規劃成功的 MLOps

AWS 方案指引

AWS 方案指引: 規劃成功的 MLOps

Table of Contents

規劃成功的 MLOps

Bruno Klein，Amazon Web Services (AWS)

2021 年 12 月 (文件歷史記錄)

在生產環境中部署機器學習 (ML) 解決方案會帶來許多標準軟體開發專案中未出現的挑戰。ML 解決方案更複雜、更棘手，一開始就正確。它們也存在於通常不穩定的環境中，其中資料分佈會隨著時間的推移而明顯偏離各種預期和非預期的原因。

許多 ML 從業人員並非來自軟體工程背景，因此這些問題會進一步加重，因此他們可能不熟悉這個產業的最佳實務，例如撰寫可測試程式碼、模組化元件，以及有效使用版本控制。這些挑戰會為 ML 團隊帶來技術債務，而且解決方案會隨著時間的推移而變得更加複雜且難以維持。

本指南列舉 ML 操作 (MLOps) 最佳實務，以協助減輕 ML 專案和工作負載中的這些挑戰。

由於 MLOps 是一個交叉切削問題，這些問題不僅會影響部署和監控程序，還影響整個模型生命週期。在本指南中，MLOps 最佳實務分為四個主要領域：

- 資料
- 訓練
- 部署
- 監控

目標業務成果

在生產環境中部署 ML 模型是一項任務，需要持續努力和專屬的團隊，才能在其生命週期內（在某些情況下甚至數年）維護這些資源。ML 模型可以從商業資料中釋放大量價值，但成本很高。為了將成本降至最低，企業應遵循軟體開發和資料科學的良好實務。他們應該注意 ML 系統的細微差別，例如資料偏離，這可讓模型在一段時間後意外執行。透過了解這些疑慮，企業可以在短期和長期中安全且靈活地達成業務目標。

ML 模型有幾種類型，而其目標產業有不同類型的 ML 任務和業務問題，因此您需要考慮每個模型和產業的不同問題集。本指南中列出的實務並非特定於模型或業務，但適用於廣泛的模型和產業，以改善部署時間、產生更高的生產力，並建置更強大的治理和安全性。

將模型放入生產是一個多學科任務，需要資料科學家、機器學習工程師、資料工程師和軟體工程師。當您建置您的 ML 團隊時，我們建議您以這些技能和背景為目標。

資料

DevOps 是一種軟體工程實務，可處理軟體的操作。DevOps 的常見元素是版本控制的程式碼、持續整合和持續交付 (CI/CD) 管道、單元測試，以及可重現的程式碼建置和部署，全都涉及程式碼。ML 模型是程式碼和資料的產品，因此資料必須符合與程式碼相同的標準。MLOps 必須解決與資料相關的問題，例如如何維護資料品質、如何識別資料中的邊緣案例、如何保護資料，以及如何讓資料更易於維護。

主題

標記

提供清楚的標籤指示

資料集可能包含模棱兩可的樣本，導致整個資料集的標籤不一致。例如，請考慮標記包含狗的影像的任務。有些範例可能只包含動物的概觀。應該以正面或負面標籤標記這些標籤嗎？ 提供清楚且客觀的指示給標籤人員，即可解決這類問題。

使用多數投票

現在，請考慮標記speech-to-text資料集的問題，該資料集包含音素上類似或與其他字詞相同的音訊，例如 know and go、shoe and two、cry and high 或 right and write。在這種情況下，標籤工具可能會不一致地標記這些樣本。

為了維持標記的高度正確性，常見的方法是使用多數投票，其中會將相同的資料範例提供給多個工作者，並彙總其結果。此方法及其更複雜的變化在部落格文章中描述 使用 Amazon SageMaker AI Ground Truth 的群眾智慧，在機器學習部落格上更準確地註釋資料。 AWS Machine Learning

分割和資料外洩

當您的模型在推論期間取得資料時，即模型處於生產環境並接收預測請求時，即不應存取資料洩漏，例如用於訓練的資料範例，或模型部署在生產環境中時無法使用的資訊。

如果您的模型在訓練資料上不小心遭到測試，資料外洩可能會導致過度擬合。過度擬合表示您的模型無法妥善一般化為看不見的資料。本節提供最佳實務，以避免資料外洩和過度擬合。

將您的資料分割成至少三個集

資料外洩的一個常見來源是在訓練期間不當分割（分割）您的資料。例如，資料科學家可能在知情或不知情的情況下，對用於測試的資料訓練模型。在這種情況下，您可能會觀察到過度擬合所造成的非常高的成功指標。若要解決此問題，您應該將資料分割成至少三個集：training、 validation和testing。

透過這種方式分割資料，您可以使用 validation集合來選擇和調整用來控制學習程序的參數 (超參數)。當您達到所需的結果或達到改善的穩定時，請對testing集合執行評估。testing 集合的效能指標應類似於其他集合的指標。這表示集合之間沒有分佈不相符的情況，且您的模型預期會在生產環境中妥善進行一般化。

使用分層分割演算法

當您將小型testing資料集的資料分割為 training、 validation和 時，或當您使用高度不平衡的資料時，請務必使用分層分割演算法。分層保證每個分割包含每個分割的大致相同數量或類別分佈。scikit-learn ML 程式庫已實作分層，Apache Spark 也是如此。

對於樣本大小，請確定驗證和測試集有足夠的資料可供評估，因此您可以得出統計上顯著的結論。例如，相對較小的資料集（少於 100 萬個範例）的常見分割大小為 70%、15% 和 15%，適用於 validation、 training和 testing。對於非常大型的資料集（超過 100 萬個範例），您可以使用 90%、5% 和 5% 來最大化可用的訓練資料。

在某些使用案例中，將資料分割為額外的集合會很有用，因為生產資料在收集期間可能發生極端、突然的分佈變更。例如，考慮為雜貨存放區項目建置需求預測模型的資料收集程序。如果資料科學團隊training在 2019 年期間收集資料，以及從 2020 年 1 月到 2020 年 3 月testing的資料，則模型可能會在testing集合中取得很好的分數。不過，在生產環境中部署模型時，某些項目的消費者模式會因為 COVID-19 大流行而大幅變更，而且模型會產生不佳的結果。在此案例中，新增另一個集合（例如 recent_testing) 作為模型核准的額外保護是合理的。此新增可能會阻止您核准因分佈不相符而立即效能不佳的生產模型。

在某些情況下，您可能想要建立其他 validation或 testing集合，其中包含特定類型的範例，例如與少數族群相關聯的資料。這些資料範例對於正確設定很重要，但可能無法在整體資料集中妥善呈現。這些資料子集稱為配量。

考慮使用 ML 模型進行信用分析的情況，該模型根據整個國家/地區的資料進行訓練，並平衡以平均地考慮目標變數的整個網域。此外，請考慮此模型可能具有 City功能。如果使用此模型的銀行將其業務

擴展到特定城市，則可能對模型在該區域的執行方式感興趣。因此，核准管道不僅應該根據整個國家/地區的測試資料來評估模型的品質，還應該評估特定城市配量的測試資料。

當資料科學家在新模型上工作時，他們可以透過在模型的驗證階段中整合代表性不足的配量，輕鬆評估模型的功能並考慮邊緣案例。

執行隨機分割時，請考慮重複的範例

另一個較不常見的洩漏來源位於可能包含太多重複樣本的資料集。在這種情況下，即使您將資料分割為子集，不同的子集可能會有相同的範例。根據重複項目的數量，過度擬合可能會誤認為一般化。

考慮在生產環境中接收推論時可能無法使用的功能

當模型使用生產環境中無法使用的功能進行訓練時，也會發生資料外洩，同時叫用推論。由於模型通常是根據歷史資料建置的，因此此資料可能會充實一些時間點不存在的其他資料欄或值。考慮信用核准模型的情況，該模型具有追蹤客戶在過去六個月內向銀行進行多少貸款的功能。如果此模型已部署並用於沒有銀行六個月歷史記錄的新客戶的信用核准，則會有資料外洩的風險。

Amazon SageMaker AI Feature Store 有助於解決此問題。您可以使用時間歷程查詢來更準確地測試模型，您可以使用這些查詢來檢視特定時間點的資料。

功能存放區

使用 SageMaker AI Feature Store 可提高團隊生產力，因為它會分離元件邊界（例如，儲存體與用量）。它還提供組織內不同資料科學團隊的功能可重複使用性。

使用時間歷程查詢

Feature Store 中的時間歷程功能有助於重現模型建置，並支援更強大的控管實務。這在組織想要評估資料譜系時很有用，類似於 Git 評估程式碼等版本控制工具的方式。時間歷程查詢也有助於組織為合規檢查提供準確的資料。如需詳細資訊，請參閱 AWS Machine Learning部落格上的了解 Amazon SageMaker AI Feature Store 的主要功能。

使用 IAM 角色

Feature Store 也有助於提高安全性，而不會影響團隊的生產力和創新。您可以使用 AWS Identity and Access Management (IAM) 角色為特定使用者或群組提供或限制對特定功能的精細存取。

例如，下列政策會限制對 Feature Store 中敏感功能的存取。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket--usw2-az1--x-s3/12345678910/
sagemaker/us-east-2/offline-store/doctor-appointments"
        }
    ]
}
```

如需使用特徵商店進行資料安全和加密的詳細資訊，請參閱 SageMaker AI 文件中的安全和存取控制。

使用單位測試

當資料科學家根據某些資料建立模型時，他們通常會假設資料的分佈，或執行徹底的分析以完全了解資料屬性。部署這些模型時，模型最終會過時。當資料集過期時，資料科學家、ML 工程師和 （在某些情況下） 自動化系統會使用從線上或離線存放區擷取的新資料重新訓練模型。

不過，此新資料的分佈可能已變更，這可能會影響目前演算法的效能。檢查這些類型問題的自動化方法是從軟體工程借用單元測試的概念。要測試的常見項目包括遺失值的百分比、分類變數的基數，以及實際值資料欄是否遵循某些預期的分佈，方法是使用假設測試統計資料 (t 測試)。您可能也想要驗證資料結構描述，以確保其未變更，也不會以無提示方式產生無效的輸入功能。

單元測試需要了解資料及其網域，以便您可以規劃在 ML 專案中執行的確切聲明。如需詳細資訊，請參閱 AWS 大數據部落格上的使用 PyDeequ 大規模測試資料品質。

培訓

MLOps 關心 ML 生命週期的操作性。因此，它必須促進資料科學家和資料工程師的工作，以建立務實的模型來滿足業務需求，並長期運作良好，而不會產生技術債務。

遵循本節中的最佳實務，以協助解決模型訓練挑戰。

主題

建立基準模型

當從業人員面臨 ML 解決方案的業務問題時，他們的第一個傾向通常是使用state-of-the-art演算法。這種做法具有風險，因為state-of-the-art演算法可能尚未經過時間測試。此外，state-of-the-art演算法通常更複雜，而且尚未充分了解，因此它可能只會比更簡單的替代模型帶來邊際改善。更好的做法是建立基準模型，以相對快速的速度進行驗證和部署，並且可以獲得專案利益相關者的信任。

當您建立基準時，我們建議您盡可能評估其指標效能。將基準模型的效能與其他自動化或手動系統進行比較，以確保其成功，並確保模型實作或專案可以中長期交付。

基準模型應與 ML 工程師進一步驗證，以確認模型可以交付為專案建立的非功能需求，例如推論時間、資料預期轉移分佈的頻率、是否可以在這些情況下輕鬆重新訓練模型，以及部署模型的方式，這將影響解決方案的成本。取得這些問題的多學科觀點，以增加您開發成功且長期執行模型的機會。

資料科學家可能傾向於將盡可能多的功能新增至基準模型。雖然這會提高模型預測所需結果的能力，但其中一些功能可能只會產生增量指標改進。許多功能，尤其是高度相關的功能，可能是冗餘的。新增太多功能會增加成本，因為它需要更多的運算資源和調校。太多功能也會影響模型的day-to-day操作，因為資料偏離變得更可能或發生得更快。

請考慮兩個輸入特徵高度相關，但只有一個特徵具有因果關係的模型。例如，預測貸款預設是否可能有輸入特徵的模型，例如客戶年齡和收入，這些特徵可能高度相關，但只應使用收入來提供或拒絕貸款。

根據這兩個功能訓練的模型可能依賴沒有因果關係的功能，例如年齡，來產生預測輸出。如果模型在生產之後，收到比訓練集平均年齡更長或更小的客戶推論請求，則可能會開始執行不佳。

此外，每個個別功能在生產時都可能經歷分佈轉移，並導致模型發生意外行為。基於這些原因，模型具有的功能越多，就越脆弱，就越傾向和過時。

資料科學家應使用關聯性指標和 Shapley 值來衡量哪些特徵為預測增加足夠的值，並應予以保留。擁有這種複雜的模型會增加回饋迴圈的機會，其中模型會變更其建模的環境。例如，消費者行為可能因為模型的建議而變更的建議系統。跨模型運作的回饋迴圈較不常見。例如，請考慮推薦電影的建議系統，以及推薦書籍的另一個系統。如果兩個模型都以相同的一組消費者為目標，它們會互相影響。

針對您開發的每個模型，請考慮哪些因素可能導致這些動態，因此您知道要在生產環境中監控哪些指標。

使用以資料為中心的方法和錯誤分析

如果您使用簡單的模型，您的 ML 團隊可以專注於改善資料本身，並採取以資料為中心的方法，而不是以模型為中心的方法。如果您的專案使用非結構化資料，例如影像、文字、音訊和其他可由人類評估的格式（相較於結構化資料，這可能更難以有效地映射到標籤），則最好能取得更好的模型效能，以執行錯誤分析。

錯誤分析涉及在驗證集上評估模型，以及檢查最常見的錯誤。這有助於識別類似資料範例的潛在群組，而模型可能無法正確進行。若要執行錯誤分析，您可以列出具有較高預測錯誤的推論，或將某個類別的範例預測為來自另一個類別的錯誤排名。

建構模型以進行快速迭代

當資料科學家遵循最佳實務時，他們可以使用新的演算法進行實驗，或在概念驗證或甚至重新訓練期間，輕鬆快速地混合和比對不同的功能。此實驗有助於生產成功。最佳實務是建立在基準模型的基礎上，採用稍微複雜的演算法並反覆新增功能，同時監控訓練和驗證集的效能，以比較實際行為與預期行為。此訓練架構可在預測能力上提供最佳平衡，並有助於讓模型盡可能簡單，且技術債務足跡更小。

為了快速迭代，資料科學家必須交換不同的模型實作，以判斷特定資料使用的最佳模型。如果您有大型團隊、短暫的截止日期和其他與專案管理相關的物流，如果沒有方法，快速反覆運算可能會很困難。

在軟體工程中，Liskov 替代原則是一種建構軟體元件之間互動的機制。此原則說明您應該能夠將介面的一個實作取代為另一個實作，而不會中斷用戶端應用程式或實作。當您為 ML 系統編寫訓練程式碼時，您可以使用此原則來建立邊界並封裝程式碼，以便您可以輕鬆取代演算法，並更有效地嘗試新的演算法。

例如，在以下程式碼中，您只需新增類別實作，即可新增實驗。

```
from abc import ABC, abstractmethod

from pandas import DataFrame


class ExperimentRunner(object):

    def __init__(self, *experiments):
        self.experiments = experiments

    def run(self, df: DataFrame) -> None:
        for experiment in self.experiments:
            result = experiment.run(df)
            print(f'Experiment "{experiment.name}" gave result {result}')


class Experiment(ABC):

    @abstractmethod
    def run(self, df: DataFrame) -> float:
        pass

    @property
    @abstractmethod
    def name(self) -> str:
        pass


class Experiment1(Experiment):

    def run(self, df: DataFrame) -> float:
        print('performing experiment 1')
        return 0

    def name(self) -> str:
        return 'experiment 1'


class Experiment2(Experiment):

    def run(self, df: DataFrame) -> float:
        print('performing experiment 2')
```

```
        return 0

    def name(self) -> str:
        return 'experiment 2'


class Experiment3(Experiment):

    def run(self, df: DataFrame) -> float:
        print('performing experiment 3')
        return 0

    def name(self) -> str:
        return 'experiment 3'


if __name__ == '__main__':
    runner = ExperimentRunner(*[
        Experiment1(),
        Experiment2(),
        Experiment3()
    ])
    df = ...
    runner.run(df)
```

追蹤您的 ML 實驗

當您使用大量實驗時，請務必衡量您觀察到的改善是否是實作變更或機會的乘積。您可以使用 Amazon SageMaker AI Experiments 輕鬆建立實驗，並將中繼資料與其建立關聯，以進行追蹤、比較和評估。

降低模型建置程序的隨機性對於偵錯、疑難排解和改善控管非常有用，因為考慮到相同的程式碼和資料，您可以更確定地預測輸出模型推論。

由於隨機權重初始化、平行運算同步性、內部 GPU 複雜性和類似的非確定性因素，因此通常無法讓訓練程式碼完全可重現。不過，正確設定隨機種子，以確保每個訓練執行從相同的點開始，並採取類似的行為，可大幅改善結果可預測性。

訓練任務疑難排解

在某些情況下，資料科學家可能很難適應非常簡單的基準模型。在這種情況下，他們可能會決定他們需要更適合複雜函數的演算法。一個好的測試是使用資料集非常小部分的基準（例如，約 10 個範例），以確保演算法過度適合此範例。這有助於排除資料或程式碼問題。

另一個用於偵錯複雜案例的實用工具是 Amazon SageMaker AI Debugger，可以擷取演算法正確性和基礎設施相關問題，例如最佳運算用量。

部署

在軟體工程中，將程式碼放入生產環境需要盡職調查，因為程式碼可能會意外發生，未預期的使用者行為可能會中斷軟體，並可能發現未預期的邊緣案例。軟體工程師和 DevOps 工程師通常會採用單元測試和復原策略來降低這些風險。使用 ML，將模型放入生產環境需要更多規劃，因為實際環境預期會偏離，而且在許多情況下，模型會根據他們嘗試改善的實際業務指標代理的指標進行驗證。

請遵循本節中的最佳實務，以協助解決這些挑戰。

主題

自動化部署週期

訓練和部署程序應完全自動化，以防止人為錯誤，並確保建置檢查持續執行。使用者不應擁有生產環境的寫入存取許可。

Amazon SageMaker AI Pipelines 並AWS CodePipeline協助為 ML 專案建立 CI/CD 管道。使用 CI/CD 管道的優點之一是，所有用來擷取資料、訓練模型和執行監控的程式碼都可以使用 Git 等工具控制版本。有時候，您必須使用相同的演算法和超參數來重新訓練模型，但資料不同。驗證您使用的演算法版本是否正確的唯一方法是使用來源控制和標籤。您可以使用 SageMaker AI 提供的預設專案範本作為 MLOps 實務的起點。

當您建立 CI/CD 管道來部署模型時，請務必使用組建識別符、程式碼版本或遞交和資料版本來標記組建成品。此實務可協助您疑難排解任何部署問題。對於在高度管制欄位中進行預測的模型，有時還需要標記。回溯工作並識別與 ML 模型相關聯的確切資料、程式碼、建置、檢查和核准的能力，有助於大幅改善控管。

CI/CD 管道任務的一部分是對正在建置的項目進行測試。雖然資料單位測試預期會在特徵存放區擷取資料之前進行，但管道仍需負責對指定模型的輸入和輸出執行測試，以及檢查關鍵指標。這類檢查的範例之一是驗證固定驗證集上的新模型，並使用已建立的閾值來確認其效能與先前模型相似。如果效能明顯低於預期，則建置應該會失敗，且模型不應進入生產環境。

CI/CD 管道的廣泛使用也支援提取請求，這有助於防止人為錯誤。當您使用提取請求時，每個程式碼變更都必須至少由其他一位團隊成員審查和核准，才能進入生產環境。提取請求也有助於識別不遵守業務規則的程式碼，以及在團隊中傳播知識。

選擇部署策略

MLOps 部署策略包括藍/綠、金絲雀、陰影和 A/B 測試。

藍/綠

藍/綠部署在軟體開發中非常常見。在此模式下，兩個系統在開發期間會保持執行：藍色是舊環境（在此案例中，是要取代的模型），而綠色是即將生產的新發行模型。因為舊系統保持運作狀態，所以可以輕鬆地復原變更，並縮短停機時間。如需 SageMaker 內容中藍/綠部署的更深入資訊，請參閱部落格文章中的使用 和 安全地部署 AWS CodePipeline 和監控 Amazon SageMaker AI 端點 AWS CodeDeploy，網址為 AWS Machine Learning部落格。

Canary

Canary 部署與 中的藍/綠部署類似，這兩者會保持兩個模型一起執行。不過，在 Canary 部署中，新模型會逐步推出給使用者，直到所有流量最終轉移到新模型為止。與藍/綠部署一樣，由於新的（和潛在的故障）模型在初始推出期間受到密切監控，並且可以在發生問題時復原，因此降低風險。在 SageMaker AI 中，您可以使用 InitialVariantWeight API 指定初始流量分佈。

影子

您可以使用陰影部署來安全地將模型帶入生產環境。在此模式中，新模型可與較舊的模型或業務流程搭配使用，並在不影響任何決策的情況下執行推論。在您將模型提升至生產環境之前，此模式可以做為最終檢查或更高保真度實驗。

當您不需要任何使用者推論意見回饋時，陰影模式非常有用。您可以執行錯誤分析，並將新模型與舊模型進行比較，藉此評估預測品質，並且可以監控輸出分佈，確認其是否如預期。若要了解如何使用 SageMaker AI 進行陰影部署，請參閱 AWS Machine Learning 部落格》中的部落格文章在 Amazon SageMaker AI 中部署陰影 ML 模型。

A/B 測試

當 ML 從業人員在其環境中開發模型時，他們最佳化的指標通常是真正重要的業務指標代理。這使得難以確定新模型是否實際改善業務成果，例如營收和點擊率，並減少使用者投訴的數量。

考慮電子商務網站的情況，其中的業務目標是盡可能銷售更多產品。審核團隊知道銷售和客戶滿意度與資訊豐富且準確的審核直接相關。團隊成員可能會提出新的審核排名演算法，以改善銷售額。透過使用 A/B 測試，他們可以將新舊演算法轉出至不同但類似的使用者群組，並監控結果，以查看從較新模型收到預測的使用者是否更有可能進行購買。

A/B 測試也有助於衡量模型過時和偏離對業務的影響。團隊可以在一些循環的情況下將新模型放入生產環境，對每個模型執行 A/B 測試，並建立年齡與效能圖表。這有助於團隊了解其生產資料中的資料偏離波動。

如需如何使用 SageMaker AI 執行 A/B 測試的詳細資訊，請參閱機器學習部落格上的部落格文章 A/B 測試生產環境中使用 Amazon SageMaker AI 的 ML 模型 AWS Machine Learning 。

考慮您的推論需求

使用 SageMaker AI，您可以選擇基礎基礎設施以不同方式部署模型。這些推論調用功能支援不同的使用案例和成本描述檔。您的選項包括即時推論、非同步推論和批次轉換，如下列各節所述。

即時推論

即時推論非常適合具有即時、互動式、低延遲需求的推論工作負載。您可以將模型部署到 SageMaker AI 託管服務，並取得可用於推論的端點。這些端點受到完整管理，支援自動擴展 （ 請參閱自動擴展 Amazon SageMaker AI 模型)，並且可以部署在多個可用區域中。

如果您有使用 Apache MXNet、PyTorch 或 TensorFlow 建置的深度學習模型，您也可以使用 Amazon SageMaker AI Elastic Inference (EI)。使用 EI，您可以將分數 GPUs 連接到任何 SageMaker AI 執行個體，以加速推論。您可以選擇用戶端執行個體來執行應用程式，並連接 EI 加速器，以針對推論需求使用正確數量的 GPU 加速。

另一個選項是使用多模型端點，提供可擴展且符合成本效益的解決方案來部署大量模型。這些端點使用共用服務容器，該容器已啟用來託管多個模型。與使用單一模型端點相比，多模型端點透過改善端點使用率來降低託管成本。它們也會降低部署開銷，因為 SageMaker AI 會管理記憶體中的載入模型，並根據流量模式進行擴展。

如需在 SageMaker AI 中部署 ML 模型的其他最佳實務，請參閱 SageMaker AI 文件中的部署最佳實務。

非同步推論

Amazon SageMaker AI 非同步推論是 SageMaker AI 中的功能，可將傳入的請求排入佇列並以非同步方式處理。此選項適用於承載大小高達 1 GB、處理時間長和近乎即時延遲需求的請求。非同步推論可讓您在無處理請求時，自動將執行個體計數擴展到零，以節省成本，因此只有在端點正在處理請求時，才需要付費。

批次轉換

當您想要執行下列動作時，請使用批次轉換：

- 預先處理資料集，以移除干擾資料集訓練或推論的雜訊或偏差。
- 從大型資料集取得推論。
- 當您不需要持久性端點時，執行推論。
- 將輸入記錄與推論建立關聯，以協助解讀結果。

監控

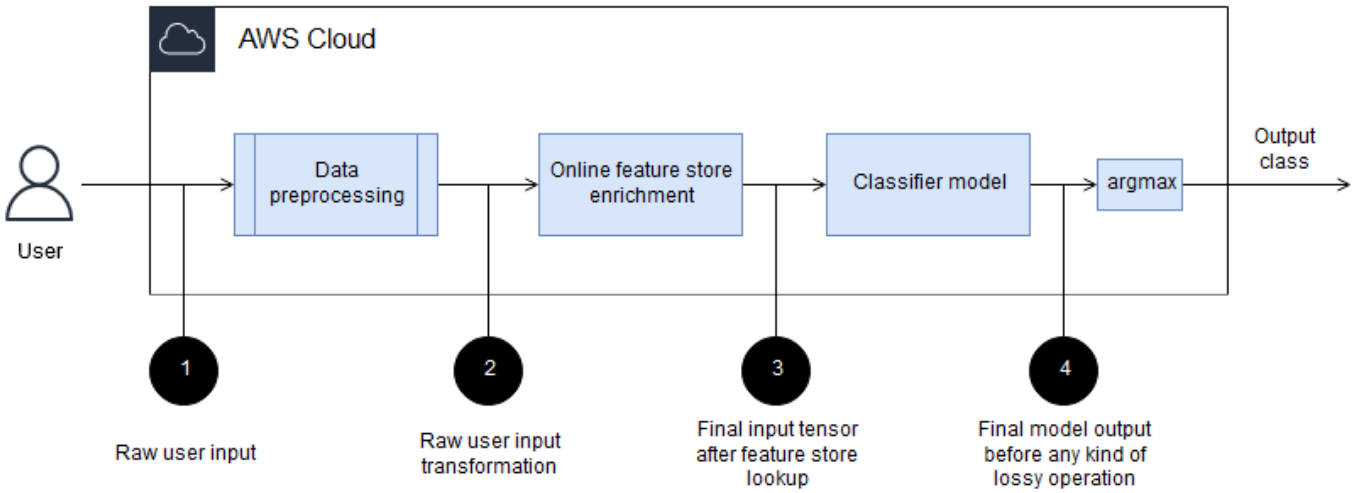
當模型已在生產中並提供商業價值時，請執行持續檢查，以識別何時必須重新訓練模型或對其採取動作。

您的監控團隊應該主動採取行動，而不是反應，以更了解環境的資料行為，並識別資料偏離的頻率、速率和突發性。團隊應該在資料中識別新的邊緣案例，這些案例可能會在訓練集、驗證集和其他邊緣案例配量中代表性不足。他們應該存放服務品質 (QoS) 指標、使用警示在發生問題時立即採取行動，並定義擷取和修改目前資料集的策略。這些實務從記錄模型的請求和回應開始，以提供故障診斷或其他洞見的參考。

理想情況下，資料轉換應在處理期間記錄於幾個關鍵階段：

- 在任何類型的預處理之前
- 在任何類型的特徵存放區擴充之後
- 在模型的所有主要階段之後
- 在模型輸出上任何類型的失真函數之前，例如 `argmax`

下圖說明這些階段。



您可以使用 SageMaker AI Model Monitor 自動擷取輸入和輸出資料，並將其存放在 Amazon Simple Storage Service (Amazon S3) 中。您可以將日誌新增至自訂服務容器，以實作其他類型的中繼記錄。

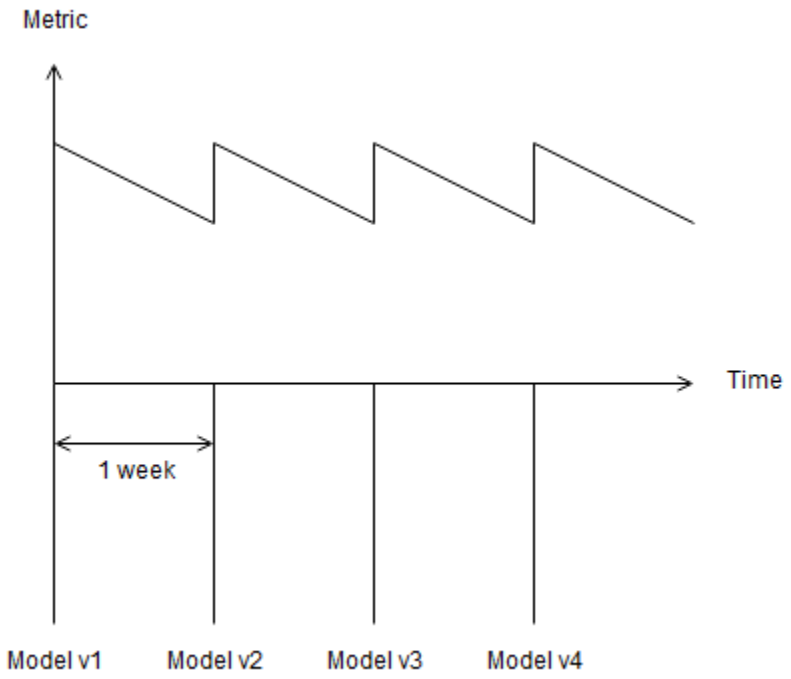
從模型記錄資料後，您可以監控分佈偏離。在某些情況下，您可以在推論後立即取得地面實況（正確標記的資料）。常見的範例是預測要向使用者顯示之最相關廣告的模型。一旦使用者離開頁面，您就可以判斷他們是否按一下廣告。如果使用者已按一下廣告，您可以記錄該資訊。在此簡單範例中，您可以使用可在訓練和部署中測量的指標，輕鬆量化模型的成功程度，例如準確性或 F1。如需有關您已標記資料之案例的詳細資訊，請參閱 SageMaker AI 文件中的監控模型品質。不過，這些簡單的案例並不常見，因為模型通常設計為最佳化數學上方便的指標，這些指標僅代表實際的業務成果。在這種情況下，最佳實務是在生產環境中部署模型時監控業務成果。

考慮檢閱排名模型的情況。如果 ML 模型的定義業務成果是在網頁頂端顯示最相關且實用的評論，您可以新增按鈕來衡量模型的成功，例如「這麼做有幫助嗎？」 每次檢閱。測量此按鈕的點擊率可能是業務成果指標，可協助您測量模型在生產中的表現。

若要監控 SageMaker AI 中輸入或輸出標籤的偏離，您可以使用 SageMaker AI Model Monitor 的資料品質功能，同時監控輸入和輸出。您也可以建置自訂容器，為 SageMaker AI Model Monitor 實作自己的邏輯。

監控模型在開發時間和執行時間接收的資料至關重要。工程師不僅應監控結構描述變更的資料，還應監控分佈不相符的資料。偵測結構描述變更比較簡單，而且可由一組規則實作，但分佈不相符通常較為棘手，尤其是因為需要您定義閾值來量化何時發出警示。在已知受監控分佈的情況下，通常最簡單的方法是監控分佈的參數。在常態分佈的情況下，即平均值和標準差。其他關鍵指標，例如遺失值的百分比、最大值和最小值也很有用。

您也可以建立持續監控任務，以取樣訓練資料和推論資料，並比較其分佈。您可以為模型輸入和模型輸出建立這些任務，並根據時間繪製資料，以視覺化任何突然或逐漸偏離。下圖說明此項目。



為了更了解資料的偏離描述檔，例如資料分佈的顯著變更頻率、速率或頻率，我們建議您持續部署新的模型版本並監控其效能。例如，如果您的團隊每週部署新的模型，並觀察到模型效能每次都大幅改善，他們可以判斷應該至少在一週內交付新的模型。

後續步驟和資源

本指南會逐步引導您在規劃要帶入生產環境的機器學習模型生命週期時，了解一些考量事項。它討論了四個領域的挑戰和最佳實務，包括資料、訓練、部署和監控，並包含其他相關的資源。

AWS 提供 Well-Architected 架構，可協助雲端架構師為各種應用程式、工作負載和技術網域建置安全、高效能、彈性且高效率的基礎設施。如需其他閱讀，請參閱 AWS Well-Architected 提供的 Machine Learning Lens。

資源

Amazon SageMaker AI 文件

- Amazon SageMaker AI 功能商店
- 功能存放區安全性和存取控制
- Shapley 值
- Amazon SageMaker AI Debugger
- Amazon SageMaker AI 管道
- Amazon SageMaker AI 預設專案範本
- SageMaker AI 即時推論
- 自動擴展 Amazon SageMaker AI 模型
- Amazon SageMaker AI 非同步推論
- SageMaker AI 模型監控

AWS 開發人員工具

- AWS CodePipeline

AWS 部落格文章

- 了解 Amazon SageMaker AI Feature Store 的主要功能
- 使用 PyDeequ 大規模測試資料品質
- Amazon SageMaker AI 實驗
- 使用 CodePipeline 和 安全地部署和監控 Amazon SageMaker 端點 AWS CodeDeploy

- 在 Amazon SageMaker AI 中部署影子 ML 模型
- 使用 Amazon SageMaker AI 在生產環境中測試 ML 模型的 A/B

文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 RSS 摘要。

變更	描述	日期
初次出版	—	2021 年 12 月 20 日

AWS 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的術語。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：將您的現場部署 Oracle 資料庫 遷移至 Amazon Aurora PostgreSQL 相容版本。
- 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：將內部部署 Oracle 資料庫 遷移至 中的 Amazon Relational Database Service (Amazon RDS) for Oracle 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：將您的現場部署 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到相同平台的雲端服務。範例：將 Microsoft Hyper-V應用程式 遷移至 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

A2A Agent-to-Agent)

支援任務委派和狀態轉移的agent-to-agent協同合作的狀態通訊協定。

ABAC

請參閱屬性型存取控制。

抽象服務

請參閱 受管服務。

ACID

請參閱原子性、一致性、隔離性、耐久性。

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，但需要比主動-被動遷移更多的工作。

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，但只有來源資料庫會在資料複寫至目標資料庫時處理來自連線應用程式的交易。目標資料庫在遷移期間不接受任何交易。

客服人員

一種 AI 系統，可以使用工具自動推理、規劃和採取行動來實現目標。

客服人員操作

在生產環境中大規模建置、測試、部署和執行 AI 代理器的操作實務。

彙總函數

在一組資料列上運作的 SQL 函數，會計算群組的單一傳回值。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

永久刪除資料集中個人資訊的程序。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

經常用於經常性問題的解決方案，其中解決方案具有反生產力、無效或比替代解決方案更有效。

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攻擊。

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 (AIOps)

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。如需有關如何在 AWS 遷移策略中使用 AIOps 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。

原子性、一致性、隔離性、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》中的 ABAC for AWS。

授權資料來源

存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、修訂或假名化資料。

可用區域

中的不同位置 AWS 區域 ，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、低延遲網路連線能力。

AWS 雲端採用架構 (AWS CAF)

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，以成功地移至雲端。 AWS CAF 將指導方針組織到六個重點領域：業務、人員、治理、平台、安全和營運。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。因此， AWS CAF 為人員開發、訓練和通訊提供指引，協助組織做好成功採用雲端的準備。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預估值的工具。 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

請參閱業務持續性規劃。

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。您可以將行為圖與 Amazon Detective 搭配使用來檢查失敗的登入嘗試、可疑的 API 呼叫和類似動作。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 Endianness。

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。此策略可協助您快速復原，並將影響降至最低。

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。有些機器人有用或有益，例如在網際網路上編製資訊索引的 Web 爬蟲程式。某些其他機器人稱為惡意機器人，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受單一方控制之機器人的網路，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，讓使用者快速取得他們通常無權存取 AWS 帳戶 之 的存取權。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS 實作打破玻璃程序指標。

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

版本對最終使用者的緩慢和增量版本。當您有信心時，您可以部署新版本，並完全取代目前的版本。

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。您可以將 CDC 用於各種用途，例如稽核或複寫目標系統中的變更以保持同步。

混沌工程

故意引入故障或破壞性事件，以測試系統的彈性。您可以使用 AWS Fault Injection Service (AWS FIS) 執行實驗，為您的 AWS 工作負載帶來壓力，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。

公民開發人員

在沒有專業技術技能的情況下，使用無程式碼/低程式碼平台建立 AI 應用程式的商業使用者。

用戶端加密

在目標 AWS 服務 接收資料之前，在本機加密資料。

雲端卓越中心 (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雲端採用 (例如，建立登陸區域、定義 CCoE、建立營運模型)
- 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

部落格文章中的 Stephen Orban 定義了這些階段： AWS 雲端 企業策略部落格上的邁向雲端優先之旅和採用階段。如需有關它們如何與 AWS 遷移策略關聯的資訊，請參閱遷移整備指南。

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。程式碼的每個版本都稱為分支。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

很少存取且通常是歷史資料的資料。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成本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影像和影片等視覺化格式分析和擷取資訊的 AI 欄位。例如，Amazon SageMaker AI 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。您通常在遷移的產品組合探索和分析階段使用 CMDB 中的資料。

一致性套件

您可以組合的 AWS Config 規則和修補動作集合，以自訂您的合規和安全檢查。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 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。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。

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更。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，提供分散式、分散式資料擁有權與集中式管理。

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 上建置資料周邊 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

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

資料主體

正在收集和處理其資料的個人。

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。

資料庫處理語言 (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。

深度防禦

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，defense-in-depth方法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 成員帳戶來管理組織的帳戶，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。

deployment

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)

一種程序，用於識別對軟體開發生命週期中的速度和品質造成負面影響的限制並排定優先順序。DVSM 擴展了最初專為精簡製造實務設計的價值串流映射程序。它著重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

虛擬呈現真實世界的系統，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

在星星結構描述中，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，其行為類似於文字。這些屬性通常用於查詢限制、篩選和結果集標記。

災難

防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，例如意外設定錯誤或惡意軟體攻擊。

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 上工作負載災難復原 AWS：雲端中的復原。

DML

請參閱資料庫處理語言。

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。如需有關如何將領域驅動的設計與 strangler fig 模式搭配使用的資訊，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。

偏離偵測

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偏離，也可以使用 AWS Control Tower 來偵測登陸區域中可能影響控管要求合規性的變更。 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。

EDI

請參閱電子資料交換。

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改善回應時間。

電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱什麼是電子資料交換。

加密

將人類可讀取的純文字資料轉換為加密文字的運算程序。

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) 委託人。這些帳戶或主體可以透過建立介面 VPC 端點私下連接至您的端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud (Amazon VPC) 文件中的建立端點服務。

企業資源規劃 (ERP)

一種系統，可自動化和管理企業的關鍵業務流程 （例如會計、MES 和專案管理）。

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 安全概念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。透過計算摘要統計並建立資料可視化來執行 EDA。

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：包含度量的資料，以及包含維度資料表外部索引鍵的資料欄。

快速失敗

一種使用頻繁和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，像是可用區域 AWS 區域、控制平面或資料平面等界限會限制故障的影響，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。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱 的機器學習模型可解譯性 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。

少量擷取提示

在要求 LLM 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。對於需要特定格式、推理或網域知識的任務，少量的提示非常有效。另請參閱零鏡頭提示。

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，以盡可能在最短的時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

FM

請參閱基礎模型。

基礎模型 (FM)

大型深度學習神經網路，已針對廣義和未標記資料的大量資料集進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及自然語言的交談。如需詳細資訊，請參閱什麼是基礎模型。

FM 閘道

控制和標準化基礎模型存取的集中式中介裝置。也稱為 LLM 閘道。

G

生成式 AI

已針對大量資料進行訓練的 AI 模型子集，可使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱什麼是生成式 AI。

地理封鎖

請參閱地理限制。

地理限制 (地理封鎖)

Amazon CloudFront 中的選項，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，而以幹線為基礎的工作流程是現代、偏好的方法。

黃金影像

系統或軟體的快照，做為部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

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

有助於跨組織單位 (OU) 來管控資源、政策和合規的高層級規則。預防性防護機制會強制執行政策，以確保符合合規標準。透過使用服務控制政策和 IAM 許可界限來將其實作。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。它們是透過使用 AWS Config AWS Security Hub CSPM、Amazon GuardDuty、、Amazon Inspector AWS Trusted Advisor和自訂 AWS Lambda 檢查來實作。

護欄 (AI)

安全機制可篩選、驗證和限制代理程式輸入和輸出，以協助確保負責任且安全的 AI 行為。

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)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統的設計目的是自動容錯移轉、持續提供高品質的效能，並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

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種來源收集和存放資料。

保留資料

從用於訓練機器學習模型的資料集中保留的部分歷史標記資料。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

human-in-the-loop (HitL)

一種工作流程模式，其中代理程式執行會在關鍵決策點暫停進行人工審核和核准。

異質資料庫遷移

將您的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 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。

熱資料

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，通常會在典型 DevOps 發行工作流程之外執行修補程式。

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

IaC

請參閱基礎設施即程式碼。

身分型政策

連接至一或多個 IAM 主體的政策，可定義其在 AWS 雲端 環境中的許可。

閒置應用程式

90 天期間 CPU 和記憶體平均使用率在 5% 至 20% 之間的應用程式。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、修補或修改現有的基礎設施。不可變基礎設施本質上比可變基礎設施更一致、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 (輸入) VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，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

2016 年 Klaus Schwab 推出的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進展，指製造程序的現代化。

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，請參閱建立工業物聯網 (IIoT) 數位轉型策略。

檢查 VPC

在 AWS 多帳戶架構中，集中式 VPC，可管理 VPCs 之間 （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 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，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 的機器學習模型可解譯性 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 為 ITSM 提供了基礎。

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。如需有關將雲端操作與 ITSM 工具整合的資訊，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獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會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。

大型語言模型 (LLM)

預先訓練大量資料的深度學習 AI 模型。LLM 可以執行多個任務，例如回答問題、摘要文件、將文字翻譯成其他語言，以及完成句子。如需詳細資訊，請參閱什麼是 LLMs。

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 個 R。

小端序系統

首先儲存最低有效位元組的系統。另請參閱 Endianness。

LLM

請參閱大型語言模型。

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、間諜軟體和鍵盤記錄器。

受管服務

AWS 服務 會 AWS 操作基礎設施層、作業系統和平台，而您會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)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

MAP

請參閱遷移加速計劃。

MCP

請參閱模型內容通訊協定。

模型內容通訊協定 (MCP)

適用於代理程式對工具通訊的無狀態通訊協定。

MCP 伺服器

透過模型內容通訊協定公開一或多個工具的服務。

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。機制是在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的建置機制。

成員帳戶

屬於組織一部分的管理帳戶 AWS 帳戶 以外的所有 AWS Organizations。帳戶一次只能是一個組織的成員。

製造執行系統

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

一種小型的獨立服務，它可透過定義明確的 API 進行通訊，通常由小型獨立團隊擁有。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 API，透過明確定義的介面進行通訊。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)

此 AWS 計畫提供諮詢支援、訓練和服務，以協助組織建立強大的營運基礎，以移至雲端，並協助抵銷遷移的初始成本。MAP 包括用於有條不紊地執行舊式遷移的遷移方法以及一組用於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營運、業務分析師和擁有者、遷移工程師、開發人員以及從事 Sprint 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：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

遷移組合評定 (MPA)

線上工具，提供驗證商業案例以遷移至 的資訊 AWS 雲端。MPA 提供詳細的組合評定 (伺服器適當規模、定價、總體擁有成本比較、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，以及波次規劃)。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。

遷移準備程度評定 (MRA)

使用 AWS CAF 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。如需詳細資訊，請參閱遷移準備程度指南。MRA 是 AWS 遷移策略的第一階段。

遷移策略

用來將工作負載遷移至 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，並請參閱動員您的組織以加速大規模遷移。

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《》中的現代化應用程式的策略 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 - 統一架構。

開放程序通訊 - 統一架構 (OPC-UA)

用於工業自動化的machine-to-machine(M2M) 通訊協定。OPC-UA 提供資料加密、身分驗證和授權機制的互通性標準。

操作水準協議 (OLA)

一份協議，闡明 IT 職能群組承諾向彼此提供的內容，以支援服務水準協議 (SLA)。

操作整備審查 (ORR)

問題和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 中的操作準備度審查 (ORR)。

操作技術 (OT)

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造中，整合 OT 和資訊技術 (IT) 系統是工業 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 建立的線索 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有 的所有事件 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 CloudTrail 文件中的建立組織追蹤。

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 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。在 AWS 遷移策略中，此架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱 OCM 指南。

原始存取控制 (OAC)

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

原始存取身分 (OAI)

CloudFront 中的一個選項，用於限制存取以保護 Amazon S3 內容。當您使用 OAI 時，CloudFront 會建立一個可供 Amazon S3 進行驗證的主體。經驗證的主體只能透過特定 CloudFront 分發來存取 S3 儲存貯體中的內容。另請參閱 OAC，它可提供更精細且增強的存取控制。

ORR

請參閱操作整備審核。

OT

請參閱操作技術。

傳出 (輸出) VPC

在 AWS 多帳戶架構中，處理從應用程式內啟動之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，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

附接至 IAM 主體的 IAM 管理政策，可設定使用者或角色擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的許可界限。

個人身分識別資訊 (PII)

當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。PII 的範例包括名稱、地址和聯絡資訊。

PII

請參閱個人身分識別資訊。

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

請參閱可程式設計邏輯控制器。

PLM

請參閱產品生命週期管理。

政策

可定義許可的物件（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）。

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。

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。

設計隱私權

透過整個開發程序將隱私權納入考量的系統工程方法。

私有託管區域

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 查詢的資訊。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。這些控制項會在佈建資源之前對其進行掃描。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，並參閱實作安全控制項中的主動控制項。 AWS

產品生命週期管理 (PLM)

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

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。

提示鏈結

使用一個 LLM 提示的輸出作為下一個提示的輸入，以產生更好的回應。此技術用於將複雜任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和相關性，並允許更精細、個人化的結果。

擬匿名化

將資料集中的個人識別符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。

發佈/訂閱 (pub/sub)

一種模式，可啟用微服務之間的非同步通訊，以改善可擴展性和回應能力。例如，在微服務型 MES 中，微服務可以將事件訊息發佈到其他微服務可訂閱的頻道。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、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

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、諮詢、告知 (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 個 R。

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)

服務中斷與服務還原之間的可接受延遲上限。

重構

請參閱 7 個 R。

區域

地理區域中的 AWS 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 您的帳戶可以使用哪些。

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 個 R。

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 個 R。

Replatform

請參閱 7 個 R。

回購

請參閱 7 個 R。

彈性

應用程式抵禦中斷或從中斷中復原的能力。在 中規劃彈性時，高可用性和災難復原是常見的考量 AWS 雲端。如需詳細資訊，請參閱AWS 雲端 彈性。

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。

負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣

矩陣，定義所有參與遷移活動和雲端操作之各方的角色和責任。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，如果您排除它，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 個 R。

淘汰

請參閱 7 個 R。

檢索增強生成 (RAG)

一種生成式 AI 技術，其中 LLM 會在產生回應之前參考訓練資料來源以外的授權資料來源。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱什麼是 RAG。

輪換

定期更新秘密的程序，讓攻擊者更難存取登入資料。

資料列和資料欄存取控制 (RCAC)

使用已定義存取規則的基本、彈性 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 (IdP) 使用的開放標準。此功能會啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS 管理主控台 或呼叫 AWS API 操作，而不必為您組織中的每個人在 IAM 中建立使用者。如需有關以 SAML 2.0 為基礎的聯合詳細資訊，請參閱 IAM 文件中的關於以 SAML 2.0 為基礎的聯合。

斯卡達

請參閱監督控制和資料擷取。

SCP

請參閱服務控制政策。

秘密

您以加密形式存放的 AWS Secrets Manager機密或限制資訊，例如密碼或使用者登入資料。它包含秘密值及其中繼資料。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 秘密中的內容？ 在 Secrets Manager 文件中。

設計安全性

透過整個開發程序將安全性納入考量的系統工程方法。

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、回應性和主動性。

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。

安全資訊與事件管理 (SIEM) 系統

結合安全資訊管理 (SIM) 和安全事件管理 (SEM) 系統的工具與服務。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，並產生提醒。

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 執行個體或輪換登入資料。

伺服器端加密

由 AWS 服務 接收資料的 在其目的地加密資料。

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。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。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

的進入點 URL 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。

服務水準協議 (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。

陰影 AI

在組織內受管管道之外建置或使用的未授權 AI 應用程式。

SIEM

請參閱安全資訊和事件管理系統。

單點故障 (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

請參閱服務層級指標。

SLO

請參閱服務層級目標。

先拆分後播種模型

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

請參閱單一故障點。

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。此結構旨在用於資料倉儲或商業智慧用途。

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。如需有關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

您 VPC 中的 IP 地址範圍。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。

系統提示

一種向 LLM 提供內容、指示或指導方針以指示其行為的技術。系統提示有助於設定內容，並建立與使用者互動的規則。

T

標籤

做為中繼資料以組織 AWS 資源的鍵/值對。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。

tool

代理程式可以叫用以在外部系統中執行操作的函數或 API。

傳輸閘道

可以用於互連 VPC 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 AWS Organizations 及其帳戶中執行任務。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 搭配使用 AWS Organizations 與其他 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。

雙比薩團隊

兩個比薩就能吃飽的小型 DevOps 團隊。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。

未區分的任務

也稱為繁重工作，這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 對等互連

兩個 VPC 之間的連線，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 Amazon VPC 文件中的什麼是 VPC 對等互連。

漏洞

危害系統安全性的軟體或硬體瑕疵。

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，通常可接受中等速度的查詢。

視窗函數

SQL 函數，對與目前記錄在某種程度上相關的資料列群組執行計算。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，然後再遷移伺服器和應用程式。

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)

儲存模型，可一次性寫入資料，並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。

Z

零時差入侵

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瑕疵或漏洞。威脅行為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。

零鏡頭提示

提供 LLM 執行任務的指示，但沒有可協助引導任務的範例 (快照)。LLM 必須使用其預先訓練的知識來處理任務。零鏡頭提示的有效性取決於任務的複雜性和提示的品質。另請參閱少量擷取提示。

殭屍應用程式

CPU 和記憶體平均使用率低於 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。