



彈性分析架構

# AWS 規範指引



# AWS 規範指引: 彈性分析架構

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
架構概觀 .....	2
了解工作負載 .....	5
應用框架 .....	7
減輕潛在故障 .....	9
了解權衡和風險 .....	9
故障模式可觀測性 .....	11
常見的緩解策略 .....	11
持續改善 .....	15
結論和資源 .....	16
文件歷史紀錄 .....	17
詞彙表 .....	18
# .....	18
A .....	18
B .....	21
C .....	22
D .....	25
E .....	28
F .....	30
G .....	31
H .....	32
I .....	33
L .....	35
M .....	36
O .....	40
P .....	42
Q .....	44
R .....	45
S .....	47
T .....	50
U .....	51
V .....	52
W .....	52
Z .....	53

---

..... liv

# 彈性分析框架

約翰·福門托，布魯諾·埃默，史蒂芬·胡珀，傑森·巴托和邁克爾·哈肯，亞馬遜 Web 服務 (AWS)

二零二三年九月([文件歷史](#))

一致且可重複的標準和流程是持續改進的重要組成部分。這對於分佈式系統的彈性也是如此。本指南的目的是引入彈性分析架構，提供一致的方式來分析失敗模式，以及它們如何影響您的工作負載。從設計到作業，在工作負載的整個生命週期中使用此架構，可協助您以一致且可重複的方式，持續將工作負載的彈性提升到更廣泛的潛在故障模式。這有助於確保您符合彈性目標，並維持工作負載所需的彈性特性。

此架構是透過 AWS 解決方案架構現場團隊與各行各業客戶合作的豐富經驗而開發的。它針對可以擁有許多職位的製造商，包括產品經理，軟件開發人員，系統工程師，運營團隊和架構師。這些是最了解正在分析的系統，服務或產品的人。在持續練習中使用該框架可以幫助您取得漸進的進步並滿足您的長期彈性目標。

架構的重點在於識別潛在的失敗模式，以及可用來減輕其影響的預防性和糾正控制項。即使不是直接受您控制的元件發生故障，例如增加相依性中的錯誤率，您也需要考慮這些失敗對工作負載的影響，以及如何設計該工作負載以回應這些失敗。最終，您應該專注於您可以回應的失敗通過使用您控制之下的緩解措施。

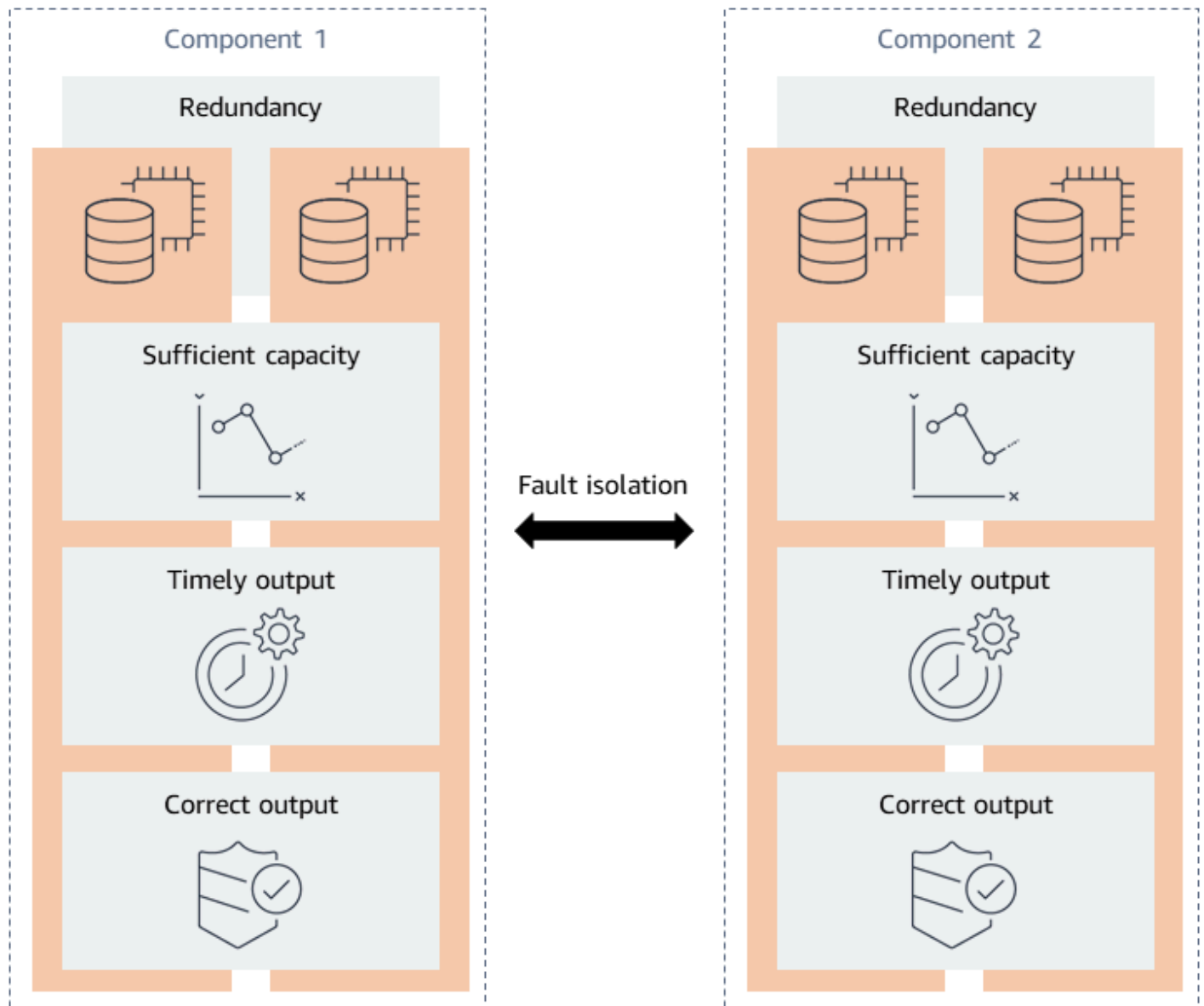
本指南概述了框架，然後討論如何識別和記錄工作負載，如何將框架應用於該工作負載，以及如何評估您發現的任何潛在故障的緩解策略。

## 內容

- [框架概述](#)
- [了解工作負載](#)
- [應用框架](#)
- [減輕潛在故障](#)
- [結論和資源](#)

## 架構概觀

彈性分析架構是透過識別工作負載所需的彈性特性而開發的。所需的屬性是您希望對系統成為真實的事物。恢復性通常是根據可用性來衡量，因此五個屬性是高可用性分散式系統的特性：備援、足夠的容量、及時輸出、正確的輸出以及故障隔離。這些屬性如下圖所示。



- 冗餘—容錯是透過可消除單點故障 (SPOFs) 的冗餘來實現。備援能夠從工作負載中的備用元件到整個應用程式堆疊的完整複本。當您考慮應用程式的備援時，請務必考慮您使用的基礎結構、資料存放區和相依性所提供的備援層級。例如，Amazon DynamoDB 和 Amazon 簡單儲存服務 (Amazon S3) 透過在一個區域中跨多個可用區域複寫資料來提供備援，以及AWS Lambda在多個可用區域中的多

個工作者節點上執行您的函數。對於您使用的每項服務，請考慮服務提供的內容以及您需要設計的內容。

- **足夠容量**— 您的工作負載需要足夠的資源才能正常運作。資源包括記憶體、CPU 週期、執行緒、儲存體、輸送量、服務配額和許多其他資源。
- **及時輸出**— 當客戶使用您的工作負載時，他們希望它能夠在合理的時間內執行其預期功能。除非服務針對延遲提供服務等級協定 (SLA)，否則他們的期望通常基於實證證據 — 也就是他們自己的經驗。這個平均客戶體驗通常被認為是系統中位數 (P50) 延遲。如果您的工作負載所花費的時間超過預期，此延遲可能會影響客戶的體驗。
- **正確的輸出**— 需要正確輸出工作負載軟體，才能提供其預期的功能。不正確或不完整的結果可能比根本沒有回應更糟糕。
- **故障隔離**— 故障隔離會在發生故障時，將影響範圍限制為預期的故障容器。它可確保工作負載的特定元件一起失敗，同時防止故障串聯到其他非預期的元件。它還有助於限制對工作負載客戶的影響範圍。故障隔離與前四個屬性有些不同，因為它接受已經發生故障但應該包含。您可以在基礎結構、相依性和軟體功能中建立故障隔離。

違反所需的屬性時，可能會導致工作負載無法使用或被認為無法使用。基於這些所需的彈性特性以及我們與許多人合作的經驗AWS客戶，我們已經確定了五個常見的故障類別：單點故障，過度負載，過度延遲，錯誤配置和錯誤，以及共享的命運，我們將其縮寫為 LIKE。這些提供了一致的方法來分類潛在的失敗模式，並在下表中說明。

失敗類別	違反	定義
單點故障 (SPOF)	備援	單一元件的故障會因為元件缺乏備援而中斷系統。
負載過大	足夠容量	過度需求或流量過度消耗資源會導致資源無法執行預期的功能。這可能包括達到限制和配額，這會導致限制和拒絕請求。
過度延遲	及時輸出	系統處理或網路流量延遲超過預期時間、服務層級目標 (SLO) 或服務層級協定 (SLA)。

配置錯誤和錯誤

正確的輸出

軟件錯誤或系統配置錯誤導致輸出不正確。

共同命運

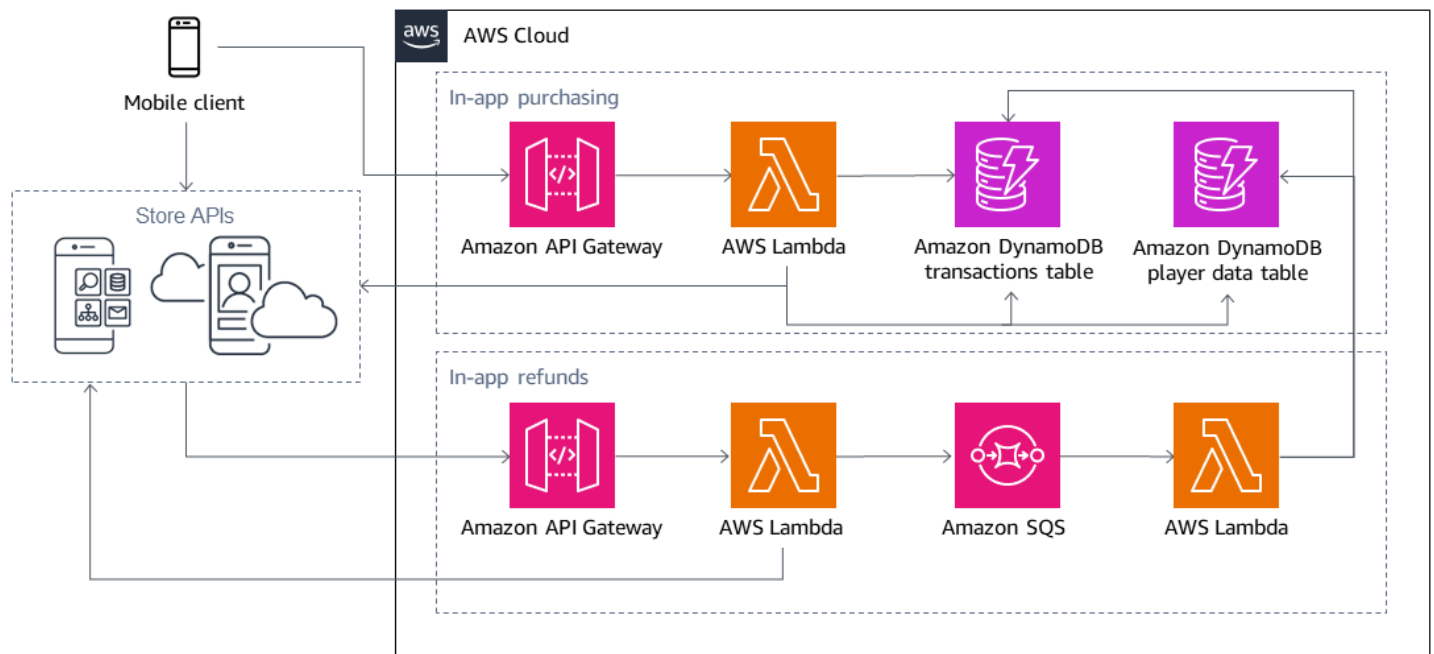
故障隔離

先前任何失敗類別所造成的錯誤會跨越預期的故障隔離界限，並串聯至系統的其他部分或其他客戶。



## 了解工作負載

若要套用架構，請先了解您要分析的工作負載。系統架構圖為記錄系統最相關的細節提供了一個起點。但是，嘗試分析整個工作負載可能很複雜，因為許多系統都有許多元件和互動。相反，我們建議您專注於用戶故事，這是從最終使用者的角度撰寫的軟體功能的非正式、一般說明。他們的目的是說明軟件功能如何為客戶提供價值。然後，您可以使用架構圖和資料流程圖來建立這些使用者故事的模型，以便更輕鬆地評估提供上述業務功能的技術元件。例如，應用內移動遊戲購買解決方案可能有兩個用戶故事：「購買應用內點數」和「獲得應用內退款」，如下圖所示。（此示例架構突出顯示如何將系統分解為用戶故事；它不是為了表示高度彈性的應用程序。）



每個使用者故事都包含四個常見元件：程式碼和設定、基礎架構、資料存放區和外部相依性。您的圖表應包括所有這些組件，並反映組件之間的相互作用。例如，如果 Amazon API 閘道端點上有過多的負載，請考慮該如何將串聯載入系統中的其他元件，例如 AWS Lambda 函數或亞馬遜動態 B 表。追蹤這些互動可協助您瞭解失敗模式如何影響使用者故事。您可以使用資料流程圖或在架構圖中使用簡單的流程箭頭，以視覺化方式擷取此流程，如上圖所示。對於每個元件，請考慮擷取詳細資料，例如正在傳輸的資訊類型、接收的資訊、通訊是同步還是非同步，以及跨越哪些故障界限。在此範例中，DynamoDB 表會在兩個使用者故事中共用，如箭頭所示，指出應用程式內退款內文中的 Lambda 元件可存取應用程式內購買故事中的 DynamoDB 表格。這意味著由於應用內購買用戶故事導致的失敗可能會由於共享命運而級聯到應用內退款用戶故事中。

此外，瞭解每個元件的基準組態也很重要。基準配置可識別條件約束，例如每秒交易的平均和最大數量、有效負載的大小上限、用戶端逾時，以及資源的預設或目前服務配額。如果您要建立新設計的模型，建議您記錄設計的功能需求並考慮限制。這可協助您瞭解失敗模式如何在元件中顯示出來。

最後，您應該根據用戶故事提供的商業價值來確定用戶故事的優先級。此優先順序可協助您首先專注於工作負載最重要的功能。然後，您可以將分析集中在作為該功能關鍵路徑一部分的工作負載元件上，並更快地利用架構來實現價值。當您遍歷該過程時，您可以按照不同的優先級檢查其他用戶故事。

# 應用框架

應用彈性分析框架的最佳方法是從一組按故障類別組織的標準問題開始，您應該詢問您正在分析的用戶故事中的每個組件。如果某些問題不適用於工作負載中的每個元件，請使用最適用的問題。

您可以從兩個角度思考失敗模式：

- 故障如何影響組件的支持用戶故事的能力？
- 故障如何影響組件與其他組件的交互？

例如，當您考慮資料存放區和過度負載時，您可能會考慮資料庫負載過多且查詢逾時的失敗模式。您也可以考慮資料庫用戶端如何使用重試或無法關閉資料庫連線來壓倒資料庫，從而耗盡連線集區。另一個範例是驗證程序，可能包含數個步驟。您需要思考多重要素驗證 (MFA) 應用程式或第三方身分識別提供者 (IdP) 的失敗會如何影響此驗證系統中的使用者故事。

當您回答下列問題時，您應該考慮失敗的來源。例如，過載是由客戶激增造成的，還是由維護活動期間節點停止服務的人工操作員造成的？您可能能夠識別每個問題中的多個失敗來源，這可能需要不同的緩和措施。當您提出問題時，請記錄您發現的潛在失敗模式、它們適用於哪些元件，以及每個失敗的來源。

## 單點故障

- 元件是否為備援架構而設計？
- 如果元件失敗，會發生什麼情況？
- 您的應用程式能否容忍單一可用區域的部分或全部損失？

## 過度延遲

- 如果此元件遇到延遲增加，或與其互動的元件延遲增加 (或網路中斷，例如 TCP 重設)，會發生什麼情況？
- 您是否已使用重試策略適當地設定逾時？
- 你失敗快還是慢？是否存在級聯效果，例如無意中將所有流量發送到受損的資源，因為它快速失敗？
- 對此組件發出的最昂貴的請求是什麼？

## 負載過大

- 什麼可以壓倒這個組件？這個組件如何壓倒其他組件？
- 如何防止將資源浪費在永遠不會成功的工作上？
- 您是否有針對元件設定的斷路器？
- 某些東西可以創建一個不可逾越的積壓嗎？
- 這個組件在哪裡可以體驗雙模態行為？
- 可以超過哪些限制或服務配額（包括儲存容量）？
- 零組件在負載下如何縮放？

### 配置錯誤和錯誤

- 如何防止錯誤配置和錯誤被部署到生產環境？
- 您是否可以自動回復錯誤的部署，或將流量從部署更新或變更的容器移開？
- 你有什麼護欄來防止操作員錯誤？
- 哪些項目（例如憑證或憑證）可以過期？

### 共同命運

- 您的故障隔離界限是什麼？
- 對部署單位所做的變更是否至少與您的預期一樣小 [故障隔離邊界](#) 但理想情況下更小，例如單機環境（故障隔離邊界內的單個實例）？
- 此元件是否在使用者故事或其他工作負載之間共用？
- 還有哪些其他組件緊密耦合到這個組件？
- 如果此元件或其相依性發生部分或灰色故障，會發生什麼情況？

提出這些問題之後，您也可以使用 YELET 來開發其他特定於您的工作負載和每個元件的問題。當您執行彈性分析時，WILLE 最適合用作思考失敗模式的結構化方式，也是靈感來源。這不是一個剛性的分類。不要花時間擔心特定失敗模式適合哪個類別-這並不重要。什麼是重要的是，你想到了失敗，並把它寫下來。沒有錯誤的答案；在盒子外面有創造力和思考是有益的。此外，不要假設已經緩解了故障模式；包括您可以想到的所有潛在故障模式。

在第一個練習中，您不太可能預測所有潛在的失敗模式。框架的多次迭代可幫助您生成更完整的模型，因此您不必嘗試解決第一遍的所有內容。您可以定期、每週或每兩週的節奏執行分析。在每個工作階段中，專注於特定的失敗模式或元件。這有助於在提高工作負載的彈性方面取得穩定、漸進的進展。收集用戶故事的潛在故障模式列表之後，您可以決定如何處理它們。

# 減輕潛在故障

既然使用者故事中的元件可能發生故障，您可以專注於緩和措施。首先，查看與您發現的每個失敗的潛在影響和可能性有關的潛在權衡。然後決定所需的觀察性層級，並選取緩和策略。權衡應包括努力檢測正確水平的觀察性和緩解策略。最後，確定定期進行彈性分析審查的正確節奏。

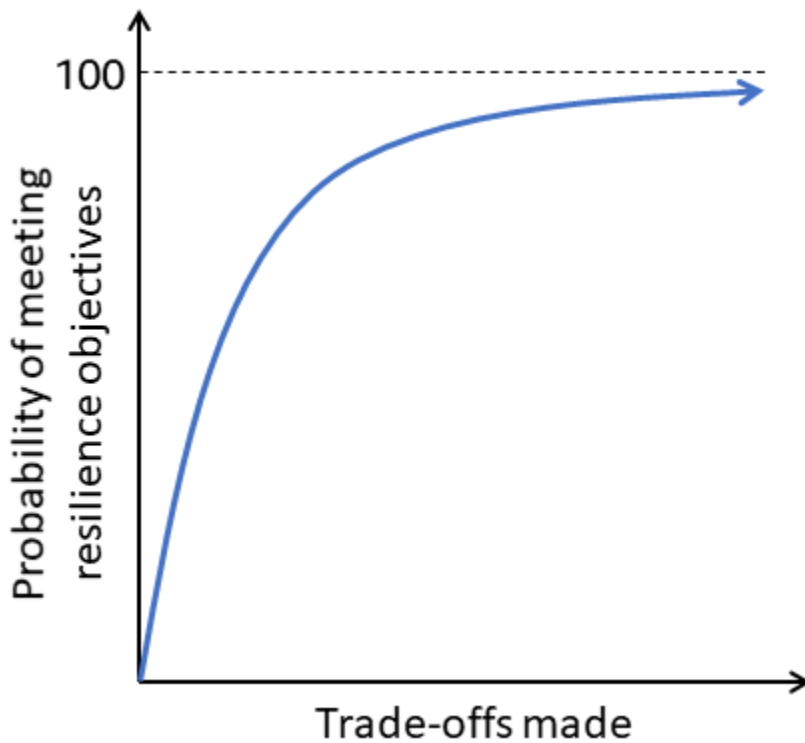
## 章節

- [了解權衡和風險](#)
- [故障模式可觀測性](#)
- [常見的緩解策略](#)
- [持續改善](#)

## 了解權衡和風險

彈性架構應該使用少數經過良好測試、簡單且可靠的機制來回應故障。為了達到最高等級的彈性，工作負載應該會自動偵測並從盡可能多的故障模式中復原。這樣做需要在執行彈性分析方面進行大量投資。這意味著實現更高的彈性水平涉及進行權衡。但是，隨著您繼續進行權衡，您會達到相對於彈性目標的回報減少的點。以下是最典型的權衡：

- 成本 — 冗餘組件，增強的可觀察性，額外的工具或增加的資源使用率將導致成本增加。
- 系統複雜性 — 偵測並回應故障模式 (包括緩解解決方案)，以及可能未使用受管理服務，可能會增加系統複雜性。
- 工程工作 — 需要額外的開發人員時間來建置解決方案，以偵測並回應故障模式。
- 營運負荷 — 監控和操作處理更多故障模式的系統可能會增加營運額外負荷，尤其是當您無法使用託管服務來緩解特定故障模式時。
- 延遲和一致性 — [建立有利於可用性的分散式系統需要在一致性和延遲方面取捨，如 PACELC 定理所述。](#)



當您考慮使用者故事中已識別失敗模式的緩和措施時，請考慮您需要進行的權衡。與安全性一樣，彈性是一個優化問題。您必須決定是否要避免，減輕，轉移或接受由識別故障帶來的風險。您可以避免某些失敗模式，您可以接受的一些失敗模式，以及一些可以轉移的失敗模式。您可以選擇緩解您識別的許多失敗模式。要確定採取哪種方法，請提出兩個問題來執行評估：發生失敗的可能性是多少？如果工作負載確實發生了什麼影響？

可能性是事件將發生多麼合理。例如，如果使用者故事具有在單一 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上運作的元件，則該元件可能會在系統作業期間的某個時間點中斷，可能是因為修補程序或作業系統錯誤所致。或者，由 Amazon 關聯式資料庫服務 (Amazon RDS) 管理的資料庫，該資料庫可同步其主要和次要執行個體之間的資料，在完全無法使用時具有較低的合理性。

影響是對事件可能造成的傷害的估計。它應該從財務和聲譽的角度進行評估，並且是相對於它影響的用戶故事的價值。例如，不堪重負的數據庫可能會對電子商務系統接受新訂單的能力產生重大影響。不過，在負載平衡器後方 20 個執行個體的叢集中遺失單一執行個體，可能會造成很小的影響。

您可以將這些問題的答案與降低風險所需的權衡成本進行比較。當您考慮到風險閾值和彈性目標而考慮此信息時，它會通知您計劃積極緩解哪些故障模式的決定。

## 故障模式可觀測性

若要緩解失敗模式，您必須先偵測它目前正在影響或即將影響您的工作負載。只有在有信號表明必須採取行動時，緩解措施才有效。這表示建立任何緩和措施的一部分至少包括驗證您具有或正在建立偵測失敗影響所需的觀察性。

您應該在兩個維度中考慮故障模式的可觀察到的症狀：

- 哪些主要指標可以通知您系統正在接近可能會很快看到影響的狀況？
- 哪些滯後指標可以在發生故障模式後盡快顯示出故障模式的影響？

例如，套用至資料庫元素的過度載入失敗可能會有連線計數做為前導指標。您可以看到連線計數的穩定增加是資料庫很快就會超出連線限制的前導指標，因此您可以採取動作，例如終止最近使用最少的連線，以減少連線計數。滯後指示器會指出何時超過資料庫連線限制，而且資料庫連線錯誤會提高。除了收集應用程式和基礎架構指標之外，還可以考慮收集[關鍵效能指標 \(KPI\)](#)，以偵測故障何時會影響您的客戶體驗。

如果可能的話，我們建議您在觀察性策略中包含這兩種類型的指標。在某些情況下，您可能無法建立領先指標，但是您應該始終計劃針對您要緩解的每個失敗都有一個滯後指標。要選擇正確的緩解措施，您還應該考慮前導或滯後指示器是否檢測到故障。例如，考慮到您網站的流量突然飆升。您可能只會看到一個滯後指標。在這種情況下，單獨自動擴展可能不是最好的緩解措施，因為部署新資源需要一些時間，而節流可以幾乎立即防止過載，並讓您的應用程式有時間來擴展或減少負載。相反，為了逐步增加流量，您會看到一個領先指標。在這種情況下，節流不適合，因為您有時間通過自動擴展系統來響應。

## 常見的緩解策略

首先，請考慮使用預防性緩和措施來防止失敗模式影響使用者故事。然後，您應該考慮糾正緩解措施。糾正緩解措施可幫助系統自我修復或適應不斷變化的條件。以下是符合復原特性之每個故障類別的常見緩解措施清單。

失敗類別	所需的彈性特性	緩和措施
單點故障 (SPOF)	備援和容錯	<ul style="list-style-type: none"><li>• 實作<a href="#">備援</a> — 例如，在 Elastic Load Balancing (ELB) 後面使用多個 EC2 執行個體。</li></ul>

- 移除 [AWS 全域服務控制平面](#) 的相依性，並且只取得 [全域服務資料平面](#) 的相依性。
- 當資源無法使用時，請使用 [正常降級](#)，以便您的系統靜態穩定到單點故障。

## 負載過大

## 足夠容量

- [關鍵緩解策略包括速率限制，負載脫落和工作優先級，持續工作，指數輪詢和重試抖動或根本不重試，使較小的服務受到控制，管理隊列深度，自動擴展，避免冷緩存和斷路器。](#)
- 您還應該考慮您的容量計劃，並考慮您可能會遇到的 AWS 資源和系統內限制的 future 容量和擴展限制。

## 過度延遲

## 及時輸出

- 實施適當配置的 [超時](#) 或調適性超時（根據當前和預測的延遲條件更改超時值，以便可能允許緩慢的依賴關係進行進展，而不是放棄緩慢的請求）。
- [從內部部署環境連線到雲端服務，並在特定路由上遇到延遲時，使用多重路徑 TCP 等技術來實作指數輪詢和重試，使用抖動、套期保值等技術，並且不丟棄工作。](#)



## 配置錯誤和錯誤

## 正確的輸出

- catch 軟件中可重複的功能錯誤的主要方法是通過諸如靜態分析，單元測試，集成測試，回歸測試，負載測試和彈性測試等機制進行嚴格測試。
- 實作基礎架構即程式碼 (IaC)、持續整合與持續交付 (CI/CD) 自動化等策略，以協助減輕設定錯誤的威脅。
- 使用部署技術，例如單機式、初期測試部署、符合故障隔離界限的部分部署，或藍/綠部署來減少錯誤設定和錯誤。

## 共同命運

## 故障隔離

- 在系統中實作容錯能力，並使用邏輯和實體容錯隔離界限，例如多個運算或容器叢集、多個 AWS 帳戶、多個 AWS Identity and Access Management (IAM) 主體、多個可用區域，以及多個 AWS 區域可用區域。
- [基於細胞的體系結構](#)和[洗牌分片等技術也可以改善故障隔離](#)。
- 考慮[鬆耦合](#)和[優雅降級](#)等模式，以防止串聯失敗。當您優先考慮用戶故事時，您還可以使用該優先級來區分對於主要業務功能至關重要的用戶故事以及可以優雅降級的用戶故事。例如，在電子商務網站中，您不希望網站上的促銷小部件受到影響處理新訂單的能力。

雖然其中一些緩和措施需要很少的工作來實施，但其他（例如採用基於細胞的架構來實現可預測的故障隔離和最小的共享命運失敗）可能需要重新設計整個工作負載，而不僅僅是特定用戶故事的組件。如前所述，重要的是衡量失敗模式的可能性和影響與您為減輕它所做的權衡。

除了適用於每個失敗模式類別的緩和技術之外，您還應該考慮復原使用者故事或整個系統所需的緩和措施。例如，失敗可能會停止工作流程，並防止將資料寫入預定目的地。在這種情況下，您可能需要操作工具來重新驅動工作流程或手動修復資料。您可能還必須在工作負載中建立檢查點機制，以協助防止發生故障時遺失資料。或者，您可能必須構建一條 andon 電源線來暫停工作流程並停止接受新工作以防止進一步的傷害。在這些情況下，您應該考慮所需的操作工具和護欄。

最後，您應該始終假設人類在制定緩解策略時會犯錯誤。儘管現代 DevOps 實踐力求自動化操作，但人類仍然必須出於各種原因與您的工作負載進行交互。不正確的人為操作可能會導致任何 SIREAM 類別中的失敗，例如在維護期間移除太多節點並導致過載，或者錯誤地設置功能標誌。這些情況實際上是預防性護欄中的失敗。根本原因分析不應該以「人類犯了錯誤」的結論結束。相反，它應該首先解決錯

誤可能出現的原因。因此，您的緩解策略應考慮人類操作員如何與工作負載元件互動，以及如何透過安全護欄來防止或將人為操作員錯誤所造成的影響降到最低。

## 持續改善

韌性是一個[連續的過程](#)。在系統的生命週期中，它運作的環境將會發生變化。為了確保您的系統保持彈性，您應該將架構整合到定期的操作和架構檢閱中。您可能會發現第一次沒有識別出來的新失敗模式，或者可能有新的或之前沒有想到的緩和措施，您可以實施。彈性分析應該是一個迭代過程，而不是一次性的練習。

您應該使用[混亂工程或遊戲日等程序](#)，以實證方式測試您的緩解策略，以驗證它們是否如預期般運作。如果您沒有嚴格的測試機制，您將無法確信緩解措施會在您需要時如預期般運作。在彈性分析期間，您可能會判斷失敗模式已由特定緩和措施處理，但測試這些假設也很重要。您應該測試使用彈性分析架構建立的現有緩和措施和新的緩和措施。

您還應該通過團隊回顧來評估您執行分析的效果。每個人都知道他們在分析過程中的工作嗎？您通過彈性分析找到的故障模式數量是否符合團隊的期望？您是否可以識別您發現的所有失敗模式的緩和措施？團隊認為這個過程有用嗎？您是否相信這會導致工作負載的彈性提高？

當發生影響工作負載可用性的實際故障事件時，請記錄特定的失敗模式、屬於失敗一部分的元件，以及使用的緩和模式。您可以在事件後分析工具中搜尋此中繼資料，以便您決定 future 要關注哪些失敗模式和元件。在整個過程中，您可以與您的客AWS戶團隊和解決方案架構師互動。

## 結論和資源

本指南提供了以持續且一致的方式執行彈性分析的框架。此架構可協助您識別單點故障、過度負載、過度延遲、錯誤設定和錯誤，以及共用命運可能會對工作負載元件造成何種影響。識別這些失敗模式可協助您決定適當的緩解策略，作為建置復原導向架構的一部分。

有關彈性分析的其他閱讀，請參閱以下鏈接：

- [彈性生命週期框架](#) ( AWS規定指導 )
- [復原力解決方AWS案](#) (解決方案庫)
- [邁向持續的韌性](#) ( 阿德里安·霍恩斯比，雲端架構師，2021年3月24日 )

# 文件歷史紀錄

下表說明本指南的重大變更。如果您想收到有關未來更新的通知，您可以訂閱[RSS 摘要](#)。

變更	描述	日期
<a href="#">初始出版</a>	—	2023年9月5 日

# AWS 規範指引詞彙表

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 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 中的 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 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。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，其中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 (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。如需如何在遷移策略中使用 AWS 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、一致性、隔離、持久性 (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 ([ABAC](#)) 文件中的 [Word for AWS](#)。IAM

## 權威性資料來源

您存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將資料從權威資料來源複製到其他位置，以處理或修改資料，例如匿名化、修訂或擬匿名化資料。

## 可用區域

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便宜的低延遲網路連線。

## 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。您可以搭配 Amazon Detective 使用行為圖表來檢查失敗的登入嘗試、可疑的 API 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[端點](#)。

### 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。有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。某些其他稱為不良機器人的機器人，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，稱為機器人繼承者或機器人運算子。Botnet 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，以測試系統的彈性。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 在部落格文章 [The Journey Toward Cloud-First](#) 和企業策略部落格上的 [採用階段](#) 中定義。AWS 雲端 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 SageMaker 則提供 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。您通常會在遷移的產品組合探索和分析階段使用來自 CMDB 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 帳戶和區域中或跨組織的單一實體。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 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](#)。

## 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、分散式的資料擁有權，並具有集中式的管理和治理。

## 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，可協助確保只有受信任身分才能從預期的網路存取受信任資源。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 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 AWS 上採用此策略時，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，a defense-in-depth 方法可能會結合多重重要素驗證、網路分割和加密。

### 委派的管理員

在 AWS Organizations 中，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)

用於識別限制並排定優先順序的程序，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 延伸了最初為精實生產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

真實世界系統的虛擬呈現，例如建築、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 [中的上的工作負載災難復原](#) [AWS：雲端中的復原](#)。

## 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 ASMX Gateway 逐步現代化舊版 Microsoft ASP.NET \(API\) 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 來偵測系統資源的偏離，也可以使用 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](#)。

### 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縮短回應時間。



## 電子資料交換 (EDI)

組織之間的商業文件自動交換。如需詳細資訊，請參閱[什麼是電子資料交換](#)。

## 加密

將純文字資料轉換為人類可讀取的運算程序。

## 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 VPC 端點，私下連線至您的端點服務。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)

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 是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欄。

## 快速失敗

使用頻繁且增量測試來縮短開發生命週期的哲學。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。

## 幾下提示

在請求 [LLM](#) 執行類似任務之前，提供少數示範任務和所需輸出的範例。此技術是內容內學習的應用，其中模型會從內嵌在提示中的範例 (快照) 中學習。對於需要特定格式設定、推理或網域知識的任務，少量擷取提示非常有效。另請參閱[零擷取提示](#)。

## 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，以及以自然語言進行交談。如需詳細資訊，請參閱[什麼是基礎模型](#)。

## G

### 生成式 AI

經過大量資料訓練的 [AI](#) 模型子集，可以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)

在 Amazon CloudFront 中，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 工作流程會被視為舊版，而以[中繼線為基礎的工作流程](#)是現代的首選方法。

### 金色影像

系統或軟體的快照，用作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

高階規則，可協助管理跨組織單位 (OUs) 的資源、政策和合規性。預防性防護機制會強制執行政策，以確保符合合規標準。其實作方式是使用服務控制政策和 IAM 許可界限。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。其實作方式包括使用 AWS Config AWS Security Hub、Amazon GuardDuty、Amazon Inspector AWS Trusted Advisor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。歷史資料是一種資料庫，用於從工廠的不同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歷程記錄、已標記的資料的一部分。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。

## 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 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的 DevOps 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

## IaC

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，可定義其在 AWS 雲端環境中的許可。

## 閒置應用程式

平均 CPU 和記憶體用量介於 5% 到 20% 之間的應用程式，為期 90 天。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，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的部署](#)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](#)建議設定具有傳入、傳出和檢查 VPCs 的網路帳戶，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

由 [Klaus Schwab](#) 於 2016 年引進的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進步，指製造流程的現代化。

## 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，集中式 VPC 可管理 VPCs (在相同或不同的 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](#)建議設定具有傳入、傳出和檢查 VPCs 的網路帳戶，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[AWS 的機器學習模型可解釋性](#)。

## 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 ITSM 的基礎。

## 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，其中每個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、彙整文件、將文字翻譯為其他語言，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 Rs](#)。

## 小端序系統

首先儲存最低有效位元組的系統。另請參閱 [端點](#)。

## 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[遷移加速計畫](#)。

## 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。機制是可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

除了屬於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

小型的獨立服務，透過定義明確的 APIs 進行通訊，通常由小型的獨立團隊擁有。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 APIs 透過定義明確的界面進行通訊。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)

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：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

## 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）。[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 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 A machine-to-machine (M2M) 通訊協定。OPC-UA 提供具有資料加密、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 群組承諾交付給彼此的協議，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[操作就緒審核 \(ORR\)](#)。

## 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 [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 AWS CloudTrail 的追蹤會記錄 AWS 帳戶組織中所有的事件 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 CloudTrail 文件中的[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至新系統和策略。在 AWS 遷移策略中，由於雲端採用專案所需的變更速度，因此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)

In CloudFront 是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 S3 儲存貯體 AWS 區域中的所有伺服器端加密 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)

In CloudFront，用於限制存取以保護您的 Amazon S3 內容的選項。當您使用 OAI 時，CloudFront 會建立 Amazon S3 可以驗證的主體。已驗證的主體只能透過特定 CloudFront 分佈存取 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[操作準備檢閱](#)。

## OT

請參閱[操作技術](#)。

## 傳出（傳出）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 Security Reference Architecture](#) 建議使用傳入、傳出和檢查 VPCs 設定您的 Network 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

連接至 IAM 主體的 IAM 管理政策，用於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](#)。

### 個人身分識別資訊 (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。PII 的範例包括名稱、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[可程式設計邏輯控制器](#)。

### 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最大許可的物件 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。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

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

容器，其中包含有關您希望 Amazon Route 53 如何回應一個或多個 DNS 內網域及其子網域的 VPCs 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](#)，並參閱在實作安全[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。

### 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。此技術用於將複雜的任務分解為子任務，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關聯性，並允許更精細、更個人化的結果。

### 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，可啟用微服務之間的非同步通訊，以提高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、諮詢、知情 \(RACI\)](#)。

## RAG

請參閱[擷取增強型生成](#)。

## 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、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 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 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。[在中規劃彈性時，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 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、負責、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：責任 (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](#)。

## 擷取增強產生 (RAG)

在產生回應之前，[LLM](#) 參考其訓練資料來源外之權威資料來源的[生成性 AI](#) 技術。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 由資料列許可和資料欄遮罩組成。

## 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)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而不必在 IAM 中為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[SAML 文件中的關於 Word 2.0 型聯合](#)。IAM

## SCADA

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？](#)。

## 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 會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 作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

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。

## 服務層級指示器 (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)

目標指標，代表服務的運作狀態，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 ASP Gateway 逐步現代化舊版 Microsoft ASMX.NET \(API\) Web 服務](#)。

## 子網

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

提供內容、指示或指引給 [LLM](#) 以指示其行為的技術。系統提示可協助設定內容，並建立與使用者互動的規則。

# 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 VPCs 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 Word 對等](#)。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。查詢此類資料時，通常可接受中等慢的查詢。

## 視窗函數

SQL 函數，對以某種方式與目前記錄相關聯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 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](#)。

# 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。威脅實施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。

## 零擷取提示

為 [LLM](#) 提供執行任務的指示，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。零擷取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 [微拍提示](#)。

## 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。