



在 Amazon RDS for My SQL、Amazon RDS for MariaDB 和 Aurora My SQL-
Compatible 中封存資料

AWS 規範指南



AWS 規範指南: 在 Amazon RDS for My SQL、Amazon RDS for MariaDB 和 Aurora My SQL-Compatible 中封存資料

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

簡介	1
概要	1
目標成果	2
從分區資料表封存	4
從未分區的表歸檔	6
將資料移至 Amazon S3	7
使用 SELECT INTO OUTFILE S3	7
使用 AWS Glue	8
存取封存資料	11
標準儲存類別	11
S3 Glacier 儲存類別	12
最佳實務	13
清除	14
資源	15
附錄一	16
附錄二	18
文件歷史紀錄	21
詞彙表	22
#	22
A	22
B	25
C	27
D	29
E	33
F	34
G	35
H	36
I	37
L	39
M	40
O	43
P	45
Q	47
R	48

S	50
T	53
U	54
V	55
W	55
Z	56
.....	lvii

將資料存檔到適用於 MySQL 的亞馬遜 RDS、亞馬遜 RDS 和極光與 MySQL 相容

Shyam Sunder Rakhecha，阿布舍克·卡爾瑪卡爾，奧利弗·弗朗西斯和索姆亞馬遜辛格亞馬遜 Web 服務 (AWS)

二零二三年四月([文件歷史](#))

存檔歷史資料的需求可能來自不同的使用案例。您的應用程式設計可能沒有封存功能，而且隨著時間的推移業務成長可能會產生大量的歷史資料。這不可避免地導致性能下降。由於組織內的合規性要求，您也可能會保留歷史資料。

本指南討論如何將歷史資料存檔到 Amazon 簡單儲存服務 (Amazon S3)，對應用程式的影響最小，並在需要時擷取存檔的資訊。

概要

本指南涵蓋了在亞馬遜網路服務 (AWS) 雲端上存檔來自亞馬遜關聯式資料庫服務 (Amazon RDS)、亞馬遜 RDS (適用於 MariaDB) 和亞馬遜極光 MySQL 相容版本的大型表格歷史資料的不同方法。在本指南中，您將學習如何封存分區資料表資料，以及未分割且常駐在大型資料表中的資料。您可以實作指南中提供的方法，以減少即時資料的大小，同時保留重要的歷史資料以供進一步分析。

定期封存資料表資料可在資料表中產生一組更精細的即時資料，進而加快讀取和寫入速度，並改善應用程式的效能。定期資料封存屬於卓越營運和效能效率的支柱[架構良好的框架](#)。當您將舊資料移至 Amazon 簡單儲存服務 (Amazon S3)，並清理 Amazon RDS 執行個體或 Aurora MySQL 相容叢集中的存檔資料時，可以節省儲存成本。這符合成本優化支柱，可協助您避免 AWS 上不必要的成本。

目標業務成果

本指南著重於以下業務成果：

- 改善使用者體驗
- 符合資料合規要求
- 降低儲存成本
- 有組織數據

改善使用者體驗

保留歷史資料的資料庫可能會因為資料表和索引較大而降低效能。當您封存歷史資料時，您會縮小資料表和索引。這會對與資料庫互動的面向客戶 API 作業產生直接的正面影響。

符合資料合規要求

金融服務、公共部門組織和醫療保健等行業都有嚴格的檔案要求。透過將常駐於 Amazon RDS for MySQL、亞馬遜 RDS (適用於 MariaDB) 或 Aurora MySQL 相容資料庫上的應用程式資料存檔到 Amazon S3，您就可以滿足受監管合規的要求，包括下列各項：

- 支付卡產業資料安全標準 (PCI DSS)
- 《健康保險可攜性與責任法案》(HIPAA) 及《經濟與臨床健康健康資訊科技 (HITECH) 法案》
- 聯邦風險與授權管理計劃 (FedRAMP)
- 一般資料保護規範 (GDPR)
- 美國聯邦資訊處理標準
- 美國國家標準與技術研究所

降低儲存成本

在 Amazon RDS 中保留資料可增加儲存成本，並且需要更高的 IOPS。如果您比較每月 GB 的儲存成本 [亞馬遜 RDS 適用於 MySQL 異地同步備份 GP2](#) 與的 [亞馬遜 S3 冰川](#) 在 us-east-1 AWS 區域中，S3 冰川儲存成本比亞馬遜 RDS 低 57 倍左右。

有組織數據

這是很好的保持信息數據，這些數據將經常被數據庫中的應用程序訪問。但是，應用程序會生成大量數據，這些數據不經常需要或變得陳舊。這些記錄可以存檔並保存到位，這是符合成本效益的，不會影響應用程式效能。

封存分區資料表中的資料

支持[分區](#)對於 InnoDB 存儲引擎，您可以使用此功能對大表進行分區。雖然在資料分割資料表上運作的 SQL 會讀取整個資料表，但資料表中的資料分割會儲存為個別的實體資料表。這使您可以自由地從表中刪除不需要的分區，而無需執行 row-by-row 刪除，以便您可以在數據庫中存檔歷史行。

請考慮下面的示例代碼。TABLE orders 存在於 orderprocessing 結構描述。它的歷史數據存在於分區中 phistorical，其中包含屬於 2021 年及更早版本的數據。在同一個表格中，應用程式層級的熱資料會出現在 2022 年每個月的即時分割區中。封存分割區中的資料 phistorical，您可以創建一個歸檔 table orders_2021_and_older 具有相同的結構 archive 結構描述。然後，您可以使用 MySQL [交換分區](#) 移動分割區 phistorical 進入那張桌子。請注意，封存資料表不會進行分割。存檔後，您可以驗證資料並將其移至 Amazon S3。

```
CREATE TABLE orders (
  orderid bigint NOT NULL AUTO_INCREMENT,
  customerid bigint DEFAULT NULL,
  .....
  .....
  order_date date NOT NULL,
  PRIMARY KEY (`orderid`, `order_date`))
PARTITION BY RANGE (TO_DAYS(order_date)) (
  PARTITION pstart VALUES LESS THAN (0),
  PARTITION phistorical VALUES LESS THAN (TO_DAYS('2022-01-01')),
  PARTITION p2022JAN VALUES LESS THAN (TO_DAYS('2022-02-01')),
  PARTITION p2022FEB VALUES LESS THAN (TO_DAYS('2022-03-01')),
  PARTITION p2022MAR VALUES LESS THAN (TO_DAYS('2022-04-01')),
  PARTITION p2022APR VALUES LESS THAN (TO_DAYS('2022-05-01')),
  PARTITION p2022MAY VALUES LESS THAN (TO_DAYS('2022-06-01')),
  PARTITION p2022JUN VALUES LESS THAN (TO_DAYS('2022-07-01')),
  PARTITION p2022JUL VALUES LESS THAN (TO_DAYS('2022-08-01')),
  PARTITION p2022AUG VALUES LESS THAN (TO_DAYS('2022-09-01')),
  PARTITION p2022SEP VALUES LESS THAN (TO_DAYS('2022-10-01')),
  PARTITION p2022OCT VALUES LESS THAN (TO_DAYS('2022-11-01')),
  PARTITION p2022NOV VALUES LESS THAN (TO_DAYS('2022-12-01')),
  PARTITION p2022DEC VALUES LESS THAN (TO_DAYS('2023-01-01')),
  PARTITION pfuture VALUES LESS THAN MAXVALUE
);
CREATE TABLE orders_2021_and_older (
  orderid bigint NOT NULL AUTO_INCREMENT,
  customerid bigint DEFAULT NULL,
  .....
```



```
.....
order_date date NOT NULL,
PRIMARY KEY (`orderid`, `order_date`));
mysql> alter table orderprocessing.orders exchange partition phistorical with table
archive.orders_2021_and_older;
Query OK, 0 rows affected (0.33 sec)
```

當您使用EXCHANGE PARTITION歸檔歷史數據的功能，我們建議採用以下最佳實踐：

- 建立個別的結構描述，以便在應用程式中儲存封存資料。此結構描述將包含存放封存資料的存檔表格。歸檔結構描述中的封存資料表結構應該與即時資料表具有相同的結構，包括其索引和主索引鍵。不過，目標封存資料表不能是資料分割資料表。MySQL 中不允許在兩個分區表之間交換分區。
- 請遵循存檔表格的命名慣例，以協助您識別儲存在其中的歷史資料。當您執行稽核任務或設計將此資料移出到 Amazon S3 的任務時，這非常有用。
- 執行EXCHANGE PARTITION當沒有流量進入您的 Aurora MySQL 相容寫入器、適用於 MySQL 的 Amazon RDS 或適用於 MariaDB 執行個體的 Amazon RDS 時，停機時間視窗中的資料定義語言 (DDL) 陳述式。

它可能是可以運行EXCHANGE PARTITION在應用程式或微服務中的低流量視窗期間。但是，應該沒有寫入，也不應該在分區表上選擇或很少選擇。現有的長時間運行選擇查詢可能會導致EXCHANGE PARTITIONDDL 等待，導致資料庫上的資源爭用。運行之前驗證所有這些條件都符合的設計腳本EXCHANGE PARTITION在您的系統上。

如果您的應用程式設計可以支援分區資料，而您目前有未分割的資料表，請考慮將資料移至分區資料表，以支援將資料歸檔。如需詳細資訊，請參閱 [MySQL 文件](#)。

封存未分割資料表中的資料

在不可能進行分區的數據庫表中，您可以使用 Percona 工具包 [PT-存檔](#) 將表的數據存檔到 MySQL 數據庫中的另一個表中的工具。

PT-Archiver 工具用於將記錄從大型表存檔到其他表或文件。這是一個讀/寫工具，這意味著它在存檔後從源表中刪除數據，因此您不必單獨管理源數據刪除。這個腳本的主要目的是在不影響現有的在線交易處理 (OLTP) 查詢負載 (見附錄 I) 的情況下，將數據插入到相同或不同服務器上的另一個表中。

您可以下載 [佩科納工具包](#) 和 [安裝](#) 它可以在您的本機電腦或 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上連線到資料庫的執行個體上。若要執行 PT 封存器工具，請使用下列語法。

```
pt-archiver --source h=<HOST>,D=<DATABASE>,t=<TABLE>,u=<USER>,p=<PASSWORD> --dest  
h=<HOST>,D=<DATABASE>,t=<TABLE> --where "'1=1'" --statistics
```

更換 HOST, DATABASE, TABLE，以及 USER 與您的源和目標數據庫的詳細信息和憑據。

您也可以使用 [AWS 批次](#) 以建立並排程表格的工作。

當您使用 pt-archiver 工具來封存資料表的資料時，請考慮下列事項：

- 在源表上使用主鍵將提高此工具的性能。如果表沒有主鍵，則可以 [創建一個唯一列的索引](#)，這將有助於 PT-歸檔器去通過表中的所有行並歸檔它們。
- 默認情況下，pt-Archiver 在歸檔表後刪除數據。在生產伺服器上執行之前，請務必使用下列方式測試您的封存工作 --dry-run。或者，您可以使用 --no-delete 選項。
- PT-Archiver 工具會根據您系統的負載來調整封存速度 (請參閱附錄 II)。負載較高時，封存效能可能會降低。

執行 pt-archiver 之後，您的封存資料應該位於歸檔結構描述中的對應資料表中。從那裡，您可以將其移動到亞馬遜 S3。

將封存的資料表資料移至 Amazon S3

Amazon Simple Storage Service (Amazon S3) 是封存資料的自然目標。它提供 99.999999999% 的耐用性，而且比資料庫儲存便宜。

此外，Amazon S3 具有內建儲存類別，根據擷取模式定價。您可以根據資料的擷取頻率，選擇將卸載的 S3 物件轉換為價格較低的儲存層。如需儲存類別和定價的詳細資訊，請參閱 [Amazon S3 文件](#)。

對於使用 MySQL 執行個體機群的應用程式，卸載至 Amazon S3 意味著在符合下列條件的資料上節省成本：

- 必須從資料庫封存，以提高效率
- 不需要立即執行，或對任何業務流程而言是少量需要的
- 由於稽核需求，必須長期保留

您可以透過下列方式封存我的 SQL 資料：

- 使用 匯出資料 `SELECT INTO OUTFILE S3`。
- 使用 AWS Glue 匯出資料。

使用 SELECT INTO OUTFILE S3 匯出資料

若要將資料直接從 Aurora My SQL-Compatible 資料庫複製到 Amazon S3，您可以使用 陳述式 `SELECT INTO OUTFILE S3`。此 SQL 陳述式可以在 資料表上執行，而且所需的資料列數目可以卸載為大小上限為 6 GB 的逗號分隔值 (CSV) 檔案。超過 6 GB 閾值時，會建立多個 .csv 檔案。

若要讓匯出正常運作，請設定下列項目：

- AWS Identity and Access Management (IAM) 角色和政策
- 授予使用者執行命令的資料庫許可
- 用於卸載的 Amazon S3 位置

如需詳細資訊，請參閱 [Amazon Aurora 文件](#)。

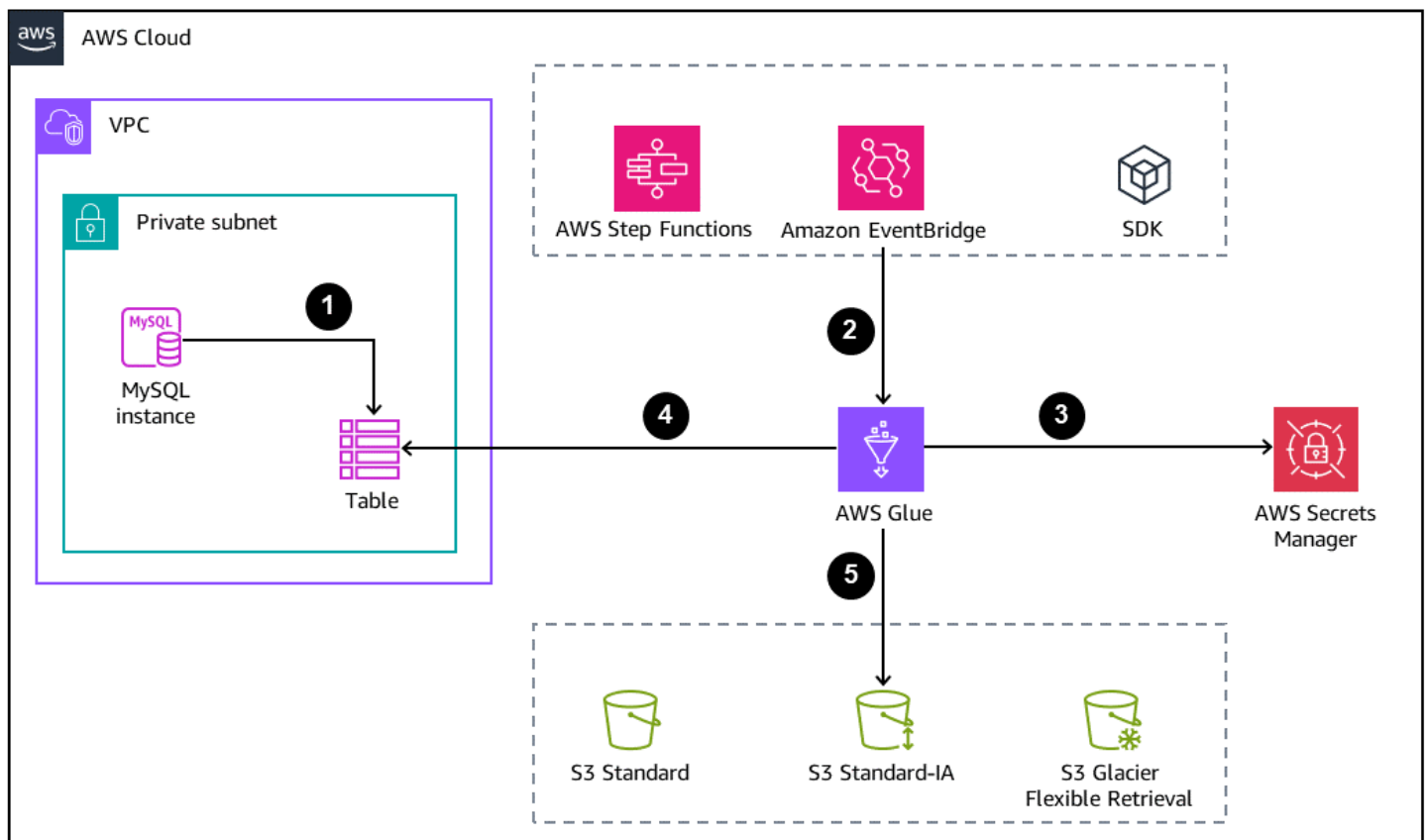
注意：若要將 S3 物件轉換為節省成本的儲存層，建議您在匯出資料的 [S3 儲存貯體中設定 Amazon S3 生命週期規則](#)。S3

使用 AWS Glue 匯出資料

您可以使用 AWS Glue 在 Amazon S3 中封存我的 SQL 資料，這是適用於大數據案例的無伺服器分析服務。AWS Glue 由 Apache Spark 提供支援許多資料庫來源的廣泛使用分散式叢集運算架構提供技術支援。

可以使用 AWS Glue 任務中的幾行程式碼，從資料庫卸載封存資料至 Amazon S3。AWS Glue 提供的最大優勢是水平可擴展性和模型 pay-as-you-go，可提供操作效率和成本最佳化。

下圖顯示資料庫封存的基本架構。



1. 我的 SQL 資料庫會建立要在 Amazon S3 中卸載的封存或備份資料表。
2. Glue AWS 任務是由下列其中一種方法啟動：
 - 同步作為 [AWS Step Functions](#) 狀態機器中的步驟
 - Amazon [EventBridge](#) 事件非同步

- 透過使用 AWSCLI 或 的手動請求 [AWS SDK](#)
3. 資料庫憑證是從 [AWS Secrets Manager](#) 擷取。
 4. Glue AWS 任務使用 Java Database Connectivity (JDBC) 連線來存取資料庫，並讀取資料表。
 5. AWS Glue 會以 Parquet 格式將資料寫入 Amazon S3，這是一種開放、單欄、節省空間的資料格式。

設定 Glue AWS 任務

若要如預期般運作，Glue AWS 任務需要下列元件和組態：

- [AWS Glue 連線](#) – 這是您附加至任務以存取資料庫的 AWS Glue Data Catalog 物件。任務可以具有多個連線，用於對多個資料庫進行呼叫。連線包含安全儲存的資料庫憑證。
- [GlueContext](#) – 這是 [SparkContext](#) GlueContext 類別上的自訂包裝，提供與 Amazon S3 和資料庫來源互動的高階 API 操作。它啟用與 Data Catalog 的整合。它也消除了依賴驅動程式進行資料庫連線的需求，該連線是在 Glue 連線中處理。此外，GlueContext 類別提供處理 Amazon S3 API 操作的方法，這在原始 SparkContext 類別中是不可能的。
- IAM 政策和角色 – 由於 AWS Glue 與其他 AWS 服務互動，因此您必須設定具有最低權限的適當角色。需要適當許可才能與 Glue AWS 互動的服務包括：
 - Amazon S3
 - AWS Secrets Manager
 - AWS 金鑰管理服務 (AWS KMS)

最佳實務

- 若要讀取要卸載大量資料列的整個資料表，建議您使用僅供讀取複本端點來提高讀取輸送量，而不會降低主要寫入器執行個體的效能。
- 若要在用於處理任務的節點數量中提高效率，請在 Glue 3.0 AWS 中開啟 [自動擴展](#)。
- 如果 S3 儲存貯體是資料湖架構的一部分，我們建議您將資料組織為實體分割區來卸載資料。分割區配置應基於存取模式。根據日期值進行分割是最常建議的作法之一。
- 將資料儲存為開放格式，例如 Parquet 或 Optimized Row Columnar (ORC)，有助於將資料提供給其他分析服務，例如 Amazon Athena 和 Amazon Redshift。
- 若要讓其他分散式服務對卸載的資料進行讀取最佳化，必須控制輸出檔案的數量。擁有較少數量的較大檔案，而不是大量小型檔案幾乎總是有利的。Spark 具有內建的組態檔案和方法來控制部分檔案的產生。

- 依定義的封存資料是經常存取的資料集。為了實現儲存的成本效率，Amazon S3 類別應轉換為價格較低的層。這可以使用兩種方法完成：
 - 卸載時同步轉換層 – 如果您事先知道卸載的資料必須作為程序的一部分進行轉換，則可以在將資料寫入 Amazon S3 的相同 AWS Glue 任務中使用 GlueContext 機制 `transition_s3_path`。
https://docs.aws.amazon.com/glue/latest/dg/aws-glue-api-crawler-pyspark-extensions-glue-context.html#aws-glue-api-crawler-pyspark-extensions-glue-context-transition_s3_path
 - 使用 [S3 生命週期](#) 非同步轉換 – 使用適當的參數設定 S3 生命週期規則，以進行 Amazon S3 儲存類別轉換和過期。在儲存貯體上設定此項目後，它將永遠保留。
- 在部署資料庫的虛擬私有雲端（VPC）內，建立並設定具有[足夠 IP 地址範圍](#)的子網路。這將避免在設定大量資料處理單位 AWS（DPUs）時，由於網路地址數量不足而導致 Glue 任務失敗。

在 Amazon S3 上存取封存資料

Amazon S3 提供許多工具來讀取資料的內容。不過，根據儲存類別，可能需要幾個預先處理步驟。本節包含下列項目：

- 使用 Glue AWS 讀取已封存的 S3 物件與標準儲存類別
- 使用 S3 批次操作，使用 S3 Glacier 儲存類別讀取封存的 S3 物件
- 最佳實務

使用標準儲存類別讀取封存的 S3 物件

使用 AWS Glue

從 MySQL 卸載至 Amazon S3 的資料會保留與關聯式資料庫管理系統 () 相同的結構剛性和一致性 RDBMS。

[AWS Glue Crawler](#) 爬過 S3 物件，推斷資料類型，並將資料表中繼資料建立為外部資料表 DDL。當您設定爬蟲程式工作時，請使用 Amazon S3 作為來源，並指定建立所有資料檔案的 S3 字首位置。在組態中，包含下列項目：

- Crawler 執行選項
- 選用的資料表字首偏好設定
- 建立資料表的目標資料庫
- IAM 具有必要許可的角色

叫用任務之後，它會掃描資料，以推斷結構描述並將其保留在 [AWS Glue Data Catalog](#) 中作為 [AWS Glue 資料表](#)。AWS Glue 資料表基本上是外部資料表，可使用分析服務查詢，例如 [Amazon Athena](#)、[Amazon Redshift Spectrum](#) 和 [Amazon EMR](#) 上的 Apache Hive 等 SQL 陳述式，例如一般資料庫資料表。如需爬蟲程式的詳細資訊，請參閱 [AWS Glue 文件](#)。

對於具有指定資料欄標頭的 .csv 檔案，產生的資料表資料欄名稱將反映相同的欄位名稱。根據資料物件中的值推斷資料類型。

對於 Parquet 檔案，結構描述會保留在資料本身內，而產生的資料表將反映相同的欄位名稱和資料類型。

或者，您可以在 Athena 中 DDL 手動執行，以使用所需的資料欄名稱和資料類型建立資料表定義。這會在 Data Catalog 中建立資料表定義。如需建立 Athena 資料表的詳細資訊，請參閱 [Amazon Athena 文件](#)。

注意：如果 CSV 檔案中缺少標頭列，爬蟲程式會將欄位名稱建立為一般 c_0、c_1、c_2、...

使用 Amazon S3 Select

您可以使用 Amazon S3 Select，透過 SQL 表達式以程式設計方式讀取 S3 物件。您可以使用 AWS CLI 命令 `select-object-content` 或使用 Boto3 SDK 等來叫用 API 操作，並從 `select_object_content` Python 叫用操作。

API 操作支援 SQL 陳述式做為參數，並且只能讀取類型 JSON 和 Parquet 的檔案。輸出可以重新導向為輸出檔案。

每個 S3 物件都會叫用這些操作。對於多個檔案，請遞迴執行操作。

如需使用 AWS 執行操作的詳細資訊 CLI，請參閱 [AWS CLI 文件](#)。如需使用 Python SDK Boto3 執行 S3 Select 的詳細資訊，請參閱 [Boto3 文件](#)。

使用 S3 Glacier 儲存類別讀取封存的 S3 物件

Amazon S3 Glacier 類別是特殊儲存類別，定價便宜，但擷取時間長。與 S3 標準物件不同，S3 Glacier 物件無法讀取為 Glue AWS 資料表。若要讓資料可用於分析查詢或報告，請先還原 S3 Glacier 物件。還原是一種非同步程序，會隨著時間發生，並具有保留期。還原物件後，可以將物件複製到與 S3 標準物件不同的位置。在保留期間之後，還原的物件會轉換回 Amazon S3 Glacier。

使用 S3 批次操作

S3 Batch Operations 會依數十億個包含 exabyte 資料的物件順序，在 Amazon S3 上啟用大規模批次操作。Amazon S3 會追蹤進度、傳送通知，並存放所有動作的詳細完成報告，提供完整受管、可稽核、無伺服器的體驗。

S3 批次操作支援 [還原](#) 操作，可針對下列儲存層啟動 S3 物件還原：

- 在 S3 Glacier Flexible Retrieval 或 S3 Glacier Deep Archive 儲存類別中封存的物件
- 透過 Archive Access 或 Deep Archive Access 方案中的 S3 Intelligent-Tiering 儲存類別封存的物件

批次操作可以透過程式設計方式在 Amazon S3 主控台上叫用。對於輸入，它需要一個 .csv 資訊清單檔案，其中包含要還原的清單物件。

您可以使用 [Amazon S3 庫存](#) 報告作為批次工作的輸入。庫存報告是針對儲存貯體設定，並且可以限制為特定字首下的物件。這是自動報告，每週或每天都會以 ORC、CSV 或 Parquet 格式產生。

如需設定庫存報告的詳細資訊，請參閱 [Amazon S3 文件](#)。如需有關使用 Boto3 建立 S3 批次操作任務的資訊，請參閱 [Boto3 文件](#)。

最佳實務

我們建議存取封存資料的下列最佳實務：

- 對於大型封存資料集，我們建議在資料之上建立 AWS Glue 資料表，以便使用 Athena 和 Amazon Redshift 等查詢引擎進行讀取。Athena 和 Amazon Redshift 都提供查詢效能的水平擴展。他們也使用模型 pay-per-query，在一次性查詢案例中具有成本效益。此外，Amazon Redshift 的機罩下具有進階查詢加速器（AQUA）引擎，可加速讀取效能，無需額外費用。
- 在 Amazon S3 中定期卸載的封存資料不應儲存為堆積傾印。相反地，它應該儲存為新的分割區。日期分割區會將資料分隔為日期維度（例如 `year=<value>/month=<value>/day=<value>`）。這在兩種情況下非常有用：
 - 如果 Glue AWS 爬蟲程式建立 Glue AWS 資料表，則這些分割區會充當虛擬資料欄。這透過限制掃描到範圍查詢中的分割區的資料來增強讀取效能。
 - 當您僅將物件子集還原為 S3 標準時，這有助於 S3 Glacier 還原操作。
- AWS 當儲存在 Amazon S3 中的封存資料實體分割時，Glue 爬蟲程式會顯示非常值。每次將資料卸載為新的字首分割區時，爬蟲程式只會掃描新的分割區，並更新該分割區的中繼資料。如果資料表的結構描述變更，這些變更會擷取在分割區層級中繼資料中。

歸檔表清理

歸檔程序的最後一個階段是清除歸檔結構描述中的表格。確認您的存檔資料已安全存檔到 Amazon S3 後，即可執行此操作。為了避免對應用程式造成任何影響，建議您在排定的停機時間或維護時段期間，或在應用程式中流量極低的時段期間刪除存檔結構描述表格。這些表不會被您的應用程式主動查詢，並且不應該因為它們對正在進行的交易的影響而引起警報。儘管如此，在停機時間執行 DDL 仍是最佳作法。

釋放大型存檔結構描述表格的儲存空間之後，Amazon Aurora 會使用 [動態調整](#) 協助您節省儲存成本。

資源

文件

- [Amazon Aurora](#)
- [AWS Glue](#)
- [Amazon Simple Storage Service \(Amazon S3\)](#)
- [使用 Amazon Athena 查詢 Amazon S3 庫存 Amazon Athena](#)
- [設定 Amazon S3 庫存](#)
- [在 Amazon S3 物件上執行大規模批次操作](#)
- [Boto3 文件](#)
- [我的SQL分割文件](#)

定價

- [Amazon RDS for MySQL Multi-AZ GP2定價](#)
- [Amazon S3 Glacier 定價](#)

工具

- [Percona Toolkit](#)
- [pt-archiver](#)
- [sysbench](#)

部落格文章

- [S3 Select 和 S3 Glacier Select – 擷取物件的子集](#)
- [使用 pg_partman 和 Amazon S3 封存和清除 Amazon RDS for PostgreSQL 和 Amazon Aurora 的 PostgreSQL 相容資料](#)

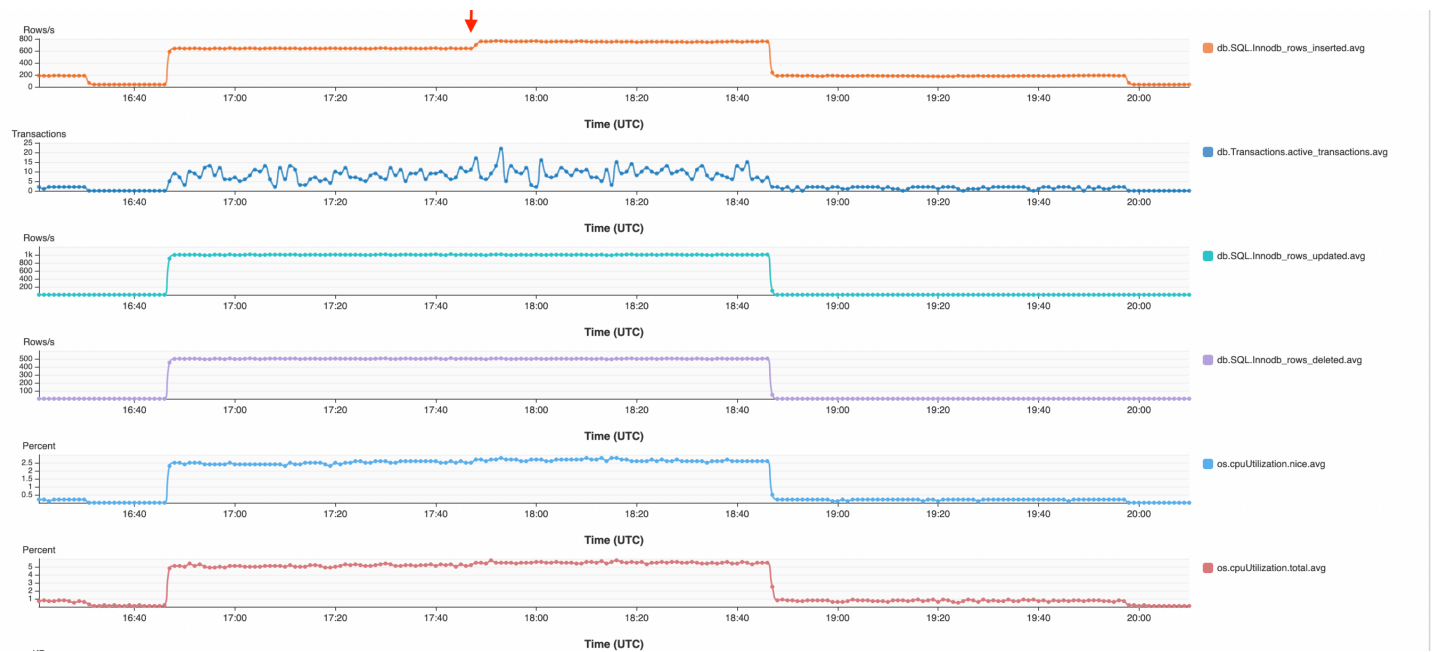
附錄一

所有測試都是在執行的亞馬遜 RDS MySQL 執行個體上執行的 db.r6g.8xlarge 實體類別。

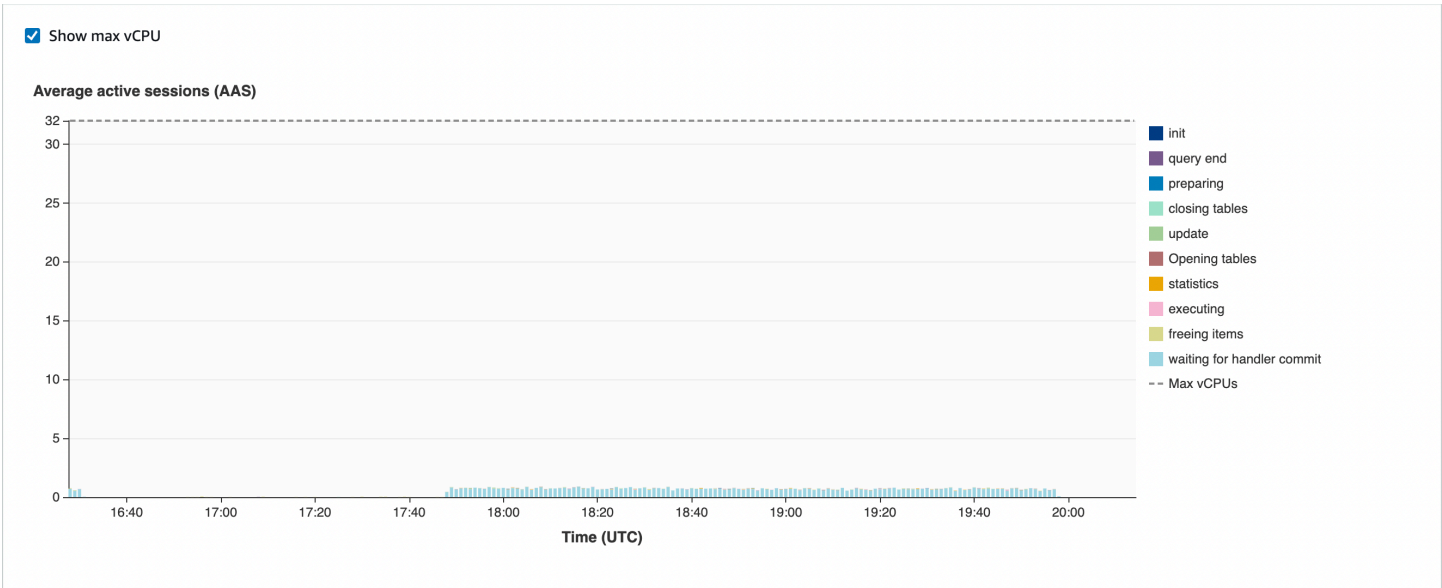
以下 [系統支柱](#) 命令用於準備和運行數據庫上的負載。

```
sysbench oltp_read_write --db-driver=mysql --mysql-db=<DATABASE> --mysql-user=<USER> --mysql-password=<PASSWORD> --mysql-host=<ENDPOINT> --tables=500 --table-size=2000000 --threads=500 prepare
sysbench oltp_read_write --db-driver=mysql --mysql-db=employees --mysql-user=admin --mysql-password=qwertyuiop --mysql-host=mysql8.cbbhujzeoxed.us-east-1.rds.amazonaws.com --tables=500 --rate=500 --time=7200 run
```

在下圖中，OLTP 工作負載正在執行，而 PT 封存程序會在標記箭頭的位置啟動。



pt-Archiver 並行執行時，CPU 使用率沒有顯著變化，因此推斷 PT-Archiver 在執行時不會影響 OLTP 查詢。



附錄二

本節提供 PT-Archiver 工具在不同情況下的基準測試結果。該[系統支柱](#)工具在這個測試中使用，把加載到數據庫上。所有測試都是在運行的亞馬遜 RDS 對 MySQL 實例執行 db.r6g.8xlarge 實體類別。

下面的 sysbench 命令是用來準備和運行在數據庫上的負載：

```
sysbench oltp_read_write --db-driver=mysql --mysql-db=<DATABASE> --mysql-user=<USER> --mysql-password=<PASSWORD> --mysql-host=<ENDPOINT> --tables=1000 --table-size=2000000 --threads=500 prepare
sysbench oltp_read_write --db-driver=mysql --mysql-db=<DATABASE> --mysql-user=<USER> --mysql-password=<PASSWORD> --mysql-host=<ENDPOINT> --tables=1000 --rate=500 --threads=500 run
```

歸檔沒有主鍵和只有一個索引 (數據庫沒有負載) 的表

```
Started at 2022-11-07T05:29:12, ended at 2022-11-07T06:03:31
Action      Count          Time           Pct
commit     600050         1715.3582     83.31
select     300025         166.5470      8.09
inserting  300024         165.4025      8.03
other      0              11.6644       0.57
```

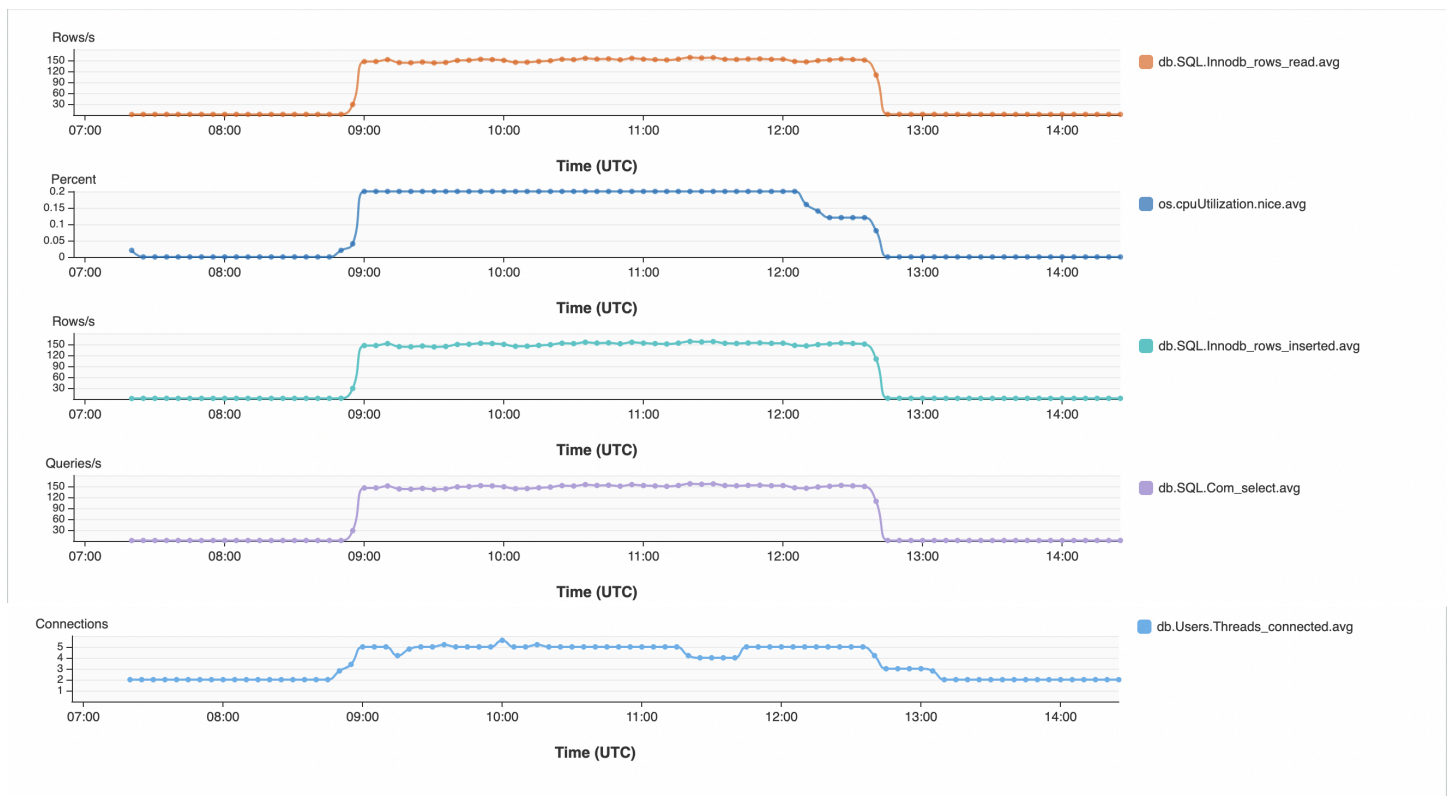
大約花了 34 分鐘的時間來存檔 300,024 行。此資料表有 200 萬個資料列，但此工具只會封存具有索引資料行唯一資料的資料列。

歸檔具有主鍵的表 (數據庫沒有負載)

```
Started at 2022-11-16T08:53:49, ended at 2022-11-16T12:38:18
Action      Count          Time           Pct
commit     4000000        11065.9534    82.16
select     2000000        1278.1854     9.49
inserting  1999999        1050.4961     7.80
other      0              74.1519       0.55
```

存檔 1,999,999 行大約需要 3 個小時，44 分鐘和 29 秒。

下圖顯示了 pt-archiver 在沒有任何負載的情況下自行運行時消耗很少的 CPU 和資源。



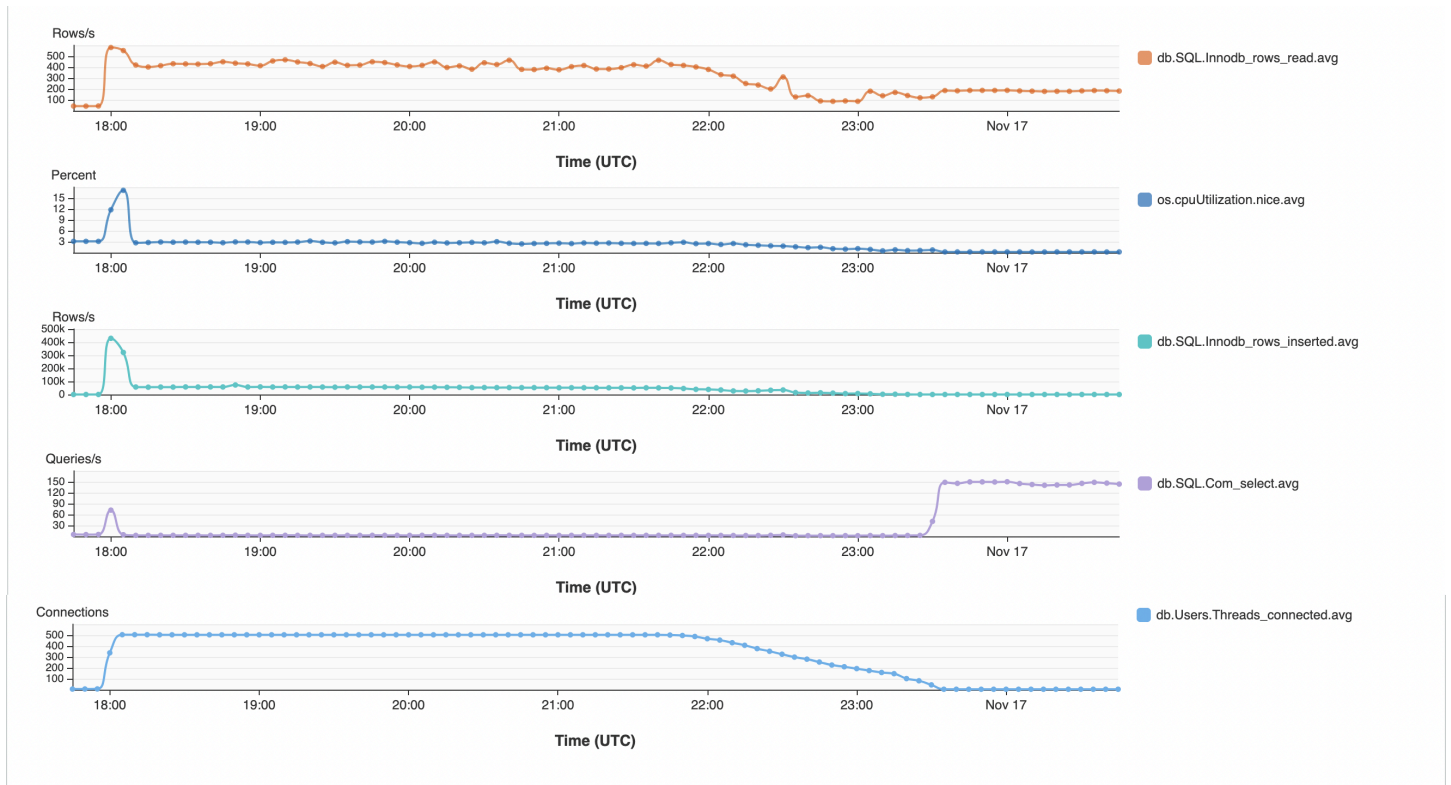
歸檔具有主鍵的表 (在數據庫上有負載)

Started at 2022-11-16T17:37:07, ended at 2022-11-17T03:20:43

Action	Count	Time	Pct
commit	4000000	19688.8362	56.23
inserting	1999999	13933.4418	39.79
select	2000000	1305.1770	3.73
other	0	89.1787	0.25

存檔 1999999 行需要大約 9 小時 43 分鐘和 36 秒。

下圖顯示在測試期間，由於 sysbench 套用的負載，CPU 使用率高達 15%。完成載入之後，pt-archiver 會繼續如預期般耗用最少的 CPU，以完成封存。



從圖表中可以看出，pt-archiver 不會在數據庫上有負載時積極存檔。

文件歷史記錄

下表描述了本指南的重大變更。如果您想要收到未來更新的通知，您可以訂閱[RSS摘要](#)。

變更	描述	日期
更新	已移除部分封存選項的相關資訊。	2024 年 10 月 21 日
初次出版	—	2023 年 6 月 8 日

AWS 規範指南詞彙表

以下是 AWS Prescriptive Guidance 所提供策略、指南和模式的常用術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

數字

7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的內部部署 Oracle 資料庫遷移至 Amazon Aurora PostgreSQL-Compatible Edition。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將您的內部部署 Oracle 資料庫遷移至中的 Oracle 的 Amazon Relational Database Service (Amazon RDS) AWS 雲端。
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 (CRM) 系統遷移至 Salesforce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將內部部署 Oracle 資料庫遷移至中 EC2 執行個體上的 Oracle AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移至相同平台的雲端服務。範例：遷移 Microsoft Hyper-V 應用程式至 AWS。
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

A

ABAC

請參閱 [屬性型存取控制](#)。

抽象服務

請參閱 [受管服務](#)。

ACID

請參閱 [原子、一致性、隔離、持久性](#)。

主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但比 [主動被動遷移](#) 需要更多工作。

主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。目標資料庫在遷移期間不接受任何交易。

彙總函數

在一組資料列上操作並計算該群組單一傳回值的 SQL 函數。彙總函數的範例包括 SUM 和 MAX。

AI

請參閱 [人工智慧](#)。

AIOps

請參閱 [人工智慧操作](#)。

匿名化

在資料集中永久刪除個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

反模式

經常用於重複性問題的解決方案，其解決方案具有反效益、無效或效果不如替代方案。

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體侵害。

應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是 [產品組合探索和分析程序](#) 的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱[什麼是人工智慧？](#)

人工智慧操作 (AI Ops)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需如何在遷移策略 AI Ops 中使用 AWS 的詳細資訊，請參閱[操作整合指南](#)。

非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

原子度、一致性、隔離、耐久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱 AWS Identity and Access Management (IAM) 文件[ABAC AWS](#)中的。

權威性資料來源

存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將資料從授權資料來源複製到其他位置，以處理或修改資料，例如匿名、編輯或假名化資料。

可用區域

中與其他可用區域中的故障 AWS 區域 隔離的不同位置，並對相同區域中的其他可用區域提供低成本、低延遲的網路連線。

AWS 雲端採用架構 (AWS CAF)

來自的指導方針和最佳實務架構 AWS ，可協助組織制定高效且有效的計劃，以成功地遷移至雲端。AWS CAF 將指導方針整理成六個重點領域：業務、人員、治理、平台、安全和操作。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。為此，AWS CAF 提供人員開發、訓練和通訊的指引，協助組織準備好成功採用雲端。如需詳細資訊，請參閱[AWS CAF 網站](#)和[AWS CAF 白皮書](#)。

AWS 工作負載資格架構 (AWS WQF)

評估資料庫遷移工作負載、建議遷移策略並提供工作估算的工具。AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

B

錯誤的機器人

旨在中斷或傷害個人或組織的[機器人](#)。

BCP

請參閱[業務連續性規劃](#)。

行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以搭配 Amazon Detective 使用行為圖表來檢查失敗的登入嘗試、可疑API的呼叫和類似的動作。如需詳細資訊，請參閱偵測文件中的[行為圖中的資料](#)。

大端序系統

首先儲存最高有效位元組的系統。另請參閱[永久性](#)。

二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 (藍色) 中執行目前的應用程式版本，並在另一個環境 (綠色) 中執行新的應用程式版本。此策略可協助您在影響最小的情況下快速復原。

機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。某些機器人很有用或有益，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。某些其他稱為壞機器人的機器人，旨在中斷或傷害個人或組織。

殭屍網路

受到**惡意軟體**感染且由單一方控制的**機器人**網路，稱為機器人繼承器或機器人運算子。殭屍網路是擴展機器人及其影響的最佳已知機制。

分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱[關於分支](#)（GitHub 文件）。

碎片存取

在特殊情況下，以及透過核准的程序，使用者取得其通常無權存取 AWS 帳戶之存取權的快速方法。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS [實作碎片程序](#) 指標。

棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和[綠地](#)策略。

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

業務如何創造價值（例如，銷售、客戶服務或營銷）。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 [AWS 上執行容器化微服務](#) 白皮書的 [圍繞業務能力進行組織](#) 部分。

業務連續性規劃（BCP）

一種解決破壞性事件（如大規模遷移）對營運的潛在影響並使業務能夠快速恢復營運的計畫。

C

CAF

請參閱[AWS 雲端採用架構](#)。

Canary 部署

版本向最終使用者緩慢且增量的版本。當您有信心時，請部署新版本並完全取代目前版本。

CCoE

請參閱[Cloud Center of Excellence](#)。

CDC

請參閱[變更資料擷取](#)。

變更資料擷取 (CDC)

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以使用 CDC 進行各種用途，例如稽核或複寫目標系統中的變更，以維持同步。

混亂工程

故意引入故障或破壞性事件，以測試系統的復原能力。您可以使用[AWS Fault Injection Service \(AWS FIS \)](#) 來執行實驗，以強調 AWS 工作負載並評估其回應。

CI/CD

請參閱[持續整合和持續交付](#)。

分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

用戶端加密

在目標 AWS 服務接收資料之前，在本機加密資料。

Cloud Center of Excellence (CCoE)

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端企業策略部落格上的[CCoE文章](#)。

雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲端運算通常連接到[邊緣運算](#)技術。

雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱[建置您的雲端操作模型](#)。

採用雲端階段

組織在遷移至 時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展您的雲端採用（例如，建立登陸區域、定義 CCoE、建立操作模型）
- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段由 Stephen Orban 在企業 AWS 雲端 策略部落格的 The [Journey Toward Cloud-First](#) 和 [採用階段](#) 部落格中定義。如需有關它們如何與 AWS 遷移策略關聯的資訊，請參閱[遷移準備指南](#)。

CMDB

請參閱[組態管理資料庫](#)。

程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲端儲存庫包括 GitHub 或 Bitbucket Cloud。每個版本的程式碼都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

冷資料

很少存取且通常為歷史資料的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且價格較低的儲存層或類別，可以降低成本。

電腦視覺 (CV)

使用機器學習從數位映像和影片等視覺化格式分析和擷取資訊的 [AI](#) 欄位。例如，AWS Panorama 提供將 CV 新增至內部部署攝影機網路的裝置，而 Amazon 則 SageMaker 提供 CV 的影像處理演算法。

組態偏離

對於工作負載，組態會從預期狀態變更。這可能會導致工作負載變得不合規，而且通常是漸進和無意的。

組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常會 CMDB 在遷移的產品組合探索和分析階段使用來自的資料。

一致性套件

您可以組合的 AWS Config 規則和修復動作集合，以自訂合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶和區域中或整個組織中的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的[一致性套件](#)。

持續整合和持續交付 (CI/CD)

自動化軟體發行程序的來源、建置、測試、暫存和生產階段的程序。CI/CD is commonly described as a pipeline. CI/CD 可協助您自動化程序、提高生產力、改善程式碼品質，以及更快交付。如需詳細資訊，請參閱[持續交付的優點](#)。CD 也可表示持續部署。如需詳細資訊，請參閱[持續交付與持續部署](#)。

CV

請參閱[電腦視覺效果](#)。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected Framework 中安全支柱的元件。如需詳細資訊，請參閱[資料分類](#)。

資料偏離

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變化。資料偏離可以降低 ML 模型預測的整體品質、準確性和公平性。

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

架構架構架構，提供分散式、分散式資料擁有權，並具有集中式管理和治理。

資料最小化

僅收集和處理嚴格必要資料的原則。在 [中實作資料最小化 AWS 雲端](#) 可以降低隱私權風險、成本和分析碳足跡。

資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱 [在上建立資料周邊 AWS](#)。

資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

資料來源

在整個資料生命週期中追蹤資料的原始伺服器 and 歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

資料主體

正在收集和處理資料的個人。

資料倉儲

支援商業智慧的資料管理系統，例如分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

資料庫操作語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱 [資料庫定義語言](#)。

深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

defense-in-depth

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在上採用此策略時 AWS，您可以在 AWS Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，方法 defense-in-depth 可能會結合多重要素驗證、網路分割和加密。

委派的管理員

在中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶，以管理組織的帳戶並管理該服務的許可。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的[可搭配 AWS Organizations 運作的服務](#)。

部署

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

開發環境

請參閱[環境](#)。

偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[偵測性控制](#)。

開發值串流映射 (DVSM)

用於識別和排序限制的程式，這些限制會對軟體開發生命週期中的速度和品質產生不利影響。DVSM 延伸了最初專為精實生產實務設計的價值串流映射程序。它專注於透過軟體開發程序建立和移動價值所需的步驟和團隊。

數位分身

真實世界系統的虛擬表示法，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

維度資料表

在[星狀結構描述](#)中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常是文字欄位或離散數字，其行為與文字類似。這些屬性通常用於查詢限制、篩選和結果集標籤。

災難

阻止工作負載或系統在其主要部署位置實現其業務目標的事件。這些事件可能是自然災難、技術故障或人為動作的結果，例如意外錯誤組態或惡意軟體攻擊。

災難復原 (DR)

您用來將[災難造成的停機時間和資料遺失降至最低的策略和程序](#)。如需詳細資訊，請參閱 AWS Well-Architected Framework [中的雲端中的工作負載災難復原 AWS：復原](#)。

DML

請參閱[資料庫操作語言](#)。

領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何搭配 strangler fig 模式使用網域驅動設計的資訊，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX \) Web 服務](#)。

DR

請參閱[災難復原](#)。

漂移偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源 中的漂移，或者您可以使用 AWS Control Tower 來[偵測您的登陸區域中可能會影響對治理要求合規性的變更](#)。
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html>

DVSM

請參閱[開發值串流映射](#)。

E

EDA

請參閱[探索性資料分析](#)。

邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與[雲端運算](#) 相比，邊緣運算可以減少通訊延遲並縮短回應時間。

加密

將純文字資料轉換為可人工讀取的運算程序。

加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

端點

請參閱[服務端點](#)。

端點服務

您可以在虛擬私有雲端（VPC）中託管以與其他使用者共用的服務。您可以使用 建立端點服務，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management（IAM）主體。這些帳戶或主體可以透過建立介面端點，私下連線至您的VPC端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud（AmazonVPC）文件中的[建立端點服務](#)。

企業資源規劃（ERP）

可自動化和**管理企業關鍵業務流程**（例如會計[MES](#)、和 專案管理）的系統。

信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service（AWS KMS）文件中的[信封加密](#)。

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如，AWS CAF 安全特徵包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱[計畫實作指南](#)。

ERP

請參閱[企業資源規劃](#)。

探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。EDA 透過計算摘要統計資料和建立資料視覺化來執行。

F

事實資料表

[星狀結構描述](#) 中的中央資料表。它存放有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含量值的資料，以及包含維度資料表外部索引鍵的資料。

快速失敗

使用頻繁且增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在中 AWS 雲端，邊界，例如可用區域 AWS 區域、控制平面或資料平面，這些邊界會限制故障的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱[AWS 故障隔離界限](#)。

功能分支

請參閱[分支](#)。

特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

功能重要性

特徵對於模型的預測有多重要。這通常表示為數值分數，可透過各種技術計算，例如 Shapley 累加解釋（SHAP）和整合式漸層。如需詳細資訊，請參閱[使用的機器學習模型可解釋性：AWS](#)。

特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

FGAC

請參閱[精細存取控制](#)。

精細存取控制（FGAC）

使用多個條件來允許或拒絕存取請求。

閃切遷移

一種資料庫遷移方法，透過[變更資料擷取](#)使用連續資料複寫，以盡可能在最短時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

G

地理封鎖

請參閱[地理限制](#)。

地理限制 (地理封鎖)

在 Amazon 中 CloudFront，此選項可防止特定國家/地區的使用者存取內容分發。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的[限制內容的地理分佈](#)。

Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被視為舊版，而以[中繼線為基礎的工作流程](#)是現代的首選方法。

綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為[棕地](#)。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

防護機制

高階規則，可協助管理跨組織單位 () 的資源、政策和合規性OUs。預防性防護機制會強制執行政策，以確保符合合規標準。它們是透過使用服務控制政策和IAM許可界限來實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。其實作方式是使用 AWS Config、AWS Security Hub、Amazon GuardDuty、AWS Trusted Advisor、Amazon Inspector 和自訂 AWS Lambda 檢查。

H

HA

請參閱[高可用性](#)。

異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如，Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分，而轉換結構描述可能是一項複雜任務。[AWS 提供有助於結構描述轉換的 AWS SCT](#)。

高可用性 (HA)

工作負載在遇到挑戰或災難時持續運作的能力，無需介入。HA 系統設計為自動容錯移轉、持續提供高品質效能，以及處理不同的負載和故障，並將效能影響降至最低。

歷史現代化

一種用於現代化和升級操作技術 (OT) 系統的方法，以更好地滿足製造業的需求。歷史記錄器是一種資料庫類型，用於從工廠的不同來源收集和存放資料。

異質資料庫遷移

將來源資料庫遷移至共用相同資料庫引擎的目標資料庫 (例如 Microsoft SQL Server 遷移至 Amazon RDS for SQL Server)。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

常用資料

經常存取的資料，例如即時資料或最近的轉譯資料。此資料通常需要高效能儲存層或類別，才能提供快速的查詢回應。

修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性，修正程式通常在典型 DevOps 的發行工作流程之外建立。

超級護理期間

在切換後，遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常，此期間的長度為 1-4 天。在超級護理期間結束時，遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

I

laC

將[基礎設施視為程式碼](#)。

身分型政策

連接至一或多個IAM主體的政策，其定義其在 AWS 雲端環境中的許可。

閒置應用程式

在 90 天內，平均CPU和記憶體用量介於 5% 到 20% 的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

IIoT

請參閱[工業物聯網](#)。

不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有基礎設施。與可變基礎設施相比，不可避免的[基礎設施](#)本質上更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[使用不可變基礎設施部署](#)最佳實務。

傳入（傳入）VPC

在 AWS 多帳戶架構中，VPC接受、檢查和路由來自應用程式外部的網路連線。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

工業 4.0

由 [Klaus Schwab](#) 於 2016 年推出的術語，指透過連線能力、即時資料、自動化、分析和 AI/ML 的進步來現代化製造程序。

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱 [建置工業物聯網 \(IIoT\) 數位轉型策略](#)。

檢查 VPC

在 AWS 多帳戶架構中，集中 VPC 管理 VPCs (在相同或不同的 中 AWS 區域)、網際網路和內部部署網路之間的網路流量檢查。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱 [什麼是 IoT?](#)

可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱 [使用的機器學習模型可解釋性 AWS](#)。

IoT

請參閱 [物聯網](#)。

IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 提供的基礎 ITSM。

IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需整合雲端操作與 ITSM 工具的相關資訊，請參閱 [操作整合指南](#)。

ITIL

請參閱 [IT 資訊庫](#)。

ITSM

請參閱 [IT 服務管理](#)。

L

標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會被明確指派安全標籤值。使用者安全標籤與資料安全標籤之間的交集決定使用者可以看到哪些資料列和資料欄。

登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱 [設定安全且可擴展的多帳戶 AWS 環境](#)。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱 [標籤型存取控制](#)。

最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的 [套用最低權限許可](#)。

隨即轉移

請參閱 [7 Rs](#)。

小端序系統

首先儲存最低有效位元組的系統。另請參閱[永久性](#)。

較低的環境

請參閱[環境](#)。

M

機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱[機器學習](#)。

主要分支

請參閱[分支](#)。

惡意軟體

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬程式、間諜軟體和鍵盤側錄程式。

受管服務

AWS 服務可 AWS 操作基礎設施層、作業系統和平台，而且您可以存取端點來存放和擷取資料。Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

製造執行系統 (MES)

用於追蹤、監控、記錄和控制生產程序的軟體系統，可將原物料轉換為工廠的成品。

MAP

請參閱[遷移加速程式](#)。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在運作時強化和改善本身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[建置機制](#)。

成員帳戶

除了屬於中組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

MES

請參閱[製造執行系統](#)。

訊息佇列遙測傳輸 (MQTT)

根據[發佈/訂閱](#)模式的輕量型 machine-to-machine (M2M) 通訊協定，適用於資源受限的 [IoT](#) 裝置。

微服務

小型、獨立的服務，透過定義明確的方式進行通訊，APIs通常由小型、獨立的團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱[使用無 AWS 伺服器服務整合微服務](#)。

微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務會使用輕量型，透過定義明確的介面進行通訊APIs。此架構中的每個微服務都可以進行更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱[在上實作微服務 AWS](#)。

Migration Acceleration Program (MAP)

提供諮詢支援、訓練和服務，以協助組織建立強大的操作基礎以遷移至雲端，並協助抵銷遷移初始成本的 AWS 計畫。MAP 包含以系統化方式執行舊版遷移的遷移方法，以及一組可自動化和加速常見遷移案例的工具。

大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是 [AWS 遷移策略](#) 的第三階段。

遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。遷移工廠團隊通常包括操作、業務分析師和擁有者、遷移工程師、開發人員和在衝刺中工作的 DevOps 專業人員。20% 至 50% 之間的

企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的[遷移工廠的討論](#)和[雲端遷移工廠指南](#)。

遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。遷移中繼資料的範例包括目標子網路、安全群組和 AWS 帳戶。

遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：EC2使用 AWS Application Migration Service 重新託管遷移至 Amazon。

遷移產品組合評估 (MPA)

線上工具，提供驗證商業案例以遷移至的資訊 AWS 雲端。MPA 提供詳細的產品組合評估（伺服器大小調整、定價、TCO比較、遷移成本分析）以及遷移規劃（應用程式資料分析和資料收集、應用程式分組、遷移優先順序和波規劃）。此[MPA工具](#)（需要登入）可供所有 AWS 顧問和APN 合作夥伴顧問免費使用。

遷移就緒狀態評估 (MRA)

使用取得組織雲端整備狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序 AWS CAF。如需詳細資訊，請參閱[遷移準備程度指南](#)。MRA 是[AWS 遷移策略](#)的第一個階段。

遷移策略

用於將工作負載遷移至的方法 AWS 雲端。如需詳細資訊，請參閱本詞彙表中的 [7 個 Rs](#) 項目，並請參閱將[組織動員以加速大規模遷移](#)。

機器學習 (ML)

請參閱[機器學習](#)。

現代化

將過時的（舊版或單一）應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱[中的應用程式現代化策略 AWS 雲端](#)。

現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱[中的評估應用程式的現代化準備 AWS 雲端](#)程度。

單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱[將單一體系分解為微服務](#)。

MPA

請參閱[遷移產品組合評估](#)。

MQTT

請參閱[訊息佇列遙測傳輸](#)。

多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」或者「這個客戶對哪種產品類別最感興趣？」

可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性，AWS Well-Architected Framework 建議使用[不可變的基礎設施](#)作為最佳實務。

O

OAC

請參閱[原始存取控制](#)。

OAI

請參閱[原始存取身分](#)。

OCM

請參閱[組織變更管理](#)。

離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

OI

請參閱[操作整合](#)。

OLA

請參閱[操作層級協議](#)。

線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

OPC-UA

請參閱[開啟程序通訊 - Unified Architecture](#)。

開放程序通訊 - Unified Architecture (OPC-UA)

工業自動化的 machine-to-machine (M2M) 通訊協定。OPC-UA 提供與資料加密、身分驗證和授權方案的互通性標準。

操作層級協議 (OLA)

闡明哪些功能性 IT 群組承諾相互交付的協議，以支援服務層級協議 (SLA)。

操作預備檢閱 (ORR)

問題及相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[操作就緒檢閱 \(ORR \)](#)。

操作技術 (OT)

與實體環境搭配使用的硬體和軟體系統，以控制工業操作、設備和基礎設施。在製造中，整合 OT 和資訊技術 (IT) 系統是 [Industry 4.0](#) 轉型的關鍵重點。

操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱[操作整合指南](#)。

組織追蹤

由建立的追蹤 AWS CloudTrail 會記錄 AWS 帳戶中組織中所有的事件 AWS Organizations。在屬於組織的每個 AWS 帳戶中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱文件中的 CloudTrail[為組織建立追蹤](#)。

組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變革採用、解決轉型問題，以及推動文化和組織變革，協助組織準備和轉換新系統和策略。在 AWS 遷移策略

中，由於雲端採用專案所需的變更速度，此架構稱為人員加速。如需詳細資訊，請參閱 [OCM指南](#)。

原始存取控制 (OAC)

在中 CloudFront，用於限制存取以保護您的 Amazon Simple Storage Service (Amazon S3) 內容的增強型選項。OAC 支援所有中的所有 S3 儲存貯體 AWS 區域，伺服器端加密使用 AWS KMS (SSE-KMS)，以及對 S3 儲存貯體的動態PUT和DELETE請求。

原始存取身分 (OAI)

在中 CloudFront，此選項用於限制存取，以保護您的 Amazon S3 內容。當您使用時OAI，會 CloudFront 建立 Amazon S3 可以驗證的主體。已驗證的主體只能透過特定 CloudFront 分發存取 S3 儲存貯體中的內容。另請參閱 [OAC](#)，它提供更精細和增強的存取控制。

ORR

請參閱[操作預備檢閱](#)。

OT

請參閱[操作技術](#)。

傳出 (輸出) VPC

在 AWS 多帳戶架構中，VPC處理從應用程式內啟動之網路連線的。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

P

許可界限

連接至IAM主體的IAM管理政策，以設定使用者或角色可擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的[許可界限](#)。

個人身分資訊 (PII)

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。的範例PII包括名稱、地址和聯絡資訊。

PII

請參閱[個人識別資訊](#)。

手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

PLC

請參閱[可程式邏輯控制器](#)。

PLM

請參閱[產品生命週期管理](#)。

政策

可以定義許可（請參閱[身分型政策](#)）、指定存取條件（請參閱[資源型政策](#)）或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱[服務控制政策](#)）的物件。

混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱[在微服務中啟用資料持久性](#)。

組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱[評估遷移準備程度](#)。

述詞

傳回 true 或的查詢條件 false，通常位於 WHERE 子句中。

述詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這可減少必須從關聯式資料庫擷取和處理的資料量，並改善查詢效能。

預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[預防性控制](#)。

委託人

中可執行動作和存取資源 AWS 的實體。此實體通常是 AWS 帳戶、IAM 角色或使用者的根使用者。如需詳細資訊，請參閱 IAM 文件中[角色術語和概念](#)中的主體。

設計隱私

一種系統工程方法，在整個工程過程中將隱私權納入考量。

私有託管區域

容器，其中包含您希望 Amazon Route 53 如何回應一個或多個 內網域及其子網域的DNS查詢的資訊VPCs。如需詳細資訊，請參閱 Route 53 文件中的[使用私有託管區域](#)。

主動控制

旨在防止部署不合規資源的[安全控制](#)。這些控制項會先掃描資源，然後再佈建。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的[控制項參考指南](#)，並請參閱在上實作安全[控制項中的主動](#)控制項。 AWS

產品生命週期管理 (PLM)

從設計、開發和啟動，到成長和成熟，再到拒絕和移除，產品整個生命週期的資料和程序管理。

生產環境

請參閱[環境](#)。

可程式設計邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

擬匿名化

將資料集中的個人識別碼取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

publish/subscribe (pub/sub)

一種模式，可在微服務之間啟用非同步通訊，以提高可擴展性和回應能力。例如，在微服務型中[MES](#)，微服務可以將事件訊息發佈到其他微服務可以訂閱的頻道。系統可以新增新的微服務，而無需變更發佈服務。

Q

查詢計劃

一系列步驟，如指示，用於存取SQL關聯式資料庫系統中的資料。

查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

R

RACI 矩陣

請參閱[負責、負責、已諮詢、知情 \(RACI \)](#)。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

RASCI 矩陣

請參閱[負責、負責、已諮詢、知情 \(RACI \)](#)。

RCAC

請參閱[資料列和資料欄存取控制](#)。

僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

重新架構師

請參閱 [7 Rs](#)。

復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

復原時間目標 (RTO)

服務中斷和服務還原之間的可接受延遲上限。

重構

請參閱 [7 Rs](#)。

區域

地理區域 AWS 的資源集合。每個 AWS 區域 都獨立於其他，以提供容錯能力、穩定性和彈性。如需詳細資訊，請參閱[指定 AWS 區域 哪些帳戶可以使用](#)。

迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

重新託管

請參閱 [7 Rs](#)。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新定位

請參閱 [7 Rs](#)。

轉譯形式

請參閱 [7 Rs](#)。

回購

請參閱 [7 Rs](#)。

彈性

應用程式抵抗中斷或從中斷中復原的能力。[在中規劃復原能力時，高可用性和災難復原](#)是常見的考量因素 AWS 雲端。如需詳細資訊，請參閱[AWS 雲端 復原能力](#)。

資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

負責任、負責、已諮詢、知情 (RACI) 矩陣

定義所有涉及遷移活動和雲端操作之各方的角色和責任的矩陣。矩陣名稱衍生自矩陣中定義的責任類型：負責人 (R)、責任 (A)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除該矩陣，則稱為 RACI 矩陣。

回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[回應性控制](#)。

保留

請參閱 [7 Rs](#)。

淘汰

請參閱 [7 Rs](#)。

輪換

定期更新[秘密](#)的程序，讓攻擊者更難存取憑證。

資料列和資料欄存取控制（RCAC）

使用已定義存取規則的基本靈活SQL表達式。RCAC 包含資料列許可和資料欄遮罩。

RPO

請參閱[復原點目標](#)。

RTO

請參閱[復原時間目標](#)。

執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

S

SAML 2.0

許多身分提供者（IdPs）使用的開放標準。此功能會啟用聯合單一登入（SSO），因此使用者可以登入 AWS Management Console 或呼叫操作，AWS API而不必IAM為您組織中的每個人建立使用者。如需 SAML 2.0 型聯合的詳細資訊，請參閱 IAM 文件中的[關於 SAML 2.0 型聯合](#)。

SCADA

請參閱 [監控控制和資料擷取](#)。

SCP

請參閱[服務控制政策](#)。

秘密

在 AWS Secrets Manager 中，以加密形式存放的機密或限制資訊，例如密碼或使用者憑證。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 [Secrets Manager 文件中的 Secrets Manager 秘密中的內容？](#)。

安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全控制有四種主要類型：[預防性](#)、[偵測性](#)、[回應性](#) 和 [主動](#)。

安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

安全資訊和事件管理（SIEM）系統

結合安全資訊管理（SIM）和安全事件管理（SEM）系統的工具和服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生警示。

安全回應自動化

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為[偵測](#)或[回應](#)式安全控制，協助您實作 AWS 安全最佳實務。自動化回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換憑證。

伺服器端加密

由接收資料的 AWS 服務 加密其目的地的資料。

服務控制政策（SCP）

為 AWS Organizations 中的組織的所有帳戶提供集中控制許可的政策。SCPs 定義管理員可委派給使用者或角色之動作的防護機制或設定限制。您可以使用 SCPs 做為允許清單或拒絕清單，以指定允許或禁止的服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中[的服務控制政策](#)。

服務端點

URL 的進入點 AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 [AWS 服務 端點](#)。

服務層級協議（SLA）

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

服務層級指標（SLI）

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標（SLO）

代表服務運作狀態的目標指標，由[服務層級指標](#) 測量。

共同責任模式

描述您與共同 AWS 承擔雲端安全與合規責任的模型。AWS 負責雲端的安全，而負責雲端的安全。如需詳細資訊，請參閱[共同責任模式](#)。

SIEM

請參閱[安全資訊和事件管理系統](#)。

單一失敗點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱[服務層級協議](#)。

SLI

請參閱[服務層級指示器](#)。

SLO

請參閱[服務層級目標](#)。

split-and-seed 模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱[中的階段式應用程式現代化方法 AWS 雲端](#)。

SPOF

請參閱[單一失敗點](#)。

星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構專為[資料倉儲](#)或商業智慧用途而設計。

Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由 [Martin Fowler 引入](#)，作為重寫單一系統時管理風險的方式。如需如何套用此模式的範例，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX \) Web 服務](#)。

子網

中 IP 地址的範圍 VPC。子網必須位於單一可用區域。

監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 [Amazon CloudWatch Synthetics](#) 來建立這些測試。

T

標籤

作為中繼資料的鍵值對，用於組織您的 AWS 資源。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱 [標記您的 AWS 資源](#)。

目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

測試環境

請參閱 [環境](#)。

訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

傳輸閘道

可用來互連 VPCs 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 [AWS Transit Gateway](#) 文件中的 [什麼是傳輸閘道](#)。

主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

受信任的存取權

將許可授予您指定的服務，以代表您在組織中執行任務 AWS Organizations，並在其帳戶中執行任務。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱 文件中的 AWS Organizations [AWS Organizations 搭配使用其他 AWS 服務](#)。

調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

雙披薩團隊

一個小型 DevOps 團隊，您可以使用兩個披薩來饋送。雙披薩團隊規模可確保軟體開發中的最佳協作。

U

不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱 [量化深度學習系統的不確定性指南](#)。

未區分的任務

也稱為繁重型，是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、維護和容量規劃。

較高的環境

請參閱 [環境](#)。

V

清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

VPC 對等

兩個之間的連線VPCs，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 [Amazon 文件中的VPC互連內容](#)。VPC

漏洞

損害系統安全性的軟體或硬體缺陷。

W

暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等慢的查詢。

視窗函數

對以某種方式與目前記錄相關聯之資料列群組執行計算的SQL函數。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器 and 應用程式。

WORM

請參閱[寫入一次，讀取許多](#)。

WQF

請參閱[AWS工作負載資格架構](#)。

寫入一次，讀取許多 (WORM)

一次性寫入資料的儲存模型，可防止刪除或修改資料。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為[不可變的](#)。

Z

零時差漏洞

利用[零時差漏洞](#)的攻擊，通常是惡意軟體。

零時差漏洞

生產系統中未緩解的缺陷或漏洞。威脅發動者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

殭屍應用程式

平均CPU和記憶體用量低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

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。