AMS 進階應用程式部署選項

AMS 進階應用程式開發人員指南

版本 September 13, 2024

AMS 進階應用程式開發人員指南: AMS 進階應用程式部署選項

Table of Contents

應用程式加入

歡迎使用 AWS Managed Services (AMS) AMS 操作計劃。本文件的目的是描述在初始聯網和存取管理設定完成後，您將應用程式加入 AMS 時可以使用的各種方法，以及選擇這些方法時應考慮的問題。

本文件旨在供系統整合商和應用程式開發人員協助判斷和製作新 AMS 客戶的應用程式程序。

什麼是應用程式加入？

AMS 應用程式加入是指視需要將資源和應用程式部署到您的 AMS 基礎設施。在 AMS 平台上架構應用程式和基礎設施與在原生平台上架構非常類似 AWS。遵循 AWS 應用程式和基礎設施設計最佳實務，同時考慮 AMS 提供的功能，將產生在 AMS 環境中託管且功能強大且可操作的應用程式。

> **Note**
>
> - 美國東部 (維吉尼亞)
> - 美國西部 (加利佛尼亞北部)
> - 美國西部 (奧勒岡)
> - 美國東部 (俄亥俄)
> - 加拿大 (中部)
> - 南美洲 (聖保羅)
> - 歐洲 (愛爾蘭)
> - 歐洲 (法蘭克福)
> - 歐洲 (倫敦)
> - 歐洲西部 （ 巴黎 ）
> - 亞太區域 (孟買)
> - 亞太區域 (首爾)
> - 亞太區域 (新加坡)
> - 亞太區域 (雪梨)
> - 亞太區域 (東京)
>
> 新區域會經常新增。若要進一步了解，請參閱 AWS 區域 和可用區域。

我們做什麼、我們不做什麼

AMS 為您提供了部署 AWS 基礎設施的標準化方法，並提供必要的持續營運管理。如需角色、責任和支援服務的完整描述，請參閱服務描述。

> **Note**
>
> 若要請求 AMS 提供額外的 AWS 服務，請提交服務請求。如需詳細資訊，請參閱提出服務請求。

- 我們做什麼：

  完成加入後，AMS 環境即可接收變更 (RFCs)、事件和服務請求。您與 AMS 服務的互動圍繞應用程式堆疊的生命週期進行。新堆疊是從預先設定的範本清單排序，啟動到特定的虛擬私有雲端 (VPC) 子網路，在操作生命週期內透過請求變更 (RFCs) 進行修改，並全年無休監控事件和事件。

  AMS 會監控和維護作用中的應用程式堆疊，包括修補，除非需要變更或停用堆疊，否則堆疊在堆疊生命週期內不需要進一步的動作。AMS 偵測到會影響堆疊運作狀態和功能的事件會產生通知，且可能需要或不需要您的動作來解決或驗證。方法問題和其他查詢可以透過提交服務請求提出。

  此外，AMS 可讓您啟用非由 AMS 管理的相容 AWS 服務。如需 AWS-AMS 相容服務的資訊，請參閱自助式佈建模式。

- 我們不執行的動作：

  雖然 AMS 透過提供許多手動和自動化選項來簡化應用程式部署，但您必須負責應用程式的開發、測試、更新和管理。AMS 為會影響應用程式的基礎設施問題提供故障診斷協助，但 AMS 無法存取或驗證您的應用程式組態。

AMS Amazon Machine Image AMIs)

AMS 每個月都會為 AMS 支援的作業系統產生更新的 Amazon Machine Image (AMIs)。此外，AMS 也會根據 CIS 層級 1 基準，為 AMS 支援的作業系統子集產生安全性增強映像 (AMIs)。若要了解哪些作業系統有可用的安全性增強映像，請參閱 AMS 安全使用者指南，該指南可透過 AWS Artifact -> 報告頁面（在左側導覽窗格中尋找報告選項）篩選 AWS Managed Services。若要存取 AWS Artifact， 可以聯絡您的 CSDM 以取得指示，或前往 AWS Artifact 入門。

若要在發行新的 AMS AMIs 時接收提醒，您可以訂閱名為「AMS AMI」的 Amazon Simple Notification Service (Amazon SNS) 通知主題。如需詳細資訊，請參閱使用 SNS 的 AMS AMI 通知。

AMS AMI 命名慣例為：`customer-ams-<operating system>-<release date> - <version>`。（例如，`customer-ams-rhel6-2018.11-3`)

僅使用開頭為 的 AMS AMIs`customer`。

AMS 建議一律使用最新的 AMI。您可以透過以下方式找到最新的 AMIs：

- 在 AMS 主控台的 AMIs頁面上查看 。
- 檢視可從 CSDM 或此 ZIP 檔案取得的最新 AMS AMI CSV 檔案：AMS 11.2024 AMI 內容和 ZIP 中的 CSV 檔案。

  如需過去的 AMI ZIP 檔案，請參閱 文件歷史記錄。
- 執行此 AMS SKMS命令（需要 AMS SKMS SDK)：

```
aws amsskms list-amis --vpc-id VPC_ID --query "Amis.sort_by(@,&Name)[?
 starts_with(Name,'customer')].[Name,AmiId,CreationTime]" --output table
```

依作業系統 (OS) 新增至 AWS AMIs 內容

- Linux AMIs：
  - AWS CLI 工具
  - NTP
  - Trend Micro Endpoint Protection Service 代理程式
  - 程式碼部署
  - PBIS/超越信任 AD 橋接器
  - SSM 代理程式
  - 重要修補程式的百勝升級
  - AMS 自訂指令碼/管理軟體（控制開機、AD 聯結、監控、安全性和記錄）
- Windows Server AMIs：
  - Microsoft .NET Framework 4.5
  - PowerShell 5.1
  - AWS 適用於 Windows PowerShell 的工具

- AMS PowerShell 模組控制開機、AD 聯結、監控、安全性和記錄
- Trend Micro Endpoint Protection Service 代理程式
- SSM 代理程式
- CloudWatch 代理程式
- EC2Config 服務 (透過 Windows Server 2012 R2)
- EC2Launch (Windows Server 2016 和 Windows Server 2019)
- EC2LaunchV2 (Windows Server 2022 及更新版本)

Linux 型 AMIs：

- Amazon Linux 2023（最新次要版本）（不支援最小 AMI)
- Amazon Linux 2（最新次要版本）
- Amazon Linux 2 (ARM64)
- Red Hat Enterprise 7（最新次要版本）
- Red Hat Enterprise 8（最新次要版本）
- Red Hat Enterprise 9（最新次要版本）
- SUSE Linux Enterprise Server 15 SP6
- Ubuntu Linux 18.04
- Ubuntu Linux 20.04
- Ubuntu Linux 22.04
- Ubuntu Linux 24.04
- Amazon Linux：如需產品概觀、定價資訊、用量資訊和支援資訊，請參閱 Amazon Linux AMI (HVM / 64 位元) 和 Amazon Linux 2。

  如需詳細資訊，請參閱 Amazon Linux 2 常見問答集。
- RedHat Enterprise Linux (RHEL)：如需產品概觀、定價資訊、用量資訊和支援資訊，請參閱 Red Hat Enterprise Linux (RHEL) 7 (HVM)。
- Ubuntu Linux 18.04：如需產品概觀、定價資訊、用量資訊和支援資訊，請參閱 Ubuntu 18.04 LTS - Bionic。
- SUSE Linux Enterprise Server for SAP 應用程式 15 SP6：
  - 每個帳戶執行下列步驟一次：
    1. 導覽至 AWS Marketplace。

2. 搜尋 SUSE 15 SAP 產品。
3. 選擇繼續以訂閱。
4. 選擇接受詞彙。

- 每次您需要啟動新的 SUSE Linux Enterprise Server for SAP Applications 15 SP6 執行個體時，請完成下列步驟： SP6
  1. 請注意訂閱的 SUSE Linux Enterprise Server for SAP Applications 15 AMI 的 AMI ID。
  2. 建立部署 | 進階堆疊元件 | EC2 堆疊 | 建立變更類型 ct-14027q0sjyt1h RFC。以您訂閱的 AWS Marketplace AMI ID 取代 *`InstanceAmiId`*。

Windows 型 AMIs：

Microsoft Windows Server (2016、2019 和 2022)，以最新的 Windows AMIs 為基礎。

如需建立 AMIs的範例，請參閱建立 AMI。

離職 AMS AMIs：

AMS 不會在離職期間取消與您共用任何 AMIs，以避免對您的任何停用情況造成影響。如果您想要從您的帳戶中移除 AMS AMIs，您可以使用 `cancel-image-launch-permission` API 來隱藏特定 AMIs。例如，您可以使用以下指令碼來隱藏先前與您的帳戶共用的所有 AMS AMIs：

```
for ami in $(aws ec2 describe-images --executable-users self --owners 027415890775 --
query 'Images[].ImageId' --output text) ;
    do
    aws ec2 cancel-image-launch-permission --image-id $ami ;
    done
```

您必須安裝 AWS CLI v2，指令碼才能在沒有錯誤的情況下執行。如需 AWS CLI 安裝步驟，請參閱安裝或更新最新版本的 AWS CLI。如需 `cancel-image-launch-permission`命令的詳細資訊，請參閱 `cancel-image-launch-permission`。

安全性增強AMIs

AMS 為 AMIs支援的作業系統子集提供以 CIS 第 1 級基準為基礎的安全性增強映像 (AMI)。若要尋找哪些作業系統有可用的安全增強型映像，請參閱 AWS Managed Services (AMS) 客戶安全指南。若要存取本指南，請開啟 AWS Artifact，選取左側導覽窗格中的報告，然後篩選 AWS Managed Services。如需如何存取的指示 AWS Artifact，請聯絡您的 CSDM 或參閱 入門 AWS Artifact以取得詳細資訊。

AMS 金鑰術語

- AMS 進階：AMS 進階文件的「服務描述」一節中所述的服務。請參閱服務描述。
- AMS 進階帳戶：始終符合 AMS 進階加入要求中所有需求的 AWS 帳戶。如需 AMS Advanced 優點、案例研究以及聯絡銷售人員的資訊，請參閱 AWS Managed Services。
- AMS Accelerate Accounts： AWS 一直符合 AMS Accelerate Onboarding Requirements 中所有需求的帳戶。請參閱 AMS Accelerate 入門。
- AWS Managed Services：AMS 和 或 AMS Accelerate。
- AWS Managed Services 帳戶：AMS 帳戶和 或 AMS Accelerate 帳戶。
- 關鍵建議： AWS 透過服務請求發出的建議，通知您需要採取動作，以防止 資源或 的潛在風險或中斷 AWS 服務。如果您決定在指定日期之前不遵循關鍵建議，則需自行負責您的決定所造成的任何傷害。
- 客戶請求的組態：任何軟體、服務或其他未識別的組態：
  - 加速：支援的組態或 AMS 加速；服務描述。
  - AMS 進階：支援的組態或 AMS 進階；服務描述。
- 事件通訊：AMS 會與您通訊事件，或者您透過在 AMS Accelerate 支援中心和 AMS 主控台中建立的事件向 AMS 請求事件。AMS 加速主控台提供儀表板上的事件和服務請求摘要，以及支援中心的連結以取得詳細資訊。
- 受管環境：AMS 進階帳戶和 或由 AMS 操作的 AMS Accelerate 帳戶。

  對於 AMS Advanced，這些包括多帳戶登陸區域 (MALZ) 和單一帳戶登陸區域 (SALZ) 帳戶。
- 帳單開始日期： 的下一個工作日 AWS 會收到您在 AWS Managed Services 加入電子郵件中請求的資訊。AWS Managed Services 加入電子郵件是指 傳送給 AWS 您的電子郵件，以收集在帳戶中啟用 AWS Managed Services 所需的資訊。

  對於您後續註冊的帳戶，帳單開始日期是 AWS Managed Services 為註冊帳戶傳送 AWS Managed Services 啟用通知後的第二天。AWS Managed Services 啟用通知會在下列情況發生：

  1. 您授予相容 AWS 帳戶的存取權，並將其交給 AWS Managed Services。
  2. AWS Managed Services 設計並建置 AWS Managed Services 帳戶。
- 服務終止：您可以透過服務請求 AWS 提供至少 AWS Managed Services 30 天的通知，以終止所有 AWS Managed Services 帳戶的 AWS Managed Services 或指定 AWS Managed Services 帳戶的 AWS Managed Services。在服務終止日期，：

  1. AWS 視適用情況，將所有 AWS Managed Services 帳戶或指定 AWS Managed Services 帳戶的控制項交給您，或

2. 適用時，各方會移除授予所有 AWS Managed Services 帳戶或指定 AWS Managed Services 帳戶 AWS 存取權 AWS Identity and Access Management 的角色。

- 服務終止日期：服務終止日期是 30 天必要終止通知期間結束後日曆月的最後一天。如果必要的終止通知期間結束在日曆月的第 20 天之後，則服務終止日期是下一個日曆月的最後一天。以下是終止日期的範例案例。
  - 如果終止通知是在 4 月 12 日提供，則 30 天的通知會在 5 月 12 日結束。服務終止日期為 5 月 31 日。
  - 如果在 4 月 29 日提供終止通知，則 30 天的通知將於 5 月 29 日結束。服務終止日期為 6 月 30 日。
- 您可用 AWS Managed Services：makes 的佈建，您可以從服務開始日期開始，存取和使用每個 AWS Managed Services 帳戶的 AWS Managed Services。 AWS
- 指定 AWS Managed Services 帳戶的終止：您可以透過服務請求 (「AWS Managed Services 帳戶終止請求」) 提供 AWS 通知，以基於任何原因終止指定 AWS Managed Services 帳戶的 AWS Managed Services。

事件管理術語：

- 事件：您的 AMS 環境中的變更。
- 警示：每當支援的 事件 AWS 服務 超過閾值並觸發警示時，就會建立警示並傳送通知到您的聯絡人清單。此外，事件會在您的事件清單中建立。
- 事件：您的 AMS 環境或 AWS Managed Services意外中斷或效能降低，導致 AWS Managed Services或您回報的影響。
- 問題：一或多個事件的共用基礎根本原因。
- 事件解決或事件解決：
  - AMS 已將與該事件相關的所有無法使用 AMS 服務或資源還原為可用狀態，或
  - AMS 已判斷無法使用的堆疊或資源無法還原至可用狀態，或
  - AMS 已啟動您授權的基礎設施還原。
- 事件回應時間：建立事件以及 AMS 透過主控台、電子郵件、服務中心或電話提供初始回應之間的時間差異。
- 事件解決時間：AMS 或您建立事件與事件解決之間的時間差異。
- 事件優先順序：AMS 或您如何將事件的優先順序設定為低、中或高。
  - 低：AMS 服務的非關鍵問題。
  - 中：您受管環境中的 AWS 服務可用，但未如預期般執行 （根據適用的服務描述）。

- 高：(1) AMS 主控台或受管環境中的一或多個 AMS APIs 無法使用；或 (2) 受管環境中的一或多個 AMS 堆疊或資源無法使用，且無法使用可防止應用程式執行其功能。

  AMS 可能會根據上述準則重新分類事件。
- 基礎設施還原：根據受影響的堆疊範本重新部署現有堆疊，並根據最後一個已知還原點啟動資料還原，除非您另有指定，否則無法解決事件。

基礎設施術語：

- 受管生產環境：客戶生產應用程式所在的客戶帳戶。
- 受管非生產環境：僅包含非生產應用程式的客戶帳戶，例如用於開發和測試的應用程式。
- AMS 堆疊：由 AMS 以單一單位管理的一或多個 AWS 資源群組。
- 不可變基礎設施：Amazon EC2 Auto Scaling 群組 (ASGs) 典型的基礎設施維護模型，其中針對每個部署替換了更新的基礎設施元件（在 AMI AWS中），而不是就地更新。不可變基礎設施的優點是所有元件都會保持同步狀態，因為它們一律從相同的基礎產生。抗擾性與建置 AMI 的任何工具或工作流程無關。
- 互斥基礎設施：典型的基礎設施維護模型，適用於非 Amazon EC2 Auto Scaling 群組且包含單一執行個體或僅包含少數執行個體的堆疊。此模型最緊密地代表傳統的硬體型系統部署，其中系統會在生命週期開始時部署，然後隨著時間的推移，更新會分層到該系統上。系統的任何更新都會個別套用到執行個體，並可能因應用程式或系統重新啟動而導致系統停機（取決於堆疊組態）。
- 安全群組：執行個體的虛擬防火牆，用於控制傳入和傳出流量。安全群組會在執行個體層級執行，而非子網路層級。因此，VPC 中子網路中的每個執行個體可以指派不同的安全群組集。
- 服務水準協議 (SLAs)：與您簽訂 AMS 合約的一部分，定義預期的服務水準。
- SLA 無法使用和無法使用：
  - 您提交的 API 請求會導致錯誤。
  - 您提交的主控台請求導致 5xx HTTP 回應（伺服器無法執行請求）。
  - 在 AMS 受管基礎設施中構成堆疊或資源的任何 AWS 服務 方案都處於「服務中斷」狀態，如服務運作狀態儀表板所示。
  - 在判斷服務點數的資格時，不會考慮直接或間接由 AMS 排除造成的無法使用。除非符合無法使用的條件，否則服務會被視為可用。
- 服務水準目標 (SLOs)：與您簽訂的一部分 AMS 合約，可定義 AMS 服務的特定服務目標。

修補詞彙：

- 強制性修補程式：重大安全性更新，以解決可能危及環境或帳戶安全狀態的問題。「重大安全性更新」是 AMS 支援之作業系統廠商評定為「重大」的安全性更新。
- 發佈的修補程式與發佈的修補程式：修補程式通常按排程發佈和發佈。緊急修補程式會在發現需要修補程式時宣佈，通常在修補程式發佈後不久宣佈。
- 修補程式附加元件：針對利用 AWS Systems Manager (SSM) 功能的 AMS 執行個體進行標籤型修補，讓您可以標記執行個體，並使用您設定的基準和視窗修補這些執行個體。
- 修補程式方法：
  - 就地修補：透過變更現有執行個體來完成的修補。
  - AMI 取代修補：透過變更現有 EC2 Auto Scaling 群組啟動組態的 AMI 參考參數來完成的修補。
- 修補程式提供者 (OS 廠商、第三方 ）：修補程式是由廠商或應用程式管理內文提供。
- 修補程式類型：
  - Critical Security Update (CSU)：受支援作業系統廠商評定為「Critical」的安全性更新。
  - 重要更新 (IU)：由受支援作業系統的廠商評定為「重要」或非安全性更新評定為「關鍵」的安全性更新。
  - 其他更新 (OU)：廠商對非 CSU 或 IU 之受支援作業系統的更新。
- 支援的修補程式：AMS 支援作業系統層級修補程式。廠商會釋出升級，以修正安全漏洞或其他錯誤，或改善效能。如需目前支援的OSs清單，請參閱支援組態。

安全術語：

- Detective Controls：由 AMS 建立或啟用的監控程式庫，可針對不符合安全、操作或客戶控制的組態持續監督客戶受管環境和工作負載，並透過通知擁有者、主動修改或終止資源來採取行動。

服務請求條款：

- 服務請求：您希望 AMS 代表您採取之動作的請求。
- 提醒通知：觸發 AMS 提醒時，AMS 發佈到您的服務請求清單頁面的通知。為您的帳戶設定的聯絡人也會透過設定的 方法 （ 例如電子郵件 ） 收到通知。如果您的執行個體/資源上有聯絡人標籤，並且已同意雲端服務交付管理員 (CSDM) 以標籤為基礎的通知，則標籤中的聯絡資訊 （ 金鑰值 ） 也會收到自動 AMS 提醒的通知。
- 服務通知：AMS 發佈至服務請求清單頁面的通知。

其他詞彙：

- AWS Managed Services介面：針對 AMS：AWS Managed Services進階主控台、AMS CM API 和支援 API。對於 AMS Accelerate： 支援 主控台和 支援 API。
- 客戶滿意度 (CSAT)：透過深入分析來通知 AMS CSAT，包括提供每個案例或通訊的案例對應率、每季調查等。
- DevOps：DevOps 是一種開發方法，在所有步驟中都強烈倡導自動化和監控。DevOps 的目標在於縮短開發週期、提高部署頻率，以及更可靠的版本，透過自動化的基礎，結合傳統上分開的開發和操作功能。當開發人員可以管理 操作，並且 操作通知開發時，問題和問題會更快地被發現和解決，而業務目標也更容易實現。
- ITIL：Information Technology Infrastructure Library （稱為 ITIL) 是一種 ITSM 架構，旨在標準化 IT 服務的生命週期。ITIL 分為五個階段，涵蓋 IT 服務生命週期：服務策略、服務設計、服務轉換、服務操作和服務改進。
- IT 服務管理 (ITSM)：一組符合 IT 服務需求的實務。
- 受管監控服務 (MMS)：AMS 會運作自己的監控系統 Managed Monitoring Service (MMS)，其會取用 AWS 運作狀態事件，並彙總 Amazon CloudWatch 資料和其他資料 AWS 服務，通知 AMS 運算子（線上全年無休） 透過 Amazon Simple Notification Service (Amazon SNS) 主題建立的任何警示。
- 命名空間：當您建立 IAM 政策或使用 Amazon Resource Name (ARNs) 時，您可以使用命名空間 AWS 服務 來識別 。您會在識別動作和資源時使用命名空間。

我的操作模型是什麼？

身為 AMS 客戶，您的組織決定分開應用程式和基礎設施操作，並使用 AMS 進行基礎設施操作。AMS 將與您的應用程式設計和開發團隊以及您的基礎設施設計團隊合作，以確保您的基礎設施操作順利執行。下圖說明此概念：

AMS 負責 AWS 基礎設施操作，而您的團隊負責您的應用程式操作。身為應用程式和基礎設施設計團隊，您必須了解部署到 AMS 基礎設施中生產環境後，誰將操作應用程式。本指南涵蓋與應用程式部署和維護相關的基礎設施設計常見方法。

AWS Managed Services 中的服務管理

主題

AMS 服務的運作方式。

AWS Managed Services 中的帳戶控管

本節涵蓋 AMS 帳戶控管。

您被指派為雲端服務交付經理 (CSDM)，該經理提供跨 AMS 的諮詢協助，並對受管環境的使用案例和技術架構有詳細的了解。CSDMs會視情況與帳戶管理員、技術帳戶管理員、AWS Managed Services 雲端架構師 (CAs) 和 AWS 解決方案架構師 (SAs) 合作，協助啟動新專案，並在整個軟體開發和操作程序中提供最佳實務建議。CSDM 是 AMS 的主要聯絡人。CSDM 的主要責任為：

- 與客戶組織和主持每月服務審查會議。
- 提供有關安全性、環境軟體更新和最佳化機會的詳細資訊。
- 擁護您的需求，包括 AMS 的功能請求。
- 回應並解決帳單和服務報告請求。
- 提供財務和容量最佳化建議的洞見。

AWS Managed Services 中的服務開始

服務起始：AWS Managed Services 帳戶的服務起始日期是第一個日曆月的第一天，之後 AWS 會通知您該 AWS Managed Services 帳戶加入要求中列出的活動已完成；前提是如果 AWS 在某個日曆月的 20 天後發出此類通知，則服務起始日期是該通知日期之後第二個日曆月的第一天。

服務起始

- R 代表負責方執行工作以達成任務。
- 我代表知情的一方，通常是在任務完成或可交付項目時收到進度通知的一方。

服務開始

步驟 #	步驟標題	描述	客戶	AMS
1.	客戶 AWS 帳戶移交	客戶建立新的 AWS 帳戶，並將其交給 AWS Managed Services	R	I
2.	AWS Managed Services 帳戶 - 設計	完成 AWS Managed Services 帳戶的設計	I	R
3.	AWS Managed Services 帳戶 - 組建	AWS Managed Services 帳戶是根據步驟 2 中的設計建置	I	R

客戶關係管理 (CRM)

AWS Managed Services (AMS) 提供客戶關係管理 (CRM) 程序，以確保與您建立和維護明確定義的關係。此關係的基礎是根據 AMS 對您業務需求的洞察。CRM 程序有助於準確且全面地了解：

- 您的業務需求以及如何滿足這些需求
- 您的功能和限制條件
- AMS 和您的不同責任與義務

CRM 程序可讓 AMS 使用一致的方法來為您提供服務，並對您與 AMS 的關係提供控管。CRM 程序包括：

- 識別您的關鍵利益相關者
- 建立控管團隊
- 與您舉行並記錄服務審查會議
- 使用呈報程序提供正式的服務投訴程序

- 實作和監控您的滿意度和意見回饋程序
- 管理您的合約

CRM 程序

CRM 程序包含下列活動：

- 識別和了解您的業務流程和需求。您與 AMS 的協議可識別您的利益相關者。
- 定義要提供的服務，以符合您的需求和要求。
- 在服務審查會議中與您開會，討論 AMS 服務範圍、SLA、合約和您的業務需求的任何變更。可能會與您舉行臨時會議，以討論績效、成就、問題和行動計劃。
- 使用我們在會議上提供的客戶滿意度調查和意見回饋來監控您的滿意度。
- 報告每月內部測量效能報告的效能。
- 與您一起檢閱服務，以判斷改善的機會。這包括經常與您通訊有關所提供 AMS 服務的等級和品質。

CRM 會議

AMS 雲端服務交付管理員 (CSDMs) 會定期與您進行會議，以討論服務追蹤（操作、安全性和產品創新）和執行追蹤 (SLA 報告、滿意度測量和業務需求的變更）。

會議	用途	Mode	參與者
每週狀態檢閱（選用）	未解決的問題或事件、修補、安全事件、問題記錄 12 週營運趨勢 (+/- 6) 應用程式運算子問題 週末排程	現場客戶location/Telecom/Chime	AMS：CSDM 和雲端架構師 (CA) 客戶指派的團隊成員（例如：雲端/基礎設施、應用程式支援、架構團隊等）
每月業務審查	檢閱服務水準效能（報告、分析和趨勢） 財務分析	現場客戶location/Telecom/Chime	AMS：CSDM、雲端架構師 (CA)、AMS 客戶團隊、AMS 技術

會議	用途	Mode	參與者
	產品藍圖 CSAT		產品經理 (TPM)（選用）、AMS OPS 經理（選用） 您：Application Operator 代表
每季業務審查	記分卡和服務水準協議 (SLA) 效能和趨勢 (6 個月） 即將到來的 3/6/9/12 個月計劃/遷移 風險與風險緩解措施 關鍵改進計劃 產品藍圖項目 符合未來方向的機會 財務 節省成本計畫 業務最佳化	現場客戶位置	AMS：CSDM、雲端架構師、AMS 客戶團隊、AMS 服務主管、AMS 操作管理員 您：Application Operator 代表、Service 代表、Service Director

CRM 會議安排

AMS CSDM 負責記錄會議，包括：

- 建立議程，包括動作項目、問題和出席者清單。
- 建立每次會議審查的動作項目清單，以確保項目按排程完成和解決。
- 在會議後的一個工作天內，透過電子郵件將會議記錄和動作項目清單分發給會議出席者。
- 將會議記錄存放在適當的文件儲存庫中。

如果沒有 CSDM，領導會議的 AMS 代表會建立和分發會議記錄。

> **Note**
>
> 您的 CSDM 會與您一起建立您的帳戶控管。

CRM 每月報告

您的 AMS CSDM 會準備並傳送每月服務效能簡報。簡報包含下列資訊：

- 報告日期
- 摘要和洞見：
  - 關鍵標註：總和作用中堆疊計數、堆疊修補狀態、帳戶加入狀態（僅限加入期間）、客戶特定問題摘要
  - 效能：事件解決、警示、修補、變更請求 (RFCs)、服務請求，以及主控台和 API 可用性的統計資料
  - 問題、挑戰、疑慮和風險：客戶特定問題狀態
  - 近期項目：客戶特定的加入或事件解決計劃
- 受管資源：堆疊的圖形和圓餅圖
- AMS 指標：監控和事件指標、事件指標、AMS SLA 遵循指標、服務請求指標、變更管理指標、儲存指標、持續性指標、Trusted Advisor 指標和成本摘要（以多種方式呈現）。功能請求。聯絡資訊。

> **Note**
>
> 除了所描述的資訊之外，您的 CSDM 也會通知您範圍或條款的任何重大變更，包括由 AMS 使用承包商進行營運活動。
>
> AMS 會產生有關 CSDM 包含在每月報告中的修補和備份的報告。在報告產生系統中，AMS 會將一些您無法存取的基礎設施新增至您的帳戶：
>
> - 報告原始資料的 S3 儲存貯體
> - Athena 執行個體，具有查詢定義來查詢資料
> - 從 S3 儲存貯體讀取原始資料的 Glue 爬蟲程式

AWS Managed Services 中的成本最佳化

AWS Managed Services 會在每月業務審查 (MBRs) 期間，每月為您提供詳細的成本使用率和節省報告。

AMS 遵循一組標準程序和機制，以識別受管帳戶中的成本節省管道，並協助您規劃和推展變更，以最佳化您的 AWS 支出。

> **Note**
>
> AMS 正在開發影片，以協助成本最佳化。第一步是為您提供 PDF 和 Excel 試算表，其中包含成本最佳化最佳實務。若要存取這些資源，請開啟成本最佳化 ZIP 檔案的快速指南。

成本最佳化架構

AMS 會遵循三個階段的方法，以最佳化您的 AWS 成本：

1. 識別受管環境中的成本最佳化管道
2. 向您介紹成本最佳化計劃
3. 協助以可衡量的方式實現成本最佳化

識別受管環境中的成本最佳化管道

AMS AWS 利用成本總管和 Trusted Advisor 等原生工具，同時利用架構最佳化、EC2 執行個體和以 AWS 帳戶為中心的最佳化等超過 20 種節省成本模式，為您建立量身打造的成本節省建議。

部分最佳化建議包括下列項目。

架構最佳化建議：

- 最佳 S3 儲存類別使用：Amazon S3 提供各種儲存類別，以根據資料存取、彈性和成本滿足各種工作負載需求。根據工作負載需求的 S3 Intelligent-Tiering 和 S3 儲存類別分析可讓您有效率地管理 S3 成本。
- 使用快取架構：在適用的情況下，利用快取執行個體可協助您取代某些資料庫執行個體，同時滿足您的 IOPS 需求。
- EBS 升級節省：將您的 EBS 磁碟區從 gp2 遷移至 gp3 可節省高達 20% 的成本，無論磁碟區大小為何，您都可以利用可預測的 3，000 IOPS 基準效能和 125 MiB/s。

- 使用彈性： AWS 提供的自動擴展功能允許有效的資源使用率和成本最佳化途徑。根據需求定期檢閱和更新執行個體擴展政策，進一步節省成本。

EC2 以執行個體為中心的建議

- 執行個體權利調整：專注於根據用量調整執行個體大小和最佳組態的建議。建議也包括使用 Amazon EC2 Auto Scaling 功能，並在適用於 Amazon S3 上的 AWS Lambda 或靜態 Web 內容時取代 EC2 執行個體。
- 執行個體排程：使用 AMS Resource Scheduler 根據時間排程自動啟動和停止執行個體有助於控制成本，尤其是在非營業時間內未使用的非生產執行個體。
- 訂閱 Savings Plans：Savings plan 是節省 AWS 用量最簡單的方式。相較於 Amazon ECEC2 Instance Savings Plans 最多可節省 72%。 Amazon EC2 Amazon SageMaker AI Savings Plans 為您的 Amazon SageMaker AI 服務用量提供高達 64% 的折扣。AMS 會根據您的 AWS 資源使用量，提供有關 Savings 計劃的適當建議。
- 預留執行個體 (RI) 用量和耗用量指引：Amazon EC2 預留執行個體 (RI) 提供相較於隨需定價的大幅折扣 （ 高達 75%)，並在用於特定可用區域時提供容量保留。
- Spot 執行個體用量：容錯工作負載可以使用 Spot 執行個體，並將價格降低至 90%。
- 閒置執行個體終止：識別和報告閒置或可終止的低使用率執行個體。

以帳戶為中心的建議

- 帳戶清除：在帳戶層級，AMS 也會識別未使用的 EBS 磁碟區、重複的 CloudTrail 追蹤、具有未使用資源的空帳戶等，並提供清除建議。
- SLA 建議：此外，AMS 會定期檢閱您的 Plus 和 Premium 帳戶，並建議為帳戶選擇正確的 SLA 層級。
- AMS 自動化最佳化：AMS 會持續最佳化用於提供 AMS 服務的 AMS 自動化和基礎設施。

向客戶展示並協助規劃

AMS 會與主要客戶利益相關者進行每月業務審查 (MBRs)，並呈現可節省成本的管道、機制和建議，以及潛在的節省成本。我們進一步與您合作，規劃所需的變更。

協助建議實作並衡量成本影響

AMS 可協助實現和衡量成本影響和最佳化變更。

您可以評估建議變更的應用程式影響、風險和成功條件，並透過 AMS 主控台提出適當的變更請求 (RFCs)。AMS 會與您合作，並在受管帳戶中實作與成本最佳化相關的變更。AMS 會測量成本影響，並在每月業務審查 (MBRs) 中包含實現的節省。

成本最佳化責任矩陣

AMS 成本最佳化中的責任。

成本最佳化 RACI

活動	客戶	AMS
編譯節省成本的建議並準備報告	I	R
呈現節省成本報告	C	R
規劃與節省成本相關的變更	R	C
評估變更的影響和風險	R	C
提高 RFCs以實作變更	R	C
檢閱 RFCs並實作變更	C	R
測試應用程式並驗證變更實作	R	C

活動	客戶	AMS
測量變更後的成本影響並向客戶展示	I	R

AWS Managed Services 中的服務時數

功能	AMS 進階
	高級方案
服務請求	全年無休
事件管理 (P2-P3)	全年無休
備份與復原	全年無休
修補管理	全年無休
監控和提醒	全年無休
自動化變更請求 (RFC)	全年無休
非自動化的變更請求 (RFC)	全年無休
雲端服務交付管理員 (CSDM)	週一至週五：08：00–17：00，當地上班時間

在 AWS Managed Services 中取得說明

AMS 一年 365 天、每週 7 天、每天 24 小時為您提供事件管理、服務請求管理和變更管理的支援（根據套用至帳戶的 AMS 服務水準協議）。

若要報告影響受管環境的 AWS 或 AMS 服務效能問題，請使用 AMS 主控台並提交事件報告。如需詳細資訊，請參閱報告事件。如需 AMS 事件管理的一般資訊，請參閱事件回應。

若要要求資訊或建議，或從 AMS 請求其他服務，請使用 AMS 主控台並提交服務請求。如需詳細資訊，請建立服務請求。如需 AMS 服務請求的一般資訊，請參閱服務請求管理。

應用程式開發

可在 AWS Managed Services (AMS) 環境中有效設計和部署應用程式的應用程式開發程序和實務。AMS 會引導您完成下列高階程序：

1. 佈建和架構要開發或整合到 AMS 受管環境的應用程式。一些考量事項：
   a. 您將如何部署應用程式？ 使用 Ansible 等部署工具進行自動化，還是直接上傳所需的檔案手動進行？
   b. 您將如何更新您的應用程式？ 使用可變方法個別更新每個執行個體，還是使用不可變方法，在 Auto Scaling 群組中使用單一更新的 AMI 更新每個執行個體？
2. 規劃和架構將使用 AWS 架構程式庫、 AWS 「Well-Architected」指引，以及 AMS 和其他雲端架構主題專家來託管應用程式的基礎設施。本指南的下列各節提供可協助您處理此問題的資訊。
3. 選取基礎設施部署方法：
   a. 完整堆疊：所有基礎設施元件會一次一起部署。
   b. Tier 和 Tie：基礎設施部署會分開部署，之後會與安全群組修改綁定在一起。這類部署也會透過彼此建置的堆疊元件序列組態來達成；例如，指定您在建立 Auto Scaling 群組時先前建立的負載平衡器。
   c. 您將採用哪些環境，例如開發、預備和生產？
4. 選擇將佈建必要堆疊或層的 AMS 變更類型 (CTs)，並準備必要的變更請求 (RFCs)。
5. 提交 RFCs以觸發基礎設施部署到適當的環境。
6. 使用選取的應用程式部署方法部署應用程式。
7. 視需要重新處理基礎設施和應用程式。
8. 將基礎設施和應用程式部署到適當的後續環境，假設您的第一個部署是非生產環境。
9. 持續維護是由操作基礎基礎設施的 AMS （以及操作應用程式） 基礎設施的操作團隊所處理。
10. 若要停用應用程式，請終止應用程式的 AMS 基礎設施。

架構良好

在 ， AWS 我們相信架構良好的系統可大幅提高業務成功的可能性。AWS 架構中心提供 中架構的專家指導 AWS 雲端。

我們建議您使用下列文章和白皮書，協助您了解建置系統時必須做出之決策的優缺點 AWS。

您是 Well-Architected 嗎？： 推出以六個支柱為基礎的 AWS Well-Architected 架構：

- 卓越營運：卓越營運支柱著重於執行和監控系統，以提供商業價值，並持續改善流程和程序。關鍵主題包括管理和自動化變更、回應事件，以及定義標準以成功管理日常操作。
- 安全：安全支柱著重於保護資訊和系統。關鍵主題包括資料的機密性和完整性、識別和管理誰可以執行許可管理、保護系統，以及建立控制以偵測安全事件。
- 可靠性：可靠性支柱著重於預防 的能力，並快速從失敗中復原以滿足業務和客戶需求。關鍵主題包括設定的基礎元素、跨專案需求、復原規劃，以及如何處理變更。
- 效能效率：效能效率支柱著重於有效率地使用 IT 和運算資源。重要主題包括根據工作負載需求選取適當的資源類型和大小、監控效能，以及做出明智的決策，以便隨著業務需求的發展維持效率。
- 成本最佳化：成本最佳化支柱著重於避免不必要的成本。關鍵主題包括了解和控制花費金錢的位置、選取最適當且正確的資源類型數量、分析一段時間內的支出，以及擴展以滿足業務需求，而不會過度花費。
- 永續性：永續性支柱著重於透過最大限度地利用佈建的資源和最大限度地減少所需的總資源，減少能源消耗並提高工作負載所有元件的效率，以持續改善永續性影響的能力。

AWS Well-Architected Framework：描述 如何 AWS 讓客戶評估和改善其雲端架構，並更好地了解其設計決策的業務影響。它解決了一般設計原則，以及六個概念領域的特定最佳實務和指導，這些概念領域將 AWS 定義為 Well-Architected Framework 的支柱。

AMS 中的應用程式層責任與基礎設施層責任

透過使用 AMS，AMS 會維護您的基礎設施，以及維護和成長所需的一切。不過，無論您需要line-of-business應用程式或產品應用程式， 都是由您開發、部署和維護的。

在 CodeDeploy 和 或 Chef AWS CloudFormation、Puppet、Ansible 或 Saltstack 等應用程式部署工具的協助下，您可以將應用程式部署至 AMS 受管基礎設施。

如需 AMS 執行和不執行之動作的詳細資訊，請參閱 我們做什麼、我們不做什麼。

AMS 中的 Amazon EC2 執行個體可變性

您和 AMS 可以透過兩種方式之一在基礎設施中維護 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體：

- 不可變：此模型使用 Amazon Machine Image (AMIs) 烘焙 （ 建立 ） 和必要的功能。部署更新時，現有執行個體會遭到破壞，並完全取代為從更新 AMI 建立的新執行個體。為了將停機時間降至最

低，此滾動程序會讓某些執行個體無法更新和存取，而其他執行個體則會更新，直到最終完全部署新的變更為止。

- 可互斥：在此模型中，基礎設施會更新為在 雲端中現有系統上部署的新程式碼。此模型混合手動推送更新和使用infrastructure-as-code來部署更新，並且不依賴新的 AMIs。

本指南稍後章節會更詳細討論這些維護模型。

搭配 AMS 資源使用 AWS Secrets Manager

在許多情況下，您可能需要與 AMS 共用秘密，例如：

- RDS 執行個體的主密碼重設
- 負載平衡器的憑證
- 從 AMS 取得 IAM 使用者的長期憑證

與 AMS 共用機密資訊最安全的方式是透過 AWS Secrets Manager；請遵循下列步驟：

1. 使用聯合存取和適用於單一帳戶登陸區域 (SALZ) 的 CustomerReadOnly 角色登入 AWS 主控台；使用下列任一角色：AWSManagedServicesSecurityOpsRole、AWSManagedServicesAdminRole 和適用於多帳戶登陸區域 (MALZ) 的 AWSManagedServicesChangeManagementRole。
2. 導覽至 AWS Secrets 管理員主控台，然後按一下儲存新的秘密。
3. 選取「其他類型的秘密」。
4. 以純文字形式輸入秘密值，然後按一下下一步。
5. 輸入秘密名稱和描述。名稱應一律以「客戶共用/*」開頭。例如「customer-shared/license-2018」。完成後，請按一下下一步繼續。
6. 使用預設 KMS 加密。
7. 讓自動輪換保持停用狀態，然後按一下下一步。
8. 檢閱並按一下儲存，以儲存秘密。
9. 在具有秘密名稱和 ARN 的 AMS 服務請求中回覆我們，以便我們識別和擷取秘密。如需建立服務請求的資訊，請參閱服務請求範例。

AMS 中的應用程式部署

在加入期間，AWS Managed Services (AMS) 會與您一起判斷您需要的基礎設施。

基本基礎設施包括 AWS 虛擬私有雲端 (VPC)、透過 ADFS 樹系信任的通訊安全性、跨兩個可用區域鏡像的基本子網路 (DMZ、共用服務和私有），並使用受管 NAT、堡壘、公有負載平衡器、 AWS Direct Connect (DX) 和必要的安全性進行設定。您的應用程式資源將部署在您的私有或客戶應用程式子網路中。您可以在 AWS Managed Services 使用者指南中進一步了解典型 AMS 架構。

基本概念完成後，您部署的基礎設施應包含應用程式和應用程式開發的所有元件。

AMS 中的應用程式部署功能

您可以在 AMS 中部署應用程式的一些方式。每個方法的詳細資訊如下。

應用程式部署功能範例

方法名稱	基礎設施部署	AMI 或金鑰元素 (s)	應用程式安裝
互斥應用程式、AMS AMI			
手動應用程式部署	完整堆疊 CT 或 Tier 和 Tie CTs	AMS 提供的 AMI	提交存取管理 CT，手動安裝應用程式。
使用應用程式代理程式（即 Chef、Puppet 等）進行 UserData 應用程式部署			使用佈建 CT 搭配安裝應用程式代理程式的 UserData 指令碼，該指令碼/代理程式會安裝應用程式。
UserData 無代理程式應用程式部署（即 Ansible、Salt SSH 等）			提交存取管理 CT、安裝應用程式代理程式。使用應用程式部署工具來部署應用程式。
互斥應用程式、自訂 AMI			

方法名稱	基礎設施部署	AMI 或金鑰元素 (s)	應用程式安裝
自訂 AMI 應用程式部署 （非 ASG)	完整堆疊 CT 或 Tier 和 Tie CTs	自訂 AMI。AMS AMI -> 使用應用程式部署工具代理程式自訂 -> 建立 EC2 執行個體 (CT) -> 建立 AMI (CT)。	應用程式部署工具（即 Chef)，利用代理程式部署應用程式。
AWS Database Migration Service (DMS) 應用程式部署	AWS DMS 同步至現有的 AMS 關聯式資料庫堆疊。	自訂 AMI	客戶或合作夥伴使用 AWS Database Migration Service；AMS 會在啟動時驗證 AMS 元件
工作負載擷取應用程式部署	合作夥伴遷移的執行個體/AMI 和客戶起始的工作負載擷取 CT。		合作夥伴遷移執行個體，在客戶 AMS 受管 VPC 中建立 AMI；客戶使用工作負載擷取 CT 在 AMS 中啟動堆疊。 如需詳細資訊，請參閱AMS 工作負載擷取 (WIGS)。
不可變的應用程式			
自訂 AMI 應用程式部署 (ASG)	完整堆疊 CT 或 Tier 和 Tie CTs	AMS AMI -> 自訂 -> 建立 EC2 執行個體 (CT) -> 建立 AMI (CT) -> 建立 Auto Scaling 群組。	Auto Scaling 使用自訂 AMI 部署應用程式 如需詳細資訊，請參閱AMS 中的 Tier 和 Tie 應用程式部署。
可變或不可變的應用程式			

方法名稱	基礎設施部署	AMI 或金鑰元素 (s)	應用程式安裝
自訂 CloudFormation 範本應用程式部署	CloudFormation 範本	AWS CloudFormation 範本 -> 自訂/準備 AMS -> 部署 | 擷取 | 從 CloudFormation 範本堆疊 | 建立 (ct-36cn2avfrrj9v)。	AMS 會使用自訂 CloudFormation 範本將應用程式部署至您的帳戶，並驗證應用程式部署。 如需詳細資訊，請參閱AMS CloudFormation 擷取。
SQL 資料庫匯入	AMS 操作 （其他 | 其他 CT)	內部部署 SQL 資料庫 -> .bak 檔案 -> AMS RDS SQL 資料庫 -> 管理 | 其他 | 其他 | 建立 (ct-1e1xtak34nx76) 以進行匯入。	AMS 會將您的現場部署資料庫匯入您的 AMS 受管 RDS 資料庫。如需詳細資訊，請參閱資料庫 (DB) 匯入 AMS RDS for Microsoft SQL Server。
資料庫遷移服務 (DMS)	AMS 操作 （多個 CTs)	內部部署資料庫 -> DMS 複寫執行個體 -> DMS 複寫子網路群組 -> DMS 目標端點 -> DMS 來源端點 -> DMS 複寫任務。	AMS 會將您的現場部署資料庫匯入 AMS 受管 S3 或目標 RDS 資料庫。如需詳細資訊，請參閱AWS Database Migration Service (AWS DMS)。
CodeDeploy 應用程式部署	CodeDeploy	應用程式 -> CodeDeploy 應用程式 -> CodeDeploy 部署群組 -> CodeDeploy 部署。	視用量、就地或藍/綠應用程式部署而定。如需詳細資訊，請參閱CodeDeploy 請求。

在 AMS 中規劃您的應用程式部署

如需啟用應用程式部署的建議問題集，請參閱 附錄：應用程式加入問卷。問題涵蓋描述您的：

- 部署摘要
- 基礎設施部署元件
- 應用程式託管平台
- 應用程式部署模型
- 應用程式相依性
- 產品應用程式的 SSL 憑證

AMS 工作負載擷取 (WIGS)

主題

- 遷移工作負載：Linux 和 Windows 的先決條件
- 遷移如何變更您的資源
- 遷移工作負載：標準程序
- 遷移工作負載：CloudEndure 登陸區域 (SALZ)
- AMS 工具帳戶 （遷移工作負載）
- 遷移工作負載：Linux 擷取前驗證
- 遷移工作負載：Windows 擷取前驗證
- 工作負載擷取堆疊：建立

搭配 AMS 雲端遷移合作夥伴使用 AMS 工作負載擷取變更類型 (CT)，將現有工作負載移至 AMS 受管 VPC。使用 AMS 工作負載擷取，您可以在將遷移的執行個體移至 AMS 之後建立自訂 AMS AMI。本節說明遷移合作夥伴和您自己為 AMS 工作負載擷取所採取的程序、先決條件和步驟。

> **Important**
>
> AMS 工作負載擷取必須支援作業系統。如需支援的作業系統，請參閱 遷移工作負載：Linux 和 Windows 的先決條件。
>
> 每個工作負載和帳戶都不同。AMS 將與您合作，為成功的結果做好準備。

下圖說明 AMS 工作負載擷取程序。

遷移工作負載：Linux 和 Windows 的先決條件

在將現場部署執行個體的複本擷取至 AWS Managed Services (AMS) 之前，必須符合特定先決條件。這些是先決條件，包括 Windows 和 Linux 作業系統之間的先決條件。

> **Note**
>
> 為了簡化判斷執行個體是否已準備好擷取的程序，已建立 Windows 和 Linux 的驗證工具。您可以下載這些工具，並直接在內部部署伺服器以及 AWS 中的 EC2 執行個體上執行。Linux Pre-WIGS Validation.zip、Windows Pre-WIGS Validation.zip。

開始之前，適用於 Linux 和 Windows：

- 執行完整的病毒掃描。
- 執行個體必須具有`customer-mc-ec2-instance-profile`執行個體描述檔。
- 安裝 Amazon EC2 Systems Manager (SSM) 代理程式，並確認 SSM 代理程式已啟動並執行。
- 建議在根磁碟區上至少 10GB 的可用磁碟空間執行 AMS 工作負載擷取 (WIGS)。在操作上，AMS 建議磁碟使用率低於 75%，並在磁碟使用率達到 85% 時發出警示。
- 與您的遷移合作夥伴一起決定擷取的時間範圍。
- 自訂 AMI 以 EC2 執行個體的形式存在於目標生產 AMS 帳戶中 （這是遷移合作夥伴的責任）。

> **Important**
>
> AMS 工作負載擷取必須支援作業系統。
>
> - 支援以下作業系統：
>   - Microsoft Windows Server：2008 R2、2012、2012 R2、2016、2019 和 2022
>   - Linux：Amazon Linux 2023、Amazon Linux 2 和 Amazon Linux、CentOS 7.x、CentOS 6.5-6.10、Oracle Linux 7：次要版本 7.5 和更新版本、Oracle Linux 8：最高達 8.3 的次要版本、RHEL 8.x、RHEL 7.x、RHEL 6.5-6.10、SUSE Linux Enterprise Server 15 SP3, SP4 和 SAP 特定版本、SUSE Linux Enterprise Server 12 SP5、Ubuntu 18.04
> - 不支援下列 AMIs：

- Amazon Linux 2023 最小 AMI。

> **Note**
>
> AMS API/CLI (amscm 和 amsskms) 端點位於 AWS N. Virginia 區域 `us-east-1`。根據身分驗證的設定方式，以及您的帳戶和資源所在的 AWS 區域，您可能需要在發出命令`--region us-east-1`時新增 。如果這是您的身分驗證方法`--profile saml`，您可能還需要新增 。

LINUX 先決條件

提交 WIGS RFC 之前，請遵守 中列出的要求，遷移工作負載：Linux 和 Windows 的先決條件並確保下列事項：

- 已安裝最新的增強型聯網驅動程式；請參閱 Linux 上的增強型聯網。
- 已移除與 AMS 元件衝突的第三方軟體元件：
  - 防毒用戶端
  - 備份用戶端
  - 虛擬化軟體 （ 例如 VM 工具或 Hyper-V 整合服務 ）
  - 存取管理軟體 （ 例如 SSSD、Centreify 或 PBIS)
- 確保 SSH 已正確設定 - 這可暫時啟用 SSH 的私有金鑰身分驗證。AMS 將此與我們的組態管理工具搭配使用。使用這些命令：

```
sudo grep -q "^PubkeyAuthentication" /etc/ssh/sshd_config && sudo sed "s/
^PubkeyAuthentication=.*/PubkeyAuthentication yes/" -i /etc/ssh/sshd_config || sudo
 sed "$ a\PubkeyAuthentication yes" -i /etc/ssh/sshd_config
```

```
sudo grep -q "^AuthorizedKeysFile" /etc/ssh/sshd_config && sudo sed "s/
^AuthorizedKeysFile=.*/AuthorizedKeysFile %h\/.ssh\/authorized_keys/" -i /etc/ssh/
sshd_config || sudo sed "$ a\AuthorizedKeysFile %h/.ssh/authorized_keys" -i /etc/ssh/
sshd_config
```

- 確保 Yum 已正確設定 - RedHat 需要授權才能使用其 Yum 儲存庫。執行個體需要透過衛星伺服器或 RedHat Cloud Server 進行授權。如果需要授權，請使用下列其中一個連結：
  - Red Hat 衛星
  - Red Hat 雲端存取

- 如果您使用 Red Hat Satellite，WIGS 需要新增 Red Hat 軟體集合 (RHSCL)。WIGS 系統使用 microSDHCCL 來新增 Python3.6 解譯器，以及系統上設定的任何項目。若要支援此解決方案，必須使用下列儲存庫：
  - rhel-server-rhscl
  - rhel-server-releases-optional

Windows 先決條件

提交 WIGS RFC 之前，請遵守 中列出的要求，遷移工作負載：Linux 和 Windows 的先決條件並確保下列事項：

- 已安裝 Powershell 第 3 版或更新版本。
- AWS EC2 Config 安裝在具有您將遷移之工作負載的執行個體上。
- 安裝支援最新一代執行個體類型的 AWS 驅動程式：PV、EMA 和 NVMe。您可以使用這些連結中的資訊：
  - 升級 Windows 執行個體上的 PV 驅動程式
  - Windows 上的增強型聯網
  - 適用於 Windows 執行個體的 AWS NVMe 驅動程式
  - 第 3 部分：升級 AWS NVMe 驅動程式
  - 第 5 部分：安裝裸機執行個體的序列連接埠驅動程式
  - 第 6 部分：更新電源管理設定
- （選用但建議） 停用關鍵服務 – 將資料庫等關鍵應用程式服務設定為停用，但請確保記錄任何變更，以便在應用程式驗證階段期間將其還原為原始啟動模式。
- （選用但建議） 從預備執行個體建立 Failsafe AMI：
  - 使用 部署 | 進階堆疊元件 | AMI | 建立
  - 在建立期間，新增標籤 Key=Name， Value=APPLICATION-ID_IngestReady
  - 等待 AMI 建立後再繼續
- 已移除與 AMS 元件衝突的第三方軟體元件：
  - 防毒用戶端
  - 備份用戶端
  - 虛擬化軟體 （例如 VM 工具或 Hyper-V 整合服務）

> **Note**
>
> 適用於 Windows 伺服器 (EMP) End-of-Support遷移計劃包含工具，可將舊版應用程式從 Windows Server 2003、2008 和 2008 R2 遷移至 AWS 上較新的支援版本，無需重構。

遷移如何變更您的資源

本節所述的擷取 RFC 會在執行個體遷移至您的 AMS 帳戶後，採取將組態新增至執行個體的下一個步驟，以便 AMS 管理它。

新增的組態是 AMS 特定的，如下所示。

對擷取的 Linux 執行個體所做的變更：

- 已安裝的軟體：
  - Cloud Init：用於設定 Jarvis Access 的私有金鑰。
  - 適用於所有支援作業系統的 Python 3（指令碼語言）(CentOS 6、RHEL 8、OracleLinux 7 除外）。
  - AWS CloudFormation Python 協助程式指令碼：AWS CloudFormation 提供指令碼，用於在 Amazon EC2 執行個體上安裝軟體和啟動服務。
  - AWS CLI：AWS CLI 是一種開放原始碼工具，建置在適用於 Python (Boto) 的 AWS 開發套件之上，可提供與 AWS 服務互動的命令。
  - AWS SSM 代理程式：SSM 代理程式會處理來自 Systems Manager 服務的請求，如請求中所指定來設定機器。
  - AWS CloudWatch Logs 代理程式：將日誌傳送至 CloudWatch。
  - AWS CodeDeploy：一種部署服務，可將應用程式部署自動化至 Amazon EC2 執行個體、內部部署執行個體或無伺服器 Lambda 函數。
  - Ruby：CodeDeploy 的必要項目
  - 系統效能工具 (sysstat)：Sysstat 包含各種公用程式來監控系統效能和用量活動。
  - AD Bridge（先前為 PowerBroker Identity Services)：將非 Microsoft 主機加入 Active Directory 網域。
  - Trend Micro Deep Security Agent：防毒軟體。
- 變更的軟體：
  - 執行個體設定為使用 UTC 時區。

對擷取的 Windows 執行個體所做的變更：

- 已安裝的軟體：
  - 適用於 Windows PowerShell 的 AWS 工具：適用於 PowerShell 的 AWS 工具可讓開發人員和管理員在 PowerShell 指令碼環境中管理其 AWS 服務和資源。
  - Trend Micro Deep Security Agent：防毒保護
  - AMS PowerShell 模組包含 PowerShell 程式碼，用於控制開機、Active Directory 聯結、監控、安全性和記錄。
- 變更的軟體：
  - 伺服器訊息區塊 (SMB) 第 1 版已停用。
  - Windows 遠端管理 (WinRM) 已啟用並設定為接聽連接埠 5986。也會建立允許此傳入連接埠的防火牆規則。
- 可能安裝或變更的軟體：
  - Microsoft .Net Framework 4.5（開發人員平台），如果版本較低，則偵測到 .Net Framework 4.5。
  - 對於 Windows 2012、廣告 Windows 2012R2，我們升級至 PowerShell 5.1。

遷移工作負載：標準程序

> **Note**
>
> 由於此程序需要雙方，本節說明每個方的任務：AMS 雲端遷移合作夥伴（遷移合作夥伴）和應用程式擁有者（您）。

1. 遷移合作夥伴，設定：
   a. 遷移合作夥伴向 AMS 提交 IAM 角色的服務請求，以遷移您的執行個體。如需提交服務請求的詳細資訊，請參閱服務請求範例。
   b. 遷移合作夥伴提交管理員存取請求。AMS Operations 團隊透過請求的 IAM 角色，為遷移合作夥伴提供您帳戶的存取權。
2. 遷移合作夥伴、遷移個別工作負載：

a. 遷移合作夥伴會使用 IAM AWS 執行個體描述檔（必須在帳戶中），透過原生 Amazon EC2 或其他遷移工具將非執行個體遷移至 AMS `customer-mc-ec2-instance-profile` 帳戶中的子網路。

b. 遷移合作夥伴提交 RFC 與部署 | 擷取 | 來自遷移合作夥伴遷移執行個體的堆疊 | 建立 CT (ct-257p9zjk14ija)；如需建立和提交此 RFC 的詳細資訊，請參閱 工作負載擷取堆疊：建立。

   RFC 的執行輸出會傳回執行個體 ID、IP 地址和 AMI ID。

   遷移合作夥伴會為您提供在帳戶中建立之工作負載的執行個體 ID。

3. 您存取和驗證遷移：

a. 使用遷移合作夥伴提供給您的執行輸出 (AMI ID、執行個體 ID 和 IP 地址），提交存取 RFC 並登入新建立的 AMS 堆疊，確認您的應用程式是否正常運作。如需詳細資訊，請參閱請求執行個體存取。

b. 如果滿足，您可以繼續使用啟動的執行個體做為 1 層堆疊和/或使用 AMI 來建立其他堆疊，包括 Auto Scaling 群組。

c. 如果對遷移不滿意，請提出服務請求並參考堆疊和 RFC IDs；AMS 將與您合作解決您的疑慮。

接下來將說明 CloudEndure 登陸區域工作負載擷取程序。

遷移工作負載：CloudEndure 登陸區域 (SALZ)

本節提供將 CloudEndure (CE) 切換執行個體的中繼遷移單一帳戶登陸區域 (SALZ) 設定為可供工作負載擷取 (WIGS) RFC 使用的相關資訊。

若要進一步了解 CloudEndure，請參閱 CloudEndure 遷移。

> **Note**
>
> 這是預先定義的、強化安全性的遷移 LZ 和模式。

事前準備：

- 客戶 AMS 帳戶

- AMS 帳戶與客戶內部部署之間的網路和存取整合
- CloudEndure 帳戶
- AMS 安全審查和簽署的預先核准工作流程，使用您的 CA 和/或 CSDM 執行 （ 例如，濫用 IAM 使用者永久登入資料，可讓您建立/刪除執行個體和安全群組 ）

> **Note**
>
> 本節說明特定的準備和遷移程序。

準備：您和 AMS 運算子：

1. 使用 Management | Other | Other | Update change type to AMS 準備變更請求 (RFC)，以取得下列資源和更新。您可以提交個別的其他 | 其他更新 RFCs，或其中一個。如需該 RFC/CT 的詳細資訊，請參閱其他 | 使用這些請求的其他更新：

   a. 在您的 AMS VPC 中指派次要 CIDR 區塊；在遷移完成後將移除的暫時 CIDR 區塊。確保 區塊不會與任何返回現場部署網路的現有路由衝突。例如，如果您的 AMS VPC CIDR 是 10.0.0.0/16，而且有路由回您的內部部署網路 10.1.0.0/16，則暫時次要 CIDR 可以是 10.255.255.0/24。如需 AWS CIDR 區塊的詳細資訊，請參閱 VPC 和子網路大小。

   b. 在初始累積的 AMS VPC 內建立新的私有子網路。範例名稱：`migration-temp-subnet`。

   c. 為僅具有本機 VPC 和 NAT （ 網際網路 ） 路由的子網路建立新的路由表，以避免在執行個體切換和可能的中斷期間與來源伺服器衝突。確保允許對網際網路的傳出流量進行修補程式下載，以便可以下載和安裝 AMS WIGS 先決條件。

   d. 更新您的 Managed AD 安全群組，以允許進出 的傳入和傳出流量`migration-temp-subnet`。同時請求更新您的 EPS 負載平衡器 (ELB) 安全群組 （ 例如：`mc-eps-McEpsElbPrivateSecurityGroup-M790XBZEEX74`)，以允許新的私有子網路 （ 即 `migration-temp-subnet`)。如果所有三個 TCP 連接埠上都不允許來自專用 CloudEndure (CE) 子網路的流量，WIGS 擷取將會失敗。

   e. 最後，請求新的 CloudEndure IAM 政策和 IAM 使用者。政策需要您的正確帳戶號碼，且`RunInstances`陳述式中的子網路 IDs 應該是：您的 <Customer Application Subnet(s) + Temp Migration Subnet>。

若要查看 AMS 預先核准的 IAM CloudEndure 政策：解壓縮 WIGS 雲端持久性登陸區域範例檔案，然後開啟 `customer_cloud_endure_policy.json`。

> **Note**
>
> 如果您想要更寬鬆的政策，請與您的 CloudArchitect/CSDM 討論您需要什麼，並視需要在提交實作政策的 RFC 之前取得 AMS 安全審查和簽署。

2. 您使用 CloudEndure for AMS 工作負載擷取的準備步驟已完成，而且如果您的遷移合作夥伴已完成準備步驟，就可以執行遷移。WIGS RFC 是由您的遷移合作夥伴提交。

> **Note**
>
> IAM 使用者金鑰不會直接共用，但必須由 AMS 運算子在畫面共用工作階段中輸入 CloudEndure 管理主控台。

準備：遷移合作夥伴和 AMS Operator：

1. 建立 CloudEndure 遷移專案。
   a. 在專案建立期間，在螢幕共用工作階段中具有 AMS 輸入 IAM 使用者登入資料。
   b. 在複寫設定 -> 選擇將啟動複寫伺服器的子網路中，選取 customer-application-x 子網路。
   c. 在複寫設定 -> 選擇要套用至複寫伺服器的安全群組中，選取兩個 Sentinel 安全群組（僅限私有和 EgressAll)。
2. 定義機器的切換選項（執行個體）。
   a. 子網路： migration-temp-subnet。
   b. 安全群組：「Sentinel」安全群組（僅限私有和 EgressAll)。

      切換執行個體必須能夠與 AMS Managed AD 和 AWS 公有端點通訊。
   c. 彈性 IP：無
   d. 公有 IP：否
   e. IAM 角色：Customer-mc-ec2-instance-profile

      IAM 角色必須允許 SSM 通訊。最好使用 AMS 預設。

f. 根據慣例設定標籤。

遷移：遷移合作夥伴：

1. 在 AMS 上建立虛擬堆疊。您可以使用堆疊 ID 來存取堡壘。
2. 在來源伺服器上安裝 CloudEndure (CE) 代理程式。如需詳細資訊，請參閱安裝 代理程式。
3. 在來源伺服器上建立本機管理員登入資料。
4. 排程短切換時段，並在就緒時按一下切換。這會完成遷移，並將使用者重新導向至目標 AWS 區域。
5. 請求堆疊 管理對虛擬堆疊的存取，請參閱管理存取請求。
6. 使用您建立的本機管理員登入資料，登入堡壘，然後登入切換執行個體。
7. 建立故障安全 AMI。如需建立 AMIs的詳細資訊，請參閱 AMI Create。
8. 準備執行個體以供擷取，請參閱 遷移工作負載：Linux 和 Windows 的先決條件。
9. 針對執行個體執行 WIGS RFC，請參閱 工作負載擷取堆疊：建立。

AMS 工具帳戶 （遷移工作負載）

您的多帳戶登陸區域工具帳戶 （使用 VPC) 有助於加速遷移工作、提高安全位置、降低成本和複雜性，以及標準化您的使用模式。

工具帳戶提供下列項目：

- 明確定義的界限，可讓您在生產工作負載之外存取系統整合商的複寫執行個體。
- 可讓您建立隔離的室，檢查工作負載是否有惡意軟體或未知的網路路由，然後再將其放入具有其他工作負載的 帳戶。
- 作為定義的帳戶設定，它可以更快地加入和設定遷移工作負載。
- 隔離的網路路由可保護來自內部部署 -> CloudEndure -> 工具帳戶 -> AMS 擷取映像的流量。擷取映像後，您可以透過 AMS 管理 | 進階堆疊元件 | AMI | 共用 (ct-1eiczxw8ihc18) RFC 將映像分享至目的地帳戶。

高階架構圖：

使用部署 | 受管登陸區域 | 管理帳戶 | 建立工具帳戶 （ 使用 VPC) 變更類型 (ct-2j7q1hgf26x5c)，快速部署工具帳戶，並在多帳戶登陸區域環境中執行個體化工作負載擷取程序。請參閱管理帳戶、工具帳戶：建立 （ 使用 VPC)。

> **Note**
>
> 我們建議您有兩個可用區域 AZs)，因為這是遷移中樞。
> 根據預設，AMS 會在每個帳戶中建立下列兩個安全群組 SGs)。確認這兩個 SGs存在。如果它們不存在，請向 AMS 團隊開啟新的服務請求，以請求它們。
>
> - SentinelDefaultSecurityGroupPrivateOnlyEgressAll
> - InitialGarden-SentinelDefaultSecurityGroupPrivateOnly
>
> 確保 CloudEndure 複寫執行個體是在有路由傳回內部部署的私有子網路中建立。您可以確認私有子網路的路由表具有傳回 TGW 的預設路由。不過，執行 CloudEndure 機器切換應該進入「隔離」私有子網路，其中沒有傳回內部部署的路由，僅允許網際網路傳出流量。請務必確保隔離子網路中發生切換，以避免內部部署資源的潛在問題。

事前準備：

1. Plus 或 Premium 支援層級。
2. 部署 AMIs 之 KMS 金鑰的應用程式帳戶 IDs。
3. 工具帳戶，如先前所述建立。

AWS Application Migration Service (AWS MGN)

AWS Application Migration Service (AWS MGN) 可透過工具帳戶佈建期間自動建立的 `AWSManagedServicesMigrationRole` IAM 角色，在您的 MALZ Tools 帳戶中使用。您可以使用 AWS MGN 遷移在支援的 Windows 和 Linux 作業系統版本上執行的應用程式和資料庫。

如需 AWS 區域 支援up-to-date，請參閱AWS 區域服務清單。

如果 AWS MGN AWS 區域 目前不支援您的偏好，或 AWS MGN 目前不支援應用程式執行所在的作業系統，請考慮改為在工具帳戶中使用 CloudEndure 遷移。

請求 AWS MGN 初始化

AWS MGN 必須在第一次使用前由 AMS 初始化。若要為新的工具帳戶請求此項目，請從工具帳戶提交管理 | 其他 | 其他 RFC，其中包含下列詳細資訊：

```
RFC Subject=Please initialize AWS MGN in this account
RFC Comment=Please click 'Get started' on the MGN welcome page here:

    https://console.aws.amazon.com/mgn/home?region=MALZ_PRIMARY_REGION#/welcome using
 all default values
    to 'Create template' and complete the initialization process.
```

一旦 AMS 成功完成 RFC 並初始化工具帳戶中的 AWS MGN，您可以使用 `AWSManagedServicesMigrationRole` 來編輯預設範本以符合您的需求。

啟用對新 AMS 工具帳戶的存取

工具帳戶建立後，AMS 會為您提供帳戶 ID。您的下一個步驟是設定新帳戶的存取權。請遵循下列步驟。

1. 將適當的 Active Directory 群組更新為適當的帳戶 IDs。

   新 AMS 建立的帳戶會佈建 ReadOnly 角色政策，以及允許使用者提交 RFCs的角色。

   工具帳戶也有額外的 IAM 角色和可用的使用者：

   - IAM 角色： `AWSManagedServicesMigrationRole`
   - IAM 使用者： `customer_cloud_endure_user`
2. 請求政策和角色，以允許服務整合團隊成員設定下一個層級的工具。

   導覽至 AMS 主控台並存檔下列 RFCs：

   a. 建立 KMS 金鑰。使用建立 KMS 金鑰 （ 自動 ） 或建立 KMS 金鑰 （ 需要檢閱 ）。

      當您使用 KMS 加密擷取的資源時，使用與其餘多帳戶登陸區域應用程式帳戶共用的單一 KMS 金鑰，可為可在目的地帳戶中解密的擷取影像提供安全性。

   b. 共用 KMS 金鑰。

      使用 管理 | 進階堆疊元件 | KMS 金鑰 | 共用 （ 需要檢閱 ） 變更類型 (ct-05yb337abq3x5)，請求將新的 KMS 金鑰分享給將存放擷取 AMIs 的應用程式帳戶。

最終帳戶設定的範例圖形：

範例 AMS 預先核准的 IAM CloudEndure 政策

若要查看 AMS 預先核准的 IAM CloudEndure 政策：解壓縮 WIGS 雲端持久性登陸區域範例檔案，然後開啟 `customer_cloud_endure_policy.json`。

測試 AMS Tools 帳戶連線和end-to-end設定

1. 從設定 CloudEndure 開始，並在將複寫至 AMS 的伺服器上安裝 CloudEndure 代理程式。
2. 在 CloudEndure 中建立專案。
3. 透過 Secrets Manager，輸入執行先決條件時共用的 AWS 登入資料。
4. 在複寫設定中：
   a. 針對選擇要套用至複寫伺服器的安全群組選項，選取兩個 AMS "Sentinel" 安全群組 （僅限私有和 EgressAll)。
   b. 定義機器的切換選項 （執行個體）。如需詳細資訊，請參閱步驟 5。剪下
   c. 子網路：私有子網路。
5. 安全群組：
   a. 選取兩個 AMS "Sentinel" 安全群組 （僅限私有和 EgressAll)。
   b. 切換執行個體必須與 AMS 受管 Active Directory (MAD) 和 AWS 公有端點通訊：
      i. 彈性 IP：無
      ii. 公有 IP：否
      iii. IAM 角色：Customer-mc-ec2-instance-profile
   c. 根據您的內部標記慣例設定標籤。
6. 在機器上安裝 CloudEndure 代理程式，並在 EC2 主控台中尋找要在 AMS 帳戶中出現的複寫執行個體。

AMS 擷取程序：

AMS Tools 帳戶衛生

在帳戶中完成共用 AMI 且不再需要複寫的執行個體之後，您會想要清除 ：

- 執行個體後 WIGs擷取：
  - 切換執行個體：在工作完成後，至少透過 AWS 主控台停止或終止此執行個體

- 擷取前 AMI 備份：擷取執行個體並終止現場部署執行個體後移除
- AMS 擷取的執行個體：共用 AMI 後關閉堆疊或終止
- AMS 擷取AMIs：與目的地帳戶共用完成後刪除

- 遷移清除結束：記錄透過開發人員模式部署的資源，以確保定期進行清除，例如：
  - 安全群組
  - 透過 Cloud-formation 建立的資源
  - 網路 ACK
  - 子網路
  - VPC
  - 路由表
  - 角色
  - 使用者和帳戶

大規模遷移 - 遷移工廠

請參閱 AWS CloudEndure 遷移工廠解決方案簡介。

遷移工作負載：Linux 擷取前驗證

您可以驗證執行個體是否已準備好擷取到您的 AMS 帳戶。工作負載擷取 (WIGS) 擷取前驗證會執行檢查，例如作業系統類型、可用的磁碟空間、是否存在衝突的第三方軟體等。執行時，WIGS 擷取前驗證會產生螢幕資料表，以及選用的日誌檔案。結果會提供每個驗證檢查的通過/失敗狀態，以及任何失敗的原因。此外，您可以自訂驗證測試以符合您的需求。

常見問答集：

- 如何使用 Linux WIGS 擷取前驗證？

  請依照下列步驟下載並使用 AMS Linux WIGS 擷取前驗證指令碼：

  1. 使用驗證指令碼下載 ZIP 檔案

     Linux WIGS 擷取前驗證 zip 檔案。
  2. 將連接的規則解壓縮到您選擇的目錄。
  3. 請遵循 readme.md 檔案中的指示。
- Linux WIGS 擷取前驗證會執行哪些驗證？

AMS Linux WIGS 擷取前驗證解決方案會驗證下列項目：

1. 開機磁碟區至少有 5 GB 可用。
2. AMS 支援作業系統。
3. 執行個體具有特定的執行個體描述檔。
4. 執行個體不包含防毒軟體或虛擬化軟體。
5. SSH 已正確設定。
6. 執行個體可存取 Yum 儲存庫。
7. 已安裝增強型聯網驅動程式。
8. 執行個體具有 SSM 代理程式且正在執行。

- 為什麼支援自訂組態檔案？

  指令碼旨在同時在內部部署實體伺服器和 AWS EC2 執行個體上執行。不過，如上表所示，在內部部署執行時，某些測試將會失敗。例如，資料中心中的實體伺服器沒有執行個體描述檔。在這類情況下，您可以編輯組態檔案來略過執行個體描述檔測試，以避免混淆。

- 如何確保我有最新版本的指令碼？

  Linux WIGS 擷取前驗證解決方案up-to-date將可在主要文件頁面上的 AMS 協助程式檔案區段中取得。

- 指令碼是唯讀的嗎？

  指令碼設計為唯讀，但其產生的日誌檔案除外，但應遵循最佳實務以在非生產環境中執行指令碼。

- WIGS 擷取前驗證是否適用於 Windows？

  是。您可以在主要文件頁面的 AMS 協助程式檔案區段下取得。

遷移工作負載：Windows 擷取前驗證

您可以使用 WIGs前驗證程式指令碼來驗證執行個體是否已準備好擷取到您的 AMS 帳戶。工作負載擷取 (WIGS) 擷取前驗證會執行檢查，例如作業系統類型、可用磁碟空間、是否存在衝突的第三方軟體等。執行時，WIGS 擷取前驗證會產生螢幕資料表和選用的日誌檔案。結果會為每個驗證檢查提供通過/失敗狀態，以及失敗原因。此外，您可以自訂驗證測試。

常見問答集：

- 如何使用 Windows WIGS 擷取前驗證？

您可以從 GUI 和 Web 瀏覽器執行驗證，也可以使用 Windows PowerShell、SSM Run Command 或 SSM Session Manager。

選項 1：從 GUI 和 Web 瀏覽器執行

若要從 GUI 和 Web 瀏覽器執行 Windows 預 WIGs，請執行下列動作：

1. 下載具有驗證指令碼的 ZIP 檔案：

   Windows WIGS 擷取前驗證 ZIP 檔案。

2. 將連接的規則解壓縮到您選擇的目錄。
3. 請遵循 README.md 檔案中的指示。

選項 2：從 Windows PowerShell、SSM Run Command 或 SSM Session Manager 執行

Windows 2016 及更新版本

1. 使用驗證指令碼下載 ZIP 檔案。

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"

$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. 從 移除現有檔案`C:\Users\AppData\Local\Temp\AWSManagedServices.PreWigs.Validation`。

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. 叫用指令碼。

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

4. 將附加檔案解壓縮至您選擇的目錄。

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

5. 以互動方式執行驗證指令碼並檢視結果。

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

6. （選用）若要擷取結束代碼區段中列出的錯誤代碼，請在沒有 RunWithoutExitCodes選項的情況下執行指令碼。請注意，此命令會終止作用中的 PowerShell 工作階段。

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

Windows 2012 R2 及更早版本

如果您執行的是 Windows Server 2012R2 或更低版本，您必須先設定 TLS，才能下載 zip 檔案。若要設定 TLS，請完成下列步驟：

1. 使用驗證指令碼下載 ZIP 檔案。

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"

$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. 如果有現有的驗證檔案，請將其移除。

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. 設定 TLS 版本。

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
```

4. 下載 WIG 驗證。

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

5. 將連接的規則解壓縮到您選擇的目錄。

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

6. 以互動方式執行驗證指令碼並檢視結果。

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

7. （選用）若要擷取結束代碼區段中列出的錯誤代碼，請在沒有 RunWithoutExitCodes 選項的情況下執行指令碼。請注意，此命令會終止作用中的 PowerShell 工作階段。

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

> **Note**
>
> 您可以下載並執行 PowerShell 指令碼。若要這樣做，請下載 pre-wigs-validation-powershell-scripts.zip。

- Windows WIGS 擷取前驗證會執行哪些驗證？

  AMS Windows WIGS 擷取前驗證解決方案會驗證下列項目：

  1. 開機磁碟區至少有 10 GB 可用。
  2. AMS 支援作業系統。
  3. 執行個體具有特定的執行個體描述檔。
  4. 執行個體不包含防毒軟體或虛擬化軟體。
  5. 至少一個網路轉接器上已啟用 DHCP。
  6. 執行個體已準備好供 Sysprep 使用。
     - 對於 2008 R2 和 2012 Base 和 R2，Sysprep 會驗證：
       - 有一個 unattend.xml 檔案
       - sppnp.dll file（若有）未損毀
       - 作業系統尚未升級
       - Sysprep 的執行次數未超過 Microsoft 指導方針的次數上限
     - 對於 2016 年及更高版本，上述所有檢查都會略過，因為這不會造成該作業系統的問題
  7. Windows 管理檢測 (WMI) 子系統運作狀態良好。
  8. 已安裝必要的驅動程式。
  9. 已安裝並執行 SSM Agent 和 。
  10. 系統會發出警告，以驗證機器是否因為 RDS 授權組態而處於寬限期。

11.已正確設定必要的登錄機碼。如需詳細資訊，請參閱擷取前驗證 zip 檔案中的 README。

- 為什麼支援自訂組態檔案？

  指令碼旨在同時在內部部署實體伺服器和 AWS EC2 執行個體上執行。不過，如上表所示，在內部部署執行時，某些測試將會失敗。例如，資料中心中的實體伺服器沒有執行個體描述檔。在這類情況下，您可以編輯組態檔案來略過執行個體描述檔測試，以避免混淆。

- 如何確保我有最新版本的指令碼？

  Windows WIGS 擷取前驗證解決方案up-to-date將在主要文件頁面上的 AMS 協助程式檔案區段下提供。

- 指令碼是唯讀的嗎？

  指令碼設計為唯讀，但其產生的日誌檔案除外，但應遵循最佳實務以在非生產環境中執行指令碼。

- Linux 是否可使用 WIGS 預先擷取驗證？

  是。Linux 版本於 2019 年 10 月 31 日啟動。您可以在主要文件頁面的 AMS 協助程式檔案區段下取得。

工作負載擷取堆疊：建立

使用主控台將執行個體遷移至 AMS 堆疊

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。

3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

> **Note**
>
> 如果 RFC 遭到拒絕，則執行輸出會包含 Amazon CloudWatch logs的連結。當不符合需求時，AMS 工作負載擷取 (WIGS) RFCs 會遭到拒絕；例如，如果在執行個體上偵測到防毒軟體。CloudWatch 日誌將包含失敗要求的相關資訊，以及要採取的修復動作。

使用 CLI 將執行個體遷移至 AMS 堆疊

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification`

"{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

您可以使用 AMS CLI 從遷移至 AMS 帳戶的非 AMS 執行個體建立 AMS 執行個體。

> **Note**
>
> 請確定您已遵循先決條件；請參閱遷移工作負載：Linux 和 Windows 的先決條件。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws amscm create-rfc --change-type-id "ct-257p9zjk14ija" --change-type-version "2.0" --
title "AMS-WIG-TEST-NO-ACTION" --execution-parameters "{\"InstanceId\":\"INSTANCE_ID\",
\"TargetVpcId\":\"VPC_ID\",\"TargetSubnetId\":\"SUBNET_ID\",\"TargetInstanceType\":
\"t2.large\",\"ApplyInstanceValidation\":true,\"Name\":\"WIG-TEST\",\"Description\":
\"WIG-TEST\",\"EnforceIMDSV2\":\"false\"}"
```

範本建立：

1. 0將此變更類型的執行參數 JSON 結構描述轉換為檔案；範例將其命名為 MigrateStackParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-257p9zjk14ija" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > MigrateStackParams.json
   ```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為如下內容：

   ```
   {
   ```

```
"InstanceId":          "MIGRATED_INSTANCE_ID",
"TargetVpcId":         "VPC_ID",
"TargetSubnetId":      "SUBNET_ID",
"Name":                "Migrated-Stack",
"Description":         "Create-Migrated-Stack",
"EnforceIMDSV2":       "false"
}
```

3. 輸出 RFC 範本 JSON 檔案；範例將其命名為 MigrateStackRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > MigrateStackRfc.json
```

4. 修改並儲存 MigrateStackRfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
"ChangeTypeId":         "ct-257p9zjk14ija",
"ChangeTypeVersion":    "2.0",
"Title":                "Migrate-Stack-RFC"
}
```

5. 建立 RFC，指定 MigrateStackRfc 檔案和 MigrateStackParams 檔案：

```
aws amscm create-rfc --cli-input-json file://MigrateStackRfc.json  --execution-
parameters file://MigrateStackParams.json
```

   您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

   新的執行個體會出現在相關 VPC 的應用程式擁有者帳戶的執行個體清單中。

6. 一旦 RFC 成功完成，請通知應用程式擁有者，讓其可以登入新的執行個體，並確認工作負載可運作。

> **Note**
>
> 如果 RFC 遭到拒絕，則執行輸出會包含 Amazon CloudWatch logs的連結。當不符合需求時，AMS 工作負載擷取 (WIGS) RFCs 會遭到拒絕；例如，如果在執行個體上偵測到防毒軟體。CloudWatch 日誌將包含失敗要求的相關資訊，以及要採取的修復動作。

提示

> **Note**
>
> 請確定您已遵循先決條件；請參閱遷移工作負載：Linux 和 Windows 的先決條件。

> **Note**
>
> 如果要遷移的執行個體上的標籤具有與 RFC 中提供的標籤相同的索引鍵，RFC 會失敗。

> **Note**
>
> 您最多可以指定四個目標 IDs、連接埠和可用區域。

> **Note**
>
> 如果 RFC 遭到拒絕，則執行輸出會包含 Amazon CloudWatch logs的連結。當不符合需求時，AMS 工作負載擷取 (WIGS) RFCs 會遭到拒絕；例如，如果在執行個體上偵測到防毒軟體。CloudWatch 日誌將包含失敗要求的相關資訊，以及要採取的修復動作。

> **Note**
>
> 如果 RFC 遭到拒絕，則執行輸出會包含 Amazon CloudWatch logs的連結。當不符合需求時，AMS 工作負載擷取 (WIGS) RFCs 會遭到拒絕；例如，如果在執行個體上偵測到防毒軟體。CloudWatch 日誌將包含失敗要求的相關資訊，以及要採取的修復動作。

如有需要，請參閱工作負載擷取 (WIGS) 失敗。

AMS CloudFormation 擷取

AMS AWS CloudFormation 擷取變更類型 (CT) 可讓您透過一些修改，使用現有的 CloudFormation 範本在 AMS 受管 VPC 中部署自訂堆疊。

主題

AMS AWS CloudFormation 擷取程序涉及下列各項：

- 準備自訂 CloudFormation 範本並將其上傳至 S3 儲存貯體，或在建立 RFC 時內嵌範本。如果您使用具有預先簽章 URL 的 S3 儲存貯體；如需詳細資訊，請參閱預先簽章。
- 將 CloudFormation 擷取變更類型提交至 RFC 中的 AMS。如需 CFN 擷取變更類型演練，請參閱 建立 CloudFormation 擷取堆疊。如需 CFN 擷取範例，請參閱 AWS CloudFormation 擷取：範例。
- 堆疊建立後，您可以更新它，並修復它上的偏離；此外，如果更新失敗，您可以明確核准和實作更新。本節說明所有這些程序。

  如需 CFN 偏離偵測的資訊，請參閱新增 – CloudFormation 偏離偵測。

> **Note**
>
> - 此變更類型現在具有 2.0 版。2.0 版為自動化，非手動執行。這可讓 CT 執行速度更快。此版本引進了兩個新參數：CloudFormationTemplate，可讓您將自訂 CloudFormation 範本貼入 RFC，而 VpcId 可讓您將 CloudFormation 擷取與 AMS 多帳戶登陸區域搭配使用。
> - 1.0 版是手動變更類型。這表示 AMS 運算子必須先採取一些動作，變更類型才能成功結束。至少需要檢閱。此版本也需要 CloudFormationTemplateS3Endpoint 參數值為預先簽章的 URL。

AWS CloudFormation 擷取準則、最佳實務和限制

若要讓 AMS 處理您的 CloudFormation 範本，有一些指導方針和限制。

指導方針

若要在執行 AWS CloudFormation 擷取時減少 AWS CloudFormation 錯誤，請遵循下列準則：

- 請勿在範本中嵌入登入資料或其他敏感資訊 – CloudFormation 範本會顯示在 AWS CloudFormation 主控台中，因此您不想在範本中嵌入登入資料或敏感資料。範本不能包含敏感資訊。只有當您使用 AWS Secrets Manager 做為 值時，才允許下列資源：
  - `AWS::RDS::DBInstance` - 【MasterUserPassword，TdeCredentialPassword】
  - `AWS::RDS::DBCluster` - 【MasterUserPassword】
  - `AWS::ElastiCache::ReplicationGroup` - 【AuthToken】

  > **Note**
  >
  > 如需有關在資源屬性中使用 AWS Secrets Manager 秘密的資訊，請參閱如何使用 AWS CloudFormation 範本和使用動態參考指定範本值來建立和擷取 Secrets Manager 中管理的 AWS 秘密。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/dynamic-references.html

- 使用 Amazon RDS 快照來建立 RDS 資料庫執行個體 – 透過這樣做，您不必提供 MasterUserPassword。
- 如果您提交的範本包含 IAM 執行個體描述檔，其字首必須是 'customer'。例如，使用名稱為 'example-instance-profile' 的執行個體描述檔會導致失敗。反之，請使用名稱為 'customer-example-instance-profile' 的執行個體描述檔。
- 請勿在 - 【UserData】 中包含任何敏感資料**`AWS::EC2::Instance`**。 UserData UserData 不應包含密碼、API 金鑰或任何其他敏感資料。這種類型的資料可以加密並存放在 S3 儲存貯體中，並使用 UserData 下載到執行個體。
- 使用 CloudFormation 範本建立 IAM 政策受到限制的支援 – IAM 政策必須由 AMS SecOps 審核和核准。目前我們僅支援使用包含預先核准許可的內嵌政策來部署 IAM 角色。在其他情況下，無法使用 CloudFormation 範本建立 IAM 政策，因為這會覆寫 AMS SecOps 程序。
- 不支援 SSH KeyPairs – Amazon EC2 執行個體必須透過 AMS 存取管理系統存取。AMS RFC 程序會驗證您的身分。您無法在 CloudFormation 範本中包含 SSH 金鑰對，因為您沒有建立 SSH 金鑰對和覆寫 AMS 存取管理模型的許可。
- 安全群組傳入規則受到限制 – 您無法擁有 0.0.0.0/0 的來源 CIDR 範圍，或具有 TCP 連接埠的可公開路由地址空間，而 TCP 連接埠不是 80 或 443。
- 撰寫 CloudFormation 資源範本時，請遵循 AWS CloudFormation 準則 – 請參閱該資源的 AWS CloudFormation 使用者指南，以確保您使用正確的資源資料類型/屬性名稱。例

如，AWS::EC2::Instance 資源中 SecurityGroupIds 屬性的資料類型是「字串值清單」，因此【「sg-aaaaaaaaaaa」】是正常的（使用括號），但「sg-aaaaaaa 不是（不使用括號）。

如需詳細資訊，請參閱 AWS 資源和屬性類型參考。

- 將自訂 CloudFormation 範本設定為使用 AMS CloudFormation 擷取 CT 中定義的參數 – 當您將 CloudFormation 範本設定為使用 AMS CloudFormation 擷取 CT 中定義的參數時，您可以使用 管理 | 自訂堆疊 | CloudFormation 範本的堆疊 | 更新 CT (ct-361tlo1k7339x)，在 CT 輸入中提交具有變更參數值的 CloudFormation 範本，以 CloudFormation 建立類似的堆疊。如需範例，請參閱「AWS CloudFormation 擷取範例：定義資源」。
- 具有預先簽章 URL 的 Amazon S3 儲存貯體端點無法過期 – 如果您使用具有預先簽章 URL 的 Amazon S3 儲存貯體端點，請確認預先簽章的 Amazon S3 URL 尚未過期。使用過期的預先簽章 Amazon S3 儲存貯體 URL 提交的 CloudFormation 擷取 RFC 會遭到拒絕。
- Wait Condition 需要訊號邏輯 – Wait Condition 用於協調堆疊資源建立與堆疊建立外部的組態動作。如果您在範本中使用等待條件資源， 會 AWS CloudFormation 等待成功訊號，如果未發出成功訊號數量，則會將堆疊建立標記為失敗。如果您使用等待條件資源，則需要有訊號的邏輯。如需詳細資訊，請參閱在範本中建立等待條件。

最佳實務

以下是您可以使用 AMS AWS CloudFormation 擷取程序來遷移資源的一些最佳實務：

- 在一個 CT 中提交 IAM 和其他政策相關資源 – 如果您可以使用 CloudFormation Ingest 等自動化 CTs 來部署 IAM 角色，我們建議您這樣做。在其他情況下，AMS 建議您收集所有 IAM 或其他政策相關資源，並將其提交至單一管理 | 其他 | 其他 | 建立變更類型 (ct-1e1xtak34nx76)。例如，合併所需的所有 IAM 角色、IAM Amazon EC2 執行個體描述檔、現有 IAM 角色的 IAM 政策更新、Amazon S3 儲存貯體政策、Amazon SNS/Amazon SQS 政策等，並提交 ct-1e1xtak34nx76 RFC，以便在未來的 CloudFormation 擷取範本中直接參考這些預先存在的資源。
- EC2 執行個體已引導並成功加入網域 – 這是自動完成的最佳實務。為了確保透過 CloudFormation 擷取堆疊啟動的 Amazon EC2 執行個體已引導並成功加入網域，AMS 包含適用於 Auto Scaling 群組資源的 CreationPolicy 和 UpdatePolicy（亦即，如果這些政策尚未存在）。
- 必須指定 Amazon RDS 資料庫執行個體參數 – 透過 AWS CloudFormation 擷取建立 Amazon RDS 資料庫時，您必須指定 `DBSnapshotIdentifier` 參數，才能從先前的資料庫快照還原。這是必要的，因為 AWS CloudFormation 擷取目前不會處理敏感資料。

如需如何將 CloudFormation 範本用於 AMS CloudFormation 範本擷取的範例，請參閱 AWS CloudFormation 擷取：範例。

範本驗證

您可以在將 CloudFormation 範本提交至 AMS 之前進行自我驗證。

提交至 AMS AWS CloudFormation 擷取的範本經過驗證，以確保可在 AMS 帳戶中安全地部署。驗證程序會檢查下列項目：

- 支援的資源 – 僅使用 AMS AWS CloudFormation 擷取支援的資源。如需詳細資訊，請參閱支援的資源。
- 支援的 AMIs – 範本中的 AMI 是 AMS 支援的 AMI。如需 AMS AMIs的詳細資訊，請參閱 AMS Amazon Machine Image AMIs)。
- AMS 共用服務子網路 – 範本不會嘗試在 AMS 共用服務子網路中啟動資源。
- 資源政策 – 沒有過度寬鬆的資源政策，例如可公開讀取或可寫入的 S3 儲存貯體政策。AMS 不允許公開讀取或寫入 S3 儲存貯體 AWS 帳戶。

使用 AWS CloudFormation Linter 驗證

您可以使用 AWS CloudFormation Linter 工具，在將 CloudFormation 範本提交至 AMS 之前進行自我驗證。

AWS CloudFormation Linter 工具是驗證 CloudFormation 範本的最佳方式，因為它可驗證資源/屬性名稱、資料類型和函數。如需詳細資訊，請參閱 aws-cloudformation/cfn-python-lint。

先前顯示的範本 AWS CloudFormation Linter 輸出如下所示：

```
$ cfn-lint -t ./testtmpl.json
E3002 Invalid Property Resources/SNSTopic/Properties/Name
./testtmpl.json:6:9
```

為了協助離線驗證 CloudFormation 範本，AMS 已為 AWS CloudFormation Linter 工具開發一組可插入的自訂驗證規則。它們位於 AMS 主控台的開發人員資源頁面上。

請依照下列步驟使用擷取 AWS CloudFormation 前驗證指令碼：

1. 安裝 AWS CloudFormation Linter 工具。如需安裝說明，請參閱 aws-cloudformation / cfn-lint 。
2. 下載具有驗證指令碼的 .zip 檔案：

   CFN Lint 自訂規則。

3. 將連接的規則解壓縮到您選擇的目錄。
4. 執行下列命令來驗證您的 CloudFormation 範本：

```
cfn-lint --template {TEMPLATE_FILE} --append-rules {DIRECTORY_WITH_CUSTOM_RULES}
```

CloudFormation 擷取堆疊：CFN 驗證器範例

這些範例可協助您準備範本以成功擷取。

格式驗證

驗證範本是否包含「資源」區段，且其下定義的所有資源都有「類型」值。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description" : "Create a SNS topic",
  "Resources": {
    "SnsTopic": {
      "Type": "AWS::SNS::Topic"
    }
  }
}
```

驗證是否允許範本的根金鑰。允許的根金鑰為：

```
[
  "AWSTemplateFormatVersion",
  "Description",
  "Mappings",
  "Parameters",
  "Conditions",
  "Resources",
  "Rules",
  "Outputs",
  "Metadata"
]
```

手動檢閱必要的驗證

如果範本包含下列資源，則自動驗證會失敗，而且您將需要手動檢閱。

從安全角度來看，顯示的政策是高風險區域。例如，除了特定使用者或群組之外，允許任何人建立物件或寫入許可的 S3 儲存貯體政策非常危險。因此，我們會驗證政策，並根據內容核准或拒絕，而且無法自動建立這些政策。我們正在調查解決此問題的可能方法。

我們目前沒有下列資源的自動驗證。

```
[
    "S3::BucketPolicy",
    "SNS::TopicPolicy",
    "SQS::QueuePolicy"
]
```

參數驗證

如果範本參數未提供值，請確認該參數必須具有預設值。

資源屬性驗證

必要屬性檢查：特定資源類型必須存在特定屬性。

- 「VPCOptions」必須存在於 `AWS::OpenSearch::Domain`
- "CludsterSubnetGroupName" 必須存在於 `AWS::Redshift::Cluster`

```
{
    "AWS::OpenSearch::Domain": [
      "VPCOptions"
    ],
    "AWS::Redshift::Cluster": [
      "ClusterSubnetGroupName"
    ]
}
```

不允許的屬性檢查：某些資源類型必須 *not* 存在。

- 「SecretString」不得存在於「AWS::SecretsManager::Secret」中
- 「MongoDbSettings」不得存在於「AWS::DMS::Endpoint」中

```
{
  "AWS::SecretsManager::Secret": [
    "SecretString"
```

```
  ],
  "AWS::DMS::Endpoint": [
    "MongoDbSettings"
  ]
}
```

SSM 參數檢查：對於下列清單中的屬性，必須透過 Secrets Manager 或 Systems Manager 參數存放區（安全字串參數）指定值：

```
{
  "RDS::DBInstance": [
    "MasterUserPassword",
    "TdeCredentialPassword"
  ],
  "RDS::DBCluster": [
    "MasterUserPassword"
  ],
  "ElastiCache::ReplicationGroup": [
    "AuthToken"
  ],
  "DMS::Certificate": [
    "CertificatePem",
    "CertificateWallet"
  ],
  "DMS::Endpoint": [
    "Password"
  ],
  "CodePipeline::Webhook": {
    "AuthenticationConfiguration": [
        "SecretToken"
    ]
  },
  "DocDB::DBCluster": [
    "MasterUserPassword"
  ]
},
```

有些屬性必須符合特定模式；例如，IAM 執行個體設定檔名稱不得以 AMS 預留字首開頭，且屬性值必須符合特定 regex，如下所示：

```
{
    "AWS::EC2::Instance": {
      "IamInstanceProfile": [
```

```
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::AutoScaling::LaunchConfiguration": {
      "IamInstanceProfile": [
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::EC2::LaunchTemplate": {
      "LaunchTemplateData.IamInstanceProfile.Name": [
        "^(?!ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|
Sentinel).+"
      ],
      "LaunchTemplateData.IamInstanceProfile.Arn": [
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile\/(?!ams|Ams|
AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    }
}
```

資源驗證

範本中只能指定允許列出的資源；這些資源如中所述支援的資源。

由於修補限制，同一堆疊中不允許 EC2 堆疊和 Auto Scaling 群組 (ASGs)。

安全群組輸入規則驗證

- 對於來自 CFN 擷取建立或堆疊更新 CT 變更類型的請求：
  - 如果 (IpProtocol 是 tcp 或 6) 和 （連接埠是 80 或 443），則CidrIP值沒有限制
  - 否則， CidrIP不能是 0.0.0.0/0
- 對於來自 Service Catalog (Service Catalog 產品） 的請求：
  - 除了 CFN 擷取建立或堆疊更新 CT 變更類型驗證之外， 中management_ports具有 中通訊協定的連接埠ip_protocols只能透過 存取allowed_cidrs：

    ```
    {
    ```

```
     "ip_protocols": ["tcp", "6", "udp", "17"],
     "management_ports": [22, 23, 389, 636, 1494, 1604, 2222, 3389, 5900, 5901,
5985, 5986],
     "allowed_cidrs": ["10.0.0.0/8", "100.64.0.0/10", "172.16.0.0/12",
"192.168.0.0/16"]
 }
```

限制

AMS AWS CloudFormation 擷取程序目前不支援下列功能。

- YAML – 不支援。僅支援 JSON 型 CloudFormation 範本。
- 巢狀堆疊 – 反之，請架構您的應用程式基礎設施以使用單一範本。或者，您也可以利用跨堆疊參考，跨多個堆疊分隔資源，其中一個資源與另一個資源有相依性。如需詳細資訊，請參閱逐步解說：請參閱另一個 AWS CloudFormation Stack 中的資源輸出。
- CloudFormation 堆疊集 – 由於安全性影響，不支援。
- 使用 CloudFormation 範本建立 IAM 資源 – 由於安全性影響，僅支援 IAM 角色。
- 敏感資料 – 不支援。請勿在範本或參數值中包含敏感資料。如果您需要參考敏感資料，請使用 Secrets Manager 來存放和擷取這些值。如需有關在資源屬性中使用 AWS Secrets Manager 秘密的資訊，請參閱如何使用 AWS CloudFormation 範本和使用動態參考指定範本值，建立和擷取在 AWS Secrets Manager 中管理的秘密 AWS CloudFormation。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/dynamic-references.html

支援的資源

AMS AWS CloudFormation 擷取程序支援下列 AWS 資源。

CloudFormation 擷取堆疊：支援的資源

AMS 工作負載擷取必須支援執行個體作業系統。僅支援此處列出的 AWS 資源。

- Amazon API Gateway
  - AWS::ApiGateway::Account
  - AWS::ApiGateway::ApiKey
  - AWS::ApiGateway::Authorizer
  - AWS::ApiGateway::BasePathMapping

- AWS::ApiGateway::ClientCertificate
- AWS::ApiGateway::Deployment
- AWS::ApiGateway::DocumentationPart
- AWS::ApiGateway::DocumentationVersion
- AWS::ApiGateway::DomainName
- AWS::ApiGateway::GatewayResponse
- AWS::ApiGateway::Method
- AWS::ApiGateway::Model
- AWS::ApiGateway::RequestValidator
- AWS::ApiGateway::Resource
- AWS::ApiGateway::RestApi
- AWS::ApiGateway::Stage
- AWS::ApiGateway::UsagePlan
- AWS::ApiGateway::UsagePlanKey
- AWS::ApiGateway::VpcLink

- Amazon API Gateway V2
  - AWS::ApiGatewayV2::Api
  - AWS::ApiGatewayV2::ApiGatewayManagedOverrides
  - AWS::ApiGatewayV2::ApiMapping
  - AWS::ApiGatewayV2::Authorizer
  - AWS::ApiGatewayV2::Deployment
  - AWS::ApiGatewayV2::DomainName
  - AWS::ApiGatewayV2::Integration
  - AWS::ApiGatewayV2::IntegrationResponse
  - AWS::ApiGatewayV2::Model
  - AWS::ApiGatewayV2::Route
  - AWS::ApiGatewayV2::RouteResponse
  - AWS::ApiGatewayV2::Stage
  - AWS::ApiGatewayV2::VpcLink

- AWS AppSync

- AWS::AppSync::ApiCache
- AWS::AppSync::ApiKey
- AWS::AppSync::DataSource
- AWS::AppSync::FunctionConfiguration
- AWS::AppSync::GraphQLApi
- AWS::AppSync::GraphQLSchema
- AWS::AppSync::Resolver

- Amazon Athena
  - AWS::Athena::NamedQuery
  - AWS::Athena::WorkGroup
- AWS Backup
  - AWS::Backup::BackupVault
- Amazon CloudFront
  - AWS::CloudFront::Distribution
  - AWS::CloudFront::CloudFrontOriginAccessIdentity
  - AWS::CloudFront::StreamingDistribution
- Amazon CloudWatch
  - AWS::CloudWatch::Alarm
  - AWS::CloudWatch::AnomalyDetector
  - AWS::CloudWatch::CompositeAlarm
  - AWS::CloudWatch::Dashboard
  - AWS::CloudWatch::InsightRule
- Amazon CloudWatch Logs
  - AWS::Logs::LogGroup
  - AWS::Logs::LogStream
  - AWS::Logs::MetricFilter
  - AWS::Logs::SubscriptionFilter
- Amazon Cognito
  - AWS::Cognito::IdentityPool
  - AWS::Cognito::IdentityPoolRoleAttachment

  - AWS::Cognito::UserPool
  - AWS::Cognito::UserPoolClient
  - AWS::Cognito::UserPoolDomain
  - AWS::Cognito::UserPoolGroup
  - AWS::Cognito::UserPoolIdentityProvider
  - AWS::Cognito::UserPoolResourceServer
  - AWS::Cognito::UserPoolRiskConfigurationAttachment
  - AWS::Cognito::UserPoolUICustomizationAttachment
  - AWS::Cognito::UserPoolUser
  - AWS::Cognito::UserPoolUserToGroupAttachment
- Amazon DocumentDB
  - AWS::DocDB::DBCluster
  - AWS::DocDB::DBClusterParameterGroup
  - AWS::DocDB::DBInstance
  - AWS::DocDB::DBSubnetGroup
- Amazon DynamoDB
  - AWS::DynamoDB::Table
- Amazon EC2
  - AWS::EC2::Volume
  - AWS::EC2::VolumeAttachment
  - AWS::EC2::Instance
  - AWS::EC2::EIP
  - AWS::EC2::EIPAssociation
  - AWS::EC2::NetworkInterface
  - AWS::EC2::NetworkInterfaceAttachment
  - AWS::EC2::SecurityGroup
  - AWS::EC2::SecurityGroupIngress
  - AWS::EC2::SecurityGroupEgress
  - AWS::EC2::LaunchTemplate

- AWS Batch

- AWS::Batch::ComputeEnvironment
- AWS::Batch::JobDefinition
- AWS::Batch::JobQueue

- Amazon Elastic Container Registry (ECR)
  - AWS::ECR::Repository
- Amazon Elastic Container Service (ECS) (Fargate)
  - AWS::ECS::CapacityProvider
  - AWS::ECS::Cluster
  - AWS::ECS::PrimaryTaskSet
  - AWS::ECS::Service
  - AWS::ECS::TaskDefinition
  - AWS::ECS::TaskSet
- Amazon Elastic File System (EFS)
  - AWS::EFS::FileSystem
  - AWS::EFS::MountTarget
- Amazon ElastiCache
  - AWS::ElastiCache::CacheCluster
  - AWS::ElastiCache::ParameterGroup
  - AWS::ElastiCache::ReplicationGroup
  - AWS::ElastiCache::SecurityGroup
  - AWS::ElastiCache::SecurityGroupIngress
  - AWS::ElastiCache::SubnetGroup
- Amazon EventBridge
  - AWS::Events::EventBus
  - AWS::Events::EventBusPolicy
  - AWS::Events::Rule
- Amazon FSx
  - AWS::FSx::FileSystem
- Amazon Inspector
  - AWS::Inspector::AssessmentTarget

- AWS::Inspector::AssessmentTemplate
- AWS::Inspector::ResourceGroup

- Amazon Kinesis Data Analytics
  - AWS::KinesisAnalytics::Application
  - AWS::KinesisAnalytics::ApplicationOutput
  - AWS::KinesisAnalytics::ApplicationReferenceDataSource
- Amazon Kinesis Data Firehose
  - AWS::KinesisFirehose::DeliveryStream
- Amazon Kinesis Data Streams
  - AWS::Kinesis::Stream
  - AWS::Kinesis::StreamConsumer
- Amazon MQ
  - AWS::AmazonMQ::Broker
  - AWS::AmazonMQ::Configuration
  - AWS::AmazonMQ::ConfigurationAssociation
- Amazon OpenSearch
  - AWS::OpenSearchService::Domain
- Amazon Relational Database Service (RDS)
  - AWS::RDS::DBCluster
  - AWS::RDS::DBClusterParameterGroup
  - AWS::RDS::DBInstance
  - AWS::RDS::DBParameterGroup
  - AWS::RDS::DBSubnetGroup
  - AWS::RDS::EventSubscription
  - AWS::RDS::OptionGroup
- Amazon Route 53
  - AWS::Route53::HealthCheck
  - AWS::Route53::HostedZone
  - AWS::Route53::RecordSet

- AWS::Route53::RecordSetGroup

  - AWS::Route53Resolver::ResolverRule
  - AWS::Route53Resolver::ResolverRuleAssociation
- Amazon Simple Storage Service (Amazon S3)
  - AWS::S3::Bucket
- Amazon Sagemaker
  - AWS::SageMaker::CodeRepository
  - AWS::SageMaker::Endpoint
  - AWS::SageMaker::EndpointConfig
  - AWS::SageMaker::Model
  - AWS::SageMaker::NotebookInstance
  - AWS::SageMaker::NotebookInstanceLifecycleConfig
  - AWS::SageMaker::Workteam
- Amazon Simple Email Service (SES)
  - AWS::SES::ConfigurationSet
  - AWS::SES::ConfigurationSetEventDestination
  - AWS::SES::ReceiptFilter
  - AWS::SES::ReceiptRule
  - AWS::SES::ReceiptRuleSet
  - AWS::SES::Template
- Amazon SimpleDB
  - AWS::SDB::Domain
- Amazon SNS
  - AWS::SNS::Subscription
  - AWS::SNS::Topic
- Amazon SQS
  - AWS::SQS::Queue
- Amazon WorkSpaces
  - AWS::WorkSpaces::Workspace
- Application AutoScaling
  - AWS::ApplicationAutoScaling::ScalableTarget

  - AWS::ApplicationAutoScaling::ScalingPolicy
- Amazon EC2 AutoScaling
  - AWS::AutoScaling::AutoScalingGroup
  - AWS::AutoScaling::LaunchConfiguration
  - AWS::AutoScaling::LifecycleHook
  - AWS::AutoScaling::ScalingPolicy
  - AWS::AutoScaling::ScheduledAction
- AWS Certificate Manager
  - AWS::CertificateManager::Certificate
- AWS CloudFormation
  - AWS::CloudFormation::CustomResource
  - AWS::CloudFormation::Designer
  - AWS::CloudFormation::WaitCondition
  - AWS::CloudFormation::WaitConditionHandle
- AWS CodeBuild
  - AWS::CodeBuild::Project
  - AWS::CodeBuild::ReportGroup
  - AWS::CodeBuild::SourceCredential
- AWS CodeCommit
  - AWS::CodeCommit::Repository
- AWS CodeDeploy
  - AWS::CodeDeploy::Application
  - AWS::CodeDeploy::DeploymentConfig
  - AWS::CodeDeploy::DeploymentGroup
- AWS CodePipeline
  - AWS::CodePipeline::CustomActionType
  - AWS::CodePipeline::Pipeline
  - AWS::CodePipeline::Webhook
- AWS Database Migration Service (DMS)
  - AWS::DMS::Certificate

  - AWS::DMS::Endpoint
  - AWS::DMS::EventSubscription
  - AWS::DMS::ReplicationInstance
  - AWS::DMS::ReplicationSubnetGroup
  - AWS::DMS::ReplicationTask

  不允許 AWS::DMS::Endpoint 資源中的 MongoDbSettings 屬性。

  下列屬性只有在 AWS Secrets Manager 解析時才允許：AWS::DMS::Certificate 資源中的 CertificatePem 和 CertificateWallet 屬性，以及 AWS::DMS::Endpoint 資源中的 Password 屬性。
- AWS Elastic Load Balancing - Application Load Balancer/Network Load Balancer
  - AWS::ElasticLoadBalancingV2::Listener
  - AWS::ElasticLoadBalancingV2::ListenerCertificate
  - AWS::ElasticLoadBalancingV2::ListenerRule
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::ElasticLoadBalancingV2::TargetGroup
- AWS Elastic Load Balancing - Classic Load Balancer
  - AWS::ElasticLoadBalancing::LoadBalancer
- AWS Elemental MediaConvert
  - AWS::MediaConvert::JobTemplate
  - AWS::MediaConvert::Preset
  - AWS::MediaConvert::Queue
- AWS Elemental MediaStore
  - AWS::MediaStore::Container
- AWS Identity and Access Management (IAM)
  - AWS::IAM::Role
- AWS Managed Streaming for Apache Kafka (MSK)
  - AWS::MSK::Cluster
- AWS Glue
  - AWS::Glue::Classifier
  - AWS::Glue::Connection
  - AWS::Glue::Crawler

- AWS::Glue::Database
- AWS::Glue::DataCatalogEncryptionSettings
- AWS::Glue::DevEndpoint
- AWS::Glue::Job
- AWS::Glue::MLTransform
- AWS::Glue::Partition
- AWS::Glue::SecurityConfiguration
- AWS::Glue::Table
- AWS::Glue::Trigger
- AWS::Glue::Workflow

- AWS Key Management Service (KMS)
  - AWS::KMS::Key
  - AWS::KMS::Alias
- AWS Lake Formation
  - AWS::LakeFormation::DataLakeSettings
  - AWS::LakeFormation::Permissions
  - AWS::LakeFormation::Resource
- AWS Lambda
  - AWS::Lambda::Alias
  - AWS::Lambda::EventInvokeConfig
  - AWS::Lambda::EventSourceMapping
  - AWS::Lambda::Function
  - AWS::Lambda::LayerVersion
  - AWS::Lambda::LayerVersionPermission
  - AWS::Lambda::Permission
  - AWS::Lambda::Version
- Amazon Redshift
  - AWS::Redshift::Cluster
  - AWS::Redshift::ClusterParameterGroup
  - AWS::Redshift::ClusterSubnetGroup

- AWS Secrets Manager
  - AWS::SecretsManager::ResourcePolicy
  - AWS::SecretsManager::RotationSchedule
  - AWS::SecretsManager::Secret
  - AWS::SecretsManager::SecretTargetAttachment
- AWS 安全中樞
  - AWS::SecurityHub::Hub
- AWS Step Functions
  - AWS::StepFunctions::Activity
  - AWS::StepFunctions::StateMachine
- AWS Systems Manager (SSM)
  - AWS::SSM::Parameter
- Amazon CloudWatch Synthetics
  - AWS::Synthetics::Canary
- AWS Transfer 系列
  - AWS::Transfer::Server
  - AWS::Transfer::User
- AWS WAF
  - AWS::WAF::ByteMatchSet
  - AWS::WAF::IPSet
  - AWS::WAF::Rule
  - AWS::WAF::SizeConstraintSet
  - AWS::WAF::SqlInjectionMatchSet
  - AWS::WAF::WebACL
  - AWS::WAF::XssMatchSet
- AWS WAF Regional
  - AWS::WAFRegional::ByteMatchSet
  - AWS::WAFRegional::GeoMatchSet
  - AWS::WAFRegional::IPSet
  - AWS::WAFRegional::RateBasedRule

- AWS::WAFRegional::RegexPatternSet
- AWS::WAFRegional::Rule
- AWS::WAFRegional::SizeConstraintSet
- AWS::WAFRegional::SqlInjectionMatchSet
- AWS::WAFRegional::WebACL
- AWS::WAFRegional::WebACLAssociation
- AWS::WAFRegional::XssMatchSet

- AWS WAFv2
  - AWS::WAFv2::IPSet
  - AWS::WAFv2::RegexPatternSet
  - AWS::WAFv2::RuleGroup
  - AWS::WAFv2::WebACL
  - AWS::WAFv2::WebACLAssociation

AWS CloudFormation 擷取：範例

在此處尋找一些詳細範例，說明如何搭配 CloudFormation 範本變更類型使用建立堆疊。

若要下載一組範例 CloudFormation 範本 AWS 區域，請參閱範例範本。

如需 AWS CloudFormation 資源的參考資訊，請參閱 AWS 資源和屬性類型參考。不過，AMS 支援一組較小的資源，如中所述AMS CloudFormation 擷取。

> **Note**
>
> AMS 建議您收集所有 IAM 或其他政策相關資源，並將其提交至單一管理 | 其他 | 其他 | 建立變更類型 (ct-1e1xtak34nx76)。例如，合併所有必要的 IAM 角色、IAM 執行個體描述檔、現有 IAM 角色的 IAM 政策更新、S3 儲存貯體政策、SNS/SQS 政策等，然後提交 ct-1e1xtak34nx76 RFC，以便在未來的 CFN 擷取範本中參考這些預先存在的資源。

主題

- AWS CloudFormation 擷取範例：定義資源
- CloudFormation 擷取範例：3 層 Web 應用程式

AWS CloudFormation 擷取範例：定義資源

使用 AMS AWS CloudFormation 擷取時，您可以自訂 CloudFormation 範本，並使用 CloudFormation 擷取變更類型 (ct-36cn2avfrrj9v) 將其提交至 RFC 中的 AMS。若要建立可多次重複使用的 CloudFormation 範本，請將堆疊組態參數新增至 CloudFormation 擷取變更類型執行輸入，而不是在 CloudFormation 範本中硬式編碼。最大的好處是您可以重複使用範本。

AMS CloudFormation 擷取變更類型輸入結構描述可讓您在 CloudFormation 範本中選擇最多 60 個參數，並提供自訂值。

此範例說明如何定義資源屬性，可用於各種 CloudFormation 範本，做為 AMS CloudFormation 擷取 CT 中的參數。本節中的範例特別顯示 SNS 主題用量。

主題

範例 1：硬式程式碼 the AWS CloudFormation SNSTopic 資源**TopicName**屬性

在此範例中，您會硬式編碼 CloudFormation 範本中的 AWS CloudFormation SNSTopic 資源TopicName屬性。請注意， `Parameters`區段為空白。

若要讓 CloudFormation 範本允許您變更新堆疊的 SNSTopic 名稱值，而不必建立新的 CloudFormation 範本，您可以使用 CloudFormation 擷取變更類型的 AMS `Parameters`區段來建立該組態。透過這樣做，您稍後會使用相同的 CloudFormation 範本來建立具有不同SNSTopic名稱的新堆疊。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
```

```
      "Properties" : {
        "TopicName" : "MyTopicName"
      }
    }
  }
}
```

範例 2：使用 SNSTopic 資源參考 AMS 變更類型中的參數

在此範例中，您可以使用 CloudFormation 範本中定義的`SNSTopic`資源`TopicName`屬性來參考 AMS 變更類型`Parameter`中的 。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
    "TopicName" : {
      "Type" : "String",
      "Description" : "Topic ID",
      "Default" : "MyTopicName"
    }
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
      "Properties" : {
        "TopicName" : { "Ref" : "TopicName"}
      }
    }
  }
}
```

範例 3：使用 AMS 擷取變更類型提交 JSON 執行參數檔案來建立 SNS 主題

在此範例中，您會使用建立 SNS 主題 的 AMS 擷取 CT 來提交 JSON 執行參數檔案`TopicName`。SNS 主題必須以此範例中所示的可修改方式在 CloudFormation 範本中定義。

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
```

```
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ],
  "TimeoutInMinutes": 60
}
```

範例 4：提交參考相同 CloudFormation 範本的新變更類型

此 JSON 範例會變更 SNS TopicName值，而不變更 CloudFormation 範本。反之，您會提交新的部署 | 擷取 | CloudFormation 範本的堆疊 | 建立參考相同 CFN 範本的變更類型。

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic2"}
  ],
  "TimeoutInMinutes": 60
}
```

範例 5：使用 CloudFormation 範本中的預設參數值

在此範例中，建立 SNS TopicName = 'MyTopicName'，因為Parameters執行參數中未提供任何TopicName值。如果您不提供Parameters定義，則會使用 CloudFormation 範本中的預設參數值。

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
```

```
  ],
  "TimeoutInMinutes": 60
}
```

CloudFormation 擷取範例：3 層 Web 應用程式

擷取標準 3-Tier Web 應用程式的 CloudFormation 範本。

這包括 Application Load Balancer、Application Load Balancer 目標群組、Auto Scaling 群組、Auto Scaling 群組啟動範本、具有 MySQL 資料庫的 Amazon Relational Database Service (RDS for SQL Server)、 AWS SSM 參數存放區和 AWS Secrets Manager。請預留 30-60 分鐘的時間來演練此範例。

先決條件

- 使用 Secrets Manager 建立包含使用者名稱和密碼與對應值的 AWS 秘密。您可以參考此包含秘密名稱 的範例 JSON 範本 (zip 檔案) `ams-shared/myapp/dev/dbsecrets`，並將其取代為您的秘密名稱。如需搭配 AMS 使用 AWS Secrets Manager 的詳細資訊，請參閱 搭配 AMS 資源使用 AWS Secrets Manager。
- 在 AWS SSM 參數存放區 (PS) 中設定必要的參數。在此範例中，私有和公有子網路`Subnet-Id`的 `VPCId`和 會存放在 SSM PS 中的路徑，例如 `/app/DemoApp/PublicSubnet1a`、`PrivateSubnet1a`、 `PublicSubnet1c`和 `PrivateSubnet1c`和 `VPCCidr`。根據您的需求更新路徑、參數名稱和值。
- 建立具有 AWS Secrets Manager 和 SSM 參數存放區路徑讀取許可的 IAM Amazon EC2 執行個體角色 （這些範例中建立和使用的 IAM 角色為 `customer-ec2_secrets_manager_instance_profile`)。如果您建立執行個體描述檔角色等 IAM 標準政策，角色名稱必須以 開頭`customer-`。若要建立新的 IAM 角色， （您可以將其命名為 或其他`customer-ec2_secrets_manager_instance_profile`項目 ） 會使用 AMS 變更類型管理 | 應用程式 | IAM 執行個體描述檔 | 建立 (ct-0ixp4ch2tiu04) CT，並連接所需的政策。您可以在 IAM 主控台中檢閱 AMS IAM 標準政策 `customer_secrets_manager_policy`和 AWS `customer_systemsmanager_parameterstore_policy`，以正常使用或做為參考。

擷取標準 3-Tier Web 應用程式的 CloudFormation 範本

1. 將連接的範例 CloudFormation JSON 範本以 zip 檔案 3-tier-cfn-ingest.zip 形式上傳至 S3 儲存貯體，並產生簽署的 S3 URL 以在 CFN 擷取 RFC 中使用。如需詳細資訊，請參閱預先簽章。當您透過 AMS 主控台提交 RFC 時，CFN 範本也可以複製/貼上至 CFN 擷取 RFC。

2. 透過 AMS 主控台或 AMS CLI 建立 CloudFormation 擷取 RFC （部署 | 擷取 | 從 CloudFormation 範本堆疊 | 建立 (ct-36cn2avfrrj9v))。CloudFormation 擷取自動化程序會驗證 CloudFormation 範本，以確保範本具有有效的 AMS 支援資源，並遵守安全標準。

   - 使用主控台 - 針對變更類型，選取部署 -> 擷取 -> 從 CloudFormation 範本堆疊 -> 建立，然後新增下列參數做為範例 （請注意，MultiAZDatabase 的預設值為 false)：

```
CloudFormationTemplateS3Endpoint: "https://s3-ap-
southeast-2.amazonaws.com/amzn-s3-demo-bucket/3-tier-cfn-ingest.json?
AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}"
VpcId: "VPC_ID"
TimeoutInMinutes: 120
IAMEC2InstanceProfile: "customer_ec2_secrets_manager_instance_profile"
MultiAZDatabase: "true"
WebServerCapacity: "2"
```

   - 使用 AWS CLI - 如需使用 RFCs 的詳細資訊 AWS CLI，請參閱建立 RFCs。例如，執行下列命令：

```
aws --profile=saml amscm create-rfc  --change-type-id ct-36cn2avfrrj9v
 --change-type-version "2.0" --title "TEST_CFN_INGEST" --execution-
parameters "{\"CloudFormationTemplateS3Endpoint\":\"https://s3-
ap-southeast-2.amazonaws.com/my-bucket/3-tier-cfn-ingest.json?
AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}\",
\"TimeoutInMinutes\":120,\"Description\":\"TEST\",\"VpcId\":\"VPC_ID\",
\"Name\":\"MY_TEST\",\"Tags\":[{\"Key\":\"env\",\"Value\":\"test\"}],
\"Parameters\":[{\"Name\":\"IAMEC2InstanceProfile\",\"Value\":
\"customer_ec2_secrets_manager_instance_profile\"},{\"Name\":\"MultiAZDatabase\",
\"Value\":\"true\"},{\"Name\":\"VpcId\",\"Value\":\"VPC_ID\"},{\"Name\":
\"WebServerCapacity\",\"Value\":\"2\"}]}" --endpoint-url https://amscm.us-
east-1.amazonaws.com/operational/ --no-verify-ssl
```

   在 AWS CloudFormation RFC 執行輸出中尋找 Application Load Balancer URL 以存取網站。如需存取 資源的資訊，請參閱存取執行個體。

建立 CloudFormation 擷取堆疊

使用主控台建立 CloudFormation 擷取堆疊

使用主控台建立 CloudFormation 擷取堆疊

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，並顯示使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 CloudFormation 擷取堆疊

使用 CLI 建立 CloudFormation 擷取堆疊

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何CreateRfc參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

1. 準備用於建立堆疊的 CloudFormation 範本，並將其上傳至 S3 儲存貯體。如需重要詳細資訊，請參閱 AWS CloudFormation 擷取準則、最佳實務和限制。
2. 建立 RFC 並將其提交至 AMS：

   - 建立並儲存執行參數 JSON 檔案，包括您想要的 CloudFormation 範本參數。下列範例將其命名為 CreateCfnParams.json.

     範例 Web 應用程式堆疊 CreateCfnParams.json 檔案：

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "VpcId": "VPC_ID",
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "TimeoutInMinutes": 120,
  "Tags": [
   {
    "Key":   "Enviroment Type"
    "Value": "Dev",
   },
   {
    "Key":   "Application"
    "Value": "PCS",
   }
  ],
  "Parameters": [
   {
    "Name": "Parameter-for-S3Bucket-Name",
```

```
    "Value":  "BUCKET-NAME"
   },
   {
    "Name": "Parameter-for-Image-Id",
    "Value":  "AMI-ID"
   }
  ],
}
```

範例 SNS 主題 CreateCfnParams.json 檔案：

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ]
}
```

3. 使用下列內容建立並儲存 RFC 參數 JSON 檔案。下列範例會將其命名為 CreateCfnRfc.json 檔案：

```
{
   "ChangeTypeId": "ct-36cn2avfrrj9v",
   "ChangeTypeVersion": "2.0",
   "Title": "cfn-ingest"
}
```

4. 建立 RFC，指定 CreateCfnRfc 檔案和 CreateCfnParams 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCfnRfc.json  --execution-
parameters file://CreateCfnParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

> **Note**
>
> 此變更類型位於 2.0 版，且為自動化（非手動執行）。這可讓 CT 執行更快進行，而新的參數 CloudFormationTemplate 可讓您將自訂 CloudFormation 範本貼入 RFC。此外，在此版本中，如果您指定自己的安全群組，我們不會連接預設 AMS 安全群組。如果您未在請求中指定自己的安全群組，AMS 會連接 AMS 預設安全群組。在 CFN Ingest v1.0 中，無論您是否提供自己的安全群組，我們一律會附加 AMS 預設安全群組。
> AMS 已啟用 17 個 AMS 自我佈建服務，可用於此變更類型。如需支援資源的資訊，請參閱 CloudFormation 擷取堆疊：支援的資源。

> **Note**
>
> 2.0 版接受不是預先簽章 URL 的 S3 端點。
> 如果您使用此舊版 CT，CloudFormationTemplateS3Endpoint 參數值必須是預先簽章的 URL。
> 產生預先簽章 S3 儲存貯體 URL (Mac/Linux) 的範例命令：
>
> ```
> export S3_PRESIGNED_URL=$(aws s3 presign DASHDASHexpires-in 86400
>  s3://BUCKET_NAME/CFN_TEMPLATE.json)
> ```
>
> 產生預先簽章 S3 儲存貯體 URL (Windows) 的範例命令：
>
> ```
> for /f %i in ('aws s3 presign DASHDASHexpires-in 86400
>  s3://BUCKET_NAME/CFN_TEMPLATE.json') do set S3_PRESIGNED_URL=%i
> ```
>
> 另請參閱為 Amazon S3 儲存貯體建立預先簽章的 URLs。

> **Note**
>
> 如果 S3 儲存貯體存在於 AMS 帳戶中，您必須使用此命令的 AMS 登入資料。例如，您可能需要在取得 AMS AWS Security Token Service (AWS STS) 登入資料`--profile saml`後附加 。

相關變更類型：核准 CloudFormation 擷取堆疊變更集、 更新 AWS CloudFormation 擷取堆疊

若要進一步了解 AWS CloudFormation，請參閱 AWS Cloud Formation。若要查看 CloudFormation 範本，請開啟 AWS CloudFormation 範本參考。

驗證 AWS CloudFormation 擷取

範本經過驗證，以確保可以在 AMS 帳戶中建立。如果通過驗證，則會更新為包含符合 AMS 所需的任何資源或組態。這包括新增 資源，例如 Amazon CloudWatch 警示，以允許 AMS Operations 監控堆疊。

如果以下任何一項成立，RFC 會遭到拒絕：

- RFC JSON 語法不正確或不遵循指定的格式。
- 提供的 S3 儲存貯體預先簽章 URL 無效。
- 範本不是有效的 AWS CloudFormation 語法。
- 範本未針對所有參數值設定預設值。
- 範本未通過 AMS 驗證。如需 AMS 驗證步驟，請參閱本主題稍後的資訊。

如果 CloudFormation 堆疊因為資源建立問題而無法建立，RFC 會失敗。

若要進一步了解 CFN 驗證和驗證程式，請參閱範本驗證和 CloudFormation 擷取堆疊：CFN 驗證程式範例。

更新 AWS CloudFormation 擷取堆疊

使用主控台更新 CloudFormation 擷取堆疊

使用主控台更新 CloudFormation 擷取堆疊

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。

- 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。

3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨（如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 更新 CloudFormation 擷取堆疊

使用 CLI 更新 CloudFormation 擷取堆疊

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參

數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

1. 準備您要用來更新堆疊的 AWS CloudFormation 範本，並將其上傳至 S3 儲存貯體。如需重要詳細資訊，請參閱 AWS CloudFormation 擷取準則、最佳實務和限制。
2. 建立 RFC 並將其提交至 AMS：

   - 建立並儲存執行參數 JSON 檔案，包括您想要的 CloudFormation 範本參數。此範例會將其命名為 UpdateCfnParams.json.

     具有內嵌參數更新的範例 UpdateCfnParams.json 檔案：

```
{
  "StackId": "stack-yjjoo9aicjyqw4ro2",
  "VpcId": "VPC_ID",
  "CloudFormationTemplate": "{\"AWSTemplateFormatVersion\":\"2010-09-09\",
\"Description\":\"Create a SNS topic\",\"Parameters\":{\"TopicName\":{\"Type
\":\"String\"},\"DisplayName\":{\"Type\":\"String\"}},\"Resources\":{\"SnsTopic
\":{\"Type\":\"AWS::SNS::Topic\",\"Properties\":{\"TopicName\":{\"Ref\":
\"TopicName\"},\"DisplayName\":{\"Ref\":\"DisplayName\"}}}}}",
  "TemplateParameters": [
    {
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1440
}
```

     包含更新 CloudFormation 範本之 S3 儲存貯體端點的 UpdateCfnParams.json 檔案範例：

```
{
  "StackId": "stack-yjjoo9aicjyqw4ro2",
  "VpcId": "VPC_ID",
  "CloudFormationTemplateS3Endpoint": "s3_url",
  "TemplateParameters": [
```

```
    {
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1080
}
```

3. 使用下列內容建立並儲存 RFC 參數 JSON 檔案。此範例會將其命名為 UpdateCfnRfc.json 檔案。

```
{
   "ChangeTypeId": "ct-361tlo1k7339x",
   "ChangeTypeVersion": "1.0",
   "Title": "cfn-ingest-template-update"
}
```

4. 建立 RFC，指定 UpdateCfnRfc 檔案和 UpdateCfnParams 檔案：

```
aws amscm create-rfc --cli-input-json file://UpdateCfnRfc.json  --execution-
parameters file://UpdateCfnParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

- 此變更類型現在位於 2.0 版。變更包括移除此 CT 版本 1.0 中使用的 AutoApproveUpdateForResources 參數，以及新增兩個新參數：AutoApproveRiskyUpdates 和 BypassDriftCheck。
- 如果 S3 儲存貯體存在於 AMS 帳戶中，您必須使用此命令的 AMS 登入資料。例如，您可能需要在取得 AMS AWS Security Token Service (AWS STS) 登入資料`--profile saml`後附加 。
- CloudFormation 範本中資源的所有`Parameter`值都必須具有一個值，可透過預設值或透過 CT 的參數區段自訂值。您可以透過建構 CloudFormation 範本資源來參考參數金鑰來覆寫參數值。如需示範執行方式的範例，請參閱 CloudFormation 擷取堆疊：CFN 驗證器範例。

重要：缺少格式中未明確提供的參數，預設為現有堆疊或範本上目前設定的值。

- 如需您可以使用 AWS CloudFormation Ingest 新增哪些自行佈建服務的清單，請參閱 CloudFormation Ingest Stack：支援的資源。

  若要進一步了解 AWS CloudFormation，請參閱 AWS Cloud Formation。

驗證 AWS CloudFormation 擷取

範本經過驗證，以確保可以在 AMS 帳戶中建立。如果通過驗證，則會更新為包含符合 AMS 所需的任何資源或組態。這包括新增 資源，例如 Amazon CloudWatch 警示，以允許 AMS Operations 監控堆疊。

如果以下任何一項成立，RFC 會遭到拒絕：

- RFC JSON 語法不正確或不遵循指定的格式。
- 提供的 S3 儲存貯體預先簽章 URL 無效。
- 範本不是有效的 AWS CloudFormation 語法。
- 範本未針對所有參數值設定預設值。
- 範本未通過 AMS 驗證。如需 AMS 驗證步驟，請參閱本主題稍後的資訊。

如果 CloudFormation 堆疊因為資源建立問題而無法建立，RFC 會失敗。

若要進一步了解 CFN 驗證和驗證程式，請參閱範本驗證和 CloudFormation 擷取堆疊：CFN 驗證程式範例。

核准 CloudFormation 擷取堆疊變更集

使用主控台核准和更新 CloudFormation 擷取堆疊

使用主控台核准和更新 CloudFormation 擷取堆疊

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。

- 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，並顯示使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。

3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。

5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 核准和更新 CloudFormation 擷取堆疊

使用 CLI 核准和更新 CloudFormation 擷取堆疊

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參

數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

1. 將此變更類型的執行參數 JSON 結構描述輸出到目前資料夾中的檔案。此範例會將其命名為 CreateAsgParams.json:

```
aws amscm create-rfc --change-type-id "ct-1404e21baa2ox" --change-
type-version "1.0" --title "Approve Update" --execution-parameters
 file://PATH_TO_EXECUTION_PARAMETERS --profile saml
```

2. 修改和儲存結構描述，如下所示：

```
{
  "StackId": "STACK_ID",
  "VpcId": "VPC_ID",
  "ChangeSetName": "UPDATE-ef81e2bc-03f6-4b17-a3c7-feb700e78faa",
  "TimeoutInMinutes": 1080
}
```

提示

> **Note**
>
> 如果堆疊中有多個資源，而且您只想要刪除一部分的堆疊資源，請使用 CloudFormation Update CT；請參閱 CloudFormation Ingest Stack：Update。您也可以提交服務請求案例，如有需要，AMS 工程師可協助您製作變更集。

若要進一步了解 AWS CloudFormation，請參閱 AWS CloudFormation。

更新 AWS CloudFormation 堆疊終止保護

使用 主控台更新 AWS CloudFormation 終止保護堆疊

以下顯示 AMS 主控台中的此變更類型。

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 更新 AWS CloudFormation 堆疊終止保護

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何CreateRfc參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

僅指定您要變更的參數。缺少參數會保留現有的值。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws amscm create-rfc \
--change-type-id "ct-2uzbqr7x7mekd" \
--change-type-version "1.0" \
--title "Enable termination protection on CFN stack" \
--execution-parameters "{\"DocumentName\":\"AWSManagedServices-
ManageResourceTerminationProtection\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ResourceId\":[\"stack-psvnq6cupymio3enl\"],\"TerminationProtectionDesiredState\":
[\"enabled\"]}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 EnableTermProCFNParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-2uzbqr7x7mekd"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    EnableTermProCFNParams.json
   ```

2. 修改並儲存 EnableTermProCFNParams 檔案，只保留您要變更的參數。例如，您可以將內容取代為如下內容：

   ```
   {
     "DocumentName": "AWSManagedServices-ManageResourceTerminationProtection",
     "Region": "us-east-1",
     "Parameters": {
   ```

```
    "ResourceId": ["stack-psvnq6cupymio3enl"],
    "TerminationProtectionDesiredState": ["enabled"]
  }
}
```

3. 將 RFC 範本輸出至目前資料夾中的檔案；此範例會將其命名為 EnableTermProCFNRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > EnableTermProCFNRfc.json
```

4. 修改並儲存 EnableTermProCFNRfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
    "ChangeTypeId": "ct-2uzbqr7x7mekd",
    "ChangeTypeVersion": "1.0",
    "Title": "Enable termination protection on CFN instance"
}
```

5. 建立 RFC，指定 EnableTermProCFNRfc 檔案和 EnableTermProCFNParams 檔案：

```
aws amscm create-rfc --cli-input-json file://EnableTermProCFNRfc.json  --execution-
parameters file://EnableTermProCFNParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

> **Note**
>
> Amazon EC2、EC2 堆疊有相關的 CT：更新終止保護。

若要進一步了解終止保護，請參閱防止堆疊遭到刪除。

在 AMS 中使用 CFN 擷取或堆疊更新 CTs自動化 IAM 部署

您可以使用這些 AMS 變更類型，在多帳戶登陸區域 (MALZ) 和單一帳戶登陸區域 (SALZ) 中部署 IAM 角色 (`AWS::IAM::Role`資源）：

- 部署 | 擷取 | CloudFormation 範本的堆疊 | 建立 (ct-36cn2avfrrj9v)

- 管理 | 自訂堆疊 | CloudFormation 範本的堆疊 | 更新 (ct-361tlo1k7339x)
- 管理 | 自訂堆疊 | 從 CloudFormation 範本堆疊 | 核准和更新 (ct-1404e21baa2ox)

在 CFN 範本中的 IAM 角色上執行的驗證：

- ManagedPolicyArns： 屬性 ManagedPolicyArns 不得存在於 中`AWS::IAM::Role`。驗證不允許將受管政策連接至要佈建的角色。相反地，可以透過 屬性政策使用內嵌政策來管理角色的許可。
- PermissionsBoundary：用於設定角色許可界限的政策只能是 AMS 支援的受管政策：`AWSManagedServices_IAM_PermissionsBoundary`。此政策可做為護欄，保護 AMS 基礎設施資源免於使用佈建的角色進行修改。使用此預設許可界限，可保留 AMS 提供的安全性優勢。

  `AWSManagedServices_IAM_PermissionsBoundary`（預設）是必要項目，如果沒有，請求會遭到拒絕。
- MaxSessionDuration：IAM 角色可設定的工作階段持續時間上限為 1 到 4 小時。AMS 技術標準要求在超過 4 小時的工作階段期間接受客戶風險。
- RoleName：以下命名空間由 AMS 保留，不能用作 IAM 角色名稱字首：

```
AmazonSSMRole,
AMS,
Ams,
ams,
AWSManagedServices,
customer_developer_role,
customer-mc-,
Managed_Services,
MC,
Mc,
mc,
SENTINEL,
Sentinel,
sentinel,
StackSet-AMS,
StackSet-Ams,
StackSet-ams,
StackSet-AWS,
StackSet-MC,
StackSet-Mc,
StackSet-mc
```

- 政策：內嵌在 IAM 角色中的內嵌政策只能包含一組由 AMS 預先核准的 IAM 動作。這是允許使用（控制政策）建立 IAM 角色的所有 IAM 動作的上限。控制政策包含：
  - AWS 受管政策 ReadOnlyAccess 中的所有動作，提供對所有 AWS 服務 和 資源的唯讀存取權
  - 下列動作具有跨帳戶 S3 動作的限制，亦即，允許的 S3 動作只能對與建立的角色位於相同帳戶中的資源執行：

```
amscm:*,
amsskms:*,
lambda:InvokeFunction,
logs:CreateLogStream,
logs:PutLogEvents,
s3:AbortMultipartUpload,
s3:DeleteObject,
s3:DeleteObjectVersion,
s3:ObjectOwnerOverrideToBucketOwner,
s3:PutObject,
s3:ReplicateTags,
secretsmanager:GetRandomPassword,
sns:Publish
```

    透過 CFN 擷取建立或更新的任何 IAM 角色都可以允許此控制政策中列出的動作，或從控制政策中列出的動作範圍縮小（低於）的動作。目前，我們允許這些可分類為唯讀動作的安全 IAM 動作，以及上述無法透過 CTs 完成且根據 AMS 技術標準預先核准的非唯讀動作。

- AssumeRolePolicyDocument：下列實體已預先核准，並可包含在信任政策中，以擔任要建立的角色：
  - 相同帳戶中的任何 IAM 實體（角色、使用者、根使用者、STS 擔任角色工作階段）都可以擔任該角色。
  - 下列 AWS 服務 可以擔任 角色：

```
apigateway.amazonaws.com,
autoscaling.amazonaws.com,
cloudformation.amazonaws.com,
codebuild.amazonaws.com,
codedeploy.amazonaws.com,
codepipeline.amazonaws.com,
datapipeline.amazonaws.com,
datasync.amazonaws.com,
dax.amazonaws.com,
dms.amazonaws.com,
```

```
ec2.amazonaws.com,
ecs-tasks.amazonaws.com,
ecs.application-autoscaling.amazonaws.com,
elasticmapreduce.amazonaws.com,
es.amazonaws.com,
events.amazonaws.com,
firehose.amazonaws.com,
glue.amazonaws.com,
lambda.amazonaws.com,
monitoring.rds.amazonaws.com,
pinpoint.amazonaws.com,
rds.amazonaws.com,
redshift.amazonaws.com,
s3.amazonaws.com,
sagemaker.amazonaws.com,
servicecatalog.amazonaws.com,
sns.amazonaws.com,
ssm.amazonaws.com,
states.amazonaws.com,
storagegateway.amazonaws.com,
transfer.amazonaws.com,
vmie.amazonaws.com
```

- 相同帳戶中的 SAML 供應商可以擔任該角色。目前，唯一支援的 SAML 供應商名稱為 `customer-saml`。

如果一或多個驗證失敗，RFC 會遭到拒絕。RFC 拒絕原因範例如下所示：

```
{"errorMessage":"[ 'LambdaRole: The maximum session duration (in seconds) should be a
 numeric value in the range 3600 to 14400 (i.e. 1 to 4 hours).', 'lambda-policy: Policy
 document is too permissive.']","errorType":"ClientError"}
```

如果您需要有關 RFC 驗證或執行失敗的協助，請使用 RFC 通訊來聯絡 AMS。如需說明，請參閱 RFC 通訊和連接（主控台）。如有任何其他問題，請提交服務請求。如需操作說明，請參閱建立服務請求。

> **Note**
>
> 作為 IAM 驗證的一部分，我們目前不會強制執行任何 IAM 最佳實務。如需 IAM 最佳實務，請參閱 IAM 中的安全最佳實務。

建立具有更寬鬆動作或強制執行 IAM 最佳實務的 IAM 角色

使用下列手動變更類型建立 IAM 實體：

- 部署 | 進階堆疊元件 | Identity and Access Management (IAM) | 建立實體或政策 (ct-3dpd8mdd9jn1r)
- 管理 | 進階堆疊元件 | Identity and Access Management (IAM) | 更新實體或政策 (ct-27tuth19k52b4)

我們建議您在提交這些手動 RFCs之前，先閱讀並了解我們的技術標準。如需存取，請參閱如何存取技術標準。

> **Note**
>
> 直接使用這些手動變更類型建立的每個 IAM 角色都屬於自己的個別堆疊，不會位於透過 CFN Ingest CT 建立其他基礎設施資源的相同堆疊中。

當無法透過自動變更類型完成更新時，透過手動變更類型更新使用 CFN 擷取建立的 IAM 角色

使用 管理 | 進階堆疊元件 | Identity and Access Management (IAM) | 更新實體或政策 (ct-27tuth19k52b4) 變更類型。

> **Important**
>
> 透過手動 CT 的 IAM 角色更新不會反映在 CFN 堆疊範本中，並導致堆疊偏離。一旦透過手動請求將角色更新為未通過驗證的狀態，只要該角色持續不符合我們的驗證，就無法使用 Stack Update CT (ct-361tlo1k7339x) 再次更新該角色。只有在 CFN 堆疊範本符合我們的驗證時，才能使用更新 CT。不過，只要未更新不符合我們驗證的 IAM 資源，且 CFN 範本通過我們的驗證，堆疊仍可透過 Stack Update CT (ct-361tlo1k7339x) 進行更新。

刪除透過擷取建立的 IAM AWS CloudFormation 角色

如果您想要刪除整個堆疊，請使用下列自動刪除堆疊變更類型。如需說明，請參閱刪除堆疊：

- 變更類型 ID：ct-0q0bic0ywqk6c
- 分類：管理 | 標準堆疊 | 堆疊 | 刪除和管理 | 進階堆疊元件 | 堆疊 | 刪除

如果您想要刪除 IAM 角色而不刪除整個堆疊，您可以從 CloudFormation 範本中移除 IAM 角色，並使用更新後的範本做為自動堆疊更新變更類型的輸入：

- 變更類型 ID：ct-361tlo1k7339x
- 分類：管理 | 自訂堆疊 | CloudFormation 範本的堆疊 | 更新

如需說明，請參閱更新 AWS CloudFormation 擷取堆疊。

CodeDeploy 請求

您可以使用 AWS CodeDeploy 建立應用程式容器，然後透過 CodeDeploy 應用程式群組進行部署。如需 CodeDeploy 的詳細資訊，請參閱 AWS CodeDeploy 文件。

使用 AWS CodeDeploy 涉及下列程序：

1. 建立 CodeDeploy 應用程式。CodeDeploy 應用程式是 CodeDeploy 使用的名稱或容器，以確保在部署期間參考正確的修訂、部署組態和部署群組。
2. 建立 CodeDeploy 部署群組。CodeDeploy 部署群組會定義一組以部署為目標的個別執行個體。AMS 對於 EC2 的 CodeDeploy 部署群組有不同的變更類型。
3. 透過 CodeDeploy 部署群組部署 CodeDeploy 應用程式。

CodeDeploy 應用程式

建立或部署 CodeDeploy 應用程式。

建立 CodeDeploy 應用程式

使用主控台建立 CodeDeploy 應用程式

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。

- 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。

3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨（如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。

5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 CodeDeploy 應用程式

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC：`aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC：`aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-0ah3gwb9seqk2" --change-type-version "1.0"
 --title "Stack-Create-CD-App" --execution-parameters "{\"Description\":\"TestCdApp\",
\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-sft6rv00000000000\",\"Name\":\"Test\",
\"TimeoutInMinutes\":60,\"Parameters\":{\"CodeDeployApplicationName\":\"Test\"}}"
```

範本建立：

1. 將 CodeDeploy 應用程式 CT 的執行參數 JSON 結構描述輸出至目前資料夾中的檔案；此範例會將其命名為 CreateCDAppParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
   ```

2. 修改並儲存 JSON 檔案，如下所示。例如，您可以將內容取代為類似以下內容：

   ```
   {
   "Description":                      "Create WP CodeDeploy App",
   "VpcId":                            "VPC_ID",
   "StackTemplateId":                  "stm-sft6rv00000000000",
   "Name":                             "WpCDApp",
   "TimeoutInMinutes":                 60,
   "Parameters":   {
       "CodeDeployApplicationName":    "WordPressCDApp"
       }
   }
   ```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateCDAppRfc.json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
   ```

4. 修改並儲存 JSON 檔案，如下所示。例如，您可以將內容取代為類似以下內容：

   ```
   {
   "ChangeTypeVersion":    "1.0",
   "ChangeTypeId":         "ct-0ah3gwb9seqk2",
   "Title":                "CD-App-Stack-RFC"
   ```

```
}
```

5. 建立 RFC，指定 CreateCDAppRfc 檔案和執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

如需 AWS CodeDeploy 的詳細資訊，請參閱使用 AWS CodeDeploy 建立應用程式。

部署 CodeDeploy 應用程式

使用主控台部署 CodeDeploy 應用程式

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。

5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 部署 CodeDeploy 應用程式

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *ID*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *ID*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-2edc3sd1sqmrb" --change-
type-version "2.0" --title "Stack-Deploy-CD-App" --execution-
parameters "{\"Description\":\"MyCDAppDeployTest\",\"VpcId\":
\"VPC_ID\",\"Name\":\"Test\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployDeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"CodeDeployDeploymentGroupName\":\"TestCDDepGroup\",
```

```
\"CodeDeployIgnoreApplicationStopFailures\":false,\"CodeDeployRevision\":
{\"RevisionType\":\"S3\",\"S3Location\":{\"S3Bucket\":\"amzn-s3-demo-bucket\",
\"S3BundleType\":\"tar\",\"S3Key\":\"TestKey\"}}}}"Test\"}}"
```

範本建立：

1. 輸出 CodeDeploy 應用程式部署 CT 的執行參數 JSON 結構描述；此範例會將其命名為 DeployCDAppParams.json:

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
```

2. 修改 JSON 檔案，如下所示。例如，您可以將內容取代為類似以下內容：

```
{
"Description":                      "Deploy WordPress CodeDeploy Application",
"VpcId":                            "VPC_ID",
"Name":                             "WP CodeDeploy Deployment Group",
"TimeoutInMinutes":                 360,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployDeploymentGroupName":    "WordPressCDDepGroup",
    "CodeDeployIgnoreApplicationStopFailures": false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket": "amzn-s3-demo-bucket",
        "S3BundleType": "zip",
        "S3Key": "wordpress.zip" }
        }
    }
}
```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 DeployCDAppRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. 修改並儲存 DeployCDAppRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "2.0",
```

```
"ChangeTypeId":          "ct-2edc3sd1sqmrb",
"Title":                 "CD-Deploy-For-CD-APP-Stack-RFC"
}
```

5. 建立 RFC，指定執行參數檔案和 DeployCDAppRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

如需詳細資訊，請參閱使用 CodeDeploy 建立部署。

CodeDeploy 部署群組

建立 CodeDeploy 應用程式群組。

建立 CodeDeploy 部署群組

使用主控台建立 CodeDeploy 部署群組

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 CodeDeploy 部署群組

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-2gd0u847qd9d2" --change-type-version
 "1.0" --title "Stack-Create-CD-Dep-Group" --execution-parameters "{\"Description
\":\"TestCdDepGroupRfc\",\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-
sp9lrk00000000000\",\"Name\":\"MyTestCDDepGroup\",\"TimeoutInMinutes\":60,\"Parameters
\":{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployAutoScalingGroups\":
[\"TestASG\"],\"CodeDeployDeploymentConfigName\":\"CodeDeployDefault.OneAtATime\",
\"CodeDeployDeploymentGroupName\":\"Test\",\"CodeDeployServiceRoleArn\":
\"arn:aws:iam::000000000:role/aws-codedeploy-role\"}}"
```

範本建立：

1. 將執行參數 JSON 結構描述輸出到目前資料夾中的檔案；此範例將其命名為 CreateCDDepGroupParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateCDDepGroupParams.json
   ```

2. 修改並儲存 JSON 檔案。例如，您可以將內容取代為類似以下內容：

   ```
   {
   "Description":                      "CreateCDDeploymentGroup",
   "VpcId":                            "VPC_ID",
   "StackTemplateId":                  "stm-sp9lrk00000000000",
   "Name":                             "WordPressCDAppGroup",
   "TimeoutInMinutes":                 60,
   "Parameters":   {
       "CodeDeployApplicationName":        "WordPressCDApp",
       "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
       "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
       "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
       "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
   codedeploy-role"
       }
   }
   ```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateCDDepGroupRfc.json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
   ```

4. 修改並儲存 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-RFC"
}
```

5. 建立 RFC，指定 CreateCDDepGroupRfc 檔案和執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

如需 AWS CodeDeploy 部署群組的詳細資訊，請參閱使用 AWS CodeDeploy 建立部署群組。

建立 EC2 的 CodeDeploy 部署群組

使用主控台為 EC2 建立 CodeDeploy 部署群組

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。

4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 為 EC2 建立 CodeDeploy 部署群組

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-00tlkda4242x7" --change-type-
version "1.0" --title "Stack-Create-CD-Ec2-Dep-Group" --execution-parameters
 "{\"Description\":\"MyTestCdDepEc2DepGroup\",\"VpcId\":\"VPC_ID\",\"Name\":
\"TestCDDepEc2Group\",\"StackTemplateId\":\"stm-n3hsoirgqeqqdbpk2\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"ApplicationName\":\"TestCDApp\",\"DeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"AutoRollbackEnabled\":\"False\",\"EC2FilterTag\":
\"Name=Test\",\"EC2FilterTag2\":\"\",\"EC2FilterTag3\":\"\",\"ServiceRoleArn\":\"\"}}"
```

範本建立：

1. 將執行參數 JSON 結構描述輸出至檔案；此範例將其命名為 CreateCDDepGroupEc2Params.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-00tlkda4242x7"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateCDDepGroupEc2Params.json
   ```

2. 修改並儲存 JSON 檔案。例如，您可以將內容取代為類似以下內容：

   ```
   {
   "Description":                    "CreateCDDepGroupEc2",
   "VpcId":                          "VPC_ID",
   "StackTemplateId":                "stm-n3hsoirgqeqqdbpk2",
   "Name":                           "CDAppGroupEc2",
   "TimeoutInMinutes":               60,
   "Parameters":   {
       "ApplicationName":        "CDAppEc2",
       "DeploymentConfigName":   "CodeDeployDefault.OneAtATime",
       "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
       "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
   codedeploy-role"
       }
   }
   ```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；此範例將其命名為 CreateCDDepGroupEc2Rfc.json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupEc2Rfc.json
   ```

4. 修改並儲存 JSON 檔案。例如，您可以將內容取代為類似以下內容：

   ```
   {
   ```

```
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-00tlkda4242x7",
"Title":                "CD-Dep-Group-For-Ec2-Stack-RFC"
}
```

5. 建立 RFC，指定 CreateCDDepGroupEc2Rfc 檔案和執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupEc2Rfc.json --
execution-parameters file://CreateCDDepGroupEc2Params.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

如需 AWS CodeDeploy 部署群組的詳細資訊，請參閱使用 AWS CodeDeploy 建立部署群組。

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) 可協助您輕鬆安全地將資料庫遷移至 AMS。您可以將資料在最廣為使用的商用和開放原始碼資料庫之間來回遷移，例如 Oracle、MySQL 和 PostgreSQL。此服務支援同質遷移，例如 Oracle 到 Oracle，以及不同資料庫平台之間的異質遷移，例如 Oracle 到 PostgreSQL 或 MySQL 到 Oracle。 AWS DMS 是一種 AWS 服務；AMS CTs可協助您在 AMS 受管帳戶中建立 AWS DMS 資源

下圖說明資料庫遷移的工作流程。

主題

- AWS Database Migration Service (AWS DMS)，在您開始之前
- AWS DMS，設定所需的資料
- AWS DMS 設定任務
- AWS DMS 管理

AWS Database Migration Service (AWS DMS)，在您開始之前

使用 AMS 規劃資料庫遷移時 AWS DMS，請考慮下列事項：

- 來源和目標端點：您需要知道來源資料庫中需要遷移到目標資料庫的資訊和資料表。AMS AWS DMS 支援基本結構描述遷移，包括建立資料表和主索引鍵。不過，AMS AWS DMS 不會在目標資料庫中自動建立次要索引、外部索引鍵、帳戶等。如需詳細資訊，請參閱資料遷移的來源和資料遷移的目標。
- 結構描述/程式碼遷移：AMS AWS DMS 不會執行結構描述或程式碼轉換。您可以使用 Oracle SQL Developer、MySQL Workbench 或 pgAdmin III 等工具轉換您的結構描述。如果您想要將現有的結構描述轉換為不同的資料庫引擎，您可以使用 AWS Schema Conversion Tool。它可以建立目標結構描述，也可以產生和建立整個結構描述：資料表、索引、檢視等。您也可以使用此工具將 PL/SQL 或 TSQL 轉換成 PgSQL 和其他格式。
- 不支援的資料類型：某些來源資料類型需要轉換為目標資料庫的同等資料類型。

AWS DMS 要考慮的案例

以下記錄的案例可協助您打造自己的資料庫遷移路徑。

- 將資料從現場部署 MySQL 伺服器遷移至 Amazon RDS MySQL：請參閱 AWS 部落格文章將現場部署 MySQL 資料遷移至 Amazon RDS （和返回）
- 從 Oracle 資料庫將資料遷移至 Amazon RDS Aurora PostgreSQL 資料庫：請參閱 AWS 部落格文章從 Oracle 資料庫遷移至 Amazon Aurora PostgreSQL 資料庫的快速簡介
- 將資料從 RDS MySQL 遷移至 S3：請參閱 AWS 部落格文章如何使用 AWS DMS 將資料從關聯式資料庫封存至 Amazon Glacier

針對資料庫遷移，您必須執行下列操作：

- 規劃資料庫遷移，包括設定複寫子網路群組。
- 配置執行遷移所有程序的複寫執行個體。
- 指定來源和目標資料庫端點。
- 建立單一任務或一組任務來定義您希望使用的資料表和複寫程序。
- 建立 AWS DMS IAM `dms-cloudwatch-logs-role`和 `dms-vpc-role` 角色。如果您使用 Amazon Redshift 做為目標資料庫，也必須建立 IAM 角色並將其新增至`dms-access-for-endpoint`您的 AWS 帳戶。如需詳細資訊，請參閱建立要與 AWS CLI 和 AWS DMS API 搭配使用的 IAM 角色。

這些演練提供使用 AMS 主控台或 AMS CLI 建立 AWS Database Migration Service () 的範例AWS DMS。提供用於建立 AWS DMS 複寫執行個體、子網路群組和任務，以及 AWS DMS 來源端點和目標端點的 CLI 命令。

若要進一步了解 AMS AWS DMS，請參閱 AWS Database Migration Service 以取得一般資訊，以及AWS Database Migration Service FAQs以取得常見問題的解答。

AWS DMS，設定所需的資料

對於以下每個 AWS DMS 演練，都需要一些常見的資料。

- `Description`：關於 資源的有意義的資訊，這與其他參數`Description`選項是分開的。
- `VpcId`：要使用的 VPC。您可以執行 SKMS API 的 ListVpcSummaries 操作 (`list-vpc-summaries`在 CLI 中），或查看 AMS 主控台中的 VPCs頁面，藉此了解這一點。如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。
- `Name`：堆疊或堆疊元件的名稱；這會成為堆疊名稱。
- `TimeoutInMinutes`：RFC 失敗之前，允許建立堆疊的分鐘數。此設定不會延遲 RFC 執行，但您必須提供足夠的時間 （例如，不要指定 "5")。
- `ChangeTypeId`、 和 `StackTemplateId`：這些是必要項目`ChangeTypeVersion`，但因 CT 而異，其值會在以下每個相關區段中提供。

AWS DMS 設定任務

AWS DMS 使用下列逐步解說來設定 。

1： AWS DMS 複寫子網路群組：建立

您可以使用 AMS 主控台或 API/CLI 來建立 AMS AWS DMS 複寫子網路群組。

建立 AWS DMS 複寫子網路群組

使用主控台建立 AWS DMS 複寫子網路群組

> **Note**
>
> 如果帳戶中不存在 `dms-vpc-role` IAM 角色，則此 CT 失敗。

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 AWS DMS 複寫子網路群組

> **Note**
>
> 如果帳戶中不存在 `dms-vpc-role` IAM 角色，則此 CT 失敗。

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何CreateRfc參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-2q5azjd8p1ag5" --change-type-version "1.0" --title "TestDMSRepSG" --execution-
parameters "{\"Description\":\"DMSTestRepSG\",\"VpcId\":\"VPC-ID\",\"Name\":\"Test
 Stack\",\"Parameters\":{\"Description\":\"DESCRIPTION\",\"SubnetIds\":[\"SUBNET-ID\",
 \"SUBNET-ID\"]},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-j637f96ls1h4oy5fj
\"}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 CreateDmsRsgParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-2q5azjd8p1ag5" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRsgParams.json
   ```

2. 修改並儲存執行參數 CreateDmsRsgParams.json 檔案。例如，您可以將內容取代為如下內容：

   ```
   {
   "Description":          "DMSTestRepSG",
   "VpcId":                "VPC_ID",
   "TimeoutInMinutes":     60,
   "StackTemplateId":      "stm-j637f96ls1h4oy5fj",
   ```

```
"Name":                 "Test RSG",
"Parameters":   {
    "Description":          "DESCRIPTION",
    "SubnetIds":            ["SUBNET_ID", "SUBNET_ID"]
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsRsgRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRsgRfc.json
```

4. 修改並儲存 CreateDmsRsgRfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2q5azjd8p1ag5",
"Title":                "DMS-RSG-Create-RFC"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsRsgRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsRsgRfc.json --execution-
parameters file://CreateDmsRsgParams.json
```

您會在回應中收到新 RFC 的 ID，並使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

- 如果帳戶中不存在 `dms-vpc-role` IAM 角色，則此 CT 失敗。
- 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

如需 DMS 複寫執行個體和子網路群組的詳細資訊，請參閱設定複寫執行個體的網路。

2： AWS DMS 複寫執行個體：建立

您可以使用 AMS 主控台或 API/CLI 來建立 AMS AWS DMS 複寫執行個體。

建立 AWS DMS 複寫執行個體

使用主控台建立 AWS DMS 複寫執行個體

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 AWS DMS 複寫執行個體

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

監控 RFC： `aws amscm get-rfc --rfc-id ID`命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-27apldkhqr0ol" --change-type-version "1.0" --title "TestDMSRepInstance" --
execution-parameters "{\"Description\":\"DMSTestRepInstance\",\"VpcId\":\"VPC-ID\",
\"Name\":\"REP-INSTANCE-NAME\",\"Parameters\":{\"InstanceClass\":\"dms.t2.micro\",
\"ReplicationSubnetGroupIdentifier\":\"TEST-REP-SG\",\"SecurityGroupIds\":\"SG-ID, SG-
ID\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-3n1j5hdrmiiiuqk6v\"}"
```

建立複寫執行個體時，您可以指定來源和目標資料存放區。來源和目標資料存放區可以位於 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、AWS S3 儲存貯體、Amazon Relational Database Service (Amazon RDS) 資料庫執行個體或內部部署資料庫。

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 CreateDmsRiParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-27apldkhqr0ol" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRiParams.json
   ```

2. 修改並儲存執行參數 CreateDmsRiParams.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"Description":          "DMSTestRepInstance",
"VpcId":                "VPC_ID",
"Name":                 "Test RI",
"StackTemplateId":      "stm-3n1j5hdrmiiiuqk6v",
"TimeoutInMinutes":     60,
"Parameters":   {
    "Description":                      "DESCRIPTION",
    "InstanceClass":                    "dms.t2.micro",
    "ReplicationSubnetGroupIdentifier": "TEST-REP-SG",
    "SecurityGroupIds":                 ["SG-ID, SG-ID"}
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsRiRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRiRfc.json
```

4. 修改並儲存 CreateDmsRiRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-27apldkhqr0ol",
"Title":                "DMS-RI-Create-RFC"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsRiRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsRiRfc.json --execution-
parameters file://CreateDmsRiParams.json
```

   您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

- 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

- 您必須在 AMS VPC 中的 EC2 執行個體上建立複寫執行個體，該執行個體具有足夠的儲存和處理能力，可執行您指派的任務，並將資料從來源資料庫遷移到目標資料庫。此執行個體的大小需求取決於遷移的資料量、執行個體需要執行的任務。當您選取 `MultiAZ`選項時，複寫執行個體會使用異地同步備份部署提供高可用性和容錯移轉支援。如需複寫執行個體的詳細資訊，請參閱使用 AWS DMS 複寫執行個體。

3： AWS DMS source 端點：建立、為 Mongo 資料庫建立、為 S3 建立

您可以使用 AMS 主控台或 API/CLI 為各種資料庫建立 AMS DMS 來源端點，我們提供三個範例。

DMS 來源端點：建立

使用主控台建立 DMS 來源端點

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 DMS 來源端點

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *ID*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *ID*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --title "MariaDB-DMS-
Source-Endpoint" --aws-account-id ACCOUNT-ID --change-type-id ct-0attesnjqy2cx --
change-type-version 1.0 --execution-parameters "{\"Description\":\"DESCRIPTION.\",
\"VpcId\":\"VPC-ID\",\"Name\":\"MariaDB-DMS-SE\",\"Parameters\":{\"EngineName\":
\"mariadb\",\"ServerName\":\"mariadb.db.example.com\",\"Port\":3306,\"Username\":
\"DB-USER\",\"Password\":\"DB-PW\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

範本建立：

1. 將此變更類型的執行參數輸出至名為 CreateDmsSeParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-0attesnjqy2cx" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeParams.json
```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"Description":          "MariaDB-DMS-SE",
"VpcId":                "VPC_ID",
"Name":                 "Test SE",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"TimeoutInMinutes":     60,
"Parameters":   {
    "Description":      "DESCRIPTION",
    "EngineName":       "mariadb",
    "ServerName":       "mariadb.db.example.com",
    "Port":             "3306",
    "Username":         "DB-USER",
    "Password":         "DB-PW",}
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsSeRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeRfc.json
```

4. 修改並儲存 CreateDmsSeRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0attesnjqy2cx",
"Title":                "MariaDB-DMS-Source-Endpoint"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsSeRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeRfc.json --execution-
parameters file://CreateDmsSeParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

建立 DMS 端點之前，請確定您的密碼不包含不支援的字元。如需詳細資訊，請參閱AWS Database Migration Service 《 使用者指南》中的建立來源和目標端點。

若要進一步了解，請參閱資料遷移的來源。

如需 S3 來源端點，請參閱 S3 的 DMS 來源端點：建立。

如需 Mongo 資料庫來源端點，請參閱 MongoDB 的 DMS 來源端點：建立。

MongoDB 的 DMS 來源端點：建立

使用主控台建立 DMS Mongo 資料庫來源端點

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 DMS Mongo 資料庫來源端點

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *ID*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *ID*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm  --profile saml --region us-east-1 create-rfc --change-type-id
 "ct-2hxcllf1b4ey0" --change-type-version "1.0" --title 'DMS_Source_MongoDB'
 --description "DESCRIPTION"  --execution-parameters "{\"Description\":
\"DMS_MongoDB_Source_Endpoint\",\"VpcId\":\"VPC_ID\",\"Name\":\"DMS-Mongo-SE\",
\"StackTemplateId\":\"stm-pud4ghhkp7395n9bc\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"DatabaseName\":\"mytestdb\",\"EngineName\":\"mongodb\",\"Port\":27017,\"ServerName
\":\"test.example.com\"}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至名為 CreateDmsSeMongoParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2hxcllf1b4ey0"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateDmsSeMongoParams.json
```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"Description":          "MongoDB-DMS-SE",
"VpcId":                "VPC_ID",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"Name":                 "Test Mongo SE",
"TimeoutInMinutes":     60,
"Parameters":   {
    "Description":      "DESCRIPTION",
    "DatabaseName":       "mytestdb",
    "EngineName":       "mongodb",
    "ServerName":       "test.example.com",
    "Port":             "27017"
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsSeMongoRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeMongoRfc.json
```

4. 修改並儲存 CreateDmsSeMongoRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2hxcllf1b4ey0",
"Title":                "DMS_Source_MongoDB"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsSeMongoRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeMongoRfc.json --execution-
parameters file://CreateDmsSeMongoParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

> **Note**
>
> 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

AMS DMS 可以使用 Mongo 或任何關聯式資料庫服務 (RDS) 作為來源端點。如需 S3 來源端點，請參閱 S3 的 DMS 來源端點：建立。

S3 的 DMS 來源端點：建立

使用主控台建立 DMS S3 來源端點

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，並顯示使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 DMS S3 來源端點

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC：`aws amscm submit-rfc --rfc-id` *ID*命令。

   監控 RFC：`aws amscm get-rfc --rfc-id` *ID*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --title "S3DMSSourceEndpoint" --
aws-account-id ACCOUNT-ID --change-type-id ct-2oxl37nphsrjz --change-type-version 1.0
 --execution-parameters "{\"Description\":\"TestS3DMS-SE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3-DMS-SE\",\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-
demo-bucket\",\"S3ExternalTableDefinition\":\"{\\\"TableCount\\\":\\\"1\\\",\\\"Tables
\\\":[{\\\"TableName\\\":\\\"employee\\\",\\\"TablePath\\\":\\\"hr/employee/\\\",\\
\"TableOwner\\\":\\\"hr\\\",\\\"TableColumns\\\":[{\\\"ColumnName\\\":\\\"Id\\\",\\
\"ColumnType\\\":\\\"INT8\\\",\\\"ColumnNullable\\\":\\\"false\\\",\\\"ColumnIsPk\\\":
\\\"true\\\"},{\\\"ColumnName\\\":\\\"LastName\\\",\\\"ColumnType\\\":\\\"STRING\\\",
\\\"ColumnLength\\\":\\\"20\\\"},{\\\"ColumnName\\\":\\\"FirstName\\\",\\\"ColumnType
```

```
\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"30\\\"},{\\\"ColumnName\\\":\\\"HireDate\
\\\",\\\"ColumnType\\\":\\\"DATETIME\\\"},{\\\"ColumnName\\\":\\\"OfficeLocation\\\",\\
\"ColumnType\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"20\\\"}],\\\"TableColumnsTotal
\\\":\\\"5\\\"}]}\",\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789101:role/ams-
ops-ct-authors-dms-s3-test-role\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

範本建立：

1. 將此變更類型的執行參數輸出至名為 CreateDmsSeS3Params.json.

```
aws amscm get-change-type-version --change-type-id "ct-2oxl37nphsrjz" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeS3Params.json
```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為如下內容：

```
{
"Description":          "TestS3DMS-SE",
"VpcId":                "VPC_ID",
"Name":                 "S3-DMS-SE",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"TimeoutInMinutes":     60,
"Parameters":   {
    "EngineName":               "s3",
    "S3BucketName":              "amzn-s3-demo-bucket",
    "S3ExternalTableDefinition":  "BUCKET-NAME",
    {"TableCount":                "1",
      "Tables":[{"TableName":"employee","TablePath":"hr/
employee/","TableOwner":"hr","TableColumns":
[{"ColumnName":"Id","ColumnType":"INT8","ColumnNullable":"false","ColumnIsPk":"true"},
{"ColumnName":"LastName","ColumnType":"STRING","ColumnLength":"20"},
{"ColumnName":"FirstName","ColumnType":"STRING","ColumnLength":"30"},
{"ColumnName":"HireDate","ColumnType":"DATETIME"},
{"ColumnName":"OfficeLocation","ColumnType":"STRING","ColumnLength":"20"}],"TableColumnsTot
    "S3ServiceAccessRoleArn":      "arn:aws:iam::123456789101:role/ams-ops-ct-
authors-dms-s3-test-role",
      }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsSeS3Rfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeS3Rfc.json
```

4. 修改並儲存 CreateDmsSeS3Rfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2oxl37nphsrjz",
"Title":                "DMS_Source_S3"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsSeS3Rfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeS3Rfc.json --execution-
parameters file://CreateDmsSeS3Params.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

> **Note**
>
> 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

AMS DMS 可以使用 S3 或任何關聯式資料庫服務 (RDS) 來源端點。如需 Mongo 資料庫來源端點，請參閱 MongoDB 的 DMS 來源端點：建立。

4： AWS DMS 目標端點：建立、為 S3 建立

您可以使用 AMS 主控台或 API/CLI 為各種資料庫建立 AMS DMS 目標端點，我們提供兩個範例。

DMS 目標端點：建立

AMS DMS 可以使用 S3 或任何關聯式資料庫服務 (RDS) 搭配
MySQL、MariaDB、Oracle、Postgresql 或 Microsoft SQL 做為目標端點。

使用主控台建立 DMS 目標端點

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 DMS 目標端點

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何CreateRfc參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-3gf8dolbo8x9p" --change-type-version "1.0" --title "TestDMSTargetEndpoint" --
execution-parameters "{\"Description\":\"TestTE\",\"VpcId\":\"VPC-ID\",\"Name\":
\"TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes\":60,
\"Parameters\":{\"EngineName\":\"mysql\",\"Password\":\"testpw123\",\"Port\":\"3306\",
\"ServerName\":\"mytestdb.d5fga0rf2wpi.ap-southeast-2.rds.amazonaws.com\",\"Username\":
\"USERNAME\"}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至名為 CreateDmsTeParams.json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-3gf8dolbo8x9p" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeParams.json
   ```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為如下內容：

   ```
   {
   "Description":          "TestTE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-knghtmmgefafdq89u",
   "Name":                 "TE-NAME",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "EngineName":       "mysql",
       "ServerName":       "sql.db.example.com",
       "Port":             "3306",
   ```

```
    "Username":         "DB-USER",
    "Password":         "DB-PW",}
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsTeRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeRfc.json
```

4. 修改並儲存 CreateDmsTeRfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-3gf8dolbo8x9p",
"Title":                "DB-DMS-Target-Endpoint"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsTeRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeRfc.json --execution-
parameters file://CreateDmsTeParams.json
```

   您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

- 此變更類型現在位於 2.0 版。
- AMS DMS 可以使用 S3 或任何關聯式資料庫服務 (RDS) 搭配 MySQL、MariaDB、Oracle、Postgresql 或 Microsoft SQL 做為目標端點。如需 S3 目標端點，請參閱 S3 的 DMS 目標端點：建立。
- 如需詳細資訊，請參閱資料遷移的目標。
- 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

S3 的 DMS 目標端點：建立

使用主控台建立 DMS S3 目標端點

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 建立 DMS S3 目標端點

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何CreateRfc參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為如下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-05muqzievnxk5" --change-type-version "1.0" --title "TestDMSTargetEndpointS3"
 --execution-parameters "{\"Description\":\"TestS3TE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-demo-bucket\",
\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789123:role/my-s3-role\"}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 CreateDmsTeS3Params.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-05muqzievnxk5" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeS3Params.json
   ```

2. 修改並儲存執行參數 CreateDmsTeS3Params.json 檔案。例如，您可以將內容取代為如下內容：

   ```
   {
   "Description":          "TestS3DMS-TE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-knghtmmgefafdq89u",
   "Name":                 "DMS-S3-TE",
   "TimeoutInMinutes":     60,
   "Parameters":   {
   ```

```
    "EngineName":       "s3",
    "S3BucketName":      "amzn-s3-demo-bucket",
    "S3ServiceAccessRoleArn":       "arn:aws:iam::123456789101:role/ams-ops-ct-
authors-dms-s3-test-role"
      }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsTeS3Rfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeS3Rfc.json
```

4. 修改並儲存 CreateDmsTeS3Rfc.json 檔案。例如，您可以將內容取代為如下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-05muqzievnxk5",
"Title":                "DMS_Target_S3"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsTeS3Rfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeS3Rfc.json --execution-
parameters file://CreateDmsTeS3Params.json
```

   您會在回應中收到新 RFC 的 ID，並使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

> **Note**
>
> 您最多可以新增 50 個標籤，但若要這樣做，您必須啟用其他組態檢視。

AMS 提供個別的變更類型，用於建立 S3 的目標端點。如需詳細資訊，請參閱使用 Amazon S3 做為 AWS Database Migration Service 的目標和使用 Amazon S3 做為 AWS DMS 目標時的額外連線屬性。

5： AWS DMS 複寫任務：建立

您可以使用 AMS 主控台或 API/CLI 來建立 AMS AWS DMS 複寫任務。

建立 AWS DMS 複寫任務

使用主控台建立 AWS DMS 複寫任務

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 AWS DMS CLI 建立複寫任務

運作方式：

1. 使用內嵌建立 （您發出包含所有 RFC 和執行參數的`create-rfc`命令） 或範本建立 （您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

監控 RFC： `aws amscm get-rfc --rfc-id ID`命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-1d2fml15b9eth" --change-type-version "1.0" --title "TestDMSRepTask" --
execution-parameters "{\"Description\":\"TestRepTask\",\"VpcId\":\"VPC-ID\",\"Name
\":\"DMSRepTask\",\"Parameters\":{\"CdcStartTime\":\1533776569\"MigrationType\":
\"full-load\",\"ReplicationInstanceArn\":\"REP_INSTANCE_ARN\",\"SourceEndpointArn
\":\"SOURCE_ENDPOINT_ARN\",\"TableMappings\":\"{\\\"rules\\\": [{\\\"rule-type
\\\": \\\"selection\\\",\\\"rule-id\\\": \\\"1\\\",\\\"rule-name\\\": \\\"1\\
\",\\\"object-locator\\\": {\\\"schema-name\\\": \\\"Test\\\",\\\"table-name\\
\": \\\"%\\\"}, \\\"rule-action\\\": \\\"include\\\"}] }\",\"TargetEndpointArn
\":\"TARGET_ENDPOINT_ARN\"},\"StackTemplateId\":\"stm-eos7uq0usnmeggdet\",
\"TimeoutInMinutes\":60}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 CreateDmsRtParams.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-1d2fml15b9eth" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRtParams.json
   ```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"Description":          "DMSTestRepTask",
"VpcId":                "VPC_ID",
"StackTemplateId":      "stm-eos7uq0usnmeggdet",
"Name":                 "Test DMS RT",
"TimeoutInMinutes":     60,
"Parameters":   {
    "CdcStartTime":           "1533776569",
    "MigrationType":          "full-load",
    "ReplicationInstanceArn": "REP_INSTANCE_ARN",
    "SourceEndpointArn":      "SOURCE_ENDPOINT_ARN",
    "TargetEndpointArn":      "TARGET_ENDPOINT_ARN"
    "TableMappings":          {"rules": [{"rule-type": "selection","rule-id":
 "1","rule-name": "1","object-locator": {"schema-name": "Test","table-name": "%"},
 "rule-action": "include"}] }",
    }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 CreateDmsRtRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRtRfc.json
```

4. 修改並儲存 CreateDmsRtRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1d2fml15b9eth",
"Title":                "DMS-RI-Create-RFC"
}
```

5. 建立 RFC，指定執行參數檔案和 CreateDmsRtRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://CreateDmsRtRfc.json --execution-
parameters file://CreateDmsRtParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

您可以建立擷取三種不同變更或資料類型的 AWS DMS 任務。如需詳細資訊，請參閱使用 AWS DMS 任務、建立任務，以及使用 AWS DMS 建立持續複寫的任務。

AWS DMS 管理

AWS DMS 管理範例。

開始 AWS DMS 複寫任務

使用主控台啟動 AWS DMS 複寫任務

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立舊版選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填寫）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 啟動 AWS DMS 複寫任務

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-1yq7hhqse71yg" --change-type-version
 "1.0" --title "Start DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StartDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"],\"StartReplicationTaskType\":[\"start-
replication\"],\"CdcStartPosition\":[\"\"],\"CdcStopPosition\":[\"\"]}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 StartDmsRtParams.json:

```
aws amscm get-change-type-version --change-type-id "ct-1yq7hhqse71yg" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > StartDmsRtParams.json
```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
  "DocumentName": "AWSManagedServices-StartDmsTask",
  "Region": "us-east-1",
  "Parameters": {
      "ReplicationTaskArn": [
        "TASK_ARN"
      ],
      "StartReplicationTaskType": [
        "start-replication"
      ],
      "CdcStartPosition": [
        ""
      ],
      "CdcStopPosition": [
        ""
      ]
  }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 StartDmsRtRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > StartDmsRtRfc.json
```

4. 修改並儲存 StartDmsRtRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
  "ChangeTypeId": "ct-1yq7hhqse71yg",
  "ChangeTypeVersion": "1.0",
  "Title": "Start DMS Replication Task"
}
```

5. 建立 RFC，指定執行參數檔案和 StartDmsRtRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://StartDmsRtRfc.json --execution-
parameters file://StartDmsRtParams.json
```

您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

您可以使用 AMS 主控台或 AMS API/CLI 啟動 AWS DMS 複寫任務。如需詳細資訊，請參閱使用 AWS DMS 任務。

停止 AWS DMS 複寫任務

使用主控台停止 AWS DMS 複寫任務

AMS 主控台中此變更類型的螢幕擷取畫面：

運作方式：

1. 導覽至建立 RFC 頁面：在 AMS 主控台的左側導覽窗格中，按一下 RFCs以開啟 RFCs清單頁面，然後按一下建立 RFC。
2. 在預設瀏覽變更類型檢視中選擇熱門的變更類型 (CT)，或在依類別選擇檢視中選擇 CT。
   - 依變更類型瀏覽：您可以在快速建立區域中按一下熱門的 CT，以立即開啟執行 RFC 頁面。請注意，您無法透過快速建立選擇較舊的 CT 版本。

     若要排序 CTs，請使用卡片或資料表檢視中的所有變更類型區域。在任一檢視中，選取 CT，然後按一下建立 RFC 以開啟執行 RFC 頁面。如果適用，建立較舊版本選項會顯示在建立 RFC 按鈕旁。
   - 依類別選擇：選取類別、子類別、項目和操作，如果適用，CT 詳細資訊方塊會開啟，其中包含使用較舊版本建立的選項。按一下建立 RFC 以開啟執行 RFC 頁面。
3. 在執行 RFC 頁面上，開啟 CT 名稱區域以查看 CT 詳細資訊方塊。需要主旨 （如果您在瀏覽變更類型檢視中選擇 CT，則會為您填入）。開啟其他組態區域以新增 RFC 的相關資訊。

   在執行組態區域中，使用可用的下拉式清單或輸入必要參數的值。若要設定選用的執行參數，請開啟其他組態區域。
4. 完成後，請按一下執行。如果沒有錯誤，RFC 成功建立的頁面會顯示已提交的 RFC 詳細資訊，以及初始的執行輸出。
5. 開啟執行參數區域以查看您提交的組態。重新整理頁面以更新 RFC 執行狀態。或者，取消 RFC 或使用頁面頂端的選項建立 RFC 的副本。

使用 CLI 停止 AWS DMS 複寫任務

運作方式：

1. 使用內嵌建立（您發出包含所有 RFC 和執行參數的`create-rfc`命令）或範本建立（您建立兩個 JSON 檔案，一個用於 RFC 參數，另一個用於執行參數），並使用兩個檔案作為輸入發出`create-rfc`命令。此處說明這兩種方法。
2. 使用傳回的 RFC ID 提交 RFC： `aws amscm submit-rfc --rfc-id` *`ID`*命令。

   監控 RFC： `aws amscm get-rfc --rfc-id` *`ID`*命令。

若要檢查變更類型版本，請使用下列命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 您可以將任何`CreateRfc`參數與任何 RFC 搭配使用，無論它們是否為變更類型結構描述的一部分。例如，若要在 RFC 狀態變更時取得通知，請將此行新增至請求的 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 參數部分（而非執行參數）。如需所有 CreateRfc 參數的清單，請參閱 AMS 變更管理 API 參考。

內嵌建立：

使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號），然後提交傳回的 RFC ID。例如，您可以將內容取代為類似以下內容：

```
aws amscm create-rfc --change-type-id "ct-1vd3y4ygbqmfk" --change-type-version
 "1.0" --title "Stop DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StopDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"]}}"
```

範本建立：

1. 將此變更類型的執行參數輸出至 JSON 檔案；此範例會將其命名為 StopDmsRtParams.json:

```
aws amscm get-change-type-version --change-type-id "ct-1vd3y4ygbqmfk" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > StopDmsRtParams.json
```

2. 修改並儲存執行參數 JSON 檔案。例如，您可以將內容取代為類似以下內容：

```
{
  "DocumentName": "AWSManagedServices-StopDmsTask",
  "Region": "us-east-1",
  "Parameters": {
      "ReplicationTaskArn": [
        "TASK_ARN"
      ]
  }
}
```

3. 將 JSON 範本輸出至目前資料夾中的檔案；此範例會將其命名為 StopDmsRtRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > StopDmsRtRfc.json
```

4. 修改並儲存 StopDmsRtRfc.json 檔案。例如，您可以將內容取代為類似以下內容：

```
{
  "ChangeTypeId": "ct-1vd3y4ygbqmfk",
  "ChangeTypeVersion": "1.0",
  "Title": "Stop DMS Replication Task"
}
```

5. 建立 RFC，指定執行參數檔案和 StopDmsRtRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://StopDmsRtRfc.json --execution-
parameters file://StopDmsRtParams.json
```

   您會在回應中收到新 RFC 的 ID，並且可以使用它來提交和監控 RFC。在您提交之前，RFC 會保持在編輯狀態，不會啟動。

提示

您可以使用 AMS 主控台或 AMS API/CLI 停止 DMS 複寫任務。如需詳細資訊，請參閱使用 AWS DMS 任務。

資料庫 (DB) 匯入 AMS RDS for Microsoft SQL Server

> **Note**
>
> AMS API/CLI (amscm 和 amsskms) 端點位於 AWS N. Virginia 區域 `us-east-1`。根據身分驗證的設定方式，以及您的帳戶和資源所在的 AWS 區域，您可能需要在發出命令`--region us-east-1`時新增 。如果這是您的身分驗證方法`--profile saml`，您可能還需要新增 。

資料庫匯入至 AMS RDS for SQL Server，程序倚賴提交為變更請求 (RFCs) 的 AMS 變更類型 (CTs)，並使用 Amazon RDS API 參數做為輸入。MicroSoft SQL Server 是一種關聯式資料庫管理系統 (RDBMS)。若要進一步了解，另請參閱：Amazon Relational Database Service (Amazon RDS) 和 rds 或 Amazon RDS API 參考。

> **Note**
>
> 在繼續下一個步驟之前，請確定每個 RFC 都已成功完成。

高階匯入步驟：

1. 將來源現場部署 MS SQL 資料庫備份至 .bak （備份） 檔案
2. 將 .bak 檔案複製到傳輸 （加密） Amazon Simple Storage Service (S3) 儲存貯體
3. 將 .bak 匯入目標 Amazon RDS MS SQL 執行個體上的新資料庫

使用要求：

- AMS 中的 MS SQL RDS 堆疊
- 具有還原選項的 RDS 堆疊 (`SQLSERVER_BACKUP_RESTORE`)
- 傳輸 S3 儲存貯體
- 具有儲存貯體存取權的 IAM 角色，允許 Amazon RDS 擔任該角色
- 安裝了 MS SQL Management Studio 來管理 RDS 的 EC2 執行個體 （可以是現場部署的工作站）

設定

完成這些任務以開始匯入程序。

1. 提交 RFC 以使用部署 | 進階堆疊元件 | RDS 資料庫堆疊 | 建立 (ct-2z60dyvto9g6c) 建立 RDS 堆疊。請勿在建立請求中使用目標資料庫名稱 (RDSDBName 參數），目標資料庫會在匯入期間建立。請務必允許足夠的空間 (RDSAllocatedStorage 參數）。如需執行此操作的詳細資訊，請參閱 AMS 變更管理指南 RDS 資料庫堆疊 | 建立。
2. 使用部署 | 進階堆疊元件 | S3 儲存 | 建立 (ct-1a68ck03fn98r) 提交 RFC 以建立傳輸 S3 儲存貯體（如果尚不存在）。如需執行此作業的詳細資訊，請參閱 AMS 變更管理指南 S3 儲存 | 建立。
3. 提交管理 | 其他 | 其他 | 更新 (ct-1e1xtak34nx76) RFC 以實作`customer_rds_s3_role`具有下列詳細資訊的 ：

   在主控台中：

   - 主旨：「若要支援 MS SQL Server 資料庫匯入，請在此帳戶`customer_rds_s3_role`實作 。
   - Transit S3 儲存貯體名稱：*BUCKET_NAME*。
   - 聯絡資訊：*EMAIL*。

   使用 CLI 的 ImportDbParams.json 檔案：

   ```
   {
          "Comment": "{"Transit S3 bucket name":"BUCKET_NAME"}",
          "Priority": "High"
        }
   }
   ```

4. 提交管理 | 其他 | 其他 | 更新請求 AMS 的 RFC，以將 SQLSERVER_BACKUP_RESTORE選項設定為步驟 1 中建立的 RDS （在此請求中使用步驟 1 輸出中的堆疊 ID 和此請求中的 `customer_rds_s3_role` IAM 角色）。
5. 提交 RFC 以建立 EC2 執行個體 （您可以使用任何現有的 EC2 或內部部署工作站/執行個體），並在執行個體上安裝 Microsoft SQL Management Studio。

匯入資料庫

若要匯入資料庫 （資料庫），請遵循下列步驟。

1. 使用 MS SQL 原生備份和還原來備份來源內部部署資料庫 （請參閱支援 SQL Server 中的原生備份和還原)。由於執行該操作，您應該有一個 .bak （備份） 檔案。
2. 使用 AWS S3 CLI 或 AWS S3 主控台將 .bak 檔案上傳至 和現有的傳輸 S3 S3 儲存貯體。如需傳輸 S3 儲存貯體的資訊，請參閱使用加密保護資料。

3. 將 .bak 檔案匯入目標 RDS for SQL Server MS SQL 執行個體上的新資料庫 （如需類型的詳細資訊，請參閱 Amazon RDS for MySQL 執行個體類型）：

   a. 登入 EC2 執行個體 （內部部署工作站） 並開啟 MS SQL Management Studio

   b. 連線至在步驟 #1 中建立為先決條件的目標 RDS 執行個體。請依照此程序進行連線：連線至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體

   c. 使用新的結構化查詢語言 (SQL) 查詢啟動匯入 （還原） 任務 （如需 SQL 查詢的詳細資訊，請參閱 SQL 簡介）。目標資料庫名稱必須是新的 （請勿使用與您先前建立的資料庫相同的名稱）。未加密的範例：

   ```
   exec msdb.dbo.rds_restore_database
           @restore_db_name=TARGET_DB_NAME,

           @s3_arn_to_restore_from='arn:aws:s3:::BUCKET_NAME/FILENAME.bak';
   ```

   d. 在不同的視窗中執行此查詢，以定期檢查匯入任務的狀態：

   ```
   exec msdb.dbo.rds_task_status;
   ```

   如果狀態變更為失敗，請在訊息中尋找失敗詳細資訊。

清除

匯入資料庫後，您可能想要移除不必要的資源，請依照下列步驟進行。

1. 從 S3 儲存貯體刪除備份檔案 (.bak)。您可以使用 S3 主控台來執行此操作。如需從 S3 儲存貯體刪除物件的 CLI 命令，請參閱 AWS CLI 命令參考中的 rm。
2. 如果您不打算使用 S3 儲存貯體，請將其刪除。如需執行此操作的步驟，請參閱刪除堆疊。
3. 如果您不打算執行 MS SQL 匯入，請提交管理 | 其他 | 其他 | 更新 (ct-0xdawir96cy7k) RFC，並請求 AMS 刪除 IAM 角色 `customer_rds_s3_role`。

AMS 中的 Tier 和 Tie 應用程式部署

Tier 和 Tie 部署可讓您獨立使用個別 RFCs 建立、設定和部署堆疊的資源，並在您進行將堆疊元件與彼此建立關聯時，使用堆疊元件IDs。

例如，若要在負載平衡器和資料庫後方部署高可用性（備援）網站，請使用 Tier 和 Tie 方法，為資料庫、負載平衡器和兩個 EC2 執行個體或 Auto Scaling 群組提交 RFCs，並使用您建立的 ELB ID 設定 EC2 執行個體或 Auto Scaling 群組。

在資源部署之後，您可以提交安全群組建立變更，以允許資源與資料庫通訊。如需建立安全群組的詳細資訊，請參閱建立安全群組。

AMS 中的完整堆疊應用程式部署

完整堆疊部署可讓您使用 CT 提交 RFC，該 CT 可一次建立和設定所需的一切。例如，若要部署剛剛描述的高可用性網站 (EC2 執行個體、負載平衡器和資料庫），您會使用 CT 來建立和設定 Auto Scaling 群組、負載平衡器、資料庫，以及所有執行個體做為堆疊運作所需的安全群組設定。接下來將說明兩個執行此操作的 AMS CTs 範例。

- 高可用性兩層堆疊 (ct-06mjngx5flwto)：此變更類型可讓您建立堆疊並設定 Auto Scaling 群組、RDS 支援的資料庫、Load Balancer和 CodeDeploy 應用程式和組態。請注意，負載平衡器不被視為層，因為它以網路設備的形式跨多個應用程式共用，CodeDeploy 函數也被視為設備。此外，它會建立 CodeDeploy 部署群組（使用您提供 CodeDeploy 應用程式的名稱），可用於部署您的應用程式。會自動建立允許資源一起運作的安全群組設定。
- 高可用性單層堆疊 (ct-09t6q7j9v5hrn)：此變更類型可讓您建立堆疊並設定 Auto Scaling 群組和 Application Load Balancer。允許資源一起運作的安全群組設定會自動建立。

使用佈建變更類型 CTs)

AMS 負責您的受管基礎設施，若要進行變更，您必須提交具有正確 CT 分類（類別、子類別、項目和操作）的 RFC。本節說明如何尋找 CTs、判斷任何 是否適合您的需求，以及在沒有 CT 的情況下請求新的 CT。

查看現有的 CT 是否符合您的需求

確定要使用 AMS 部署的內容後，下一步是研究現有的 CTs和 CloudFormation 範本，以查看解決方案是否已存在。

建立 RFC 時，您必須指定 CT。您可以使用 AWS Management Console 或 AMS API/CLI。接下來將說明使用兩者的範例。

您可以使用 主控台或 API/CLI 來尋找變更類型 ID (CT) 或版本。有兩種方法：搜尋或選擇分類。對於這兩種選擇類型，您可以選擇最常使用、最近使用或按字母順序排序搜尋。

YouTube 影片：如何使用 AWS Managed Services CLI 建立 RFC，以及在哪裡可以找到 CT 結構描述？

在 AMS 主控台的 RFCs -> 建立 RFC 頁面上：

- 選取依變更類型瀏覽（預設值）時：
  - 使用快速建立區域從 AMS 最熱門CTs 中選取。按一下標籤，隨即開啟執行 RFC 頁面，並自動為您填入主旨選項。視需要完成其餘選項，然後按一下執行以提交 RFC。
  - 或者，向下捲動至所有變更類型區域，並開始在選項方塊中輸入 CT 名稱，您不需要具有確切或完整的變更類型名稱。您也可以輸入相關字詞，依變更類型 ID、分類或執行模式（自動或手動）搜尋 CT。

    選取預設卡檢視後，相符的 CT 卡會在您輸入時顯示，選取卡片並按一下建立 RFC。選取資料表檢視後，選擇相關的 CT，然後按一下建立 RFC。這兩種方法都會開啟執行 RFC 頁面。
- 或者，若要探索變更類型選擇，請按一下頁面頂端的依類別選擇，以開啟一系列的下拉式選項方塊。
- 選擇類別、子類別、項目和操作。該變更類型的資訊方塊會顯示頁面底部的面板。
- 當您準備好時，請按 Enter，並顯示相符的變更類型清單。
- 從清單中選擇變更類型。該變更類型的資訊方塊會出現在頁面底部。
- 在您擁有正確的變更類型之後，請選擇建立 RFC。

> **Note**
>
> 必須安裝 AMS CLI，這些命令才能運作。若要安裝 AMS API 或 CLI，請前往 AMS 主控台開發人員資源頁面。如需 AMS CM API 或 AMS SKMS API 的參考資料，請參閱《使用者指南》中的 AMS 資訊資源一節。您可能需要新增身分驗證`--profile`選項，例如 `aws amsskms` *`ams-cli-command`* `--profile SAML`。您可能還需要新增 `--region`選項，因為所有 AMS 命令都用盡 us-east-1；例如 `aws amscm` *`ams-cli-command`* `--region=us-east-1`。

> **Note**
>
> AMS API/CLI (amscm 和 amsskms) 端點位於 AWS N. Virginia 區域 `us-east-1`。根據身分驗證的設定方式，以及您的帳戶和資源所在的 AWS 區域，您可能需要在發出命令`--region us-east-1`時新增 。如果這是您的身分驗證方法`--profile saml`，您可能還需要新增 。

若要使用 AMS CM API 搜尋變更類型（請參閱 ListChangeTypeClassificationSummaries) 或 CLI：

您可以使用篩選條件或查詢來搜尋。ListChangeTypeClassificationSummaries 操作具有 `Category`、`Item`、 `Subcategory`和 的篩選條件選項`Operation`，但值必須完全符合現有的值。若要在使用 CLI 時獲得更靈活的結果，您可以使用 `--query`選項。

使用 AMS CM API/CLI 變更類型篩選

屬性	有效值	有效/預設條件	備註
ChangeTypeId	代表 ChangeTypeId 的任何字串（例如：ct-abc123xyz7890)	等於	如需變更類型 IDs，請參閱變更類型參考。 如需變更類型 IDs，請參閱尋找變更類型或 CSIO。
類別	任何自由格式文字	包含	不支援每個個別欄位中的規則運算式。不區分大小寫的搜尋
Subcategory			
項目			
作業			

1. 以下是列出變更類型分類的一些範例：

   下列命令會列出所有變更類型類別。

   ```
   aws amscm list-change-type-categories
   ```

   下列命令會列出屬於指定類別的子類別。

   ```
   aws amscm list-change-type-subcategories --category CATEGORY
   ```

   下列命令會列出屬於指定類別和子類別的項目。

   ```
   aws amscm list-change-type-items --category CATEGORY --subcategory SUBCATEGORY
   ```

2. 以下是使用 CLI 查詢搜尋變更類型的一些範例：

下列命令會搜尋項目名稱中包含 "S3" 的 CT 分類摘要，並以資料表形式建立類別、子類別、項目、操作和變更類型 ID 的輸出。

```
aws amscm list-change-type-classification-summaries --query
 "ChangeTypeClassificationSummaries [?contains(Item, 'S3')].
[Category,Subcategory,Item,Operation,ChangeTypeId]" --output table
```

```
+-------------------------------------------------------------+
|               ListChangeTypeClassificationSummaries         |
+----------+-------------------------+--+------+--------------+
|Deployment|Advanced Stack Components|S3|Create|ct-1a68ck03fn98r|
+----------+-------------------------+--+------+--------------+
```

3. 然後，您可以使用變更類型 ID 來取得 CT 結構描述並檢查參數。下列命令會將結構描述輸出至名為 CreateS3Params.schema.json.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3Params.schema.json
```

如需有關使用 CLI 查詢的資訊，請參閱如何使用 --query Option 篩選輸出和查詢語言參考 JMESPath Specification。

4. 在您擁有變更類型 ID 之後，建議您驗證變更類型的版本，以確保它是最新版本。使用此命令來尋找指定變更類型的版本：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CHANGE_TYPE_ID
```

若要尋找AutomationStatus特定變更類型的 ，請執行此命令：

```
aws amscm --profile saml get-change-type-version --change-type-id CHANGE_TYPE_ID --
query "ChangeTypeVersion.{AutomationStatus:AutomationStatus.Name}"
```

若要尋找ExpectedExecutionDurationInMinutes特定變更類型的 ，請執行此命令：

```
aws amscm --profile saml get-change-type-version --change-type-id ct-14027q0sjyt1h
 --query "ChangeTypeVersion.{ExpectedDuration:ExpectedExecutionDurationInMinutes}"
```

一旦找到您認為適當的 CT，請查看與其相關聯的執行參數 JSON 結構描述，以了解它是否解決您的使用案例。

使用此命令將 CT 結構描述輸出至以 CT 命名的 JSON 檔案；此範例會輸出建立 S3 儲存結構描述：

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
--query "ChangeTypeVersion.ExecutionInputSchema" --output text >
CreateBucketParams.json
```

讓我們仔細看看此結構描述提供的內容。

S3 儲存貯體建立結構描述

```
{
  "$schema": "http://json-schema.org/draft-04/sch
ema#",
"name": "Create S3 Storage
"description": "Use to create an Amazon Simple
 Storage Service stack.",
  "type": "object",
  "properties": {
    "Description": {
      "description": "The description of the
 stack.",
      "type": "string",
      "minLength": 1,
      "maxLength": 500
    },
    "VpcId": {
      "description": "ID of the VPC to create the S3
 Bucket in, in the form vpc-a1b2c3d4e5f67890e.",
      "type": "string",
      "pattern": "^vpc-[a-z0-9]{17}$"
    },
    "StackTemplateId": {
      "description": "Required value: stm-s2b72
beb000000000.",
      "type": "string",
      "enum": ["stm-s2b72beb000000000"]
    },
    "Name":{
      "description": "The name of the stack to
 create.",
```

結構描述從 CT ("description") 開始，它會告訴您結構描述的用途。在此情況下， 會建立 S3 儲存堆疊。

接下來，您有可指定的必要和選用屬性。會提供預設屬性值。所需的屬性會列在結構描述的結尾。

在 StackTemplateId 區域中，您會看到此 CT 和結構描述有一個特定的堆疊範本，其 ID 是必要的屬性值。

結構描述可讓您為建立的堆疊加上標籤，以供內部簿記之用。此外，備份等某些選項需要 Key：backup 和 Value：true 的標籤。如需深入資訊，請參閱標記您的 Amazon EC2 資源。

```
      "type": "string",
      "minLength": 1,
      "maxLength": 255
    },
    "Tags": {
      "description": "Up to seven tags (key/value
pairs) for the stack.",
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "Key": {
            "type": "string",
            "minLength": 1,
            "maxLength": 127
          },
          "Value": {
            "type": "string",
            "minLength": 1,
            "maxLength": 255
          }
        },
        "additionalProperties": false,
        "required": [
          "Key",
          "Value"
        ]
      },
      "minItems": 1,
      "maxItems": 7
    },
    "TimeoutInMinutes": {
      "description": "The amount of time, in minutes,
to allow for creation of the stack.",
      "type": "number",
      "minimum": 0,
      "maximum": 60
    },
    "Parameters": {
      "description": "Specifications for the
stack.",
      "type": "object",
      "properties": {
        "AccessControl": {
```

CT JSON 結構描述的參數區段是您提供執行參數的位置。

對於此結構描述，只有 ACL 和 BucketName 是必要的執行參數。

```
          "description": "The canned (predefined)
 access control list (ACL) to assign to the bucket.",
          "type": "string",
          "enum": [
            "Private",
            "PublicRead",
            "AuthenticatedRead",
            "BucketOwnerRead"
          ]
        },
        "BucketName": {
          "description": "A name for the bucket.
 The bucket name must contain only lowercase letters,
 numbers, periods (.), and hyphens (-).",
          "type": "string",
          "pattern": "^[a-z0-9]([-.a-z0-9]+)[a-z
0-9]$",
          "minLength": 3,
          "maxLength": 63
        }
      },
      "additionalProperties": false,
      "required": [
        "AccessControl",
        "BucketName"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "Description",
    "VpcId",
    "StackTemplateId",
    "Name",
    "TimeoutInMinutes",
    "Parameters"
  ]
}
```

請求新的 CT

檢查結構描述後，您可以決定它無法提供足夠的參數來建立您想要的部署。如果是這種情況，請檢查現有的 CloudFormation 範本，尋找更接近您想要的範本。一旦您知道所需的其他參數，請提交管理 | 其他 | 其他 | 建立 CT。

> **Note**
>
> 所有其他 | 其他建立和更新 CTs會收到 AMS 操作員的注意，他們將與您聯絡以討論新的 CT。

若要提交新 CT 的請求，請透過一般存取 AMS 主控台，AWS Management Console然後遵循下列步驟。

1. 在左側導覽中，按一下 RFCs。

   RFCs儀表板頁面隨即開啟。
2. 按一下 Create (建立)。

   建立變更請求頁面隨即開啟。
3. 在類別下拉式清單中選取管理，並為子類別和項目選取其他。針對操作，選擇建立。RFC 需要經過核准才能實作。
4. 輸入您想要 CT 的原因資訊，例如：根據現有的建立 S3 儲存體 CT，請求允許自訂 ACLs 的修改後建立 S3 儲存體 CT。這應該會產生新的 CT：部署 | 進階堆疊元件 | S3 儲存 | 建立 S3 自訂 ACL。這個新的 CT 可能是公有的。
5. 請按 Submit (提交)。

   RFC 儀表板上會顯示您的 RFC。

測試新的 CT

一旦 AWS Managed Services 建立了該新的 CT，您就可以透過提交 RFC 進行測試。如果您使用 AMS 將新的 CT 預先核准，您可以直接遵循標準 RFC 提交，並留意結果（如需提交 RFCs的詳細資訊，請參閱建立和提交 RFC)。如果新的 CT 未預先核准（您希望確保在未明確核准的情況下永遠不會執行），則每次您想要執行時，都需要與 AMS 討論其實作。

快速入門

主題

使用 AMS 變更類型的組合，您可以完成複雜的任務。

您可以使用 AMS 變更管理系統，為多帳戶登陸區域 (MALZ) 或單一帳戶登陸區域 (SALZ) 帳戶設定 AMS 資源排程器。程序會有所不同。此外，執行檔案傳輸和跨帳戶快照。

AMS Resource Scheduler 快速入門

使用此快速入門指南實作 AMS Resource Scheduler，這是一種標籤型執行個體排程器，可在 AMS Advanced 中節省成本。

AMS 資源排程器是以 AWS 執行個體排程器為基礎。

AMS Resource Scheduler 術語

開始之前，最好先熟悉 AMS Resource Scheduler 術語：

- 期間：每個排程必須至少包含一個期間 （ 定義執行個體應執行的時間 ）。排程可以包含多個期間。當排程中使用多個期間時，當至少一個期間規則為 true 時，資源排程器會套用適當的啟動動作。
- 時區：如需要在稍後參考的 DefaultTimezone 參數中使用的可接受時區值清單，請參閱 TZ 資料庫時區清單的 TZ 欄。
- 休眠：設定為啟用休眠且符合休眠需求的真實 EC2 執行個體時，會休眠 (suspend-to-disk)。檢查 EC2 主控台，以了解您的執行個體是否已啟用休眠。對執行 Amazon Linux 的已停止 Amazon EC2 執行個體使用休眠。
- 強制執行：設為 true 時，根據定義的排程，如果資源排程器在執行期間之外手動啟動，則會停止執行中的資源，如果資源在執行期間手動停止，則會啟動資源。
- retain_running：設為 true 時，如果執行個體是在期間開始之前手動啟動，則 會防止 Resource Scheduler 在執行期間結束時停止執行個體。例如，如果設定期間從上午 9 點到下午 5 點的執行個體在上午 9 點之前手動啟動，Resource Scheduler 不會在下午 5 點停止執行個體。
- ssm-maintenance-window：將 AWS Systems Manager 維護時段新增為排程的執行期間。當您指定與部署堆疊位於相同帳戶和 AWS 區域中的維護時段名稱，以排程您的 Amazon EC2 執行個體時，

如果沒有其他執行期間指定執行個體應執行，且維護事件已完成，則 Resource Scheduler 會在維護時段開始之前啟動執行個體，並在維護時段結束時停止執行個體。

Resource Scheduler 會使用您在初始組態期間指定的 AWS Lambda 頻率，判斷在維護時段開始執行個體之前多久。如果您將頻率 AWS CloudFormation 參數設定為 10 分鐘或更短，Resource Scheduler 會在維護時段前 10 分鐘啟動執行個體。如果您將頻率設定為大於 10 分鐘，Resource Scheduler 啟動執行個體的分鐘數與您指定的頻率相同。例如，如果您將 Systems Manager 維護時段頻率設定為 30 分鐘，資源排程器會在維護時段前 30 分鐘啟動執行個體。

如需詳細資訊，請參閱AWS Systems Manager 維護時段。

- override-status：暫時覆寫資源排程器設定的排程開始和停止動作。如果您將 欄位設定為執行中，資源排程器會啟動，但不會停止適用的執行個體。執行個體會執行，直到您手動停止為止。如果您將覆寫狀態設定為停止，資源排程器會停止，但不會啟動適用的執行個體。在您手動啟動執行個體之前，執行個體不會執行。

AMS 資源排程器實作

若要部署 AMS Resource 排程器解決方案，請遵循下列步驟。

1. 提交部署 | AMS 資源排程器 | 解決方案 | 部署 (ct-0ywnhc8e5k9z5) RFC，並提供下列參數：
   - SchedulingActive：是，用於啟用資源排程，否，用於停用。預設為是。
   - ScheduledServices：輸入以逗號分隔的服務清單來排程資源。有效值包括自動擴展、ec2 和 rds 的組合。預設為 autoscaling、ec2、rds。
   - TagName：將資源排程結構描述與服務資源建立關聯的標籤金鑰名稱。預設為排程。

     > **Note**
     >
     > 您的資源排程器部署只會在具有此標籤的資源上運作。

   - DefaultTimezone：用作預設時區的時區名稱，格式為 US/Pacific。預設為 UTC。
2. 收到確認步驟 1 中的 RFC 已成功執行後，您可以提交期間 | 新增變更類型。
3. 最後，提交 RFC，將排程新增至步驟 2 中建立的期間。使用排程 | 新增變更類型。

AMS Resource Scheduler 實作和使用FAQs

AMS Resource Scheduler 的常見問題。

問：如果我啟用休眠，但 EC2 執行個體不支援它，會發生什麼情況？

答：休眠將內容從執行個體記憶體 (RAM) 儲存到您的 Amazon Elastic Block Store (Amazon EBS) 根磁碟區。如果此欄位設定為 true，當 Resource Scheduler 停止執行個體時，執行個體會休眠。

如果您將 Resource Scheduler 設定為使用休眠，但您的執行個體未啟用休眠，或不符合休眠先決條件，Resource Scheduler 會記錄警告，且執行個體會在沒有休眠的情況下停止。如需詳細資訊，請參閱讓執行個體進入休眠。

問：如果我同時設定 override_status 和強制執行，會發生什麼情況？

答：如果您將 override_status 設定為執行，並將 enforcedto true （防止執行個體在執行期間之外手動啟動），Resource Scheduler 會停止執行個體。

如果您將 override_status 設定為已停止，並將 enforcedto true （防止執行個體在執行期間手動停止），Resource Scheduler 會重新啟動執行個體。

> **Note**
>
> 如果強制執行為 false，則會套用設定的覆寫行為。

問：部署 AMS 資源排程器之後，如何停用或啟用帳戶中的資源排程器？

答：若要停用或啟用 AMS 資源排程器：

- 若要停用：使用狀態建立 RFC | 停用。請務必將 SchedulerState 設定為 DISABLE
- 若要啟用：使用狀態建立 RFC | 啟用。請務必將 SchedulerState 設定為 ENABLE

問 如果 AMS 資源排程器期間落在我的修補維護時段內，會發生什麼情況？

答：Resource Scheduler 會根據其設定的排程運作。如果設定為在修補進行時停止執行個體，則會停止執行個體，除非修補時段在修補開始之前新增為排程的期間。換句話說，除非已設定指定的期間，否則 Resource Scheduler 不會自動啟動任何已停止的執行個體以進行修補。為了避免與修補維護時段衝突，請將修補配置的時段新增至 資源排程器排程做為期間。若要將期間新增至現有排程，請使用期間 | 新增來建立 RFC。

問 如果我需要為不同的 EC2 執行個體設定不同的排程，是否可以在我的帳戶中設定多個排程？

答：是，您可以建立多個排程。根據需求，每個排程可以有多個期間。在帳戶中啟用 AMS Resource Scheduler 時，會設定標籤金鑰。例如，如果標籤金鑰是「排程」，標籤值可能會根據對應至 AMS

Resource Scheduler 排程名稱的不同排程而有所不同。若要新增排程，您可以使用管理 | AMS 資源排程器 | 排程 | 新增 (ct-2bxelbn765ive) 變更類型來建立 RFC，請參閱排程 | 新增。

問：我可以在哪裡找到 AMS Resource Scheduler 支援的所有不同變更類型？

答：AMS 具有資源排程器變更類型，可將 AMS 資源排程器部署至您的帳戶；啟用或停用；定義、新增、更新和刪除要與其搭配使用的排程和期間；以及描述（取得詳細描述）排程和期間。

設定跨帳戶備份（區域內）

AWS Backup 支援將快照從一個帳戶複製到相同 AWS 區域內的另一個帳戶，只要兩個帳戶位於相同的 AWS Organization 中即可。例如，在 AMS Advanced 多帳戶登陸區域 (MALZ) 中，您可以使用此快速入門在相同的 AWS 區域內設定跨帳戶快照複本。

如需詳細資訊，請參閱 AWS Backup 和 AWS Organizations 帶來跨帳戶備份功能

您可以複製快照跨帳戶進行災難復原 (DR)。您可能需要將快照保留在相同的 AWS 區域內，但跨越帳戶邊界進行資料保護。

概觀：

在高階，這些是 AMS 中跨帳戶備份的步驟：

- 在託管 AMS 登陸區域的 AWS 區域中建立目的地帳戶以託管備份（步驟 1)
- 建立 KMS 金鑰以加密目的地帳戶中的備份（步驟 3)
- 在與您 AMS Advanced 登陸區域相同區域的目的地帳戶中建立備份保存庫（步驟 4)
- 在 管理帳戶中啟用跨帳戶設定（步驟 5)
- 建立或修改來源帳戶備份計劃和規則（步驟 6)

> **Note**
>
> 確定來源和目的地帳戶都位於相同的區域。如果您想要跨區域複製備份，請聯絡您的 CA 或 CSDM。

若要啟用和設定跨帳戶備份：

1. 建立目的地帳戶以託管備份；如果您已有此類帳戶，可以略過此步驟。若要建立帳戶，請使用部署 | 受管登陸區域 | 管理帳戶 | 建立應用程式帳戶 （ 使用 VPC) 變更類型 (ct-1zdasmc2ewzrs)，從您的管理付款人帳戶提交 RFC。
2. 【選用】 如果在來源帳戶 （ 例如 Prod) 中加密資源或快照，請與目的地帳戶共用用於加密的 KMS 金鑰。若要這樣做，請使用 管理 | 進階堆疊元件 | KMS 金鑰 | 更新變更類型 (ct-3ovo7px2vsa6n) 提交 RFC。
3. 在目的地帳戶中，建立用於 Backup Vault 加密的 KMS 金鑰。若要這樣做，請使用 部署 | 進階堆疊元件 | KMS 金鑰 | 建立 （ 自動 ） 變更類型 (ct-1d84keiri1jhg) 提交 RFC。
4. 在目的地帳戶中，使用先前建立的金鑰建立 Backup Vault。您可以使用 CFN 擷取自動化變更類型、部署 | 擷取 | 從 CloudFormation 範本堆疊 | 建立 (ct-36cn2avfrrj9v) 來建立 AWS Backup Vault。 CloudFormation 在相同的請求中，需要修改保存庫存取政策，以允許來源帳戶存取保存庫。(s) 以下是範例政策：

   備份保存庫的 CloudFormation 範本範例：

```
{
  "Description": "Test infrastructure",
  "Resources": {
  "BackupVaultForTesting": {
    "Type": "AWS::Backup::BackupVault",
    "Properties": {
      "BackupVaultName": "backup-vault-for-test",
      "EncryptionKeyArn" : "arn:aws:kms:us-east-2:123456789012:key/227d8xxx-
aefx-44ex-a09x-b90c487b4xxx",
        "AccessPolicy" : {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "AllowSrcAccountPermissionsToCopy",
              "Effect": "Allow",
              "Action": "backup:CopyIntoBackupVault",
              "Resource": "*",
              "Principal": {
                "AWS": ["arn:aws:iam::987654321098:root"]
              }
            }
          ]
        }
      }
    }
  }
```

```
}
```

5. 從您的管理付款人帳戶，啟用跨帳戶備份。若要這樣做，請使用管理 | AWS Backup | 備份計畫 | 啟用跨帳戶複製（管理帳戶） 變更類型 (ct-2yja7ihh30ply) 提交 RFC。
6. 最後，從來源備份的來源帳戶建立備份計劃的規則或規則，以管理備份以跨帳戶複製快照。若要這樣做，請使用部署 | AWS Backup | 備份計畫 | 建立變更類型 (ct-2hyozbpa0sx0m) 提交 RFC。如果您需要更新現有的備份計劃，請使用 Management | Other | Other | Update change type (ct-0xdawir96cy7k) 提交 RFC，並附上此資訊：

   1. 備份計劃名稱以及要更新的規則名稱。
   2. 目的地/ICE 帳戶備份文件庫 ARN。
   3. 您想要在目標 ICE 保存庫中保留快照的保留日/月。

教學課程

主題

- 主控台教學課程：高可用性兩層堆疊 (Linux/RHEL)
- 主控台教學課程：部署 Tier 和 Tie WordPress 網站
- CLI 教學課程：高可用性兩層堆疊 (Linux/RHEL)
- CLI 教學課程：部署 Tier 和 Tie WordPress 網站

下列教學課程詳細說明使用 CLI 和主控台，以及部署 Linux 或 RHEL Amazon EC2 Auto Scaling 群組 (ASG)，建立具有高可用性 (ct-06mjngx5flwto) 的雙層堆疊的步驟。類似的tier-and-tie教學會遵循每個（一個用於主控台，另一個用於 CLI)，這些教學使用不同的 CTs，其建立順序可讓您在建立資源時將資源繫結在一起。

您可以在 managedservices/latest/ctref/ Change Type Reference 中找到所有 CT 選項的描述，包括 ChangeTypeId。

主控台教學課程：高可用性兩層堆疊 (Linux/RHEL)

本節說明如何使用 AMS 主控台將高可用性 (HA) WordPress 網站部署至 AMS 環境。

> **Note**
>
> 此部署演練已在 AMZN Linux 和 RHEL 環境中進行測試。

任務和所需 RFCs的摘要：

1. 建立基礎設施 (HA 雙層堆疊)
2. 為 CodeDeploy 應用程式建立 S3 儲存貯體
3. 建立 WordPress 應用程式套件並將其上傳至 S3 儲存貯體
4. 使用 CodeDeploy 部署應用程式
5. 存取 WordPress 網站並登入以驗證部署
6. 向下拉動部署

您可以在 AMS 變更類型參考中找到所有 CT 選項的描述，包括 ChangeTypeId。

開始之前

部署 | 進階堆疊元件 | 高可用性兩層堆疊 | 建立 CT 會建立 Auto Scaling 群組、負載平衡器、資料庫，以及 CodeDeploy 應用程式名稱和部署群組（具有您提供應用程式相同的名稱）。如需 CodeDeploy 的資訊，請參閱什麼是 CodeDeploy？

本演練使用高可用性兩層堆疊 RFC，其中包含UserData並同時說明如何建立 CodeDeploy 可以部署的 WordPress 套件。

範例中UserData顯示的 透過查詢位於 https：//http://169.254.169.254/latest/meta-data/ 的 EC2 執行個體中繼資料服務，從執行中的執行個體中取得執行個體 ID、區域等執行個體中繼資料。使用者資料指令碼中的此行：`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`，會將中繼資料服務的可用區域名稱擷取到支援區域的 $REGION 變數，並使用它來完成下載 CodeDeploy 代理程式的 S3 儲存貯體 URL。169.254.169.254 IP 只能在 VPC 內路由（所有 VPCs都可以查詢服務）。如需 服務的資訊，請參閱執行個體中繼資料和使用者資料。另請注意，輸入為 UserData 的指令碼會以「根」使用者身分執行，不需要使用「sudo」命令。

此演練會將下列參數保留為預設值（顯示）：

- Auto Scaling 群組：`Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization, ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75`。
- Load Balancer：`HealthCheckInterval=30, HealthCheckTimeout=5`。
- 資料庫：`BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`。
- 應用程式：`DeploymentConfigName=CodeDeployDefault.OneAtATime`。

變數參數：

主控台提供開始時間的 ASAP 選項，此演練建議使用它。ASAP 會在通過核准後立即執行 RFC。

> **Note**
>
> 您可以選擇以不同於所示的方式設定許多參數。範例中顯示的參數值已經過測試，但可能不適合您。範例中只會顯示所需的值。應變更###字型中的值，因為這些值專屬於您的帳戶。

建立基礎設施

此程序使用高可用性雙層堆疊 CT，後面接著建立 S3 儲存體 CT。

在開始之前收集下列資料，可讓部署更快進行。

必要資料 HA 堆疊：

- AutoScalingGroup：
  - UserData：此值在本教學課程中提供。它包含用於設定 CodeDeploy 資源並啟動 CodeDeploy 代理程式的命令。
  - AMI-ID：此值決定 Auto Scaling 群組 (ASG) 將啟動的 EC2 執行個體作業系統。在您的帳戶中選取以「customer-」開頭的 AMI，並且是您想要的作業系統。在 AMS 主控台 VPCs -> VPCs詳細資訊頁面中尋找 AMI IDs。此逐步解說適用於設定為使用 Amazon Linux 或 RHEL AMI 的 ASGs。
- 資料庫：
  - 這些參數 DBEngine、En EngineVersion 和 LicenseModel 應根據您的情況設定，但範例中顯示的值已經過測試。教學課程分別使用這些值：*MySQL*、*8.0.16*、*general-public-license*。
  - 部署應用程式套件時，需要這些參數：DBName、MasterUserPassword 和 MasterUsername。教學課程分別使用這些值：*wordpressDB*、*p4ssw0rd*、*admin*。請注意，DBName 只能包含英數字元。
  - 當您輸入 RDS 資料庫的 MasterUsername 時，它會以純文字顯示，因此請儘快登入資料庫，並變更密碼以確保您的安全。
  - 對於 RDSSubnetIds請使用兩個私有子網路。在每個項目之後按「Enter」一次輸入一個項目。使用 尋找子網路 IDs 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-subnet-summaries) 或 AMS 主控台 VPCs -> VPC 詳細資訊頁面。
- LoadBalancer：
  - 將此參數公有設為 true，因為教學課程使用公有 ELB 子網路。

- ELBSubnetIds：使用兩個公有子網路。在每個項目之後按「Enter」一次輸入一個項目。使用 尋找子網路 IDs 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-subnet-summaries) 或 AMS 主控台 VPCs -> VPC 詳細資訊頁面。

- 應用程式：ApplicationName 值會設定 CodeDeploy 應用程式名稱和 CodeDeploy 部署群組名稱。您可以使用它來部署應用程式。它在帳戶中必須是唯一的。若要檢查您的帳戶是否有 CodeDeploy 名稱，請參閱 CodeDeploy 主控台。此範例使用 *WordPress*，但如果您將使用該值，請確定它尚未使用。

1. 啟動高可用性堆疊。

   a. 在建立 RFC 頁面上，從清單中選取類別部署、子類別標準堆疊、項目高可用性雙層堆疊和操作建立。

   b. 重要：選擇進階並如所示設定值。

      您只需輸入星號 (*) 選項的值，測試的值會顯示在範例中；您可以將不需要的空白選項保留空白。

   c. 對於 RFC 描述區段：

      ```
      Subject: WP-HA-2-Tier-RFC
      ```

   d. 在資源資訊區段中，設定 AutoScalingGroup、資料庫、LoadBalancer、應用程式和標籤的參數。

      此外，「AppName」標籤金鑰的目的是讓您輕鬆地在 EC2 主控台中搜尋 ASG 執行個體；您可以呼叫此標籤金鑰「Name」或任何其他您想要的金鑰名稱。請注意，您最多可以新增 50 個標籤。

      ```
      UserData:
          #!/bin/bash
          REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/
       | sed 's/[a-z]$//')
          yum -y install ruby httpd
          chkconfig httpd on
          service httpd start
          touch /var/www/html/status
          cd /tmp
          curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
          chmod +x ./install
          ./install auto
      ```

```
    chkconfig codedeploy-agent on
    service codedeploy-agent start
AmiId:                AMI-ID
Description:          WP-HA-2-Tier-Stack

Database:
    LicenseModel:     general-public-license (USE RADIO BUTTON)
    EngineVersion:    8.0.16
    DBEngine:         MySQL
    RDSSubnetIds:     PRIVATE_AZ1 PRIVATE_AZ2 (ENTER ONE AT A TIME PRESSING
 "ENTER" AFTER EACH)
    MasterUserPassword:  p4ssw0rd
    MasterUsername:      admin
    DBName:              wordpressDB

LoadBalancer:
    Public:               true (USE RADIO BUTTON)
    ELBSubnetIds:         PUBLIC_AZ1 PUBLIC_AZ2

Application:
    ApplicationName:      WordPress

Tags:
    Name:                 WP-Rhel-Stack
```

e. 完成後按一下提交。

2. 登入您建立的資料庫並變更密碼。
3. 啟動 S3 儲存貯體堆疊。

   在開始之前收集下列資料，可讓部署更快進行。

   必要資料 S3 儲存貯體：

   - VPC-ID：此值決定 S3 儲存貯體的位置。使用 尋找具有 AMS SKMS API 參考的 VPC IDs，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-vpc-summaries) 或 AMS 主控台 VPCs頁面中的報告索引標籤。
   - BucketName：此值會設定 S3 儲存貯體名稱，您可以使用它來上傳應用程式套件。它在帳戶區域必須是唯一的，且不能包含大寫字母。將您的帳戶 ID 包含在 BucketName 中不是必要項目，但之後更容易識別儲存貯體。若要查看帳戶中存在哪些 S3 儲存貯體名稱，請前往您帳戶的 Amazon S3 主控台。

a. 在建立 RFC 頁面上，從 RFC CT 挑選清單選取類別部署、子類別進階堆疊元件、項目 S3 儲存和操作建立。

b. 保留預設的基本選項，並如所示設定值。

```
Subject:              S3-Bucket-WP-HA-RFC
Description:          S3BucketForWordPressBundles
BucketName:           ACCOUNT_ID-BUCKET_NAME
AccessControl:        Private
VpcId:                VPC_ID
Name:                 S3-Bucket-WP-HA-Stack
TimeoutInMinutes:     60
```

c. 完成後按一下提交。使用此變更類型部署的儲存貯體允許完整讀取/寫入存取整個帳戶。

建立、上傳和部署應用程式

首先，建立 WordPress 應用程式套件，然後使用 CodeDeploy CTs 來建立和部署應用程式。

1. 下載 WordPress、擷取檔案並建立 ./scripts 目錄。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：貼`https://github.com/WordPress/WordPress/archive/master.zip`到瀏覽器視窗並下載 zip 檔案。

   建立要在其中組合套件的暫時目錄。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

   Windows：建立「WordPress」目錄，稍後您將使用目錄路徑。

2. 將 WordPress 來源解壓縮至「WordPress」目錄，並建立 ./scripts 目錄。

   Linux：

```
unzip master.zip -d /tmp/WordPress_Temp
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
rm -rf /tmp/WordPress_Temp
rm -f master
cd /tmp/WordPress
mkdir scripts
```

Windows：前往您建立的「WordPress」目錄，並在該處建立「scripts」目錄。

如果您在 Windows 環境中，請務必將指令碼檔案的中斷類型設定為 Unix (LF)。在記事本 ++ 中，這是視窗右下角的選項。

3. 在 WordPress 目錄中建立 CodeDeploy appspec.yml 檔案（如果複製範例，請檢查縮排，每個空間計數）。重要：確保「來源」路徑正確，可將 WordPress 檔案（在本例中為 WordPress 目錄中）複製到預期的目的地 (/var/www/html/WordPress)。在此範例中，appapppec.yml 檔案位於具有 WordPress 檔案的 目錄中，因此只需要 "/"。此外，即使您為 Auto Scaling 群組使用 RHEL AMI，也請保持原狀。Apppec.yml 檔案範例：

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. 在 WordPress ./scripts 目錄中建立 bash 檔案指令碼。

   首先，config_wordpress.sh使用下列內容建立 （ 如果您願意，可以直接編輯 wp-config.php 檔案 ）。

   > **Note**
   >
   > 將 *DBName* 取代為 HA 堆疊 RFC 中指定的值 （ 例如 wordpress)。
   > 將 *DB_MasterUsername* 取代為 HA 堆疊 RFC 中指定的MasterUsername值 （ 例如 admin)。
   > 將 *DB_MasterUserPassword* 取代為 HA Stack RFC 中指定的MasterUserPassword值 （ 例如 p4ssw0rd)。
   > 將 *DB_ENDPOINT* 取代為 HA Stack RFC 執行輸出中的端點 DNS 名稱 （ 例如 srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com)。您可以使用 GetRfc 操作 (CLI：get-rfc --rfc-id RFC_ID) 或在先前提交的 HA Stack RFC 的 AMS 主控台 RFC 詳細資訊頁面中找到此項目。

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 在相同的目錄中install_dependencies.sh，使用下列內容建立 ：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 會在啟動時安裝為使用者資料的一部分，以允許運作狀態檢查從頭開始運作。

6. 在相同的目錄中`start_server.sh`，使用下列內容建立 ：

   - 對於 Amazon Linux 執行個體，請使用：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 對於 RHEL 執行個體，請使用此 （額外的命令是允許 SELINUX 接受 WordPress 的政策）：

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. 在相同的目錄中`stop_server.sh`，使用下列內容建立 ：

   ```
   #!/bin/bash
   service httpd stop
   ```

8. 建立 zip 套件。

   Linux：

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows：前往您的「WordPress」目錄，選取所有檔案並建立 zip 檔案，請務必將其命名為 wordpress.zip。

1. 將應用程式套件上傳至 S3 儲存貯體

   套件必須就位，才能繼續部署堆疊。

您會自動存取您建立的任何 S3 儲存貯體執行個體。您可以透過堡壘（請參閱存取執行個體) 或透過 S3 主控台存取，並使用drag-and-drop上傳 CodeDeploy 套件，或瀏覽並選取檔案。

您也可以在 shell 視窗中使用以下命令；請確定您有 zip 檔案的正確路徑：

```
aws s3 cp wordpress/wordpress.zip s3://BUCKET_NAME/
```

2. 部署 WordPress CodeDeploy 應用程式套件

所需的資料碼部署應用程式部署：

- CodeDeployApplicationName：您提供 CodeDeploy 應用程式的名稱。
- CodeDeployGroupName：由於 CodeDeploy 應用程式和群組都是從您在 HA 堆疊 RFC 中提供 CodeDeploy 應用程式的名稱建立的，因此這是與 CodeDeployApplicationName 相同的名稱。
- S3Bucket體：您提供 S3 儲存貯體的名稱。
- S3BundleType 和 S3Key：這些是您部署的 WordPress 應用程式套件的一部分。
- VpcId：相關的 VPC。

a. 在建立 RFC 頁面上，從 RFC CT 挑選清單選取類別部署、子類別應用程式、項目 CodeDeploy 應用程式和操作部署。

b. 保留預設的基本選項，並如所示設定值。

> **Note**
>
> 參考先前建立的 CodeDeploy 應用程式、CodeDeploy 部署群組、S3 儲存貯體和套件。

```
Subject:                                WP-CD-Deploy-RFC
Description:                            DeployWordPress
S3Bucket:                               BUCKET_NAME
S3Key:                                  wordpress.zip
S3BundleType:                           zip
CodeDeployApplicationName:              WordPress
CodeDeployDeploymentGroupName:          WordPress
CodeDeployIgnoreApplicationStopFailures: false
RevisionType:                           S3
```

```
VpcId:                                  VPC_ID
Name:                                   WP-CD-Deploy-Op
TimeoutInMinutes:                       60
```

c. 完成後按一下提交。

驗證應用程式部署

導覽至先前建立負載平衡器的端點 (LoadBalancerCName)，並使用 WordPress 部署的路徑：/WordPress。例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

您應該會看到如下的頁面：

向下拉動高可用性部署

若要縮減部署，您可以針對 HA 兩層堆疊和 S3 儲存貯體提交刪除堆疊 CT，並請求刪除 RDS 快照（這些快照會在十天後自動刪除，但在那裡需要支付少量費用）。收集 HA 堆疊IDs，然後遵循下列步驟。 S3 請參閱堆疊 | 刪除。

主控台教學課程：部署 Tier 和 Tie WordPress 網站

本節說明如何使用 AMS 主控台將高可用性 (HA) WordPress 網站部署至 AMS 環境。這組指示包含建立必要 WordPress CodeDeploy 相容套件（例如 zip) 檔案的範例。資源的佈建遵循 順序，可讓您將資源繫結在一起以形成「tiers」。

> **Note**
>
> 此部署演練旨在與 AMZN Linux 作業系統搭配使用。
> 基本變數參數會標示為###；不過，您可能想要修改其他參數以符合您的情況。

任務和所需 RFCs的摘要：

1. 建立基礎設施：

a. 建立 MySQL RDS 資料庫叢集
b. 建立負載平衡器
c. 建立 Auto Scaling 群組並將其與負載平衡器綁定
d. 為 CodeDeploy 應用程式建立 S3 儲存貯體

2. 建立 WordPress 應用程式套件 （ 不需要 RFC)
3. 使用 CodeDeploy 部署 WordPress 應用程式套件：
   a. 建立 CodeDeploy 應用程式
   b. 建立 CodeDeploy 部署群組
   c. 將 WordPress 應用程式套件上傳至 S3 儲存貯體 （ 不需要 RFC)
   d. 部署 CodeDeploy 應用程式
4. 驗證部署
5. 向下拉動部署

您可以在 AMS 變更類型參考中找到所有 CT 選項的描述，包括 ChangeTypeId。

使用主控台建立 RFC （ 基本 ）

這些是每次使用主控台建立 RFC 時，您必須遵循的一些步驟。

1. 按一下左側導覽窗格中的 RFCs 以開啟 RFCs清單頁面，然後按一下建立 RFC。

   建立 RFC 頁面隨即開啟。
2. 選擇瀏覽變更類型 （ 預設 ） 或依類別選擇。
3. 瀏覽變更類型：

   a. 按一下快速建立選項，以其中一個最常用的變更類型開始 RFC。

      該變更類型的一般組態區域隨即開啟，主旨行已填入。若要查看變更類型詳細資訊，請開啟頁面頂端的區域。

   b. 使用所有變更類型區域。

      篩選、切換卡片或資料表檢視，或排序變更類型。當您找到所需的值時，請選取該值，然後按一下頁面頂端的建立 RFC。

      該變更類型的一般組態區域隨即開啟，主旨行已填入。若要查看變更類型詳細資訊，請開啟頁面頂端的區域。

4. 依類別選擇：

   a. 選取適當的類別、子類別、項目和操作。

      變更類型詳細資訊方塊會出現在頁面底部。

   b. 按一下頁面底部的建立 RFC。

   c. 該變更類型的一般組態區域隨即開啟，主旨行已填入。若要查看變更類型詳細資訊，請開啟頁面頂端的區域。

5. 為了確保某些人員收到 RFC 進度的通知，請填寫電子郵件地址。若要新增變更類型的詳細資訊，請填寫描述。開啟其他組態區域，以新增有關 RFC 的詳細資訊。
6. 針對排程，選取盡快執行此變更或排程此變更。如果您選取盡快執行此變更，RFC 會在通過核准後立即執行。如果您選取排程此變更類型，則會顯示挑選行事曆、時間和時區，並在提交後依排程啟動 RFC。
7. 在執行組態區域中，設定變更類型參數。若要查看選用參數，請開啟其他組態區域。
8. 準備就緒時，請按一下執行。

建立基礎設施

登入目標 AMS 帳戶的 AWS 主控台，然後登入帳戶的 AMS 主控台。

下列程序說明以您使用資源 IDs建置基礎設施的方式建立 RDS 資料庫、負載平衡器和 Auto Scaling 群組。

建立 RDS 堆疊

請參閱 RDS 堆疊 | 建立。

建立 ELB 堆疊

啟動公有 ELB。

必要資料：

- `VpcId`：您正在使用的 VPC，這應與先前使用的 VPC 相同。
- `ELBSubnetIds`：負載平衡器將分配流量的子網路陣列。選擇公有或私有子網路。使用 尋找子網路 IDs 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-subnet-summaries) 或 AMS 主控台 VPCs -> VPC 詳細資訊頁面。
- `VpcId`：您正在使用的 VPC，這應與先前使用的 VPC 相同。

1. 在建立 RFC 頁面上，選取類別部署、子類別進階堆疊元件、項目負載平衡器 (ELB) 堆疊，然後按一下建立。選擇進階，並接受所有預設值（包括沒有值的預設值），但接下來顯示的預設值除外。

```
Subject:                          WP-ELB-RFC
ELBSubnetIds:                     PUBLIC_AZ1
                                PUBLIC_AZ2
ELBScheme                         true
ELBCookieExpirationPeriod         600
VpcId:                            VPC_ID
Name:                             WP-Public-ELB
```

2. 完成後按一下提交。

建立 Auto Scaling 群組堆疊

啟動 Auto 擴展群組。

必要資料：

- VpcId：您正在使用的 VPC，這應與先前使用的 VPC 相同。
- AMI-ID：此值決定 Auto Scaling 群組 (ASG) 將啟動的 EC2 執行個體類型。請務必在您帳戶中選取以「customer-」開頭的 AMI，且為您想要的作業系統。使用 尋找 AMI IDs 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-amis) 或 AMS 主控台 VPCs VPCs 詳細資訊頁面。此逐步解說適用於設定為使用 Linux AMI 的 ASGs。
- ASGLoadBalancerNames：您先前建立的負載平衡器 - 透過查看 EC2 主控台 -> 負載平衡器（左側導覽中）尋找名稱。請注意，這不是您先前建立 ELB 時指定的 "Name"。

1. 在建立 RFC 頁面上，選取類別部署、子類別進階堆疊元件、項目自動擴展群組，然後按一下建立。選擇進階，並接受所有預設值（包括沒有值的預設值），但接下來顯示的預設值除外。

> **Note**
>
> 指定最新的 AMS AMI。指定先前建立的 ELB。

```
Subject:                                WP-ASG-RFC
```

```
ASGSubnetIds:                    PRIVATE_AZ1
                                                    PRIVATE_AZ2
ASGAmiId:                        AMI_ID
VpcId:                           VPC_ID
Name:                            WP_ASG
ASGLoadBalancerNames:            ELB_NAME
ASGUserData:
#!/bin/bash
REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed
 's/[a-z]$//')
yum -y install ruby httpd
chkconfig httpd on
service httpd start
touch /var/www/html/status
cd /tmp
curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
chmod +x ./install
./install auto
chkconfig codedeploy-agent on
service codedeploy-agent start
```

2. 完成後按一下提交。

建立 S3 堆疊

啟動 S3 儲存貯體。S3 儲存貯體是您上傳建立的應用程式套件的位置。

必要資料：

- VPC-ID：此值會決定 S3 儲存貯體的位置，這應該與先前使用的 VPC 相同。
- AccessControl：預設 AccessControl 清單 (ACL) 選項為 Private、 和 PublicRead。如需詳細資訊，請參閱 Amazon Simple Storage Service 固定 ACL。
- BucketName：此值會設定 S3 儲存貯體名稱，您可以使用它來上傳應用程式套件。它在帳戶區域必須是唯一的，且不能包含大寫字母。將您的帳戶 ID 包含在 BucketName 中不是必要項目，但之後更容易識別儲存貯體。若要查看帳戶中存在哪些 S3 儲存貯體名稱，請前往您帳戶的 Amazon S3 主控台。

1. 在建立 RFC 頁面上，選取類別部署、子類別進階堆疊元件、項目 S3 儲存，然後按一下建立。

   您可以將預設參數選項保留在 Basic 以接受預設，如所述。若要設定不同的值，請選擇進階。

> **Note**
>
> 使用此變更類型部署的儲存貯體允許完整讀取/寫入存取整個帳戶，可能需要新的變更類型，以允許更多受限的存取許可。

```
Subject:            S3-Bucket-RFC
BucketName:         ACCOUNT_ID-codedeploy-bundles
AccessControl:      Private

VpcId:              VPC_ID
Name:               S3BucketForWP
```

2. 完成後按一下提交。

建立 WordPress CodeDeploy 套件

本節提供建立應用程式部署套件的範例。

1. 下載 WordPress、解壓縮檔案並建立 ./scripts 目錄。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：貼`https://github.com/WordPress/WordPress/archive/master.zip`到瀏覽器視窗並下載 zip 檔案。

   建立要在其中組合套件的暫時目錄。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

   Windows：建立「WordPress」目錄，稍後您將使用目錄路徑。

2. 將 WordPress 來源解壓縮至「WordPress」目錄，並建立 ./scripts 目錄。

   Linux：

```
unzip master.zip -d /tmp/WordPress_Temp
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
rm -rf /tmp/WordPress_Temp
rm -f master
cd /tmp/WordPress
mkdir scripts
```

Windows：前往您建立的「WordPress」目錄，並在該處建立「scripts」目錄。

如果您在 Windows 環境中，請務必將指令碼檔案的中斷類型設定為 Unix (LF)。在記事本 ++ 中，這是視窗右下角的選項。

3. 在 WordPress 目錄中建立 CodeDeploy apppec.yml 檔案（如果複製範例，請檢查縮排，每個空格都會計算）。重要：確保「來源」路徑正確，可將 WordPress 檔案（在本例中為 WordPress 目錄中）複製到預期的目的地 (/var/www/html/WordPress)。在此範例中，appapppec.yml 檔案位於具有 WordPress 檔案的 目錄中，因此只需要 "/"。此外，即使您為 Auto Scaling 群組使用 RHEL AMI，也請保持原狀。Apppec.yml 檔案範例：

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. 在 WordPress ./scripts 目錄中建立 bash 檔案指令碼。

   首先，config_wordpress.sh使用下列內容建立 （ 如果您願意，可以直接編輯 wp-config.php 檔案 ）。

   > **Note**
   >
   > 將 *DBName* 取代為 HA 堆疊 RFC 中指定的值 （ 例如 wordpress)。
   > 將 *DB_MasterUsername* 取代為 HA 堆疊 RFC 中指定的MasterUsername值 （ 例如 admin)。
   > 將 *DB_MasterUserPassword* 取代為 HA 堆疊 RFC 中指定的MasterUserPassword值 （ 例如 p4ssw0rd)。
   > 將 *DB_ENDPOINT* 取代為 HA Stack RFC 執行輸出中的端點 DNS 名稱 （ 例如 srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com)。您可以使用 GetRfc 操作 (CLI：get-rfc --rfc-id RFC_ID) 或在先前提交的 HA Stack RFC 的 AMS 主控台 RFC 詳細資訊頁面中找到此項目。

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 在相同的目錄中install_dependencies.sh，使用下列內容建立 ：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 會在啟動時安裝為使用者資料的一部分，以允許從頭開始運作的運作狀態檢查。

6. 在相同的目錄中`start_server.sh`，使用下列內容建立 ：

   - 對於 Amazon Linux 執行個體，請使用：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 對於 RHEL 執行個體，請使用此 （額外的命令是允許 SELINUX 接受 WordPress 的政策）：

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. 在相同的目錄中`stop_server.sh`，使用下列內容建立 ：

   ```
   #!/bin/bash
   service httpd stop
   ```

8. 建立 zip 套件。

   Linux：

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows：前往您的「WordPress」目錄，選取所有檔案並建立 zip 檔案，請務必將其命名為 wordpress.zip。

使用 CodeDeploy 部署 WordPress 應用程式套件

CodeDeploy 是一種 AWS 部署服務，可將應用程式部署自動化至 Amazon EC2 執行個體。此程序的這個部分涉及建立 CodeDeploy 應用程式、建立 CodeDeploy 部署群組，然後使用 CodeDeploy 部署應用程式。

建立 CodeDeploy 應用程式

CodeDeploy 應用程式只是 AWS CodeDeploy 使用的名稱或容器，以確保在部署期間參考正確的修訂、部署組態和部署群組。在此情況下，部署組態是您先前建立的 WordPress 套件。

必要資料：

- `VpcId`：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- `CodeDeployApplicationName`：在帳戶中必須是唯一的。查看 CodeDeploy 主控台以檢查現有的應用程式名稱。

1. 建立適用於 WordPress 的 CodeDeploy 應用程式

   在建立 RFC 頁面上，從 RFC CT 挑選清單選取類別部署、子類別應用程式、項目 CodeDeploy 應用程式和操作建立。選擇基本，然後如所示設定值。完成後按一下提交。

   ```
   Subject:                        CD-WP-App-RFC
   CodeDeployApplicationName:      WordPress
   VpcId:                          VPC_ID
   Name:                           WP-CD-App
   ```

2. 完成後按一下提交。

建立 CodeDeploy 部署群組

建立 CodeDeploy 部署群組。

CodeDeploy 部署群組會定義一組以部署為目標的個別執行個體。

必要資料：

- `VpcId`：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- `CodeDeployApplicationName`：使用您先前建立的值。
- `CodeDeployAutoScalingGroups`：使用您先前建立的 Auto Scaling 群組名稱。

- CodeDeployDeploymentGroupName：部署群組的名稱。此名稱在與部署群組建立關聯的每個應用程式中都必須是獨一無二的。
- CodeDeployServiceRoleArn：使用範例中提供的公式。

1. 在建立 RFC 頁面上，選取類別部署、子類別應用程式、項目 CodeDeploy 部署群組，以及從 RFC CT 挑選清單中建立操作。選擇進階並設定顯示的值 (RFC 只需要主旨)。完成後按一下提交。

> **Note**
>
> 參考此格式的 CodeDeploy 服務角色 ARN，"arn:aws:iam::085398962942:role/aws-codedeploy-role"並使用先前為 "ASG_NAME" 建立的 Auto Scaling 群組名稱。

```
Description:                        Create CodeDeploy Deployment Group for WP
CodeDeployApplicationName:          WordPress
CodeDeployAutoScalingGroups:        ASG_NAME
CodeDeployDeploymentConfigName:     CodeDeployDefault.HalfAtATime
CodeDeployDeploymentGroupName:      WP CD Group
CodeDeployServiceRoleArn:           arn:aws:iam::ACCOUNT_ID:role/aws-codedeploy-role

VpcId:                              VPC_ID
Name:                               WP Deployment Group
```

2. 完成後按一下提交。

上傳 WordPress 應用程式

您會自動存取您建立的任何 S3 儲存貯體執行個體。您可以透過堡壘 （請參閱存取執行個體) 或透過 S3 主控台存取，然後上傳 CodeDeploy 套件。套件必須就位，才能繼續部署堆疊。此範例使用先前建立的儲存貯體名稱。

您可以使用此 AWS 命令來壓縮套件：

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

使用 CodeDeploy 部署 WordPress 應用程式

部署 CodeDeploy 應用程式。

必要資料：

- VPC-ID：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- CodeDeployApplicationName：使用您先前建立之 CodeDeploy 應用程式的名稱。
- CodeDeployDeploymentGroupName：使用您先前建立的 CodeDeploy 部署群組名稱。
- S3Location （您上傳應用程式套件的位置）S3Bucket：：您先前建立的 BucketName，S3BundleType以及S3Key：您放在 S3 存放區上套件的類型和名稱。

1. 部署 WordPress CodeDeploy 應用程式套件

   在建立 RFC 頁面上，從 RFC CT 挑選清單選取類別部署、子類別應用程式、項目 CodeDeploy 應用程式和操作部署。選擇基本，然後如所示設定值。完成後按一下提交。

   > **Note**
   >
   > 參考先前建立的 CodeDeploy 應用程式、CodeDeploy 部署群組、S3 儲存貯體和套件。

Subject:	WP-CD-Deploy-RFC
CodeDeployApplicationName:	*WordPress*
CodeDeployDeploymentGroupName:	*WPCDGroup*
RevisionType:	S3
S3Bucket:	*ACCOUNT_ID-codedeploy-bundles*
S3BundleType:	zip
S3Key:	wordpress.zip
VpcId:	*VPC_ID*
Name:	WordPress

2. 完成後按一下提交。

驗證應用程式部署

導覽至先前建立負載平衡器的端點 (ELB CName)，並使用 WordPress 部署的路徑：/WordPress。例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

向下傾斜應用程式部署

若要縮減部署，請針對 RDS 資料庫堆疊、應用程式負載平衡器、Auto Scaling 群組、S3 儲存貯體，以及 Code Deploy 應用程式和 group--ix RFCs提交 Delete Stack CT。此外，您可以為要刪除的 RDS 快照提交服務請求（它們會在 10 天後自動刪除，但在那裡需要支付少量費用）。收集所有 的堆疊 IDs，然後遵循下列步驟。請參閱堆疊 | 刪除。

CLI 教學課程：高可用性兩層堆疊 (Linux/RHEL)

本節說明如何使用 AMS CLI 將高可用性 (HA) 雙層堆疊部署至 AMS 環境。

> **Note**
>
> 此部署演練已在 AMZN Linux 和 RHEL 環境中進行測試。

任務和所需 RFCs的摘要：

1. 建立基礎設施 (HA 雙層堆疊）
2. 為 CodeDeploy 應用程式建立 S3 儲存貯體
3. 建立 WordPress 應用程式套件並將其上傳至 S3 儲存貯體
4. 使用 CodeDeploy 部署應用程式
5. 存取 WordPress 網站並登入以驗證部署

開始之前

部署 | 進階堆疊元件 | 高可用性 兩層堆疊進階 | 建立 CT 會建立 Auto Scaling 群組、負載平衡器、資料庫，以及 CodeDeploy 應用程式名稱和部署群組（與您提供應用程式的名稱相同）。如需 CodeDeploy 的資訊，請參閱什麼是 CodeDeploy？

本演練使用包含 UserData 的高可用性兩層堆疊（進階）RFC，並說明如何建立 CodeDeploy 可以部署的 WordPress 套件。

範例中`UserData`顯示的 透過查詢位於 https：//http://169.254.169.254/latest/meta-data/ 的 EC2 執行個體中繼資料服務，從執行中的執行個體中取得執行個體 ID、區域等執行個體中繼資料。使用者資料指令碼中的此行：`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`，會將中繼資料服務的可用區域

名稱擷取到支援區域的 $REGION 變數，並使用它來完成下載 CodeDeploy 代理程式的 S3 儲存貯體 URL。169.254.169.254 IP 只能在 VPC 內路由（所有 VPCs都可以查詢服務）。如需 服務的資訊，請參閱執行個體中繼資料和使用者資料。另請注意，輸入為 UserData 的指令碼會以「根」使用者身分執行，不需要使用「sudo」命令。

此演練會將下列參數保留為預設值（顯示）：

- Auto Scaling 群組：`Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization, ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75`。
- Load Balancer：`HealthCheckInterval=30, HealthCheckTimeout=5`。
- 資料庫：`BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`。
- 應用程式：`DeploymentConfigName=CodeDeployDefault.OneAtATime`。
- S3 儲存貯體：`AccessControl=Private`。

其他設定：

`RequestedStartTime RequestedEndTime` 如果您想要排程 RFC：您可以使用 Time.is 來判斷正確的 UTC 時間。所提供的範例必須適當調整。如果已超過開始時間，RFC 將無法繼續。或者，您可以將這些值保留為關閉，以建立 ASAP RFC，在通過核准後立即執行。

> **Note**
>
> 您可以選擇以不同於所示的方式設定許多參數。範例中顯示的參數值已經過測試，但可能不適合您。

建立基礎設施

在開始之前收集下列資料，可讓部署更快進行。

必要資料 HA 堆疊：

- AutoScalingGroup：
  - UserData：此值在本教學課程中提供。它包含用於設定 CodeDeploy 資源並啟動 CodeDeploy 代理程式的命令。
  - AMI-ID：此值決定 Auto Scaling 群組 (ASG) 將啟動的 EC2 執行個體類型。請務必在您帳戶中選取以「customer-」開頭的 AMI，且為您想要的作業系統。使用 尋找 AMI IDs 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 操作 (CLI：list-amis) 或 AMS 主控台 VPCs VPCs詳細資訊頁面。此逐步解說適用於設定為使用 Linux AMI 的 ASGs。
- 資料庫：
  - 這些參數 DBEngine、 和 LicenseModel應根據您的情況設定EngineVersion，但範例中顯示的值已經過測試。
  - 部署應用程式套件時，MasterUserPassword需要這些參數、MasterUsername、、RDSSubnetIds DBName和 。對於 RDSSubnetIds請使用兩個私有子網路。
- LoadBalancer：
  - 這些參數 DBEngine、 和 LicenseModel應根據您的情況設定EngineVersion，但範例中顯示的值已經過測試。
  - ELBSubnetIds：使用兩個公有子網路。
- 應用程式： ApplicationName值會設定 CodeDeploy 應用程式名稱和 CodeDeploy 部署群組名稱。您可以使用它來部署應用程式。它在帳戶中必須是唯一的。若要檢查您的帳戶是否有 CodeDeploy 名稱，請參閱 CodeDeploy 主控台。此範例使用「WordPress」，但如果您將使用該值，請確定該值尚未在使用中。

此程序使用高可用性雙層堆疊 （進階） CT (ct-06mjngx5flwto) 和建立 S3 儲存 CT (ct-1a68ck03fn98r)。從您的已驗證帳戶，遵循命令列的這些步驟。

1. 啟動基礎設施堆疊。

   a. 將 HA 兩層堆疊 CT 的執行參數 JSON 結構描述輸出到名為 CreateStackParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-06mjngx5flwto"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateStackParams.json
```

b. 修改結構描述。視需要取代*##*。例如，針對 ASG 將建立的 EC2 執行個體，使用您想要的作業系統。將 記錄ApplicationName為稍後用來部署應用程式的方式。請注意，您最多可以新增 50 個標籤。

```
{
"Description":      "HA two tier stack for WordPress",
"Name":             "WordPressStack",
"TimeoutInMinutes":  360,
"Tags": [
        {
            "Key": "ApplicationName",
            "Value": "WordPress"
        }
    ],
"AutoScalingGroup": {
            "AmiId":    "AMI-ID",
            "UserData": "#!/bin/bash \n
            REGION=$(curl 169.254.169.254/latest/meta-data/placement/
availability-zone/ | sed 's/[a-z]$//') \n
            yum -y install ruby httpd \n
            chkconfig httpd on \n
            service httpd start \n
            touch /var/www/html/status \n
            cd /tmp \n
            curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/
install \n
            chmod +x ./install \n
            ./install auto \n
            chkconfig codedeploy-agent on \n
            service codedeploy-agent start"
    },
    "LoadBalancer": {
        "Public":               true,
        "HealthCheckTarget":    "HTTP:80/status"
    },
    "Database":     {
        "DBEngine":             "MySQL",
        "DBName":               "wordpress",
```

```
        "EngineVersion":        "8.0.16 ",
        "LicenseModel":         "general-public-license",
        "MasterUsername":       "admin",
        "MasterUserPassword":   "p4ssw0rd"
    },
    "Application":  {
    "ApplicationName":  "WordPress"
        }
}
```

c. 將 CreateRfc JSON 範本輸出到目前資料夾中名為 CreateStackRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateStackRfc.json
```

d. 如下所示修改並儲存 RFC 範本，您可以刪除和取代內容。請注意，RequestedStartTime和 現在RequestedEndTime是選用的；排除它們會建立 ASAP RFC，在核准後立即執行（通常會自動執行）。若要提交排定的 RFC，請新增這些值。

```
{
"ChangeTypeVersion":    "3.0",
"ChangeTypeId":         "ct-06mjngx5flwto",
"Title":                "HA-Stack-For-WP-RFC"
}
```

e. 建立 RFC，指定 CreateStackRfc.json 檔案和 CreateStackParams.json 執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateStackRfc.json --execution-
parameters file://CreateStackParams.json
```

您會在回應中收到 RFC ID。儲存後續步驟的 ID。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id  RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

g. 若要檢查 RFC 狀態，請執行

```
aws amscm get-rfc --rfc-id RFC_ID
```

請記下 RFC ID。

2. 啟動 S3 儲存貯體

在開始之前收集下列資料，可讓部署更快進行。

必要資料 S3 儲存貯體：

- VPC-ID：此值會決定 S3 儲存貯體的位置。使用您先前使用的相同 VPC ID。
- BucketName：此值會設定 S3 儲存貯體名稱，您可以使用它來上傳應用程式套件。它在帳戶區域必須是唯一的，且不能包含大寫字母。將您的帳戶 ID 包含在 BucketName 中不是必要項目，但之後更容易識別儲存貯體。若要查看帳戶中存在哪些 S3 儲存貯體名稱，請前往您帳戶的 Amazon S3 主控台。

a. 將 S3 儲存體的 JSON 結構描述輸出至名為 CreateS3StoreParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3StoreParams.json
```

b. 如下所示修改結構描述，您可以刪除和取代內容。適當地取代 *VPC_ID*。範例中的值已經過測試，但可能不適合您。

> **Tip**
>
> BucketName 必須是帳戶區域中唯一的，且不能包含大寫字母。將您的帳戶 ID 包含在 BucketName 中不是必要項目，但之後更容易識別儲存貯體。若要查看帳戶中存在哪些 S3 儲存貯體名稱，請前往您帳戶的 Amazon S3 主控台。

```
{
"Description":      "S3BucketForWordPressBundle",
"VpcId":            "VPC_ID",
"StackTemplateId":  "stm-s2b72beb000000000",
"Name":             "S3BucketForWP",
"TimeoutInMinutes":  60,
"Parameters":   {
    "AccessControl":    "Private",
    "BucketName":       "ACCOUNT_ID-BUCKET_NAME"
```

```
    }
}
```

c. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中名為 CreateS3StoreRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateS3StoreRfc.json
```

d. 修改並儲存 CreateS3StoreRfc.json 檔案，您可以刪除並取代內容。請注意，RequestedStartTime和 現在RequestedEndTime是選用的；排除它們會建立 ASAP RFC，在核准後立即執行（通常會自動執行）。若要提交排定的 RFC，請新增這些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1a68ck03fn98r",
"Title":                "S3-Stack-For-WP-RFC"
}
```

e. 建立 RFC，指定 CreateS3StoreRfc.json 檔案和 CreateS3StoreParams.json 執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateS3StoreRfc.json  --
execution-parameters file://CreateS3StoreParams.json
```

您會在回應中收到新 RFC 的 RfcId。儲存後續步驟的 ID。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

g. 若要檢查 RFC 狀態，請執行

```
aws amscm get-rfc --rfc-id RFC_ID
```

建立、上傳和部署應用程式

首先，建立 WordPress 應用程式套件，然後使用 CodeDeploy CTs 來建立和部署應用程式。

1. 下載 WordPress、擷取檔案並建立 ./scripts 目錄。

Linux 命令：

```
wget https://github.com/WordPress/WordPress/archive/master.zip
```

Windows：貼https://github.com/WordPress/WordPress/archive/master.zip到瀏覽器視窗並下載 zip 檔案。

建立要在其中組合套件的暫時目錄。

Linux：

```
mkdir /tmp/WordPress
```

Windows：建立「WordPress」目錄，稍後您將使用目錄路徑。

2. 將 WordPress 來源解壓縮至「WordPress」目錄，並建立 ./scripts 目錄。

   Linux：

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows：前往您建立的「WordPress」目錄，並在該處建立「scripts」目錄。

   如果您在 Windows 環境中，請務必將指令碼檔案的中斷類型設定為 Unix (LF)。在記事本 ++ 中，這是視窗右下角的選項。

3. 在 WordPress 目錄中建立 CodeDeploy appspec.yml 檔案（如果複製範例，請檢查縮排，每個空間計數）。重要：確保「來源」路徑正確，可將 WordPress 檔案（在本例中為 WordPress 目錄中）複製到預期的目的地 (/var/www/html/WordPress)。在此範例中，appapppec.yml 檔案位於具有 WordPress 檔案的 目錄中，因此只需要 "/"。此外，即使您為 Auto Scaling 群組使用 RHEL AMI，也請保持原狀。Apppec.yml 檔案範例：

   ```
   version: 0.0
   os: linux
   files:
     - source: /
       destination: /var/www/html/WordPress
   ```

```
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. 在 WordPress ./scripts 目錄中建立 bash 檔案指令碼。

   首先，config_wordpress.sh使用下列內容建立（如果您願意，可以直接編輯 wp-config.php 檔案）。

   > **Note**
   >
   > 將 *DBName* 取代為 HA 堆疊 RFC 中指定的值（例如 wordpress)。
   > 將 *DB_MasterUsername* 取代為 HA 堆疊 RFC 中指定的MasterUsername值（例如 admin)。
   > 將 *DB_MasterUserPassword* 取代為 HA Stack RFC 中指定的MasterUserPassword值（例如 p4ssw0rd)。
   > 將 *DB_ENDPOINT* 取代為 HA Stack RFC 執行輸出中的端點 DNS 名稱（例如 srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com)。您可以使用 GetRfc 操作 (CLI：get-rfc --rfc-id RFC_ID) 或在先前提交的 HA Stack RFC 的 AMS 主控台 RFC 詳細資訊頁面中找到此項目。

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
```

```
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 在相同的目錄中`install_dependencies.sh`，使用下列內容建立 ：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 會在啟動時安裝為使用者資料的一部分，以允許運作狀態檢查從頭開始運作。

6. 在相同的目錄中`start_server.sh`，使用下列內容建立 ：

   - 對於 Amazon Linux 執行個體，請使用：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 對於 RHEL 執行個體，請使用此 （額外的命令是允許 SELINUX 接受 WordPress 的政策）：

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. 在相同的目錄中`stop_server.sh`，使用下列內容建立 ：

```
#!/bin/bash
service httpd stop
```

8. 建立 zip 套件。

   Linux：

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

Windows：前往您的「WordPress」目錄，選取所有檔案並建立 zip 檔案，請務必將其命名為 wordpress.zip。

1. 將應用程式套件上傳至 S3 儲存貯體。

   套件必須備妥，才能繼續部署堆疊。

   您會自動存取您建立的任何 S3 儲存貯體執行個體。您可以透過堡壘或透過 S3 主控台存取，並使用drag-and-drop或瀏覽上傳 WordPress 套件至 ，然後選取 zip 檔案。

   您也可以在 shell 視窗中使用以下命令；請確定您有 zip 檔案的正確路徑：

   ```
   aws s3 cp wordpress.zip s3://BUCKET_NAME/
   ```

2. 部署 WordPress 應用程式套件。

   在開始之前收集下列資料，可讓部署更快進行。

   必要資料：

   - VPC-ID：此值會決定 S3 儲存貯體的位置。使用您先前使用的相同 VPC ID。
   - CodeDeployApplicationName 和 CodeDeployApplicationName：您在 HA 2-Tier堆疊 RFC 中使用的ApplicationName值會設定 CodeDeployApplicationName 和 CodeDeployDeploymentGroupName。此範例使用「WordPress」，但您可能已使用不同的值。
   - S3Location：對於 S3Bucket，請使用您先前建立BucketName的 。S3BundleType 和 S3Key來自您放在 S3 存放區的套件。

   a. 輸出 CodeDeploy 應用程式的執行參數 JSON 結構描述，將 CT 部署到名為 DeployCDAppParams.json.

      ```
      aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb"
       --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
       DeployCDAppParams.json
      ```

b. 如下所示修改結構描述並將其儲存為 ，您可以刪除並取代內容。

```
{
"Description":                          "DeployWPCDApp",
"VpcId":                                "VPC_ID",
"Name":                                 "WordPressCDAppDeploy",
"TimeoutInMinutes":                     60,
"Parameters":   {
    "CodeDeployApplicationName":                "WordPress",
    "CodeDeployDeploymentGroupName":            "WordPress",
    "CodeDeployIgnoreApplicationStopFailures":   false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket":     "BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key":        "wordpress.zip" }
        }
    }
}
```

c. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中名為 DeployCDAppRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

d. 修改並儲存 DeployCDAppRfc.json 檔案，您可以刪除和取代內容。請注意，RequestedStartTime和 現在RequestedEndTime是選用的；排除它們會建立 ASAP RFC，在核准後立即執行 （通常會自動執行）。若要提交排定的 RFC，請新增這些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-RFC"
}
```

e. 建立 RFC，指定 DeployCDAppRfc 檔案和 DeployCDAppParams 執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

您會在回應中收到新 RFC 的 RfcId。儲存後續步驟的 ID。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

g. 若要檢查 RFC 狀態，請執行

```
aws amscm get-rfc --rfc-id RFC_ID
```

驗證應用程式部署

導覽至先前建立負載平衡器的端點 (ELB CName)，並使用 WordPress 部署的路徑：/WordPress。例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

向下傾斜應用程式部署

完成教學課程後，您會想要拆解部署，才不會向您收取資源的費用。

以下是一般堆疊刪除操作。您會想要提交兩次，一次用於 HA 2-Tier堆疊，一次用於 S3 儲存貯體堆疊。最後，請提交服務請求，刪除 S3 儲存貯體的所有快照 （包括服務請求中的 S3 儲存貯體堆疊 ID)。它們會在 10 天後自動刪除，但提早刪除會節省一些成本。

本演練提供使用 AMS 主控台刪除 S3 堆疊的範例；此程序適用於使用 AMS 主控台刪除任何堆疊。

> **Note**
>
> 如果刪除 S3 儲存貯體，必須先清空物件。

必要資料：

- `StackId`：要使用的堆疊。您可以查看 AMS 主控台堆疊頁面，透過左側導覽中的連結取得。使用 AMS SKMS API/CLI，執行 AMS SKMS API 參考，請參閱 AWS Artifact Console. 操作中的報告索引標籤 (`list-stack-summaries` CLI 中的)。
- 此演練的變更類型 ID 為 `ct-0q0bic0ywqk6c`，版本為 "1.0"，若要了解最新版本，請執行此命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

內嵌建立：

- 使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號）。E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- 使用建立 RFC 操作中傳回的 RFC ID 提交 RFC。在提交之前，RFC 會保持 Editing 狀態，且不會採取任何動作。

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- 監控 RFC 狀態並檢視執行輸出：

```
aws amscm get-rfc --rfc-id RFC_ID
```

範本建立：

1. 將 RFC 範本輸出到目前資料夾中的檔案；範例將其命名為 DeleteStackRfc.json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
   ```

2. 修改並儲存 DeleteStackRfc.json 檔案。由於刪除堆疊只有一個執行參數，因此執行參數可以位於 DeleteStackRfc.json 檔案本身（不需要使用執行參數建立單獨的 JSON 檔案）。

   ExecutionParameters JSON 延伸中的內部引號必須以反斜線 (\) 逸出。沒有開始和結束時間的範例：

   ```
   {
   "ChangeTypeVersion":    "1.0",
   "ChangeTypeId":         "ct-0q0bic0ywqk6c",
   "Title":                "Delete-My-Stack-RFC"
   "ExecutionParameters":  "{
           \"StackId\":\"STACK_ID\"}"
   }
   ```

3. 建立 RFC：

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

您會在回應中收到新 RFC 的 RfcId。例如：

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

儲存後續步驟的 ID。

4. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您在命令列不會收到確認。

5. 若要監控請求的狀態和檢視執行輸出：

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

CLI 教學課程：部署 Tier 和 Tie WordPress 網站

本節說明如何使用 AMS CLI 將高可用性 (HA) WordPress 網站部署至 AMS 環境。這組指示包含建立必要 WordPress CodeDeploy 相容套件 （ 例如 zip) 檔案的範例。

> **Note**
>
> 此部署演練旨在與 AMZN Linux 環境搭配使用。
> 基本變數參數會標示為###；不過，您可能想要修改其他參數以符合您的情況。

任務和必要 RFCs的摘要：

1. 建立基礎設施：
   a. 建立 RDS Stack (CLI)
   b. 建立負載平衡器

c. 建立 Auto Scaling 群組並將其與負載平衡器綁定
d. 為 CodeDeploy 應用程式建立 S3 儲存貯體
2. 建立 WordPress 應用程式套件 （ 不需要 RFC)
3. 使用 CodeDeploy 部署 WordPress 應用程式套件：
a. 建立 CodeDeploy 應用程式
b. 建立 CodeDeploy 部署群組
c. 將 WordPress 應用程式套件上傳至 S3 儲存貯體 （ 不需要 RFC)
d. 部署 CodeDeploy 應用程式
4. 驗證部署
5. 向下拉動部署

遵循來自已驗證帳戶的命令列中的所有步驟。

使用 CLI 建立 RFC

如需建立 RFCs的詳細資訊，請參閱建立 RFCs；如需常見 RFC 參數的說明，請參閱 RFC 常見參數。

建立基礎設施

下列程序說明以您使用資源 IDs建置基礎設施的方式建立 RDS 資料庫、負載平衡器和 Auto Scaling 群組。

建立 RDS Stack (CLI)

請參閱 RDS 堆疊 | 建立。

建立 ELB 堆疊

啟動公有負載平衡器 (ELB)。請參閱Load Balancer (ELB) 堆疊 | 建立。

建立 Auto Scaling 群組堆疊

啟動 Auto 擴展群組。

請參閱 Auto Scaling 群組 | 建立。

建立 S3 存放區

啟動 S3 儲存貯體。S3 儲存貯體是您上傳建立的應用程式套件的位置。請參閱 S3 儲存 | 建立。

為 CodeDeploy 建立 WordPress 應用程式套件

本節提供建立應用程式部署套件的範例。

1. 下載 WordPress、解壓縮檔案並建立 ./scripts 目錄。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：貼`https://github.com/WordPress/WordPress/archive/master.zip`入瀏覽器視窗並下載 zip 檔案。

   建立要在其中組合套件的暫時目錄。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

   Windows：建立「WordPress」目錄，稍後您將使用目錄路徑。
2. 將 WordPress 來源解壓縮至「WordPress」目錄，並建立 ./scripts 目錄。

   Linux：

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows：前往您建立的「WordPress」目錄，並在該處建立「scripts」目錄。

   如果您在 Windows 環境中，請務必將指令碼檔案的中斷類型設定為 Unix (LF)。在記事本 ++ 中，這是視窗右下角的選項。
3. 在 WordPress 目錄中建立 CodeDeploy appspec.yml 檔案（如果複製範例，請檢查縮排，每個空間計數）。重要：確保「來源」路徑正確，可將 WordPress 檔案（在本例中為 WordPress 目錄中）複製到預期的目的地 (/var/www/html/WordPress)。在此範例中，appapppec.yml 檔案位於

具有 WordPress 檔案的 目錄中，因此只需要 "/"。此外，即使您為 Auto Scaling 群組使用 RHEL AMI，也請保持原狀。Apppec.yml 檔案範例：

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. 在 WordPress ./scripts 目錄中建立 bash 檔案指令碼。

   首先，`config_wordpress.sh`使用下列內容建立（如果您願意，可以直接編輯 wp-config.php 檔案）。

   > **Note**
   >
   > 將 *DBName* 取代為 HA 堆疊 RFC 中指定的值（例如 `wordpress`)。
   > 將 *DB_MasterUsername* 取代為 HA 堆疊 RFC 中指定的`MasterUsername`值（例如 `admin`)。
   > 將 *DB_MasterUserPassword* 取代為 HA 堆疊 RFC 中指定的`MasterUserPassword`值（例如 `p4ssw0rd`)。
   > 將 *DB_ENDPOINT* 取代為 HA Stack RFC 執行輸出中的端點 DNS 名稱（例如 `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com`)。您可以使用

GetRfc 操作 (CLI：get-rfc --rfc-id RFC_ID) 或在先前提交的 HA Stack RFC 的 AMS 主控台 RFC 詳細資訊頁面中找到此項目。

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 在相同的目錄中`install_dependencies.sh`，使用下列內容建立 ：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 會在啟動時安裝為使用者資料的一部分，以允許運作狀態檢查從頭開始。

6. 在相同的目錄中`start_server.sh`，使用下列內容建立 ：

- 對於 Amazon Linux 執行個體，請使用：

```
#!/bin/bash
service httpd start
```

- 對於 RHEL 執行個體，請使用此 （額外的命令是允許 SELINUX 接受 WordPress 的政策）：

```
#!/bin/bash
setsebool -P  httpd_can_network_connect_db 1
setsebool -P  httpd_can_network_connect 1
chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
restorecon -Rv /var/www/html
```

```
service httpd start
```

7. 在相同的目錄中stop_server.sh，使用下列內容建立 ：

```
#!/bin/bash
service httpd stop
```

8. 建立 zip 套件。

   Linux：

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

   Windows：前往您的「WordPress」目錄，選取所有檔案並建立 zip 檔案，請務必將其命名為 wordpress.zip。

使用 CodeDeploy 部署 WordPress 應用程式套件

CodeDeploy 是一種 AWS 部署服務，可將應用程式部署自動化至 Amazon EC2 執行個體。此程序的這個部分涉及建立 CodeDeploy 應用程式、建立 CodeDeploy 部署群組，然後使用 CodeDeploy 部署應用程式。

建立 CodeDeploy 應用程式

CodeDeploy 應用程式只是 AWS CodeDeploy 使用的名稱或容器，以確保在部署期間參考正確的修訂、部署組態和部署群組。在此情況下，部署組態是您先前建立的 WordPress 套件。

必要資料：

- VpcId：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- CodeDeployApplicationName：在帳戶中必須是唯一的。查看 CodeDeploy 主控台以檢查現有的應用程式名稱。
- ChangeTypeId 和 ChangeTypeVersion：此演練的變更類型 ID 為 ct-0ah3gwb9seqk2，若要了解最新版本，請執行此命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0ah3gwb9seqk2
```

1. 將 CodeDeploy 應用程式 CT 的執行參數 JSON 結構描述輸出到目前資料夾中的檔案；範例將其命名為 CreateCDAppParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
```

2. 修改並儲存 JSON 檔案，如下所示；您可以刪除並取代內容。

```
{
"Description":                          "Create WordPress CodeDeploy App",
"VpcId":                                "VPC_ID",
"StackTemplateId":                      "stm-sft6rv00000000000",
"Name":                                 "WordPressCDApp",
"TimeoutInMinutes":                     60,
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. 將 CreateRfc 的 JSON 範本輸出到目前資料夾中的檔案；範例將其命名為 CreateCDAppRfc.json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
```

4. 修改並儲存 JSON 檔案，如下所示；您可以刪除並取代內容。請注意，RequestedStartTime和 現在RequestedEndTime是選用的；排除它們會導致 RFC 在核准後立即執行 （通常會自動發生）。若要提交「排程」RFC，請新增這些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0ah3gwb9seqk2",
"Title":                "CD-App-For-WP-Stack-RFC"
}
```

5. 建立 RFC，指定 CreateCDAppRfc 檔案和執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

   您會在回應中收到新 RFC 的 RFC ID。儲存後續步驟的 ID。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

7. 提交 RFC：

```
aws amscm get-rfc --rfc-id RFC_ID
```

建立 CodeDeploy 部署群組

建立 CodeDeploy 部署群組。

CodeDeploy 部署群組會定義一組以部署為目標的個別執行個體。

必要資料：

- `VpcId`：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- `CodeDeployApplicationName`：使用您先前建立的值。
- `CodeDeployAutoScalingGroups`：使用您先前建立的 Auto Scaling 群組名稱。
- `CodeDeployDeploymentGroupName`：部署群組的名稱。此名稱在與部署群組建立關聯的每個應用程式中都必須是獨一無二的。
- `CodeDeployServiceRoleArn`：使用範例中提供的公式。
- `ChangeTypeId` 和 `ChangeTypeVersion`：此演練的變更類型 ID 為 `ct-2gd0u847qd9d2`，若要了解最新版本，請執行此命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2gd0u847qd9d2
```

1. 將執行參數 JSON 結構描述輸出到目前資料夾中的檔案；範例將其命名為 CreateCDDepGroupParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateCDDepGroupParams.json
```

2. 修改並儲存 JSON 檔案，如下所示；您可以刪除並取代內容。

```
{
"Description":                      "CreateWPCDDeploymentGroup",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sp9lrk00000000000",
"Name":                             "WordPressCDAppGroup",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
    "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
    "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
    "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
codedeploy-role"
    }
}
```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；範例名稱為 CreateCDDepGroupRfc.json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
```

4. 修改並儲存 JSON 檔案，如下所示；您可以刪除並取代內容。請注意，RequestedStartTime和 現在RequestedEndTime是選用的；排除它們會導致 RFC 在核准後立即執行（通常會自動發生）。若要提交「排程」RFC，請新增這些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-For-WP-Stack-RFC"
}
```

5. 建立 RFC，指定 CreateCDDepGroupRfc 檔案和執行參數檔案：

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

您會在回應中收到新 RFC 的 RFC ID。儲存後續步驟的 ID。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

7. 檢查 RFC 狀態：

```
aws amscm get-rfc --rfc-id RFC_ID
```

上傳 WordPress 應用程式

您會自動存取您建立的任何 S3 儲存貯體執行個體。您可以透過堡壘（請參閱存取執行個體) 或透過 S3 主控台存取，然後上傳 CodeDeploy 套件。套件必須就位，才能繼續部署堆疊。此範例使用先前建立的儲存貯體名稱。

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

使用 CodeDeploy 部署 WordPress 應用程式

部署 CodeDeploy 應用程式。

一旦您擁有 CodeDeploy 應用程式套件和部署群組，請使用此 RFC 來部署應用程式。

必要資料：

- VPC-ID：您正在使用的 VPC，這應該與先前使用的 VPC 相同。
- CodeDeployApplicationName：使用您先前建立之 CodeDeploy 應用程式的名稱。
- CodeDeployDeploymentGroupName：使用您先前建立的 CodeDeploy 部署群組名稱。
- S3Location（您上傳應用程式套件的位置）S3Bucket：：您先前建立的 BucketName，S3BundleType以及S3Key：您放在 S3 存放區上套件的類型和名稱。
- ChangeTypeId 和 ChangeTypeVersion：此演練的變更類型 ID 為 ct-2edc3sd1sqmrb，若要了解最新版本，請執行此命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2edc3sd1sqmrb
```

1. 將 CodeDeploy 應用程式部署 CT 的執行參數 JSON 結構描述輸出至目前資料夾中的檔案；範例會將其命名為 DeployCDAppParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
```

2. 修改 JSON 檔案，如下所示；您可以刪除和取代內容。對於 S3Bucket，請使用您先前建立BucketName的 。

```
{
"Description":                        "Deploy WordPress CodeDeploy Application",
"VpcId":                              "VPC_ID",
"Name":                               "WP CodeDeploy Deployment Group",
"TimeoutInMinutes":                   60,
"Parameters":   {
    "CodeDeployApplicationName":         "WordPressCDApp",
    "CodeDeployDeploymentGroupName":     "WordPressCDDepGroup",
    "CodeDeployIgnoreApplicationStopFailures": false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket": "ACCOUNT_ID.BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key": "wordpress.zip" }
        }
    }
}
```

3. 將 CreateRfc 的 JSON 範本輸出至目前資料夾中的檔案；範例名稱為 DeployCDAppRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. 修改並儲存 DeployCDAppRfc.json 檔案；您可以刪除和取代內容。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-Stack-RFC",
"RequestedStartTime":   "2017-04-28T22:45:00Z",
"RequestedEndTime":     "2017-04-28T22:45:00Z"
}
```

5. 建立 RFC，指定執行參數檔案和 DeployCDAppRfc 檔案：

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

您會在回應中收到新 RFC 的 RfcId。儲存後續步驟的 ID。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您不會收到任何輸出。

驗證應用程式部署

導覽至先前建立負載平衡器的端點 (ELB CName)，並使用 WordPress 部署的路徑：/WordPress。例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

向下傾斜應用程式部署

若要縮減部署，請針對 RDS 資料庫堆疊、應用程式負載平衡器、Auto Scaling 群組、S3 儲存貯體，以及 Code Deploy 應用程式和 group--ix RFCs提交 Delete Stack CT。此外，您可以為要刪除的 RDS 快照提交服務請求（它們會在 10 天後自動刪除，但在那裡需要支付少量費用）。收集所有 的堆疊 IDs，然後遵循下列步驟。

此逐步解說提供使用 AMS 主控台刪除 S3 堆疊的範例；此程序適用於使用 AMS 主控台刪除任何堆疊。

> **Note**
>
> 如果刪除 S3 儲存貯體，必須先清空物件。

必要資料：

- `StackId`：要使用的堆疊。您可以查看 AMS 主控台堆疊頁面，透過左側導覽中的連結取得。使用 AMS SKMS API/CLI，執行 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。操作 (`list-stack-summaries` CLI 中的)。

- 此演練的變更類型 ID 為 ct-0q0bic0ywqk6c，版本為 "1.0"，若要了解最新版本，請執行此命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

內嵌建立：

- 使用內嵌提供的執行參數發出建立 RFC 命令（在內嵌提供執行參數時逸出引號）。E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- 使用建立 RFC 操作中傳回的 RFC ID 提交 RFC。在提交之前，RFC 會保持 Editing 狀態，且不會採取任何動作。

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- 監控 RFC 狀態並檢視執行輸出：

```
aws amscm get-rfc --rfc-id RFC_ID
```

範本建立：

1. 將 RFC 範本輸出到目前資料夾中的檔案；範例將其命名為 DeleteStackRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
```

2. 修改並儲存 DeleteStackRfc.json 檔案。由於刪除堆疊只有一個執行參數，因此執行參數可以位於 DeleteStackRfc.json 檔案本身（不需要使用執行參數建立單獨的 JSON 檔案）。

   ExecutionParameters JSON 延伸中的內部引號必須以反斜線 (\) 逸出。沒有開始和結束時間的範例：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
```

```
}
```

3. 建立 RFC：

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

您會在回應中收到新 RFC 的 RfcId。例如：

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

儲存後續步驟的 ID。

4. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，您在命令列不會收到確認。

5. 若要監控請求的狀態和檢視執行輸出：

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

應用程式維護

部署基礎設施後，從 QA 到預備階段到生產，在所有 AMS 環境中以一致的方式進行更新是一大挑戰。

本節提供 AMS 工作負載擷取程序的概觀，以及您可以用來讓雲端基礎設施層保持最新狀態的一些不同方法範例。

應用程式維護策略

部署應用程式的方式會影響您維護應用程式的方式。本節提供應用程式維護的一些策略。

環境更新可以涉及下列任何變更：

- 安全性更新
- 您應用程式的新版本
- 應用程式組態變更
- 依存項目的更新

> **Note**
>
> 對於任何應用程式部署，無論使用何種方法，一律事先提出服務請求，讓 AMS 知道您要部署應用程式。

不可變與可變應用程式安裝範例

運算執行個體互斥性	應用程式安裝方法	AMI
Mutable	使用 CodeDeploy	AMS 提供的
	手動	
	使用 Chef 或 Puppet，以提取為基礎	
	使用 Ansible 或 Salt，以推送為基礎	
固定	使用 Golden AMI	自訂（根據 AMS 提供的）

啟用 CodeDeploy 的 AMI 的可互斥部署

AWS CodeDeploy 是一項服務，可將程式碼部署自動化至任何執行個體，包括 Amazon EC2 執行個體和執行內部部署的執行個體。您可以使用 CodeDeploy 搭配 AMS 來建立和部署 CodeDeploy 應用程式。請注意，AMS 提供 CodeDeploy 應用程式的預設執行個體描述檔。

- Amazon Linux （第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

第一次使用 CodeDeploy 之前，您必須先完成數個設定步驟：

1. 安裝或升級 AWS CLI
2. 為 AWS CodeDeploy 建立服務角色，您可以在部署中使用服務角色 ARN

您可以在變更類型參考中找到所有 CT 選項IDs。

> **Note**
>
> 目前，您必須搭配此解決方案使用 Amazon S3 儲存體。

此處概述了基本步驟，而程序在 AMS 使用者指南中詳細說明。

1. 建立 Amazon S3 儲存貯體。CT：ct-1a68ck03fn98r。S3 儲存貯體必須已啟用版本控制 （如需執行此作業的相關資訊，請參閱啟用儲存貯體版本控制)。
2. 將您的綁定 CodeDeploy 成品放在其上。您可以使用 Amazon S3 主控台執行此操作，而無需透過 AMS 請求存取權。或者使用此命令的變體：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 尋找 AMS `customer-` AMI；使用下列其中一項：

   - AMS 主控台：相關 VPC 的 VPC 詳細資訊頁面
   - AMS API 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 或 CLI：`aws amsskms list-amis`

4. 建立 Autoscaling 群組 (ASG)。CT：ct-2tylseo8rxfsc。指定 AMS AMI、將負載平衡器設定為具有開放連接埠、`customer-mc-ec2-instance-profile`為 指定 `ASGIAMInstanceProfile`。
5. 建立 CodeDeploy 應用程式。CT：ct-0ah3gwb9seqk2。參數包含應用程式名稱，例如 `WordpressProd`。
6. 建立 CodeDeploy 部署群組。CT：ct-2gd0u847qd9d2。參數包括 CodeDeploy 應用程式名稱、ASG 名稱、組態類型名稱和服務角色 ARN。
7. 部署 CodeDeploy 應用程式。CT：ct-2edc3sd1sqmrb。參數包括 CodeDeploy 應用程式名稱、組態類型名稱、部署群組名稱、修訂類型，以及 CodeDeploy 成品所在的 S3 儲存貯體位置。

可互斥部署、手動設定和更新的應用程式執行個體

此應用程式部署策略是簡單且手動的應用程式執行個體更新。這些是基本步驟。

您可以在變更類型參考中找到所有 CT 選項IDs。

> **Note**
>
> 目前，您必須搭配此解決方案使用 Amazon S3 儲存體。

這裡概述了基本步驟；AMS 使用者指南中詳細說明了各種程序。

1. 建立 Amazon S3 儲存貯體。CT：ct-1a68ck03fn98r。S3 儲存貯體必須已啟用版本控制 （ 如需執行此作業的相關資訊，請參閱啟用儲存貯體版本控制)。
2. 將綁定的應用程式成品放在其上 （ 您的應用程式啟動和工作所需的一切 ）。您可以使用 Amazon S3 主控台執行此操作，而無需透過 AMS 請求存取權。或者使用此命令的變體：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 尋找 AMS AMI，所有 都會有 CodeDeploy。若要尋找「客戶」AMI，請使用下列其中一項：

   - AMS 主控台：相關 VPC 的 VPC 詳細資訊頁面
   - AMS API 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 或 CLI：`aws amsskms list-amis`

4. 使用該 AMI 建立 EC2 執行個體。CT：ct-14027q0sjyt1h。指定 AMS AMI、設定標籤，`Key=backup, Value=true`並指定 `InstanceProfile` 參數`customer-mc-ec2-instance-profile`的 。請注意傳回的執行個體 ID。

5. 請求管理員存取執行個體。CT：ct-1dmlg9g1l91h6。您將需要帳戶的 FQDN。如果您不確定 FQDN 是什麼，可以透過下列方式找到它：
   - 使用適用於 Directory Services 的 AWS 管理主控台（在安全和身分下）目錄名稱索引標籤。
   - 執行其中一個命令（傳回目錄類別；DC+DC+DC=FQDN)：Windows： `whoami /fqdn`或 Linux：`hostname --fqdn`。
6. 登入執行個體，請參閱《AMS 使用者指南》中的透過堡壘存取執行個體。
7. 將綁定的應用程式檔案從 S3 儲存貯體下載至執行個體。
8. 使用 AMS 的服務請求請求立即備份，您將需要知道執行個體 ID。
9. 當您需要更新應用程式時，請將新檔案載入 S3 儲存貯體，然後遵循步驟 3 到 8。

使用提取式部署工具設定的 AMI 進行互斥部署

此策略依賴 Managed Services Create EC2 CT 中的 `InstanceUserData` 參數。如需使用此參數的詳細資訊，請參閱使用使用者資料設定執行個體。此範例假設提取型應用程式部署工具，例如 Chef 或 Puppet。

所有 AMS AMIs 都支援 CodeDeploy 代理程式。以下是支援的 AMIs清單：

- Amazon Linux（第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

您可以在變更類型參考中找到所有 CT 選項IDs。

> **Note**
>
> 目前，您必須搭配此解決方案使用 Amazon S3 儲存體。

此處概述了基本步驟，而程序在 AMS 使用者指南中詳細說明。

1. 建立 Amazon S3 儲存貯體。CT：ct-1a68ck03fn98r。S3 儲存貯體必須已啟用版本控制（如需執行此作業的相關資訊，請參閱啟用儲存貯體版本控制)。

2. 將您的綁定 CodeDeploy 成品放在其上。您可以使用 Amazon S3 主控台執行此操作，而無需透過 AMS 請求存取權。或者使用此命令的變體：

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. 尋找 AMS `customer-` AMI；使用下列其中一項：
   - AMS 主控台：相關 VPC 的 VPC 詳細資訊頁面
   - AMS API 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 或 CLI：`aws amsskms list-amis`
4. 建立 EC2 執行個體。CT：ct-14027q0sjyt1h；設定標籤 `Key=backup, Value=true`，並使用 `InstanceUserData` 參數指定引導和其他指令碼（下載 Chef/Puppet 代理程式等），並包含必要的授權金鑰。您可以在 AMS 使用者指南的變更管理區段中，找到建立 HA 兩層部署的範例。或者，請求存取和登入執行個體，並使用必要的部署成品進行設定。請記住，提取型部署命令會從執行個體上的代理程式傳送至您的公司主伺服器，而且可能需要授權才能通過堡壘。您可能需要向 AMS 提出服務請求，才能在沒有堡壘的情況下請求安全群組/AD 群組存取。
5. 重複步驟 4 建立另一個 EC2 執行個體，並使用部署工具主伺服器進行設定。
6. 當您需要更新應用程式時，請使用 部署工具將更新推展到您的執行個體。

使用推送式部署工具設定的 AMI 進行互斥部署

此策略依賴 Managed Services Create EC2 CT 中的 `InstanceUserData` 參數。如需使用此參數的詳細資訊，請參閱使用使用者資料設定執行個體。此範例假設提取型應用程式部署工具，例如 Chef 或 Puppet。

您可以在變更類型參考中找到所有 CT 選項IDs。

> **Note**
>
> 目前，您必須搭配此解決方案使用 Amazon S3 儲存。

此處概述了基本步驟，而程序詳述於 AMS 使用者指南中。

1. 建立 Amazon S3 儲存貯體。CT：ct-1a68ck03fn98r。S3 儲存貯體必須已啟用版本控制（如需執行此作業的相關資訊，請參閱啟用儲存貯體版本控制）。

2. 將您的綁定 CodeDeploy 成品放在其上。您可以使用 Amazon S3 主控台執行此操作，而無需透過 AMS 請求存取權。或者使用此命令的變體：

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. 尋找 AMS AMI，所有 都會有 CodeDeploy。若要尋找「客戶」AMI，請使用：
   - AMS 主控台：相關 VPC 的 VPC 詳細資訊頁面
   - AMS API 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 或 CLI：`aws amsskms list-amis`
4. 建立 EC2 執行個體。CT：ct-14027q0sjyt1h；設定標籤 `Key=backup, Value=true`，並使用 `InstanceUserData` 參數來執行引導和其他指令碼，包括授權金鑰、SALT 堆疊 （引導小兵—如需詳細資訊，請參閱使用 Cloud-Init 在 Linux EC2 上引導 Salt) 或 Ansible （安裝金鑰對—如需詳細資訊，請參閱 Ansible 和動態 Amazon EC2 庫存管理入門)。或者，請求存取並登入執行個體，並使用必要的部署成品進行設定。請記住，推送型命令來自您的企業子網路到您的執行個體，您可能需要設定授權，才能通過堡壘。您可能需要向 AMS 提出服務請求，才能在沒有堡壘的情況下請求安全群組/AD 群組存取。
5. 重複步驟 4 建立另一個 EC2 執行個體，並使用部署工具主伺服器進行設定。
6. 當您需要更新應用程式時，請使用 部署工具將更新推展到您的執行個體。

使用黃金 AMI 進行不可避免的部署

此策略採用「黃金」AMI，您已設定為如同您希望所有應用程式執行個體的行為。例如，使用此黃金 AMI 建立的執行個體會自行加入正確的網域和 DNS、自行設定、重新啟動和啟動所有必要的系統。當您想要更新應用程式執行個體時，您可以重新建立黃金 AMI，並使用它推出全新的應用程式執行個體。

所有 AMS AMIs 都支援 CodeDeploy 代理程式。以下是支援的 AMIs清單：

- Amazon Linux （第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

您可以在變更類型參考中找到所有 CT 選項IDs。

> **Note**
>
> 目前，您必須搭配此解決方案使用 Amazon S3 儲存。

1. 建立 Amazon S3 儲存貯體。CT：ct-1a68ck03fn98r。S3 儲存貯體必須已啟用版本控制（如需執行此作業的相關資訊，請參閱啟用儲存貯體版本控制)。
2. 將綁定的應用程式成品放在其上（您的應用程式啟動和工作所需的一切）。您可以使用 Amazon S3 主控台執行此操作，而無需透過 AMS 請求存取權。或者使用此命令的變體：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 尋找 AMS `customer-` AMI；使用下列其中一項：
   - AMS 主控台：相關 VPC 的 VPC 詳細資訊頁面
   - AMS API 如需 AMS SKMS API 參考，請參閱 AWS 成品主控台中的報告索引標籤。 或 CLI：`aws amsskms list-amis`
4. 使用該 AMI 建立 EC2 執行個體。CT：ct-14027q0sjyt1h。指定 AMS AMI、設定標籤`Key=backup, Value=true`，並為 指定 `customer-mc-ec2-instance-profile InstanceProfile`。請注意傳回的執行個體 ID。
5. 請求管理員存取執行個體。CT：ct-1dmlg9g1l91h6。您將需要帳戶的 FQDN。如果您不確定 FQDN 是什麼，可以透過下列方式找到它：
   - 使用 AWS Management Console for Directory Services（在安全和身分下）目錄名稱索引標籤。
   - 執行其中一個命令（傳回目錄類別；DC+DC+DC=FQDN)：Windows： `whoami /fqdn`或 Linux：`hostname --fqdn`。
6. 登入執行個體，請參閱《AMS 使用者指南》中的存取執行個體。
7. 從 S3 儲存貯體下載至綁定應用程式檔案的執行個體。設定執行個體，以便在開機時自行部署功能完整的應用程式。
8. 在執行個體上建立黃金 AMI。CT：ct-3rqqu43krekby。如需詳細資訊，請參閱 AMI | Create。
9. 設定 Auto Scaling 群組以使用該 AMI 建立新的執行個體。CT：ct-2tylseo8rxfsc。當您需要更新應用程式時，請遵循此程序並請求 AMS 更新 ASG 以使用新的黃金 AMI；使用管理 | 其他 | 為此更新 CT。

更新策略

您可以使用幾個不同的策略來更新 AMS 受管環境中的應用程式或執行個體。

- 排程停機時間：此簡單策略涉及排程應用程式離線的時間並手動更新。若要這樣做，請提交管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 請求以停止所需的執行個體。進行必要的更新，然後提交另一個管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 請求來啟動執行個體。

- 藍/綠：此策略需要您擁有備援環境 （兩個功能完整的環境），並使用網域名稱系統 (DNS) 或 Web 防火牆 (WAF) 更新將流量重新導向，讓一個環境離線。更新一個環境，然後再次重新導向以更新另一個環境。

  若要進一步了解，請參閱 AWS CodeDeploy 推出藍/綠部署。

- 使用新的 AMI 滾動更新：您可以在其中自訂新的 AMI （請參閱建立 AMI)，然後請求 AMS 將其部署到您的 Auto Scaling 群組。使用管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 來執行此操作。

AWS Managed Services 資源排程器

使用 AWS Managed Services (AMS) Resource Scheduler 來排程自動啟動和停止您帳戶中的 AutoScaling 群組、Amazon EC2 執行個體和 RDS 執行個體。這有助於降低資源不應全年無休執行的基礎設施成本。解決方案建置在 上的執行個體排程器 AWS之上，但包含 AMS 需求特有的其他功能和自訂。

> **Note**
>
> 根據預設，AMS Resource Scheduler 不會與不屬於 AWS CloudFormation 堆疊的資源互動。資源必須是以「stack-」、「sc-」或「SC-」開頭的堆疊的一部分。若要排程不屬於 CloudFormation 堆疊的資源，您可以將 Resource Scheduler 堆疊參數更新`ScheduleNonStackResources`為 `Yes`。

AMS Resource Scheduler 使用期間和排程：

- 期間定義資源排程器執行的時間，例如開始時間、結束時間和當月天數。
- 排程包含您定義的期間，以及其他組態，例如 SSM 維護時段、時區、休眠設定等；並指定資源應在何時執行，並指定設定的期間規則。

您可以使用 AMS Resource Scheduler 的自動化變更類型 (CTs) 來設定這些期間和排程。

如需 AMS Resource Scheduler 可用設定的完整詳細資訊，請參閱 解決方案元件中對應的 AWS 執行個體排程器文件。如需解決方案的架構檢視，請參閱 Architecture overview.html 中對應的 AWS 執行個體排程器文件。

部署 AMS 資源排程器

若要部署 AMS 資源排程器，請使用自動變更類型 (CT)：部署 | AMS 資源排程器 | 解決方案 | 部署 (ct-0ywnhc8e5k9z5) 來引發 RFC，然後部署帳戶中的解決方案。執行 RFC 後，包含具有預設組態之 AMS Resource Scheduler 資源的 CloudFormation 堆疊會自動佈建至您的帳戶。如需 Resource Scheduler 變更類型的詳細資訊，請參閱 AMS Resource Scheduler。

> **Note**
>
> 若要了解您的帳戶中是否已部署 AMS Resource Scheduler，請檢查該帳戶的 AWS Lambda 主控台，並尋找 AMSResourceScheduler 函數。

在您的帳戶中佈建 AMS Resource Scheduler 之後，我們建議您檢閱預設組態，並視需要根據您的偏好設定自訂組態，例如標籤索引鍵、時區、排程服務等。如需建議自訂的詳細資訊，請參閱 自訂 AMS 資源排程器，下一步。

若要進行自訂組態，或僅確認資源排程器組態，

自訂 AMS 資源排程器

我們建議您使用更新 AMS Resource Scheduler 變更類型來自訂 AMS Resource Scheduler 的下列屬性，請參閱 AMS Resource Scheduler。

- 標籤名稱：Resource Scheduler 用來將執行個體排程與 資源建立關聯的標籤名稱。預設值為排程。
- 排程服務：以逗號分隔的清單，列出 Resource Scheduler 可以管理的服務。預設值為 "ec2，rds，autoscaling"。有效值為 "ec2"、"rds" 和 "autoscaling"
- 預設時區：指定資源排程器要使用的預設時區。預設值為 UTC。
- 使用 CMK：以逗號分隔的 Amazon KMS 客戶受管金鑰 (CMK) ARNs 清單，可將許可授予資源排程器。
- 使用 LicenseManager：可以授予該資源排程器以逗號分隔的 AWS Licence Manager ARNs 清單。

> **Note**
>
> AMS 可能會隨時釋出功能和修正，讓 AMS Resource Scheduler 在您的帳戶中保持最新狀態。發生這種情況時，您對 AMS 資源排程器進行的任何自訂都會保留。

使用 AMS 資源排程器

若要在部署解決方案後設定 AMS Resource Scheduler，請使用自動化 Resource Scheduler CTs 來建立、刪除、更新和描述（取得詳細資訊）AMS Resource Scheduler 期間（資源排程器執行的時間）和排程（設定的期間和其他選項）。如需使用 AMS Resource Scheduler 變更類型的範例，請參閱 AMS Resource Scheduler。

若要選取要由 AMS Resource Scheduler 管理的資源，請在部署和排程建立之後，使用 AMS 標籤建立 CTs，以您在部署期間提供的標籤索引鍵標記 Auto Scaling 群組、Amazon RDS 堆疊和 Amazon EC2 資源，並將定義的排程作為標籤值。標記資源後，資源會根據您定義的資源排程器排程排程為啟動或停止。

使用 AMS Resource Scheduler 無需額外費用。不過，解決方案會使用數個 AWS 服務 ，而且系統會在使用這些資源時向您收取費用。如需詳細資訊，請參閱架構概觀。

若要選擇退出 AMS Resource Scheduler：

- 對於暫時選擇退出或停用：使用自動化管理 | AMS 資源排程器 | 狀態 | 停用變更類型 (ct-14v49adibs4db) 提交 RFC
- 對於永久移除：提交管理 | 其他 | 其他 | 更新（需要檢閱）(ct-0xdawir96cy7k) RFC 請求從資源排程器版本自動化系統移除

AMS 資源排程器成本估算器

為了追蹤成本節省，AMS Resource Scheduler 具有每小時計算排程器管理之 Amazon EC2 和 RDS 資源預估成本節省的元件。然後，此節省成本的資料會發佈為 CloudWatch 指標 (`AMS/ResourceScheduler`)，以協助您追蹤資料。成本節省估算器只會預估執行個體執行時數的節省。它不會考慮任何其他成本，例如與資源相關聯的資料傳輸成本。

使用 Resource Scheduler 啟用成本節省估算器。它會每小時執行一次，並從中擷取成本和用量資料 AWS Cost Explorer。從該資料中，它會計算每個執行個體類型的每小時平均成本，然後在未排定的情況下執行整天的預估成本。節省成本是預測成本與 Cost Explorer 在特定日期實際報告成本之間的差異。

例如，如果執行個體 A 使用 Resource Scheduler 設定為從上午 9 點到下午 5 點執行，即指定日期的 8 小時。Cost Explorer 會將成本報告為 $1，用量報告為 8。因此，每小時的平均成本為 0.125 美元。如果未使用 Resource Scheduler 排程執行個體，則執行個體會在當天執行 24 小時。在這種情況下，成本會是 24x0.125 = $3。資源排程器可協助您節省 2 美元的成本。

為了節省成本估算器僅從 Cost Explorer 擷取由 Resource Scheduler 管理的資源的成本和用量，Resource Scheduler 用於目標資源的標籤索引鍵需要啟用為帳單儀表板中的成本分配標籤。如果帳戶屬於組織，則需要在組織的管理帳戶中啟用標籤金鑰。如需執行此操作的資訊，請參閱啟用使用者定義的成本分配標籤和使用者定義的成本分配標籤

在標籤金鑰啟用為成本分配標籤之後， AWS 帳單會開始追蹤資源排程器所管理資源的成本和用量，並在該資料可用之後，成本節省估算器會開始計算成本節省，並在 CloudWatch 中的`AMS/ResourceScheduler`指標命名空間下發佈資料。

成本估算器秘訣

成本節省估算器不接受折扣，例如預留執行個體、節省計劃等，並將其計算納入考量。估算器會從 Cost Explorer 取得用量成本，並計算資源每小時的平均成本。如需詳細資訊，請參閱了解您的 AWS 成本資料集：備忘單

為了節省成本估算器僅從 Cost Explorer 擷取由 Resource Scheduler 管理的資源的成本和用量，Resource Scheduler 用於目標資源的標籤索引鍵需要啟用為帳單儀表板中的成本分配標籤。如果帳戶屬於組織，則需要在組織的管理帳戶中啟用標籤金鑰。如需執行此作業的資訊，請參閱使用者定義的成本分配標籤。如果未啟用成本分配標籤，估算器就無法計算節省成本並發佈指標，即使已啟用也一樣。

AMS Resource Scheduler 最佳實務

排程 Amazon EC2 執行個體

- 執行個體關閉行為必須設定為 `stop` ，而不是設定為 `terminate`。對於使用 AMS Amazon EC2 建立自動變更類型 (ct-14027q0sjyt1h) 建立的`stop`執行個體，這是預先設定為 ，並且可以透過將 `InstanceInitiatedShutdownBehavior` 屬性設定為 ，為使用 AWS CloudFormation 擷取建立的 Amazon EC2 執行個體設定 `stop`。如果執行個體已將關閉行為設定為 `terminate`，則當資源排程器停止執行個體時，執行個體將會結束，且排程器將無法啟動它們。
- AMS Resource Scheduler 不會個別處理屬於 Auto Scaling 群組的 Amazon EC2 執行個體，即使它們已加上標籤。
- 如果目標執行個體根磁碟區使用 KMS 客戶主金鑰 (CMK) 加密，則需要將額外的`kms:CreateGrant`許可新增至您的 Resource Scheduler IAM 角色，排程器才能啟動此類執行個

體。根據預設，此許可不會新增至角色，以改善安全性。如果您需要此許可，請使用 Management | AMS Resource Scheduler | Solution | Update change type 提交 RFC，並指定 KMS CMKs 的 ARNs 逗號分隔清單。

排程 Auto Scaling 群組

- AMS Resource Scheduler 會啟動或停止 Auto Scaling 群組的自動擴展，而不是群組中的個別執行個體。也就是說，排程器會還原 Auto Scaling 群組的大小（開始）或將大小設定為 0（停止）。
- 使用指定的標籤標記 AutoScaling 群組，而不是群組中的執行個體。
- 在停止期間，AMS Resource Scheduler 會存放 Auto Scaling 群組的最小、預期和最大容量值，並將最小和預期容量設定為 0。在啟動期間，排程器會還原停止期間的 Auto Scaling 群組大小。因此，Auto Scaling 群組執行個體必須使用適當的容量組態，以便執行個體的終止和重新啟動不會影響 Auto Scaling 群組中執行的任何應用程式。
- 如果在執行期間修改 Auto Scaling 群組（最小或最大容量），排程器會存放新的 Auto Scaling 群組大小，並在停止排程結束時還原群組時使用。

排程 Amazon RDS 執行個體

- 排程器可以在停止 RDS 執行個體之前擷取快照（不適用於 Aurora 資料庫叢集）。此功能預設為開啟，且建立 RDS 執行個體快照 AWS CloudFormation 範本參數設為 true。快照會保留到下次停止 Amazon RDS 執行個體並建立新的快照為止。

  排程器可以啟動/停止屬於叢集或 Amazon RDS Aurora 資料庫或多可用區域 (Multi-AZ) 組態中的 Amazon RDS 執行個體。不過，當排程器無法停止 Amazon RDS 執行個體時，請檢查 Amazon RDS 限制，尤其是多可用區域執行個體。若要排程 Aurora 叢集啟動或停止，請使用排程 Aurora 叢集範本參數（預設為 true)。Aurora 叢集（而非叢集內的個別執行個體）必須使用初始組態期間定義的標籤索引鍵和排程名稱做為標籤值來標記，以排程該叢集。

  每個 Amazon RDS 執行個體都有每週維護時段，在此期間會套用任何系統變更。在維護時段期間，Amazon RDS 會自動啟動已停止超過七天的執行個體，以套用維護。請注意，維護事件完成後，Amazon RDS 不會停止執行個體。

  排程器允許指定是否要將 Amazon RDS 執行個體的偏好維護時段新增為其排程的執行期間。如果沒有其他執行期間指定執行個體應執行，且維護事件已完成，解決方案將在維護時段開始時啟動執行個體，並在維護時段結束時停止執行個體。

  如果維護事件未在維護時段結束時完成，執行個體將在維護事件完成後的排程間隔之前執行。

> **Note**
>
> 排程器不會驗證資源是否已啟動或停止。它會發出 API 呼叫並繼續。如果 API 呼叫失敗，它會記錄錯誤以進行調查。

應用程式安全考量

應用程式安全性包括考量應用程式需要執行哪些許可、防火牆規則、應啟用哪些 IAM 角色才能存取應用程式。

若要進一步了解一般 AWS 安全性，請參閱安全性、身分和合規的最佳實務。

組態管理的存取權

AWS Managed Services (AMS) 致力於為您提供輕鬆的基礎設施，讓您不必擔心安全問題、修補問題、備份問題等。若要這樣做，AMS 建議將 IAM 角色降至最低，如果使用應用程式部署工具，則僅允許特定群組或主伺服器存取執行您應用程式的執行個體。

應用程式存取防火牆規則

如同作業系統 (OS)，所有應用程式存取都應該使用 Active Directory (AD) 群組進行管理。使用 Amazon Relational Database Service (Amazon RDS) 作為範例，您必須破壞鏡像 （ 複寫 ） 以新增使用者。最佳方法是在 AD 中建立群組，並在資料庫建立時新增群組。在 AMS AD 中擁有群組意味著您可以建立 CTs以進行應用程式存取。如需 AD 官方分組策略的資訊，請參閱使用群組巢狀策略 – 群組策略的 AD 最佳實務。

若要進一步了解網域樹狀目錄和父/子網域，請參閱網域和樹系的運作方式。

下列規則說明適合與位於子網域中的使用者進行多網域樹系信任的解決方案。

Windows 執行個體

這些是為您的 Windows 父系和子系網域控制站設定的規則。

父系網域控制站，Windows

FROM：父系網域控制站 TO：Windows 堆疊和共用服務子網路

來源連接埠	目標連接埠	通訊協定
88	49152 - 65535	TCP

來源連接埠	目標連接埠	通訊協定
389	49152 - 65535	UDP

FROM： 堆疊子網路，包括共用服務 TO： Windows 樹系根網域控制站

來源連接埠	目標連接埠	通訊協定
49152 - 65535	88	TCP
49152 - 65535	389	UDP

子網域控制站，Windows

FROM：子網域控制站 TO：Windows AWS 網域控制站

來源連接埠	目標連接埠	通訊協定
49152 - 65535	53	TCP
49152 - 65535	88	TCP
49152 - 65535	389	UDP

FROM：子網域控制站 TO：Windows 堆疊和共用服務子網路

來源連接埠	目標連接埠	通訊協定
88	49152 - 65535	TCP
135	49152 - 65535	TCP
389	49152 - 65535	TCP
389	49152 - 65535	UDP
445	49152 - 65535	TCP

來源連接埠	目標連接埠	通訊協定
49152 - 65535	49152 - 65535	TCP

FROM： 堆疊子網路，包括共用服務 TO： Windows 子網域控制站

來源連接埠	目標連接埠	通訊協定
49152 - 65535	88	TCP
49152 - 65535	135	TCP
49152 - 65535	389	TCP
49152 - 65535	389	UDP
49152 - 65535	445	TCP
49152 - 65535	49152 - 65535	TCP

Linux 執行個體

這些是為您的 Linux 父系和子系網域控制站設定的規則。

所有測試都是使用 Amazon Linux 執行。雖然 Windows 的動態連接埠範圍是 49152 到 65535，但許多 Linux 核心會使用連接埠範圍 32768 到 61000。執行以下命令以檢視 IP 連接埠範圍。

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

Linux 父系網域控制站

FROM：父系網域控制站 TO：Linux 堆疊和共用服務子網路

來源連接埠	目標連接埠	通訊協定
389	32768 - 61000	UDP
88	32768 - 61000	TCP

FROM： 堆疊子網路，包括共用服務 TO： Linux 樹系根網域控制站

來源連接埠	目標連接埠	通訊協定
32768 - 61000	88	TCP
32768 - 61000	389	UDP

子網域控制器，Linux

FROM：子網域控制站 TO：Linux AWS 網域控制站

來源連接埠	目標連接埠	通訊協定
49152 - 65535	53	TCP
49152 - 65535	88	TCP
389	49152 - 65535	UDP
49152 - 65535	389	UDP

FROM：子網域控制器 TO：Linux 堆疊和共用服務子網路

來源連接埠	目標連接埠	通訊協定
88	32768 - 61000	TCP
389	32768 - 61000	UDP

FROM： Stack 子網路，包括共用服務 TO： Linux 子網域控制器

來源連接埠	目標連接埠	通訊協定
32768 - 61000	88	TCP
32768 - 61000	389	UDP

AMS 輸出流量管理

根據預設，AMS 私有和客戶應用程式子網路目的地 CIDR 為 0.0.0.0/0 的路由具有網路位址轉譯 (NAT) 閘道做為目標。AMS 服務、TrendMicro 和修補是必須具有網際網路輸出存取權的元件，以便 AMS 能夠提供其服務，而 TrendMicro 和作業系統可以取得更新。

AMS 支援透過客戶管理的輸出裝置將輸出流量轉移到網際網路，只要：

- 它充當隱含（例如透明）代理。

  以及

- 它允許 AMS HTTP 和 HTTPS 相依性（列於本節），以允許 AMS 受管基礎設施的持續修補和維護。

部分範例如下：

- 傳輸閘道 (TGW) 具有預設路由，指向多帳戶登陸區域網路帳戶中透過 AWS Direct Connect 連線的客戶受管內部部署防火牆。
- TGW 有一個預設路由，指向多帳戶登陸區域輸出 VPC 中的 AWS 端點，利用 AWS PrivateLink，指向另一個 AWS 帳戶中的客戶受管代理。
- TGW 預設路由指向另一個 AWS 帳戶中的客戶受管防火牆，並以site-to-site連接做為多帳戶登陸區域 TGW 的連接。

AMS 已識別對應的 AMS HTTP 和 HTTPS 相依性，並持續開發和精簡這些相依性。請參閱 egressMgmt.zip。除了 JSON 檔案之外，ZIP 還包含 README。

> **Note**
>
> - 此資訊並不全面 - 此處未列出某些必要的外部網站。
> - 請勿在拒絕清單或封鎖策略下使用此清單。
> - 此清單旨在做為輸出篩選規則集的起點，預期報告工具將用於精確判斷實際流量與清單的差異。

若要詢問有關篩選輸出流量的資訊，請傳送電子郵件給您的 CSDM：ams-csdm@amazon.com。

安全群組

在 AWS VPCs 中，AWS 安全群組充當虛擬防火牆，控制一或多個堆疊（執行個體或一組執行個體）的流量。啟動堆疊時，它會與一或多個安全群組相關聯，以決定允許哪些流量到達該群組：

- 對於公有子網路中的堆疊，預設安全群組接受來自所有位置（網際網路）的 HTTP (80) 和 HTTPS (443) 流量。堆疊也接受來自您公司網路和 AWS 堡壘的內部 SSH 和 RDP 流量。然後，這些堆疊可以透過任何連接埠輸出到網際網路。它們也可以輸出到您的私有子網路和公有子網路中的其他堆疊。
- 私有子網路中的堆疊可以輸出到私有子網路中的任何其他堆疊，而堆疊中的執行個體可以完全透過任何通訊協定互相通訊。

> **Important**
>
> 私有子網路上堆疊的預設安全群組允許私有子網路中的所有堆疊與該私有子網路中的其他堆疊通訊。如果您想要限制私有子網路中堆疊之間的通訊，您必須建立新的安全群組來描述限制。例如，如果您想要限制與資料庫伺服器的通訊，以便該私有子網路中的堆疊只能透過特定連接埠從特定應用程式伺服器通訊，請請求特殊安全群組。本節將說明如何執行此操作。

預設安全群組

MALZ

下表說明堆疊的預設傳入安全群組 (SG) 設定。SG 名為 "SentinelDefaultSecurityGroupPrivateOnly-vpc-ID"，其中 *ID* 是 AMS 多帳戶登陸區域帳戶中的 VPC ID。允許所有流量透過此安全群組傳出至 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly"（允許堆疊子網路中的所有本機流量）。

第二個安全群組 "SentinelDefaultSecurityGroupPrivateOnly" 允許所有流量傳出至 0.0.0.0/0。

> **Tip**
>
> 如果您要為 AMS 變更類型選擇安全群組，例如 EC2 建立或 OpenSearch 建立網域，您可以使用此處所述的其中一個預設安全群組，或您建立的安全群組。您可以在 AWS EC2 主控台或 VPC 主控台中找到每個 VPC 的安全群組清單。

還有其他預設安全群組用於內部 AMS 用途。

AMS 預設安全群組（傳入流量）

類型	通訊協定	連接埠範圍	來源
所有流量	全部	全部	SentinelDefaultSecurityGroupPrivateOnly（限制相同安全群組成員的傳出流量）
所有流量	全部	全部	SentinelDefaultSecurityGroupPrivateOnlyEgressAll（不限制傳出流量）
HTTP、HS、SSH、	TCP	80 / 443（來源 0.0.0.0/0） 允許從堡壘存取 SSH 和 RDP	SentinelDefaultSecurityGroupPublic（不限制傳出流量）
MALZ 堡壘：			
SSH	TCP	22	SharedServices VPC CIDR 和 DMZ VPC CIDR，以及客戶提供的內部部署 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
SALZ 堡壘：			
SSH	TCP	22	mc-initial-garden-LinuxBastionSG
SSH	TCP	22	mc-initial-garden-LinuxBastionDMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionDMZSG

SALZ

下表說明堆疊的預設傳入安全群組 (SG) 設定。SG 名為 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly-*ID*"，其中 *ID* 是唯一的識別符。允許所有流量透過此安

全群組傳出至 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly"（允許堆疊子網路中的所有本機流量）。

第二個安全群組 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnlyEgressAll-ID" 允許所有流量傳出至 0.0.0.0/0。

> **Tip**
>
> 如果您要為 AMS 變更類型選擇安全群組，例如 EC2 建立或 OpenSearch 建立網域，您可以使用此處所述的其中一個預設安全群組，或您建立的安全群組。您可以在 AWS EC2 主控台或 VPC 主控台中找到每個 VPC 的安全群組清單。

還有其他預設安全群組用於內部 AMS 用途。

AMS 預設安全群組（傳入流量）

類型	通訊協定	連接埠範圍	來源
所有流量	全部	全部	SentinelDefaultSecurityGroupPrivateOnly（限制相同安全群組成員的傳出流量）
所有流量	全部	全部	SentinelDefaultSecurityGroupPrivateOnlyEgressAll（不限制傳出流量）
HTTP、HS、SSH、	TCP	80 / 443（來源 0.0.0.0/0) 允許從堡壘存取 SSH 和 RDP	SentinelDefaultSecurityGroupPublic（不限制傳出流量）
MALZ 堡壘：			
SSH	TCP	22	SharedServices VPC CIDR 和 DMZ VPC CIDR，以及客戶提供的內部部署 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	

類型	通訊協定	連接埠範圍	來源
SALZ 堡壘：			
SSH	TCP	22	mc-initial-garden-LinuxBastionSG
SSH	TCP	22	mc-initial-garden-LinuxBastionDMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionDMZSG

建立、變更或刪除安全群組

您可以請求自訂安全群組。如果預設安全群組不符合應用程式或組織的需求，您可以修改或建立新的安全群組。這類請求會被視為需要核准，並由 AMS 操作團隊審核。

若要在堆疊和 VPCs 外部建立安全群組，請使用`Deployment | Advanced stack components | Security group | Create (review required)`變更類型 (ct-1oxx2g2d7hc90) 提交 RFC。

針對 Active Directory (AD) 安全群組修改，請使用下列變更類型：

- 若要新增使用者：使用管理 | Directory Service | 使用者和群組 | 新增使用者至群組 【ct-24pi85mjtza8k】 提交 RFC
- 若要移除使用者：使用 Management | Directory Service | User and group | Remove user from group 【ct-2019s9y3nfml4】 提交 RFC

> **Note**
>
> 使用「需要檢閱」CTs時，AMS 建議您使用 ASAP 排程選項（在主控台中選擇 ASAP，在 API/CLI 中保留開始和結束時間空白），因為這些 CTs 需要 AMS 運算子來檢查 RFC，並在核准和執行之前與您通訊。如果您排程這些 RFCs，請務必至少允許 24 小時。如果未在排定的開始時間之前進行核准，則會自動拒絕 RFC。

尋找安全群組

若要尋找連接到堆疊或執行個體的安全群組，請使用 EC2 主控台。找到堆疊或執行個體後，您可以看到連接到該堆疊或執行個體的所有安全群組。

如需在命令列尋找安全群組並篩選輸出的方法，請參閱 describe-security-groups。

附錄：應用程式加入問卷

使用此問卷來描述您的部署元素和結構，以便 AMS 可以判斷需要哪些基礎設施元件。Line-of-Business (LoB) 應用程式的加入要求與產品應用程式明顯不同，因此本問卷旨在解決兩者。

主題

部署摘要

部署的描述。例如：

- 此帳戶適用於Line-of-Business(LoB) 應用程式部署 （ 而不是產品應用程式部署 ）。
- 部署涉及帳戶公有/DMZ 子網路內的自動擴展 ARP （ 已驗證的反向代理 ）。
- Web 和應用程式伺服器將部署在帳戶的私有子網路中。
- Amazon RDS (Amazon Relational Database Service) 執行個體也會部署在帳戶的私有子網路內。
- 伺服器 (ARP、Web、應用程式、資料庫、負載平衡器等 ） 會分成不同的安全群組。
- 帳戶需要跨可用區域 (AZs) 分散的 HA （ 高可用性 ） 設計，也就是異地同步備份。

基礎設施部署元件

需要設定哪些不同的元件來支援您的應用程式？

- 區域：需要哪些 AWS 區域 或 區域？
- 高可用性 (HA)：將使用哪些可用區域？
- Virtual Private Cloud (VPC)：VPC 的 CIDR 區塊是什麼？
- 需要哪些伺服器執行個體？

  - 已驗證的反向代理 (ARP)：作業系統、AMI、執行個體類型、子網路 ID、安全群組、輸入連接埠？
  - 應用程式部署工具伺服器：OS、AMI、執行個體類型、子網路 ID、安全群組、輸入連接埠 (Chef、Puppet) 或輸出連接埠 (Ansible、Saltstack) 連接埠？
  - Amazon RDS with MySQL：資料庫版本、用量類型、執行個體類別、子網路 ID、安全群組、資料庫執行個體 ID、儲存體大小、異地同步備份、身分驗證類型、加密？
  - 儲存：您的應用程式是否無狀態？ 您需要 S3 儲存貯體嗎？ 您需要持久性儲存嗎？ 您需要 EBS 磁碟區上的靜態資料加密嗎？ 您需要資料庫加密嗎？
  - 外部 （受管服務 VPC) 伺服器端點：SMTP？ LDAP？
  - 網路需求：網路篩選 （根據安全群組？)？ Web 流量檢查 （傳入？傳出？)？
- 標記：應該使用哪些標籤將資源分組為邏輯集合？ 例如，應用程式堆疊的所有資源。為您的使用案例選取標籤；例如`backup=true`，啟用備份。此外，您必須使用 標籤`name=value`，您建立的任何 EC2 執行個體才能在主控台中顯示名稱。
- 安全群組：
  - 需要哪些安全群組？
  - 安全群組輸入規則？
  - 安全群組輸出規則？

應用程式託管平台

針對您的應用程式託管平台，請考慮下列可能的需求：

- 資料庫已加密？
- 由誰管理的加密金鑰？
- 所有資料傳輸中和靜態加密？
- 所有使用者是否透過 HTTPS 存取系統？
- 您的安全營運團隊核准的所有system-to-system互動？

應用程式部署模型

規劃應用程式部署方式的考量。請參閱我的操作模型是什麼？

- 自動化或手動？ 無部署自動化表示無自動擴展。如果您請求存取並登入並手動更新應用程式，則更新會失敗。AMS 期望您轉返更新，或透過服務請求提醒我們，以便我們為您提供協助。
- 如果自動化，架構是什麼？ 指令碼？ 以代理程式為基礎的 (puppet/chef)？ 無代理程式 (SALT/Ansible)？ CodeDeploy？ 代理程式型和無代理程式部署工具需要建立個別的執行個體，並將其部署為工具的主要伺服器。AMS 希望您了解成功部署工具所需的所有元素；不過，我們非常樂於協助解決相關的基礎設施問題。
- 您的Line-of-Business應用程式 （ 您用來建立和管理應用程式的應用程式 ） 是否需要修補？

應用程式相依性

您需要Line-of-Business(LoB) 應用程式的執行個體嗎？ 對於產品應用程式？

您的產品應用程式需要什麼才能正常運作？

- 網路層級相依性：例如， AWS Direct Connect
- 套件相依性：例如， `pip`
- 此應用程式依賴的應用程式：例如，MySql
- 防火牆相依性？

您的 LoB 應用程式需要什麼才能正常運作？

- 網路層級相依性：例如， AWS Direct Connect
- 套件相依性：例如，Firefox Saucy
- 此應用程式依賴的應用程式：例如，MySql
- 防火牆相依性？

產品應用程式的 SSL 憑證

您的伺服器需要哪些 SSL 憑證，才能讓您的應用程式 (LoB 和產品 ） 達到執行和存取所需的一切？

- Auto Scaling 群組？
- 資料庫 (Amazon RDS)？
- Load Balancer？
- 部署工具伺服器？

- Web 應用程式防火牆 (AWS WAF)？
- 其他執行個體？

例如，針對上面列出的每個執行個體，您可能需要下列憑證：

WAF （憑證 1) - > ELB-Ext （憑證 2) - > ARP （憑證 3) - > ELB-Int （憑證 4) -> 網站 （憑證 5) - > ELB-Int （憑證 6) -> Web 服務 （憑證 7)。

文件歷史記錄

下表說明此 AMS 版本的文件。

- API 版本：2019-05-21
- 文件最近更新時間：2023 年 2 月 16 日

變更	描述	連結
TOC 連結已移除	TOC AWS 詞彙表連結已移除。	2025 年 8 月 8 日
更新內容：遷移工作負載：Windows 擷取前驗證	更新章節，納入使用 WIGs前驗證程式指令碼來驗證 Windows 執行個體是否已準備好擷取至 AMS 帳戶的詳細步驟；	遷移工作負載：Windows 擷取前驗證
已更新內容、DMS 組態	有關必要角色 dms-vpc-role 的重要注意事項。	1： AWS DMS 複寫子網路群組：建立
已更新內容、CFN Ingest 支援的資源	新增 OpenSearch。	支援的資源
已更新內容、遷移工作負載	更新擷取前驗證的指示。	遷移工作負載：Windows 擷取前驗證
已更新內容 CFN 擷取。	從 CFN 擷取內容中移除限制的「支援的資源」。	CloudFormation 擷取堆疊：支援的資源
更新支援的 Windows 版本	新增對 Windows Server 2022 的支援。	AMS Amazon Machine Image AMIs)、遷

變更	描述	連結
		移工作負載：Linux 和 Windows 的先決條件 與 遷移工作負載：Windows 擷取前驗證
已更新內容、資源排程器。	更新了指示，以使用專用部署 CT ct-0ywnhc 8e5k9z5，適用於 SALZ 和 MALZ。	AMS Resource Scheduler 快速入門
已更新內容、工作負載擷取。	更新支援的 SUSE Linux 版本。	遷移工作負載：Linux 和 Windows 的先決條件
已更新內容 Database Migration Service。	已新增至先決條件，並針對實用性和可用性進行多項變更。	AWS Database Migration Service (AWS DMS)
已更新內容、工作負載擷取。	Linux Pre-WIGS Validation Zip 已更新。	遷移工作負載：Linux 和 Windows 的先決條件
已更新內容。	更新 Linux 的 WIGS 驗證前 zip。此外，新增 Windows Server 2008 R2 做為支援的作業系統。	遷移工作負載：Linux 和 Windows 的先決條件
新內容	快速入門和教學課程已從已淘汰的 AMS 進階變更管理指南移至此處。	快速入門, 教學課程.

變更	描述	連結
已更新內容	部署 | 進階堆疊元件 | 資料庫遷移服務 (DMS) | 開始複寫任務 (ct-1yq7hqse71yg) 更新以指出 DocumentName 和區域是必要的參數；先前，它們錯誤地列為選用。	資料庫遷移服務 (DMS) | 開始複寫任務
已更新內容	CloudFormation 擷取 更新以指出兩個新的支援資源：AWS::Route53Resolver::ResolverRuleAssociation 和 AWS::Route53Resolver::ResolverRule。	支援的資源
已更新內容	遷移工作負載：Windows 擷取前驗證	Sysprep 資訊已更新為更具體。 遷移工作負載：Windows 擷取前驗證
已更新內容	管理 | 自訂堆疊 | CloudFormation 範本的堆疊 | 核准變更集和更新 (ct-1404e21baa2ox) ChangeSetName 參數的 CT 演練說明已更新為額外資訊。	CloudFormation 範本的堆疊 | 核准變更集和更新
	提供 Ubuntu 18.04 和 Oracle Linux 8.3	遷移工作負載：Linux 和 Windows 的先決條件
新內容：	透過 CFN 擷取和堆疊更新 CTs IAM 部署。	2022 年 2 月 10 日

變更	描述	連結
Database Migration Service (DMS) 複寫任務	變更類型已更新，因此規則表達式允許包含連字號的任務 ARNs。 開始 AWS DMS 複寫任務和資料庫遷移服務 (DMS) | 停止複寫任務。	2022 年 1 月 13 日
Linux WIGS 擷取前驗證	zip 檔案已更新。遷移工作負載：Linux 擷取前驗證。	2022 年 1 月 13 日
已修正的連結	資料庫（資料庫）匯入 AMS SQL RDS -> 設定區段有一些錯誤的連結。	2022 年 1 月 13 日

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。