 &CacheClusterId=my-cluster
 &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications
 &Version=2014-12-01
```



```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyCacheCluster](#)。

## 啟用和停用 Amazon SNS通知

您可以為叢集開啟或關閉通知。下列程序說明如何停用 Amazon SNS通知。

### 啟用和停用 Amazon SNS通知 ( 主控台 )

#### 使用 停用 Amazon SNS通知 AWS Management Console

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看執行 Memcached 的叢集清單，請在導覽窗格中選擇 Memcached。  
若要查看執行 Valkey 或 Redis 的叢集清單OSS，請在導覽窗格中選擇 Valkey 或 Redis OSS。
3. 選擇您要修改通知之叢集左邊的方塊。
4. 選擇 Modify (修改)。
5. 在通知主題 下的修改叢集中，選擇停用通知。 SNS
6. 選擇 Modify (修改)。

### 啟用和停用 Amazon SNS通知 ( AWS CLI )

若要停用 Amazon SNS通知，請使用 命令 `modify-cache-cluster` 搭配下列參數：

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-cluster \
 --notification-topic-status inactive
```

## 針對 Windows :

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-cluster ^
 --notification-topic-status inactive
```

### 啟用和停用 Amazon SNS通知 (ElastiCache API)

若要停用 Amazon SNS通知，請使用下列參數呼叫 `ModifyCacheCluster` 動作：

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

此呼叫會傳回類似以下的輸出：

### Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheCluster
&ApplyImmediately=false
&CacheClusterId=my-cluster
&NotificationTopicStatus=inactive
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

## 檢視 ElastiCache 事件

ElastiCache 會記錄與您的叢集執行個體、安全群組和參數群組相關的事件。此資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的描述。您可以使用 ElastiCache 主控台、AWS CLI `describe-events` 命令或 ElastiCache API 動作 `DescribeEvents` 輕鬆從日誌擷取事件。

下列程序說明如何檢視過去 24 小時（1440 分鐘）的所有 ElastiCache 事件。

### 檢視 ElastiCache 事件（主控台）

下列程序使用 ElastiCache 主控台顯示事件。

#### 使用 ElastiCache 主控台檢視事件

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。

在事件畫面上，清單的每一列代表一個事件，並顯示事件來源、事件類型（`cache-cluster` `cache-parameter-group` `cache-security-group`、或 `cache-subnet-group`）、事件 GMT 時間，以及事件的描述。

您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

### 檢視 ElastiCache 事件（AWS CLI）

若要使用產生 ElastiCache 事件清單 AWS CLI，請使用命令 `describe-events`。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個快取叢集事件。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內的所有事件。

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

`describe-events` 命令的輸出會類似下列內容。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

```
{
 "Events": [
 {
 "SourceIdentifier": "my-mem-cluster",
 "SourceType": "cache-cluster",
 "Message": "Finished modifying number of nodes from 1 to 3",
 "Date": "2020-06-09T02:01:21.772Z"
 },
 {
 "SourceIdentifier": "my-mem-cluster",
 "SourceType": "cache-cluster",
 "Message": "Added cache node 0002 in availability zone us-west-2a",
 "Date": "2020-06-09T02:01:21.716Z"
 },
 {
 "SourceIdentifier": "my-mem-cluster",
 "SourceType": "cache-cluster",
 "Message": "Added cache node 0003 in availability zone us-west-2a",
 "Date": "2020-06-09T02:01:21.706Z"
 },
 {
 "SourceIdentifier": "my-mem-cluster",
 "SourceType": "cache-cluster",
 "Message": "Increasing number of requested nodes",
 "Date": "2020-06-09T01:58:34.178Z"
 },
 {
 "SourceIdentifier": "mycluster-0003-004",
 "SourceType": "cache-cluster",
 "Message": "Added cache node 0001 in availability zone us-west-2c",
 "Date": "2020-06-09T01:51:14.120Z"
 },
 {
 "SourceIdentifier": "mycluster-0003-004",
 "SourceType": "cache-cluster",
 "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
 "Date": "2020-06-09T01:51:14.095Z"
 },
 {
 "SourceIdentifier": "mycluster-0003-004",
 "SourceType": "cache-cluster",
 "Message": "Cache cluster created",
 "Date": "2020-06-09T01:51:14.094Z"
 }
]
}
```

```
 },
 {
 "SourceIdentifier": "mycluster-0001-005",
 "SourceType": "cache-cluster",
 "Message": "Added cache node 0001 in availability zone us-west-2b",
 "Date": "2020-06-09T01:42:55.603Z"
 },
 {
 "SourceIdentifier": "mycluster-0001-005",
 "SourceType": "cache-cluster",
 "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
 "Date": "2020-06-09T01:42:55.576Z"
 },
 {
 "SourceIdentifier": "mycluster-0001-005",
 "SourceType": "cache-cluster",
 "Message": "Cache cluster created",
 "Date": "2020-06-09T01:42:55.574Z"
 },
 {
 "SourceIdentifier": "mycluster-0001-004",
 "SourceType": "cache-cluster",
 "Message": "Added cache node 0001 in availability zone us-west-2b",
 "Date": "2020-06-09T01:28:40.798Z"
 },
 {
 "SourceIdentifier": "mycluster-0001-004",
 "SourceType": "cache-cluster",
 "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
 "Date": "2020-06-09T01:28:40.775Z"
 },
 {
 "SourceIdentifier": "mycluster-0001-004",
 "SourceType": "cache-cluster",
 "Message": "Cache cluster created",
 "Date": "2020-06-09T01:28:40.773Z"
 }
]
}
```

如需可用參數和允許參數值這類項目的詳細資訊，請參閱 [describe-events](#)。

## 檢視 ElastiCache 事件 (ElastiCache API)

若要使用產生 ElastiCache 事件清單 ElastiCache API，請使用 DescribeEvents 動作。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼會列出 40 個最新的快取叢集事件。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeEvents
&MaxRecords=40
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SourceType=cache-cluster
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內的快取叢集事件。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeEvents
&Duration=1440
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SourceType=cache-cluster
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

上述動作產生的輸出應該會類似下列內容。

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">
 <DescribeEventsResult>
 <Events>
 <Event>
 <Message>Cache cluster created</Message>
 <SourceType>cache-cluster</SourceType>
 <Date>2015-02-02T18:22:18.202Z</Date>
 <SourceIdentifier>mem01</SourceIdentifier>
 </Event>

 (...output omitted...)
```

```
</Events>
</DescribeEventsResult>
<ResponseMetadata>
 <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
</ResponseMetadata>
</DescribeEventsResponse>
```

如需可用參數和允許參數值這類項目的詳細資訊，請參閱 [DescribeEvents](#)。

## 事件通知和 Amazon SNS

ElastiCache 當快取叢集上發生重大事件時，可以使用 Amazon Simple Notification Service ( SNS ) 發佈訊息。此功能可用於重新整理用戶端機器上的伺服器清單，這些機器連線至快取叢集的個別快取節點端點。

### Note

如需 Amazon Simple Notification Service ( SNS ) 的詳細資訊，包括定價和 Amazon SNS 文件連結的相關資訊，請參閱 [Amazon SNS 產品頁面](#)。

通知會發佈至指定的 Amazon SNS 主題。下列是通知的需求：

- 只能為 ElastiCache 通知設定一個主題。
- 擁有 Amazon SNS 主題 AWS 的帳戶，必須與擁有啟用通知之快取叢集的帳戶相同。
- 您發佈的 Amazon SNS 主題無法加密。

### Note

可以將加密（靜態）Amazon SNS 主題連接至叢集。不過，主題與 ElastiCache 主控台的狀態會顯示為非作用中，當將訊息 ElastiCache 推送至主題時，這會有效地取消主題與叢集的關聯。


- Amazon SNS 主題必須與 ElastiCache 叢集位於相同的區域。

## ElastiCache 事件

下列 ElastiCache 事件會觸發 Amazon SNS 通知。如需事件的詳細資訊，請參閱 [檢視 ElastiCache 事件](#)。

事件名稱	訊息	描述
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	快取節點已新增至快取叢集，且已就緒可供使用。
ElastiCache : AddCacheNodeFailed 由於可用 IP 地址不足	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	由於可用的 IP 地址不足，而無法新增快取節點。
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	一或多個快取叢集參數已變更。
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	快取叢集的佈建已完成，且快取叢集中的快取節點已就緒並可供使用。
ElastiCache : CacheClusterProvisioningFailed 由於網路狀態不相容	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	已嘗試將新的快取叢集啟動到不存在的虛擬私有雲端 ( VPC )。
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	快取叢集的擴展已成功完成。
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	快取叢集上的擴展操作失敗。
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	已發生下列其中一項事件： <ul style="list-style-type: none"> <li>• 授權給快取叢集的快取安全群組清單已修改。</li> <li>•</li> </ul>



事件名稱	訊息	描述
		<p>與快取EC2叢集相關聯的任何快取安全群組上已授權一或多個新的安全群組。</p> <ul style="list-style-type: none"> <li>一或多個EC2安全群組已從與快取叢集相關聯的任何快取安全群組中撤銷。</li> </ul>
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機降級或無法連線，並已開始取代快取節點。</p> <div data-bbox="1068 772 1507 993" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>取代快取節點DNS的項目不會變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。不過，即使 ElastiCache 已取代快取節點，某些快取用戶端程式庫仍可能停止使用快取節點；在此情況下，應用程式應在發生此事件時重新整理伺服器清單。</p>

事件名稱	訊息	描述
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機降級或無法連線，且已完成取代快取節點。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b> 取代快取節點DNS的項目不會變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。不過，即使 ElastiCache 已取代快取節點，某些快取用戶端程式庫仍可能停止使用快取節點；在此情況下，應用程式應在發生此事件時重新整理伺服器清單。</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>一或多個快取節點已重新開機。</p> <p>訊息 (Memcached) : "Cache node %s shutdown" 然後，第二個訊息 : "Cache node %s restarted"</p>
ElastiCache : CertificateRenewalComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:CertificateRenewalComplete	Amazon CA 憑證已成功續約。
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	已成功建立複寫群組。

事件名稱	訊息	描述
ElastiCache:DeleteCacheClusterComplete	ElastiCache:DeleteCacheClusterComplete : <i>cluster-name</i>	已完成刪除快取叢集和所有與其相關聯的快取節點。
ElastiCache : FailoverComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:FailoverComplete : <i>mycluster</i>	已成功容錯移轉至複本節點。
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	叢集中的複本數量已增加。
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	將複本新增至叢集的程序已經開始。
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	叢集中原先已排程替換的節點，已不再排程替換。
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>叢集中原先已排程替換的節點，已重新排程在通知中所述的新視窗期間進行替換。</p> <p>如需您可以採取哪些動作的資訊，請參閱<a href="#">更換節點 ( Valkey 和 RedisOSS )</a>。</p>

事件名稱	訊息	描述
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	您叢集中的節點，已排程在通知中所述的視窗期間進行替換。  如需您可以採取哪些動作的資訊，請參閱 <a href="#">更換節點 ( Valkey 和 RedisOSS )</a> 。
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	快取節點已從快取叢集移除。
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	已順利完成複寫群組上的向上擴展操作。
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	複寫群組上的擴展操作失敗。
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	節點有可用的自助更新。
ElastiCache : SnapshotComplete ( OSS僅限 Valkey 或 Redis )	ElastiCache:SnapshotComplete : <i>cluster-name</i>	快取快照已順利完成。
ElastiCache : SnapshotFailed ( OSS僅限 Valkey 或 Redis )	SnapshotFailed : <i>cluster-name</i>	快取快照已失敗。請參閱叢集的快取事件，以取得失敗原因的詳細資訊。  如果您描述快照，請查看 <a href="#">DescribeSnapshots</a> ，其狀態將是 failed。

## 相關主題

- [檢視 ElastiCache 事件](#)

## 日誌傳送

### Note

Valkey 7.x 和更新版本支援慢速日誌，以及使用引擎 6.0 版更新版本的 Redis OSS快取叢集和複寫群組。

Valkey 7.x 和 Redis OSS快取叢集和使用引擎 6.2 版以後的複寫群組支援引擎日誌。

日誌交付可讓您將 [SLOWLOG](#)或引擎日誌串流至兩個目的地之一：

- Amazon Data Firehose
- Amazon CloudWatch Logs

當您使用 [建立或修改叢集](#)時，您可以啟用和設定日誌交付 ElastiCache APIs。每個日誌項目都會以兩種格式之一傳送至指定的目的地：JSON或TEXT。

定期從引擎擷取固定數量的慢速日誌項目。根據為引擎參數slowlog-max-len 指定的值，可能無法將其他慢速日誌項目傳送至目的地。

您可以隨時使用 AWS 主控台或其中一個修改 或 APIs，選擇變更交付組態[modify-cache-cluster](#)或停用日誌交付[modify-replication-group](#)。

您必須設定 apply-immediately 參數以進行所有日誌傳送修改。

### Note

啟用日誌交付時，即使日誌直接交付至 Amazon Data Firehose，也會收取 Amazon CloudWatch Logs 費用。如需詳細資訊，請參閱 [Amazon CloudWatch Pricing](#) 中的已終明日誌一節。

## 慢速日誌項目的內容

慢速日誌包含下列資訊：

- CacheClusterId – 快取叢集的 ID
- CacheNodeId – 快取節點的 ID
- Id - 每個慢速日誌項目的唯一漸進式識別碼
- Timestamp - 處理記錄命令的 Unix 時間戳記
- Duration - 執行所需的時間長度，以微秒為單位
- Command - 用戶端使用的命令。例如，`set foo bar`其中 `foo`是金鑰，而 `bar`是 `value`。會使用 ElastiCache 取代實際的金鑰名稱和值(2 more arguments)，以避免敏感資料暴露。
- ClientAddress – 用戶端 IP 地址和連接埠
- ClientName – 如果透過 `CLIENT SETNAME`命令設定 用戶端名稱

## 引擎日誌項目的內容

ElastiCache 引擎日誌包含下列資訊：

- CacheClusterId – 快取叢集的 ID
- CacheNodeId – 快取節點的 ID
- 日誌層級 – LogLevel 可以下列其中一項：VERBOSE("-")、NOTICE("\*")、WARNING("#")。
- 時間 – 記錄訊息UTC的時間。時間格式如下："DD MMM YYYY hh:mm:ss.ms UTC"
- Role – 從中發出日誌之節點的角色。它可以是下列其中一項：主要程式的「M」、複本的「S」、在 RDB/AOF 或前哨程式的「X」上工作的寫入器子程序的「C」。
- 訊息 – 引擎日誌訊息。

## 設定記錄功能的許可

您需要在IAM使用者/角色政策中包含下列IAM許可：

- `logs:CreateLogDelivery`
- `logs:UpdateLogDelivery`
- `logs>DeleteLogDelivery`
- `logs:GetLogDelivery`

- `logs:ListLogDeliveries`

如需許可和政策的詳細資訊，請參閱[存取管理概觀：許可與政策](#)。

## 日誌類型和日誌格式規格

### 慢速日誌

慢速日誌同時支援 JSON 和 TEXT

下列顯示 JSON 格式範例：

```
{
 "CacheClusterId": "logslowxxxxmsxj",
 "CacheNodeId": "0001",
 "Id": 296,
 "Timestamp": 1605631822,
 "Duration (us)": 0,
 "Command": "GET ... (1 more arguments)",
 "ClientAddress": "192.168.12.104:55452",
 "ClientName": "logslowxxxxmsxj##"
}
```

下列顯示 TEXT 格式範例：

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

### 引擎日誌

引擎日誌同時支援 JSON 和 TEXT

下列顯示 JSON 格式範例：

```
{
 "CacheClusterId": "xxxxxxxxzy-engine-log-test",
 "CacheNodeId": "0001",
 "LogLevel": "VERBOSE",
 "Role": "M",
 "Time": "12 Nov 2020 01:28:57.994 UTC",
 "Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.
Check back later"
```

```
}
```

下列顯示TEXT格式範例：

```
xxxxxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua script detected that is still in execution after 10000 milliseconds.
```

## ElastiCache 記錄目的地

本節說明您可以為 ElastiCache 日誌選擇的記錄目的地。每節都會提供為目的地類型設定記錄的指引，以及目的地類型專屬行為的相關資訊。設定記錄目的地之後，您可以將其規格提供給 ElastiCache 記錄組態，以開始記錄目的地。

### 主題

- [Amazon CloudWatch Logs](#)
- [Amazon Data Firehose](#)

## Amazon CloudWatch Logs

- 您可以指定將交付日誌的 CloudWatch Logs 日誌群組。
- 來自多個 Valkey 或 Redis OSS叢集和複寫群組的日誌可以交付至相同的日誌群組。
- 系統會為快取叢集或複寫群組中的每個節點建立新的記錄串流，並將日誌傳送至個別的記錄串流。日誌串流名稱將使用以下格式：`elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

### 將日誌發佈至 CloudWatch 日誌的許可

您必須具有下列許可設定，才能設定 ElastiCache 將日誌傳送至 CloudWatch Logs 日誌群組：

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "logs:CreateLogDelivery",
 "logs:GetLogDelivery",
```



```

 "logs:UpdateLogDelivery",
 "logs>DeleteLogDelivery",
 "logs>ListLogDeliveries"
],
 "Resource": [
 "*"
],
 "Effect": "Allow",
 "Sid": "ElastiCacheLogging"
},
{
 "Sid": "ElastiCacheLoggingCWL",
 "Action": [
 "logs:PutResourcePolicy",
 "logs:DescribeResourcePolicies",
 "logs:DescribeLogGroups"
],
 "Resource": [
 "*"
],
 "Effect": "Allow"
}
]
}

```

如需詳細資訊，請參閱[傳送至日誌的 CloudWatch 日誌](#)。

## Amazon Data Firehose

- 您可以指定要交付日誌的 Firehose 交付串流。
- 來自多個 Valkey 或 Redis OSS 叢集和複寫群組的日誌可以交付至相同的交付串流。
- 來自同一個快取叢集或複寫群組內每個節點的日誌都會傳送至相同的交付串流。您可以根據每個日誌訊息中包含的 `cache-cluster-id` 和 `cache-node-id` 來區分不同快取節點的日誌訊息。
- 目前亞太區域（大阪）區域不提供傳送至 Firehose 的日誌。

### 將日誌發佈至 Firehose 的許可

您必須具有下列許可，才能設定 ElastiCache 將日誌傳送至 Amazon Kinesis Data Firehose 交付串流。

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "logs:CreateLogDelivery",
 "logs:GetLogDelivery",
 "logs:UpdateLogDelivery",
 "logs>DeleteLogDelivery",
 "logs:ListLogDeliveries"
],
 "Resource": [
 "*"
],
 "Effect": "Allow",
 "Sid": "ElastiCacheLogging"
 },
 {
 "Sid": "ElastiCacheLoggingFHSLR",
 "Action": [
 "iam:CreateServiceLinkedRole"
],
 "Resource": "*",
 "Effect": "Allow"
 },
 {
 "Sid": "ElastiCacheLoggingFH",
 "Action": [
 "firehose:TagDeliveryStream"
],
 "Resource": "Amazon Kinesis Data Firehose delivery stream ARN",
 "Effect": "Allow"
 }
]
}

```

## 使用主控台指定日誌傳送

使用 AWS Management Console，您可以依照的步驟建立 Valkey 或 Redis OSS (停用叢集模式) 叢集，[建立 Valkey \(停用叢集模式\) 叢集 \(主控台\)](#) 或使用的步驟建立 Valkey 或 Redis OSS (啟用叢集模式) 叢集 [建立 Valkey 或 Redis OSS \(啟用叢集模式\) 叢集 \(主控台\)](#)。在任一種情況下，您都可以執行下列動作來設定日誌傳送；

1. 在進階設定下，選擇日誌，然後檢查慢速日誌或引擎日誌。
2. 在日誌格式下，選擇文字或 JSON。
3. 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
4. 在日誌目的地下，選擇建立新，並輸入您的 Amazon S3 儲存貯體名稱、CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱，或選擇選擇現有，然後選擇您的 CloudWatch Logs 群組名稱或 Kinesis Data Firehose 串流名稱。

修改叢集時：

您可以選擇啟用/停用日誌傳送或變更目的地類型、格式或目的地：

1. 登入主控台，並在開啟 ElastiCache 主控台<https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格中，選擇 Valkey 叢集或 Redis OSS叢集。
3. 從叢集清單中選擇您要修改的叢集。請選擇叢集名稱而不是名稱旁邊的核取方塊。
4. 在 Cluster name (叢集名稱) 頁面上，選擇 Logs (日誌) 索引標籤。
5. 若要啟用/停用慢速日誌，請選擇 Enable slow logs (啟用慢速日誌) 或 Disable slow logs (停用慢速日誌)。
6. 若要啟用/停用引擎日誌，請選擇 Enable engine logs (啟用引擎日誌) 或 Disable engine logs (停用引擎日誌)。
7. 若要變更您的組態，請選擇 Modify slow logs (修改慢速日誌) 或 Modify engine logs (修改引擎日誌)：
  - 在目的地類型下，選擇 CloudWatch Logs 或 Kinesis Firehose。
  - 在日誌目的地下，選擇建立新，然後輸入您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。或選擇選取現有，然後選擇您的 CloudWatchLogs 日誌群組名稱或 Kinesis Data Firehose 串流名稱。

## 使用 指定日誌交付 AWS CLI

慢速日誌

建立複寫群組，以慢速將日誌傳送至 CloudWatch Logs。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
```

```
--replication-group-id test-slow-log \
--replication-group-description test-slow-log \
--engine redis \
--cache-node-type cache.r5.large \
--num-cache-clusters 2 \
--log-delivery-configurations '{
 "LogType":"slow-log",
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{
 "CloudWatchLogsDetails":{
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
}'
```

針對 Windows :

```
aws elasticache create-replication-group ^
--replication-group-id test-slow-log ^
--replication-group-description test-slow-log ^
--engine redis ^
--cache-node-type cache.r5.large ^
--num-cache-clusters 2 ^
--log-delivery-configurations '{
 "LogType":"slow-log",
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{
 "CloudWatchLogsDetails":{
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
}'
```

修改複寫群組以將慢速日誌交付至 CloudWatch 日誌

若為 Linux、macOS 或 Unix :

```
aws elasticache modify-replication-group \
--replication-group-id test-slow-log \
--apply-immediately \
--log-delivery-configurations '
```

```
{
 "LogType":"slow-log",
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{"
 "CloudWatchLogsDetails":{"
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
}'
```

針對 Windows :

```
aws elasticache modify-replication-group ^
--replication-group-id test-slow-log ^
--apply-immediately ^
--log-delivery-configurations '
{
 "LogType":"slow-log",
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{"
 "CloudWatchLogsDetails":{"
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
}'
```

修改複寫群組以停用緩慢日誌傳送

若為 Linux、macOS 或 Unix :

```
aws elasticache modify-replication-group \
--replication-group-id test-slow-log \
--apply-immediately \
--log-delivery-configurations '
{
 "LogType":"slow-log",
 "Enabled":false
}'
```

針對 Windows :

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
 {
 "LogType":"slow-log",
 "Enabled":false
 }'
```

## 引擎日誌

建立複寫群組，並將引擎日誌傳送至 CloudWatch Logs。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-replication-group \
 --replication-group-id test-slow-log \
 --replication-group-description test-slow-log \
 --engine redis \
 --cache-node-type cache.r5.large \
 --num-cache-clusters 2 \
 --log-delivery-configurations '{
 "LogType":"engine-log",
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{
 "CloudWatchLogsDetails":{
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
 }'
```

針對 Windows：

```
aws elasticache create-replication-group ^
 --replication-group-id test-slow-log ^
 --replication-group-description test-slow-log ^
 --engine redis ^
 --cache-node-type cache.r5.large ^
 --num-cache-clusters 2 ^
 --log-delivery-configurations '{
 "LogType":"engine-log",
```

```
 "DestinationType":"cloudwatch-logs",
 "DestinationDetails":{
 "CloudWatchLogsDetails":{
 "LogGroup":"my-log-group"
 }
 },
 "LogFormat":"json"
 }'
```

修改複寫群組，將引擎日誌交付至 Firehose

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
{
 "LogType":"engine-log",
 "DestinationType":"kinesis-firehose",
 "DestinationDetails":{
 "KinesisFirehoseDetails":{
 "DeliveryStream":"test"
 }
 },
 "LogFormat":"json"
}'
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
{
 "LogType":"engine-log",
 "DestinationType":"kinesis-firehose",
 "DestinationDetails":{
 "KinesisFirehoseDetails":{
 "DeliveryStream":"test"
 }
 },
 "LogFormat":"json"
}'
```

```
}'
```

## 修改複寫群組以切換為引擎格式

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
 {
 "LogType":"engine-log",
 "LogFormat":"json"
 }'
```

針對 Windows：

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
 {
 "LogType":"engine-log",
 "LogFormat":"json"
 }'
```

## 修改複寫群組以停用引擎日誌傳送

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
 {
 "LogType":"engine-log",
 "Enabled":false
 }'
```

針對 Windows：

```
aws elasticache modify-replication-group ^
```



```
--replication-group-id test-slow-log ^
--apply-immediately ^
--log-delivery-configurations '
{
 "LogType":"engine-log",
 "Enabled":false
}'
```

## 監控與 CloudWatch 指標搭配使用

ElastiCache 提供指標，可讓您監控叢集。您可以透過存取這些指標 CloudWatch。如需的詳細資訊 CloudWatch，請參閱 [CloudWatch 文件](#)。

ElastiCache 同時提供主機層級指標（例如，CPU 用量）和快取引擎軟體特有的指標（例如，快取取得和快取遺失）。每隔 60 秒會針對每個快取節點測量及發佈這些指標。

### Important

您應該考慮在特定關鍵指標上設定 CloudWatch 警示，以便在快取叢集效能開始下降時收到通知。如需詳細資訊，請參閱本指南中的 [應監控哪些指標？](#)。

### 主題

- [主機層級指標](#)
- [Valkey 和 Redis 的指標 OSS](#)
- [Memcached 的指標](#)
- [應監控哪些指標？](#)
- [選擇指標統計資料與期間](#)
- [監控 CloudWatch 叢集和節點指標](#)

## 主機層級指標

AWS/ElastiCache 命名空間包含下列個別快取節點的主機層級指標。每隔 60 秒會針對每個快取節點測量及發佈這些指標。

另請參閱

- [Valkey 和 Redis 的指標 OSS](#)

指標	描述	單位
CPUUtilization	整個主機的使用CPU率百分比。由於 Valkey 和 Redis OSS是單執行緒，我們建議您監控具有 4 個或更多的節點的EngineCPUUtilization 指標vCPUs。	百分比
CPUCreditBalance	<p>執行個體自啟動或啟動以來累積的累積CPU點數。對於 T2 標準，CPUCreditBalance也包含已累積的啟動點數。</p> <p>獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。額度餘額有最大值限制，它取決於執行個體大小。到達限制之後，任何獲得的新額度都會遭到捨棄。如果是 T2 Standard，啟動額度不會計入此限制。</p> <p>中的點數CPUCreditBalance可供執行個體用於超出其基準CPU使用率的爆量。</p> <p>CPU 點數指標僅以五分鐘的頻率提供。</p> <p>這些指標不適用於 T2 高載效能執行個體。</p>	額度 ( v CPU 分鐘 )
CPUCreditUsage	<p>執行個體為CPU使用而花費的CPU點數。一個CPU點數等於一個 vCPU 在一分鐘內以 100% 使用率執行，或相等的 vCPUs、使用率和時間組合 ( 例如一個 vCPU 在兩分鐘內以 50% 使用率執行，或兩個 v 在兩分鐘內以 25% 使用率 vCPUs 執行 )。</p> <p>CPU 點數指標僅以五分鐘的頻率提供。如果您要指定大於 5 分鐘的期間，請使用總和統計數字代替平均統計數字。</p>	額度 ( v CPU 分鐘 )

指標	描述	單位
	這些指標不適用於 T2 高載效能執行個體。	
FreeableMemory	主機上可用的記憶體數量。這是衍生自作業系統報告為可自由使用的 RAM、緩衝區和快取。	位元組
NetworkBytesIn	主機已從網路讀取的位元組數。	位元組
NetworkBytesOut	執行個體在所有網路界面上送出的位元組數目。	位元組
NetworkPacketsIn	執行個體在所有網路界面上收到的封包數目。此指標識別單一執行個體上的傳入流量 (封包數目)。	計數
NetworkPacketsOut	執行個體在所有網路界面上送出的封包數目。此指標識別單一執行個體上的傳出流量 (封包數目)。	計數
NetworkBandwidthInAllowanceExceeded	因傳入的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。	計數
NetworkConntrackAllowanceExceeded	因為連線追蹤超過執行個體的上限且無法建立新的連線，而丟棄的封包數目。這可能會導致傳送或傳回執行個體流量的封包遺失。	計數
NetworkBandwidthOutAllowanceExceeded	因傳出的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。	計數
NetworkPacketsPerSecondAllowanceExceeded	因雙向每秒封包數超過執行個體的上限而排入佇列或丟棄的封包數目。	計數
NetworkMaxBytesIn	每分鐘接收位元組的每秒爆量上限。	位元組
NetworkMaxBytesOut	每分鐘傳輸位元組的每秒爆量上限。	位元組
NetworkMaxPacketsIn	每分鐘每秒的爆量接收封包上限。	計數
NetworkMaxPacketsOut	每分鐘傳輸封包的每秒爆量上限。	計數

指標	描述	單位
SwapUsage	主機已使用的交換空間的量。	位元組

## Valkey 和 Redis 的指標 OSS

Amazon ElastiCache 命名空間包含下列 Valkey 和 Redis OSS 指標。使用 Valkey 引擎時，這些指標相同。

除了 ReplicationLag 和 EngineCPUUtilization 之外，這些指標是從 info 命令衍生而來。每個指標都是在快取模式層級進行計算。

如需 info 命令的完整文件，請參閱 <https://http://valkey.io/commands/info>。

另請參閱

- [主機層級指標](#)

指標	描述	單位
ActiveDefragHits	作用中重組程序每分鐘執行的值重新配置次數。這是衍生自的 active_defrag_hits 統計資料 <a href="#">INFO</a> 。	Number
AuthenticationFailures	OSS 使用 AUTH 命令向 Valkey 或 Redis 進行身分驗證的失敗嘗試總數。您可以使用 <a href="#">ACLLOG</a> 命令來尋找個別身分驗證失敗的詳細資訊。建議對此設定警示，以偵測未經授權的存取嘗試。	計數
	由 Valkey 或 Redis 配置 OSS 用於所有目的的位元組總數，包括資料集、緩衝區等。	位元組
BytesUsedForCache	Dimension: Tier=Memory 適用於使用的 Valkey 或 Redis OSS 叢集中的資料分層 <a href="#">ElastiCache</a> ：用於記憶體快取的位元組總數。這是的 used_memory 統計值 <a href="#">INFO</a> 。	位元組

指標	描述	單位
	Dimension: Tier=SSD 對於使用的 Valkey 或 Redis OSS 叢集中的資料分層 <a href="#">ElastiCache</a> : 用於快取的位元組總數 SSD。	位元組
BytesReadFromDisk	每分鐘從磁碟讀取的位元組總數。僅支援使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。	位元組
BytesWrittenToDisk	每分鐘寫入磁碟的位元組總數。僅支援使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。	位元組
CacheHits	主字典中的成功唯讀索引鍵查詢次數。這是衍生自的 <code>keyspace_hits</code> 統計資料 <a href="#">INFO</a> 。	計數
CacheMisses	主字典中的未成功唯讀索引鍵查詢次數。這是衍生自的 <code>keyspace_misses</code> 統計資料 <a href="#">INFO</a> 。	計數
CommandAuthorizationFailures	使用者嘗試執行他們沒有呼叫許可的命令失敗總次數。您可以使用 <a href="#">ACLOG</a> 命令找到個別身分驗證失敗的詳細資訊。建議對此設定警示，以偵測未經授權的存取嘗試。	計數
CacheHitRate	指示 Valkey 或 Redis OSS 執行個體的使用效率。如果快取率低於大約 0.8，表示有大量金鑰移出、過期或不存在。這是使用 <code>cache_hits</code> 和 <code>cache_misses</code> 統計資料以下列方式計算的： $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$ 。	百分比
ChannelAuthorizationFailures	使用者嘗試存取他們沒有存取許可的金鑰失敗總次數。您可以使用 <a href="#">ACLOG</a> 命令來尋找個別身分驗證失敗的詳細資訊。建議對此設定警示，以偵測未經授權的存取嘗試。	計數

指標	描述	單位
CurrConnections	用戶端連線的數量，不包括來自僅供讀取複本的連線。ElastiCache 會使用二到四個連線來監控叢集。這是衍生自的 <code>connected_clients</code> 統計資料 <a href="#">INFO</a> 。	計數
CurrItems	快取中的項目數。這衍生自 <code>keyspace</code> 統計，將整個金鑰空間中的所有金鑰加總。	計數
	Dimension: Tier=Memory 適用於使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。記憶體中的項目數。	計數
	Dimension: Tier=SSD (固態硬碟) 適用於使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。中的項目數量 SSD。	計數
CurrVolatileItems	所有資料庫中已設定 <code>tll</code> 的金鑰總數。這是衍生自 <code>expires</code> 統計資料，將整個金鑰空間中所有金鑰與 <code>tll</code> 集加總。	計數
DatabaseCapacityUsagePercentage	<p>使用中叢集之總資料容量百分比。</p> <p>在資料分層執行個體上，指標計算為 <math>(\text{used\_memory} - \text{mem\_not\_counted\_for\_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})</math>，其中 <code>maxmemory</code> <code>used_memory</code> 和取自 <a href="#">INFO</a>。</p> <p>在所有其他情況下，會使用計算指標 <code>used_memory/maxmemory</code>。</p>	百分比


指標	描述	單位
DatabaseCapacityUsageCountedForEvictPercentage	<p>使用中叢集的總資料容量百分比，不包括用於額外負荷和的記憶體COB。此指標的計算方式為：</p> $\frac{\text{used\_memory} - \text{mem\_not\_counted\_for\_evict}}{\text{maxmemory}}$ <p>在資料分層執行個體上，指標的計算方式為：</p> $\frac{(\text{used\_memory} + \text{SSD used})}{(\text{maxmemory} + \text{SSD total capacity})}$ <p><code>used_memory</code> 和 <code>maxmemory</code> 的來源 <a href="#">INFO</a></p>	百分比
DatabaseMemoryUsagePercentage	<p>叢集使用之記憶體百分比。這是使用 <code>used_memory/maxmemory</code> 的計算<a href="#">INFO</a>。</p>	百分比
DatabaseMemoryUsageCountedForEvictPercentage	<p>使用中叢集的記憶體百分比，不包括用於額外負荷和的記憶體COB。這是使用 <code>used_memory-mem_not_counted_for_evict/maxmemory</code> 的計算<a href="#">INFO</a>。</p>	百分比
DB0AverageTTL	<p><code>avg_ttl</code> DBO 從 <a href="#">INFO</a>命令的統計資料公開 <code>keyspace</code>。複本不會使金鑰過期，而會等待主節點讓金鑰過期。當主要節點過期金鑰（或因為而將其移除LRU）時，它會合成命令，該DEL命令會傳輸至所有複本。因此，複本節點的DB0AverageTTL為0，因為它們不會過期金鑰，因此不會追蹤TTL。</p>	毫秒

指標	描述	單位
EngineCPUUtilization	<p>提供 Valkey 或 Redis OSS引擎執行緒的使用 CPU率。由於 Valkey 和 Redis OSS是單執行緒，因此您可以使用此指標來分析程序本身的負載。此EngineCPUUtilization 指標可提供更精確的程序可見性。您可以搭配 CPUUtilization 指標使用它。CPUUtilization 會公開伺服器執行個體的整體CPU使用率，包括其他作業系統和管理程序。對於具有四個 vCPUs 或更多節點的較大節點類型，請使用 EngineCPU Utilization 指標來監控和設定擴展的閾值。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>在 ElastiCache 主機上，背景程序會監控主機以提供受管資料庫體驗。這些背景程序可能會佔用CPU工作負載的很大一部分。這對於超過兩個的大型主機來說並不重要vCPUs。但可能會影響 2 個vCPUs 或更少的小型主機。如果您只監控EngineCPUUtilization 指標，則不會知道主機因來自 Valkey 或 Redis 的高CPU用量OSS和來自背景監控程序的高CPU用量而過載的情況。因此，我們建議監控兩個vCPUs或更少主機的CPUUtilization 指標。</p> </div>	百分比
Evictions	因 maxmemory 限制而移出的金鑰數目。這是衍生自的evicted_keys 統計資料 <a href="#">INFO</a> 。	計數
GlobalDatastoreReplicationLag	這是次要區域的主節點和主要區域的主節點之間的延遲。對於啟用叢集模式的 Valkey 或 Redis OSS，延遲表示碎片之間的最大延遲。	秒鐘



指標	描述	單位
IamAuthenticationExpirations	過期IAM且經過驗證的 Valkey 或 Redis OSS連線總數。您可以在使用者指南中找到有關 <a href="#">使用 驗證 IAM</a> 的詳細資訊。	計數
IamAuthenticationThrottling	限流 IAM- 驗證的 Valkey 或 Redis OSSAUTH或 HELLO請求總數。您可以在使用者指南中找到有關 <a href="#">使用 驗證 IAM</a> 的詳細資訊。	計數
IsMaster	指出節點是否為目前碎片/叢集的主節點。指標可能是 0 (非主要) 或 1 (主要)。	計數
KeyAuthorizationFailures	使用者嘗試存取他們沒有存取許可的金鑰失敗總次數。您可以使用 <a href="#">ACLOG</a> 命令來尋找個別身分驗證失敗的詳細資訊。建議對此設定警示，以偵測未經授權的存取嘗試。	計數
KeysTracked	Valkey 或 Redis 金鑰追蹤所追蹤的OSS金鑰數目，以的百分比表示tracking-table-max-keys。金鑰追蹤用來協助用戶端快取，並在金鑰修改時通知用戶端。	計數
MemoryFragmentationRatio	指示配置 Valkey 或 Redis OSS引擎記憶體的效率。某些閾值表示不同的行為。建議的值是具有 1.0 以上的片段。這是從 mem_fragmentation_ratio statistic 的計算 <a href="#">INFO</a> 。	Number

指標	描述	單位
NewConnections	<p>在此期間內，伺服器已接受的連線總數。這是衍生自的total_connections_received 統計資料<a href="#">INFO</a>。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>如果您使用的是 ElastiCache ( Redis OSS ) 第 5 版或更舊版本，則此指標報告的 2 到 4 個連線會由 ElastiCache 用來監控叢集。不過，使用 ElastiCache ( Redis OSS ) 第 6 版或更新版本時，ElastiCache 用於監控叢集的連線不包含在此指標中。</p> </div>	計數
NumItemsReadFromDisk	每分鐘從磁碟檢索的項目總數。僅支援使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。	計數
NumItemsWrittenToDisk	每分鐘寫入磁碟的項目總數。僅支援使用 <a href="#">中的資料分層 ElastiCache</a> 的叢集。	計數
MasterLinkHealthStatus	此狀態有兩個值：0 或 1。值 0 表示主節點中的 ElastiCache 資料未與 OSS 上的 Valkey 或 Redis 同步 EC2。值為 1 表示資料同步。若要完成遷移，請使用 <a href="#">CompleteMigration</a> API 操作。	Boolean
Reclaimed	金鑰過期事件總數。這是衍生自的expired_keys 統計資料 <a href="#">INFO</a> 。	計數
ReplicationBytes	針對複寫組態中的節點，ReplicationBytes 會報告主節點傳送給其所有複本的位元組數。此指標代表複寫群組上的寫入負載。這是衍生自的master_repl_offset 統計資料 <a href="#">INFO</a> 。	位元組

指標	描述	單位
ReplicationLag	此指標僅適用於以讀取複本形式執行的節點。它代表複本要多久的時間 (秒) 才會套用主要節點變更。對於 Valkey 7.2 及更新版本，以及 Redis OSS引擎 5.0.6 版更新版本，延遲可以毫秒為單位進行測量。	秒鐘
SaveInProgress	只要進行背景儲存 (延伸或無延伸)，此二進位指標就會傳回 1，否則為 0。背景儲存程序一般是在快照與同步期間使用。這些操作可能會導致效能降低。使用 SaveInProgress 指標，您可以診斷效能降低是否為背景儲存程序所造成。這是衍生自的 <code>rdb_bgsave_in_progress</code> 統計資料 <a href="#">INFO</a> 。	Boolean
TrafficManagementActive	指示 ElastiCache ( Redis OSS ) 是否透過調整配置給傳入命令、監控或複寫的流量來主動管理流量。當傳送的命令超過 Valkey 或 Redis 可以處理的數目時，會管理流量，OSS並用於維持引擎的穩定性和最佳操作。任何資料點為 1 可能表示節點的規模不足以因應所提供的工作負載。  <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>如果此指標保持作用中狀態，請評估叢集，以決定是否需要縱向擴展或橫向擴展。相關指標包括 <code>NetworkBandwidthOutAllowanceExceeded</code> 和 <code>EngineCPUUtilization</code> 。</p> </div>	Boolean

## EngineCPUUtilization 可用性

AWS 下列區域適用於所有支援的節點類型。

區域	區域名稱
us-east-2	美國東部 (俄亥俄)
us-east-1	美國東部 (維吉尼亞北部)
us-west-1	美國西部 (加利佛尼亞北部)
us-west-2	美國西部 (奧勒岡)
ap-northeast-1	亞太區域 (東京)
ap-northeast-2	亞太區域 (首爾)
ap-northeast-3	亞太區域 (大阪)
ap-east-1	亞太區域 (香港)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-1	亞太區域 (新加坡)
ap-southeast-2	亞太區域 (悉尼)
ap-southeast-3	亞太區域 (雅加達)
ca-central-1	加拿大 (中部)
cn-north-1	中國 (北京)
cn-northwest-2	中國 (寧夏)
me-south-1	Middle East (Bahrain)
eu-central-1	歐洲 (法蘭克福)
eu-west-1	歐洲 (愛爾蘭)
eu-west-2	歐洲 (倫敦)
eu-west-3	歐洲 (巴黎)

區域	區域名稱
eu-south-1	歐洲 (米蘭)
af-south-1	非洲 (開普敦)
eu-north-1	歐洲 (斯德哥爾摩)
sa-east-1	南美洲 (聖保羅)
us-gov-west-1	AWS GovCloud (美國西部)
us-gov-east-1	AWS GovCloud (美國東部)

這些是來自 `info commandstats` 的特定命令類型彙整。命令統計資料區段會根據命令類型提供統計資料，包括呼叫次數、這些命令耗用的總CPU時間，以及每個命令執行的平均CPU耗用量。對於每個命令類型，都會新增下列行：`cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`。

下列延遲指標是使用來自的 `commandstats` 統計資料計算 [INFO](#)。它們的計算方式如下： $\text{delta}(\text{usec})/\text{delta}(\text{calls})$ 。`delta` 計算為一分鐘內的差異。延遲定義為 ElastiCache 處理命令所花費CPU的時間。請注意，對於使用資料分層的叢集，從中擷取項目所花費的時間SSD不包含在這些測量中。

如需可用命令的完整清單，請參閱 Valkey 文件中的 [命令](#)。

指標	描述	單位
ClusterBasedCmds	叢集類型命令總數。這是透過加總對叢集 ( <code>cluster slot</code> 、 <code>cluster info</code> 等) 執行動作的所有命令，從 <code>commandstats</code> 統計資料衍生而來。	計數
ClusterBasedCmdsLatency	以叢集為基礎之命令的延遲。	微秒
EvalBasedCmds	以 <code>eval</code> 為基礎之命令的命令總數。這是透過加總 <code>commandstats eval</code> 、衍生自統計資料 <code>evalsha</code> 。	計數

指標	描述	單位
EvalBasedCmdsLatency	以 eval 為基礎之命令的延遲。	微秒
GeoSpatialBasedCmds	以 geospatial- 為基礎的之命令的命令總數。這是衍生自commandstats 統計資料。加總了下列 geo 類型的所有命令而得出：geoadd、geodist、geohash、geopos、georadius 及 georadiusbymember。	計數
GeoSpatialBasedCmdsLatency	以 geospatial 為基礎之命令的延遲。	微秒
GetTypeCmds	read-only 類型命令的總數。這透過加總所有read-only類型命令 commandstats ( get、lrange、等 ) hget scard來衍生自統計數字。	計數
GetTypeCmdsLatency	讀取命令的延遲。	微秒
HashBasedCmds	雜湊類型命令總數。這透過加總對一或多個雜湊 ( hget、hkeys、hdel、等 ) hvals採取行動的所有命令，衍生自commandstats 統計資料。	計數
HashBasedCmdsLatency	以 hash 為基礎之命令的延遲。	微秒
HyperLogLogBasedCmds	以 HyperLogLog 為基礎的命令總數。這是透過加總所有pf類型的命令 commandstats ( pfadd、pfmerge、pfcount等 ) 來衍生自統計資料。	計數
HyperLogLogBasedCmdsLatency	HyperLogLog以 為基礎的命令延遲。	微秒
JsonBasedCmds	JSON 命令的總數，包括讀取和寫入命令。這透過加總對JSON索引鍵執行動作的所有JSON命令，從commandstats 統計資料衍生而來。	計數

指標	描述	單位
JsonBasedCmdsLatency	所有JSON命令的延遲，包括讀取和寫入命令。	微秒
JsonBasedGetCmds	JSON 唯讀命令的總數。這透過加總對JSON金鑰採取行動的所有JSON讀取命令，從commandstats 統計衍生而來。	計數
JsonBasedGetCmdsLatency	JSON 唯讀命令的延遲。	微秒
JsonBasedSetCmds	JSON 寫入命令的總數。這是透過加總對JSON金鑰採取行動的所有JSON寫入命令，從commandstats 統計資料衍生而來。	計數
JsonBasedSetCmdsLatency	JSON 寫入命令的延遲。	微秒
KeyBasedCmds	金鑰類型命令總數。這透過加總對多個資料結構（del、rename、等）expire中的一或多個索引鍵執行動作的所有命令，從commandstats 統計資訊衍生而來。	計數
KeyBasedCmdsLatency	以 key 為基礎之命令的延遲。	微秒
ListBasedCmds	清單類型命令總數。這透過加總對一或多個清單（lindex、lrange、ltrim、等）lpush執行動作的所有命令，衍生自commandstats 統計資料。	計數
ListBasedCmdsLatency	以 list 為基礎之命令的延遲。	微秒
NonKeyTypeCmds	非金鑰型命令的總數。這透過加總所有不對金鑰採取行動的命令，例如 acldbsize或，從commandstats 統計資料衍生而來info。	計數
NonKeyTypeCmdsLatency	命令的 non-key-based延遲。	微秒

指標	描述	單位
PubSubBasedCmds	pub/sub 功能的命令總數。這透過加總用於 pub/sub 功能的所有命令來衍生自commandstats 統計資料：psubscribe、publish、pubsub、punsubscribe、ssubscribe、spublish subscribe和unsubscribe。	計數
PubSubBasedCmdsLatency	以 pub/sub 為基礎之命令的延遲。	微秒
SetBasedCmds	集合類型命令總數。這透過加總對一或多個集 ( scard、sdiff、sunion、等 ) sadd採取行動的所有命令，衍生自commandstats 統計資料。	計數
SetBasedCmdsLatency	以 set 為基礎之命令的延遲。	微秒
SetTypeCmds	write 類型命令的總數。這透過加總在資料 ( set、hset、lpop、等 ) sadd上操作的所有命令mutative類型，從commandstats 統計資訊衍生而來。	計數
SetTypeCmdsLatency	寫入命令的延遲。	微秒
SortedSetBasedCmds	有序集合類型命令總數。這透過加總對一或多個排序集 ( zcount、zrange、zadd、等 ) zrank執行動作的所有命令，衍生自commandstats 統計資料。	計數
SortedSetBasedCmdsLatency	以 sorted 為基礎之命令的延遲。	微秒
StringBasedCmds	字串類型命令總數。這透過加總對一或多個字串 ( strlen、setrange、等 ) setex執行動作的所有命令，從commandstats 統計資料衍生而來。	計數



指標	描述	單位
StringBasedCmdsLatency	以 string 為基礎之命令延遲。	微秒
StreamBasedCmds	串流類型命令總數。這透過加總對一或多個串流資料類型 ( xrange、xlen、xdel、等 ) xadd採取行動的所有命令，衍生自commandstats 統計資料。	計數
StreamBasedCmdsLatency	以 stream 為基礎之命令的延遲。	微秒

## Memcached 的指標

AWS/ElastiCache 命名空間包含下列 Memcached 指標。

AWS/ElastiCache namespace 包含下列衍生自 Memcached 統計資料命令的指標。每個指標都是在快取模式層級進行計算。

另請參閱

- [主機層級指標](#)

指標	描述	單位
BytesReadIntoMemcached	快取節點從網路讀取的位元組數。	位元組
BytesUsedForCacheItems	用於存放快取項目的位元組數。	位元組
BytesWrittenOutFromMemcached	快取節點寫入至網路的位元組數。	位元組
CasBadval	快取收到的 CAS ( 檢查和設定 ) 請求數目，其中 Cas 值與儲存的 Cas 值不相符。	計數

指標	描述	單位
CasHits	快取已接收到，其請求的金鑰已找到且 Cas 值匹配的 Cas 請求的數量。	計數
CasMisses	快取已接收到其請求的金鑰未找到的 Cas 請求的數量。	計數
CmdFlush	快取已接收到的 flush 命令的數量。	計數
CmdGet	快取已接收到的 get 命令的數量。	計數
CmdSet	快取接收到的 set 命令數量。	計數
CurrConnections	<p>即時連接至快取的連接數量計數。ElastiCache 使用兩到三個連線來監控叢集。</p> <p>除了以上之外，memcached 會建立一些內部連線，等於用於節點類型的兩倍執行緒。各種節點類型的執行緒計數，可於 Nodetype Specific Parameters 適用的參數群組中查看。</p> <p>連線總數是用戶端連線、監控的連線和以上提及的內部連線的總和。</p>	計數
CurrItems	目前儲存於快取的項目數量計數。	計數
DecrHits	在請求的金鑰已找到的情況下，快取接收到的減少請求的數量。	計數
DecrMisses	在請求的金鑰未找到的情況下，快取接收到的減少請求的數量。	計數
DeleteHits	在請求的金鑰已找到的情況下，快取接收到的刪除請求的數量。	計數
DeleteMisses	在請求的金鑰未找到的情況下，快取接收到的刪除請求的數量。	計數

指標	描述	單位
Evictions	快取已移除以提供空間給新的寫入的未逾期項目的數量。	計數
GetHits	在請求的金鑰已找到的情況下，快取接收到的取得請求的數量。	計數
GetMisses	快取已接收到其請求的金鑰未找到的取得請求的數量。	計數
IncrHits	在請求的金鑰已找到的情況下，快取接收到的增加請求的數量。	計數
IncrMisses	在請求的金鑰未找到的情況下，快取接收到的增加請求的數量。	計數
Reclaimed	快取已移除以提供空間給新的寫入的逾期項目的數量。	計數

針對 Memcached 1.4.14 提供以下額外的指標。

指標	描述	單位
BytesUsedForHash	雜湊表目前使用的位元組數量。	位元組
CmdConfigGet	累積的 config get 請求的數量。	計數
CmdConfigSet	累積的 config set 請求的數量。	計數
CmdTouch	累積的 touch 請求的數量。	計數
CurrConfig	目前已儲存的組態的數量。	計數
EvictedUnfetched	從最近使用過的快取 (LRU) 中移出的有效項目數量，這些項目在設定後從未觸碰過。	計數
ExpiredUnfetched	從中回收的過期項目數量LRU，這些項目在設定後從未觸碰過。	計數

指標	描述	單位
SlabsMoved	已移除的 slab 頁面的總數。	計數
TouchHits	已被碰觸並賦予新的過期時間的金鑰的數量。	計數
TouchMisses	已被碰觸但未被找到的項目的數量。	計數

AWS/ElastiCache namespace 包含下列計算的快取層級指標。

指標	描述	單位
NewConnections	快取已接收到的新連線的數量。這是衍伸自藉由記錄 total_connections 在一段期間內的變更而獲得的 memcached total_connections 統計資訊。由於為預留的連線，因此這一律至少為 1 ElastiCache。	計數
NewItems	快取已儲存的新項目的數量。這是衍伸自藉由記錄 total_items 在一段期間內的變更而獲得的 memcached total_items 統計資訊。	計數
UnusedMemory	<p>未被資料使用的記憶體量。這是衍伸自 Memcached 統計資訊 limit_maxbytes 和藉由減去 limit_maxbytes 位元組而獲得的。</p> <p>由於 Memcached 額外負荷除了資料使用的記憶體之外，也 UnusedMemory 不應被視為可用於其他資料的記憶體量。即使您還有一些未使用的記憶體，也可能會遇到被移出的情形。</p> <p>如需詳細資訊，請參閱 <a href="#">Memcached 項目記憶體使用量</a>。</p>	位元組

## 應監控哪些指標？

下列 CloudWatch 指標可提供 ElastiCache 效能的良好洞見。在大多數情況下，我們建議您為這些指標設定 CloudWatch 警示，以便在發生效能問題之前採取修正動作。

### 要監控的指標

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage \( Valkey 和 RedisOSS \)](#)
- [移出](#)
- [CurrConnections](#)
- [記憶體 \( Valkey 和 RedisOSS \)](#)
- [網路](#)
- [Latency \(延遲\)](#)
- [複寫](#)
- [流量管理 \( Valkey 和 RedisOSS \)](#)

### CPUUtilization

此為主機層級指標，以百分比報告。如需詳細資訊，請參閱[主機層級指標](#)。

### Valkey 和 Redis OSS

對於 2 vCPUs 或更少的較小節點類型，請使用 CPUUtilization 指標來監控工作負載。

一般而言，我們建議您將閾值設定為可用的 90%CPU。由於 Valkey 和 Redis OSS 都是單執行緒，因此實際閾值應該計算為節點總容量的分數。例如，假設您使用擁有二核心的節點類型。在這種情況下，的閾值 CPUUtilization 為  $90/2$  或 45%。

您將需要根據您所使用快取節點中的核心數，來判斷您自己的閾值。若您超過此閾值，並且您的主要工作負載是來自讀取請求，請透過新增僅供讀取複本來向外擴展您的快取叢集。若主要工作負載是來自寫入請求，取決於您的叢集組態，我們建議您：

- Valkey 或 Redis OSS ( 停用叢集模式 ) 叢集：使用較大的快取執行個體類型來擴展規模。
- Valkey 或 Redis OSS ( 啟用叢集模式 ) 叢集：新增更多碎片，將寫入工作負載分散到更多主要節點。

**Tip**

與其使用主機層級指標 CPUUtilization，Valkey 和 Redis OSS 使用者可以使用指標 EngineCPUUtilization，該指標會報告 Valkey 或 Redis OSS 引擎核心上的用量百分比。若要查看此指標是否可在您的節點上使用，以及如需詳細資訊，請參閱 [Valkey 和 Redis 的指標OSS](#)。

對於 4 個 vCPUs 以上的較大節點類型，您可能想要使用 EngineCPUUtilization 指標，該指標會報告 Valkey 或 Redis OSS 引擎核心上的用量百分比。若要查看此指標是否在您的節點上可用，以及如需詳細資訊，請參閱 [Redis 的指標。OSS](#)

### Memcached

因為 Memcached 為多執行緒，此指標可高達 90%。如果您超過此閾值，請使用較大的快取節點類型來擴展快取叢集，或新增更多快取節點來擴展。

### EngineCPUUtilization

對於具有 4 個 vCPUs 或更多節點的較大節點類型，您可能想要使用 EngineCPUUtilization 指標，該指標會報告 Redis OSS 引擎核心上的用量百分比。若要查看此指標是否可在您的節點上使用，以及如需詳細資訊，請參閱 [Valkey 和 Redis 的指標OSS](#)。

如需詳細資訊，請參閱使用 [Amazon 監控 Amazon ElastiCache \(RedisOSS\) 最佳實務 CloudWatch CPUs](#) 一節。

### SwapUsage ( Valkey 和 RedisOSS )

此為主機層級指標，以位元組報告。如需詳細資訊，請參閱 [主機層級指標](#)。

FreeableMemory CloudWatch 指標接近 0 ( 即低於 100MB 或 SwapUsage 指標大於 FreeableMemory 指標，表示節點處於記憶體壓力下。如果發生此情況，請見下列主題：

- [確保您有足夠的記憶體來製作 Valkey 或 Redis OSS 快照](#)
- [管理 Valkey 和 Redis 的預留記憶體 OSS](#)

### 移出

此為快取引擎指標。建議您根據應用程式需求，親自判斷此指標的警示閾值。

如果您使用 Memcached 並超過您選擇的閾值，請使用較大的節點類型來擴展叢集，或新增更多節點來擴展叢集。

## CurrConnections

此為快取引擎指標。建議您根據應用程式需求，親自判斷此指標的警示閾值。

越來越多的 CurrConnections 可能表示您的應用程式發生問題；您需要調查應用程式行為以解決此問題。

如需詳細資訊，請參閱使用 [Amazon 監控 Amazon ElastiCache \( RedisOSS \) 最佳實務 CloudWatch](#) 的連線一節。

## 記憶體 ( Valkey 和 RedisOSS )

記憶體是 Valkey 和 Redis 的核心面向OSS。為避免資料遺失以及因應資料集的未來成長而調整，了解叢集的記憶體使用率是必要的。有關節點記憶體使用率的統計資料，請參閱 [INFO](#) 命令的記憶體區段。

如需詳細資訊，請參閱 [使用 Amazon 監控 Amazon ElastiCache \( RedisOSS \) 最佳實務 CloudWatch](#) 的記憶體一節。

## 網路

叢集網路頻寬容量的決定因素之一，是您選取的節點類型。如需節點網路容量的詳細資訊，請參閱 [Amazon ElastiCache 定價](#)。

如需詳細資訊，請參閱 [使用 Amazon 監控 Amazon ElastiCache \( RedisOSS \) 最佳實務 CloudWatch](#) 的網路一節。

## Latency (延遲)

您可以使用提供每個資料結構彙總延遲的一組 CloudWatch 指標來測量命令的延遲。這些延遲指標是使用 Valkey commandstats [INFO](#) 命令的統計資料計算。

如需詳細資訊，請參閱 [使用 Amazon ElastiCache 監控 Amazon 最佳實務 CloudWatch](#) 的延遲一節。

## 複寫

遭複寫的資料量可透過 ReplicationBytes 指標顯示。雖然此指標代表複寫群組上的寫入負載，但並沒有提供複寫運作狀態的深入分析。針對這個用途，您可以使用 ReplicationLag 指標。

如需詳細資訊，請參閱[使用 Amazon 監控 Amazon ElastiCache \( RedisOSS \) 最佳實務的 CloudWatch 複寫章節](#)。

## 流量管理 ( Valkey 和 RedisOSS )

ElastiCache ( Redis OSS ) 會在傳送的傳入命令數量超過 Valkey 或 Redis 可以處理的節點時，自動針對節點管理流量OSS。這樣做是為了讓引擎保持最佳運作狀態和穩定性。

若在節點上主動管理流量，則指標 TrafficManagementActive 會發出資料點 1。這表示節點的規模可能不足以因應所提供的工作負載。如果此指標長時間維持 1，請評估叢集，以決定是否需要縱向擴展或橫向擴展。

如需詳細資訊，請參閱[指標](#)頁面上的 TrafficManagementActive 指標。



## 選擇指標統計資料與期間

雖然 CloudWatch 可讓您為每個指標選擇任何統計數字和期間，但並非所有組合都很有用。例如，的平均、最小和最大統計資料 CPU Utilization 很有用，但總和統計資料則沒有。

所有 ElastiCache 範例都會針對每個個別快取節點發佈 60 秒的持續時間。針對任何 60 秒期間，快取節點指標僅會包含一個單一樣本。

如需如何擷取您快取節點指標的進一步資訊，請參閱 [監控 CloudWatch 叢集和節點指標](#)。

## 監控 CloudWatch 叢集和節點指標

ElastiCache 和 CloudWatch 已整合，因此您可以收集各種指標。您可以使用 [監控這些指標 CloudWatch](#)。

### Note

下列範例需要 CloudWatch 命令列工具。如需 CloudWatch 和 [的詳細資訊](#)，請參閱 [CloudWatch 產品頁面](#) 以下載開發人員工具。

下列程序說明如何使用 CloudWatch 來收集過去一小時快取叢集的儲存空間統計資料。

### Note

以下範例提供的 StartTime 和 EndTime 值僅做為說明之用。您必須將快取節點的開始和結束時間值替換為適合的值。

如需 ElastiCache 限制的資訊，請參閱 [AWS 的服務限制](#) ElastiCache。

## 監控 CloudWatch 叢集和節點指標（主控台）

收集快取叢集的使用 CPU 率統計資料

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 選取您要檢視其指標的快取節點。

**Note**

選取 20 個以上的節點時，會停用主控台上的指標檢視。

- a. 在 AWS 管理主控台的快取叢集頁面上，按一下一或多個快取叢集的名稱。

快取叢集的詳細資訊頁面隨即出現。

- b. 按一下視窗頂端的 Nodes (節點) 標籤。
- c. 在詳細資訊視窗上的 Nodes (節點) 標籤，選取您要檢視其指標的快取節點。

可用的 CloudWatch 指標清單會出現在主控台視窗底部。

- d. 按一下 CPU 使用量指標。

CloudWatch 主控台會開啟，顯示您選擇的指標。若要變更顯示的指標，可以使用 Statistic (統計數字) 和 Period (期間) 下拉式清單方塊和 Time Range (時間範圍) 索引標籤。

## 使用 監控 CloudWatch 叢集和節點指標 CloudWatch CLI

### 收集快取叢集的使用 CPU 率統計資料

- 若為 Linux、macOS 或 Unix：

```
aws cloudwatch get-metric-statistics \
 --namespace AWS/ElastiCache \
 --metric-name CPUUtilization \
 --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
 --statistics=Average \
 --start-time 2018-07-05T00:00:00 \
 --end-time 2018-07-06T00:00:00 \
 --period=3600
```

### 針對 Windows：

```
aws cloudwatch get-metric-statistics ^
 --namespace AWS/ElastiCache ^
 --metric-name CPUUtilization ^
```

```
--dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' ^
--statistics=Average ^
--start-time 2018-07-05T00:00:00 ^
--end-time 2018-07-06T00:00:00 ^
--period=3600
```

## 使用 監控 CloudWatch 叢集和節點指標 CloudWatch API

### 收集快取叢集的使用CPU率統計資料

- GetMetricStatistics 使用下列參數呼叫 CloudWatch API ( 請注意 , 開始和結束時間僅顯示為範例 ; 您需要替換自己的適當開始和結束時間 ) :
  - Statistics.member.1=Average
  - Namespace=AWS/ElastiCache
  - StartTime=2013-07-05T00:00:00
  - EndTime=2013-07-06T00:00:00
  - Period=60
  - MeasureName=CPUUtilization
  - Dimensions=CacheClusterId=mycachecluster,CacheNodeId=0002

### Example

```
http://monitoring.amazonaws.com/
?Action=GetMetricStatistics
&SignatureVersion=4
&Version=2014-12-01
&StartTime=2018-07-05T00:00:00
&EndTime=2018-07-06T23:59:00
&Period=3600
&Statistics.member.1=Average
&Dimensions.member.1="CacheClusterId=mycachecluster"
&Dimensions.member.2="CacheNodeId=0002"
&Namespace=&AWS;/ElastiCache
&MeasureName=CPUUtilization
&Timestamp=2018-07-07T17%3A48%3A21.746Z
&AWS;AccessKeyId=<&AWS; Access Key ID>
```

`&Signature=<Signature>`

## 的配額 ElastiCache

AWS 您的帳戶具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

若要檢視的配額 ElastiCache，請開啟 [Service Quotas 主控台](#)。在導覽窗格中，選擇 AWS 服務，然後選擇 ElastiCache。

若要請求增加配額，請參閱 Service Quotas 使用者指南中的 [請求提高配額](#)。如果 Service Quotas 中尚未提供配額，請使用 [增加服務配額表單](#)。

AWS 您的帳戶具有與相關的下列配額 ElastiCache。

資源	預設
每個區域的無伺服器快取	40
每個快取每天的無伺服器快照，Redis	24
每個區域的節點	300
每個叢集的節點、Memcached	60
每個執行個體類型每個叢集的節點、Valkey 或 Redis OSS ( 啟用叢集模式 )	90
每個碎片的節點、Valkey 或 Redis OSS ( 停用叢集模式 )	6
每區域參數群組數	300
每區域安全群組數	50
每區域子網路群組數	300
每個子網路群組的子網路	20
每個使用者群組的使用者，Redis	100
使用者數目上限，Redis	1000

資源	預設
使用者群組數目上限, Redis	100

## 參考資料

本節中的主題涵蓋使用 Amazon ElastiCache API 和 的 ElastiCache 章節 AWS CLI。本節也包含常見錯誤訊息及服務通知。

- [使用 ElastiCache API](#)
- [ElastiCache API 參考](#)
- [ElastiCache AWS CLI 參考的 區段](#)
- [Amazon ElastiCache 錯誤訊息](#)
- [通知](#)

## 使用 ElastiCache API

本節提供如何使用和實作 ElastiCache 操作的任務導向描述。如需這些操作的完整說明，請參閱 [Amazon ElastiCache API 參考](#)。

### 主題

- [使用查詢 API](#)
- [可用程式庫](#)
- [對應用程式進行疑難排解](#)

## 使用查詢 API

### 查詢參數

HTTP 查詢型請求是使用動詞 GET或 HTTP POST和名為 的查詢參數的HTTP請求Action。

每一個查詢請求都須包括一些常用參數，來處理動作的身分驗證和選取。

部分操作有數個參數清單。這些清單是使用 param.*n* 表示法來指定的。的值 *n* 是從 1 開始的整數。

### 查詢請求身分驗證

您只能傳送查詢請求，HTTPS而且您必須在每個查詢請求中包含簽章。本節說明如何建立簽章。下列程序所述的方法也稱為「簽章第 4 版」。

下列為用來對 AWS 進行驗證要求的基本步驟。這會假設您已向 註冊，AWS 並具有存取金鑰 ID 和秘密存取金鑰。

### 查詢身分驗證程序

1. 寄件者建構對 的請求 AWS。
2. 寄件者會計算請求簽章，即具有 SHA-1 雜湊函數的雜湊型訊息驗證碼（HMAC），如本主題下一節所定義。
3. 請求的寄件者會將請求資料、簽章和存取金鑰 ID（使用的秘密存取金鑰的金鑰識別符）傳送至 AWS。
4. AWS 使用存取金鑰 ID 來查詢秘密存取金鑰。
5. AWS 會使用用於計算請求中簽章的相同演算法，從請求資料和秘密存取金鑰產生簽章。
6. 如果簽章相符，則會將請求視為真實請求。若比較失敗，則會捨棄該要求，且 AWS 會傳回錯誤回應。

#### Note

如果請求包含 Timestamp 參數，針對請求計算出的簽章就會在其值的 15 分鐘後過期。  
如果請求包含 Expires 參數，簽章就會於 Expires 參數指定的時間過期。

### 計算請求簽章

1. 建立標準化查詢字串，以在本程序稍後使用：
  - a. 依參數名稱以自然位元組順序排序 UTF-8 查詢字串元件。這些參數可以來自 GETURI 或來自 POST 內文（當 Content-Type 是 Application/ 時 x-www-form-urlencoded）。
  - b. URL 根據下列規則編碼參數名稱和值：
    - i. 請勿 URL 編碼 3986 RFC 定義的任何未保留字元。這些未預留字元包括 A-Z、a-z、0-9、連字號 (-)、底線 (\_)、句點 (.) 及波狀符號 (~)。
    - ii. 對所有其他含有 %XY 的字元執行百分比編碼，其中 X 和 Y 是十六進位字元 0-9 和大寫 A-F。
    - iii. 以 %XY%ZA 格式編碼延伸 UTF-8 個字元的百分比...
    - iv. 將空白字元百分比編碼為 %20 (而非常見編碼機制使用的 +)。
  - c. 使用等於符號 (=) (ASCII 字元 61) 將編碼參數名稱與編碼值分開，即使參數值為空。



- d. 將名稱值對與 ( 和 ) ( ASCII 代碼 38 ) 分開。
2. 根據下列虛擬文法建立要簽署的字串 ( "\n" 代表ASCII新行 )。

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

HTTPRequestURI 元件是 的HTTP絕對路徑元件，URI最多可達 ，但不包含查詢字串。如果HTTPRequestURI是空的，請使用正斜線 ( / )。

3. 計算 RFC 2104 相容HMAC，其中包含您剛建立的字串、以私密存取金鑰作為金鑰，以及以SHA256 或 SHA1作為雜湊演算法。

如需詳細資訊，請參閱 <https://www.ietf.org/rfc/rfc2104.txt>。

4. 將結果值轉換成 base64。
5. 將該值包含為請求中的 Signature 參數值。

例如，以下是範例請求 (為了清楚起見，已加入分行符號)。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
```

針對上述查詢字串，您會計算下列字串的HMAC簽章。

```
GET\n
elasticache.amazonaws.com\n
Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
```

```
&X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache%2Faws4_request
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-amz-date
 content-type:
 host:elasticache.us-west-2.amazonaws.com
 user-agent:CacheServicesAPICommand_Client
x-amz-content-sha256:
x-amz-date:
```

結果為下列簽署的請求。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

如需簽章程序和計算請求簽章的詳細資訊，請參閱 [Signature 第 4 版簽署程序](#) 主題和副主題。

## 可用程式庫

AWS 提供軟體開發套件（SDKs）給偏好使用語言特定 APIs 而非查詢 建置應用程式的軟體開發人員 API。這些功能 SDKs 提供基本函數（不包含在中 APIs），例如請求身分驗證、請求重試和錯誤處理，以便更輕鬆地開始使用。SDKs 和其他資源可用於下列程式設計語言：

- [Java](#)
- [Windows 和 。NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

如需其他語言的資訊，請參閱[範本程式碼與程式庫](#)。

## 對應用程式進行疑難排解

ElastiCache 提供特定和描述性錯誤，協助您在與 互動時疑難排解問題 ElastiCache API。

### 擷取錯誤

通常，您想要應用程式在您花費任何時間處理結果之前，先檢查請求是否已產生錯誤。找出是否發生錯誤的最簡單方法是從 尋找回應中的Error節點 ElastiCache API。

XPath 語法提供搜尋Error節點是否存在的簡單方法，以及擷取錯誤碼和訊息的簡單方法。下列程式碼片段使用 Perl 和 XML : : XPath 模組來判斷請求期間是否發生錯誤。如果發生錯誤，程式碼會列印回應中的第一個錯誤碼和訊息。

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ($xp->find("//Error"))
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

### 對秘訣進行故障診斷

我們建議下列程序來診斷和解決 的問題 ElastiCache API。

- 確認 ElastiCache 是否正確執行。

若要執行此操作，只需開啟瀏覽器視窗，然後將查詢請求提交至 ElastiCache 服務（例如 <https://elasticache.amazonaws.com>）。MissingAuthenticationTokenException 或 500 內部伺服器錯誤會確認服務可用並回應請求。

- 檢查請求的結構。

每個 ElastiCache 操作在參考 中都有ElastiCache API參考頁面。再次檢查您是否正確使用參數。若要提供有關可能出錯的概念，請查看範例請求或使用者案例，來查看那些範例是否執行類似操作。

- 查看論壇。

ElastiCache 有一個討論論壇，您可以在其中搜尋其他人在過程中遇到的問題的解決方案。若要檢視論壇，請參閱

<https://forums.aws.amazon.com/> .

## 設定 ElastiCache 命令列介面

本節說明執行命令列工具的事前準備、如何取得命令列工具、如何設定工具及其環境，以及使用工具的一系列常見範例。

只有在您前往 AWS CLI 的時，才遵循本主題中的指示 ElastiCache。

### Important

Amazon ElastiCache Command Line Interface ( CLI ) 不支援 2014-09-30 API版之後的任何 ElastiCache 改進。若要從命令列使用較新的 ElastiCache 功能，請使用[AWS 命令列介面](#)。

### 主題

- [必要條件](#)
- [取得命令列工具](#)
- [設定工具](#)
- [為工具提供憑證](#)
- [環境變數](#)

## 必要條件

本文件假設您可以在 Linux/UNIX 或 Windows 環境中工作。Amazon ElastiCache 命令列工具也適用於以 UNIX為基礎的 Mac OS X，但本指南中不包含特定的 Mac OS X 指示。

以慣例而言，所有命令列文字前方都會加上通用的 **PROMPT>** 命令列提示字元，您機器實際的命令列提示字元可能有所不同。我們也使用 **\$** 來指示 Linux/UNIX特定命令，以及 **C:\>** Windows 特定命令。此命令的範例輸出則直接列出，前方沒有任何前綴字。

## Java 執行階段環境

本指南使用的命令列工具，需要 Java (版本 5 或更新版本) 來執行。可接受 JRE或 JDK 安裝。若要檢視和下載JREs一系列平台，包括 Linux/UNIX 和 Windows，請參閱 [Java SE Downloads](#)。

## 設定 Java Home 變數

命令列工具依存環境變數 (JAVA\_HOME) 來尋找 Java Runtime。此環境變數應設定為目錄的完整路徑，其中包含名為 `bin` 的子目錄，其繼而包含可執行檔 `java` (在 Linux 和上UNIX) 或 `java.exe` (在 Windows 上) 可執行檔。

## 設定 Java Home 變數

### 1. 設定 Java Home 變數。

- 在 Linux 和上UNIX，輸入下列命令：

```
$ export JAVA_HOME=<PATH>
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set JAVA_HOME=<PATH>
```

### 2. 執行 `$JAVA_HOME/bin/java -version` 並檢查輸出，來確認路徑設定。

- 在 Linux/上UNIX，您會看到類似下列的輸出：

```
$ $JAVA_HOME/bin/java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

- 在 Windows 上，您將會看到類似以下的輸出：

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

## 取得命令列工具

命令列工具可在 [ElastiCache 開發人員工具 網站](#) 上以 ZIP 檔案形式提供。這些工具以 Java 編寫，並包含適用於 Windows 2000/XP/Vista/Windows 7、Linux/ UNIX 和 Mac 的 Shell 指令碼 OS X。ZIP 檔案為獨立式，不需要安裝；只需下載 zip 檔案並將其解壓縮至本機機器上的目錄即可。

## 設定工具

命令列工具依賴環境變數（AWS\_ELASTICACHE\_HOME）來尋找支援程式庫。在您使用此工具之前，您必須先設定此環境變數。將它設為您解壓縮命令列工具的目錄路徑。此目錄名為 ElastiCacheCli-A.B.nnnn（A、B 和 n 是版本/發行版本編號），並包含名為 bin 和 lib 的子目錄。

### 設定 AWS\_ELASTICACHE\_HOME 環境變數

- 開啟命令列視窗，然後輸入下列其中一個命令來設定 AWS\_ELASTICACHE\_HOME 環境變數。
  - 在 Linux 和 上 UNIX，輸入下列命令：

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

為了讓工具更易於使用，我們建議您將工具的 BIN 目錄新增至您的系統 PATH。本指南的其餘部分假設 BIN 目錄位於您的系統路徑中。

### 將工具的 BIN 目錄新增至您的系統路徑

- 輸入下列命令，將工具的 BIN 目錄新增至您的系統 PATH。
  - 在 Linux 和 上 UNIX，輸入下列命令：

```
$ export PATH=$PATH:&AWS;_ELASTICACHE_HOME/bin
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

**Note**

Windows 環境變數會在您關閉命令視窗時重設。建議您將它設為永久值。請查閱您 Windows 版本的文件以取得詳細資訊。

**Note**

包含空格的路徑必須包裝在雙引號中，例如：  
"C:\Program Files\Java"

## 為工具提供憑證

命令列工具需要 AWS 帳戶隨附的 AWS 存取金鑰和秘密存取金鑰。您可以透過使用命令列或從位於您本機系統上的登入資料檔案來取得它們。

部署包含範本檔案 `${AWS_ELASTICACHE_HOME}/credential-file-path.template`，您需要使用資訊進行編輯。以下是範本檔案的內容：

```
AWS AccessKeyId=<Write your AWS access ID>
AWS SecretKey=<Write your AWS secret key>
```

**⚠ Important**

在上 UNIX，將許可限制為憑證檔案的擁有者：

```
$ chmod 600 <the file created above>
```

使用憑證檔案設定時，您需要設定 `AWS_CREDENTIAL_FILE` 環境變數，以便 ElastiCache 工具可以找到您的資訊。

### 設定 `AWS_CREDENTIAL_FILE` 環境變數

#### 1. 設定 環境變數：

- 在 Linux 和 上UNIX，使用以下命令更新變數：

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- 在 Windows 上，請使用下列命令來設定變數：

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. 檢查您的設定正常運作，請執行以下命令：

```
elasticache --help
```

您應該會看到所有 ElastiCache 命令的使用頁面。

## 環境變數

環境變數在編寫指令碼、設定預設值或暫時覆寫預設值時很有用。

除了 `AWS_CREDENTIAL_FILE` 環境變數之外，ElastiCache 命令列介面隨附的大多數 API 工具都支援下列變數：

- `EC2_REGION` — 要使用 AWS 的區域。
- `AWS_ELASTICACHE_URL` — URL 用於服務呼叫的。如果指定 `EC2_REGION` 或傳遞 `--region` 參數，則不需要指定不同的區域端點。

下列範例示範如何設定環境變數 `EC2_REGION`，以設定 API 工具所使用的區域：

Linux、OS X，或 Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```



## Amazon ElastiCache 錯誤訊息

Amazon 會傳回下列錯誤訊息 ElastiCache。您可能會收到由 ElastiCache、AWS 其他服務、Valkey、Redis OSS 或 Memcached 傳回的其他錯誤訊息。如需來自 以外來源的錯誤訊息說明 ElastiCache，請參閱產生錯誤訊息之來源的文件。

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Manual snapshot quota exceeded](#)
- [Insufficient cache cluster capacity](#)

錯誤訊息：超過叢集節點配額。此區域中每個叢集最多可以有 %n 個節點。

原因：您嘗試建立或修改某個叢集，結果該叢集含有的節點超過 %n 個。

解決方案：變更要求，讓叢集的節點數不超過 %n 個。或者，如果您需要超過 %n 個節點，請使用 [Amazon ElastiCache Node 請求表單 提出請求](#)。

如需詳細資訊，請參閱 中的 [Amazon ElastiCache 限制](#) Amazon Web Services 一般參考。

錯誤訊息：超過客戶節點配額。在此區域中，您最多可以有 %n 個節點 或者，您已達到此區域 %s 個節點的配額。

原因：您嘗試建立或修改某個叢集，結果您帳戶在此區域中跨所有叢集的節點超過 %n 個。

解決方案：變更要求，讓此帳戶在區域中跨所有叢集的節點總數不超過 %n 個。或者，如果您需要超過 %n 個節點，請使用 [Amazon ElastiCache Node 請求表單 提出請求](#)。

如需詳細資訊，請參閱 中的 [Amazon ElastiCache 限制](#) Amazon Web Services 一般參考。

錯誤訊息：此叢集在 24 小時內擷取的最高手動快照數已達到上限 或此節點在 24 小時內擷取的最高手動快照數已達到其配額 %n

原因：您嘗試擷取叢集的手動快照，但您已擷取超過此叢集在 24 小時期間允許的手動快照數上限。

解決方案：等候 24 小時，再嘗試擷取叢集的其他手動快照。或者，如果您需要立即擷取手動快照，請擷取具有相同資料之其他節點的快照，例如叢集中的不同節點。

### 錯誤訊息：InsufficientCacheClusterCapacity

原因：AWS 目前隨需容量不足，無法執行您的請求。

解決方案：

- 等候幾分鐘，然後再次提交您的請求；容量會頻繁轉移。
- 以降低的節點或碎片 (節點群組) 數提交新請求。例如，如果您要提出啟動 15 個節點的單一請求，請改嘗試提出 3 個啟動 5 個節點的請求，或 15 個啟動 1 個節點的請求。
- 如果啟動的是叢集，請提交新的請求，而不要指定可用區域。
- 如果啟動的是叢集，請使用不同的節點類型 (您可以在後面的階段進行擴展) 來提交新的請求。如需詳細資訊，請參閱[擴展 ElastiCache](#)。

## 通知

本主題涵蓋您可能感興趣的 ElastiCache 通知。通知是一種狀態或事件，在大多數的情況下，通常都是暫時性的，只會持續存在直到找到解決方案或實作解決方案。通知通常會有一個開始日期和解決日期，在解決日期之後，通知便不再重要。任何一個通知可能會與您相關，也可能不相關。我們建議一套實作準則，您可以透過遵循此準則來改善叢集的效能。

通知不會宣告新的或改進的 ElastiCache 特徵或功能。

### 一般 ElastiCache 通知

目前沒有非引擎特有的未完成 ElastiCache 通知。

### ElastiCache ( Memcached ) 通知

下列 ElastiCache 通知是 Memcached 引擎特有的。

ElastiCache ( Memcached ) 特定通知

- [警示：Memcached LRU 爬蟲程式造成分割故障](#)

## 警示：Memcached LRU 爬蟲程式造成分割故障

**⚠ 提醒日期：**2017 年 2 月 28 日

在某些情況下，您的叢集可能會在 Memcached LRU Crawler 中顯示具有分割錯誤的不穩定狀態。此為 Memcached 引擎內的問題，並且已存在一段時間。在預設啟用 LRU Crawler 時，問題在 Memcached 1.4.33 中變得明顯。

如果您遇到此問題，建議您停用 LRU Crawler，直到修正為止。若要執行此作業，請在命令列使用 `lru_crawler disable`，或是修改 `lru_crawler` 參數值 (建議)。

解決日期：

解決方法：

## ElastiCache ( RedisOSS ) 特定通知

目前沒有未完成的 ElastiCache ( Redis OSS ) 通知。

# ElastiCache 文件歷史記錄

- API 版本：2015-02-02
- 最新文件更新時間：2023 年 11 月 27 日

下表說明 ElastiCache 2018 年 3 月之後使用者指南每個版本的重要變更。如需本文件更新通知，您可以訂閱RSS摘要。

## 最近 ElastiCache 更新

變更	描述	日期
<a href="#">支援 ElastiCache 搭配 Valkey</a>	ElastiCache 現在支援 Valkey。Valkey 7.2.6 與 Redis OSS 7.2 相容如需詳細資訊，請參閱 <a href="#">Valkey</a> 。	2024 年 10 月 8 日
<a href="#">大小彈性預留節點</a>	ElastiCache 現在支援 <a href="#">Size Flexible 預留節點</a> 。如需詳細資訊，請參閱 <a href="#">Amazon ElastiCache Pricing</a> 。	2024 年 10 月 1 日
<a href="#">ElastiCache ( Redis OSS ) 新增了對其他 C7gn 節點大小的支援</a>	ElastiCache ( Redis OSS ) 新增了對其他 C7gn 節點大小的支援。	2024 年 1 月 10 日
<a href="#">ElastiCache ( Redis OSS ) 現在支援建立無伺服器快取</a>	您現在可以建立無伺服器快取，它可簡化快取管理並立即擴展，以支援需求最嚴苛的應用程式。如需詳細資訊，請參閱 <a href="#">選擇部署選項</a> 。作為此功能的一部分，已將 <a href="#">新許可</a> 新增至 AmazonElastiCacheFullAccess ElastiCacheServiceRolePolicy，並允許無伺服器快取與受管VPC端點建立關聯。此外，	2023 年 11 月 27 日

已新增許可來支援改良後使用 AmazonElastiCacheFullAccess 政策的主控台體驗。

[ElastiCache \( Memcached \)](#)  
[現在支援建立無伺服器快取](#)

您現在可以建立無伺服器快取，它可簡化快取管理並立即擴展，以支援需求最嚴苛的應用程式。如需詳細資訊，請參閱[選擇部署選項](#)。作為此功能的一部分，已將[新許可](#)新增至 AmazonElastiCacheFullAccess ElastiCacheServiceRolePolicy，並允許無伺服器快取與受管VPC端點建立關聯。此外，已新增許可來支援改良後使用 AmazonElastiCacheFullAccess 政策的主控台體驗。

2023 年 11 月 27 日

[ElastiCache \( Redis OSS \)](#)  
[現在支援修改叢集模式](#)

您現在可以將叢集從停用叢集模式 ( CMD ) 遷移至已啟用叢集模式 ( CME )。如需詳細資訊，請參閱[修改叢集模式](#)。

2023 年 5 月 11 日

[ElastiCache \( Redis OSS \)](#)  
[現在支援修改傳輸中加密設定](#)

您現在可以變更 Redis OSS叢集的TLS組態，而不需要重建或重新佈建叢集，也不會影響應用程式的可用性。如需詳細資訊，請參閱[啟用現有叢集的傳輸中加密](#)。

2022 年 12 月 28 日

[ElastiCache \( Redis OSS \)](#)  
[現在支援使用 驗證使用者 IAM](#)

IAM 身分驗證可讓您使用 AWS IAM 身分驗證 ElastiCache ( Redis OSS ) 的連線。這可讓您強化安全模型，並簡化許多管理安全任務。如需詳細資訊，請參閱[使用 驗證IAM](#)。

2022 年 11 月 16 日

[ElastiCache \( Redis OSS \)](#)  
[現在支援 Redis OSS 7](#)

此版本為 Amazon ElastiCache ( Redis OSS ) 帶來了多項新功能：Redis OSS 函數、ACL 改進和陰影 Pub/Sub。如需詳細資訊，請參閱[ElastiCache \( Redis OSS \) 7.0 版](#)。

2022 年 11 月 8 日

[ElastiCache \( Redis OSS \)](#)  
[現在支援 IPV6](#)

ElastiCache 支援網際網路通訊協定第 4 版和第 6 版 ( IPv4 和 IPv6 )，可讓您將叢集設定為僅接受 IPv4 連線、僅接受 IPv6 連線或同時接受 IPv4 和 IPv6 連線 ( 雙堆疊 )。IPv6 支援在 [Nitro 系統](#) 上建置的所有執行個體上使用 Redis OSS 引擎 6.2 版以上的工作負載。透過 ElastiCache 存取 無需額外費用 IPv6。如需詳細資訊，請參閱[選擇網路類型](#)。

2022 年 11 月 7 日

## [ElastiCache \( Memcached \)](#) [現在支援 IPV6](#)

ElastiCache 支援網際網路通訊協定第 4 版和第 6 版 ( IPv4 和 IPv6 )，可讓您將叢集設定為僅接受 IPv4 連線、僅接受 IPv6 連線或同時接受 IPv4 和 IPv6 連線 ( 雙堆疊 )。IPv6 支援在 [Nitro 系統](#) 上建置的所有執行個體上使用 Memcached 引擎 1.6.6 版更新的工作負載。透過 ElastiCache 存取 無需額外費用 IPv6。如需詳細資訊，請參閱 [選擇網路類型](#)。

2022 年 11 月 7 日

## [ElastiCache \( Memcached \)](#) [現在支援傳輸中加密](#)

傳輸中加密是選用的功能，可讓您在資料最易遭受攻擊的點，也就是資料在地點間傳輸時，增加資料的安全性。Memcached 1.6.12 和更新版本支援。如需詳細資訊，請參閱 [ElastiCache 傳輸中加密 \( TLS \)](#)。

2022 年 5 月 26 日

## [ElastiCache \( Redis OSS \)](#) [現在支援原生 JavaScript](#) [Object Notation \( JSON \) 格式](#)

原生 JavaScript 物件表示法 ( JSON ) 格式是一種簡單、無結構描述的方式，可編碼 Redis OSS 叢集內的複雜資料集。您可以在 Redis OSS 叢集內使用 JavaScript Object Notation ( JSON ) 格式原生存放和存取資料，並更新存放在這些叢集中的 JSON 資料，而不需要管理自訂程式碼來序列化和還原序列化資料。如需詳細資訊，請參閱 [入門 JSON](#)。

2022 年 5 月 25 日

## [ElastiCache 現在支援 PrivateLink](#)

AWS PrivateLink 可讓您在沒有網際網路閘道、NAT裝置、VPN連線或 AWS Direct Connect 連線的情況下私有存取 ElastiCache API操作。如需詳細資訊，請參閱適用於 Redis 的 [Amazon ElastiCache API 和介面VPC端點 \( AWS PrivateLink \)](#) OSS或適用於 Memcached 的 [Amazon ElastiCache API 和介面VPC端點 \( AWS PrivateLink \)](#)。

2022 年 1 月 24 日

## [ElastiCache \( Redis OSS \) 現在支援 Redis OSS 6.2 和資料分層](#)

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個 Redis OSS引擎版本 ElastiCache。ElastiCache ( Redis OSS ) 6.2 包括使用 x86 節點類型搭配 8 個 vCPUs 以上節點或 Graviton2 節點類型搭配 4 個 vCPUs 以上啟用 TLS的叢集的效能改善。ElastiCache ( Redis OSS ) 也會引入資料分層。您可以使用資料分層作為較低成本的方式，將叢集擴展至最多數百 TB 的容量。如需詳細資訊，請參閱 [ElastiCache \( Redis OSS \) 6.2 版 \( 增強版 \)](#) 和 [資料分層](#)。

2021 年 11 月 23 日



## [支援 Auto Scaling](#)

ElastiCache ( Redis OSS ) 現在支援 Auto Scaling 。  
ElastiCache ( Redis OSS ) 自動擴展是自動增加或減少 ElastiCache ( Redis OSS ) 服務中所需碎片或複本的能力。ElastiCache 會利用 Application Auto Scaling 服務來提供此功能。如需詳細資訊，請參閱 [Auto Scaling ElastiCache \( Redis OSS \) 叢集](#)。

2021 年 8 月 19 日

## [支援交付 Redis OSS Slow 日誌](#)

ElastiCache 現在可讓您將 Redis 串流OSSSLOWLOG 至兩個目的地之一：Amazon Data Firehose 或 Amazon CloudWatch Logs。如需詳細資訊，請參閱 [日誌傳送](#)。

2021 年 4 月 22 日

## [支援標記資源和條件索引鍵](#)

ElastiCache 現在支援標記，協助您管理叢集和其他 ElastiCache 資源。如需詳細資訊，請參閱 [標記您的 ElastiCache 資源](#)。ElastiCache 也會引入條件金鑰的支援。您可以指定決定 IAM 政策如何生效的條件。如需這些條件索引鍵的詳細資訊，請參閱 [使用條件索引鍵](#)。

2021 年 4 月 7 日

## [支援標記資源和條件索引鍵](#)

ElastiCache 現在支援標記，協助您管理叢集和其他 ElastiCache 資源。如需詳細資訊，請參閱[標記您的 ElastiCache 資源](#)。ElastiCache 也會引入條件金鑰的支援。您可以指定決定 IAM 政策如何生效的條件。如需這些條件索引鍵的詳細資訊，請參閱[使用條件索引鍵](#)。

2021 年 4 月 7 日

## [ElastiCache 現在可在 AWS Outposts 上使用](#)

[AWS Outpost](#) 將原生 AWS 服務、基礎設施和操作模型帶入幾乎所有資料中心、共同位置空間或內部部署設施。您可以在 Outposts ElastiCache 上部署，以設定、操作和使用內部部署快取，就像在雲端一樣。如需詳細資訊，請參閱[使用 Outposts for Redis OSS](#)或[使用 Outposts for Memcached](#)。

2020 年 10 月 8 日

## [ElastiCache 現在支援 Redis OSS 6](#)

Amazon ElastiCache ( Redis OSS ) 推出 Amazon 支援的下一個 Redis OSS 引擎版本 ElastiCache。此版本包含[使用角色型存取控制驗證使用者身分](#)、無版本支援、用戶端快取，以及顯著的操作改善功能。如需詳細資訊，請參閱[ElastiCache \( Redis OSS \) 6.0 版 \( 增強版 \)](#)。

2020 年 10 月 7 日

## [ElastiCache 現在支援 Local Zones](#)

本機區域是地理位置接近您使用者的 AWS 區域延伸。您可以透過建立新的子網路並將其指派給 Local Zone，將任何虛擬私有雲端（VPC）從父 AWS 區域延伸至 Local Zones。如需詳細資訊，請參閱[使用本地區域](#)。

2020 年 9 月 25 日

## [ElastiCache \( Redis OSS \) 現在支援將您的 Redis OSS 叢集環境擴展到 500 個節點或 500 個碎片](#)

Redis OSS Cluster 模式可讓您使用組態，將資料分割到多個碎片，並提供更好的可擴展性、效能和可用性。此功能適用於所有 AWS 區域的 Amazon ElastiCache ( RedisOSS ) 5.0.6 版及所有現有和新的 ElastiCache ( RedisOSS ) 叢集環境。如需詳細資訊，請參閱[Redis OSS Nodes 和 Shards](#)。

2020 年 8 月 13 日

## [ElastiCache 現在支援資源層級許可](#)

您現在可以在 AWS Identity and Access Management ( IAM ) 政策中指定 ElastiCache 資源來限制使用者許可的範圍。如需詳細資訊，請參閱[資源層級許可](#)。

2020 年 8 月 12 日

## [ElastiCache \( Redis OSS \) 新增其他 Amazon CloudWatch 指標](#)

ElastiCache ( Redis OSS ) 現在支援新的 CloudWatch 指標，包括 PubSubCmds 和 HyperLogLogBasedCmds 。如需完整清單，請參閱[Redis 的指標。OSS](#)

2020 年 6 月 10 日

[ElastiCache 現在支援自動更新 ElastiCache 叢集](#)

Amazon ElastiCache 現在支援在服務更新的 ElastiCache 「建議套用截止日期」過後自動更新叢集。ElastiCache 將使用您的維護時段來排程適用叢集的自動更新。如需詳細資訊，請參閱[自助更新](#)。

2020 年 5 月 13 日

[ElastiCache \( Redis OSS \) 現在支援 Global Datastore for Redis OSS](#)

Global Datastore for Redis OSS 功能提供全受管、快速、可靠且安全的跨 AWS 區域複寫。使用此功能，您可以為 ElastiCache ( Redis OSS ) 建立跨區域僅供讀取複本叢集，以跨 AWS 區域啟用低延遲讀取和災難復原。您可以建立、修改和說明全域資料存放區。您也可以從全域資料存放區新增或移除 AWS 區域，並將 AWS 區域提升為全域資料存放區中的主要區域。如需詳細資訊，請參閱[使用全域資料存放區跨 AWS 區域複寫](#)。

2020 年 3 月 16 日

[ElastiCache \( Redis OSS \) 現在支援 Redis 5.0.6 OSS 版](#)

如需詳細資訊，請參閱[ElastiCache \( Redis OSS \) 5.0.6 版 \( 增強版 \)](#)。

2019 年 12 月 18 日

[Amazon ElastiCache 現在支援 T3-Standard 快取節點](#)

您現在可以在 Amazon 中啟動新一代一般用途爆量 T3-Standard 快取節點 ElastiCache。Amazon EC2 的 T3-Standard 執行個體提供基準 CPU 效能層級，並可隨時爆量 CPU 用量，直到累積的額度用盡為止。如需詳細資訊，請參閱[支援的節點類型](#)。

2019 年 11 月 12 日

[Amazon ElastiCache 現在支援修改現有 ElastiCache \(RedisOSS\) 伺服器上的 AUTH 權杖](#)

ElastiCache (Redis OSS) 5.0.6 現在可讓您透過設定和輪換新權杖來修改身分驗證權杖。您現在可以在字符使用期間修改啟用的字符。您也可以以在啟用傳輸中加密，且之前設定時沒有驗證字符的現有叢集中加入全新的字符。此程序有兩個步驟，您可以藉由此程序設定並輪換字符，而且不會中斷用戶端要求。目前不支援此功能 AWS CloudFormation。如需詳細資訊，請參閱[使用 Redis OSS AUTH 命令驗證使用者](#)。

2019 年 10 月 30 日

[Amazon ElastiCache 現在支援從 Amazon OSS 上的 Redis 進行線上資料遷移 EC2](#)

您現在可以使用線上遷移將資料從 Amazon OSS 上的自我託管 Redis 遷移 EC2 至 Amazon ElastiCache。如需詳細資訊，請參閱[線上遷移至 ElastiCache](#)。

2019 年 10 月 28 日

[ElastiCache \(Redis OSS\) 引入 Redis OSS Cluster 模式的線上垂直擴展。](#)

您現在可以隨需擴展或縮減碎片 Redis OSS 叢集。ElastiCache (Redis OSS) 透過變更節點類型來調整叢集的大小，而叢集會繼續保持連線，並提供傳入請求。如需詳細資訊，請參閱[透過修改節點類型來進行線上垂直擴展](#)。

2019 年 8 月 20 日

[ElastiCache \(Redis OSS\)](#)  
[現在可讓使用者為您的 Amazon ElastiCache \(Redis OSS\) 叢集使用單一讀取器端點。](#)

此功能可讓您透過單一叢集層級端點，將所有讀取流量導向您的 ElastiCache (RedisOSS) 叢集，以利用負載平衡和更高的可用性。如需詳細資訊，請參閱[尋找連線端點](#)。

2019 年 6 月 13 日

[ElastiCache \(Redis OSS\)](#)  
[現在允許使用者依自己的排程套用服務更新](#)

有了這項功能，您就能夠選擇在您所選的時間 (而不只是在維護時段期間) 套用可用的服務更新。這將最大限度地減少服務中斷，特別是在尖峰業務流程期間，並有助於確保您的叢集在 ElastiCache 支援的合規計劃中時保持合規。如需詳細資訊，請參閱 [Amazon 中的自助式更新 ElastiCache](#) 和 [Amazon 的合規驗證 ElastiCache](#)。

2019 年 6 月 4 日

[ElastiCache 標準預留執行個體方案：部分預付、所有預付和無預付。](#)

預留執行個體可讓您根據 ElastiCache 執行個體類型和 AWS 區域，靈活地將 Amazon 執行個體保留一年或三年。如需詳細資訊，請參閱使用[預留節點管理成本](#)。

2019 年 1 月 18 日

[ElastiCache \(Redis OSS\)](#)  
[支援每個 Redis OSS 叢集最多 250 個節點](#)

節點或碎片限制最多可增加至每個 ElastiCache (Redis OSS) 叢集 250 個。如需詳細資訊，請參閱[碎片](#)。

2018 年 11 月 19 日

[ElastiCache \( Redis OSS \)  
支援自動容錯移轉和備份和還  
原所有 T2 節點](#)

ElastiCache ( Redis OSS )  
引入對自動容錯移轉、建立快  
照以及備份和還原所有 T2 節  
點的支援。如需詳細資訊，  
請參閱 [ElastiCache \( Redis  
OSS \) 備份、還原和快照](#)。

2018 年 11 月 19 日

[ElastiCache \( Redis OSS \)  
支援 M5 和 R5 節點](#)

ElastiCache ( Redis OSS )  
現在支援以 AWS Nitro System  
為基礎的 M5 和 R5 節點、一  
般用途和記憶體最佳化執行個  
體類型。如需詳細資訊，請參  
閱 [支援的節點類型](#)。

2018 年 10 月 23 日

[支援動態變更僅供讀取複本的  
數量](#)

ElastiCache ( Redis OSS )  
已新增支援，可在沒有叢集停  
機時間的情況下，從任何叢集  
新增和移除僅供讀取複本。如  
需此版本中這些和其他變更的  
詳細資訊，請參閱 ElastiCache  
( Redis OSS ) 使用者指南 中  
的 [變更複本數量](#)。另請參閱 參  
考 [IncreaseReplicaCount](#) 中的  
[DecreaseReplicaCount](#) 和  
ElastiCache API

2018 年 9 月 17 日

[聯RAMP準會合規認證](#)

ElastiCache ( Redis OSS )  
現在已通過 FedRAMP 合規  
認證。如需詳細資訊，請參閱  
[Amazon 的合規驗證 ElastiCac  
he](#)。

2018 年 8 月 30 日

[Valkey 或 Redis OSS \( 啟用叢  
集模式 \) 引擎升級](#)

Amazon ElastiCache ( Redis  
OSS ) 新增了升級 Valkey 或  
Redis OSS ( 啟用叢集模式 )  
引擎版本的支援。如需詳細資  
訊，請參閱 [升級引擎版本](#)。

2018 年 8 月 20 日

<a href="#">PCI DSS 合規認證</a>	ElastiCache ( Redis OSS ) 現在已通過PCIDSS合規認證。如需詳細資訊，請參閱 <a href="#">Amazon 的合規驗證 ElastiCache</a> 。	2018 年 7 月 5 日
<a href="#">支援 ElastiCache ( Redis OSS ) 4.0.10</a>	ElastiCache ( Redis OSS ) 現在支援 Redis OSS 4.0.10，包括加密和在單一版本中調整大小的線上叢集。如需詳細資訊，請參閱 <a href="#">ElastiCache ( Redis OSS ) 4.0.10 版 ( 增強版 )</a> 。	2018 年 6 月 14 日
<a href="#">使用者指南重組</a>	現在已重組單一ElastiCache 使用者指南，以便 Redis OSS ( <a href="#">ElastiCache ( RedisOSS ) 使用者指南</a> ) 和 Memcached ( <a href="#">ElastiCache ( Memcached ) 使用者指南</a> ) 有單獨的使用者指南。Command <a href="#">AWS CLI Reference : elasticache</a> 區段和 <a href="#">Amazon ElastiCache API Reference</a> 中的文件結構保持不變。	2018 年 4 月 20 日
<a href="#">支援 EngineCPUUtilization 指標</a>	ElastiCache ( Redis OSS ) 新增了指標 EngineCPU Utilization，報告目前使用中 CPU容量的百分比。如需詳細資訊，請參閱 <a href="#">Redis 的指標。OSS</a>	2018 年 4 月 9 日

下表說明 ElastiCache 使用者指南在 2018 年 3 月之前的重要變更。



變更	描述	變更日期
支援亞太區域 (大阪—當地)。	<p>ElastiCache 已新增對亞太區域 (大阪當地) 區域的支援。亞太區域 (大阪) 目前支援單一可用區域，且僅能透過邀請來取得。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2018 年 2 月 12 日
支援歐洲 (巴黎)。	<p>ElastiCache 已新增對 EU (巴黎) 區域的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2017 年 12 月 18 日
支援中國 (寧夏) 區域	<p>Amazon ElastiCache 已新增對中國 (寧夏) 區域的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2017 年 12 月 11 日
支援服務連結角色	<p>此版本 ElastiCache 新增了對服務連結角色 ( ) 的支援SLR。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">使用 Amazon 的服務連結角色 ElastiCache</a></li> <li>• <a href="#">設定您的許可 (僅限新 ElastiCache 使用者)</a></li> </ul>	2017 年 12 月 7 日
支援 R4 節點類型	<p>此版本的 ElastiCache 新增支援支援的所有 AWS 區域中的 R4 節點類型 ElastiCache。您能以隨需或預留快取節點形式購買 R4 節點類型。如需詳細資訊，請參閱下列內容：</p>	2017 年 11 月 20 日

變更	描述	變更日期
	<ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> <li>• <a href="#">Redis OSS節點類型特定參數</a></li> </ul>	
ElastiCache ( Redis OSS ) 3.2.10 和支援線上轉印	<p>Amazon ElastiCache ( Redis OSS ) 新增對 ElastiCache ( Redis OSS ) 3.2.10 的支援。ElastiCache ( Redis OSS ) 也引進線上叢集調整大小，以在叢集繼續提供傳入 I/O 請求時，從叢集新增或移除碎片。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">線上叢集大小調整</a></li> <li>• <a href="#">Valkey 或 Redis 的線上重新共享 OSS ( 啟用叢集模式 )</a></li> </ul>	2017 年 11 月 9 日
HIPAA 資格	<p>ElastiCache ( Redis OSS ) 3.2.6 版現在在您的叢集上啟用加密時已通過HIPAA資格認證。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">Amazon 的合規驗證 ElastiCache</a></li> <li>• <a href="#">Amazon 中的資料安全 ElastiCache</a></li> </ul>	2017 年 11 月 2 日

變更	描述	變更日期
ElastiCache ( Redis OSS ) 3.2.6 和 支援加密	<p>ElastiCache 新增對 ElastiCache ( Redis OSS ) 3.2.6 的支援，其中包含兩個加密功能：</p> <ul style="list-style-type: none"> <li>• 傳輸中加密會在您的資料傳輸時進行加密，例如在叢集內的節點間傳輸，或是在叢集與應用程式間傳輸。</li> <li>• 待用加密可在同步和備份操作期間加密您磁碟上的資料。</li> </ul> <p>如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">Amazon 中的資料安全 ElastiCache</a></li> <li>• <a href="#">支援的引擎和版本</a></li> </ul>	2017 年 10 月 25 日
連接模式主題	<p>ElastiCache 文件會新增主題，涵蓋存取 ElastiCacheAmazon 中叢集各種模式 VPC。</p> <p>如需詳細資訊，請參閱 使用者指南 <a href="#">在 Amazon 中存取 ElastiCache 快取的存取模式 VPC</a> 中的 ElastiCache</p>	2017 年 4 月 24 日
支援 Memcached 1.4.34	<p>ElastiCache 新增 支援 Memcached 1.4.34 版，該版本將許多修正納入較早的 Memcached 版本。</p> <p>如需詳細資訊，請參閱 上的 <a href="#">Memcached 的 Memcached 1.4.34 版本備註</a>GitHub。</p>	2017 年 4 月 10 日

變更	描述	變更日期
支援測試自動容錯移轉	<p>ElastiCache 新增了測試支援複寫之 Redis OSS 叢集上的自動容錯移轉的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">測試自動容錯移轉</a> (在 ElastiCache 使用者指南中)</li> <li>• <a href="#">TestFailover</a> 在ElastiCache API參考 中。</li> <li>• AWS CLI 參考中的 <a href="#">test-failover</a>。</li> </ul>	2017 年 4 月 4 日
增強 Redis OSS 還原	<p>ElastiCache 新增了增強的 Redis OSS 備份和還原，並採用叢集大小調整。此功能支援將備份還原至與用來建立備份的叢集具有不同碎片數的叢集。(對於 API 和 CLI，此功能可以還原不同數量的節點群組，而不是不同數量的碎片。) 此更新也支援不同的 Redis OSS 插槽組態。如需詳細資訊，請參閱<a href="#">從備份還原到新的快取</a>。</p>	2017 年 3 月 15 日
新的 Redis OSS 記憶體管理參數	<p>ElastiCache 新增了新的 Redis OSS 參數 <code>reserved-memory-percent</code>，這可讓您更輕鬆地管理預留記憶體。此參數適用於所有版本的 ElastiCache (Redis OSS)。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">管理 Valkey 和 Redis 的預留記憶體 OSS</a></li> <li>• <a href="#">Redis OSS 3.2.4 的新參數</a></li> </ul>	2017 年 3 月 15 日
支援 Memcached 1.4.33	<p>ElastiCache 新增對 Memcached 1.4.33 版的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">Memcached 1.4.33 版</a></li> <li>• <a href="#">Memcached 1.4.33 版新增參數</a></li> </ul>	2016 年 12 月 20 日

變更	描述	變更日期
支援西歐 (倫敦) 區域	<p>ElastiCache 新增對 EU ( 倫敦 ) 區域的支援。目前只支援 T2 和 M4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"><li>• <a href="#">支援的區域</a></li><li>• <a href="#">支援的快取節點類型</a></li></ul>	2016 年 12 月 13 日
支援加拿大 (蒙特婁) 區域	<p>ElastiCache 新增對加拿大 ( 蒙特婁 ) 區域的支援。此 AWS 區域目前僅支援節點類型 M4 和 T2。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"><li>• <a href="#">支援的區域</a></li><li>• <a href="#">支援的快取節點類型</a></li></ul>	2016 年 12 月 8 日
支援 M4 和 R3 節點類型	<p>ElastiCache 新增對南美洲 ( 聖保羅 ) 區域的 R3 和 M4 節點類型和中國 ( 北京 ) 區域的 M4 節點類型的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"><li>• <a href="#">支援的區域</a></li><li>• <a href="#">支援的快取節點類型</a></li></ul>	2016 年 11 月 1 日
美國東部 2 (俄亥俄) 區域支援	<p>ElastiCache 新增對 M4, T22 和 R3 節點類型的美國東部 ( 俄亥俄 ) 區域 ( us-east-2 ) 的支援。R3 如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"><li>• <a href="#">支援的區域</a></li><li>• <a href="#">支援的快取節點類型</a></li></ul>	2016 年 10 月 17 日

變更	描述	變更日期
支援 Redis OSS 叢集	<p>ElastiCache 新增對 Redis OSS Cluster ( 增強版 ) 的支援。使用 Redis OSS Cluster 的客戶可以將資料分割為最多 15 個碎片 ( 節點群組 )。每個碎片支援的複寫為每個碎片最多 5 個僅供讀取複本。Redis OSS Cluster 自動容錯移轉時間約為四分之一，與舊版相同。</p> <p>此版本包括重新設計的管理主控台，並使用符合業界使用方式的術語。</p> <p>如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">比較 Memcached 和 Redis OSS</a></li> <li>• <a href="#">ElastiCache 元件和功能</a> - 請注意說明節點、碎片、叢集及複寫的小節。</li> <li>• <a href="#">ElastiCache 術語</a></li> </ul>	2016 年 10 月 12 日
支援 M4 節點類型	<p>ElastiCache 在支援的大多數 AWS 區域中，新增了對 M4 系列節點類型的支援。您能以隨需或預留快取節點形式購買 M4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> <li>• <a href="#">Redis OSS 節點類型特定參數</a></li> </ul>	2016 年 8 月 3 日
支援孟買地區	<p>ElastiCache 新增對亞太區域 ( 孟買 ) 區域的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> <li>• <a href="#">Redis OSS 節點類型特定參數</a></li> </ul>	2016 年 6 月 27 日

變更	描述	變更日期
快照匯出	<p>ElastiCache 新增匯出 Redis OSS快照的功能，以便您可以從外部存取它ElastiCache。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">匯出備份</a> Amazon ElastiCache 使用者指南中的</li> <li>• <a href="#">CopySnapshot</a> Amazon ElastiCache API 參考中的</li> </ul>	2016 年 5 月 26 日
節點類型擴展	ElastiCache 新增擴展 Redis OSS節點類型的功能。如需詳細資訊，請參閱 <a href="#">擴展 ElastiCache</a> 。	2016 年 3 月 24 日
輕鬆升級引擎	ElastiCache 新增輕鬆升級 Redis OSS快取引擎的功能。如需詳細資訊，請參閱 <a href="#">的版本管理 ElastiCache</a> 。	2016 年 3 月 22 日
支援 R3 節點類型	ElastiCache 在中國（北京）區域和南美洲（聖保羅）區域中新增了對 R3 節點類型的支援。如需詳細資訊，請參閱 <a href="#">支援的快取節點類型</a> 。	2016 年 3 月 16 日
ElastiCache 使用 Lambda 函數存取	新增了有關設定 Lambda 函數以在 Amazon ElastiCache 中存取的教學課程VPC。如需詳細資訊，請參閱 <a href="#">其他 ElastiCache 教學課程和影片</a> 。	2016 年 2 月 12 日
支援 Redis OSS 2.8.24	<p>ElastiCache 新增了對 Redis 2.8.24 OSS版的支援，並新增了自 Redis 2.8.23 OSS 以來的改進。改良功能包括錯誤修正以及對惡意記憶體存取地址的記錄支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">ElastiCache ( Redis OSS ) 2.8.24 版 ( 增強版 )</a></li> <li>• <a href="#">Redis OSS 2.8 版本備註</a></li> </ul>	2016 年 1 月 20 日
支援亞太區域 (首爾)	ElastiCache 新增對具有 t2、m3 和 r3 節點類型的亞太區域（首爾）（ap-northeast-2）區域的支援。	2016 年 1 月 6 日

變更	描述	變更日期
Amazon ElastiCache 主控台變更。	由於較新的 Redis OSS 版本提供更好且更穩定的使用者體驗，因此 Redis 2.6.13、2.8.6 和 2.8.19 OSS 版不再列在 ElastiCache 管理主控台中。如需其他選項和詳細資訊，請參閱 <a href="#">支援的引擎和版本</a> 。	2015 年 12 月 15 日
支援 Redis OSS 2.8.23。	ElastiCache 新增了對 Redis 2.8.23 OSS 版的支援，並新增了自 Redis 2.8.22 OSS 以來的改進。改良功能包括錯誤修正以及對新參數 <code>close-on-slave-write</code> 的支援，若啟用此參數，則會在用戶端嘗試寫入唯讀複本時中斷其連線。如需詳細資訊，請參閱 <a href="#">ElastiCache ( Redis OSS ) 2.8.23 版 ( 增強版 )</a> 。	2015 年 11 月 13 日
支援 Redis OSS 2.8.22。	<p>ElastiCache 新增了對 Redis 2.8.22 OSS 版的支援，並 ElastiCache 新增了自 2.8.21 版以來的增強功能和改進功能。改良功能包括：</p> <ul style="list-style-type: none"> <li>• 實作無分支儲存程序；當可用記憶體偏少可能導致無分支儲存失敗時，此程序可確保成功儲存。</li> <li>• 其他 CloudWatch 指標 — <code>SaveInProgress</code> 和 <code>ReplicationBytes</code>。</li> <li>• 若要啟用部分同步，Redis OSS 參數 <code>repl-backlog-size</code> 現在適用於所有叢集。</li> </ul> <p>如需完整變更清單和詳細資訊，請參閱 <a href="#">ElastiCache ( Redis OSS ) 2.8.22 版 ( 增強版 )</a>。</p> <p>此文件版本包括文件的重組和 ElastiCache 命令列界面 ( CLI ) 文件的移除。如需使用命令列，請參閱的 <a href="#">AWS 命令列</a> ElastiCache。</p>	2015 年 9 月 28 日



變更	描述	變更日期
支援 Memcached 1.4.28。	ElastiCache 新增對 Memcached 1.4.24 版和 Memcached 自 1.4.14 版以來改進的支援。此版本新增了對最近使用至少 (LRU) 快取管理作為背景任務的支援，選擇 jenkins 或 murm3 作為雜湊演算法、新命令和其他錯誤修正。如需詳細資訊，請參閱 <a href="#">Memcached 版本備註</a> 。	2015 年 8 月 27 日
使用 5.6 PHP 支援 Memcached Auto Discovery	此版本的 Amazon ElastiCache 新增對 5.6 PHP 版 Memcached Auto Discovery 用戶端的支援。如需詳細資訊，請參閱 <a href="#">編譯 ElastiCache 叢集用戶端的原始程式碼 PHP</a> 。	2015 年 7 月 29 日
支援 Redis OSS 2.8.21	ElastiCache 新增對 Redis OSS 版和 Redis 自 2.8.21 版以來 OSS 改進的支援 2.8.19。此 Redis OSS 版本包含數個錯誤修正。如需詳細資訊，請參閱 <a href="#">Redis OSS 2.8 版本備註</a> 。	2015 年 7 月 29 日
新主題： ElastiCache 從外部存取 AWS	新增了有關如何從外部存取 ElastiCache 資源的新主題 AWS。如需詳細資訊，請參閱 <a href="#">ElastiCache 從外部存取 AWS</a> 。	2015 年 7 月 9 日
新增節點取代訊息	ElastiCache 新增三個與排程節點取代相關的訊息：ElastiCacheNodeReplacementScheduled、ElastiCache：NodeReplacementRescheduled 和 ElastiCache：NodeReplacementCanceled。  如需節點排定取代時可採取的詳細資訊和動作，請參閱 ElastiCache 的 <a href="#">事件通知</a> 和 <a href="#">Amazon SNS</a> 。	2015 年 6 月 11 日

變更	描述	變更日期
支援 Redis OSS v. 2.8.19。	<p>ElastiCache 新增對 Redis 2.8.19 OSS版和 Redis 2.8.6 版以來OSS改進的支援。此支援也包含下列支援：</p> <ul style="list-style-type: none"> <li>• HyperLogLog 資料結構，搭配 Redis OSS命令 PFADD、PFCOUNT和 PFMERGE。</li> <li>• 使用新命令 ZRANGEBYLEX、ZLEXCOUNT和的 Lexicographic 範圍查詢ZREMRANGEBYLEX。</li> <li>• 引進了許多錯誤修正，也就是當背景儲存（bgsave）子程序意外終止SYNC時，透過失敗主節點，防止主節點傳送過時的資料至複本節點。</li> </ul> <p>如需的詳細資訊 HyperLogLog，請參閱 <a href="#">Redis OSS新資料結構：。HyperLogLog</a></p> <p>如需 PFADD、PFCOUNT和的詳細資訊 PFMERGE，請參閱 <a href="#">Redis OSS 文件</a>並按一下 HyperLogLog。</p>	2015 年 3 月 11 日
支援成本分配標籤	ElastiCache 新增對成本分配標籤的支援。如需詳細資訊，請參閱 <a href="#">使用成本配置標籤監控成本</a> 。	2015 年 2 月 9 日
支援 AWS GovCloud（美國西部）區域	ElastiCache 新增對 AWS GovCloud（美國西部）（us-gov-west-1）區域的支援。	2015 年 1 月 29 日
支援歐洲（法蘭克福）區域	ElastiCache 新增對歐洲（法蘭克福）（eu-central-1）區域的支援。	2015 年 1 月 19 日

變更	描述	變更日期
Redis OSS複寫群組的多可用區支援	ElastiCache 將對多可用區從主要節點的支援新增至 Redis 複OSS寫群組中的僅供讀取複本。ElastiCache 監控複寫群組的運作狀態。如果主要故障，ElastiCache 會自動將複本提升為主要複本，然後取代複本。如需詳細資訊，請參閱 <a href="#">搭配 Valkey 和 Redis ElastiCache 使用多可用區域，將中的停機時間降到最低 OSS</a> 。	2014 年 10 月 24 日
AWS CloudTrail 記錄支援的API呼叫	ElastiCache 新增使用 AWS CloudTrail 記錄所有 ElastiCache API呼叫的支援。如需詳細資訊，請參閱 <a href="#">使用記錄 Amazon ElastiCache API 呼叫 AWS CloudTrail</a> 。	2014 年 9 月 15 日
支援新的執行個體大小	ElastiCache 新增對其他一般用途（ T2 ）執行個體的支援。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎 ElastiCache 參數</a> 。	2014 年 9 月 11 日
Memcached 支援更靈活的節點置放	ElastiCache 新增跨多個可用區域建立 Memcached 節點的支援。	2014 年 7 月 23 日
支援新的執行個體大小	ElastiCache 新增對其他一般用途（ M3 ）執行個體和記憶體最佳化（ R3 ）執行個體的支援。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎 ElastiCache 參數</a> 。	2014 年 7 月 1 日
PHP 自動探索	新增 5.5 PHP版自動探索的支援。	2014 年 5 月 13 日
Redis OSS叢集的備份和還原	在此版本中，ElastiCache 可讓客戶建立 Redis OSS叢集的快照，並使用這些快照建立新叢集。備份是特定時間點的叢集複本，由叢集中繼資料和 Redis OSS快取中的所有資料組成。備份是儲存在 Amazon S3 中，客戶可以隨時透過快照將資料還原到新的叢集中。如需詳細資訊，請參閱 <a href="#">快照和還原</a> 。	2014 年 4 月 24 日

變更	描述	變更日期
Redis OSS 2.8.6	ElastiCache 除了 Redis OSS 2.6.13 之外，還支援 Redis OSS 2.8.6。使用 Redis OSS 2.8.6，客戶可以改善僅供讀取複本的復原能力和容錯能力，並支援部分重新同步，以及使用者定義的僅供讀取複本數目下限，必須隨時可用。Redis OSS 2.8.6 也提供完整的支援 publish-and-subscribe，其中用戶端可以收到伺服器上發生的事件通知。	2014 年 3 月 13 日
Redis OSS 快取引擎	<p>ElastiCache 除了 Memcached 之外，還提供 Redis OSS 快取引擎軟體。目前使用 Redis 的客戶 OSS 可以將新的 ElastiCache Redis OSS 快取叢集與現有資料從 Redis OSS 快照檔案進行「種子」，以減輕遷移至受管 ElastiCache 環境的次數。</p> <p>若要支援 Redis OSS ElastiCache API 複寫功能，現在支援複寫群組。客戶可以使用主要 Redis OSS 快取節點建立複寫群組，並新增一或多個可自動與主要節點中快取資料同步的僅供讀取複本節點。您可以將讀取密集型應用程式卸載至僅供讀取複本，以降低主要節點的負載。僅供讀取複本也可以在主要快取節點故障時保護資料，避免遺失。</p>	2013 年 9 月 3 日
支援預設 Amazon Virtual Private Cloud (VPC)	在此版本中，ElastiCache 與 Amazon Virtual Private Cloud ( ) 完全整合 VPC。對於新客戶，VPC 預設會在 Amazon 中建立快取叢集。如需詳細資訊，請參閱 <a href="#">Amazon VPCs 和安全性 ElastiCache</a> 。	2013 年 1 月 8 日
PHP 支援快取節點自動探索	快取節點自動探索的初始版本已提供對 Java 程式的支援。在此版本中，ElastiCache 將快取節點自動探索支援帶至 PHP。	2013 年 1 月 2 日

變更	描述	變更日期
支援 Amazon Virtual Private Cloud ( VPC )	在此版本中，叢集可以在 ElastiCache Amazon Virtual Private Cloud ( ) 中啟動VPC。根據預設，新的客戶快取叢集會自動在 Amazon 中建立 VPC；現有客戶可以按照VPC自己的速度遷移至 Amazon。如需詳細資訊，請參閱 <a href="#">Amazon VPCs和安全性 ElastiCache</a> 。	2012 年 12 月 20 日
快取節點自動探索和新的快取引擎版本	ElastiCache 提供快取節點自動探索 - 用戶端程式自動判斷叢集中所有快取節點，以及啟動和維護所有這些節點連線的能力。  此版本也提供新的快取引擎版本：Memcached 1.4.14 版。這款新的快取引擎提供增強型 slab 重新平衡功能、效能和可擴展性大幅提升，以及多項錯誤修正。您可以設定多種新的快取參數。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎 ElastiCache 參數</a> 。	2012 年 11 月 28 日
新增快取節點類型	本版本提供四種額外的快取節點類型。	2012 年 11 月 13 日
預留快取節點	此版本新增對預留快取節點的支援。 q	2012 年 4 月 5 日
新指南	這是 Amazon ElastiCache 使用者指南 的第一個版本。	2011 年 8 月 22 日

# AWS 詞彙表

如需最新的 AWS 術語，請參閱AWS 詞彙表 參考 中的[AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。