



在 AWS 上為醫療保健建立擷取增強生成解決方案

# AWS 方案指引



# AWS 方案指引: 在 AWS 上為醫療保健建立擷取增強生成解決方案

Copyright © 2025 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
病患照護和生產力 .....	1
人才管理 .....	1
機會和挑戰 .....	3
醫療保健中生成式 AI 應用程式的機會 .....	3
進階映像分析 .....	3
將解決方案工業化的挑戰 .....	3
使用案例：建置醫療智慧應用程式 .....	5
解決方案概觀 .....	5
步驟 1：探索資料 .....	7
步驟 2：建立醫學知識圖表 .....	7
步驟 3：建置內容擷取代理程式 .....	12
Amazon Bedrock 代理程式 .....	13
LangChain 代理程式 .....	14
步驟 4：建立知識庫 .....	15
使用 OpenSearch Service .....	15
建立 RAG 架構 .....	16
步驟 5：產生回應 .....	19
符合 AWS Well-Architected 架構 .....	20
使用案例：預測重新入學率 .....	21
解決方案概觀 .....	21
步驟 1：預測患者結果 .....	23
步驟 2：預測患者行為 .....	24
步驟 3：預測患者重新入院 .....	26
步驟 4：計算傾向分數 .....	28
符合 AWS Well-Architected 架構 .....	31
使用案例：管理人才 .....	32
解決方案概觀 .....	32
步驟 1：建立技能描述檔 .....	34
步驟 2：探索role-to-skill的相關性 .....	34
步驟 3：建議訓練 .....	36
符合 AWS Well-Architected 架構 .....	36
開發解決方案 .....	38
Amazon Q Developer .....	38

多擷取器 RAG 設計 .....	38
ReAct 代理程式 .....	40
評估解決方案 .....	42
評估資訊擷取 .....	42
評估多個擷取器 .....	42
使用 LLM .....	43
資源 .....	44
AWS 文件 .....	44
AWS 部落格文章 .....	44
其他資源 .....	44
貢獻者 .....	45
編寫 .....	45
檢閱 .....	45
技術寫入 .....	45
文件歷史紀錄 .....	46
詞彙表 .....	47
# .....	47
A .....	47
B .....	50
C .....	51
D .....	54
E .....	57
F .....	59
G .....	60
H .....	61
I .....	62
L .....	64
M .....	65
O .....	69
P .....	71
Q .....	73
R .....	73
S .....	76
T .....	79
U .....	80
V .....	81

---

W .....	81
Z .....	82
.....	lxxxiii

# 在上 AWS 為醫療保健建立擷取增強生成解決方案

Amazon Web ServicesAccenture、 和 Cadiem([貢獻者](#))

2025 年 3 月 ([文件歷史記錄](#))

在大型語言模型 (LLMs) 和生成式 AI 之前，在醫療保健產業中開發自動化和高精確度應用程式的任務是一項挑戰。傳統方法非常依賴手動資料輸入和分析。分析醫療影像和病患記錄的複雜性需要大量的人工介入，這通常會導致分散和效率低下的工作流程。AI 技術的進展可協助您大規模建置超個人化應用程式。醫療保健應用程式現在可以與醫療知識庫整合、以更高的準確性解譯診斷影像，以及使用預測模型預測患者結果。

本指南探討 LLMs 如何透過您可以使用的擷取增強生成應用程式來徹底改變醫療保健 AWS 服務。擷取增強生成 (RAG) 是一種生成式 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。RAG 應用程式以實際知識為基礎模型的輸出，可減少幻覺並提高回應相關性。在醫療保健領域中，RAG 可用來提供準確且 up-to-date 醫療資訊，確保醫療保健供應商可以存取最新的研究和臨床準則。透過將資料轉換為可行的洞見並自動化複雜的程序，這些技術有助於增強患者護理、簡化操作並改善醫療專業人員的生產力。

在 [Amazon Bedrock](#) 中，您可以微調 LLMs 並將其與智慧型代理程式整合，以建立進階醫療保健解決方案。本指南強調 [Amazon OpenSearch Service](#) 與 [Amazon Neptune](#) 之間的協同作用，示範這些服務如何透過增強的搜尋相關性和進階多來源資料擷取來提升 RAG 解決方案。您可以協調使用 Amazon Bedrock 代理程式的完整 Amazon Bedrock 解決方案 [LangChain](#)，並無縫協調不同資料儲存庫的互動。此整合示範結合專業服務，以建立更有效且高效率的 AI 驅動系統的能力。

## 病患照護和生產力

本指南提供兩種用於病患照護和生產力的實際使用案例：[病患資料擴增](#)和[預測重新入院風險](#)。它提供策略藍圖，以大規模實作這些解決方案，為醫療保健組織提供工業化 AI 驅動程序的明確途徑。透過這些洞察，醫療保健機構可以使用進階 AI 技術來建立更有效率、更智慧的工作流程。

## 人才管理

本指南還概述了重新技能化的策略，並使醫療保健工作者能夠將生成式 AI 無縫整合到其日常工作中。這可以提高生產力和患者護理品質。透過讓人力資源具備有效使用進階 AI 工具的技能，醫療保健組織可以最大限度地提高投資回報，並推動患者護理的創新。

此採用 AI 技術的[人才管理解決方案](#)包含下列主要功能：

- 智慧型人才恢復剖析器 – 透過使用 Amazon Bedrock 中可用的進階 LLMs，此工具可從恢復有效率地擷取和分析關鍵人才技能和屬性。此工具可以簡化招聘程序。
- 人才知識庫 – 採用 Amazon Neptune 技術，此動態資料庫可提供人員配置層級、技能分佈和產業趨勢的即時洞見。這可協助您做出有關人力資源管理的資料驅動型決策。
- 學習建議引擎 – 此 AI 驅動工具可識別組織內的技能差距，並為醫療工作人員建議個人化的訓練計畫。此工具可促進持續的專業開發，並協助您的員工適應不斷發展的醫療保健技術。

這些 AI 驅動功能共同協助最佳化人力資源效能，並以更高的智慧和效率徹底改變人才管理。

## 機會和挑戰

Amazon Bedrock 可以提供增強的生產力、可擴展性、成本效益和資料驅動的洞見。Amazon Bedrock 可讓醫療保健組織在各種使用案例中有效使用 LLMs，從內容建立和資料分析到自動化決策。本指南提供克服常見生成式 AI 挑戰的方法，例如資料品質問題、基礎設施可擴展性、模型效能的維護，以及從概念驗證轉換到生產期間的持續改進要求。

## 醫療保健中生成式 AI 應用程式的機會

醫療保健產業已準備好進行轉型轉移，這是由生成式 AI 應用程式提供的機會所驅動。生成式 AI 有可能增強患者護理、簡化操作並加速醫學研究。透過使用進階 AI 模型，醫療保健供應商可以自動化增強醫療記錄。完整且 up-to-date 患者歷史記錄有助於更準確的診斷和治療計畫。AI 驅動的影像分析，例如解譯聲紋和其他醫療影像，可以提供快速且精確的洞見、減少醫療專業人員的工作負載，並將人為錯誤的風險降至最低。

除了診斷和處理之外，生成式 AI 可以在預測分析中發揮關鍵作用。預測分析可協助醫療保健組織預測患者結果，並相應地個人化護理計畫。此技術也可以最佳化管理程序，從管理患者資料到簡化提供者和患者之間的通訊。透過將生成式 AI 解決方案與現有的醫療保健系統整合，醫療院所可以實現更高的效率、降低成本，最終提供更高品質的護理。AI 與醫療保健的整合不只是一種增強功能，而是邁向更智慧、回應能力和以患者為中心的護理的基礎轉變。

## 進階映像分析

將 Amazon Bedrock 與 Amazon Neptune 和 Amazon OpenSearch Service 等資料存放區結合，可協助您解決醫療保健領域中進階影像分析的複雜性。資訊擷取解決方案可以增強疾病探索程序，並透過評估診斷影像和解譯聲紋來增強解譯準確性。解決方案可以將視覺和文字評估資料與醫生的手動患者評估審核整合。

## 將解決方案工業化的挑戰

在醫療保健中工業化 AI 解決方案時要解決的主要障礙是資料品質和可用性。醫療保健資料通常以分段、不一致的格式存在。確保 AI 模型能夠存取乾淨、結構化和代表性的資料，對於在真實世界案例中維持效能至關重要。基礎設施可擴展性可能會成為挑戰，因為生產環境。這些環境需要處理大量的即時病患資料，同時提供快速的回應時間，並維持資料隱私權法規的合規性，例如健康保險流通與責任法案 (HIPAA)。此外，隨著不斷演進的醫療資訊和患者資料，AI 模型需要重新訓練和更新，以保持相關性並

提供準確的建議。最後，將這些 AI 解決方案整合到現有的醫療保健系統中可能很複雜，因為互通性問題以及需要與目前的臨床工作流程保持一致。此整合需要技術和操作變更。

## 使用案例：使用擴增的患者資料建置醫療智慧應用程式

生成式 AI 可以透過增強臨床和管理功能，協助提高患者護理和員工生產力。AI 驅動的影像分析，例如解譯聲紋，可加速診斷程序並提高準確性。它可以提供關鍵洞見，以支援及時的醫療介入。

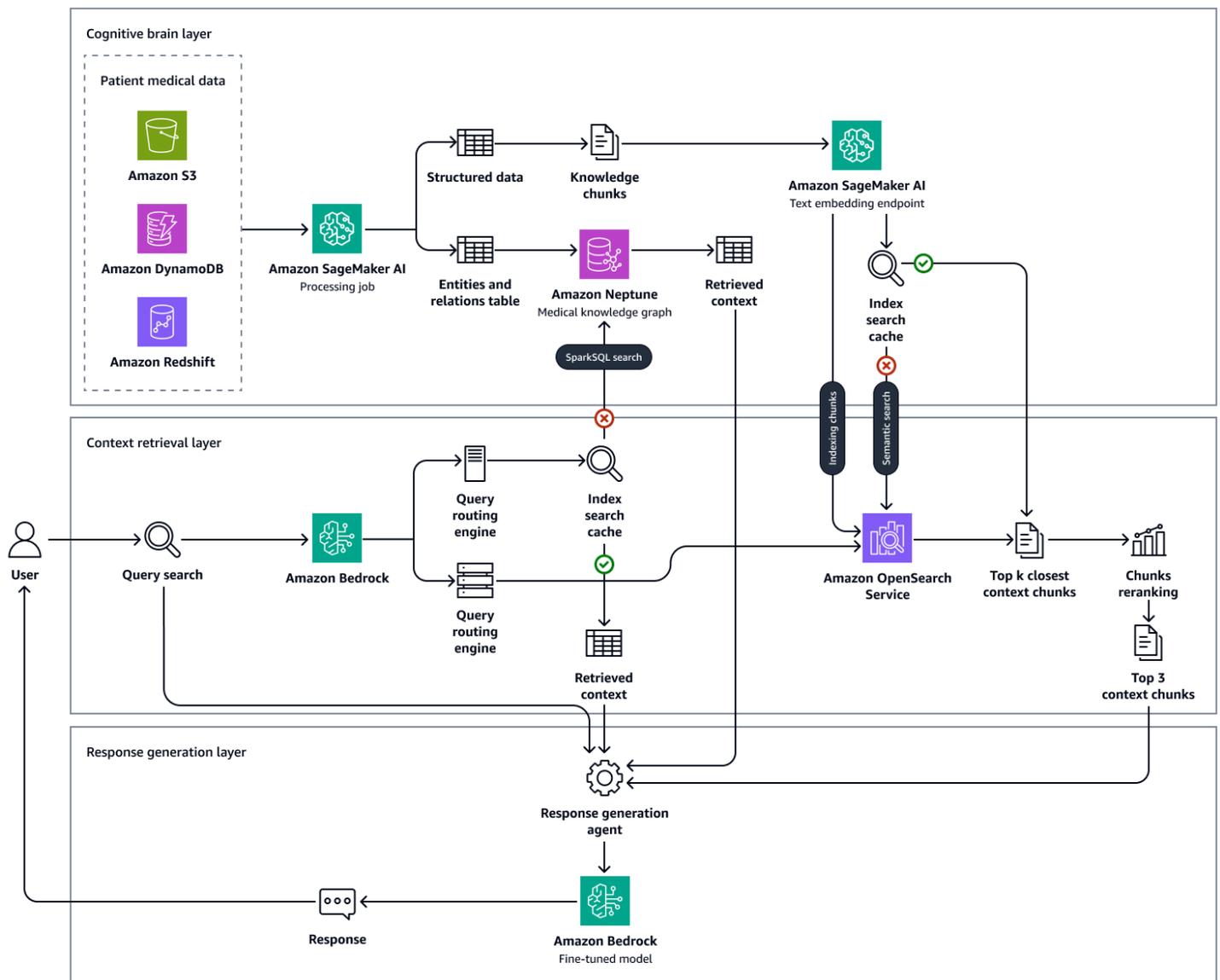
當您將生成式 AI 模型與知識圖表結合時，您可以自動化電子患者記錄的時間組織。這可協助您整合來自醫生與患者互動、症狀、診斷、實驗室結果和影像分析的即時資料。這會為醫生提供全面的患者資料。此資料可協助醫生做出更準確且及時的醫療決策，同時提高患者成果和醫療保健供應商的生產力。

### 解決方案概觀

AI 可以透過合成患者資料和醫學知識來為醫生和臨床醫生提供有價值的洞察。此擷取增強生成 (RAG) 解決方案是一種醫學智慧引擎，可從數百萬個臨床互動中取用一組全面的患者資料和知識。它利用生成式 AI 的力量來建立以證據為基礎的洞見，以改善患者護理。它旨在增強臨床工作流程、減少錯誤並改善患者結果。

解決方案包含由 LLMs 提供的自動化影像處理功能。此功能可減少醫務人員手動搜尋類似的診斷影像和分析診斷結果所花費的時間。

下圖顯示此解決方案的end-to-end-workflow。它使用 Amazon Neptune、Amazon SageMaker AI、Amazon OpenSearch Service 和 Amazon Bedrock 中的基礎模型。對於與 Neptune 中的醫學知識圖表互動的內容擷取代理程式，您可以在 Amazon Bedrock 代理程式和LangChain代理程式之間進行選擇。



在我們對範例醫療問題的實驗中，我們觀察到使用 Neptune、OpenSearch 向量資料庫存放臨床知識庫和 Amazon Bedrock LLMs 中維護的知識圖表的方法所產生的最終回應是基於事實，並透過減少誤報並提高真誤報來更加準確。此解決方案可以產生以證據為基礎的患者健康狀態洞察，並旨在增強臨床工作流程、減少錯誤並改善患者結果。

建置此解決方案包含下列步驟：

- [步驟 1：探索資料](#)
- [步驟 2：建立醫學知識圖表](#)
- [步驟 3：建置內容擷取代理程式來查詢醫學知識圖表](#)
- [步驟 4：建立即時描述性資料的知識庫](#)

## • [步驟 5：使用 LLMs 回答醫療問題](#)

# 步驟 1：探索資料

您可以使用許多開放原始碼醫療資料集來支援開發醫療保健 AI 驅動的解決方案。這類資料集之一是 [MIMIC-IV 資料集](#)，這是公有可用的電子運作狀態記錄 (EHR) 資料集，廣泛用於醫療保健研究社群。MIMIC-IV 包含詳細的臨床資訊，包括來自病患記錄的任意文字放電備註。您可以使用這些記錄來實驗文字總和和實體擷取技術。這些技術可協助您從非結構化文字中擷取醫療資訊（例如患者症狀、受管藥物和規定的處理方式）。

您也可以使用資料集，該資料集提供已標註、去識別化的患者出院摘要，這些摘要專門為研究目的而策劃。放電摘要資料集可協助您實驗實體擷取，讓您從文字中識別關鍵醫療實體（例如條件、程序和藥物）。[步驟 2：建立醫學知識圖表](#) 本指南說明如何使用從 MIMIC-IV 擷取的結構化資料和放電摘要資料集來建立醫學知識圖表。此醫療知識圖表是醫療保健專業人員進階查詢和決策支援系統的骨幹。

除了文字型資料集之外，您還可以使用影像資料集。例如，[Musculoskeletal Radiographs \(MURA\) 資料集](#)，這是骨骼多檢視放射影像的完整資料庫。使用這類影像資料集，透過醫療影像解碼技術來實驗診斷評估。這些解碼技術對於早期診斷疾病至關重要，例如肌肉骨骼疾病、重大疾病和骨質疏鬆。透過微調醫療影像資料集上的視覺和語言基礎模型，您可以偵測診斷影像中的異常。這有助於系統為臨床醫生提供早期且準確的診斷洞察。透過使用影像和文字資料集，您可以建立 AI 驅動的醫療保健應用程式，能夠同時處理文字和影像資料，以改善患者護理。

## 步驟 2：建立醫學知識圖表

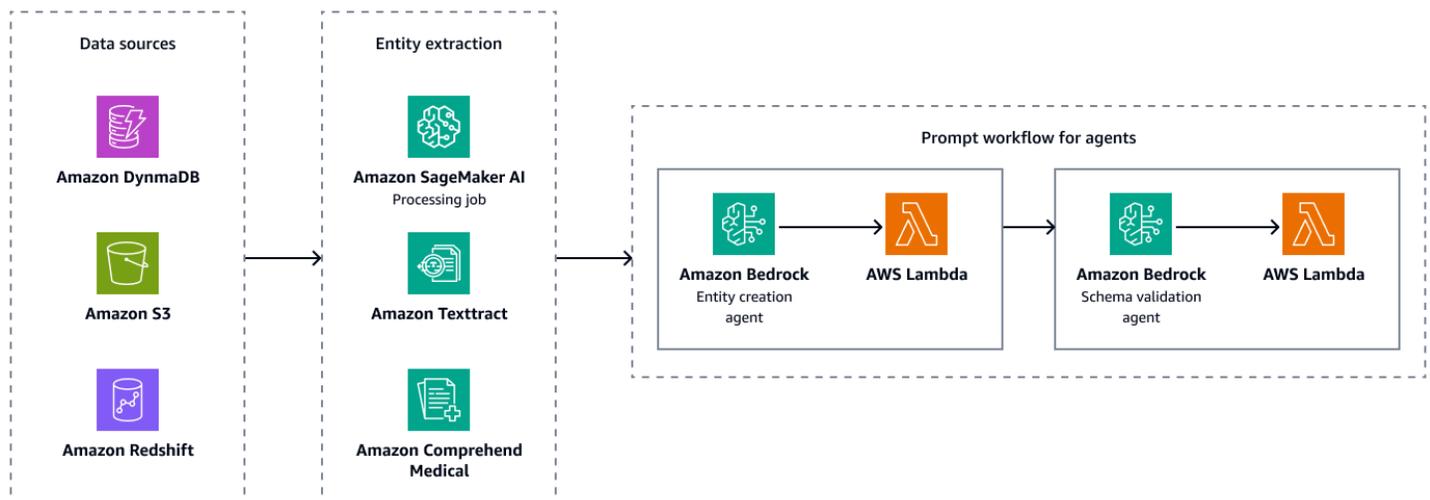
對於任何想要根據大量知識庫建置決策支援系統的醫療保健組織，關鍵挑戰是尋找和擷取臨床筆記、醫療日誌、放電摘要和其他資料來源中存在的醫療實體。您也需要從這些醫療記錄中擷取時間關係、主體和確定性評估，以有效地使用擷取的實體、屬性和關係。

第一步是使用幾個基礎模型的快照提示，例如 Amazon Bedrock 中的 Llama 3，從非結構化醫療文字中擷取醫療概念。少量的提示是當您向 LLM 提供少量範例，示範任務和所需輸出，然後再要求它執行類似的任務時。使用 LLM 型醫療實體擷取器，您可以剖析非結構化醫療文字，然後產生醫療知識實體的結構化資料表示。您也可以存放用於下游分析和自動化的患者屬性。實體擷取程序包含下列動作：

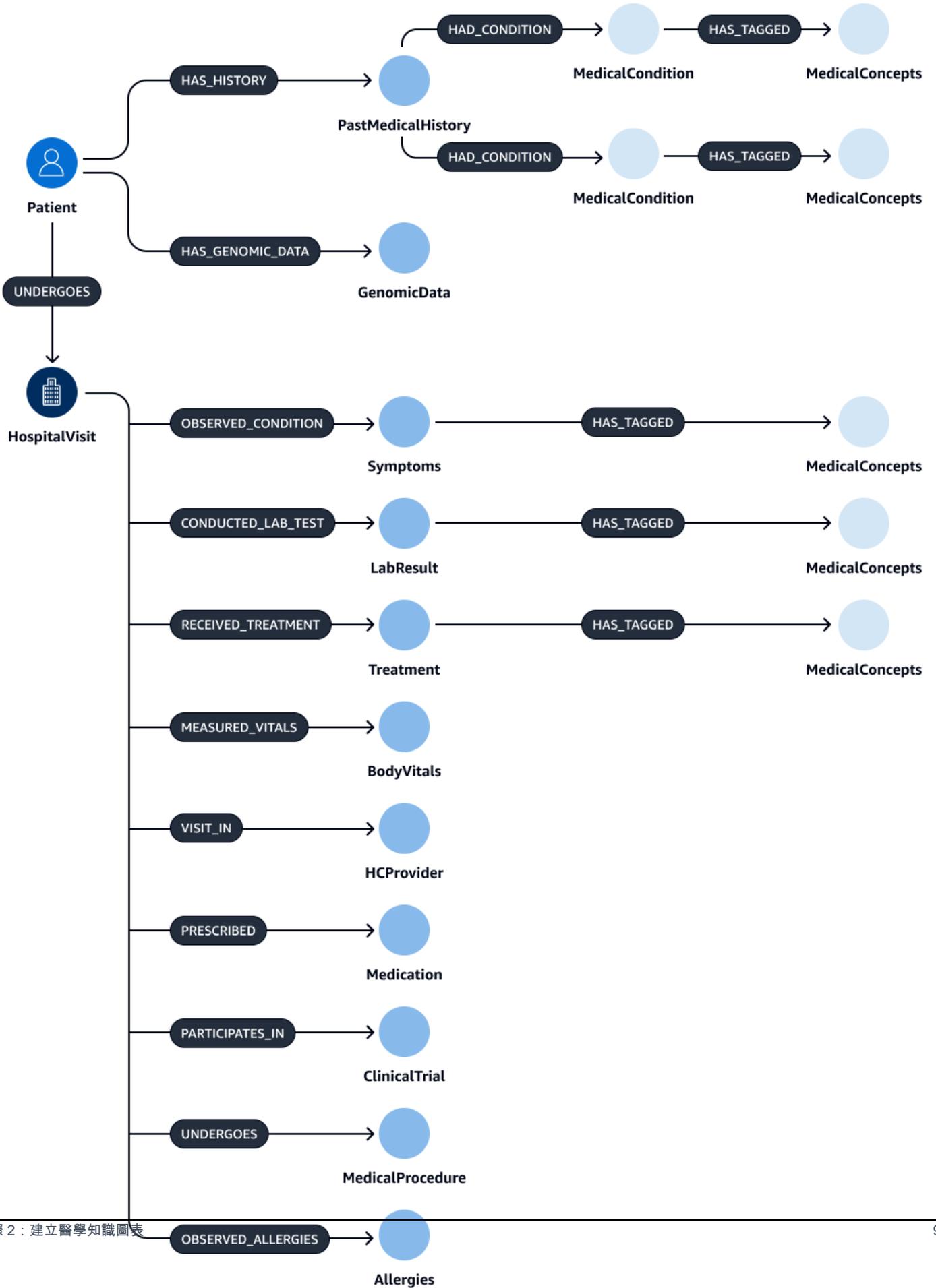
- 擷取有關醫療概念的資訊，例如疾病、藥物、醫療設備、用量、藥物頻率、藥物持續時間、症狀、醫療程序及其臨床相關屬性。
- 擷取功能特徵，例如擷取的實體、主體和確定性評估之間的時間關係。
- 展開標準醫療詞彙，例如：

- 來自 RxCUI) [RxNorm](#)
- [國際疾病分類、第 10 次修訂、臨床修改 \(ICD-10-CM\) 的程式碼](#)
- [Medical Subject Headings \(MeSH\) 的條款](#)
- [系統化醫學命名法臨床術語 \(SNOMED CT\) 的概念](#)
- 來自[統一醫療語言系統 \(UMLS\)](#) 的程式碼
- 摘要放電備註，並從文字記錄衍生醫學洞見。

下圖顯示實體擷取和結構描述驗證步驟，以建立實體、屬性和關係的有效配對組合。您可以在 Amazon Simple Storage Service (Amazon S3) 中存放非結構化資料，例如放電摘要或病患備註。您可以在 Amazon Redshift 和 Amazon DynamoDB 中存放結構化資料，例如企業資源規劃 (ERP) 資料、電子患者記錄和實驗室資訊系統。您可以建置 Amazon Bedrock 實體建立代理程式。此代理程式可以整合 Amazon SageMaker AI 資料擷取管道、Amazon Textract 和 Amazon Comprehend Medical 等服務，從結構化和非結構化資料來源中擷取實體、關係和屬性。最後，您可以使用 Amazon Bedrock 結構描述驗證代理程式來確保擷取的實體和關係符合預先定義的圖形結構描述，並維護節點邊緣連線和相關聯屬性的完整性。



在擷取和驗證實體、關係和屬性之後，您可以連結它們來建立subject-object-predicate三元組。您可以將此資料擷取至 Amazon Neptune 圖形資料庫，如下圖所示。[圖形資料庫](#)經過最佳化，可存放和查詢資料項目之間的關係。



您可以使用此資料建立全面的知識圖表。[知識圖表](#)可協助您組織和查詢各種連線資訊。例如，您可以建立具有下列主要節點的知識圖表：HospitalVisit、PastMedicalHistory、Symptoms、Medication、MedicalProcedures和 Treatment。

下表列出您可能從放電備註中擷取的實體及其屬性。

實體	Attributes
Patient	PatientID , Name, Age, Gender, Address, ContactInformation
HospitalVisit	VisitDate , Reason, Notes
HealthcareProvider	ProviderID , Name, Specialty , ContactInformation , Address, AffiliatedInstitution
Symptoms	Description , RiskFactors
Allergies	AllergyType , Duration
Medication	MedicationID , Name, Description , Dosage, SideEffects , Manufacturer
PastMedicalHistory	ContinuingMedicines
MedicalCondition	ConditionName , Severity, Treatment Received , DoctorinCharge , HospitalName , MedicinesFollowed
BodyVitals	HeartRate , BloodPressure , RespiratoryRate , BodyTemperature , BMI
LabResult	LabResultID , PatientID , TestName, Result, Date
ClinicalTrial	TrialID, Name, Description , Phase, Status, StartDate , EndDate

實體	Attributes
GenomicData	GenomicDataID , PatientID , Sequencedata , VariantInformation
Treatment	TreatmentID , Name, Description , Type, SideEffects
MedicalProcedure	ProcedureID , Name, Description , Risks, Outcomes
MedicalConcepts	UMLSCodes , MedicalVocabularies

下表列出實體可能擁有的關係及其對應的屬性。例如，Patient 實體可能會連線到具有 [UNDERGOES] 關係的 HospitalVisit 實體。此關係的屬性為 VisitDate。

主體實體	關係	物件實體	Attributes
Patient	[UNDERGOES]	HospitalVisit	VisitDate
HospitalVisit	[VISIT_IN]	HealthcareProvider	ProviderName , Location, ProviderID , VisitDate
HospitalVisit	[OBSERVED_CONDITION]	Symptoms	Severity, CurrentStatus , VisitDate
HospitalVisit	[RECEIVED_TREATMENT]	Treatment	Duration, Dosage, VisitDate
HospitalVisit	[PRESCRIBED]	Medication	Duration, Dosage, Adherence , VisitDate
Patient	[HAS_HISTORY]	PastMedicalHistory	無

主體實體	關係	物件實體	Attributes
PastMedicalHistory	[HAD_CONDITION]	MedicalCondition	DiagnosisDate , CurrentStatus
HospitalVisit	[PARTICIPATES_IN]	ClinicalTrial	VisitDate , Status, Outcomes
Patient	[HAS_GENOMIC_DATA]	GenomicData	CollectionDate
HospitalVisit	[OBSERVED_ALLERGIES]	Allergies	VisitDate
HospitalVisit	[CONDUCTED_LAB_TEST]	LabResult	VisitDate , AnalysisDate , Interpretation
HospitalVisit	[UNDERGOES]	MedicalProcedure	VisitDate , Outcome
MedicalCondition	[HAS_TAGGED]	MedicalConcepts	無
LabResult	[HAS_TAGGED]	MedicalConcepts	無
Treatment	[HAS_TAGGED]	MedicalConcepts	無
Symptoms	[HAS_TAGGED]	MedicalConcepts	無

### 步驟 3：建置內容擷取代理程式來查詢醫學知識圖表

在您建置醫療圖形資料庫之後，下一步是建置圖形互動的代理程式。客服人員會擷取醫生或臨床醫生輸入的查詢正確且必要的內容。設定這些從知識圖表擷取內容的代理程式有幾個選項：

- [Amazon Bedrock 代理程式](#)
- [LangChain 代理程式](#)

## 用於圖形互動的 Amazon Bedrock 代理程式

Amazon Bedrock [代理程式](#) 可與 Amazon Neptune 圖形資料庫無縫搭配使用。您可以透過 Amazon Bedrock [動作群組](#) 執行進階互動。動作群組會透過呼叫執行 Neptune openCypher 查詢的 AWS Lambda 函數來啟動程序。

若要查詢知識圖表，您可以使用兩種不同的方法：直接查詢執行或使用內容內嵌進行查詢。這些方法可以獨立套用或合併，視您的特定使用案例和排名條件而定。透過結合這兩種方法，您可以為 LLM 提供更全面的內容，從而改善結果。以下是兩種查詢執行方法：

- 無內嵌的直接 Cypher 查詢執行 – Lambda 函數會直接針對 Neptune 執行查詢，無需任何內嵌型搜尋。以下是此方法的範例：

```
MATCH (p:Patient)-[u:UNDERGOES]->(h:HospitalVisit) WHERE h.Reason = 'Acute Diabetes'
AND date(u.VisitDate) > date('2024-01-01')
RETURN p.PatientID, p.Name, p.Age, p.Gender, p.Address, p.ContactInformation
```

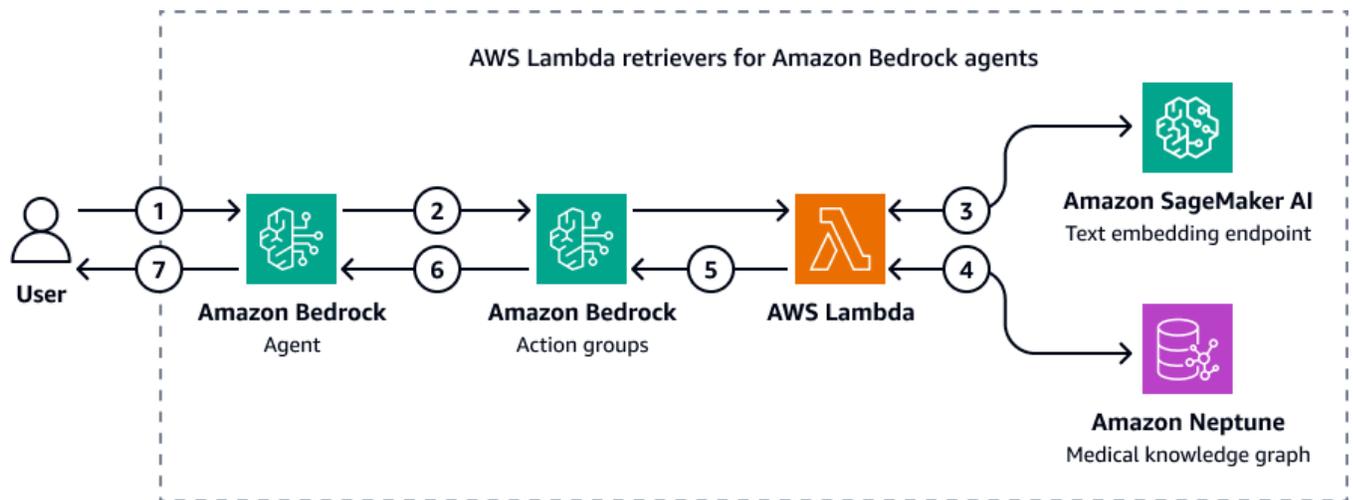
- 使用內嵌搜尋直接執行 Cypher 查詢 – Lambda 函數使用內嵌搜尋來增強查詢結果。此方法透過整合內嵌來增強查詢執行，這是一種密集的資料向量表示法。當查詢需要語意相似性或超出完全相符範圍的理解時，內嵌特別有用。您可以使用預先訓練或自訂訓練的模型，為每個醫療條件產生內嵌。以下是此方法的範例：

```
CALL { WITH "Acute Diabetes" AS query_term RETURN search_embedding(query_term) AS
similar_reasons }

MATCH (p:Patient)-[u:UNDERGOES]->(h:HospitalVisit) WHERE h.Reason IN similar_reasons
AND date(u.VisitDate) > date('2024-01-01')
RETURN p.PatientID, p.Name, p.Age, p.Gender, p.Address, p.ContactInformation
```

在此範例中，`search_embedding("Acute Diabetes")` 函數會擷取語意上接近「急性糖尿病」的條件。這有助於查詢尋找具有糖尿病前期或新陳能綜合徵等條件的患者。

下圖顯示 Amazon Bedrock 代理程式如何與 Amazon Neptune 互動，以便對醫學知識圖表執行 Cypher 查詢。



該圖顯示以下工作流程：

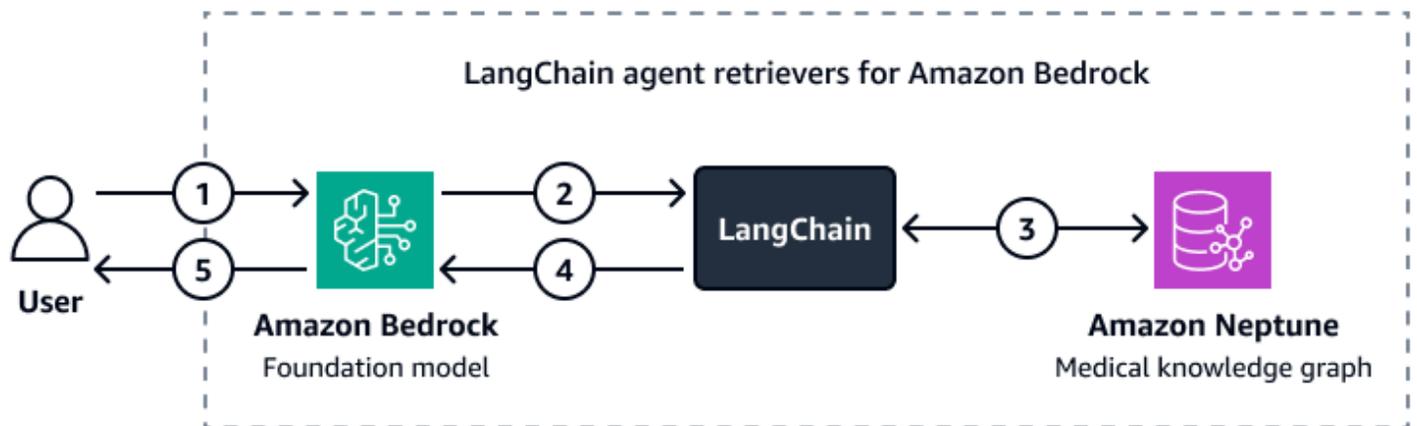
1. 使用者向 Amazon Bedrock 代理程式提交問題。
2. Amazon Bedrock 代理程式會將問題和輸入篩選條件變數傳遞至 Amazon Bedrock 動作群組。這些動作群組包含與 Amazon SageMaker AI 文字內嵌端點和 Amazon Neptune 醫療知識圖表互動的 AWS Lambda 函數。
3. Lambda 函數與 SageMaker AI 文字內嵌端點整合，以在 openCypher 查詢中執行語意搜尋。它會使用基礎LangChain代理程式，將自然語言查詢轉換為 openCypher 查詢。
4. Lambda 函數會查詢 Neptune 醫療知識圖表以取得正確的資料集，並從 Neptune 醫療知識圖表接收輸出。
5. Lambda 函數會將 Neptune 的結果傳回 Amazon Bedrock 動作群組。
6. Amazon Bedrock 動作群組會將擷取的內容傳送至 Amazon Bedrock 代理程式。
7. Amazon Bedrock 代理程式會使用原始使用者查詢和從知識圖表擷取的內容來產生回應。

## LangChain 圖形互動的代理程式

您可以LangChain與 Neptune 整合，以啟用以圖形為基礎的查詢和擷取。這種方法可以透過使用 Neptune 中的圖形資料庫功能來增強 AI 驅動的工作流程。自訂LangChain擷取器可做為中介裝置。Amazon Bedrock 中的基礎模型可以使用直接 Cypher 查詢和更複雜的圖形演算法與 Neptune 互動。

您可以使用自訂擷取器來精簡LangChain代理程式與 Neptune 圖形演算法的互動方式。例如，您可以使用少量的提示，這可協助您根據特定模式或範例來量身打造基礎模型的回應。您也可以套用 LLM 識別篩選條件，以精簡內容並改善回應的精確度。這可以改善與複雜圖形資料互動時整體擷取程序的效率和準確性。

下圖顯示自訂LangChain代理程式如何協調 Amazon Bedrock 基礎模型與 Amazon Neptune 醫療知識圖表之間的互動。



該圖顯示以下工作流程：

1. 使用者向 Amazon Bedrock 和LangChain客服人員提交問題。
2. Amazon Bedrock 基礎模型使用代理LangChain程式提供的 Neptune 結構描述來產生使用者問題的查詢。
3. LangChain 代理程式會根據 Amazon Neptune 醫療知識圖表執行查詢。
4. LangChain 代理程式會將擷取的內容傳送至 Amazon Bedrock 基礎模型。
5. Amazon Bedrock 基礎模型使用擷取的內容來產生使用者問題的答案。

## 步驟 4：建立即時描述性資料的知識庫

接下來，您將建立即時、描述性醫生與患者互動備註、診斷影像評估和實驗室分析報告的知識庫。此知識庫是[向量資料庫](#)。透過使用向量資料庫，可將描述性醫學知識以索引化的向量化形式存放，醫療保健供應商可以有效率地從大量儲存庫查詢和存取相關資訊。這些向量化表示式可協助您擷取語意相似的資料。照護提供者可以快速瀏覽臨床備註、醫療影像和實驗室結果。這透過提供內容相關資訊的即時存取，提高診斷和治療計劃的準確性和速度，從而加速明智的決策。

### 使用 OpenSearch Service 醫療知識庫

[Amazon OpenSearch Service](#) 可以管理大量高維度醫療資料。它是一種受管服務，可促進高效能搜尋和即時分析。它非常適合做為 RAG 應用程式的向量資料庫。OpenSearch Service 可做為後端工具來管理大量非結構化或半結構化資料，例如醫療記錄、研究文章和臨床備註。其進階語意搜尋功能可協助您擷取內容相關資訊。這使得它在臨床決策支援系統、患者查詢解決工具和醫療保健知識管理系統等應

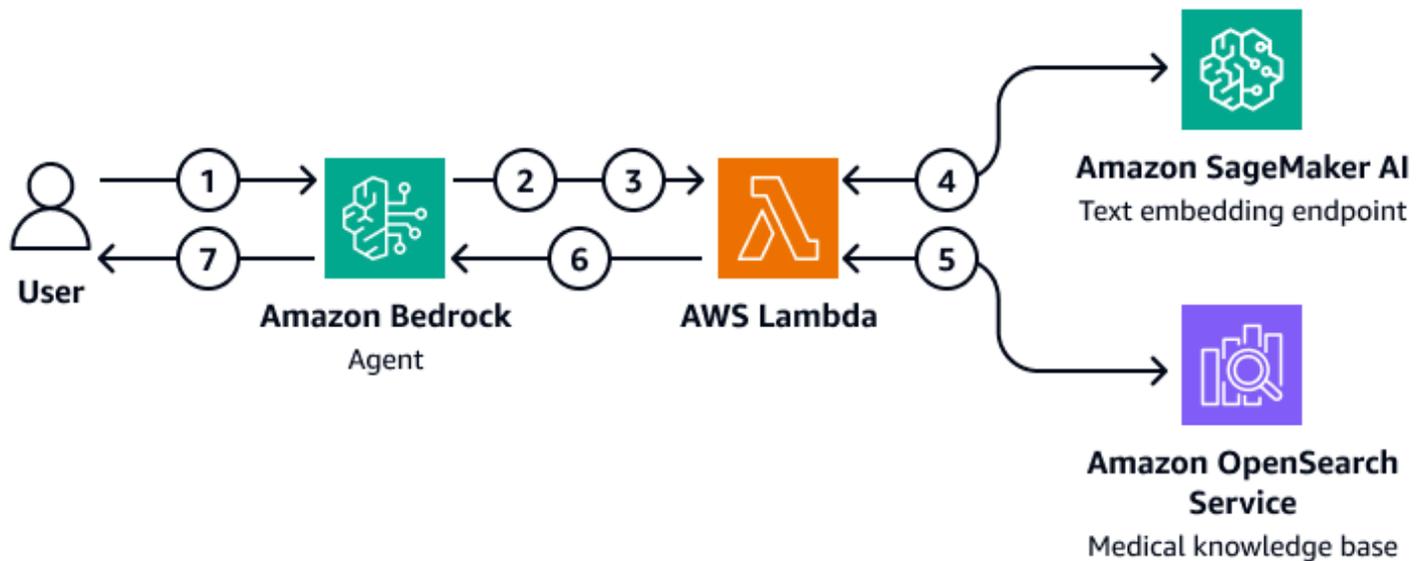
用程式中特別有用。例如，臨床醫生可以快速找到符合特定症狀或治療協定的相關患者資料或研究。這有助於臨床醫生做出由 up-to-date 和相關資訊通知的決策。

OpenSearch Service 可以擴展和處理即時資料索引和查詢。這使得它非常適合動態醫療保健環境，這些環境需要及時存取準確的資訊。此外，它具有多模式搜尋功能，最適合需要多個輸入的搜尋，例如醫療影像和醫生備註。為醫療保健應用程式實作 OpenSearch Service 時，請務必定義精確的欄位和映射，以最佳化資料索引和擷取。欄位代表個別資料片段，例如病患記錄、醫療歷史記錄和診斷代碼。映射會定義存放和查詢這些欄位的方式（內嵌形式或原始形式）。對於醫療保健應用程式，建立可適應各種資料類型的映射至關重要，包括結構化資料（例如數值測試結果）、半結構化資料（例如病患備註）和非結構化資料（例如醫療影像）

在 OpenSearch Service 中，您可以透過策劃的提示來執行全文 [神經搜尋](#) 查詢，以搜尋醫療記錄、臨床筆記或研究論文，以快速找到特定症狀、治療或病患歷史記錄的相關資訊。神經搜尋查詢會使用內建神經網路模型，自動處理輸入提示和映像的內嵌。這有助於其了解和擷取多模態資料中更深層的語意關係，相較於其他搜尋查詢演算法，可提供更了解內容和精確的搜尋結果，例如 k 近鄰 (k-NN) 搜尋。

## 建立 RAG 架構

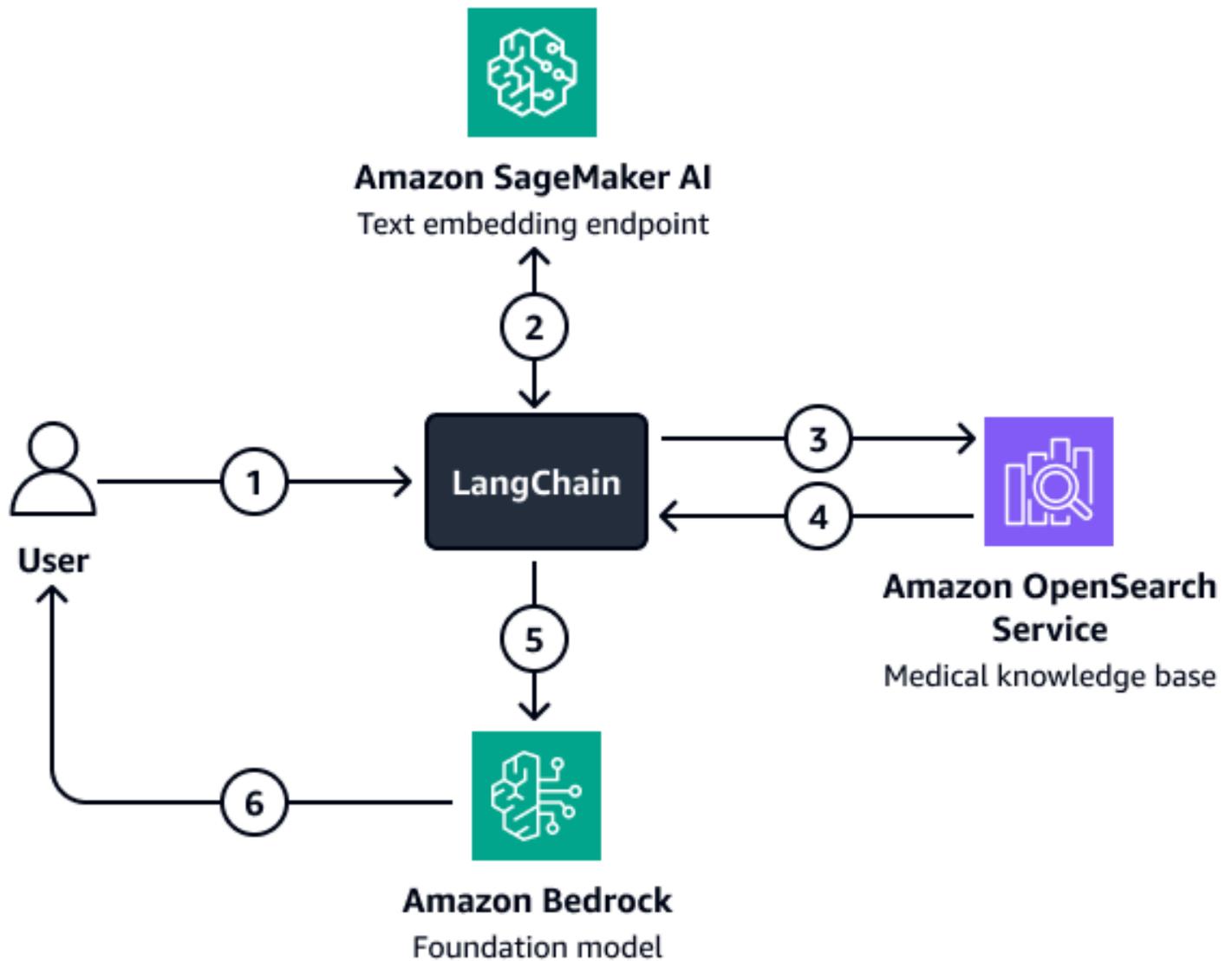
您可以部署自訂 RAG 解決方案，該解決方案使用 Amazon Bedrock 代理程式在 OpenSearch Service 中查詢醫療知識庫。若要達成此目的，您可以建立可與互動並查詢 OpenSearch Service 的 AWS Lambda 函數。Lambda 函數透過存取 SageMaker AI 文字內嵌端點來內嵌使用者的輸入問題。Amazon Bedrock 代理程式會將其他查詢參數做為輸入傳遞至 Lambda 函數。函數會查詢 OpenSearch Service 中的醫療知識庫，其會傳回相關的醫療內容。設定 Lambda 函數之後，請在 Amazon Bedrock 代理程式中將其新增為動作群組。Amazon Bedrock 代理程式會接受使用者的輸入、識別必要的變數、將變數和問題傳遞至 Lambda 函數，然後啟動函數。函數會傳回內容，以協助基礎模型為使用者的問題提供更準確的答案。



該圖顯示以下工作流程：

1. 使用者向 Amazon Bedrock 代理程式提交問題。
2. Amazon Bedrock 代理程式會選取要啟動的動作群組。
3. Amazon Bedrock 代理程式會啟動 AWS Lambda 函數，並將參數傳遞給該函數。
4. Lambda 函數會啟動 Amazon SageMaker AI 文字內嵌模型，以內嵌使用者問題。
5. Lambda 函數會將內嵌文字和其他參數和篩選條件傳遞至 Amazon OpenSearch Service。Amazon OpenSearch Service 會查詢醫療知識庫，並將結果傳回 Lambda 函數。
6. Lambda 函數會將結果傳回給 Amazon Bedrock 代理程式。
7. Amazon Bedrock 代理程式中的基礎模型會根據結果產生回應，並將回應傳回給使用者。

對於涉及更複雜篩選的情況，您可以使用自訂LangChain擷取器。透過設定直接載入的 OpenSearch Service 向量搜尋用戶端來建立此擷取器LangChain。此架構可讓您傳遞更多變數，以建立篩選條件參數。設定擷取器之後，請使用 Amazon Bedrock 模型和擷取器來設定擷取問答鏈。此鏈透過將使用者輸入和潛在篩選條件傳遞給擷取器，協調模型和擷取器之間的互動。擷取器會傳回相關內容，以協助基礎模型回答使用者的問題。



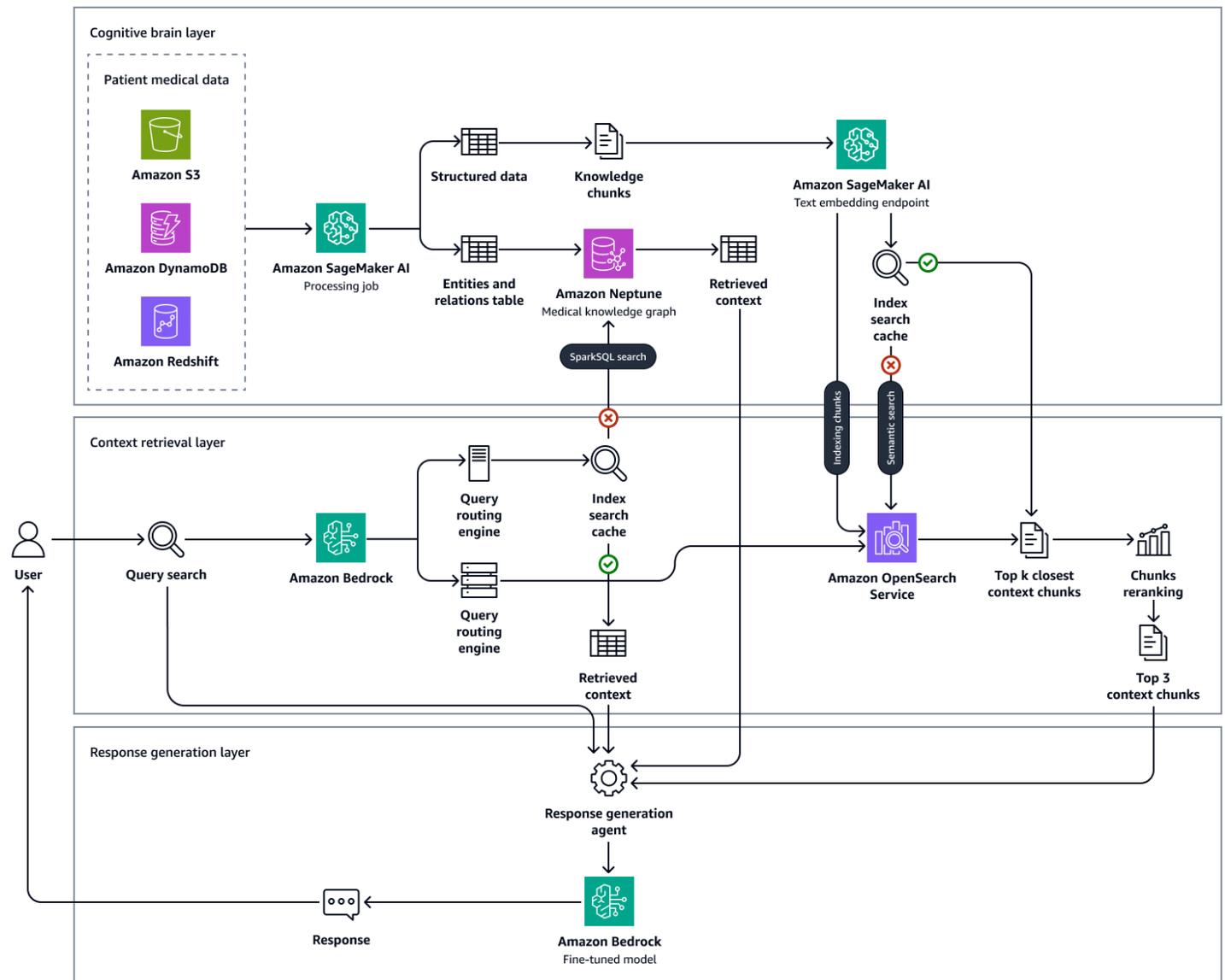
該圖顯示以下工作流程：

1. 使用者向LangChain擷取程式代理程式提交問題。
2. LangChain 擷取器代理程式會將問題傳送至 Amazon SageMaker AI 文字內嵌端點，以內嵌問題。
3. LangChain 擷取程式代理程式會將內嵌文字傳遞給 Amazon OpenSearch Service。
4. Amazon OpenSearch Service 會將擷取的文件傳回給LangChain擷取器代理程式。
5. LangChain 擷取器代理程式會將使用者問題和擷取的內容傳遞至 Amazon Bedrock 基礎模型。
6. 基礎模型會產生回應並將其傳送給使用者。

## 步驟 5：使用 LLMs 回答醫療問題

上述步驟可協助您建置醫療智慧應用程式，以擷取病患的醫療記錄，並摘要相關藥物和潛在診斷。現在，您可以建置產生層。此層使用 Amazon Bedrock 中 LLM 的生成功能，例如 Llama 3，來擴增應用程式的輸出。

當臨床醫生輸入查詢時，應用程式的內容擷取層會從知識圖表執行擷取程序，並傳回與病患歷史記錄、人口統計特性、症狀、診斷和結果相關的最高記錄。從向量資料庫中，它還從醫學研究和學術書籍的龐大組合中擷取即時、描述性的醫生與患者互動備註、診斷影像評估洞察、實驗室分析報告摘要和洞察。這些最常擷取的結果、臨床醫生的查詢和提示（根據查詢的性質量身打造來策劃答案），接著會傳遞到 Amazon Bedrock 中的基礎模型。這是回應產生層。LLM 使用擷取的內容來產生對臨床醫生查詢的回應。下圖顯示此解決方案中步驟的 end-to-end 工作流程。



您可以針對醫療智慧應用程式必須處理的一系列使用案例，在 Amazon Bedrock 中使用預先訓練的基礎模型，例如 Llama 3。指定任務最有效的 LLM 會根據使用案例而有所不同。例如，預先訓練的模型可能足以總結病患醫生對話、搜尋藥物和病患歷史記錄，以及從內部醫療資料集和科學知識內文中擷取洞見。不過，其他複雜的使用案例可能需要微調 LLM，例如即時實驗室評估、醫療程序建議，以及患者結果的預測。您可以在醫療網域資料集上進行訓練，以微調 LLM。特定或複雜的醫療保健和生命科學需求可推動這些微調模型的開發。

如需微調 LLM 或選擇已針對醫療網域資料進行訓練之現有 LLM 的詳細資訊，請參閱[使用大型語言模型進行醫療保健和生命科學使用案例](#)。

## 符合 AWS Well-Architected 架構

解決方案符合 [AWS Well-Architected Framework](#) 的所有六個支柱，如下所示：

- 卓越營運 – 架構會分離，以實現高效的監控和更新。Amazon Bedrock 代理程式和 AWS Lambda 可協助您快速部署和復原工具。
- 安全性 – 此解決方案的設計符合醫療保健法規，例如 HIPAA。您也可以實作加密、精細存取控制和 Amazon Bedrock 護欄，以協助保護病患資料。
- 可靠性 – AWS 受管服務，例如 Amazon OpenSearch Service 和 Amazon Bedrock，提供持續模型互動的基礎設施。
- 效能效率 – RAG 解決方案會使用最佳化語意搜尋和 Cypher 查詢快速擷取相關資料，同時代理程式路由器會識別使用者查詢的最佳路由。
- 成本最佳化 – Amazon Bedrock 和 RAG 架構中的 pay-per-token 模型可降低推論和預先訓練成本。
- 永續性 – 使用無伺服器基礎設施和 pay-per-token 運算可將資源使用量降至最低，並增強永續性。

## 使用案例：預測患者結果和重新入院率

AI 驅動的預測分析透過預測患者結果並啟用個人化治療計劃，提供進一步的好處。這可以改善患者滿意度和健康結果。透過將這些 AI 功能與 Amazon Bedrock 和其他技術整合，醫療保健供應商可以實現顯著的生產力提高、降低成本並提高患者護理的整體品質。

您可以在[知識圖表](#)中存放醫療資料，例如病患歷史記錄、臨床備註、藥物和處理方式。透過結合對 LLMs 的深入內容理解與醫學知識圖表中的結構化時間資料，醫療保健供應商可以獲得對個別患者模式的其他洞見。使用預測分析，您可以及早識別潛在的不遵循或處理複雜性，並產生個人化的重新入學傾向分數。

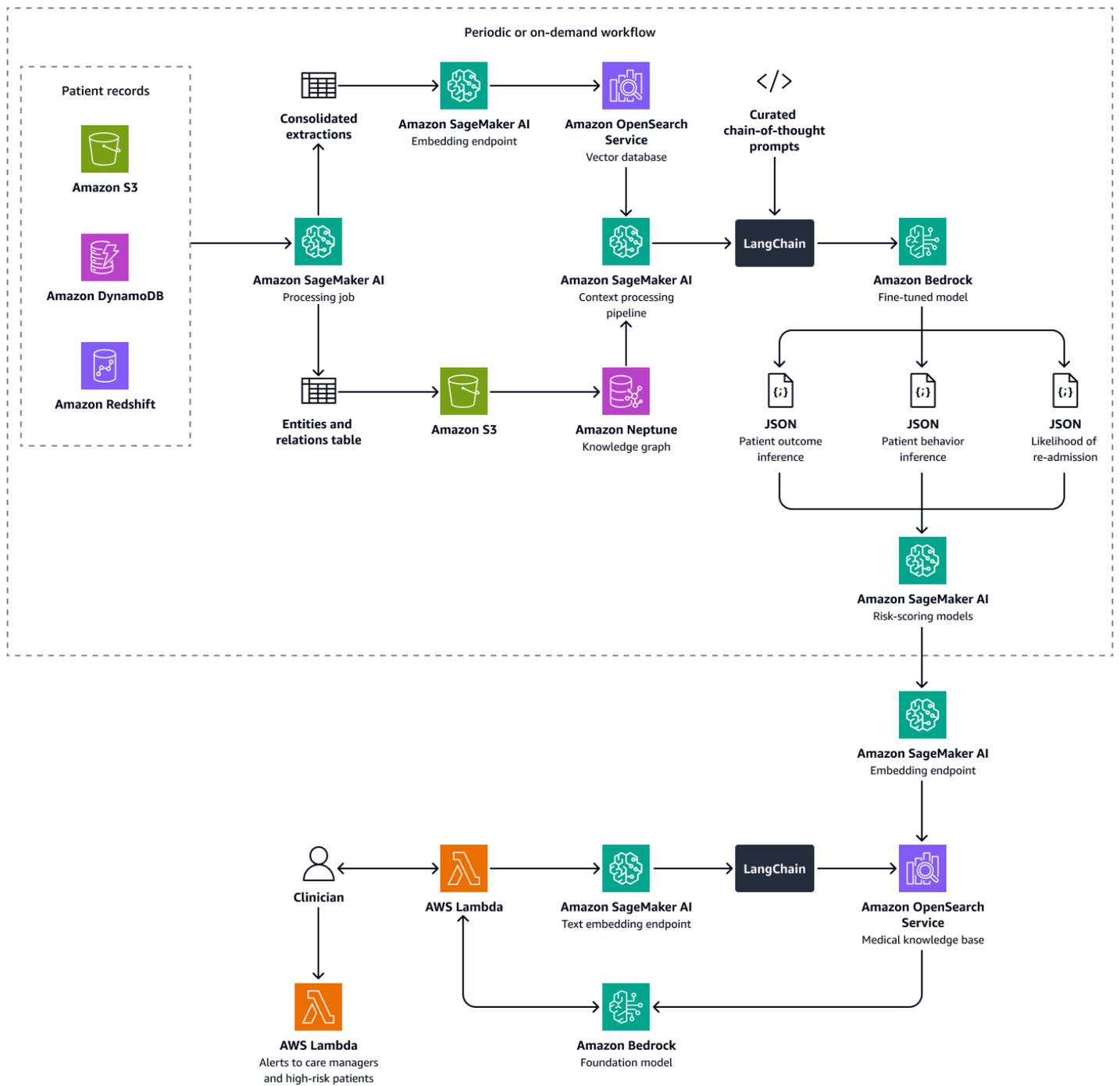
此解決方案可協助您預測重新入學的可能性。這些預測可以改善患者結果並降低醫療保健成本。此解決方案也可以協助醫院臨床醫生和管理員將注意力集中在具有較高重新入院風險的患者。它還有助於他們透過提醒、自助服務和資料驅動的動作，向這些患者啟動主動介入。

### 解決方案概觀

此解決方案使用多擷取器擷取增強生成 (RAG) 架構來分析患者資料。它預測個別患者重新入院的可能性，並協助您計算醫院層級的重新入院傾向分數。此解決方案整合了下列功能：

- 知識圖表 – 存放結構化、按時間排序的患者資料，例如醫院事件、先前的重新入院、症狀、實驗室結果、規定的處理方式，以及藥物依從性歷史記錄
- 向量資料庫 – 存放非結構化臨床資料，例如出院摘要、醫生備註，以及錯過預約或報告藥物副作用的記錄
- 微調 LLM – 使用知識圖表中的結構化資料和向量資料庫中的非結構化資料，以產生有關患者行為、治療遵循和重新入院可能性的推論

風險評分模型會將 LLM 的推論量化為數值分數。您可以將分數彙總為醫院層級的重新入學傾向分數。此分數會定義每位病患的風險暴露，您可以定期或視需要進行計算。所有推論和風險分數都會編製索引並存放在 Amazon OpenSearch Service 中，以便照護管理員和臨床醫生可以擷取它。透過將對話式 AI 代理程式與此向量資料庫整合，臨床醫生和護理經理可以順暢地擷取個別患者層級、全設施層級或醫療專科的洞見。您也可以根據風險分數設定自動提醒，以鼓勵主動介入。



建置此解決方案包含下列步驟：

- [步驟 1：使用醫學知識圖表預測病患結果](#)
- [步驟 2：預測患者對處方藥或治療的行為](#)
- [步驟 3：預測患者重新入院的可能性](#)
- [步驟 4：計算醫院重新入院傾向分數](#)

## 步驟 1：使用醫學知識圖表預測病患結果

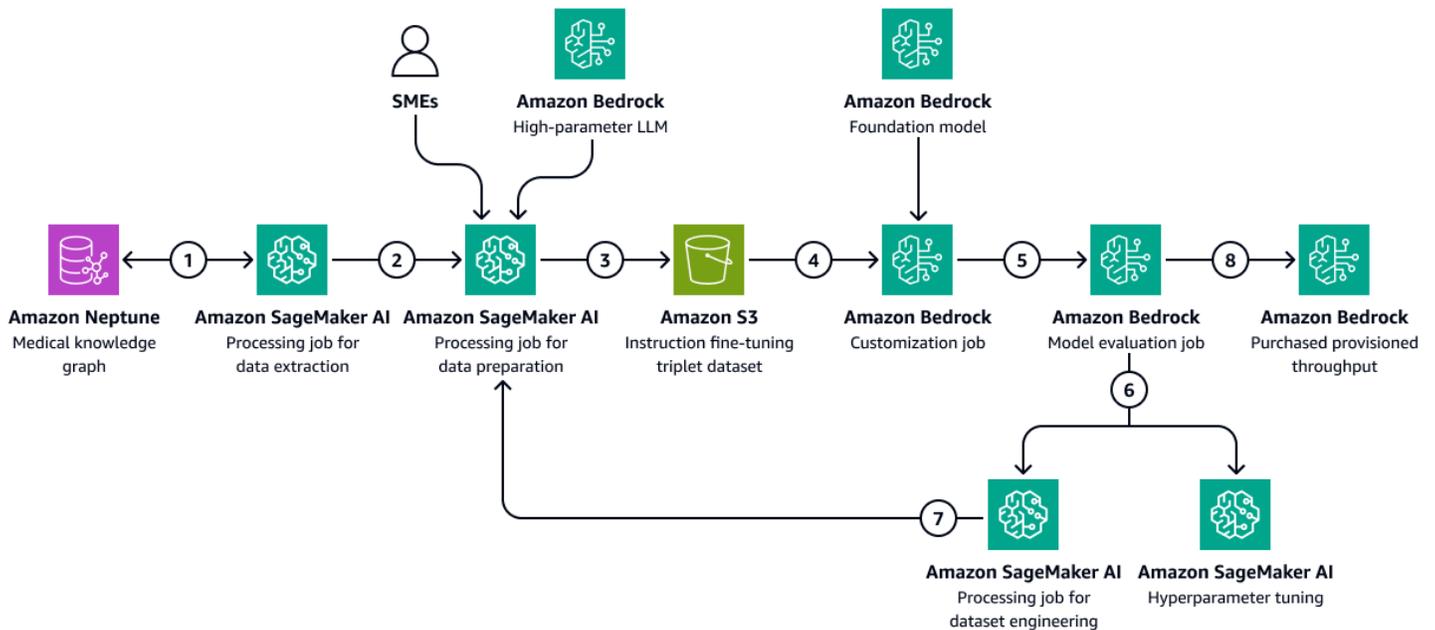
在 [Amazon Neptune](#) 中，您可以使用知識圖表來儲存一段時間內有關患者就診和結果的臨時知識。建置和存放知識圖形最有效的方法是使用圖形模型和圖形資料庫。圖形資料庫專為存放和導覽關係而建置。圖形資料庫可讓您更輕鬆地建立和管理高度連線的資料，並具有靈活的結構描述。

知識圖表可協助您執行時間序列分析。以下是圖形資料庫的關鍵元素，用於暫時預測患者結果：

- 歷史資料 – 先前診斷、繼續藥物、先前使用藥物，以及病患的實驗室結果
- 患者門診（時間） – 門診日期、症狀、觀察到的敏感、臨床備註、診斷、程序、治療、處方藥和實驗室結果
- 症狀和臨床參數 – 臨床和以症狀為基礎的資訊，包括嚴重性、進展模式，以及患者對藥物的反應

您可以使用醫學知識圖表中的洞見來微調 Amazon Bedrock 中的 LLM，例如 Llama 3。您可以使用有關患者對一組藥物或治療隨時間回應的循序患者資料來微調 LLM。使用標記的資料集，將中的一組藥物或治療和患者-臨床互動資料分類為預先定義的類別，以指示患者的運作狀態。這些類別的範例包括運作狀態惡化、改善或穩定的進度。當臨床醫生輸入有關患者及其症狀的新內容時，微調後的 LLM 可以使用訓練資料集的模式來預測潛在的患者結果。

下圖顯示使用醫療保健特定的訓練資料集，在 Amazon Bedrock 中微調 LLM 所涉及的循序步驟。此資料可能包括患者醫療情況，以及隨著時間的推移對治療的反應。此訓練資料集有助於模型對患者結果進行一般性預測。



該圖顯示以下工作流程：

1. Amazon SageMaker AI 資料擷取任務會查詢知識圖表，以擷取不同患者隨著時間對一組藥物或處理方式回應的時間資料。
2. SageMaker AI 資料準備任務整合了 Amazon Bedrock LLM 和主題專家 (SMEs) 的輸入。任務會將從知識圖表擷取的資料分類為預先定義的類別（例如，運作狀態惡化、改善或穩定的進度），以指出每個病患的運作狀態。
3. 任務會建立微調資料集，其中包含從知識圖表擷取的資訊、chain-of-thought 提示和病患結果類別。它會將此訓練資料集上傳至 Amazon S3 儲存貯體。
4. Amazon Bedrock 自訂任務會使用此訓練資料集來微調 LLM。
5. Amazon Bedrock 自訂任務整合了訓練環境中選擇的 Amazon Bedrock 基礎模型。它會啟動微調任務，並使用訓練資料集和您設定的訓練超參數。
6. Amazon Bedrock 評估任務會使用預先設計的模型評估架構來評估微調後的模型。
7. 如果模型需要改善，在仔細考慮訓練資料集之後，訓練任務會使用更多資料再次執行。如果模型未顯示增量效能改善，也請考慮修改訓練超參數。
8. 模型評估符合業務利益相關者定義的標準後，您會將微調後的模型託管至 Amazon Bedrock 佈建的輸送量。

## 步驟 2：預測患者對處方藥或治療的行為

微調 LLMs 可以處理臨時醫學知識圖表中的臨床備註、放電摘要和其他患者特定文件。他們可以評估患者是否可能遵循處方藥或治療。

此步驟使用在 中建立的知識圖表 [步驟 1：使用醫學知識圖表預測病患結果](#)。知識圖表包含來自病患設定檔的資料，包括病患作為節點的歷史遵循。它還包括不遵守藥物或處理方式的執行個體、藥物的副作用、缺乏藥物的存取或成本障礙，或複雜用量方案作為此類節點的屬性。

微調 LLMs 可以使用來自醫學知識圖表的過去處方履行資料，以及來自 Amazon OpenSearch Service 向量資料庫的臨床備註描述性摘要。這些臨床備註可能會提及經常錯過的預約或不遵守治療。LLM 可以使用這些備註來預測未來不遵守的可能性。

1. 準備輸入資料，如下所示：
  - 結構化資料 – 從醫學知識圖表中擷取最近的患者資料，例如過去三次的就診和實驗室結果。
  - 非結構化資料 – 從 Amazon OpenSearch Service 向量資料庫中擷取最新的臨床備註。
2. 建構包含病患歷史記錄和目前內容的輸入提示。以下是提示範例：

You are a highly specialized AI model trained in healthcare predictive analytics. Your task is to analyze a patient's historical medical records, adherence patterns, and clinical context to predict the **likelihood of future non-adherence** to prescribed medications or treatments.

### ### **Patient Details**

- **Patient ID:** {patient\_id}
- **Age:** {age}
- **Gender:** {gender}
- **Medical Conditions:** {medical\_conditions}
- **Current Medications:** {current\_medications}
- **Prescribed Treatments:** {prescribed\_treatments}

### ### **Chronological Medical History**

- **Visit Dates & Symptoms:** {visit\_dates\_symptoms}
- **Diagnoses & Procedures:** {diagnoses\_procedures}
- **Prescribed Medications & Treatments:** {medications\_treatments}
- **Past Adherence Patterns:** {historical\_adherence}
- **Instances of Non-Adherence:** {past\_non\_adherence}
- **Side Effects Experienced:** {side\_effects}
- **Barriers to Adherence (e.g., Cost, Access, Dosing Complexity):** {barriers}

### ### **Patient-Specific Insights**

- **Clinical Notes & Discharge Summaries:** {clinical\_notes}
- **Missed Appointments & Non-Compliance Patterns:** {missed\_appointments}

### ### **Let's think Step-by-Step to predict the patient behaviour**

1. You should first analyze past adherence trends and patterns of non-adherence.
2. Identify potential barriers, such as financial constraints, medication side effects, or complex dosing regimens.
3. Thoroughly examine clinical notes and documented patient behaviors that may hint at non-adherence.
4. Correlate adherence history with prescribed treatments and patient conditions.
5. Finally predict the likelihood of non-adherence based on these contextual insights.

### ### **Output Format (JSON)**

Return the prediction in the following structured format:

```
```json
{
  "patient_id": "{patient_id}",
  "likelihood_of_non_adherence": "{low | moderate | high}",
  "reasoning": "{detailed_explanation_based_on_patient_history}"
}
```

```
}
```

3. 將提示傳遞至微調的 LLM。LLM 會處理提示並預測結果。以下是來自 LLM 的範例回應：

```
{  
  "patient_id": "P12345",  
  "likelihood_of_non_adherence": "high",  
  "reasoning": "The patient has a history of missed appointments, has reported side effects to previous medications. Additionally, clinical notes indicate difficulty following complex dosing schedules."  
}
```

4. 剖析模型的回應，以擷取預測的結果類別。例如，上一個步驟中範例回應的類別可能是不遵守的可能性很高。
5. (選用) 使用模型日誌或其他方法來指派可信度分數。日誌是屬於特定類別的項目的非標準化機率。

## 步驟 3：預測患者重新入院的可能性

由於醫療保健管理的高成本，以及其對患者良好狀態的影響，重新入院是主要的考量。運算醫院重新入院率是衡量醫療保健供應商患者照護品質和效能的一種方式。

若要計算重新加入率，您定義了一個指標，例如 7 天的重新加入率。此指標是在出院後七天內返回醫院進行非計劃就診的已入院患者百分比。為了預測患者重新入院的機會，微調後的 LLM 可能會耗用您在中建立的醫學知識圖表中的暫時資料[步驟 1：使用醫學知識圖表預測病患結果](#)。此知識圖表會維護病患事件、程序、藥物和症狀的時間記錄。這些資料記錄包含下列項目：

- 自病患上次出院以來的持續時間
- 病患對過去治療和藥物的反應
- 隨著時間的推移，症狀或條件的進展

您可以處理這些時間序列事件，透過策劃的系統提示來預測患者重新入院的可能性。提示會將預測邏輯提供給微調的 LLM。

1. 準備輸入資料，如下所示：

- 依從性歷史記錄 – 從醫學知識圖表中擷取藥物收件日期、藥物補充頻率、診斷和藥物詳細資訊、時間性醫療史，以及其他資訊。
- 行為指標 – 擷取並包含有關錯過預約和患者報告副作用的臨床備註。

## 2. 建構輸入提示，其中包含遵循歷史記錄和行為指標。以下是提示範例：

You are a highly specialized AI model trained in healthcare predictive analytics. Your task is to analyze a patient's historical medical records, clinical events, and adherence patterns to predict the **likelihood of hospital readmission** within the next few days.

### ### **Patient Details**

- **Patient ID:** {patient\_id}
- **Age:** {age}
- **Gender:** {gender}
- **Primary Diagnoses:** {diagnoses}
- **Current Medications:** {current\_medications}
- **Prescribed Treatments:** {prescribed\_treatments}

### ### **Chronological Medical History**

- **Recent Hospital Encounters:** {encounters}
- **Time Since Last Discharge:** {time\_since\_last\_discharge}
- **Previous Readmissions:** {past\_readmissions}
- **Recent Lab Results & Vital Signs:** {recent\_lab\_results}
- **Procedures Performed:** {procedures\_performed}
- **Prescribed Medications & Treatments:** {medications\_treatments}
- **Past Adherence Patterns:** {historical\_adherence}
- **Instances of Non-Adherence:** {past\_non\_adherence}

### ### **Patient-Specific Insights**

- **Clinical Notes & Discharge Summaries:** {clinical\_notes}
- **Missed Appointments & Non-Compliance Patterns:** {missed\_appointments}
- **Patient-Reported Side Effects & Complications:** {side\_effects}

### ### **Reasoning Process – You have to analyze this use case step-by-step.**

1. First assess **time since last discharge** and whether recent hospital encounters suggest a pattern of frequent readmissions.
2. Second examine **recent lab results, vital signs, and procedures performed** to identify clinical deterioration.
3. Third analyze **adherence history**, checking if past non-adherence to medications or treatments correlates with readmissions.
4. Then identify **missed appointments, self-reported side effects, or symptoms worsening** from clinical notes.
5. Finally predict the **likelihood of readmission** based on these contextual insights.

### ### **Output Format (JSON)**

```
Return the prediction in the following structured format:
```

```
```json
{
  "patient_id": "{patient_id}",
  "likelihood_of_readmission": "{low | moderate | high}",
  "reasoning": "{detailed_explanation_based_on_patient_history}"
}
```

3. 將提示傳遞至微調的 LLM。LLM 會處理提示，並預測重新入院的可能性和原因。以下是來自 LLM 的範例回應：

```
{
  "patient_id": "P67890",
  "likelihood_of_readmission": "high",
  "reasoning": "The patient was discharged only 5 days ago, has a history of more than two readmissions to hospitals where the patient received treatment. Recent lab results indicate abnormal kidney function and high liver enzymes. These factors suggest a medium risk of readmission."
}
```

4. 將預測分類為標準化規模，例如低、中或高。
5. 檢閱 LLM 提供的推理，並識別有助於預測的關鍵因素。
6. 將定性輸出映射至量化分數。例如，非常高可能等於 0.9 的機率。
7. 使用驗證資料集，根據實際重新入學率校正模型輸出。

## 步驟 4：計算醫院重新入院傾向分數

接下來，您可以計算每位病患的醫院重新入院傾向分數。此分數反映在先前步驟中執行的三個分析的淨影響：潛在的患者結果、患者對藥物和治療的行為，以及患者重新入院的可能性。透過將患者層級重新入院傾向分數彙總至專科層級，然後在醫院層級，您可以獲得臨床醫生、護理經理和管理員的洞察。醫院重新入院傾向分數可協助您依設施、專科或條件評估整體效能。然後，您可以使用此分數來實作主動介入。

1. 將權重指派給每個不同的因素（結果預測、遵循可能性、重新加入）。以下是範例權重：

- 結果預測權重：0.4
- 依從性預測權重：0.3
- 重新加入可能性權重：0.3

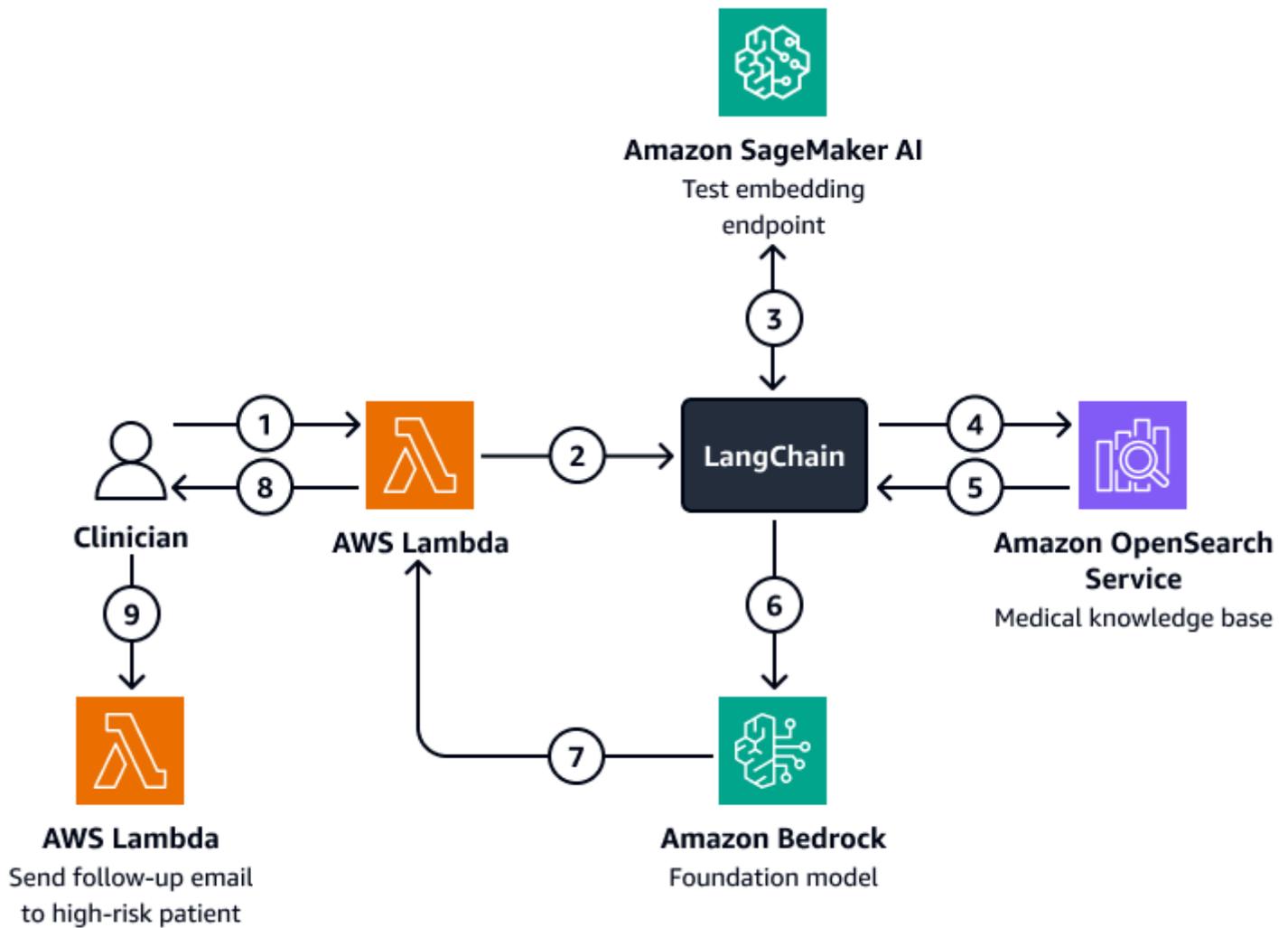
2. 使用下列計算來計算複合分數：

$$\text{ReadmissionPropensityScore} = (\text{OutcomeScore} \times \text{OutcomeWeight}) + (\text{AdherenceScore} \times \text{AdherenceWeight}) + (\text{ReadmissionLikelihoodScore} \times \text{ReadmissionLikelihoodWeight})$$

3. 確保所有個別分數都位於相同的規模，例如 0 到 1。
4. 定義動作的閾值。例如，高於 0.7 的分數會啟動提醒。

根據上述分析和患者重新入學傾向分數，臨床醫生或護理經理可以設定提醒，根據計算的分數來監控其個別患者。如果超過預先定義的閾值，則會在達到該閾值時收到通知。這有助於護理管理員在為其患者建立出院護理計劃時保持主動而不是被動。在 Amazon OpenSearch Service 向量資料庫中以索引形式儲存病患結果、行為和重新入院傾向分數，以便照護管理員可以使用對話式 AI 代理程式無縫擷取。

下圖顯示對話式 AI 代理器的工作流程，臨床醫生或照護管理員可用來擷取有關患者結果、預期行為和重新入院傾向的洞察。使用者可以擷取患者層級、部門層級或醫院層級的洞見。AI 代理器會擷取這些洞見，這些洞見以索引形式存放在 Amazon OpenSearch Service 向量資料庫中。代理程式使用查詢來擷取相關資料並提供量身打造的回應，包括為具有高重新入院風險的患者建議的動作。根據風險等級，代理程式也可以為患者和護理提供者設定提醒。



該圖顯示以下工作流程：

1. 臨床醫生向對話式 AI 代理器提出問題，該代理程式包含 AWS Lambda 函數。
2. Lambda 函數會啟動LangChain代理程式。
3. LangChain 代理程式會將使用者的問題傳送至 Amazon SageMaker AI 文字內嵌端點。端點內嵌問題。
4. LangChain 代理程式會將內嵌問題傳遞給 Amazon OpenSearch Service 中的醫學知識庫。
5. Amazon OpenSearch Service 會將與使用者查詢最相關的特定洞見傳回給LangChain客服人員。
6. LangChain 代理程式會將查詢和擷取的內容從知識庫傳送至 Amazon Bedrock 基礎模型。
7. Amazon Bedrock 基礎模型會產生回應，並將其傳送至 Lambda 函數。
8. Lambda 函數會將回應傳回給臨床醫生。
9. 臨床醫生會啟動 Lambda 函數，將追蹤電子郵件傳送給具有重新入院高風險的患者。

## 符合 AWS Well-Architected 架構

追蹤患者行為和預測醫院重新入院率的架構整合 AWS 服務了醫學知識圖表和 LLMs，可改善醫療成果，同時符合 [AWS Well-Architected Framework](#) 的六個支柱：

- 卓越營運 – 解決方案是解耦的自動化系統，使用 Amazon Bedrock 和 AWS Lambda 進行即時提醒。
- 安全性 – 此解決方案的設計符合醫療保健法規，例如 HIPAA。您也可以實作加密、精細存取控制和 Amazon Bedrock 護欄，以協助保護病患資料。
- 可靠性 – 架構使用容錯、無伺服器 AWS 服務。
- 效能效率 – Amazon OpenSearch Service 和微調的 LLMs 可以提供快速準確的預測。
- 成本最佳化 – 無伺服器技術和pay-per-inference模型有助於將成本降至最低。雖然使用微調 LLM 的可能會產生額外費用，但模型會使用 RAG 方法來減少微調程序所需的資料和運算時間。
- 永續性 – 架構透過使用無伺服器基礎設施將資源消耗降至最低。它還支援有效率、可擴展的醫療保健操作。

## 使用案例：管理和提升您的醫療保健人員技能

實作人才轉型和提升技能策略，有助於員工繼續熟悉在醫療和醫療保健服務中使用新技術和實務。主動提升技能計畫可確保醫療專業人員可以提供高品質的病患護理、最佳化營運效率，以及符合法規標準。此外，人才轉型促進持續學習的文化。這對於適應不斷變化的醫療保健環境和解決新興的公共健康挑戰至關重要。傳統的訓練方法，例如課堂式訓練和靜態學習模組，可為廣泛的受眾提供統一的內容。他們通常缺乏個人化的學習路徑，這對於解決個別從業人員的特定需求和熟練程度至關重要。這個one-size-fits-all策略可能會導致分離和次佳的知識保留。

因此，醫療保健組織必須採用創新、可擴展且技術驅動的解決方案，這些解決方案可以判斷每位員工在其目前狀態和潛在未來狀態中的差距。這些解決方案應該建議超個人化學習途徑和正確的一組學習內容。這可以有效地為人力資源做好未來醫療保健的準備。

在醫療保健產業中，您可以套用生成式 AI，以協助您了解和提升人力資源。透過大型語言模型 (LLMs) 和進階擷取器的連線，組織可以了解他們目前擁有的技能，並識別未來可能需要的關鍵技能。此資訊可協助您透過招聘新工作者和提升目前人力的技能來彌補差距。使用 Amazon Bedrock 和知識圖表，醫療保健組織可以開發領域特定的應用程式，以促進持續學習和技能開發。

此解決方案所提供的知識可協助您有效管理人才、最佳化人力資源績效、推動組織成功、識別現有技能，以及制定人才策略。此解決方案可協助您在幾週內執行這些任務，而不是幾個月。

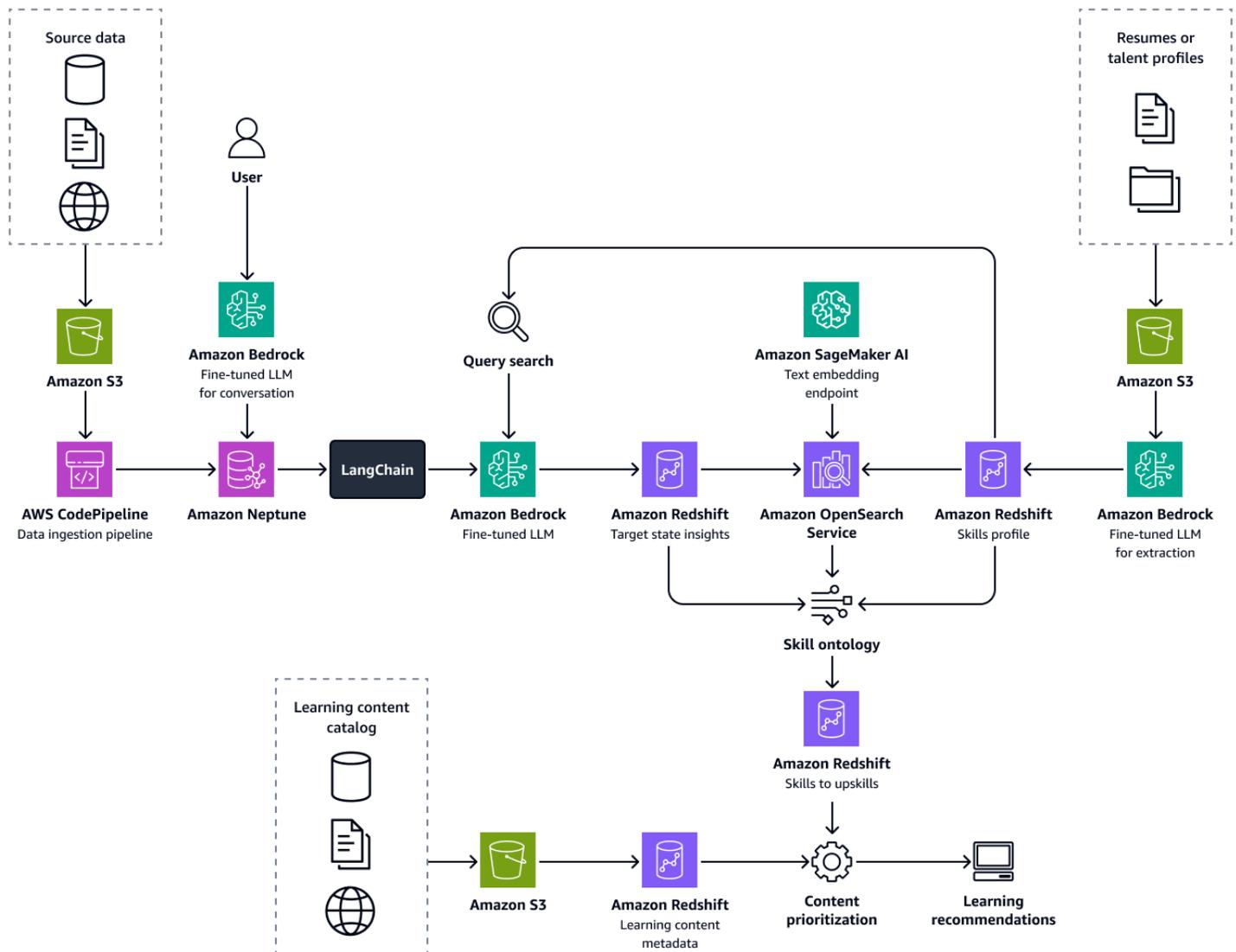
## 解決方案概觀

此解決方案是一種醫療保健人才轉型架構，包含下列元件：

- 智慧型繼續剖析器 – 此元件可以讀取候選者的繼續，並精確擷取候選資訊，包括技能。在專屬訓練資料集上使用 Amazon Bedrock 中微調 Llama 2 模型建置的智慧型資訊擷取解決方案，涵蓋來自超過 19 個產業的履歷和人才設定檔。此 LLM 型程序透過自動化恢復的手動檢閱程序，並將最佳候選項目配對至開放角色，節省數百小時。
- 知識圖表 – 建立在 Amazon Neptune 上的知識圖表，這是人才資訊的統一儲存庫，包括組織和產業的角色和技能分類，使用技能定義、角色及其屬性、關係和邏輯限制來擷取醫療保健人才的語意。
- 技能本質 – 候選技能與理想目前狀態或未來狀態技能（使用知識圖表擷取）之間的技能鄰近性探索是透過衡量候選技能和目標狀態技能之間語意相似性的本質演算法達成。
- 學習途徑和內容 – 此元件是一種學習建議引擎，可根據識別的技能差距，從任何供應商的學習材料目錄中建議正確的學習內容。透過分析技能差距並建議優先學習內容來識別每個候選者的最佳提升技能途徑，以便在轉換到新角色期間為每個候選者實現無縫且持續的專業發展。

此雲端型自動化解決方案採用機器學習服務、LLMs、知識圖表和擷取增強生成 (RAG) 技術。它可以在最短時間內擴展以處理數十或數千個恢復、建立即時候選設定檔、識別其目前或潛在未來狀態的差距，然後有效地建議正確的學習內容來填補這些差距。

下圖顯示架構的end-to-end流程。解決方案是以 Amazon Bedrock 中經過微調LLMs 為基礎。這些 LLMs會從 Amazon Neptune 中的醫療保健人才知識庫擷取資料。資料驅動演算法針對每個候選項目的最佳學習途徑提出建議。



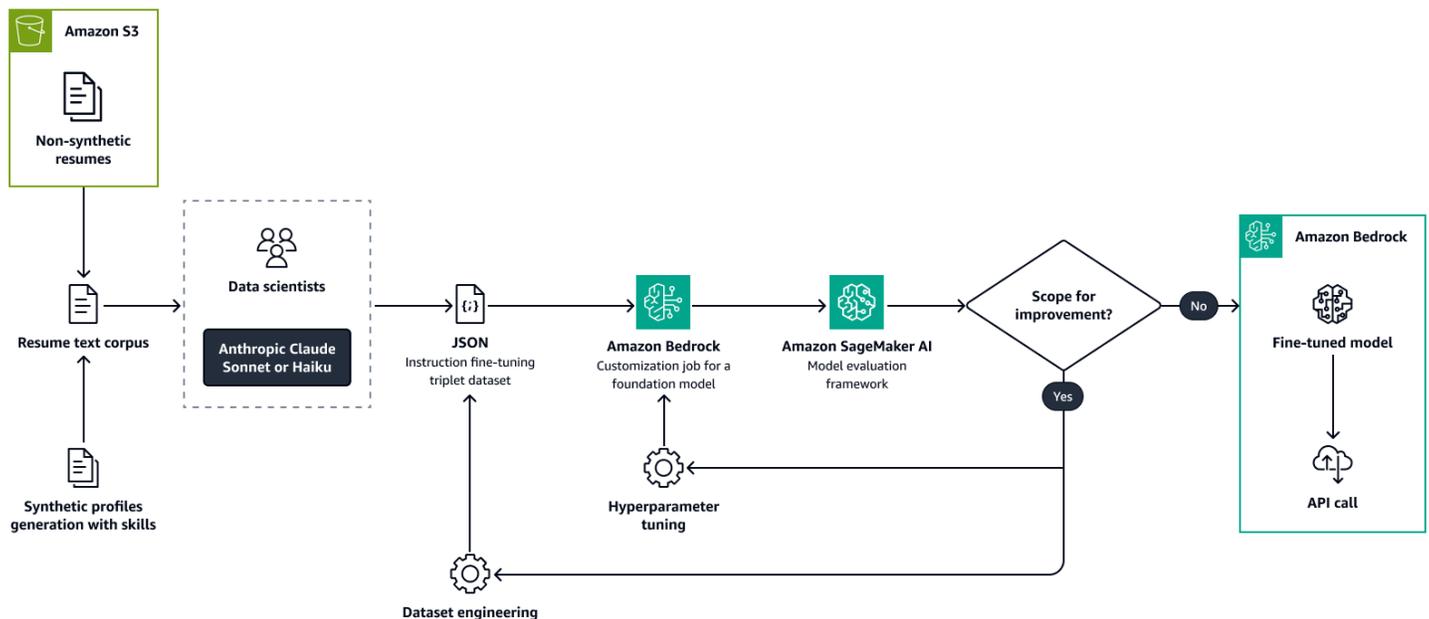
建置此解決方案包含下列步驟：

- [步驟 1：擷取人才資訊和建立技能描述檔](#)
- [步驟 2：從知識圖表探索role-to-skill的相關性](#)
- [步驟 3：識別技能差距並建議訓練](#)

## 步驟 1：擷取人才資訊和建立技能描述檔

首先，使用自訂資料集微調 Amazon Bedrock 中的大型語言模型，例如 Llama 2。這會針對使用案例調整 LLM。在訓練期間，您準確且一致地從候選履歷或類似人才設定檔中擷取關鍵人才屬性。這些人才屬性包括技能、目前角色標題、具有日期範圍的體驗標題、教育和認證。如需詳細資訊，請參閱《Amazon Bedrock 文件》中的[自訂模型，以改善其使用案例的效能](#)。

下圖顯示使用 Amazon Bedrock 微調恢復剖析模型的程序。實際和合成建立的恢復都會傳遞至 LLM，以擷取金鑰資訊。一組資料科學家會根據原始原始原始文字驗證擷取的資訊。擷取的資訊接著會使用 [chain-of-thought](#) 提示和原始文字來串連，以衍生訓練資料集進行微調。然後，此資料集會傳遞至 Amazon Bedrock 自訂任務，該任務會微調模型。Amazon SageMaker AI 批次任務會執行模型評估架構，以評估微調後的模型。如果模型需要改進，任務會使用更多資料或不同的超參數再次執行。評估符合標準後，您可以透過 Amazon Bedrock 佈建的輸送量託管自訂模型。



## 步驟 2：從知識圖表探索role-to-skill的相關性

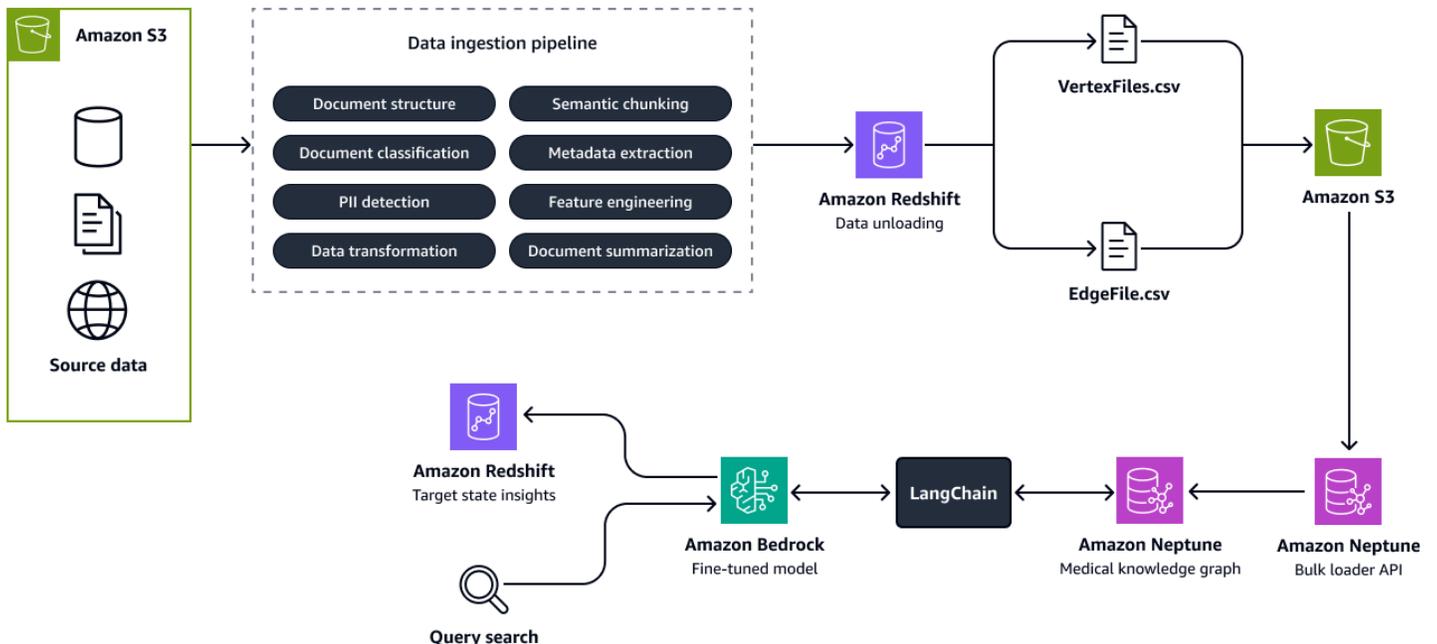
接下來，您將建立知識圖表，其中封裝組織和醫療保健產業中其他組織的技能和角色分類。這個豐富的知識庫來自 [Amazon Redshift](#) 中的彙總人才和組織資料。您可以從各種人力市場資料提供者以及組織特定的結構化和非結構化資料來源收集人才資料，例如企業資源規劃 (ERP) 系統、人力資源資訊系統 (HRIS)、員工恢復、任務描述和人才架構文件。

在 [Amazon Neptune](#) 上建置知識圖表。節點代表技能和角色，邊緣代表它們之間的關係。將此圖形填入中繼資料，以包含組織名稱、產業、工作系列、技能類型、角色類型和產業標籤等詳細資訊。

接下來，您將開發圖形擷取增強生成 (Graph RAG) 應用程式。圖形 RAG 是一種從圖形資料庫擷取資料的 RAG 方法。以下是 Graph RAG 應用程式的元件：

- 與 Amazon Bedrock 中的 LLM 整合 – 應用程式使用 Amazon Bedrock 中的 LLM 進行自然語言理解和查詢產生。使用者可以使用自然語言與系統互動。這可讓非技術利益相關者存取。
- 協調和資訊擷取 – 使用 [LlamaIndex LangChain](#) 或協調器來促進 LLM 和 Neptune 知識圖表之間的整合。他們會管理將自然語言查詢轉換為 [openCypher](#) 查詢的程序。然後，他們會在知識圖表上執行查詢。使用提示詞工程來指示 LLM 建構 openCypher 查詢的最佳實務。這有助於最佳化查詢以擷取相關子圖，其中包含有關查詢角色和技能的所有相關實體和關係。
- 洞見產生 – Amazon Bedrock 中的 LLM 會處理擷取的圖形資料。它會針對查詢的角色和相關技能，產生有關目前狀態和專案未來狀態的詳細洞見。

下圖顯示從來源資料建置知識圖表的步驟。您可以將結構化和非結構化來源資料傳遞至資料擷取管道。管道會擷取資訊並將其轉換為與 Amazon Neptune 相容的 CSV 大量負載形式。大量載入器 API 會將存放在 Amazon S3 儲存貯體中的 CSV 檔案上傳至 Neptune 知識圖表。對於與人才未來狀態、相關角色或技能相關的使用者查詢，Amazon Bedrock 中的微調 LLM 會透過 LangChain 協調器與知識圖表互動。協調器會從知識圖表擷取相關內容，並將回應推送至 Amazon Redshift 中的洞察資料表。LangChain 協調器，例如 [GraphQAChain](#)，會將使用者的自然語言查詢轉換為 openCypher 查詢，以查詢知識圖表。Amazon Bedrock 微調模型會根據擷取的內容產生回應。



## 步驟 3：識別技能差距並建議訓練

在此步驟中，您可以準確計算醫療保健專業人員目前狀態與潛在未來狀態角色之間的鄰近性。若要這樣做，您可以透過比較個人的技能集與任務角色來執行技能親和性分析。在 [Amazon OpenSearch Service](#) 向量資料庫中，您會儲存技能分類資訊和技能中繼資料，例如技能描述、技能類型和技能叢集。使用 Amazon Bedrock 內嵌模型，例如 [Amazon Titan Text Embeddings 模型](#)，將已識別的關鍵技能內嵌至向量。透過向量搜尋，您可以擷取目前狀態技能和目標狀態技能的描述，並執行本體分析。此分析提供目前和目標狀態技能對之間的接近分數。對於每對，您可以使用計算的內科分數來識別技能親和性中的差距。然後，您會建議提升技能的最佳路徑，候選者可以在角色轉換期間考慮這些路徑。

對於每個角色，建議用於提升技能或重新技能的正確學習內容涉及一種系統化方法，從建立完整的學習內容目錄開始。此目錄存放在 Amazon Redshift 資料庫中，會彙總來自各種提供者的內容，並包含中繼資料，例如內容持續時間、難度等級和學習模式。下一個步驟是擷取每個內容提供的關鍵技能，然後將其對應至目標角色所需的個別技能。您可以透過技能鄰近性分析來分析內容所提供的涵蓋範圍，藉此達成此映射。此分析會評估內容所教導的技能與角色所需技能的一致性。中繼資料在為每個技能選擇最適當的內容時扮演關鍵角色，確保學習者收到符合其學習需求的量身打造建議。在 LLMs 從內容中繼資料中擷取技能、執行功能工程，以及驗證內容建議。這可改善提升技能或重新技能程序中的準確性和相關性。

## 符合 AWS Well-Architected 架構

解決方案符合 [AWS Well-Architected Framework](#) 的所有六個支柱：

- 卓越營運 – 模組化的自動化管道可增強卓越營運。管道的關鍵元件會解耦和自動化，讓模型更新更快速且更容易監控。此外，自動化訓練管道支援更快速的微調模型版本。
- 安全性 – 此解決方案會處理敏感和個人身分識別資訊 (PII)，例如履歷中的資料和人才設定檔。在 [AWS Identity and Access Management \(IAM\)](#) 中，實作精細存取控制政策，並確保只有獲得授權的人員才能存取此資料。
- 可靠性 – 解決方案使用 AWS 服務，例如 Neptune、Amazon Bedrock 和 OpenSearch Service，即使在高需求期間也能提供容錯能力、高可用性和不間斷的洞見存取。
- 效能效率 – Amazon Bedrock 和 OpenSearch Service 向量資料庫中的微調 LLMs 旨在快速準確地處理大型資料集，以提供及時、個人化的學習建議。
- 成本最佳化 – 此解決方案使用 RAG 方法，可減少持續預先訓練模型的需求。系統只會微調特定程序，例如從恢復和結構輸出擷取資訊，而不是重複微調整個模型。這可大幅節省成本。透過將資源密集型模型訓練的頻率和規模降至最低，以及使用 pay-per-use 的雲端服務，醫療保健組織可以最佳化其營運成本，同時維持高效能。

- 永續性 – 此解決方案使用可擴展的雲端原生服務，以動態方式配置運算資源。這可減少能源消耗和環境影響，同時仍支援大規模、資料密集的人才轉型計畫。

# 為醫療保健開發和協調生成式 AI 解決方案

若要建置本指南中的解決方案，您必須建置使用微調 LLMs RAG 架構，以提供增強型病患資料、臨床和診斷洞察，以及預測病患結果給醫療保健供應商。這需要整合多個 AWS 服務和工具，才能建立有凝聚力且高效率的工作流程。本節討論下列項目：

- [Amazon Q Developer](#) – 使用 Amazon Q Developer 在開發過程中解決工程問題和程式碼錯誤。
- [多擷取器 RAG 設計](#) – 設計和實作使用多個擷取器來擷取使用者問題的正確醫療內容的 RAG 解決方案。
- [ReAct 代理程式](#) – 實作結合推理與動態動作的代理程式。

## Amazon Q Developer

建置生成式 AI 解決方案時，可能很難建立 AI 代理器和連線金鑰服務。不過，[Amazon Q Developer](#) 透過提供進階生成式 AI 助理的存取權，協助資料科學家和 AI 工程師。Amazon Q 可以快速準確地解決使用者問題和程式碼錯誤，這可協助您最佳化 LLM 開發程序。Amazon Q 為建立使用 Amazon Bedrock 基礎模型的應用程式的開發人員提供了顯著的優勢。它可以簡化工作流程並增強程式碼品質。它可自動產生 Python 指令碼和基礎設施做為程式碼 (IaC) 組態，大幅減少開發時間和工作量。透過進階重構功能，Amazon Q 可以改善程式碼效能、識別安全漏洞，並確保開發人員遵守最佳實務。此外，它透過提供內容感知的建議和說明來促進初學者的學習和採用，使複雜的編碼任務更容易存取和提高效率。

## 多擷取器 RAG 設計

在生成式 AI 應用程式中，多擷取器 RAG 管道可以有效地從多個資料來源擷取資訊，以協助醫療保健供應商和臨床醫生回答醫療問題。此管道使用不同類型的擷取器，從不同的知識庫提取相關資料。每個擷取工具都專門從醫學研究和學術文字中擷取特定類型的資訊，例如病患歷史記錄、診斷見解、臨床筆記或內容。

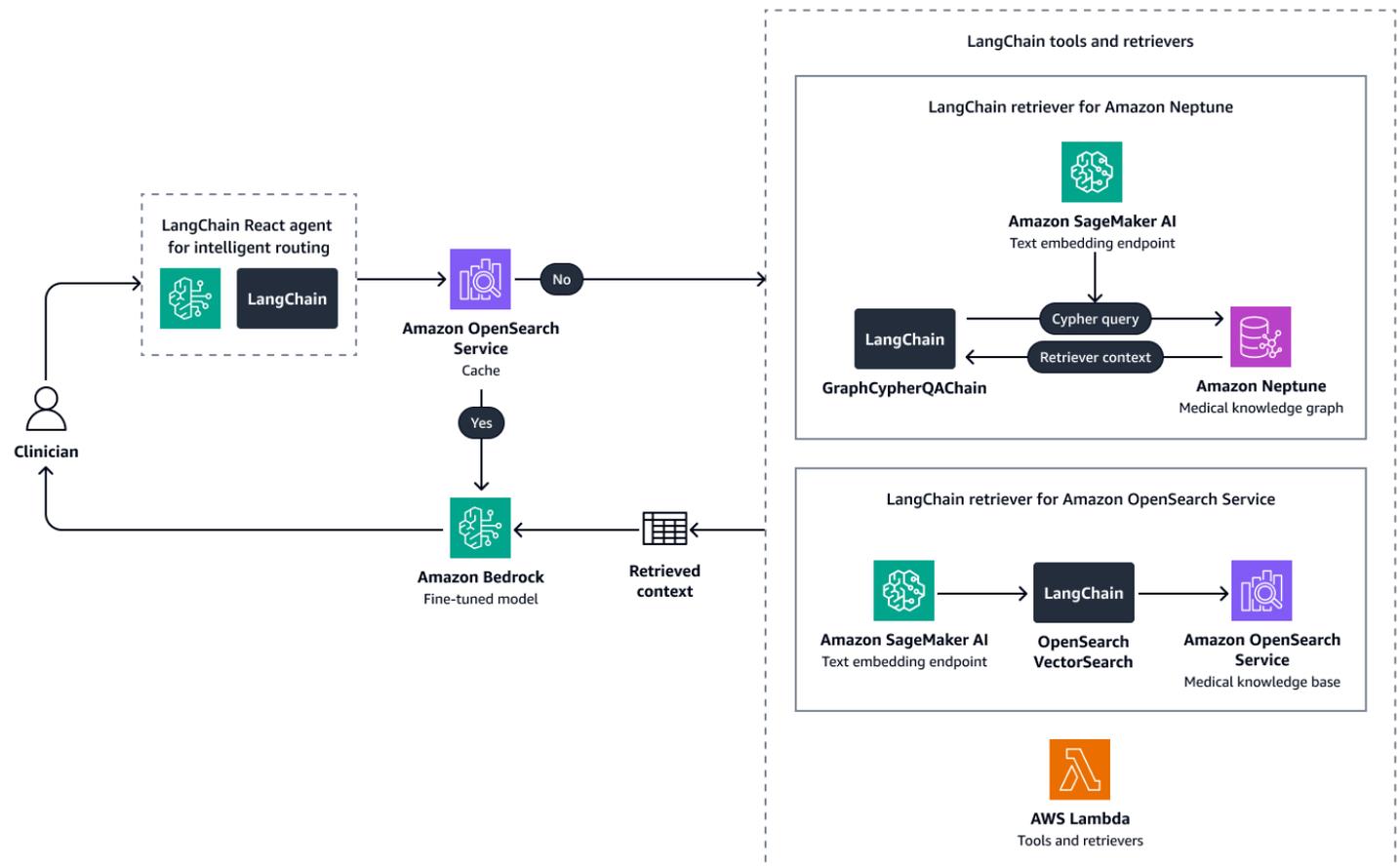
使用資料的性質和特定應用程式需求來判斷您的使用案例適用的正確後端知識庫。Amazon OpenSearch Service 向量資料庫非常適合大量非結構化或半結構化醫療資料，包括影像診斷評估摘要、放電摘要、臨床報告、醫學研究和學術文字內容。另一方面，圖形資料庫服務，例如 Amazon Neptune，非常適合需要深入探索實體之間時間關係的醫療保健使用案例，例如患者、患者歷程記錄、醫療保健供應商、藥物、症狀和治療。

此管道的關鍵元件是使用者查詢意圖預測。這可確保系統將查詢路由到正確的擷取器鏈。例如，如果臨床醫生詢問患者治療歷史記錄、症狀、與醫院的互動、重新入院的可能性或潛在的患者結果，則查詢意

圖預測模組會識別此意圖。它會將請求導向至擷取器鏈，該鏈可從醫學知識圖表擷取病患記錄或時間處理資料。或者，如果問題是關於疾病探索、特定診斷評估，或學術教科書中特定臨床程序的詳細資訊，則查詢會路由至可從 OpenSearch Service 向量資料庫擷取此資訊的擷取器鏈。您可以使用 [的工具呼叫](#) 功能 LangChain，將自訂工具繫結至可將使用者問題分類為預先定義意圖的 Amazon Bedrock LLM。

此多擷取器 RAG 系統包含專為管理特定知識庫存取權而設計的 LangChain 代理程式。您可以使用 LangChain 協調 Amazon Bedrock LLM、不同擷取器和工具之間的互動。LangChain 包含工具呼叫類別，可協助您建立自訂工具，例如意圖分類器、Neptune 的擷取器、OpenSearch Service 的擷取器，或任何其他可以開發的工具，以結構化格式分類使用者意圖和從特定知識庫存取資料。然後，將這些工具提供給類別，以建立理由和動作 (ReAct) 代理程式。ReAct 代理程式會處理使用者問題、規劃循序步驟來回答問題，然後反覆執行可用的工具，並處理工具回應以最終回答使用者查詢。

下圖顯示多擷取器 RAG 系統如何有效擷取知識和智慧型查詢解決方案。LangChain ReAct 代理程式會分析使用者的意圖、制定執行的結構化計畫，並選取最相關的擷取工具。系統會查詢先前的問題快取，並根據關鍵屬性檢查類似查詢，例如患者 ID、醫療情況和就診日期。如果找到高度相似的問題，則會直接擷取對應的答案。否則，代理程式會執行適當的擷取器。對於擷取以患者為中心的資訊，例如治療歷史記錄、症狀、醫院互動或重新入院可能性，系統會使用圖形擷取器。對於診斷評估、臨床程序和結構化醫療問題清單，代理程式會採用向量資料庫擷取器。在需要結合兩個資料存放區內容知識的情況下，為了產生全面的回應，系統會使用混合擷取策略來整合知識圖表和向量資料庫的結果。



## ReAct 代理程式

原因和動作 (ReAct) 代理程式是專為多面向 RAG 應用程式所設計。這些代理程式提供推理和動態動作的強大組合，尤其是涉及 step-by-step 邏輯資訊擷取工作流程的複雜應用程式。如需詳細資訊，請參閱 [ReAct : Synergizing Reasoning and Acting in Language Models](#)。

在醫療和醫療保健環境中，臨床醫生或醫生的查詢通常是多面向的。例如，臨床醫生可能會問：「對類似的有高血壓和第 2 類糖尿病的患者提供了哪些治療？」識別使用者意圖後，也就是擷取高血壓和第 2 類糖尿病的處理方式，AI 代理器需要將此查詢分割為子任務，然後選擇最有效的擷取策略。在此情況下，AI 代理器應識別最相關的節點（例如病患年齡、性別、條件、處理方式和藥物），然後查詢這些實體及其屬性和關係的圖表。ReAct 代理程式非常有用，因為它們結合 LLM 的推理（邏輯推論）功能與動作（查詢或與外部資源或知識庫互動）。

若要回答使用者查詢：「針對相似的糖尿病患者提供哪些處理方式？」，下列範例說明 ReAct 代理程式的運作方式：

1. 客服人員推理 – ReAct 客服人員推斷問題涉及擷取有關條件（糖尿病和高血壓）的資訊。它考慮患者年齡、治療、藥物和要分析的期間。

2. 客服人員動作 – 客服人員使用 openCypher 來查詢知識圖表，了解第 2 類糖尿病和高血壓特有的處理方式。它也會擷取所管理藥物、醫院就診日期、藥物的副作用、已知的患者結果，以及類似患者（例如性別和年齡相同的患者）的交叉參考資料。
3. 客服人員觀察 – 從知識圖表中，客服人員擷取最近六個月的表格式資料，這些資料是關於提供給同時有高血壓和第 2 類糖尿病的患者。
4. 代理程式推理 – 若要從擷取的記錄對結果進行排名，代理程式會識別重要屬性，例如延遲、藥物的副作用或已知的患者結果。
5. 代理程式動作 – 代理程式會根據已識別的屬性和透過系統提示授予的預先定義邏輯來重新排序記錄。
6. 回應產生 – Amazon Bedrock 中的 LLM 會根據 ReAct 代理程式準備的內容產生回應。

# 評估醫療保健的生成式 AI 解決方案

評估您建置的醫療保健 AI 解決方案對於確保其在實際醫療環境中有效、可靠且可擴展至關重要。使用系統性方法來評估解決方案每個元件的效能。以下是您可以用來評估解決方案的方法和指標摘要。

## 主題

- [評估資訊的擷取](#)
- [使用多個擷取器評估 RAG 解決方案](#)
- [使用 LLM 評估解決方案](#)

## 評估資訊的擷取

評估資訊擷取解決方案的效能，例如[智慧型繼續剖析器](#)和[自訂實體擷取器](#)。您可以使用測試資料集來測量這些解決方案回應的一致性。如果您沒有涵蓋多樣化醫療保健人才設定檔和患者醫療記錄的資料集，您可以使用 LLM 的推理功能來建立自訂測試資料集。例如，您可以使用模型等大型參數模型 Anthropic Claude 來產生測試資料集。

以下是可用於評估資訊擷取模型的三個關鍵指標：

- 準確性和完整性 – 這些指標會評估輸出擷取 Ground Truth 資料中正確和完整資訊的程度。這包括檢查擷取資訊的正確性，以及擷取資訊中是否存在所有相關詳細資訊。
- 相似性與相關性 – 這些指標會評估輸出與基本事實資料 (相似性) 之間的語意、結構和內容相似性，以及輸出符合並處理基本事實資料的內容、內容和意圖 (相關性) 的程度。
- 調整的召回或擷取速率 – 這些速率會以經驗判斷模型正確識別地面真實資料中的目前值數量。速率應包含對模型擷取之所有 false 值的懲罰。
- 精確度分數 – 精確度分數可協助您判斷與真陽性相比，預測中存在多少誤報。例如，您可以使用精確度指標來測量擷取技能熟練度的正確性。

## 使用多個擷取器評估 RAG 解決方案

若要評估系統擷取相關資訊的程度，以及如何使用該資訊產生準確且情境適當的回應，您可以使用下列指標：

- 回應相關性 – 測量使用擷取內容產生的回應與原始查詢的相關性。

- 內容精確度 – 在總擷取結果中，評估與查詢相關的擷取文件或程式碼片段的比例。更高的內容精確度表示擷取機制可有效選取相關資訊。
- 誠實 – 評估產生的回應在擷取內容中反映資訊的準確度。換句話說，請測量回應對來源資訊是否仍為 true。

## 使用 LLM 評估解決方案

您可以使用稱為 LLM-as-a-judge 的技術來評估生成式 AI 解決方案的文字回應。它涉及使用 LLMs 來評估和評估模型輸出的效能。此技術使用 Amazon Bedrock 的功能來提供對各種屬性的判斷，例如回應品質、一致性、遵循性、準確性和對人類偏好或基本事實資料的完整性。您可以使用 [chain-of-thought\(CoT\)](#) 和 [少量擷取](#) 提示技術進行全面評估。提示會指示 LLM 使用評分 Rubric 來評估產生的回應，而提示中的幾個樣本示範實際的評估程序。提示也包含 LLM 評估器要遵循的準則。例如，您可以考慮使用下列一或多個使用 LLM 來判斷產生的回應的評估技術：

- 配對比較 – 為 LLM 評估者提供一個醫療問題，以及由您建立之 RAG 系統的不同反覆版本所產生的多個回應。提示 LLM 評估者根據回應品質、一致性和對原始問題的遵守來判斷最佳回應。
- 單一答案分級 – 此技術非常適合您需要評估分類準確性的使用案例，例如患者結果分類、患者行為分類、患者重新入院可能性和風險分類。使用 LLM 評估器來單獨分析個別分類或分類，並評估它針對基本事實資料提供的原因。
- 參考引導分級 – 為 LLM 評估者提供一系列需要描述性回答的醫療問題。建立這些問題的範例回應，例如參考答案或理想的回應。提示 LLM 評估器將 LLM 產生的回應與參考答案或理想的回應進行比較，並提示 LLM 評估器對產生的回應進行準確度、完整性、相似性、相關性或其他屬性的分級。此技術可協助您評估產生的回應是否符合明確定義的標準或示例性答案。

# 資源

## AWS 文件

- [Amazon Bedrock 文件](#)
- [Amazon Neptune 文件](#)
- [Amazon OpenSearch Service 文件](#)
- [套用 AWS Well-Architected Framework for Amazon Neptune](#) (AWS 方案指引 )
- [Amazon OpenSearch Service 的操作最佳實務](#) (OpenSearch Service 文件 )
- [將 Amazon Comprehend Medical 和 LLMs 用於醫療保健和生命科學](#) (AWS 方案指引 )

## AWS 部落格文章

- [使用 Amazon Bedrock 中提供的新 Amazon Titan Text Premier 模型建置 RAG 和代理程式型生成式 AI 應用程式](#)
- [使用 Amazon Neptune 從資料倉儲建置知識圖表來補充商業智慧](#)
- [使用知識圖表透過 Amazon Bedrock 和 Amazon Neptune 建置 GraphRAG 應用程式](#)

## 其他資源

- [將擷取增強的生成與海洋中的大型語言模型整合：推動實務應用程式](#) (PubMed Central , 國家醫學圖書館 )
- [簡介 LangChain](#)(LangChain 文件 )

## 貢獻者

### 編寫

- Nitu Nivedita , Accenture 資料與 AI 人工智慧主管常務董事
- Manoj Appully , Cadiem 創辦人兼技術長
- Conor Folan , Accenture 資料與 AI 顧問
- Deepak Krishna AR , Accenture 資料與 AI 顧問
- Almore Cato , Accenture 資料與 AI 經理
- Soonam Kurian , 首席解決方案架構師 AWS

### 檢閱

- Sally Lin | Accenture 資料與 AI 資料科學資深經理
- TerryHuang | Accenture 資料與 AI 資料科學經理
- William Lorenz , 合作夥伴解決方案架構師 , AWS

### 技術寫入

- 資深技術作者 , Lilly AbouHarb AWS

# 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
<a href="#">初次出版</a>	—	2025 年 3 月 14 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：將您的現場部署 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

### 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、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

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、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 (IAM) 文件》中的[ABAC for AWS](#)。

## 授權資料來源

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。

## 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)

一種工具，可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。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，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。有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，讓使用者能夠快速存取他們通常無權存取 AWS 帳戶的。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改的中繼資料的程序。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。

### 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

這些階段由 Stephen Orban 於部落格文章 [The Journey Toward Cloud-First 和 Enterprise Strategy 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 遷移策略相關的詳細資訊，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 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，以管理組織的帳戶和管理該服務的許可。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)

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 延伸了原本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

真實世界系統的虛擬呈現，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中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

一種運算程序，可將人類可讀取的純文字資料轉換為加密文字。

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

## 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 工作流程會被視為舊版，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、Amazon GuardDuty、Amazon Inspector、AWS Trusted Advisor 和自訂 AWS Lambda 檢查來實作。

# 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。

## 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

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

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。

## IloT

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 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 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、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](#)。

## 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

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。

## 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)

一種 AWS 計畫，提供諮詢支援、訓練和服務，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，請參閱此詞彙表中的 [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[不可變基礎設施](#)做為最佳實務。

## 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[開放程序通訊 - 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。在製造業中，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 儲存貯體中的所有伺服器端加密 AWS KMS (SSE-KMS) 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)

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。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。

## 依設計的隱私權

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)

管理產品整個生命週期的資料和程序，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](#) 提示的輸出做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 [擷取增強產生](#)。

## 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，定義所有涉及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 Rs](#)。

## 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 Management Console 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](#)。

## 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，例如密碼或使用者登入資料。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 秘密中的什麼內容？](#)。

## 依設計的安全性

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

由接收資料的 AWS 服務 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](#)。

## SIEM

請參閱[安全資訊和事件管理系統](#)。

## 單一故障點 (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。

## 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。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 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)

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。