使用者指南

AWS 截止日期雲

版本 latest

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 截止日期雲: 使用者指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS 截止日期雲？

期限雲端是 AWS 服務 您可以直接從數位內容建立管道和工作站在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上建立和管理轉譯專案和任務的方式。

截止日期雲端提供主控台介面、本機應用程式、命令列工具和 API。您可以使用 Deteffate Cloud 建立、管理和監控伺服器陣列、叢集、工作、使用者群組和儲存區。您也可以指定硬體需求、為特定工作負載建立環境，以及將生產所需的內容建立工具整合到您的 Deteffate Cloud 管道中。

截止日期雲提供了一個統一的界面，可以在一個地方管理所有渲染項目。您可以管理使用者、將專案指派給使用者，以及授與工作角色的權限。

主題

截止日期雲的功能

以下是 Deptionate Cloud 可協助您執行和管理視覺化計算工作負載的一些關鍵方式：

- 快速建立您的伺服器陣列、佇列和叢集。監控其狀態，並深入瞭解伺服器陣列和工作的作業。
- 集中管理截止日期雲端使用者和群組，並指派權限。
- 使用管理專案使用者和外部身分提供者的登入安全性 AWS IAM Identity Center。
- 使用 AWS Identity and Access Management (IAM) 政策和角色，安全地管理專案資源的存取。
- 使用標籤來組織並快速尋找專案資源。
- 管理專案資源使用情況和專案的預估成本。
- 提供廣泛的運算管理選項，以支援雲端或親自轉譯。

截止日期雲端的概念和術語

為了協助您開始使用 AWS 截止日期雲端，本主題說明其一些重要概念和術語。

預算經理

預算管理器是截止日期雲監視器的一部分。使用預算管理程式來建立和管理預算。您也可以使用它來限制活動以保持在預算範圍內。

截止日期雲客戶端庫

用戶端程式庫包含用於管理期限雲端的命令列介面和程式庫。功能包括根據「開啟 Job 說明」規格將工作組合提交至 Deminate Cloud、下載工作附件輸出，以及使用命令列介面監視伺服器陣列。

數位內容建立應用程式 (DCC)

數位內容建立應用程式 (DCCS) 是您建立數位內容的協力廠商產品。DCC 的範例為MayaNuke、和 Houdini。截止日期雲為特定的 DCC 提供作業提交者集成的插件。

伺服器陣列

伺服器陣列是您專案資源所在的位置。它由隊列和艦隊組成。

機群

叢集是執行轉譯的工作者節點群組。工作者節點處理工作。一個叢集可以關聯到多個佇列，而一個佇列可以與多個叢集相關聯。

任務

工作是轉譯要求。使用者提交工作。工作包含概述為步驟和工作的特定工作屬性。

Job 附件

工作附件是截止日期雲端功能，您可以使用它來管理工作的輸入和輸出。在彩現過程中，Job 檔案會作為工作附件上載。這些文件可以是紋理，3D 模型，照明裝備和其他類似的項目。

任務屬性

Job 屬性是您在提交彩現工作時定義的設定。一些範例包括影格範圍、輸出路徑、工作附件、可彩現相機等。屬性會根據提交轉譯的 DCC 而有所不同。

任務範本

作業範本會定義執行階段環境，以及在截止日期 Cloud 工作中執行的所有程序。

佇列

佇列是提交作業所在位置並排定要呈現的位置。佇列必須與叢集相關聯，才能建立成功的轉譯。一個佇列可以與多個叢集相關聯。

佇列-艦隊關聯

佇列與叢集相關聯時，就會有一個佇列-叢集關聯。使用關聯可將工作者從叢集排定到該佇列中的工作。您可以啟動和停止關聯以控制工作排程。

步驟

步驟是要在作業中執行的一個特定程序。

截止日期雲端提交者

截止日期雲提交者是一個數字內容創建（DCC）插件。藝術家使用它來從他們熟悉的第三方 DCC 介面提交工作。

標籤

標籤是您可以指派給 AWS 資源的標籤。每個標籤都包含一個鍵和一個您定義的可選值。

使用標籤，您可以用不同的方式對資 AWS 源進行分類。例如，您可以為帳戶的 Amazon EC2 執行個體定義一組標籤，以協助您追蹤每個執行個體的擁有者和堆疊層級。

您也可以依目的、擁有者或環境來分類 AWS 資源。當您有許多相同類型的資源時，此方法非常有用。您可以根據指派給該資源的標籤，快速識別特定資源。

任務

工作是彩現步驟的單一元件。

以使用量為基礎的授權 (UBL)

使用型授權 (UBL) 是一種隨需授權模式，適用於特定第三方產品。這種模式是按需付費，並按照您使用的小時和分鐘數向您收費。

使用總管

使用資源管理器是截止日期雲監視器的功能。它提供了您的成本和用量的大致估算值。

工作程序

工作者屬於艦隊，並執行截止日期雲端指派的任務以完成步驟和工作。工作者會將任務操作的日誌存放在 Amazon CloudWatch 日誌中。工作人員也可以使用任務附件功能，將輸入和輸出同步到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。

開始使用期限雲端

使用期限雲端快速建立具有預設設定和資源的渲染伺服器陣列，例如 Amazon EC2 執行個體組態和 Amazon Simple Storage Service (Amazon S3) 儲存貯體。

您也可以在建立彩現農場時定義設定和資源。此方法比使用預設設定和資源花費更多的時間，但可讓您擁有更多控制權。

熟悉截止日期雲端概念和術語之後，請參閱入門以取得建立伺服器陣列、新增使用者和實用資訊連結的 step-by-step 指示。

存取期限雲端

您可以透過下列任何一種方式存取期限雲端：

- 截止日期雲端主控台 — 在瀏覽器中存取主控台以建立伺服器陣列及其資源，以及管理使用者存取權。如需詳細資訊，請參閱入門。
- 截止日期雲端監控 — 管理彩現工作，包括更新優先順序和工作狀態。監視伺服器陣列並檢視記錄和工作狀態。對於擁有擁有者權限的使用者，截止日期雲端監視器也提供探索使用情況和建立預算的存取。期限雲端監視器可同時作為網頁瀏覽器和桌面應用程式使用。
- AWS SDK 和 AWS CLI — 使用 AWS Command Line Interface (AWS CLI) 從本機系統上的命令列呼叫截止日期 Cloud API 作業。如需詳細資訊，請參閱設定開發人員工作站。

相關服務

截止日期雲端適用於以下項目 AWS 服務：

- Amazon CloudWatch — 使用 CloudWatch，您可以監控您的項目和相關 AWS 資源。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon EC2 — 這 AWS 服務 提供了在雲中運行應用程序的虛擬服務器。您可以將專案設定為將 Amazon EC2 執行個體用於工作負載。如需詳細資訊，請參閱 Amazon EC2 執行個體。
- Amazon EC2 Auto Scaling — 使用 Auto Scaling，您可以根據執行個體的需求發生變化，自動增加或減少執行個體數量。Auto Scaling 有助於確保您正在執行所需數量的執行個體，即使執行個體失敗也是如此。如果您啟用「使用截止日期雲端自動調整」功能，Auto Scaling 啟動的執行個體會自動向工作負載註冊。同樣地，Auto Scaling 終止的執行個體也會自動從工作負載中取消註冊。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南。

- AWS PrivateLink— 在虛擬私人雲端 (VPC) 和內部部署網路之間 AWS PrivateLink 提供私有連線，而不會將流量暴露到公用網際網路。 AWS 服務 AWS PrivateLink 可讓您輕鬆連接不同帳戶和 VPC 之間的服務。如需詳細資訊，請參閱 AWS PrivateLink。
- Amazon S3 — Amazon S3 是一種對象存儲服務。截止日期雲端使用 Amazon S3 儲存貯體來存放任務附件。
- IAM 身分中心 — IAM 身分中心 AWS 服務 可讓使用者從單一位置存取所有指派帳戶和應用程式的單一登入存取權。您也可以集中 AWS Organizations管理中所有帳戶的多帳戶存取和使用者權限。如需詳細資訊，請參閱 AWS IAM Identity Center 常見問答集。

期限雲端的運作方式

您可以使用 Detecate Cloud，直接從數位內容建立 (DCC) 管道和工作站建立和管理轉譯專案和任務。

您可以使用 AWS SDK、 AWS Command Line Interface (AWS CLI) 或截止日期雲端工作提交者，將工作提交至截止日期雲端。截止日期雲支持開放 Job 描述（OpenJD）作業模板規範。如需詳細資訊，請參閱GitHub網站上的開啟 Job 描述。

截止日期雲提供工作提交者。工作提交者是一種 DCC 外掛程式，可從協力廠商 DCC 介面 (例如或) 提交轉譯工作。Maya Nuke有了提交者，藝術家就可以從協力廠商介面將轉譯工作提交到 Depitage Cloud，在這裡管理專案資源和監控工作，全都集中在同一個位置。

您可以使用 Detection Cloud 伺服器陣列建立佇列和叢集、管理使用者，以及管理專案資源使用量和成本。伺服器陣列由佇列和艦隊組成。佇列是提交作業所在位置並排定要呈現的位置。叢集是執行工作以完成作業的工作者節點群組。佇列必須與叢集關聯，才能呈現工作。單一叢集可支援多個佇列，而多個叢集可支援一個佇列。

工作由步驟組成，每個步驟都包含特定的任務。您可以透過 Deption Cloud 監視器存取工作、步驟和工作的狀態、記錄和其他疑難排解指標。

期限雲端中的權限

截止日期雲支持以下內容：

- 使用 AWS Identity and Access Management (IAM) 管理對其 API 作業的存取
- 使用整合來管理員工使用者的存取 AWS IAM Identity Center

任何人都必須擁有該專案和相關聯伺服器陣列的存取權，才能處理專案。截止日期雲端與 IAM 身分中心整合，以管理員工身份驗證和授權。您可以直接將使用者新增至 IAM 身分中心，也可以將使用者連線到現有的身分提供者 (IdP)，例如Okta或Active Directory。IT 管理員可以將存取權限授與不同層級的使用者和群組。每個後續層級都包含先前層級的權限。下列清單說明從最低層級到最高層級的四個存取層級：

- 檢視者 — 查看伺服器陣列、佇列、叢集及其有權存取之工作中資源的權限。檢視者無法送出或變更工作。
- 貢獻者 — 與檢視者相同，但有權將工作提交至佇列或伺服器陣列。
- 管理員 — 與參與者相同，但有權編輯佇列中的工作，他們有權存取，並授與他們有權存取的資源的權限。
- 擁有者 — 與管理員相同，但可以檢視和建立預算以及查看使用情況。

> **Note**
>
> 這些權限不會授予使用者修改截止日期雲端基礎結構的存取權 AWS Management Console 或權限。

使用者必須擁有伺服器陣列的存取權，才能存取相關聯的佇列和叢集。使用者存取權會分別指派給伺服器陣列中的佇列和叢集。

您可以將使用者新增為個人或群組的一部分。將群組新增至伺服器陣列、叢集或佇列可讓您更輕鬆地管理大型人員群組的存取權限。例如，如果您有一個專案團隊正在處理特定專案，則可以將每個專案團隊成員加入至群組。然後，您可以針對對應的伺服器陣列、叢集或佇列授與整個群組的存取權限。

截止日期雲端的軟體支援

截止日期 Cloud 適用於任何可從命令列介面執行並使用參數值控制的軟體應用程式。截止日期 Cloud 支援將工作描述為具有參數化 (例如跨框架範圍) 到工作中的軟體指令碼步驟的工作的OpenJD規格。使用 Deputation Cloud 工具和功能，將工作指示組OpenJD合到工作套件中，以便從協力廠商軟體應用程式建立、執行和授權步驟。

工作需要授權才能呈現。截止日期 Cloud 針對一系列軟體應用程式授權提供以使用量為基礎的授權 (UBL)，根據使用情況按小時以分鐘為單位計費。有了截止日期雲端，您也可以視需要使用自己的軟體授權。如果工作無法存取授權，則不會轉譯並產生錯誤，並且會在 Deption Cloud 監視器的工作記錄中顯示。

開始使用期限雲端

若要在 AWS 截止日期雲端中建立伺服器陣列，您可以使用截止日期雲端主控台或 AWS Command Line Interface (AWS CLI)。使用主控台獲得建立伺服器陣列 (包括佇列和叢集) 的引導式體驗。您可以使用直 AWS CLI 接與服務搭配使用，或是開發您自己的工具，以搭配使用截止日期雲端。

若要建立伺服器陣列並使用期限雲端監視器，請為期限雲端設定您的帳戶。您只需要為每個帳戶設定一次截止日期雲端監控基礎結構。您可以在伺服器陣列中管理專案，包括伺服器陣列及其資源的使用者存取權。

若要在不設定期限雲端監視器基礎結構的情況下建立伺服器陣列，請為 Dependpoint Cloud 設定開發人員工作站

若要使用最少的資源建立伺服器陣列以接受工作，請在主控台首頁中選取快速入門。 設定截止日期雲端監控引導您完成這些步驟。這些伺服器陣列的起始時間為佇列和自動關聯的叢集。這種方法是創建沙箱風格農場進行實驗的便捷方法。

主題

- 設定您的 AWS 帳戶
- 設定截止日期雲端監控
- 為期限雲端設定開發人員工作站
- 設定截止日期雲端提交者
- 使用伺服器陣列

設定您的 AWS 帳戶

AWS 帳戶 將您的設定為使用 AWS 期限雲端。

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

當您第一次建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務 和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。

> **Important**
>
> 強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

設定截止日期雲端監控

若要開始使用，您需要建立截止日期雲端監視器基礎結構並定義您的伺服器陣列。您也可以執行其他選擇性步驟，包括新增群組和使用者、選擇服務角色，以及將標籤新增至資源。

步驟 1：設定顯示器

雲端監視器用於授權使 AWS IAM Identity Center 用者的期限。您用於截止日期雲端的 IAM 身分識別中心執行個體必須與監視器 AWS 區域 相同。如果您的主控台在建立監視器時使用不同的區域，您會收到變更 IAM 身分中心區域的提醒。

顯示器的基礎結構由下列元件組成：

- 監視器顯示名稱：監視器顯示名稱是識別監視器的方式，例如AnyCompany 監視器。監視器的名稱也會決定您的監視器 URL。

  > **Important**
  >
  > 完成設定後，您無法變更監視器顯示名稱。

- 監視器 URL：您可以使用監視器 URL 存取您的監視器。此 URL 是以「監視器」顯示名稱為基礎，例如 https://anycompanymonitor.awsapps.com。

> **Important**
>
> 完成設定後，您無法變更監視器 URL。

- AWS 區域：AWS 區域是 AWS 資料中心集合的實際位置。當您設定監視器時，[地區] 會預設為離您最近的位置。我們建議您變更「地區」，使其位於離您的使用者最近的位置。這樣可以減少延遲並提高數據傳輸速度。 AWS IAM Identity Center 必須在與截止日期雲端相 AWS 區域 同的啟用狀態。

> **Important**
>
> 完成設定期限雲端後，您就無法變更您的地區。

完成本節中的工作以設定監視器的基礎結構。

若要設定監視器的基礎結構

1. 登入以啟動「歡迎AWS Management Console使用截止日期雲端」設定，然後選擇「下一步」。
2. 輸入監視器顯示名稱 — 例如**AnyCompany Monitor**。
3. (選擇性) 若要變更監視器名稱，請選擇編輯 URL。
4. (選擇性) 若要變更為離您的AWS 區域使用者最近的位置，請選擇 [變更地區]。
   a. 選取離您使用者最近的地區。
   b. 選擇「套用區域」。

   - (選擇性) 若要新增群組和使用者，請選取(選擇性) 新增群組和使用者。
   - (選擇性) 若要進一步自訂您的監視器設定，請選取其他設定。
5. 如果您已準備好 步驟 2：定義伺服器陣列詳細，請選擇 [下一步]。

(選擇性) 新增群組和使用者

在完成截止日期雲端監視器設定之前，您可以新增監視器使用者並將其新增至群組。

安裝完成後，您可以建立新的使用者和群組，以及管理使用者，例如為他們指派群組、權限和應用程式，或從監視器中刪除使用者。

其他設定

截止日期雲端設定包括其他設定。使用這些設定，您可以檢視截止日期 Cloud 安裝程式對您的所有變更 AWS 帳戶、設定監視器使用者角色，以及變更加密金鑰類型。

AWS IAM Identity Center

AWS IAM Identity Center 是用於管理使用者和群組的雲端單一登入服務。IAM 身分中心也可與您的企業單一登入 (SSO) 提供者整合，讓使用者可以使用其公司帳戶登入。

截止日期雲端預設會啟用 IAM 身分中心，而且必須設定和使用截止日期雲端。您用於截止日期雲端的 IAM 身分識別中心執行個體必須與監視器 AWS 區域 相同。如需詳細資訊，請參閱「什麼是 AWS IAM Identity Center」。

設定服務存取角色

AWS 服務可以假設服務角色代表您執行動作。截止日期雲端需要監控使用者角色，才能讓使用者存取監視器中的資源。

您可以將 AWS Identity and Access Management (IAM) 受管政策附加到監控使用者角色。這些原則可讓使用者執行特定動作，例如在特定截止日期雲端應用程式中建立工作。由於應用程式取決於受管理原則中的特定條件，因此如果您不使用受管理的原則，應用程式可能無法如預期般執行。

您可以在完成設定後隨時變更監視器使用者角色。如需使用者角色的詳細資訊，請參閱 IAM 角色。

下列索引標籤包含兩種不同使用案例的指示。若要建立並使用新的服務角色，請選擇 [新增服務角色] 索引標籤。若要使用現有的服務角色，請選擇現有的服務角色索引標籤。

New service role

若要建立和使用新的服務角色

1. 選取建立並使用新的服務角色。
2. (選擇性) 輸入服務使用者角色名稱。
3. 如需角色的詳細資訊，請選擇 [檢視權限詳細資料]

Existing service role

若要使用現有的服務角色

1. 選取 [使用現有的服務角色]。
2. 開啟下拉式清單以選擇現有的服務角色。
3. (選擇性) 選擇 IAM 主控台中的檢視，以取得有關該角色的詳細資訊。

步驟 2：定義伺服器陣列詳細

返回截止日期雲端主控台，完成下列步驟以定義伺服器陣列詳細資料。

1. 在伺服器陣列詳細資料中，新增伺服器陣列的名稱。
2. 在說明中，輸入伺服器陣列說明。清楚的說明可協助您快速識別伺服器陣列的用途。
3. (選擇性) 根據預設，您的資料會使用 AWS 擁有並管理您的安全性的金鑰加密。您可以選擇 [自訂加密設定 (進階)] 以使用現有的金鑰或建立您管理的新金鑰。

   如果您選擇使用核取方塊自訂加密設定，請輸入 AWS KMS ARN，或選擇建立新的 KMS 金鑰 AWS KMS 來建立新金鑰。
4. (選擇性) 選擇 [新增標籤]，將一或多個標籤新增至伺服器陣列。
5. 請選擇下列其中一個選項：

   - 選取 [略過檢閱和建立] 以檢閱並建立您的伺服器陣列。
   - 選取「下一步」以繼續執行其他選擇性步驟。

(選擇性) 步驟 3：定義佇列詳細資訊

佇列負責追蹤工作的進度和排程工作。

1. 從佇列詳細資料開始，提供佇列的名稱。
2. 在說明中，輸入佇列說明。清楚的說明可協助您快速識別佇列的用途。
3. 對於 Job 務附件，您可以建立新的 Amazon S3 儲存貯體，或選擇現有的 Amazon S3 儲存貯體。如果您沒有現有的 Amazon S3 儲存貯體，則需要建立一個儲存貯體。

   a. 若要建立新的 Amazon S3 儲存貯體，請選取建立新的任務儲存貯體。您可以在「根字首」欄位中定義工作時段的名稱。我們建議您呼叫值區**deadlinecloud-job-attachments-[MONITORNAME]**。

      您只能使用小寫字母和破折號。不可使用空格或特殊字元。

b. 若要搜尋並選取現有的 Amazon S3 儲存貯體，請選取「從現有的 Amazon S3 儲存貯體中選擇」。然後，選擇瀏覽 S3 搜尋現有儲存貯體。顯示可用的 Amazon S3 儲存貯體清單時，請選取要用於佇列的 Amazon S3 儲存貯體。

4. 如果您使用客戶管理的叢集，請選取 [啟用與客戶管理的叢集關聯]。

   - 若為客戶管理的叢集，請新增已設定佇列的使用者，然後設定 POSIX 及/或 Windows 認證。或者，您可以選取核取方塊來略過執行身分功能。

5. 您的佇列需要權限才能代表您存取 Amazon S3。建議您為每個佇列建立新的服務角色。

   a. 對於新角色，請完成以下步驟。

      i. 選取建立並使用新的服務角色。

      ii. 輸入佇列角色的角色名稱，或使用提供的角色名稱。

      iii. (選擇性) 新增佇列角色說明。

      iv. 您可以選擇檢視權限詳細資料，以檢視佇列角色的 IAM 許可。

   b. 或者，您可以選擇現有的服務角色。

6. (選擇性) 使用名稱和值配對新增佇列環境的環境變數。

7. (選擇性) 使用索引鍵和值配對為佇列新增標籤。

輸入所有佇列詳細資訊之後，請選取 [下一步]。

(選擇性) 步驟 4：定義叢集詳細資訊

叢集會分配 Worker 來執行您的轉譯工作。如果您的轉譯工作需要叢集，請核取 [建立叢集] 核取方塊。

1. 車隊詳情

   a. 為您的叢集提供 [名稱] 和 [選擇性說明]。

   b. 選取運算資源應擴展的方式。服務管理選項允許截止日期雲端自 auto 擴展您的計算資源。客戶管理選項讓您可以控制自己的計算擴展。

2. 在「執行個體選項」區段中，選擇 Spot 或隨需。Amazon EC2 隨需執行個體提供更快的可用性，而且 Amazon EC2 競價型執行個體更能節省成本。

3. 對於 Auto Scaling 叢集中的執行個體數量，請選擇執行個體數目下限和執行個體數目上限。

   我們強烈建議您一律設定執行個體數量下限，**0**以避免產生額外費用。

4. 您的叢集需要代表您寫入 CloudWatch 的權限。我們建議您為每個叢集建立新的服務角色。

a. 對於新角色，請完成以下步驟。

i. 選取建立並使用新的服務角色。

ii. 輸入叢集角色的角色名稱，或使用提供的角色名稱。

iii. (選擇性) 新增叢集角色說明。

iv. 您可以選擇檢視權限詳細資料，以檢視叢集角色的 IAM 許可。

b. 或者，您可以使用現有的服務角色。

5. (選擇性) 使用金鑰和值配對為叢集新增標籤。

輸入所有叢集詳細資料後，請選取 [下一步]。

(選擇性) 步驟 5：設定 Worker 需求

定義 Worker 實例的需求。

1. 檢閱作業系統 (OS) 和 CPU 架構設定以進行感知。
2. 根據您的硬體需求，更新 vCPUs 的最小和上限數目。
3. 根據您的硬體需求，更新最小和最大記憶體數目 (GiB)。
4. 您可以透過允許或排除 Worker 執行個體類型來篩選執行個體類型。在這兩個篩選選項中，您最多可以篩選 10 個 Amazon EC2 執行個體類型。
5. 在其他需求 (選用) 下，您可以依據大小 (GiB)、IOPS 和輸送量 (MIB/s) 來定義根 EBS 磁碟區。
6. 設定完所有 Worker 需求之後，選擇 [下一步] 以定義群組的存取層級。

(選擇性) 步驟 6：定義存取層級

如果您有群組連線到監視器，您可以定義其存取層級。截止日期雲端功能的使用權限由存取層級管理。您可以將不同的存取層級指派給使用者群組。

1. 使用截止日期雲端伺服器陣列存取層級功能表來選取群組的權限層級。
2. 選擇 [下一步] 以繼續並檢閱輸入的所有伺服器陣列

步驟 7：檢閱並建立

檢閱所有輸入的資訊以建立伺服器陣列。準備就緒後，請選擇 [建立農場]。

伺服器陣列的建立進度會顯示在 [伺服器陣列] 頁面上。當您的伺服器陣列可供使用時，會顯示成功訊息。

為期限雲端設定開發人員工作站

在本教學課程中，您將使用 AWS CloudShell 建立簡單的開發人員伺服器陣列，並執行 Worker Agent。然後，您可以提交並執行包含參數和附件的簡單工作、新增服務受管理的叢集，以及在完成後清理伺服器陣列資源。

以下各節將向您介紹截止日期雲端的不同功能，以及它們如何運作和協同運作。遵循這些步驟對於開發和測試新的工作負載和自訂非常有用。

主題

步驟 1：建立截止日期雲端伺服器陣列

若要在 Deptionate Cloud 中建立開發人員伺服器陣列和佇列資源， AWS 請使用 AWS Command Line Interface (AWS CLI)，如下列程序所示。您也會建立 AWS Identity and Access Management (IAM) 角色和客戶管理叢集 (CMF)，並將叢集與佇列建立關聯。然後，您可以設定 AWS CLI 並確認您的伺服器陣列已按照指定的方式進行設定和運作。

您可以使用此伺服器陣列來探索 Deputation Cloud 的功能，然後開發和測試新的工作負載、自訂和管道整合。

若要建立伺服器陣列

1. 安裝和配置 AWS Command Line Interface （ AWS CLI ），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.
2. 為您的伺服器陣列建立名稱，並將該伺服器陣列名稱新增至`~/.bashrc`。這將使其可用於其他終端會話。

```
echo "DEV_FARM_NAME=DeveloperFarm" >> ~/.bashrc
source ~/.bashrc
```

3. 建立伺服器陣列資源，並將其伺服器陣列識別碼新增至~/.bashrc。

```
aws deadline create-farm \
    --display-name "$DEV_FARM_NAME"

echo "DEV_FARM_ID=\$(aws deadline list-farms \
        --query \"farms[?displayName=='\$DEV_FARM_NAME'].farmId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

4. 建立佇列資源，並將其佇列 ID 新增至 ~/.bashrc.

```
aws deadline create-queue \
    --farm-id $DEV_FARM_ID \
    --display-name "$DEV_FARM_NAME Queue" \
    --job-run-as-user '{"posix": {"user": "job-user", "group": "job-group"},
 "runAs":"QUEUE_CONFIGURED_USER"}'

echo "DEV_QUEUE_ID=\$(aws deadline list-queues \
        --farm-id \$DEV_FARM_ID \
        --query \"queues[?displayName=='\$DEV_FARM_NAME Queue'].queueId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

5. 為叢集建立 IAM 角色。此角色為叢集中的 Worker 主機提供必要的安全性認證，以便從佇列執行工作。

```
aws iam create-role \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --assume-role-policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "credentials.deadline.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
```

```
                }
            ]
        }'
aws iam put-role-policy \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --policy-name WorkerPermissions \
    --policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": [
                        "deadline:AssumeFleetRoleForWorker",
                        "deadline:UpdateWorker",
                        "deadline:DeleteWorker",
                        "deadline:UpdateWorkerSchedule",
                        "deadline:BatchGetJobEntity",
                        "deadline:AssumeQueueRoleForWorker"
                    ],
                    "Resource": "*",
                    "Condition": {
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "logs:CreateLogStream"
                    ],
                    "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
                    "Condition": {
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "logs:PutLogEvents",
                        "logs:GetLogEvents"
```

```
                    ],
                    "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
                    "Condition": {
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                }
            ]
        }'
```

6. 建立客戶管理的叢集 (CMF)，並將其叢集 ID 新增至~/.bashrc。

```
FLEET_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
        --query "Account" --output text):role/${DEV_FARM_NAME}FleetRole"
aws deadline create-fleet \
    --farm-id $DEV_FARM_ID \
    --display-name "$DEV_FARM_NAME CMF" \
    --role-arn $FLEET_ROLE_ARN \
    --max-worker-count 5 \
    --configuration \
        '{
            "customerManaged": {
                "mode": "NO_SCALING",
                "workerCapabilities": {
                    "vCpuCount": {"min": 1},
                    "memoryMiB": {"min": 512},
                    "osFamily": "linux",
                    "cpuArchitectureType": "x86_64"
                }
            }
        }'

echo "DEV_CMF_ID=\$(aws deadline list-fleets \
        --farm-id \$DEV_FARM_ID \
        --query \"fleets[?displayName=='\$DEV_FARM_NAME CMF'].fleetId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

7. 確保您可以訪問期限雲。

```
pip install deadline
```

8. 將 CMF 與您的佇列產生關聯。

```
aws deadline create-queue-fleet-association \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --fleet-id $DEV_CMF_ID
```

9. 若要將預設伺服器陣列設定為伺服器陣列識別碼，並將佇列設定為您先前建立的佇列識別碼，請使用下列命令。

```
deadline config set defaults.farm_id $DEV_FARM_ID
deadline config set defaults.queue_id $DEV_QUEUE_ID
```

10. (選擇性) 若要確認伺服器陣列已根據您的規格進行設定，請使用下列命令：

- 列出所有農場 — **`deadline farm list`**
- 列出預設伺服器陣列中的所有佇列 — **`deadline queue list`**
- 列出預設伺服器陣列中的所有艦隊 — **`deadline fleet list`**
- 取得預設伺服器陣列 — **`deadline farm get`**
- 獲取默認隊列-**`deadline queue get`**
- 獲取與默認隊列關聯的所有艦隊-**`deadline fleet get`**

步驟 2：在截止日期雲中以開發人員模式運行工作代理

在您可以執行提交至開發人員伺服器陣列上佇列的工作之前，您必須在背景工作者主機上以開發人員模式執行 Dependate Cloud Worker 代理程式。 AWS

在本教學課程的其餘部分中，您將使用兩個 AWS CloudShell 索引標籤在開發人員伺服器陣列上執行 AWS CLI 作業。在第一個選項卡中，您可以提交工作。在第二個索引標籤中，您可以執行 Worker 代理程式。

> **Note**
>
> 如果您讓工 CloudShell 作階段閒置超過 20 分鐘，它將逾時並停止 Worker 代理程式。若要重新啟動 Worker 代理程式，請遵循下列程序中的指示。

在開發人員模式中執行 Worker 代理程式

1. 安裝和配置 AWS Command Line Interface （ AWS CLI ），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.

2. 在伺服器陣列仍在第一個 CloudShell 索引標籤中開啟的情況下，開啟第二個 CloudShell 索引標籤，然後建立`demoenv-logs`和`demoenv-persist`目錄。

```
mkdir ~/demoenv-logs
mkdir ~/demoenv-persist
```

3. 從 PyPI 下載並安裝期限雲端工作者代理程式套件：

> **Note**
>
> 在上Windows，需要將代理程式檔案安裝到 Python 的全域網站套件目錄中。目前不支援 Python 虛擬環境。

```
python -m pip install deadline-cloud-worker-agent
```

4. 若要允許 Worker 代理程式建立執行中工作的暫存目錄，請建立目錄：

```
sudo mkdir /sessions
sudo chmod 750 /sessions
sudo chown cloudshell-user /sessions
```

5. 在開發人員模式下執行截止日期雲端工作者代理程式`DEV_FARM_ID`，`DEV_CMF_ID`其中包含您新增到`~/.bashrc`.

```
deadline-worker-agent \
    --farm-id $DEV_FARM_ID \
    --fleet-id $DEV_CMF_ID \
    --run-jobs-as-agent-user \
    --logs-dir ~/demoenv-logs \
    --persistence-dir ~/demoenv-persist
```

當 Worker 代理程式初始化，然後輪詢 `UpdateWorkerSchedule` API 作業時，會顯示下列輸出：

```
INFO    Worker Agent starting
[2024-03-27 15:51:01,292][INFO    ] # Worker Agent starting
```

```
[2024-03-27 15:51:01,292][INFO    ] AgentInfo
Python Interpreter: /usr/bin/python3
Python Version: 3.9.16 (main, Sep  8 2023, 00:00:00)  - [GCC 11.4.1 20230605 (Red
 Hat 11.4.1-2)]
Platform: linux
...
[2024-03-27 15:51:02,528][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params={'assignedSessions': {}, 'cancelSessionActions': {},
 'updateIntervalSeconds': 15} ...
[2024-03-27 15:51:17,635][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params=(Duplicate removed, see previous response) ...
[2024-03-27 15:51:32,756][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params=(Duplicate removed, see previous response) ...
...
```

6. 選取您的第一個 CloudShell 索引標籤，然後列出叢集中的 Worker。

```
deadline worker list --fleet-id $DEV_CMF_ID
```

會顯示如下的輸出：

```
Displaying 1 of 1 workers starting at 0

- workerId: worker-8c9af877c8734e89914047111f
  status: STARTED
  createdAt: 2023-12-13 20:43:06+00:00
```

在生產組態中，Definition Cloud Worker 代理程式需要在主機上將多個使用者和組態目錄設定為系統管理使用者。您可以覆寫這些設定，因為您在自己的開發伺服器陣列中執行工作 (只有您可以存取)。

步驟 3：使用截止日期雲端提交和執行工作

若要使用 AWS 期限雲端執行工作，請使用下列程序。使用第一個 AWS CloudShell 索引標籤，將工作提交至開發人員伺服器陣列 使用第二個 CloudShell 索引標籤來檢視 Worker 代理程式輸出。

主題

- 提交simple_job樣本
- 提交一個simple_job帶有參數
- 使用檔案 I/O 建立簡單的工作套裝軟體

提交simple_job樣本

建立伺服器陣列並執行背景工作者代理程式之後，您可以將simple_job範例提交至截止日期雲端。

將simple_job樣本提交到期限雲

1. 安裝和配置 AWS Command Line Interface（AWS CLI），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.
2. 從下載樣本 GitHub。

```
cd ~
git clone https://github.com/aws-deadline/deadline-cloud-samples.git
```

3. 選擇您的第一個 CloudShell 索引標籤，然後瀏覽至工作套件範例目錄。

```
cd ~/deadline-cloud-samples/job_bundles/
```

4. 提交simple_job樣本。

```
deadline bundle submit simple_job
```

5. 選擇第二個 CloudShell 索引標籤，以檢視有關呼叫BatchGetJobEntities、取得工作階段和執行工作階段動作的記錄輸出。

```
...
[2024-03-27 16:00:21,846][INFO    ] # Session.Starting
 # [session-053d77cef82648fe2] Starting new Session.
 [queue-3ba4ff683ff54db09b851a2ed8327d7b/job-d34cc98a6e234b6f82577940ab4f76c6]
[2024-03-27 16:00:21,853][INFO    ] # API.Req # [deadline:BatchGetJobEntity]
 resource={'farm-id': 'farm-3e24cfc9bbcd423e9c1b6754bc1',
 'fleet-id': 'fleet-246ee60f46d44559b6cce010d05', 'worker-id':
 'worker-75e0fce9c3c344a69bff57fcd83'} params={'identifiers': [{'jobDetails':
 {'jobId': 'job-d34cc98a6e234b6f82577940ab4'}}]} request_url=https://
scheduling.deadline.us-west-2.amazonaws.com/2023-10-12/farms/
farm-3e24cfc9bbcd423e /fleets/fleet-246ee60f46d44559b1 /workers/worker-
 75e0fce9c3c344a69b /batchGetJobEntity
[2024-03-27 16:00:22,013][INFO    ] # API.Resp # [deadline:BatchGetJobEntity](200)
 params={'entities': [{'jobDetails': {'jobId': 'job-d34cc98a6e234b6f82577940ab6',
 'jobRunAsUser': {'posix': {'user': 'job-user', 'group': 'job-group'},
 'runAs': 'QUEUE_CONFIGURED_USER'}, 'logGroupName': '/aws/deadline/
farm-3e24cfc9bbcd423e9c1b6754bc1/queue-3ba4ff683ff54db09b851a2ed83', 'parameters':
```

```
 '*REDACTED*', 'schemaVersion': 'jobtemplate-2023-09'}}], 'errors': []}
 request_id=a3f55914-6470-439e-89e5-313f0c6
[2024-03-27 16:00:22,013][INFO    ] # Session.Add #
 [session-053d77cef82648fea9c69827182] Appended new SessionActions.
 (ActionIds: ['sessionaction-053d77cef82648fea9c69827182-0'])
 [queue-3ba4ff683ff54db09b851a2ed8b/job-d34cc98a6e234b6f82577940ab6]
[2024-03-27 16:00:22,014][WARNING ] # Session.User #
 [session-053d77cef82648fea9c69827182] Running as the Worker Agent's
 user. (User: cloudshell-user) [queue-3ba4ff683ff54db09b851a2ed8b/job-
d34cc98a6e234b6f82577940ac6]
[2024-03-27 16:00:22,015][WARNING ] # Session.AWSCreds #
 [session-053d77cef82648fea9c69827182] AWS Credentials are not available: Queue has
 no IAM Role. [queue-3ba4ff683ff54db09b851a2ed8b/job-d34cc98a6e234b6f82577940ab6]
[2024-03-27 16:00:22,026][INFO    ] # Session.Logs #
 [session-053d77cef82648fea9c69827182] Logs streamed to: AWS CloudWatch
 Logs. (LogDestination: /aws/deadline/farm-3e24cfc9bbcd423e9c1b6754bc1/
queue-3ba4ff683ff54db09b851a2ed83/session-053d77cef82648fea9c69827181)
 [queue-3ba4ff683ff54db09b851a2ed83/job-d34cc98a6e234b6f82577940ab4]
[2024-03-27 16:00:22,026][INFO    ] # Session.Logs #
 [session-053d77cef82648fea9c69827182] Logs streamed to: local
 file. (LogDestination: /home/cloudshell-user/demoenv-logs/
queue-3ba4ff683ff54db09b851a2ed8b/session-053d77cef82648fea9c69827182.log)
 [queue-3ba4ff683ff54db09b851a2ed83/job-d34cc98a6e234b6f82577940ab4]
...
```

> **Note**
>
> 只會顯示 Worker 代理程式的記錄輸出。執行工作的階段作業有個別的記錄。

6. 選擇您的第一個索引標籤，然後檢查 Worker 代理程式所寫入的記錄檔。

   a. 瀏覽至 Worker 代理程式記錄目錄並檢視其內容。

   ```
   cd ~/demoenv-logs
   ls
   ```

   b. 列印 Worker 代理程式建立的第一個記錄檔。

   ```
   cat worker-agent-bootstrap.log
   ```

此檔案包含背景工作者代理程式輸出，說明如何呼叫 Dependate Cloud API 在叢集中建立背景工作者資源，然後擔任叢集角色。

c. 當 Worker 代理程式加入叢集時，列印記錄檔輸出。

```
cat worker-agent.log
```

此記錄檔包含 Worker Agent 所執行之所有動作的相關輸出，但不包含從其執行工作之佇列的輸出，但這些資源的 ID 除外。

d. 在與佇列資源 ID 相同的目錄中，列印每個階段作業的記錄檔。

```
cat $DEV_QUEUE_ID/session-*.log
```

如果工作成功，記錄檔輸出將類似下列內容：

```
 cat $DEV_QUEUE_ID/$(ls -t $DEV_QUEUE_ID | head -1)
2024-03-27 16:00:22,026 WARNING Session running with no AWS Credentials.
2024-03-27 16:00:22,404 INFO
2024-03-27 16:00:22,405 INFO ================================================
2024-03-27 16:00:22,405 INFO --------- Running Task
2024-03-27 16:00:22,405 INFO ================================================
2024-03-27 16:00:22,406 INFO ------------------------------------------------
2024-03-27 16:00:22,406 INFO Phase: Setup
2024-03-27 16:00:22,406 INFO ------------------------------------------------
2024-03-27 16:00:22,406 INFO Writing embedded files for Task to disk.
2024-03-27 16:00:22,406 INFO Mapping: Task.File.runScript -> /sessions/
session-053d77cef82648fea9c698271812a/embedded_fileswa_gj55_/tmp2u9yqtsz
2024-03-27 16:00:22,406 INFO Wrote: runScript -> /sessions/
session-053d77cef82648fea9c698271812a/embedded_fileswa_gj55_/tmp2u9yqtsz
2024-03-27 16:00:22,407 INFO ------------------------------------------------
2024-03-27 16:00:22,407 INFO Phase: Running action
2024-03-27 16:00:22,407 INFO ------------------------------------------------
2024-03-27 16:00:22,407 INFO Running command /sessions/
session-053d77cef82648fea9c698271812a/tmpzuzxpslm.sh
2024-03-27 16:00:22,414 INFO Command started as pid: 471
2024-03-27 16:00:22,415 INFO Output:
2024-03-27 16:00:22,420 INFO Welcome to AWS Deadline Cloud!
2024-03-27 16:00:22,571 INFO
2024-03-27 16:00:22,572 INFO ================================================
2024-03-27 16:00:22,572 INFO --------- Session Cleanup
2024-03-27 16:00:22,572 INFO ================================================
```

```
2024-03-27 16:00:22,572 INFO Deleting working directory: /sessions/
session-053d77cef82648fea9c698271812a
```

7. 列印工作的相關資訊。

```
deadline job get
```

當您提交工作時，系統會將其儲存為預設值，因此您不需要輸入工作 ID。

提交一個simple_job帶有參數

您可以使用參數提交工作。在下列程序中，您可以編輯simple_job範本以包含自訂訊息、提交simple_job，然後列印工作階段記錄檔以檢視訊息。

若要使用參數提交simple_job樣本

1. 選取您的第一個 CloudShell 索引標籤，然後瀏覽至工作套件範例目錄。

```
cd ~/deadline-cloud-samples/job_bundles/
```

2. 列印simple_job範本的內容。

```
cat simple_job/template.yaml
```

帶有Message參數的parameterDefinitions部分應如下所示：

```
parameterDefinitions:
- name: Message
  type: STRING
  default: Welcome to AWS Deadline Cloud!
```

3. 提交具有參數值的simple_job樣本，然後等待工作完成執行。

```
deadline bundle submit simple_job \
    -p "Message=Greetings from the developer getting started guide."
```

4. 若要查看自訂訊息，請檢視最新的工作階段記錄檔。

```
cd ~/demoenv-logs
cat $DEV_QUEUE_ID/$(ls -t $DEV_QUEUE_ID | head -1)
```

使用檔案 I/O 建立簡單的工作套裝軟體

彩現工作需要讀取場景定義、彩現影像，然後將該影像儲存至輸出檔案。您可以使工作計算輸入的雜湊值，而不是彩現影像，以模擬此動作。

使用檔案 I/O 建立簡單的工作套裝軟體

1. 選取您的第一個 CloudShell 索引標籤，然後瀏覽至工作套件範例目錄。

```
cd ~/deadline-cloud-samples/job_bundles/
```

2. 使用新名稱複製一份simple_file_job。simple_job

```
cp -r simple_job simple_file_job
```

3. 編輯工作樣板，如下所示：

> **Note**
>
> 我們建議您使nano用這些步驟。如果您偏好使用Vim，則必須使用設定其貼上模式:set paste。

a. 在文字編輯器中開啟範本。

```
nano simple_file_job/template.yaml
```

b. 新增下列type、objectType、和dataFlowparameterDefinitions。

```
- name: InFile
  type: PATH
  objectType: FILE
  dataFlow: IN
- name: OutFile
  type: PATH
  objectType: FILE
  dataFlow: OUT
```

c. 將以下bash腳本命令添加到從輸入文件讀取並寫入輸出文件的文件末尾。

```
        # hash the input file, and write that to the output
```

```
        sha256sum "{{Param.InFile}}" > "{{Param.OutFile}}"
```

更新後template.yaml應完全符合以下內容：

```
specificationVersion: 'jobtemplate-2023-09'
name: Simple File Job Bundle Example
parameterDefinitions:
- name: Message
  type: STRING
  default: Welcome to AWS Deadline Cloud!
- name: InFile
  type: PATH
  objectType: FILE
  dataFlow: IN
- name: OutFile
  type: PATH
  objectType: FILE
  dataFlow: OUT
steps:
- name: WelcomeToDeadlineCloud
  script:
    actions:
      onRun:
        command: '{{Task.File.runScript}}'
    embeddedFiles:
    - name: runScript
      type: TEXT
      runnable: true
      data: |
        #!/usr/bin/env bash
        echo "{{Param.Message}}"

        # hash the input file, and write that to the output
        sha256sum "{{Param.InFile}}" > "{{Param.OutFile}}"
```

> **Note**
>
> 如果您要調整中的間距template.yaml，請確定您使用空格而非縮排。

d. 儲存檔案，然後結束文字編輯器。

4. 為輸入和輸出檔案提供參數值，以提交簡單檔案工作。

```
deadline bundle submit simple_file_job \
    -p "InFile=simple_job/template.yaml" \
    -p "OutFile=hash.txt"
```

5. 列印工作的相關資訊。

```
deadline job get
```

- 您將看到如下輸出：

```
parameters:
  Message:
    string: Welcome to AWS Deadline Cloud!
  InFile:
    path: /local/home/cloudshell-user/BundleFiles/JobBundle-Examples/simple_job/
template.yaml
  OutFile:
    path: /local/home/cloudshell-user/BundleFiles/JobBundle-Examples/hash.txt
```

- 雖然您只提供相對路徑，但參數已設定完整路徑。當路徑具有類型時，會將目前工作目錄 AWS CLI 連接到作為參數提供的任何路徑PATH。
- 在其他終端機視窗中執行的 Worker Agent 會接收並執行工作。此操作將創建`hash.txt`文件，您可以使用以下命令查看該文件。

```
cat hash.txt
```

該命令將打印類似於以下內容的輸出。

```
eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3 /local/home/
cloudshell-user/BundleFiles/JobBundle-Examples/simple_job/template.yaml
```

步驟 4：在截止日期雲端中執行含有工作附件的工作

許多伺服器陣列會使用共用檔案系統，在提交工作的主機與執行工作的主機之間共用檔案。例如，在前面的`simple_file_job`範例中，本機檔案系統會在終端機視窗之間共用， AWS CloudShell 終端機視窗會在您提交工作的索引標籤一中執行，以及在您執行 Worker Agent 的標籤 2 中執行。

當提交者工作站和 Worker 主機位於相同的區域網路上時，共用檔案系統是有利的。如果您將資料儲存在存取資料的工作站附近的內部部署，則使用雲端伺服器陣列表示您必須透過高延遲 VPN 共用檔案系統，或在雲端中同步檔案系統。這些選項都不易於設置或操作。

AWS 截止日期雲提供了一個簡單的解決方案與工作附件，這是類似於電子郵件附件。使用工作附件，您可以將資料附加至工作。然後，截止日期雲端會處理在 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體中傳輸和存放任務資料的詳細資訊。

內容建立工作流程通常是反覆運算的，這意味著使用者提交包含一小部分修改檔案的工作。由於 Amazon S3 儲存貯體將任務附件存放在可內容定址的儲存中，因此每個物件的名稱均以物件資料的雜湊值為基礎，而目錄樹狀結構的內容則以連接到任務的資訊清單檔案格式儲存。

若要執行含有工作附件的工作，請完成以下步驟。

主題

- 將工作附件組態新增至佇列
- simple_file_job使用工作附件提交
- 了解任務附件在 Amazon S3 中的存放方式

將工作附件組態新增至佇列

若要啟用佇列中的工作附件，請將工作附件組態新增至帳號中的佇列資源。

若要將工作附件組態新增至佇列

1. 安裝和配置 AWS Command Line Interface（AWS CLI），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.
2. 選擇您的第一個 CloudShell 索引標籤，然後輸入下列其中一個命令，將 Amazon S3 儲存貯體用於任務附件。
    - 如果您沒有現有的私有 Amazon S3 儲存貯體，可以建立和使用新的 S3 儲存貯體。

```
DEV_FARM_BUCKET=$(echo $DEV_FARM_NAME \
    | tr '[:upper:]' '[:lower:]')-$(xxd -l 16 -p /dev/urandom)
if [ "$AWS_REGION" == "us-east-1" ]; then LOCATION_CONSTRAINT=
else LOCATION_CONSTRAINT="--create-bucket-configuration \
    LocationConstraint=${AWS_REGION}"
fi
aws s3api create-bucket \
    $LOCATION_CONSTRAINT \
```

```
    --acl private \
    --bucket ${DEV_FARM_BUCKET}
```

- 如果您已經擁有私有 Amazon S3 儲存貯體，則可以使用儲存貯體*MY_BUCKET_NAME*的名稱來使用它。

```
DEV_FARM_BUCKET=MY_BUCKET_NAME
```

3. 建立或選擇 Amazon S3 儲存貯體之後，請新增儲存貯體名稱，`~/.bashrc`讓儲存貯體可用於其他終端機工作階段。

```
echo "DEV_FARM_BUCKET=$DEV_FARM_BUCKET" >> ~/.bashrc
```

4. 為佇列建立 AWS Identity and Access Management (IAM) 角色。

```
aws iam create-role --role-name "${DEV_FARM_NAME}QueueRole" \
    --assume-role-policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "credentials.deadline.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }'
aws iam put-role-policy \
    --role-name "${DEV_FARM_NAME}QueueRole" \
    --policy-name S3BucketsAccess \
    --policy-document \
            '{
                "Version": "2012-10-17",
                "Statement": [
                {
                    "Action": [
                        "s3:GetObject*",
                        "s3:GetBucket*",
                        "s3:List*",
                        "s3:DeleteObject*",
                        "s3:PutObject",
```

```
                    "s3:PutObjectLegalHold",
                    "s3:PutObjectRetention",
                    "s3:PutObjectTagging",
                    "s3:PutObjectVersionTagging",
                    "s3:Abort*"
                ],
                "Resource": [
                    "arn:aws:s3:::'$DEV_FARM_BUCKET'",
                    "arn:aws:s3:::'$DEV_FARM_BUCKET'/*"
                ],
                "Effect": "Allow"
            }
        ]
        }'
```

5. 更新佇列以包含工作附件設定和 IAM 角色。

```
QUEUE_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
        --query "Account" --output text):role/${DEV_FARM_NAME}QueueRole"
aws deadline update-queue \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --role-arn $QUEUE_ROLE_ARN \
    --job-attachment-settings \
        '{
            "s3BucketName": "'$DEV_FARM_BUCKET'",
            "rootPrefix": "JobAttachments"
        }'
```

6. 確認您已更新佇列。

```
deadline queue get
```

如下所示的輸出：

```
...
jobAttachmentSettings:
  s3BucketName: DEV_FARM_BUCKET
  rootPrefix: JobAttachments
roleArn: arn:aws:iam::ACCOUNT_NUMBER:role/DeveloperFarmQueueRole
...
```

simple_file_job使用工作附件提交

使用工作附件時，工作服務包必須提供足夠的資訊來決定工作的資料流程，例如使用PATH參數。在的情況下simple_file_job，您已編輯template.yaml檔案，以告知 Perfday Cloud 資料流程位於輸入檔案和輸出檔案中。

將工作附件組態新增至佇列後，您可以提交包含工作附件的 simple_file_job 範例。執行此操作之後，您可以檢視記錄和工作輸出，以確認simple_file_job包含工作附件的工作正在運作。

若要提交含有工作附件的簡單檔案工作套裝軟體

1. 選擇您的第一個選 CloudShell 項卡，然後打開JobBundle-Samples目錄。
2.
```
cd ~/AmazonDeadlineCloud-DocumentationAndSamples/JobBundle-Samples
```
3. 將簡單檔案提交至佇列。當系統提示您確認上傳時，請輸入**y**。

```
deadline bundle submit simple_file_job \
    -p InFile=simple_job/template.yaml \
    -p OutFile=hash-jobattachments.txt
```
4. 若要檢視工作附件資料傳輸階段作業記錄輸出，請選擇第二個 CloudShell 標籤。

```
JOB_ID=$(deadline config get defaults.job_id)
SESSION_ID=$(aws deadline list-sessions \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --job-id $JOB_ID \
        --query "sessions[0].sessionId" \
        --output text)
cat ~/demoenv-logs/$DEV_QUEUE_ID/$SESSION_ID.log
```
5. 列出在工作階段中執行的工作階段動作。

```
aws deadline list-session-actions \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --job-id $JOB_ID \
    --session-id $SESSION_ID
```

   如下所示的輸出：

```
{
    "sessionactions": [
        {
            "sessionActionId": "sessionaction-123-0",
            "status": "SUCCEEDED",
            "startedAt": "<timestamp>",
            "endedAt": "<timestamp>",
            "progressPercent": 100.0,
            "definition": {
                "syncInputJobAttachments": {}
            }
        },
        {
            "sessionActionId": "sessionaction-123-1",
            "status": "SUCCEEDED",
            "startedAt": "<timestamp>",
            "endedAt": "<timestamp>",
            "progressPercent": 100.0,
            "definition": {
                "taskRun": {
                    "taskId": "task-abc-0",
                    "stepId": "step-def"
                }
            }
        }
    ]
}
```

第一個工作階段動作會下載輸入工作附件，而第二個動作會像之前一樣執行工作，然後上傳輸出工作附件。

6. 列出輸出目錄。

```
ls *.txt
```

如顯示hash.txt的輸出，但hash-jobattachments.txt不存在。

7. 從最近的工作下載輸出。

```
deadline job download-output
```

8. 檢視下載檔案的輸出。

```
cat hash-jobattachments.txt
```

如下所示的輸出：

```
eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3  /tmp/openjd/
session-123/assetroot-abc/simple_job/template.yaml
```

了解任務附件在 Amazon S3 中的存放方式

您可以使用 AWS Command Line Interface (AWS CLI) 上傳或下載任務附件的資料，這些附件存放在 Amazon S3 儲存貯體。了解截止日期雲端在 Amazon S3 上存放任務附件如何協助您開發工作負載和管道整合。

檢查截止日期雲端任務附件在 Amazon S3 中的存放方式

1. 選擇您的第一個 CloudShell 索引標籤，然後開啟工作套件範例目錄。

```
cd ~/AmazonDeadlineCloud-DocumentationAndSamples/JobBundle-Samples
```

2. 檢查工作屬性。

```
deadline job get
```

如下所示的輸出：

```
parameters:
  Message:
    string: Welcome to Amazon Deadline Cloud!
  InFile:
    path: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
JobBundle-Samples/simple_job/template.yaml
  OutFile:
    path: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
JobBundle-Samples/hash-jobattachments.txt
attachments:
  manifests:
  - rootPath: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
JobBundle-Samples
    rootPathFormat: posix
```

```
    outputRelativeDirectories:
    - .
    inputManifestPath: farm-3040c59a5b9943d58052c29d907a645d/queue-
cde9977c9f4d4018a1d85f3e6c1a4e6e/Inputs/
f46af01ca8904cd8b514586671c79303/0d69cd94523ba617c731f29c019d16e8_input.xxh128
    inputManifestHash: f95ef91b5dab1fc1341b75637fe987ee
  fileSystem: COPIED
```

附件欄位包含資訊清單結構清單，這些結構描述工作執行時所使用的輸入和輸出資料路徑。查看rootPath以查看提交工作的機器上的本機目錄路徑。若要查看包含資訊清單檔案的 Amazon S3 物件尾碼，請查看inputManifestFile。資訊清單檔案包含工作輸入資料之目錄樹狀結構快照的中繼資料。

3. 漂亮地列印 Amazon S3 資訊清單物件，以查看任務的輸入目錄結構。

```
MANIFEST_SUFFIX=$(aws deadline get-job \
     --farm-id $DEV_FARM_ID \
     --queue-id $DEV_QUEUE_ID \
     --job-id $JOB_ID \
     --query "attachments.manifests[0].inputManifestPath" \
     --output text)
 aws s3 cp s3://$DEV_FARM_BUCKET/JobAttachments/Manifests/$MANIFEST_SUFFIX - | jq .
```

如下所示的輸出：

```
{
     "hashAlg": "xxh128",
     "manifestVersion": "2023-03-03",
     "paths": [
     {
         "hash": "2ec297b04c59c4741ed97ac8fb83080c",
         "mtime": 1698186190000000,
         "path": "simple_job/template.yaml",
         "size": 445
     }
     ],
     "totalSize": 445
 }
```

4. 建構保留輸出任務附件資訊清單的 Amazon S3 前綴，並在其下列出物件。

```
SESSION_ACTION=$(aws deadline list-session-actions \
```

```
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --job-id $JOB_ID \
    --session-id $SESSION_ID \
    --query "sessionActions[?definition.taskRun != null] | [0]")
STEP_ID=$(echo $SESSION_ACTION | jq -r .definition.taskRun.stepId)
TASK_ID=$(echo $SESSION_ACTION | jq -r .definition.taskRun.taskId)
TASK_OUTPUT_PREFIX=JobAttachments/Manifests/$DEV_FARM_ID/$DEV_QUEUE_ID/$JOB_ID/
$STEP_ID/$TASK_ID/
aws s3api list-objects-v2 --bucket $DEV_FARM_BUCKET --prefix $TASK_OUTPUT_PREFIX
```

輸出任務附件不會直接從任務資源參考，而是根據伺服器陣列資源 ID 放置在 Amazon S3 儲存貯體中。

5. 獲取特定會話操作 ID 的最新清單對象密鑰，然後漂亮地打印清單對象。

```
SESSION_ACTION_ID=$(echo $SESSION_ACTION | jq -r .sessionActionId)
 MANIFEST_KEY=$(aws s3api list-objects-v2 \
     --bucket $DEV_FARM_BUCKET \
     --prefix $TASK_OUTPUT_PREFIX \
     --query "Contents[*].Key" --output text \
     | grep $SESSION_ACTION_ID \
     | sort | tail -1)
 MANIFEST_OBJECT=$(aws s3 cp s3://$DEV_FARM_BUCKET/$MANIFEST_KEY -)
 echo $MANIFEST_OBJECT | jq .
```

您將在輸出中看到文件hash-jobattachments.txt的屬性，如下所示：

```
{
     "hashAlg": "xxh128",
     "manifestVersion": "2023-03-03",
     "paths": [
     {
         "hash": "f60b8e7d0fabf7214ba0b6822e82e08b",
         "mtime": 1698785252554950,
         "path": "hash-jobattachments.txt",
         "size": 182
     }
     ],
     "totalSize": 182
 }
```

每個任務運行時，您的工作只會有一個清單對象，但一般來說，每次任務運行時可能會有更多對象。

6. 在前置詞下檢視內容定址 Amazon S3 儲存輸出。Data

```
FILE_HASH=$(echo $MANIFEST_OBJECT | jq -r .paths[0].hash)
FILE_PATH=$(echo $MANIFEST_OBJECT | jq -r .paths[0].path)
aws s3 cp s3://$DEV_FARM_BUCKET/JobAttachments/Data/$FILE_HASH -
```

如下所示的輸出：

```
eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3  /tmp/openjd/
session-123/assetroot-abc/simple_job/template.yaml
```

步驟 5：將服務受管理的叢集新增至 Develop Cloud 中的開發人員伺服器陣列

AWS CloudShell 無法提供足夠的運算容量來測試較大的工作負載。它也未配置為與在多個 Worker 主機上分發任務的工作搭配使用。

您可以將 Auto Scaling 服務受管叢集 (SMF) 新增至開發人員伺服器陣列 CloudShell，而不必使用。SMF 可為較大的工作負載提供足夠的運算容量，並且可以處理需要在多個 Worker 主機上分配工作任務的工作。除非您關閉 CMF 工作程式，否則排程器會同時使用 SMF 和 CMF 工作程式來執行工作。

將服務管理的叢集新增至開發人員伺服器陣列

1. 安裝和配置 AWS Command Line Interface（AWS CLI），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.
2. 選擇您的第一個 AWS CloudShell 索引標籤，然後建立服務受管理的叢集，並將其叢集 ID 新增至.bashrc。此動作可讓其他終端機工作階段使用。

```
FLEET_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
        --query "Account" --output text):role/${DEV_FARM_NAME}FleetRole"
 aws deadline create-fleet \
     --farm-id $DEV_FARM_ID \
     --display-name "$DEV_FARM_NAME SMF" \
```

```
    --role-arn $FLEET_ROLE_ARN \
    --max-worker-count 5 \
    --configuration \
        '{
            "serviceManagedEc2": {
                "instanceCapabilities": {
                    "vCpuCount": {
                        "min": 2,
                        "max": 4
                    },
                    "memoryMiB": {
                        "min": 512
                    },
                    "osFamily": "linux",
                    "cpuArchitectureType": "x86_64"
                },
                "instanceMarketOptions": {
                    "type": "spot"
                }
            }
        }'

echo "DEV_SMF_ID=$(aws deadline list-fleets \
        --farm-id $DEV_FARM_ID \
        --query "fleets[?displayName=='$DEV_FARM_NAME SMF'].fleetId \
        | [0]" --output text)" >> ~/.bashrc
source ~/.bashrc
```

3. 將 SMF 與您的佇列產生關聯。

```
aws deadline create-queue-fleet-association \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --fleet-id $DEV_SMF_ID
```

4.

> **Note**
>
> 除非您關閉 CMF 工作程式，否則排程器會同時使用 SMF 和 CMF 工作程式來執行工作。

提交simple_file_job到隊列。當系統提示您確認上傳時，請輸入**y**。

```
deadline bundle submit simple_file_job \
    -p InFile=simple_job/template.yaml \
    -p OutFile=hash-jobattachments.txt
```

5. 確認 SMF 是否正常運作。

```
deadline fleet get
```

- 工作人員可能需要幾分鐘才能開始。
- queueFleetAssociationsStatus適用於您客戶管理的車隊和服務管理車隊ACTIVE。
- SMF autoScalingStatus 將從變更GROWING為STEADY。

您的狀態看起來會類似下列內容：

```
fleetId: fleet-2cc78e0dd3f04d1db427e7dc1d51ea44
farmId: farm-63ee8d77cdab4a578b685be8c5561c4a
displayName: DeveloperFarm SMF
description: ''
status: ACTIVE
autoScalingStatus: STEADY
targetWorkerCount: 0
workerCount: 0
minWorkerCount: 0
maxWorkerCount: 5
```

6. 檢視您提交之工作的記錄。此日誌存放在 Amazon CloudWatch 日誌的日誌中，而不是 CloudShell 檔案系統中。

```
JOB_ID=$(deadline config get defaults.job_id)
SESSION_ID=$(aws deadline list-sessions \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --job-id $JOB_ID \
        --query "sessions[0].sessionId" \
        --output text)
aws logs tail /aws/deadline/$DEV_FARM_ID/$DEV_QUEUE_ID \
    --log-stream-names $SESSION_ID
```

步驟 6：在截止日期雲端中清理您的伺服器陣列資源

若要開發和測試新的工作負載和管線整合，您可以繼續使用您為本教學課程建立的 Dependout Cloud 開發人員伺服器陣列。如果您不再需要開發人員伺服器陣列，可以刪除其資源，包括 Amazon Logs 中的伺服器陣列、叢集、佇列、 AWS Identity and Access Management (IAM) 角色和 CloudWatch 日誌。刪除這些資源後，您將需要再次開始教學課程，才能使用這些資源。如需詳細資訊，請參閱 為期限雲端設定開發人員工作站。

清除開發人員伺服器陣列資源

1. 安裝和配置 AWS Command Line Interface（AWS CLI），如果你還沒有。如需相關資訊，請參閱安裝或更新至最新版本的 AWS CLI.
2. 選擇您的第一個 CloudShell 索引標籤，然後停止佇列的所有佇列-叢集關聯。

```
FLEETS=$(aws deadline list-queue-fleet-associations \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --query "queueFleetAssociations[].fleetId" \
        --output text)
for FLEET_ID in $FLEETS; do
    aws deadline update-queue-fleet-association \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --fleet-id $FLEET_ID \
        --status STOP_SCHEDULING_AND_CANCEL_TASKS
done
```

3. 列出佇列叢集關聯。

```
aws deadline list-queue-fleet-associations \
     --farm-id $DEV_FARM_ID \
     --queue-id $DEV_QUEUE_ID
```

您可能需要重新執行命令，直到輸出報告為止"status": "STOPPED"，然後才能繼續執行下一個步驟。此程序可能需要幾分鐘的時間才能完成。

```
{
    "queueFleetAssociations": [
        {
            "queueId": "queue-abcdefgh01234567890123456789012id",
            "fleetId": "fleet-abcdefgh01234567890123456789012id",
```

```
            "status": "STOPPED",
            "createdAt": "2023-11-21T20:49:19+00:00",
            "createdBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName",
            "updatedAt": "2023-11-21T20:49:38+00:00",
            "updatedBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName"
        },
        {
            "queueId": "queue-abcdefgh01234567890123456789012id",
            "fleetId": "fleet-abcdefgh01234567890123456789012id",
            "status": "STOPPED",
            "createdAt": "2023-11-21T20:32:06+00:00",
            "createdBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName",
            "updatedAt": "2023-11-21T20:49:39+00:00",
            "updatedBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName"
        }
    ]
}
```

4. 刪除佇列的所有佇列-叢集關聯。

```
for FLEET_ID in $FLEETS; do
     aws deadline delete-queue-fleet-association \
         --farm-id $DEV_FARM_ID \
         --queue-id $DEV_QUEUE_ID \
         --fleet-id $FLEET_ID
 done
```

5. 刪除與佇列相關聯的所有叢集。

```
for FLEET_ID in $FLEETS; do
     aws deadline delete-fleet \
         --farm-id $DEV_FARM_ID \
         --fleet-id $FLEET_ID
 done
```

6. 刪除佇列。

```
aws deadline delete-queue \
     --farm-id $DEV_FARM_ID \
```

```
    --queue-id $DEV_QUEUE_ID
```

7. 刪除伺服器陣列。

```
aws deadline delete-farm \
    --farm-id $DEV_FARM_ID
```

8. 刪除伺服器陣列的其他 AWS 資源。

   a. 刪除叢集 AWS Identity and Access Management (IAM) 角色。

```
aws iam delete-role-policy \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --policy-name WorkerPermissions
aws iam delete-role \
    --role-name "${DEV_FARM_NAME}FleetRole"
```

   b. 刪除佇列 IAM 角色。

```
aws iam delete-role-policy \
    --role-name "${DEV_FARM_NAME}QueueRole" \
    --policy-name S3BucketsAccess
aws iam delete-role \
    --role-name "${DEV_FARM_NAME}QueueRole"
```

   c. 刪除 Amazon CloudWatch 日誌日誌群組。每個佇列和叢集都有自己的記錄群組。

```
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_QUEUE_ID"
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_CMF_ID"
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_SMF_ID"
```

設定截止日期雲端提交者

此程序適用於想要安裝、設定和啟動 AWS 截止日期雲端提交者的管理員和藝術家。截止日期雲提交者是一個數字內容創建（DCC）插件。藝術家使用它來從他們熟悉的第三方 DCC 介面提交工作。

> **Note**
>
> 此程序必須在藝術家將用於提交彩現的所有工作站上完成。

主題

步驟 1：安裝截止日期雲端提交者

下列各節會引導您完成安裝期限雲端提交者的步驟。

下載提交者安裝程式

您必須先下載提交者安裝程式，才能安裝截止日期雲端提交者。目前，截止日期雲端提交者安裝程式僅支援Windows和。Linux

1. 登入 AWS Management Console 並開啟截止日期雲端主控台。
2. 在側邊導覽窗格中，選擇 [下載]。
3. 找到截止日期雲端提交者安裝程式區段。
4. 選取電腦作業系統的安裝程式，然後選擇 [下載]。

(選擇性) 驗證下載軟體的真偽

若要確認您下載的軟體是否正版，請針對Windows或使用下列程序Linux。

> **Note**
>
> 您可以使用這些指示先驗證安裝程式，然後在下一節 (步驟 2) 下載截止日期雲端監視器之後驗證。

Windows

若要驗證下載檔案的真實性，請完成以下步驟。

1. 在下列命令中，*file*以您要驗證的檔案取代。例如 ***C:\PATH\TO\MY\DeadlineCloudSubmitter-windows-x64-installer.exe*** 。另外，請*signtool-sdk-version*替換為已安裝的 SignTool SDK 版本。例如 **10.0.22000.0**。

   ```
   "C:\Program Files (x86)\Windows Kits\10\bin\signtool-sdk-version\x86\signtool.exe" verify /vfile
   ```

2. 例如，您可以執行下列命令，以驗證截止日期雲端提交者安裝程式檔案：

   ```
   "C:\Program Files (x86)\Windows Kits\10\bin
   \10.0.22000.0\x86\signtool.exe" verify /v DeadlineCloudSubmitter-windows-x64-installer.exe
   ```

Linux

若要驗證下載檔案的真實性，請使用gpg命令列工具。

1. 執行下列命令，匯入截止日期雲端提交者安裝程式的OpenPGP金鑰：

```
 gpg --import --armor <<EOF
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGX6GQsBEADduUtJgqSXI+q76O6fsFwEYKmbnlyL0xKvlq32EZuyv0otZo5L
le4m5Gg52AzrvPvDiUTLooAlvYeozaYyirIGsK08Ydz0Ftdjroiuh/mw9JSJDJRI
rnRn5yKet1JFezkjopA3pjsTBP6lW/mb1bDBDEwwwtH0x9lV7A03FJ9T7Uzu/qSh
qO/UYdkafro3cPASvkqgDt2tCvURfBcUCAjZVFcLZcVD5iwXacxvKsxxS/e7kuVV
I1+VGT8Hj8XzWYhjCZxOLZk/fvpYPMyEEujN0fYUp6RtMIXve0C9awwMCy5nBG2J
eE2Ol5DsCpTaBd4Fdr3LWcSs8JFA/YfP9auL3NczOozPoVJt+fw8CBlVIXO0J7l5
hvHDjcC+5v0wxqAlMG6+f/SX7CT8FXK+L3iOJ5gBYUNXqHSxUdv8kt76/KVmQa1B
Akl+MPKpMq+lhw++S3G/lXqwWaDNQbRRw7dSZHymQVXvPp1nsqc3hV7KlOM+6s6g
1g4mvFY4lf6DhptwZLWyQXU8rBQpojvQfiSmDFrFPWFi5BexesuVnkGIolQoklKx
AVUSdJPVEJCteyy7td4FPhBaSqT5vW3+ANbr9b/uoRYWJvn17dN0cc9HuRh/Ai+I
nkfECo2WUDLZ0fEKGjGyFX+todWvJXjvc5kmE9Ty5vJp+M9Vvb8jd6t+mwARAQAB
tCxBV1MgRGVhZGxpbmUgQ2xvdWQgPGF3cy1kZWFkbGluZUBhbWF6b24uY29tPokC
VwQTAQgAQRYhBLhAwIwpqQeWoHH6pfbNPOa3bzzvBQJl+hkLAxsvBAUJA8JnAAUL
CQgHAgIiAgYVCgkICwIDFgIBAh4HAheAAAoJEPbNPOa3bzzvKswQAJXzKSAY8sY8
F6Eas2oYwIDDdDurs8FiEnFghjUEO6MTt9AykF/jw+CQg2UzFtEyObHBymhgmhXE
3buVeom96tgM3ZDfZu+sxi5pGX6oAQnZ6riztN+VpkpQmLgwtMGpSMLl3KLwnv2k
WK8mrR/fPMkfdaewB7A6RIUYiW33GAL4KfMIs8/vIwIJw99NxHpZQVoU6dFpuDtE
1OuxGcCqGJ7mAmo6H/YawSNp2Ns80gyqIKYo7o3LJ+WRroIRlQyctq8gnR9JvYXX
42ASqLq5+OXKo4qh81blXKYqtc176BbbSNFjWnzIQgKDgNiHFZCdcOVgqDhwO15r
NICbqqwwNLj/Fr2kecYx180Ktpl0jOOw5IOyh3bf3MVGWnYRdjvA1v+/CO+55N4g
```

```
z0kf50Lcdu5RtqV10XBCifn28pecqPaSdYcssYSRl5DLiFktGbNzTGcZZwITTKQc
af8PPdTGtnnb6P+cdbW3bt9MVtN5/dgSHLThnS8MPEuNCtkTnpXshuVuBGgwBMdb
qUC+HjqvhZzbwns8dr5WI+6HWNBFgGANn6ageYl58vVp0UkuNP8wcWjRARciHXZx
ku6W2jPTHDWGNrBQO2Fx7fd2QYJheIPPAShHcfJO+xgWCof45D0vAxAJ8gGg9Eq+
gFWhsx4NSHn2gh1gDZ41Ou/4exJ1lwPM
=uVaX
-----END PGP PUBLIC KEY BLOCK-----
EOF
```

2. 判斷是否信任0penPGP金鑰。決定是否信任上述鍵時需要考慮的因素包括以下幾點：

   - 您用來從本網站取得 GPG 金鑰的網際網路連線是安全的。
   - 您訪問本網站的設備是安全的。
   - AWS 已採取措施確保在本網站上託管0penPGP公鑰的安全。

3. 如果您決定信任OpenPGP金鑰，請使用gpg類似下列範例來編輯金鑰以信任的金鑰：

```
$ gpg --edit-key 0xB840C08C29A90796A071FAA5F6CD3CE6B76F3CEF

    gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.


    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud example@example.com

    gpg> trust
    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com

    Please decide how far you trust this user to correctly verify other users'
 keys
    (by looking at passports, checking fingerprints from different sources,
 etc.)

      1 = I don't know or won't say
      2 = I do NOT trust
      3 = I trust marginally
      4 = I trust fully
      5 = I trust ultimately
```

```
  m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                     trust: ultimate      validity: unknown
[ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com
Please note that the shown key validity is not necessarily correct
unless you restart the program.

gpg> quit
```

4. 驗證安裝

   若要驗證安裝程式，請完成以下步驟：

   a. 返回截止日期雲端主控台下載頁面，並下載截止日期雲端提交者安裝程式的簽章檔案。

   b. 執行下列指令，驗證截止日期雲端提交者安裝程式的簽章：

   ```
   gpg --verify ./DeadlineCloudSubmitter-linux-x64-
   installer.run.sig ./DeadlineCloudSubmitter-linux-x64-
   installer.run
   ```

5. 驗證截止日期雲監控

   > **Note**
   >
   > 您可以使用簽名檔案或平台特定方法來驗證截止日期雲端監視器的下載。有關平台特定方法，請根據您下載的文件類型查看Linux (AppImage)選項卡或選項卡。Linux (DEB)

   若要使用簽章檔案驗證截止日期雲端監視器桌面應用程式，請完成以下步驟：

   a. 返回截止日期雲端主控台下載頁面並下載對應的 .sig 檔案，然後執行

      對於 .deb：

      ```
      gpg --verify ./deadline-cloud-
      monitor_<APP_VERSION>_amd64.deb.sig ./deadline-cloud-
      monitor_<APP_VERSION>_amd64.deb
      ```

對於. AppImage:

```
gpg --verify ./deadline-cloud-
monitor_<APP_VERSION>_amd64.AppImage.sig ./deadline-cloud-
monitor_<APP_VERSION>_amd64.AppImage
```

b. 確認輸出看起來類似下列內容：

```
gpg: Signature made Mon Apr 1 21:10:14 2024 UTC

gpg: using RSA key B840C08C29A90796A071FAA5F6CD3CE6B7
```

如果輸出包含片語`Good signature from "AWS Deadline Cloud"`，表示簽章已成功驗證，您可以執行 Dependpoint Cloud 監視器安裝指令碼。

Linux (DEB)

若要驗證使用 Linux .deb 二進位檔的套件，請先完成索引標籤中的步驟 1-3。Linux

dpkg 是大多數debian基Linux礎發行版中的核心套件管理工具。您可以使用工具驗證 .deb 檔案。

1. 從「截止日期雲端主控台下載」頁面，下載截止日期雲端監控 .deb 檔案。
2. 以您要驗證的 .deb 檔案版本取***<APP_VERSION>***代。

   ```
   dpkg-sig --verify deadline-cloud-monitor_<APP_VERSION>_amd64.deb
   ```
3. 輸出將類似於：

   ```
   Processing deadline-cloud-monitor_1.1.1_amd64.deb... GOODSIG
   _gpgbuilder B840C08C29A90796A071FAA5F6CD3C 171200
   ```
4. 若要驗證 .deb 檔案，請確認輸出中存`GOODSIG`在該檔案。

Linux (AppImage)

若要驗證使用Linux. AppImage 二進位，請先完成Linux索引標籤中的步驟 1-3。

1. 從「截止日期雲端主控台下載」頁面，下載截止日期雲端監控。 AppImage 文件。
2. 若要取代為<APP_VERSION>的版本。 AppImage 您要驗證的檔案，請完成下列步驟：

   a. 從寫入簽名。 AppImage 文件到一個 .sig 文件中。

```
./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
--appimage-signature > ./deadline-cloud-
monitor_<APP_VERSION>_amd64_.AppImage.sig
```

b. 使用生成的 .sig 文件使用以下命令進行驗證。

```
gpg --verify ./deadline-cloud-
monitor_<APP_VERSION>_amd64.AppImage.sig
```

c. (選擇性) 如果顯示「拒絕權限」錯誤，請使用下列命令新增執行權限。

```
chmod +x ./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
```

d. 確認輸出看起來類似下列內容：

```
gpg: Signature made Mon Apr 1 21:10:14 2024 UTC

gpg: using RSA key B840C08C29A90796A071FAA5F6CD3CE6B7
```

如果輸出包含片語`Good signature from "AWS Deadline Cloud"`，表示簽章已成功驗證，您可以執行 Dependpoint Cloud 監視器安裝指令碼。

安裝截止日期雲端提交者

您可以使Windows用或安裝截止日期雲端提交者。Linux使用安裝程式，您可以安裝下列提交者：

- 瑪雅
- 核子彈
- 胡迪尼 19.5
- 按鍵快照 12
- 攪拌機 3.6
- 虛幻引擎 5

Windows

1. 在檔案瀏覽器中，導覽至安裝程式下載的資料夾，然後選取`DeadlineCloudSubmitter-windows-x64-installer.exe`。

   a. 如果顯示 Windows 保護您的電腦彈出式視窗，請選擇 [更多資訊]。

b. 仍然選擇「運行」。

2. AWS 截止日期「雲端提交者安裝精靈」開啟後，選擇「下一步」。
3. 完成下列其中一個步驟來選擇安裝範圍：
   - 若只要為目前使用者安裝，請選擇 [使用者]。
   - 若要為所有使用者安裝，請選擇 [系統]。

     如果您選擇「系統」，則必須結束安裝程式，然後以系統管理員身分重新執行，方法是完成下列步驟：

     a. 右鍵單擊**DeadlineCloudSubmitter-windows-x64-installer.exe**，然後選擇以管理員身份運行。

     b. 輸入您的系統管理員認證，然後選擇 [是]。

     c. 選擇 [系統] 做為安裝範圍。
4. 選取安裝範圍之後，請選擇 [下一步]。
5. 再次選擇「下一步」以接受安裝目錄。
6. 選取整合式提交者 Nuke，或您要安裝的提交者。
7. 選擇下一步。
8. 檢閱安裝，然後選擇 [下一步]。
9. 再次選擇 [下一步]，然後選擇 [完成]。

Linux

> **Note**
>
> 截止日期雲端整合安裝Nuke程式Linux和截止日期雲端監視器只能安裝在至少具有 GLIBC 2.31 的Linux發行版上。

1. 開啟終端機視窗。
2. 若要執行安裝程式的系統安裝，請輸入指令，**sudo -i**然後按 Enter 以成為 root。
3. 導覽至您下載安裝程式的位置。

   例如 **cd /home/*USER*/Downloads**。

4. 若要使安裝程式可執行，請輸入**chmod +x DeadlineCloudSubmitter-linux-x64-installer.run**。
5. 若要執行期限雲端提交者安裝程式，請輸入。**./DeadlineCloudSubmitter-linux-x64-installer.run**
6. 安裝程式開啟時，請依照畫面上的提示完成安裝精靈。

您可以安裝此處未列出的其他提交者。我們使用截止日期雲庫來構建提交者。您可以在 aw GitHub s-deadline 組織中找到這些庫和提交者的源代碼。

步驟 2：安裝和設置截止日期雲監視器

您可以使用或安裝截止日期雲監視器桌面應用Windows程序Linux。

Windows

1. 如果您尚未登入，請登入 AWS Management Console 並開啟「截止日期雲端主控台」。
2. 在左側導覽窗格中，選擇 [下載]。
3. 在「截止日期雲端監控」區段中，選取電腦作業系統適用的檔案。
4. 若要下載截止日期雲端監視器，請選擇下載。

Linux

若要 AppImage 在 RPM 發行版上安裝期限雲端監視器

1. 下載最新的截止日期雲監視器 AppImage。
2. 若要製作 AppImage 可執行檔，請輸入**chmod a+x deadline-cloud-monitor_*<APP_VERSION>*_amd64.AppImage**。
3. 若要設定正確的 SSL 憑證路徑，請輸入**sudo ln -sf /etc/ssl/certs/ca-bundle.crt /etc/ssl/certs/ca-certificates.crt**。

AppImage 在 Debian 發行版上安裝期限雲監視器

1. 下載最新的截止日期雲監視器 AppImage。

2.

> **Note**
>
> 此步驟適用於 Ubuntu 22 及以上版本。對於其他版本的 Ubuntu，請跳過此步驟。

若要安裝資料庫 2，請輸入 **sudo apt udpate**

sudo apt install libfuse2.

3. 若要製作 AppImage 可執行檔，請輸入**chmod a+x deadline-cloud-monitor_*<APP_VERSION>*_amd64.AppImage**。

在 Debian 發行版上安裝期限雲監控 Debian 軟件包

1. 下載最新的截止日期雲監控 Debian 軟件包。
2.

> **Note**
>
> 此步驟適用於 Ubuntu 22 及以上版本。對於其他版本的 Ubuntu，請跳過此步驟。

若要安裝 Libssl1.1，請輸入 **wget http://nz2.archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.*<APP_VERSION>*.1f-1ubuntu2.22_amd64.deb**

sudo dpkg -i libssl1.*<APP_VERSION>*.1f-1ubuntu2.22_amd64.deb.

3. 要安裝期限雲監視器 Debian 軟件包，請輸入 **sudo apt update**

sudo apt install ./deadline-cloud-monitor_*<APP_VERSION>*_amd64.deb.

4. 如果在具有未滿足相依性的套件上安裝失敗，請修正損毀的套件，然後執行下列命令。

sudo apt --fix-missing update

sudo apt update

sudo apt install -f

完成下載後，您可以驗證下載軟件的真實性。請參閱步驟 1 中的驗證下載軟體的真實性。

下載截止日期雲監視器並驗證真實性後，請使用以下步驟設置截止日期雲監視器。

設定截止日期雲端監控

1. 打開截止日期雲監視器。
2. 當系統提示您建立新設定檔時，請完成以下步驟。

   a. 在 URL 輸入中輸入您的監視器 URL，看起來像 **https://*MY-MONITOR*.deadlinecloud.amazonaws.com/**

   b. 輸入設定檔名稱。

   c. 選擇建立設定檔。

   您的設定檔已建立，而且您的認證現在會與使用您建立的設定檔名稱的任何軟體共用。
3. 建立截止日期雲端監控設定檔後，您就無法變更設定檔名稱或工作室 URL。如果您需要進行變更，請改為執行下列動作：

   a. 刪除設定檔。在左側導覽窗格中，選擇「期限雲端監控」、「設定」、「刪除」。

   b. 使用您想要的變更建立新的設定檔。
4. 在左側導覽窗格中，使用 > 截止日期雲端監視器選項執行下列作業：

   - 變更截止日期雲端監控設定檔以登入不同的監視器。
   - 啟用自動登入，這樣您就不必在後續開啟期限雲端監視器時輸入監視器 URL。
5. 關閉截止日期雲端監視器視窗。它會繼續在後台運行，並每隔 15 分鐘同步您的憑據。
6. 針對您計劃用於彩現專案的每個數位內容建立 (DCC) 應用程式，完成以下步驟：

   a. 在截止日期雲端提交者中，開啟截止日期雲端工作站設定。

   b. 在工作站組態中，選取您在「截止日期雲端監視器」中建立的設定檔。您的截止日期雲端認證現在會與此 DCC 共用，您的工具應如預期般運作。

步驟 3：啟動截止日期雲端提交者

下列各節將引導您完成在Blender、Nuke、和中啟動截止日期雲端提交者外掛程式的步驟。Maya Houdini

若要啟動截止日期雲端提交者 Blender

> **Note**
>
> 使用服務管理叢集的Conda環境提供 Support 援。Blender如需詳細資訊，請參閱 預設Conda佇列環境。

1. 打開 Blender.
2. 打開具有資產根目錄中存在的依賴關係的Blender場景。
3. 在「彩現」功能表中，選取「期限雲」對話方塊。
   a. 如果您尚未在截止日期雲端提交者中進行驗證，「身份證明狀態」會顯示 NEEDS_LOGIN。
   b. 選擇 Login (登入)。
   c. 登入瀏覽器視窗隨即顯示。使用您的使用者認證登入。
   d. 選擇 Allow (允許)。您現在已登入，「身份證明狀態」 會顯示為「已驗證」。
4. 選擇提交。

若要啟動截止日期雲端提交者 Foundry Nuke

> **Note**
>
> 使用服務管理叢集的Conda環境提供 Support 援。Nuke如需詳細資訊，請參閱 預設Conda佇列環境。

1. 打開 Nuke.
2. 打開具有資產根目錄中存在依賴關係的Nuke腳本。
3. 選擇 Thinkbox，然後選擇提交到期限雲端以啟動提交者。
   a. 如果您尚未在截止日期雲端提交者中進行驗證，身份證明狀態將顯示為 NEEDS_LOGIN。
   b. 選擇 Login (登入)。
   c. 在登入瀏覽器視窗中，使用您的使用者認證登入。
   d. 選擇 Allow (允許)。您現在已登入，「身份證明狀態」 會顯示為「已驗證」。
4. 選擇提交。

若要啟動截止日期雲端提交者 Maya

> **Note**
>
> 使用服務管理叢集的Conda環境提供 Support 援Maya和Arnold for Maya(MtoA)提供。如需詳細資訊，請參閱 預設Conda佇列環境。

1. 打開 Maya.
2. 設置您的項目，並打開資產根目錄中存在的文件。
3. 選擇視窗 → 設置/首選項 → 插件管理器。
4. 搜尋 DeadlineCloud「提交者」。
5. 若要載入截止日期雲端提交者外掛程式，請選取已載入。
    a. 如果您尚未在截止日期雲端提交者中進行驗證，身份證明狀態將顯示為 NEEDS_LOGIN。
    b. 選擇 Login (登入)。
    c. 登入瀏覽器視窗隨即顯示。使用您的使用者認證登入。
    d. 選擇 Allow (允許)。您現在已登入，「身份證明狀態」會顯示為「已驗證」。
6. (選擇性) 若要在每次開啟時載入截止日期雲端提交者外掛程式Maya，請選擇「自動載入」。
7. 選取截止日期雲端架，然後選取綠色按鈕以啟動提交者。

若要啟動截止日期雲端提交者 Houdini

> **Note**
>
> 使用服務管理叢集的Conda環境提供 Support 援。Houdini如需詳細資訊，請參閱 預設Conda佇列環境。

1. 打開 Houdini.
2. 在「網路編輯器」中，選取 /out 網路。
3. 按 Tab 鍵，然後輸入**`deadline`**。
4. 選取「截止日期雲端」選項，並將其連線至您現有的網路。
5. 按兩下「期限雲端」節點。

若要啟動截止日期雲端提交者 KeyShot

這假設您已經下載了截止日期雲和 PySide 2。

1. 將檔案複製或連結至 AWS 截止日期 Cloud.py 到指令碼至指令碼資料夾的無效雲端/金鑰擷取指令碼指令碼/指令碼。 KeyShot

   例如，在上Windows，腳本文件夾的位置將是**C:/Users/*USER*/Documents/KeyShot 12/Scripts**。

2. 設定下列環境變數。

   a. 將環境變量設置**DEADLINE_PYTHON**為 Python 安裝路徑，其中死線雲和 PySide 2 所在的位置。

      例如，在上Windows，如果使用 Python 3.10，則該命令可能是**set DEADLINE_PYTHON=C:/Users/*USER*/AppData/Local/Programs/Python/Python310/python**。

   b. 將環境變數設定**DEADLINE_KEYSHOT**為 keyshot_ 提交程式資料夾的路徑。

      例如，在上Windows，如果源在您的桌面上，則命令可能是**set DEADLINE_KEYSHOT=C:/Users/*USER*/Desktop/deadline-cloud-for-keyshot/src/deadline/keyshot_submitter**。

3. 設定環境變數後，啟動KeyShot。
4. 若要從中啟動提交者，請選擇「編寫指令碼主控台」 KeyShot Windows、「提交至 AWS 截止日期雲端」和「執行」。

若要啟動截止日期雲端提交者 Unreal Engine

這假設您已下載截止日期雲端。

1. 建立或開啟您用於Unreal Engine專案的資料夾。
2. 開啟命令列並執行下列命令：

   - `git clone https://github.com/aws-deadline/deadline-cloud-for-unreal-engine`
   - `cd deadline-cloud-for-unreal/test_projects`
   - `git lfs fetch –all`

3. 若要下載的外掛程式Unreal Engine，請開啟Unreal Engine專案資料夾，然後啟動 deadline-cloud-forunreal/test_projects/pull_ue_plugin.bat。

   這將插件文件放在 C:/LocalProjects/UnrealDeadlineCloudTest/Plugins/UnrealDeadlineCloudService中。

4. 若要下載提交者，請開啟 UnrealDeadlineCloudService 資料夾，然後執行。**deadline-cloud-forunreal/ test_projects/Plugins/UnrealDeadlineCloudService/install_unreal_submitter.bat**

5. 若要從啟動提交者Unreal Engine，請完成以下步驟：

   a. 選擇 「編輯」 >「專案設定」。

   b. 在搜尋列中，輸入 **movie render pipeline**。

   c. 調整下列「影片演算管線」設定：

      i. 對於「預設遠端執行程式」，請輸入**MoviePipelineDeadlineCloudRemoteExecutor**。

      ii. 對於預設執行程式 Job，請輸入 **MoviePipelineDeadlineCloudExecutorJob**

      iii. 對於「預設 Job 設定類別」，請選擇加號，然後輸入**DeadlineCloudRenderStepSetting**。

   使用這些設定，您可以從中選擇截止日期雲外掛程式Unreal Engine。

使用伺服器陣列

如果您已遵循所有入門指示，則已設定所需的一切，以便開始將工作從本機工作站提交至伺服器陣列，然後監視這些作業和資源。如需有關提交各種工作或監視的詳細資訊，請參閱下面的相關主題。

- 任務
- 使用監視器

使用截止日期雲端監視器

AWS 截止日期雲端監視器為您提供視覺化運算工作的整體檢視。您可以使用它來監視和管理工作、檢視叢集上的工作者活動、追蹤預算和使用情況，以及下載工作結果。

每個佇列都有一個工作監視器，可顯示工作、步驟和工作的狀態。監視器提供直接從監視器管理工作的方法。您可以進行優先順序變更、取消工作和重新查詢工作。

截止日期 Cloud 監視器有一個顯示工作摘要狀態的表格，您也可以選取工作來查看詳細的工作記錄，協助疑難排解工作的問題。

您可以使用「截止日期雲端」監視器，將結果下載到工作建立時指定的工作站上的位置。

截止日期雲端監視器也可協助您監控使用情況並管理成本。如需詳細資訊，請參閱 管理截止日期雲端的預算和用量。

主題

共用截止日期雲端監控器 URL

設定期限雲端服務時，依預設會建立一個 URL，以開啟帳戶的截止日期雲端監視器。使用此 URL 在瀏覽器或桌面上打開顯示器。與其他使用者共用 URL，以便他們可以存取截止日期雲端監視器。

您必須先授與使用者存取權，才能開啟截止日期雲端監視器。若要授與存取權，請將使用者新增至監視器的授權使用者清單，或將他們新增至具有監視存取權的群組。如需詳細資訊，請參閱 管理期限雲端中的使用者。

共用監視器 URL

1. 開啟截止日期雲端主控台。
2. 從 [開始使用] 中，選擇 [移至期限雲端儀表板]。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在「帳戶概覽」區段中，選擇「帳戶詳細資料」
5. 複製 URL，然後安全地將 URL 傳送給需要存取截止日期雲端監視器的任何人。

開啟截止日期雲端監視器

您可以透過下列任一方式開啟截止日期雲端監視器：

- 主控台 — 登入 AWS Management Console 並開啟截止日期雲端主控台。
- 網頁 — 移至您在設定期限雲端時建立的監視器 URL。
- 監視器 — 使用桌面截止日期雲監視器。

使用主控台時，您必須能夠 AWS 使用 AWS Identity and Access Management 身分登入，然後使用 AWS IAM Identity Center 認證登入監視器。如果您只有 IAM 身分中心登入資料，則必須使用監控 URL 或桌面應用程式登入。

開啟截止日期雲端監視器 (網頁)

1. 使用瀏覽器開啟您在設定期限雲端時建立的監視器 URL。
2. 使用您的使用者認證登入。

開啟截止日期雲端監視器 (主控台)

1. 開啟截止日期雲端主控台。
2. 在導覽窗格中，選取伺服器陣列。
3. 選取伺服器陣列，然後選擇 [管理工作] 以開啟截止日期雲端監視頁面。
4. 使用您的使用者認證登入。

開啟截止日期雲端監視器 (桌面)

1. 開啟截止日期雲端主控台。

-或-

從監視器 URL 開啟截止日期雲端監視器-網頁。

2. • 在截止日期雲端主控台上，執行下列動作：
   1. 在監視器中，選擇 [前往截止日期雲端儀表板]，然後從左側功能表選擇 [下載]。
   2. 在截止日期雲端監控中，選擇桌面的監視器版本。
   3. 選擇 Download (下載)。
   - 在截止日期雲監視器-Web 上，執行以下操作：
     - 從左側功能表中選擇「工作站設定」。如果看不到「工作站」設定項目，請使用箭頭開啟左側功能表。
     - 選擇 Download (下載)。
     - 從選取作業系統中，選擇您的作業系統。
3. 下載截止日期雲監視器-桌面。
4. 下載並安裝顯示器後，請在計算機上打開它。

   - 如果這是您第一次開啟截止日期雲端監視器，您必須提供監視器 URL 並建立設定檔名稱。接下來，您使用截止日期雲端認證登入監視器。
   - 建立設定檔之後，您可以選取設定檔來開啟監視器。您可能需要輸入截止日期雲端認證。

在期限雲端中檢視佇列和車隊詳細資料

您可以使用截止日期雲端監視器來檢視伺服器陣列中佇列和叢集的組態。您也可以使用監視器來查看佇列中的工作清單或叢集中的 Worker。

您必須擁有檢視佇列和叢集詳細資料的VIEWING權限。如果未顯示詳細資料，請聯絡您的系統管理員以取得正確的權限。

檢視佇列詳細資訊

1. 開啟截止日期雲端監視器.
2. 從伺服器陣列清單中，選擇包含您感興趣之佇列的伺服器陣列。
3. 在佇列清單中，選擇要顯示其詳細資訊的佇列。若要比較兩個或多個佇列的組態，請選取一個以上的核取方塊。
4. 若要查看佇列中的工作清單，請從佇列清單或詳細資料面板中選擇佇列名稱。

如果監視器已開啟，您可以從左側導覽窗格的 [佇列] 清單中選取佇列。

檢視機群詳細資訊

1. 開啟截止日期雲端監視器.
2. 從伺服器陣列清單中，選擇包含您感興趣之叢集的伺服器陣列。
3. 在伺服器陣列資源中，選擇 [叢集]。
4. 在艦隊清單中，選擇要顯示其詳細資料的機隊。若要比較兩個或多個叢集的組態，請選取一個以上的核取方塊。
5. 若要查看叢集中的工作者清單，請從艦隊清單或詳細資料面板中選擇車隊名稱。

如果監視器已開啟，您可以從左側導覽窗格的 [叢集] 清單中選取車隊。

在截止日期雲端中檢視和管理工作、步驟和工作

當您選取佇列時，「截止日期雲端」監視器的「工作監視器」區段會顯示該佇列中的工作、工作中的步驟，以及每個步驟中的工作。當您選取工作、步驟或工作時，可以使用「動作」功能表來管理每個工作、步驟或工作。

若要開啟工作監視器，請按照步驟檢視其中的佇列在期限雲端中檢視佇列和車隊詳細資料，然後選取要處理的工作、步驟或工作。

對於工作、步驟和工作，您可以執行下列動作：

- 將狀態變更為「重新啟動」、「成功」、「失敗」或「已取消」。
- 從工作、步驟或工作下載已處理的輸出。
- 複製工作、步驟或工作的 ID。

對於選取的工作，您可以：

- 封存工作。
- 修改工作屬性，例如變更優先順序或檢視步驟與步驟相依性。
- 使用工作的參數檢視其他詳細資訊。

如需詳細資訊，請參閱 在截止日期雲中查看工作詳細。

對於每個步驟，您可以：

- 檢視步驟的相依性。步驟的相依性必須在執行步驟之前完成。

如需詳細資訊，請參閱 檢視截止日期雲端中的步驟。

對於每個任務，您可以：

- 檢視工作的記錄。
- 檢視工作參數。

如需詳細資訊，請參閱 在截止日期雲端中檢視工作。

在截止日期雲中查看工作詳細

截止日期雲端監視器中的「Job 監視器」頁面提供下列資訊：

- 工作進度的整體檢視。
- 構成工作之步驟和工作的檢視。

從清單中選擇工作以檢視工作的步驟清單，然後從步驟清單中選擇步驟以檢視工作的工作。選擇項目後，您可以使用該項目的「動作」功能表來檢視詳細資訊。

檢視工作詳細資訊

1. 按照步驟檢視中的佇列在期限雲端中檢視佇列和車隊詳細資料。
2. 在瀏覽窗格中，選取您提交工作的佇列。
3. 使用下列其中一種方法選取工作：

    a. 從「工作」清單中，選取要檢視其詳細資訊的工作。
    b. 在搜尋欄位中，輸入與工作相關聯的任何文字，例如建立工作的工作名稱或使用者。從顯示的結果中，選取您要檢視的工作。

工作的詳細資訊包括工作中的步驟以及每個步驟中的工作。您可以使用 「動作」功能表執行下列作業：

- 變更工作的狀態。

- 檢視和修改工作的屬性。您可以檢視工作中步驟之間的相依性，以及變更工作的優先順序。一般而言，優先順序較高的工作會更快完成。
- 檢視送出工作時所設定之工作的參數。
- 下載工作的輸出。當您下載工作的輸出時，它會包含工作中的步驟和工作所產生的所有輸出。

檢視截止日期雲端中的步驟

使用 AWS 截止日期雲端監視器來檢視處理工作中的步驟。在「Job 監視器」中，「步驟」清單會顯示組成所選工作的步驟清單。當您選取步驟時，「工作」清單會顯示步驟中的工作。

若要檢視步驟

1. 請按照中在截止日期雲中查看工作詳細的步驟檢視工作清單。
2. 從 Jobs (任務) 清單中選擇一項任務。
3. 從「步驟」清單中選取一個步驟。

您可以使用 「動作」功能表執行下列作業：

- 變更步驟的狀態。
- 下載步驟的輸出。當您下載步驟的輸出時，它會包含步驟中工作所產生的所有輸出。
- 檢視步驟的相依性。相依性表格會顯示所選步驟開始前必須完成的步驟清單，以及等待此步驟完成的步驟清單。

在截止日期雲端中檢視工作

使用 AWS 截止日期雲端監視器來檢視處理工作中的工作。在「Job 監視器」中，「工作」清單會顯示構成「步驟」清單中所選步驟的工作。

若要檢視工作

1. 請按照中在截止日期雲中查看工作詳細的步驟檢視工作清單。
2. 從 Jobs (任務) 清單中選擇一項任務。
3. 從「步驟」清單中選取一個步驟。
4. 從 [工作] 清單中選取工作。

您可以使用 「動作」功能表執行下列作業：

- 變更工作的狀態。
- 檢視工作記錄。如需詳細資訊，請參閱 在截止日期雲中查看日誌。
- 檢視建立工作時所設定的參數。
- 下載任務的輸出。當您下載工作的輸出時，它只會包含所選工作所產生的輸出。

在截止日期雲中查看日誌

記錄檔為您提供有關工作狀態和處理的詳細資訊。在 AWS 截止日期雲端監控中，您可以看到下列兩種類型的記錄檔：

- 工作階段記錄會詳細說明動作的時間表，包括：
  - 設定動作，例如附件同步處理和載入軟體環境
  - 執行工作或一組工作
  - 關閉動作，例如關閉 Worker 上的環境

  會話包括至少一個任務的處理，並且可以包括多個任務。工作階段日誌也會顯示 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體類型、vCPU 和記憶體的相關資訊。工作階段記錄也包含工作階段中使用之 Worker 的記錄連結。
- Worker 記錄會提供 Worker 在其生命週期內處理之動作時間表的詳細資訊。Worker 記錄可以包含多個工作階段的相關資訊

您可以下載工作階段和 Worker 記錄，以便離線檢查它們。

若要檢視工作階段記

1. 請按照中在截止日期雲中查看工作詳細的步驟檢視工作清單。
2. 從 Jobs (任務) 清單中選擇一項任務。
3. 從「步驟」清單中選取一個步驟。
4. 從 [工作] 清單中選取工作。
5. 從「動作」功能表中選擇「檢視記錄檔」。

「時間表」段落會顯示工作的動作摘要。若要查看工作階段中執行的更多工作，並查看工作階段的關閉動作，請選擇 [檢視所有工作的記錄]。

若要從工作檢視背景工作者記錄

1. 請按照中在截止日期雲中查看工作詳細的步驟檢視工作清單。
2. 從 Jobs (任務) 清單中選擇一項任務。
3. 從「步驟」清單中選取一個步驟。
4. 從 [工作] 清單中選取工作。
5. 從「動作」功能表中選擇「檢視記錄檔」。
6. 選擇工作階段資訊。
7. 選擇 [檢視工作者記錄]

若要從叢集詳細資料檢視 Worker 記錄

1. 請按照中的步在期限雲端中檢視佇列和車隊詳細資料驟檢視叢集。
2. 從 [Worker] 清單中選取 [Worker ID]。
3. 從 [動作] 功能表中，選擇 [檢視 Worker 記錄]。

在截止日期雲下載完成的輸出

工作完成後，您可以使用 AWS 截止日期雲端監視器將結果下載到您的工作站。輸出檔案會以您在建立工作時指定的名稱和位置儲存。

輸出檔案會無限期儲存。若要降低儲存成本，請考慮為佇列的 Amazon S3 儲存貯體建立 S3 生命週期組態。如需詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的管理儲存生命週期。

若要下載工作、步驟或工作的已完成輸出

1. 請按照中在截止日期雲中查看工作詳細的步驟檢視工作清單。
2. 選取您要下載輸出的工作、步驟或工作。
   - 如果您選取工作，您可以下載該工作所有步驟中所有工作的所有輸出。
   - 如果您選取步驟，您可以下載該步驟中所有工作的所有輸出。
   - 如果您選取工作，您可以下載該個別工作的輸出。
3. 從「動作」功能表中選擇「下載輸出」。
4. 輸出將下載到提交作業時設定的位置。

> **Note**
>
> 目前僅支援使用功能表下載輸Windows出Linux。如果您有Mac並且選擇 [下載] 輸出功能表項目，則會出現一個視窗，顯示可用來下載轉譯輸出的 AWS CLI 命令。

截止日期雲農場

伺服器陣列是佇列的容器，可管理執行工作的工作和執行工作之計算資源叢集。

主題

建立伺服器陣列

1. 在截止日期雲端主控台中，選擇移至儀表板。
2. 在截止日期雲端儀表板的「農場」區段中，選擇「動作 → 建立伺服器陣列」
   - 或者，在左側面板中選擇伺服器陣列和其他資源，然後選擇建立伺服器陣列。
3. 為您的伺服器陣列新增名稱。
4. 在說明中，輸入伺服器陣列說明。清楚的說明可協助您快速識別伺服器陣列的用途。
5. (選擇性) 根據預設，您的資料會使用 AWS 擁有並管理您的安全性的金鑰加密。您可以選擇 [自訂加密設定 (進階)] 以使用現有的金鑰或建立您管理的新金鑰。

   如果您選擇使用核取方塊自訂加密設定，請輸入 AWS KMS ARN，或選擇建立新 AWS KMS 的 KMS 金鑰來建立新的金鑰。
6. (選擇性) 選擇 [新增標籤]，將一或多個標籤新增至伺服器陣列。
7. 選擇 [建立農場]。建立之後，會顯示伺服器陣列。

刪除伺服器陣列

1. 從截止日期雲端儀表板中，選擇伺服器陣列和其他資源。
2. 在伺服器陣列清單中，選取要刪除的一或多個伺服器陣列，然後選擇 [刪除]。

編輯伺服器陣列

1. 從截止日期雲端儀表板中，選擇伺服器陣列和其他資源。

2. 在伺服器陣列清單中，選取要刪除的一或多個伺服器陣列，然後選擇 [編輯]。
3. 在顯示的編輯視窗中，變更伺服器陣列名稱或說明，然後選擇 [儲存變更]。

期限雲端佇列

佇列是管理和處理工作的伺服器陣列資源。

若要使用佇列，您應該已經設定了監視器和伺服器陣列。

主題

建立佇列

1. 在截止日期雲端主控台儀表板中，選取您要為其建立佇列的伺服器陣列。
   - 或者，在左側面板中選擇伺服器陣列和其他資源，然後選取您要為其建立佇列的伺服器陣列。
2. 在 [佇列] 索引標籤中選擇 [建立佇列]。
3. 輸入佇列的名稱。
4. 在說明中，輸入佇列說明。說明可協助您識別佇列的用途。
5. 對於 Job 務附件，您可以建立新的 Amazon S3 儲存貯體，或選擇現有的 Amazon S3 儲存貯體。
   a. 若要建立新的 Amazon S3 儲存貯體
      i. 選取「建立新的工作時段」。
      ii. 輸入值區的名稱。我們建議您為值區命名`deadlinecloud-job-attachments-[MONITORNAME]`。
      iii. 輸入根前置詞以定義或變更佇列的根位置。
   b. 若要選擇現有的 Amazon S3 儲存貯體
      i. 選取 [選擇現有的 S3 儲存貯體] > [瀏覽 S3]。
      ii. 從可用儲存貯體清單中選取佇列的 S3 儲存貯體。
6. (選擇性) 若要將佇列與客戶管理的叢集建立關聯，請選取「啟用與客戶管理的叢集關聯」。

7. 如果您啟用與客戶管理的叢集關聯，則必須完成下列步驟。

> **Important**
>
> 強烈建議您指定執行身分功能的使用者和群組。如果不這樣做，它會降低伺服器陣列的安全性狀態，因為這些工作可以完成工作代理程式可以執行的所有動作。如需有關潛在安全威脅的詳細資訊，請參閱以使用者和群組身分執行工作。

a. 對於以使用者身分執行：

若要提供佇列工作的證明資料，請選取已設定佇列的使用者。

或者，若要選擇不設定您自己的認證並以 Worker 代理程式使用者身分執行工作，請選取 Worker 代理程式使用者。

b. (選擇性) 針對以使用者身分證明執行，輸入使用者名稱和群組名稱，以提供佇列工作的認證。

如果您正在使用Windows叢集，則必須建立 AWS Secrets Manager 包含執行身分使用者密碼的密碼。請依照下列指示建立密碼。使用的名稱取代`####`具。`jobRunAsUser`

i. 以管理員身份打開PowerShell或命令提示符。

ii. 建立使用者。

```
net user jobuser /add
```

iii. 設定密碼。

```
net user jobuser *
```

iv. 為使用者建立本機設定檔和主目錄。執行下列命令，並在出現提示時輸入使用者的密碼。

```
runas /profile /user:jobuser "cmd.exe /C"
```

8. 要求預算有助於管理佇列的成本。選取 [不需要預算] 或 [需要預算]。

9. 您的佇列需要權限才能代表您存取 Amazon S3。您可以建立新的服務角色或使用現有的服務角色。如果您沒有現有的服務角色，請建立並使用新的服務角色。

a. 若要使用現有的服務角色，請選取 [選擇服務角色]，然後從下拉式清單中選取角色。

b. 若要建立新的服務角色，請選取 [建立並使用新的服務角色]，然後輸入角色名稱和說明。

10. (選擇性) 若要為佇列環境新增環境變數，請選擇 [新增環境變數]，然後為您新增的每個變數輸入名稱和值。
11. (選擇性) 選擇 [新增標記]，將一或多個標籤新增至佇列。
12. 若要建立預設Conda佇列環境，請保持核取方塊保持選取狀態。若要深入了解佇列環境，請參閱建立佇列環境。如果您要為客戶管理的叢集建立佇列，請清除核取方塊。
13. 選擇建立佇列。

建立佇列環境

佇列環境是設定叢集 Worker 的一組環境變數和指令。您可以使用佇列環境為佇列中的工作提供軟體應用程式、環境變數和其他資源。

建立佇列時，您可以選擇建立預設Conda佇列環境。此環境可讓服務管理叢集存取合作夥伴 DCC 應用程式和轉譯器的套件。如需詳細資訊，請參閱 預設Conda佇列環境。

您可以使用主控台或直接編輯 json 或 YAML 範本來新增佇列環境。此程序說明如何使用主控台建立環境。

1. 若要將佇列環境新增至佇列，請導覽至佇列並選取佇列環境索引標籤。
2. 選擇動作，然後選擇使用表單建立新的
3. 輸入佇列環境的名稱和說明。
4. 選擇 [新增環境變數]，然後為您新增的每個變數輸入名稱和值。
5. (選擇性) 輸入佇列環境的優先順序。優先順序表示此佇列環境將在 Worker 上執行的順序。優先順序較高的佇列環境會先執行。
6. 選擇 [建立佇列環境]。

預設Conda佇列環境

當您建立與服務管理的叢集相關聯的佇列時，您可以選擇新增預設佇列環境，Conda以支援在虛擬環境中為工作下載和安裝套件。

Conda提供來自頻道的套件。通道是儲存套件的位置。截止日期雲端提供一個管道`deadline-cloud`，用於託管支援合作夥伴 DCC 應用程式和轉譯器的套件。這些軟件包是：

- 攪拌機

  - blender=3.6
  - blender-openjd
- 胡迪尼
  - houdini=19.5
  - houdini-openjd
- Maya
  - maya=2024
  - maya-mtoa=2024.5.3
  - maya-openjd
- 核彈
  - nuke=15
  - nuke-openjd

當您將工作提交至具有預設Conda環境的佇列時，環境會將兩個參數新增至工作。這些參數會指定在處理工作之前，用來設定工作環境的Conda套裝軟體和通道。參數如下：

- CondaPackages— 以空格分隔的套件符合規格清單，例如blender=3.6或numpy>1.22。預設值為空白，可略過建立虛擬環境。
- CondaChannels— 以空格分隔的Conda頻道清單deadline-cloud，例如conda-forge、或s3://*DOC-EXAMPLE-BUCKET*/conda/channel。預設值為deadline-cloud提供合作夥伴 DCC 應用程式和轉譯器的服務管理叢集可使用的通道。

當您使用整合式提交者將工作從 DCC 傳送至 Deptional Cloud 時，提交者會根據 DCC 應用程式和提交者填入CondaPackages參數的值。例如，如果您正在使用 Blender，則CondaPackage參數設定為blender=3.6.* blender-openjd=0.4.*。

刪除佇列

> **Warning**
>
> 如果刪除佇列，則無法復原佇列中的工作。刪除佇列也會刪除該佇列中的工作。

1. 從截止日期雲端儀表板中，選擇伺服器陣列和其他資源。

2. 在伺服器陣列清單中，選取包含要刪除之佇列的伺服器陣列。
3. 選取佇列，然後選擇 [刪除]。
4. 在確認視窗中，選擇 Delete (刪除)。您的佇列和佇列中的所有工作都會被刪除。

編輯佇列

1. 從截止日期雲端儀表板中，選擇伺服器陣列和其他資源。
2. 在伺服器陣列清單中，選取包含要編輯之佇列的伺服器陣列。
3. 選取佇列，然後選擇 [編輯]。
4. 您可以編輯名稱、說明、預算需求、執行身分使用者選項，以及指定的服務角色。您也可以將現有叢集與佇列建立關聯。
5. 選擇儲存變更。

建立佇列與叢集的關聯

1. 選取要與叢集建立關聯的佇列。
2. 若要選取要與佇列建立關聯的叢集，請選擇「關聯叢集」。
3. 選擇 [選取艦隊] 下拉式清單。會顯示可用叢集的清單。
4. 從可用叢集清單中，選取您要與佇列建立關聯的一或多個叢集旁邊的核取方塊。
5. 選擇關聯。叢集關聯狀態現在應該是 [關聯]。

管理截止日期雲端叢集

本節說明如何針對期限雲端管理服務管理的叢集 (SMF) 和客戶管理叢集 (CMF)。

您可以設定兩種截止日期雲端叢集類型：

- 服務管理的叢集是工作者群組，這些工作者擁有此服務所提供的預設設定，即截止日期雲端。這些預設設定的設計目的是要有效率且符合成本效益。
- 客戶管理的叢集 (CMF) 是您所管理的員工群。CMF 可以位於 AWS 基礎設施、內部部署或共置的資料中心內。CMF 提供車隊的完全控制和責任。這包括佈建、作業、管理和解除委任叢集中的工作人員。

主題

管理截止日期雲端服務管理的叢集

服務管理的叢集是擁有由截止日期雲端提供的預設設定的工作者群組。這些預設設定是為了有效率且符合成本效益而設計。

1. 若要建立服務管理的叢集 (SMF)，請瀏覽至您要在其中建立叢集的伺服器陣列。
2. 選取 [叢集] 索引標籤。
3. 選擇 Create fleet (建立機群)。
4. 輸入叢集的「名稱」。
5. 輸入 Description (描述)。清晰的描述可以幫助您快速識別車隊的目的。
6. 選取服務管理的叢集類型。
7. 為您的叢集選擇競價型或隨需執行個體市場選項。Spot 執行個體是未預留容量，您可以以折扣價格使用，但可能會因隨需請求而中斷。隨需執行個體按第二個定價，但沒有長期承諾，也不會中斷。根據預設，叢集會使用 Spot 執行個體。
8. 選擇性設定擴展叢集的執行個體數目上限，以便佇列中的工作可用容量。我們建議您保留最少數量的執行個體，**0**以確保叢集在沒有任何作業排入佇列時釋放所有執行個體。
9. 如需叢集的服務存取權，請選取現有角色或建立新角色。服務角色提供認證給叢集中的執行個體，授與他們處理工作的權限，以及監視器中的使用者，以便他們能夠讀取記錄資訊。

10. 選擇下一步。
11. 輸入叢集所需的最小和最大 vCPU。
12. 輸入叢集所需的最小和最大記憶體。
13. 選擇性您可以選擇允許或排除叢集中的特定執行個體類型，以確保僅將這些執行個體類型用於此叢集。
14. 選用您可以指定將連接至此叢集中工作者的 Amazon 彈性區塊存放區 (Amazon EBS) gp3 磁碟區的大小。如需詳細資訊，請參閱 EBS 使用者指南。
15. 選擇下一步。
16. 選擇性定義自訂 Worker 需求，以定義此叢集的功能，這些功能可與工作提交上指定的自訂主機需求結合使用。如果您計劃將叢集連接到自己的授權伺服器，則其中一個範例是特定的授權類型。
17. 選擇下一步。
18. 選擇性若要將叢集與佇列產生關聯，請從下拉式清單中選取佇列。如果佇列是以預Conda設佇列環境設定，系統會自動為您的叢集提供支援合作夥伴 DCC 應用程式和轉譯器的套件。如需提供套件的清單，請參閱預設Conda佇列環境。
19. 選擇下一步。
20. 選擇性若要將標籤新增至叢集，請選擇 [新增標籤]，然後輸入該標籤的金鑰和值。
21. 選擇下一步。
22. 檢閱您的叢集設定，然後選擇 [建立叢集]。建立後，您的叢集便會顯示出來。

VFX Reference Platform 相容性

VFX Reference Platform這是視覺特效產業常見的目標平台。若要將執行 Amazon Linux 2023 的標準服務管理叢集 Amazon EC2 執行個體與支援的軟體搭配使用VFX Reference Platform，在使用服務管理的叢集時，請記住以下考量事項。

每年更新VFX Reference Platform一次。這些使用 AL2023 的考量因素，包括截止日期雲端服務管理的機隊，是以 2022 年至 2024 年參考平台為基礎。如需詳細資訊，請參閱 VFX Reference Platform。

> **Note**
>
> 如果要為客戶管理的叢集建立自訂 Amazon Machine Image (AMI)，則可以在準備 Amazon EC2 執行個體時新增這些需求。

若要在 AL2023 Amazon EC2 執行個體上使用VFX Reference Platform支援的軟體，請考慮下列事項：

- 使用 AL2023 安裝的 glibc 版本相容於執行階段使用，但不適用於建置與 VFX Reference Platform CY2024 或更早版本相容的軟體。
- Python 務管理的叢集提供了與 VFX Reference Platform CY2022 和 CY2024 相容的服務管理叢集。服務管理的叢集中不提供 Python 3.7 和 3.10。需要它們的軟體必須在佇列或工作環境中提供 Python 安裝。
- 服務管理叢集中提供的某些 Boost 程式庫元件為 1.75 版，與. VFX Reference Platform 如果您的應用程式使用 Boost，您必須提供您自己的程式庫版本以確保相容性。
- Intel TBB 更新 3 是在服務管理的叢集中提供的。此功能與 VFX Reference Platform CY2022、CY2023 和 CY2024 相容。
- 服務管理的叢集不會提供其他具VFX Reference Platform有指定版本的程式庫。您必須向程式庫提供服務管理的叢集中使用的任何應用程式。如需程式庫清單，請參閱參考平台。

管理截止日期雲端客戶管理的叢集

本節說明如何管理期限雲端的客戶管理叢集 (CMF)。

CMF 是您管理的員工隊伍。CMF 可以位於 AWS 基礎設施、內部部署或共置的資料中心內。CMF 提供車隊的完全控制和責任。這包括佈建、作業、管理和解除委任叢集中的工作人員。

主題

- 建立客戶管理的叢集
- 背景工作主機設定和組態
- 管理對 Windows 工作使用者密碼的存取
- 安裝和設定工作所需的軟體
- 設定 AWS 認證
- 建立 Amazon Machine Image
- 使用 Amazon EC2 自動擴展群組建立叢集基礎設施
- Connect 客戶管理的叢集連線到授權端點

建立客戶管理的叢集

若要建立客戶管理的叢集 (CMF)，請完成以下步驟。

Deadline Cloud console

使用截止日期雲端主控台建立客戶管理的叢集

1. 開啟截止日期雲端主控台。
2. 選取「農場」。會顯示可用伺服器陣列的清單。
3. 選取您要在其中工作的伺服器陣列名稱。
4. 選取 [叢集] 索引標籤。
5. 選擇 Create fleet (建立機群)。
6. 輸入叢集的「名稱」。
7. (選擇性) 輸入叢集的「說明」。
8. 針對「機隊類型」選取「客戶管理
9. 選取「Auto Scaling」類型。如需詳細資訊，請參閱用 EventBridge 來處理 Auto Scaling 事件。
   - 無擴展：您正在建立內部部署叢集，並希望選擇退出截止日期雲端 Auto Scaling。
   - 擴展建議：您正在建立一個亞馬遜彈性運算雲端 (Amazon EC2) 叢集。
10. 選擇您車隊的服務存取權限。
    a. 我們建議針對每個叢集使用 [建立和使用新的服務角色] 選項，以進行更精細的權限控制。預設會選取此選項。
    b. 您也可以選取 [選擇服務角色]，以使用現有的服務角色。
11. 檢視您的選擇，然後選擇「下一步」。
12. 選取叢集的作業系統。所有車隊的工作人員都必須擁有通用的操作系統。
13. 選取主機 CPU 架構。
14. 為此叢集中的工作者主機選取下列硬體需求。
    a. 選取最小和最大 vCPU 和記憶體硬體需求，以符合叢集的工作負載需求。
    b. (選擇性) 選取 GPU 需求，然後輸入最小和最大 GPU。
15. 檢視您的選擇，然後選擇「下一步」。
16. (選擇性) 定義自訂 Worker 需求。
17. 使用下拉式清單，選取一或多個要與叢集建立關聯的佇列。

> **Note**
>
> 我們建議您只將叢集與全部位於相同信任界限的佇列產生關聯。這可確保在同一個 Worker 上執行作業之間具有強大的安全性界限。

18. 複查佇列關聯，然後選取下一步。
19. (選擇性) 對於預設 Conda 佇列環境，我們會為您的佇列建立環境，以安裝工作要求的 Conda 套件。

> **Note**
>
> Conda 佇列環境是用來安裝工作要求的 Conda 套件。一般而言，您應該取消核取與 CMF 相關聯之佇列上的 Conda 佇列環境，因為 CMF 預設不會安裝必要的 Conda 命令。

20. (選擇性) 將標籤新增至您的 CMF。如需詳細資訊，請參閱標記 AWS 資源。
21. 檢閱您的叢集組態並進行任何變更。
22. 選擇 Create fleet (建立機群)。
23. 選取 [艦隊] 索引標籤，然後記下 [叢集 ID]。

AWS CLI

若要使用建 AWS CLI 立客戶管理的叢集

1. 開啟 AWS CLI.
2. 編輯 `fleet-trust-policy.json`.

   a. 新增下列 IAM 政策，將##文字取代為您的 AWS 帳戶 ID 和截止日期雲端伺服器陣列 ID。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "credentials.deadline.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "ACCOUNT_ID"
                },
                "ArnEquals": {
                    "aws:SourceArn":
 "arn:aws:deadline:*:ACCOUNT_ID:farm/FARM_ID"
                }
            }
        }
    ]
}
```

b. 儲存您的變更。

3. 編輯 `create-cmf-fleet.json`.

a. 新增下列 IAM 政策。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "deadline:AssumeFleetRoleForWorker",
                "deadline:UpdateWorker",
                "deadline:DeleteWorker",
                "deadline:UpdateWorkerSchedule",
                "deadline:BatchGetJobEntity",
                "deadline:AssumeQueueRoleForWorker"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
```

```
            "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        }
    ]
}
```

b. 儲存您的變更。

4. 為叢集中的員工新增 IAM 角色以供使用。

```
aws iam create-role --role-name FleetWorkerRoleName --assume-role-policy-
document file://fleet-trust-policy.json
aws iam put-role-policy --role-name FleetWorkerRoleName --policy-name
 FleetWorkerPolicy --policy-document file://fleet-policy.json
```

5. 編輯 `create-fleet-request.json`.

a. 新增下列 IAM 政策，將斜體文字取代為 CMF 的值。

> **Note**
>
> 您可以在*##*到。`create-cmf-fleet.json`
> 對於 *OS_FAMIL* Y，您必須選擇`linux`、`macos`或之一。`windows`

```
{
```

```
    "farmId": "FARM_ID",
    "displayName": "FLEET_NAME",
    "description": "FLEET_DESCRIPTION",
    "roleArn": "ROLE_ARN",
    "minWorkerCount": 0,
    "maxWorkerCount": 10,
    "configuration": {
        "customerManaged": {
            "mode": "NO_SCALING",
            "workerCapabilities": {
                "vCpuCount": {
                    "min": 1,
                    "max": 4
                },
                "memoryMiB": {
                    "min": 1024,
                    "max": 4096
                },
                "osFamily": "OS_FAMILY",
                "cpuArchitectureType": "x86_64",
            },
        },
    }
}
```

b. 儲存您的變更。

6. 建立您的車隊。

```
aws deadline create-fleet --cli-input-json file://create-fleet-request.json
```

背景工作主機設定和組態

工作者主機是指執行期限雲端背景工作者的主機。本節說明如何設定 Worker 主機，並針對您的特定需求進行設定。每個工作者主機都會執行稱為 Worker Agent 的程式。工人代理負責：

- 管理工作者生命週期。
- 同步分配的工作，其進度和結果。
- 監控執行中的工作。
- 將記錄轉送至設定的目的地。

我們建議您使用提供的期限雲端背景工作者代理程式。Worker 代理程式是開放原始碼的，我們鼓勵您提出功能要求，但您也可以開發和自訂以符合您的需求。

若要完成下列各節中的工作，您需要下列項目：

Linux

- Linux基於 Amazon Elastic Compute Cloud (Amazon EC2) 實例。我們推薦 Amazon 2023.
- sudo權限。
- Python 3.9 或以上。

Windows

- Windows基於 Amazon Elastic Compute Cloud (Amazon EC2) 實例。我們推薦Windows Server 2022。
- 工作者主機的管理員存取權
- 為所有用戶安裝了 Python 3.9 或更高版本

建立並設定 Python 虛擬環境

你可以創建一個 Python 虛擬環境，Linux如果你已經安裝了 Python 3.9 或更高版本，並把它放在你的PATH.

若要建立並啟動 Python 虛擬環境

1. 開啟 AWS CLI.
2. 建立並啟動 Python 虛擬環境。

```
python3 -m venv /opt/deadline/worker
source /opt/deadline/worker/bin/activate
pip install --upgrade pip
```

安裝期限雲端工作者代理

在設定 Python 並建立虛擬環境之後Linux，請安裝截止日期雲端背景工作者代理程式 Python 套件。

若要安裝工作者代理程式 Python 套件

1. 開啟終端機。

   a. 開啟Linux，以使用root者身分開啟終端機 (或使用sudo/su)

   b. 開啟Windows，開啟系統管理員命令提示字元或 PowerShell終端機。

2. 從 PyPI 下載並安裝期限雲端工作者代理程式套件：

> **Note**
>
> 在上Windows，代理程式檔案必須安裝到 Python 的全域網站套件目錄中。目前不支援 Python 虛擬環境。

```
python -m pip install deadline-cloud-worker-agent
```

設定期限雲端工作者代理程式

您可以透過三種方式設定期限雲端背景工作者代理程式設定。我們建議您使用通過設置的操作系統install-deadline-worker。

命令列引數 — 您可以在從命令列執行「截止日期雲端工作者代理程式」時指定引數。某些組態設定無法透過命令列引數使用。要查看所有可用的命令行參數，請輸入deadline-worker-agent --help以查看所有可用的命令行參數。

環境變數 — 您可以透過設定開頭為的環境變數來設定截止日期 Cloud Worker 代理程式DEADLINE_WORKER_。例如，您可以使用export DEADLINE_WORKER_VERBOSE=true將 Worker 代理程式的輸出設定為詳細資訊。有關實例和詳細資訊，請參閱中/etc/amazon/deadline/worker.toml.example的Linux或中C:\ProgramData\Amazon\Deadline\Config\worker.toml.example的Windows。

配置檔案 — 當您安裝 Worker 代理程式時，它會建立位於/etc/amazon/deadline/worker.toml上Linux或C:\ProgramData\Amazon\Deadline\Config\worker.toml上的配置檔案Windows。Worker 代理程式會在啟動時載入此組態檔案。您可以使用範例組態檔案 (開/etc/amazon/deadline/worker.toml.example啟Linux或開啟Windows)，針C:\ProgramData\Amazon\Deadline\Config\worker.toml.example對您的特定需求量身打造預設 Worker 代理程式組態檔案。

最後，我們建議您啟用 Worker 代理程式的 auto 關機。這可讓 Worker 叢集在需要時向上擴充，並在轉譯工作完成時關閉。自動調整功能有助於確保您只在需要時使用資源。

啟用 auto 關機

作為使**root**用者：

- 使用參數安裝 Worker 代理程式**--allow-shutdown**。

  Linux

  輸入：

  ```
  /opt/deadline/worker/bin/install-deadline-worker \
    --farm-id FARM_ID \
    --fleet-id FLEET_ID \
    --region REGION \
    --allow-shutdown
  ```

  Windows

  輸入：

  ```
  install-deadline-worker ^
    --farm-id FARM_ID ^
    --fleet-id FLEET_ID ^
    --region REGION ^
    --allow-shutdown
  ```

建立工作使用者和群組

本節說明代理程式使用者與佇列中jobRunAsUser定義的使用者之間所需的使用者和群組關係。

截止日期 Cloud Worker 代理程式應以主機上的專用代理程式特定使用者身分執行。您應該jobRunAsUser設定「截止日期雲端佇列」的內容，以便 Worker 以特定作業系統使用者和群組的身分執行佇列工作。這表示您可以控制工作擁有的共用檔案系統權限。它還提供了作業與 Worker Agent 使用者之間的重要安全性界限。

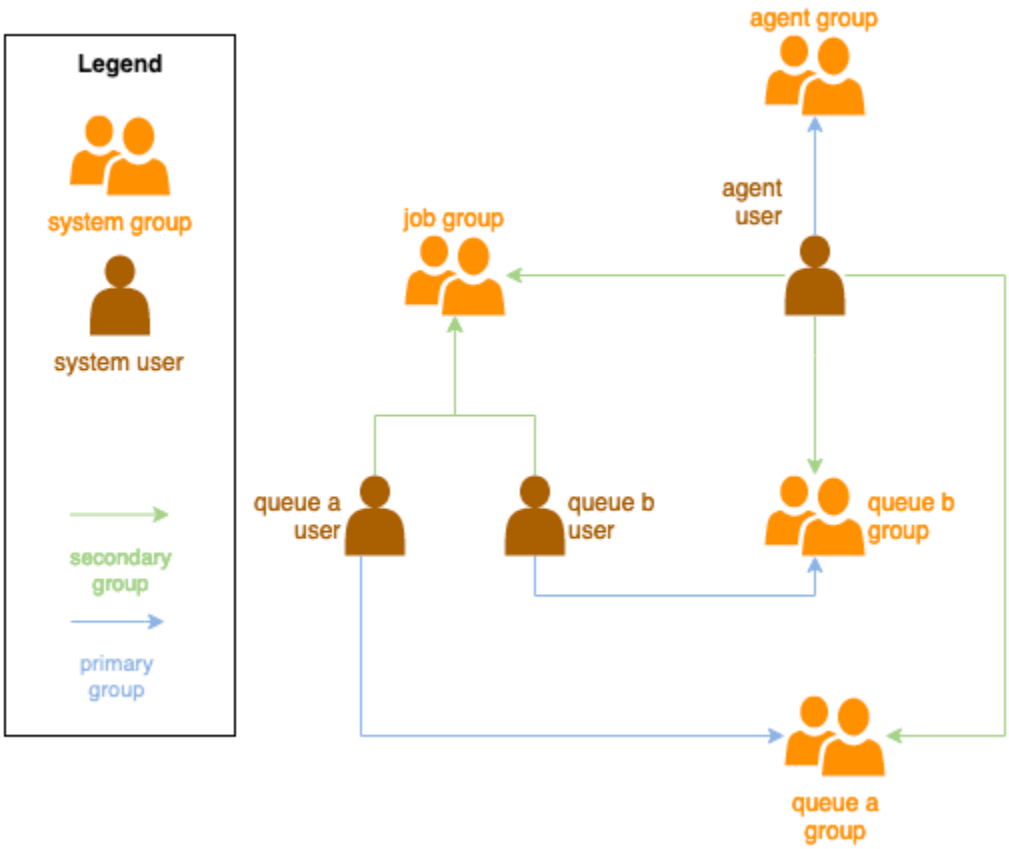
Linux工作使用者和群組

若要設定您的代理程式使用者jobRunAsUser，並確定您符合下列需求：

- 每個群組都有一個群組jobRunAsUser，它是其對應的主要群組jobRunAsUser。
- 代理程式-使用者屬於 Worker 取得工jobRunAsUser作之佇列的主要群組。基於安全性最佳做法，我們建議您將其作為代理程式使用者的次要群組。此共用群組可讓 Worker 代理程式在工作執行時提供檔案供工作使用。
- A jobRunAsUser 不屬於代理程式-使用者的主要群組。針對安全性最佳做法：
  - Worker 代理程式所寫入的敏感檔案是由代理程式的主要群組所擁有。
  - 如果jobRunAsUser屬於此群組，且 Worker 代理程式寫入的檔案可由提交至 Worker 上執行之佇列的工作存取。
- 預設的 [AWS 區域] 應符合 Worker 所屬的伺服器陣列的 [區域]。如需詳細資訊，請參閱組態和認證檔案設定。

  這應該適用於：
  - 代理程式-使用者
  - Worker 上的所有佇列jobRunAsUser帳戶
- 代理程式使用者可以執行sudo命令為. jobRunAsUser

下圖說明代理程式使用者與叢集關聯之佇列的使jobRunAsUser用者與群組之間的關係。



Windows 使用者

若要使用使用Windows者作為jobRunAsUser，它必須符合下列需求：

- 所有佇列jobRunAsUser使用者都必須存在。
- 他們的密碼必須與佇列JobRunAsUser欄位中指定的密碼值相符。如需指示，請參閱中的步驟 7 建立佇列。
- 代理程式使用者必須能夠以這些使用者身分登入。

管理對 Windows 工作使用者密碼的存取

當您使用 Windows 設定佇列時jobRunAsUser，您必須指定密碼 AWS Secrets Manager 碼。此密碼的值應為以下格式的 JSON 編碼物件：

```
{
    "password": "JOB_USER_PASSWORD"
}
```

若要讓 Worker 以佇列的設定方式執行工作jobRunAsUser，叢集的 IAM 角色必須具有權限才能取得密碼的值。如果使用客戶管理的 KMS 金鑰加密密碼，則叢集的 IAM 角色也必須具有使用 KMS 金鑰解密的權限。

強烈建議遵循這些秘密的最低權限原則。這意味著訪問獲取隊列 jobRunAsUser → windows → 的秘密值passwordArn應該是：

- 在叢集與佇列之間建立佇列-叢集關聯時，授與叢集角色
- 刪除叢集與佇列之間的佇列-叢集關聯時，已從叢集角色撤銷

此外，當 AWS 密碼不再使用時，應刪除包含jobRunAsUser密碼的秘密管理員密碼。

授與密碼密碼的存取權

當佇列和叢集相關聯時，雲端叢集需要存取佇列密碼密碼機密中所儲存的密碼。jobRunAsUser我們建議您使用 AWS Secrets Manager 資源原則來授與叢集角色的存取權。如果您嚴格遵守此準則，則更容易判斷哪些叢集角色可以存取密碼。

若要授予密碼存取權

1. 開啟 AWS 密碼管理員主控台以取得密碼。
2. 在「資源權限」區段中，新增表單的政策陳述式：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    // ...
    {
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "FLEET_ROLE_ARN"
      },
      "Action" : "secretsmanager:GetSecretValue",
      "Resource" : "*"
    }
    // ...
  ]
}
```

撤銷對密碼密碼的存取

當叢集不再需要佇列的存取權時，請移除佇列密碼密碼的存取權jobRunAsUser。我們建議您使用 AWS Secrets Manager 資源原則來授與叢集角色的存取權。如果您嚴格遵守此準則，則更容易判斷哪些叢集角色可以存取密碼。

若要撤銷對密碼的存取權

1. 開啟 AWS 密碼管理員主控台以取得密碼。
2. 在 [資源權限] 區段中，移除表單的政策陳述式：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    // ...
    {
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "FLEET_ROLE_ARN"
      },
      "Action" : "secretsmanager:GetSecretValue",
      "Resource" : "*"
    }
    // ...
  ]
}
```

安裝和設定工作所需的軟體

設定截止日期 Cloud Worker 代理程式之後，您可以使用執行作業所需的任何軟體來準備背景工作主機。

當您將工作送至具有相關聯的佇列時jobRunAsUser，該工作會以該使用者的身分執行。所有指令都必須在該PATH使用者中可用。

在 Linux 上，您可以在下列其中一項中PATH為使用者指定：

- 他們的~/.bashrc或 ~/.bash_profile
- 系統組態檔案，例如/etc/profile.d/*和 /etc/profile
- 殼層啟動指令碼：/etc/bashrc.

在 Windows 上，您可以在下列其中一個項目中PATH為使用者指定：

- 他們的用戶特定環境變量
- 系統範圍的環境變量

安裝數位內容建立工具轉接器

截止日期雲端為數位內容建立 (DCC) 應用程式提供第一方整合支援。若要在客戶管理的叢集上使用這些整合，您必須安裝 DCC 軟體和介面卡。

在客戶管理的機群上安裝 DCC 配接器

1. 打開終端。
   a. 在 Linux 上，以使用root者身分開啟終端機 (或使用sudo/su)
   b. 在 Windows 上，開啟系統管理員命令提示字元或 PowerShell 終端機。
2. 安裝期限雲端轉接器套件。

```
pip install deadline deadline-cloud-for-maya deadline-cloud-for-nuke deadline-
cloud-for-blender
```

設定 AWS 認證

本節說明如何設定 AWS 認證。

工作者生命週期的這個初始階段是啟動載入。在這個階段，Worker Agent 軟體會在您的叢集中建立 Worker，並從叢集的角色取得 AWS 認證以供進一步操作。

AWS credentials for Amazon EC2

若要設定 Amazon EC2 的 AWS 登入資料

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中選取角色，然後選取建立角色。
3. 選擇AWS 服務。
4. 選取 EC2 做為服務或使用案例，然後選取下一步。
5. 附加受AWSDeadlineCloud-WorkerHost AWS 管理的策略。

On-premise AWS credentials

若要設定 AWS 內部部署認證

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中選取角色，然後選取建立角色。
3. 選取 AWS 帳戶，然後選取下一步。
4. 附加受AWSDeadlineCloud-WorkerHost AWS 管理的策略。
5. 為 AWS IAM 使用者產生 IAM 存取權和秘密金鑰：
   a. 如需 IAM 角色任何地方，請參閱隨處可見 IAM 角色。
   b. 如需在主機上設定登入資料的最安全方式，請參閱從 AWS Identity and Access Management 角色隨處取得臨時安全登入資料。
   c. 您也可以使用 CLI 做為替代身份驗證，有關詳情，請參閱使用 IAM 使用者登入資料進行身份驗
6. 將這些金鑰儲存在 Worker 主機檔案系統上的代理程式使用者 AWS 認證檔案中。
   a. 在 Linux 上，它位於 ~/.aws/credentials
   b. 在視窗上，它位於 %USERPROFILE%\.aws\credentials

> **Note**
>
> 只有安裝 Worker 代理程式的作業系統使用者名稱 (deadline-worker-agent) 才能存取認證。

```
# Replace keys below
[default]
aws_access_key_id=ACCESS_KEY_ID
aws_secret_access_key=SECRET_ACCESSS_KEY
```

7. 變更deadline-worker-agent擁有者和權限。

> **Note**
>
> 如果您在安裝 Worker 代理程式時變更了 OS 使用者 (`deadline-worker-agent`) 名稱，請改用該名稱。

建立 Amazon Machine Image

若要建立 Amazon Machine Image (AMI) 以在 Amazon Elastic Compute Cloud (Amazon EC2) 客戶管理叢集 (CMF) 中使用，請完成本節中的任務。您必須先建立 Amazon EC2 執行個體，才能繼續進行。如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的啟動執行個體。

> **Important**
>
> 建AMI立 Amazon EC2 執行個體的連接磁碟區建立快照。執行個體上安裝的任何軟體都會持續存在，因此執行個體會在您從AMI. 我們建議採用修補策略，並在申請到您的機隊之前定期AMI使用更新的軟件更新任何新軟件。

準備 Amazon EC2 實例

在建置之前AMI，您必須刪除 Worker 狀態。背景工作者代理程式啟動之間會持續存在，如果此狀態持續存在於AMI，則從它啟動的所有實例將共享相同的狀態。

我們也建議您刪除任何現有的記錄檔。當您準備 AMI 時，日誌檔案可以保留在 Amazon EC2 執行個體上。刪除這些檔案可在診斷使用 AMI 的 Worker 叢集中可能發生的問題時，將混淆降到最低。

您也應該啟用工作者代理程式系統服務，以便在 Amazon EC2 啟動時啟動期限雲端工作者代理程式。

最後，我們建議您啟用 Worker 代理程式 auto 關機。這可讓 Worker 叢集在需要時向上擴充，並在轉譯工作完成時關閉。這種 auto 擴展有助於確保您只在需要時使用資源。

若要準備亞馬遜 EC2 執行個體

1. 開啟 Amazon EC2 主控台。
2. 啟動 Amazon EC2 執行個體。如需詳細資訊，請參閱啟動執行個體。
3. 設定主機以連線至您的身分識別提供者 (IdP)，然後掛載所需的任何共用檔案系統。
4. 按照自學課程安裝期限雲端工作者代理，然後配置工作者代理，和建立工作使用者和群組。

5. 如果您要準備以 Amazon Linux 2023 為AMI基礎的軟體來執行與 VFX 參考平台相容的軟體，則需要更新數個需求。如需相關資訊，請參閱VFX Reference Platform 相容性。
6. 開啟終端機。
   a. 在 Linux 上，以使用root者身分開啟終端機 (或使用sudo/su)
   b. 在 Windows 上，開啟系統管理員命令提示字元或 PowerShell終端機。
7. 確保 Worker 服務未運行，並配置為在啟動時啟動：
   a. 在 Linux 上，執行

```
systemctl stop deadline-worker
systemctl enable deadline-worker
```

   b. 在視窗上，執行

```
sc.exe stop DeadlineWorker
sc.exe config DeadlineWorker start= auto
```

8. 刪除工作站狀態。
   a. 在 Linux 上，執行

```
rm -rf /var/lib/deadline/*
```

   b. 在視窗上，執行

```
del /Q /S %PROGRAMDATA%\Amazon\Deadline\Cache\*
```

9. 刪除記錄檔。
   a. 在 Linux 上，執行

```
rm -rf /var/log/amazon/deadline/*
```

   b. 在視窗上，執行

```
del /Q /S %PROGRAMDATA%\Amazon\Deadline\Logs\*
```

10. 在 Windows 上，建議執行「開始」功能表中的 Amazon EC2Launch 設定應用程式，以完成執行個體的最終主機準備和關閉。

> **Note**
>
> 您必須選擇不使用 Sysprep 的 [關機]，並且永遠不要選擇 [使用 Sysprep 關機]。使用 Sysprep 關閉會導致所有本機使用者變得無法使用。如需詳細資訊，請參閱 Windows 執行個體使用者指南中「建立自訂 AMI」主題的「開始之前」一節。

建置 AMI

若要建置 AMI

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中選取執行個體，然後選取您的執行個體。
3. 選擇執行個體狀態，然後選擇停止例項
4. 執行個體已停止之後，請選擇「動作」。
5. 選擇映像和範本，然後選擇建立映像。
6. 輸入影像名稱。
7. (選擇性) 輸入圖片說明。
8. 選擇 Create image (建立映像)。

使用 Amazon EC2 自動擴展群組建立叢集基礎設施

本節說明如何建立 Amazon EC2 Auto Scaling 叢集。

使用下面的 AWS CloudFormation YAML 範本建立 Amazon EC2 自動擴展 (Auto Scaling 動擴展) 群組、具有兩個子網路的 Amazon Virtual Private Cloud (Amazon VPC)、一個執行個體設定檔和一個執行個體存取角色。若要在子網路中使用「自動調整」(Auto Scaling) 啟動執行個體，

您應該檢閱並更新執行個體類型清單，以符合您的彩現需求。

若要建立 Amazon EC2 Auto Scaling 叢集

1. 開啟主 AWS CloudFormation 控台，網址為 https://console.aws.amazon.com/cloudformation。
2. 建立具有參數`Farm ID`、`Fleet ID`和的 CloudFormation 範本`AMI ID`。

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: Amazon Deadline Cloud customer-managed fleet
Parameters:
  FarmId:
    Type: String
    Description: Farm ID
  FleetId:
    Type: String
    Description: Fleet ID
  AMIId:
    Type: String
    Description: AMI ID for launcing Workers
Resources:
  deadlineVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: 100.100.0.0/16
  deadlineWorkerSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: !Join
        - ' '
        - - Security Group created for deadline workers in fleet
          - !Ref FleetId
      GroupName: !Join
        - ''
        - - deadlineWorkerSecurityGroup-
          - !Ref FleetId
      SecurityGroupEgress:
        - CidrIp: 0.0.0.0/0
          IpProtocol: '-1'
      SecurityGroupIngress: []
      VpcId: !Ref deadlineVPC
  deadlineIGW:
    Type: 'AWS::EC2::InternetGateway'
    Properties: {}
  deadlineVPCGatewayAttachment:
    Type: 'AWS::EC2::VPCGatewayAttachment'
    Properties:
      VpcId: !Ref deadlineVPC
      InternetGatewayId: !Ref deadlineIGW
  deadlinePublicRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref deadlineVPC
```

```
  deadlinePublicRoute:
    Type: 'AWS::EC2::Route'
    Properties:
      RouteTableId: !Ref deadlinePublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref deadlineIGW
    DependsOn:
      - deadlineIGW
      - deadlineVPCGatewayAttachment
  deadlinePublicSubnet0:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref deadlineVPC
      CidrBlock: 100.100.16.0/22
      AvailabilityZone: !Join
        - ''
        - - !Ref 'AWS::Region'
          - a
  deadlineSubnetRouteTableAssociation0:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref deadlinePublicRouteTable
      SubnetId: !Ref deadlinePublicSubnet0
  deadlinePublicSubnet1:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref deadlineVPC
      CidrBlock: 100.100.20.0/22
      AvailabilityZone: !Join
        - ''
        - - !Ref 'AWS::Region'
          - c
  deadlineSubnetRouteTableAssociation1:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref deadlinePublicRouteTable
      SubnetId: !Ref deadlinePublicSubnet1
  deadlineInstanceAccessAccessRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: !Join
        - '-'
        - - deadline
          - InstanceAccess
```

```
          - !Ref FleetId
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              Service: ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Path: /
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy'
        - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
        - 'arn:aws:iam::aws:policy/AWSDeadlineCloud-WorkerHost'
  deadlineInstanceProfile:
    Type: 'AWS::IAM::InstanceProfile'
    Properties:
      Path: /
      Roles:
        - !Ref deadlineInstanceAccessAccessRole
  deadlineLaunchTemplate:
    Type: 'AWS::EC2::LaunchTemplate'
    Properties:
      LaunchTemplateName: !Join
        - ''
        - - deadline-LT-
          - !Ref FleetId
      LaunchTemplateData:
        NetworkInterfaces:
          - DeviceIndex: 0
            AssociatePublicIpAddress: true
            Groups:
              - !Ref deadlineWorkerSecurityGroup
            DeleteOnTermination: true
        ImageId: !Ref AMIId
        InstanceInitiatedShutdownBehavior: terminate
        IamInstanceProfile:
          Arn: !GetAtt
            - deadlineInstanceProfile
            - Arn
        MetadataOptions:
          HttpTokens: required
          HttpEndpoint: enabled

  deadlineAutoScalingGroup:
```

```
    Type: 'AWS::AutoScaling::AutoScalingGroup'
    Properties:
      AutoScalingGroupName: !Join
        - ''
        - - deadline-ASG-autoscalable-
          - !Ref FleetId
      MinSize: 0
      MaxSize: 10
      VPCZoneIdentifier:
        - !Ref deadlinePublicSubnet0
        - !Ref deadlinePublicSubnet1
      NewInstancesProtectedFromScaleIn: true
      MixedInstancesPolicy:
        InstancesDistribution:
          OnDemandBaseCapacity: 0
          OnDemandPercentageAboveBaseCapacity: 0
          SpotAllocationStrategy: capacity-optimized
          OnDemandAllocationStrategy: lowest-price
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref deadlineLaunchTemplate
            Version: !GetAtt
              - deadlineLaunchTemplate
              - LatestVersionNumber
          Overrides:
            - InstanceType: m5.large
            - InstanceType: m5d.large
            - InstanceType: m5a.large
            - InstanceType: m5ad.large
            - InstanceType: m5n.large
            - InstanceType: m5dn.large
            - InstanceType: m4.large
            - InstanceType: m3.large
            - InstanceType: r5.large
            - InstanceType: r5d.large
            - InstanceType: r5a.large
            - InstanceType: r5ad.large
            - InstanceType: r5n.large
            - InstanceType: r5dn.large
            - InstanceType: r4.large
      MetricsCollection:
        - Granularity: 1Minute
          Metrics:
            - GroupMinSize
```

```
            - GroupMaxSize
            - GroupDesiredCapacity
            - GroupInServiceInstances
            - GroupTotalInstances
            - GroupInServiceCapacity
            - GroupTotalCapacity
```

3. 建立 IAM 角色後，您需要確認下列事項：

   - 附加至員工 Amazon EC2 執行個體的 IAM 角色的登入資料可供該工作者上執行的所有程序使用，其中包括任務。Worker 應具有最少的操作權限：`deadline:CreateWorker`和 `deadline:AssumeFleetRoleForWorker.`
   - Worker 代理程式會取得佇列角色的認證，並設定它們以供執行工作使用。Amazon EC2 執行個體設定檔角色不應包含任務所需的許可。

使用截止日期雲端擴展建議功能自動擴展 Amazon EC2 叢集

截止日期雲端利用 Amazon EC2 Auto Scaling (Auto Scaling) 群組自動擴展 Amazon EC2 客戶管理叢集 (CMF)。您必須設定叢集模式，並在帳戶中部署所需的基礎結構，才能讓叢集 auto 擴充。您部署的基礎架構將適用於所有艦隊，因此您只需設置一次即可。

基本工作流程是：您將叢集模式設定為 auto 擴充，然後在建議的叢集大小變更時 (其中一個 EventBridge 事件包含叢集識別碼、建議的叢集大小和其他中繼資料)，Perfate Cloud 就會傳送該叢集的事件。您將有一個 EventBridge 規則來篩選相關事件，並讓 Lambda 使用它們。Lambda 將與 Amazon EC2 自 Au `AutoScalingGroup` to Scaling 整合，以自動擴展 Amazon EC2 叢集。

將車隊模式設定為 **EVENT_BASED_AUTO_SCALING**

將您的叢集模式設定為`EVENT_BASED_AUTO_SCALING`。您可以使用主控台來執行此操作，或使用直接呼叫`CreateFleet`或 `UpdateFleet` API。 AWS CLI 設定模式後，只要建議的叢集大小變更，Perfate Cloud 就會開始傳送 EventBridge 事件。

- 範例`UpdateFleet`命令：

```
aws deadline update-fleet \
  --farm-id FARM_ID \
  --fleet-id FLEET_ID \
  --configuration file://configuration.json
```

- 範例`CreateFleet`命令：

```
aws deadline create-fleet \
  --farm-id FARM_ID \
  --display-name "Fleet name" \
  --max-worker-count 10 \
  --configuration file://configuration.json
```

以下是上述 CLI 指令中configuration.json使用的範例 (--configuration file://configuration.json)。

- 若要在叢集上啟用 Auto Scaling，您應該將模式設定為EVENT_BASED_AUTO_SCALING。
- 這workerCapabilities是建立 CMF 時指定給 CMF 的預設值。如果您需要增加 CMF 可用的資源，可以變更這些值。

設定叢集模式後，Detecout Cloud 會開始發出該叢集的叢集大小建議事件。

```
{
    "customerManaged": {
        "mode": "EVENT_BASED_AUTO_SCALING",
        "workerCapabilities": {
            "vCpuCount": {
                "min": 1,
                "max": 4
            },
            "memoryMiB": {
                "min": 1024,
                "max": 4096
            },
            "osFamily": "linux",
            "cpuArchitectureType": "x86_64",
        }
    }
}
```

使用範本部署 Auto Scaling AWS CloudFormation 模堆疊

您可以設定 EventBridge 規則來篩選事件、使用事件和控制 Auto Scaling 的 Lambda，以及用來儲存未處理事件的 SQS 佇列。使用下列 AWS CloudFormation 範本來部署堆疊中的所有項目。成功部署資源後，您可以提交工作，叢集會自動擴充。

```
Resources:
  AutoScalingLambda:
    Type: 'AWS::Lambda::Function'
    Properties:
      Code:
        ZipFile: |-
          """
          This lambda is configured to handle "Fleet Size Recommendation Change"
          messages. It will handle all such events, and requires
          that the ASG is named based on the fleet id. It will scale up/down the fleet
          based on the recommended fleet size in the message.

          Example EventBridge message:
          {
              "version": "0",
              "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
              "detail-type": "Fleet Size Recommendation Change",
              "source": "aws.deadline",
              "account": "111122223333",
              "time": "2017-12-22T18:43:48Z",
              "region": "us-west-1",
              "resources": [],
              "detail": {
                  "farmId": "farm-12345678900000000000000000000000",
                  "fleetId": "fleet-12345678900000000000000000000000",
                  "oldFleetSize": 1,
                  "newFleetSize": 5,
              }
          }
          """

          import json
          import boto3
          import logging

          logger = logging.getLogger()
          logger.setLevel(logging.INFO)

          auto_scaling_client = boto3.client("autoscaling")

          def lambda_handler(event, context):
              logger.info(event)
              event_detail = event["detail"]
```

```
              fleet_id = event_detail["fleetId"]
              desired_capacity = event_detail["newFleetSize"]

              asg_name = f"deadline-ASG-autoscalable-{fleet_id}"
              auto_scaling_client.set_desired_capacity(
                  AutoScalingGroupName=asg_name,
                  DesiredCapacity=desired_capacity,
                  HonorCooldown=False,
              )

              return {
                  'statusCode': 200,
                  'body': json.dumps(f'Successfully set desired_capacity for {asg_name}
to {desired_capacity}')
              }
      Handler: index.lambda_handler
      Role: !GetAtt
        - AutoScalingLambdaServiceRole
        - Arn
      Runtime: python3.11
    DependsOn:
      - AutoScalingLambdaServiceRoleDefaultPolicy
      - AutoScalingLambdaServiceRole
  AutoScalingEventRule:
    Type: 'AWS::Events::Rule'
    Properties:
      EventPattern:
        source:
          - aws.deadline
        detail-type:
          - Fleet Size Recommendation Change
      State: ENABLED
      Targets:
        - Arn: !GetAtt
            - AutoScalingLambda
            - Arn
          DeadLetterConfig:
            Arn: !GetAtt
              - UnprocessedAutoScalingEventQueue
              - Arn
          Id: Target0
          RetryPolicy:
            MaximumRetryAttempts: 15
  AutoScalingEventRuleTargetPermission:
```

```
    Type: 'AWS::Lambda::Permission'
    Properties:
      Action: 'lambda:InvokeFunction'
      FunctionName: !GetAtt
        - AutoScalingLambda
        - Arn
      Principal: events.amazonaws.com
      SourceArn: !GetAtt
        - AutoScalingEventRule
        - Arn
  AutoScalingLambdaServiceRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Action: 'sts:AssumeRole'
            Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
        Version: 2012-10-17
      ManagedPolicyArns:
        - !Join
          - ''
          - - 'arn:'
            - !Ref 'AWS::Partition'
            - ':iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
  AutoScalingLambdaServiceRoleDefaultPolicy:
    Type: 'AWS::IAM::Policy'
    Properties:
      PolicyDocument:
        Statement:
          - Action: 'autoscaling:SetDesiredCapacity'
            Effect: Allow
            Resource: '*'
        Version: 2012-10-17
      PolicyName: AutoScalingLambdaServiceRoleDefaultPolicy
      Roles:
        - !Ref AutoScalingLambdaServiceRole
  UnprocessedAutoScalingEventQueue:
    Type: 'AWS::SQS::Queue'
    Properties:
      QueueName: deadline-unprocessed-autoscaling-events
    UpdateReplacePolicy: Delete
    DeletionPolicy: Delete
```

```
  UnprocessedAutoScalingEventQueuePolicy:
    Type: 'AWS::SQS::QueuePolicy'
    Properties:
      PolicyDocument:
        Statement:
          - Action: 'sqs:SendMessage'
            Condition:
              ArnEquals:
                'aws:SourceArn': !GetAtt
                  - AutoScalingEventRule
                  - Arn
            Effect: Allow
            Principal:
              Service: events.amazonaws.com
            Resource: !GetAtt
              - UnprocessedAutoScalingEventQueue
              - Arn
        Version: 2012-10-17
      Queues:
        - !Ref UnprocessedAutoScalingEventQueue
```

Connect 客戶管理的叢集連線到授權端點

AWS 截止日期雲端 (截止日期雲端) 使用型授權伺服器會為選取的協力廠商產品提供隨選授權。這使您可以隨時隨地付款。您只會在使用的時間內變更。

只要截止日期雲端工作者可以與授權伺服器通訊，以期限雲端使用為基礎的授權伺服器就可以與任何叢集類型一起使用。這會在服務管理叢集中自動設定。只有客戶管理的機隊才需要此設定。

若要建立授權伺服器，您需要下列項目：

- 伺服器陣列 VPC 的安全性群組，可允許第三方授權的流量。
- 具有連結政策的 AWS Identity and Access Management (IAM) 角色，可存取截止日期雲端授權端點作業。

主題

- 步驟 1：建立安全性群組
- 步驟 2：設定授權端點
- 步驟 3：將轉譯應用程式 Connect 到端點

步驟 1：建立安全性群組

使用 Amazon VPC 主控台 (https://console.aws.amazon.com/vpc/) 為伺服器陣列的虛擬私人雲端建立安全群組。設定安全性群組以允許下列輸入規則：

- 歐特克瑪雅和阿諾德 — 2701-2702,
- 歐特克 3DS 最大-2704，TCP，IPv4
- 鑄造核彈 — 6101、TCP、IPv4
- 西德福斯胡迪尼，咒語和噶瑪 — 1715-1717, IPv4

每個輸入規則的來源都是叢集的 Worker 安全性群組。

如需有關建立安全群組的詳細資訊，請參閱 Amazon Virtual Private Cloud 使用者指南中的建立安全群組。

步驟 2：設定授權端點

授權端點可讓您存取協力廠商產品的授權伺服器。授權要求會傳送至授權端點。端點會將它們路由到適當的授權伺服器。授權伺服器會追蹤使用限制和權利。您建立的每個授權端點需支付費用。如需詳細資訊，請參閱 Amazon VPC 定價。

您可以使 AWS Command Line Interface 用適當的權限從中建立授權端點。如需建立授權端點的必要政策，請參閱允許建立授權端點的策略。

您可以使用 AWS CloudShell (https://console.aws.amazon.com/cloudshell/) 或任何其他 AWS CLI 環境，使用下列 AWS Command Line Interface 命令來設定授權端點。

1. 建立授權端點。將安全群組識別碼、子網路識別碼和 VPC ID 取代為您先前建立的值。如果您使用多個子網路，請使用空格分隔它們。

```
aws deadline create-license-endpoint \
  --security-group-id SECURITY_GROUP_ID \
  --subnet-ids SUBNET_ID1 SUBNET_ID2 \
  --vpc-id VPC_ID
```

2. 使用下列命令確認端點已成功建立。記住虛擬私人雲端端點的 DNS 名稱。

```
aws deadline get-license-endpoint \
  --license-endpoint-id LICENSE_ENDPOINT_ID
```

3. 檢視可用計量產品的清單：

```
aws deadline list-available-metered-products
```

4. 使用下列命令將計量產品新增至授權端點。

```
aws deadline put-metered-product \
--license-endpoint-id LICENSE_ENDPOINT_ID \
--product-id PRODUCT_ID
```

您可以使用以下`remove-metered-product`命令從授權端點移除產品：

```
aws deadline remove-metered-product \
  --license-endpoint-id LICENSE_ENDPOINT_ID \
  --productId PRODUCT_ID
```

您可以使用以下`delete-license-endpoint`命令刪除授權端點：

```
aws deadline delete-license-endpoint \
 --license-endpoint-id LICENSE_ENDPOINT_ID
```

步驟 3：將轉譯應用程式 Connect 到端點

設定授權端點後，應用程式使用的方式與使用協力廠商授權伺服器的方式相同。通常，您可以透過將環境變數或其他系統設定 (例如 Microsoft Windows 登錄機碼) 設定為授權伺服器連接埠和位址來規劃應用程式的授權伺服器。

若要取得授權端點 DNS 名稱，請使用下列 AWS CLI 命令。

```
aws deadline get-license-endpoint
```

或者，您也可以使用 Amazon 虛擬私人雲端主控台 (https://console.aws.amazon.com/vpc/) 來識別上一個步驟中由截止日期雲端 API 建立的 VPC 端點。

組態範例

Example — 歐特克瑪雅和阿諾德

將環境變數設定`ADSKFLEX_LICENSE_FILE`為：

```
2702@VPC_Endpoint_DNS_Name:2701@VPC_Endpoint_DNS_Name
```

> **Note**
>
> 對於 Windows Worker，請使用分號（；）而不是冒號（:）來分隔端點。

Example — 歐特克 3DS 最大

將環境變數設定ADSKFLEX_LICENSE_FILE為：

```
2704@VPC_Endpoint_DNS_Name
```

Example — 鑄造核彈

將環境變數設定foundry_LICENSE為若6101@*VPC_Endpoint_DNS_Name*要測試授權是否正常運作，您可以在終端機中執行 Nuke：

```
~/nuke/Nuke14.0v5/Nuke14.0 -x
```

Example — SiDEFX 胡迪尼, 咒語, 和噶瑪

執行以下命令：

```
/opt/hfs19.5.640/bin/hserver -S
 "http://VPC_Endpoint_DNS_Name:1715;http://VPC_Endpoint_DNS_Name:1716;http://
VPC_Endpoint_DNS_Name:1717;"
```

若要測試授權是否正常運作，您可以透過以下指令呈現 Houdini 場景：

```
/opt/hfs19.5.640/bin/hython ~/forpentest.hip -c "hou.node('/out/mantra1').render()"
```

管理期限雲端中的使用者

AWS 截止日期 Cloud 用 AWS IAM Identity Center 於管理使用者和群組。IAM 身分中心是雲端式單一登入服務，可與您的企業單一登入 (SSO) 供應商整合。透過整合，使用者可以使用其公司帳戶登入。

截止日期雲端預設會啟用 IAM 身分中心，而且必須設定和使用截止日期雲端。如需詳細資訊，請參閱管理您的身分識別來源。

您 AWS Organizations 的組織擁有者必須負責管理可存取截止日期雲端監視器的使用者和群組。您可以使用 IAM 身分中心或截止日期雲端主控台來建立和管理這些使用者和群組。如需詳細資訊，請參閱什麼是 AWS Organizations。

您可以使用 Deption Cloud 主控台建立和移除可使用監視器管理伺服器陣列、佇列和叢集的使用者和群組。當您將使用者新增到期限雲端時，他們必須先使用 IAM 身分中心重設密碼，才能取得存取權。

主題

- 管理監視器的使用者和群組
- 管理伺服器陣列、佇列和叢集的使用者和群組

管理監視器的使用者和群組

組 Organizations 擁有者可以使用截止日期雲端主控台來管理可存取截止日期雲端監視器的使用者和群組。您可以從現有的 IAM 身分中心使用者和群組中進行選擇，也可以從主控台新增使用者和群組。

1. 登入 AWS Management Console 並開啟截止日期雲端主控台。在主頁面的 [開始使用] 區段中，選擇 [設定期限雲端] 或 [前往儀表板]。
2. 在左側導覽窗格中，選擇 [使用者管理]。依預設，會選取「群組」(Groups) 標籤。

根據要採取的動作，選擇「群組」標籤或「使用者」標籤。

Monitor groups

建立群組

1. 選擇 Create group (建立群組)。
2. 輸入群組名稱。IAM 身分中心組織中的群組之間的名稱必須是唯一的。

若要移除群組

1. 選取要移除的群組。
2. 選擇移除。
3. 在確認對話方塊中，選擇 [移除群組]。

> **Note**
>
> 您要從 IAM 身分中心移除群組。群組成員無法再登入截止日期雲端或存取伺服器陣列資源。

Monitor users

新增使用者

1. 選擇 Users (使用者) 索引標籤。
2. 選擇 Add users (新增使用者)。
3. 輸入新使用者的名稱、電子郵件地址和使用者名稱。
4. 如有需要，請選擇一或多個要新增使用者的 IAM 身分中心群組。
5. 選擇 [傳送邀請]，傳送電子郵件給新使用者，其中包含加入 IAM 身分中心組織的指示。

移除使用者

1. 選取要從監視器移除的使用者。
2. 選擇移除。
3. 在確認對話方塊中，選擇 [移除使用者]。

> **Note**
>
> 您要從 IAM 身分中心移除使用者。使用者無法再登入到期限雲端監視器或存取伺服器陣列資源。

管理伺服器陣列、佇列和叢集的使用者和群組

1. 如果您尚未登入，請登入 AWS Management Console 並開啟截止日期雲端主控台。
2. 在左側導覽窗格中，選擇 [伺服器陣列和其他資源]。
3. 選取要管理的伺服器陣列。選擇伺服器陣列名稱以開啟詳細資料頁面。您可以使用搜尋列搜尋伺服器陣列。
4. 若要管理佇列或叢集，請選擇 [佇列] 或 [叢集] 索引標籤，然後選擇要管理的佇列或叢集。
5. 選擇存取管理索引標籤。依預設，會選取「群組」(Groups) 標籤。若要管理使用者，請將切換移至 [使用者]。

根據要採取的動作，選擇「群組」標籤或「使用者」標籤。

如需存取層級定義，請參閱權限。

Groups

若要新增群組

1. 選取群組切換。
2. 選擇 Add group (新增群組)。
3. 從下拉式清單中，選取要新增的群組。
4. 對於群組存取層級，請選擇下列其中一個選項：
   - Viewer (檢視者)
   - Contributor (作者群)
   - 經理
   - 擁有者
5. 選擇新增。

若要移除群組

1. 選取要移除的群組。
2. 選擇移除。
3. 在確認對話中，選擇 Remove (移除)。

Users

新增使用者

1. 若要新增使用者，請選擇 [新增使用者]。
2. 從下拉式清單中，選取要新增至伺服器陣列的使用者。
3. 針對使用者存取層級，選擇下列其中一個選項：
   - Viewer (檢視者)
   - Contributor (作者群)
   - 經理
   - 擁有者
4. 選擇新增。使用者會新增至您的伺服器陣列。

若要移除使用者

1. 選取要移除的使用者。
2. 在「移除」確認對話方塊中選擇「移除」。然後將使用者從選取的伺服器陣列中移除。

您也可以使用 IAM 身分中心主控台 https://console.aws.amazon.com/singlesignon/，為使用者和群組新增或移除伺服器陣列許可。

截止日期雲工作

工作是 AWS 截止日期雲端用來排程和執行可用背景工作的一組指示。建立工作時，您可以選擇要傳送工作的伺服器陣列和佇列。您也會提供 JSON 或 YAML 檔案，以提供背景工作處理的指示。截止日期雲接受按照打開職位描述（OpenJD）規範的 Job 模板來描述工作。有關更多信息，請參閱 GitHub 網站上的打開 Job 描述文檔。

工作包括：

- 步驟 — 定義要在 Worker 上執行的指令碼。步驟可能有需求，例如最小 Worker 記憶體或其他需要先完成的步驟。每個步驟都有一或多個工作。
- 任務 — 發送給 Worker 執行的工作單位。工作是步驟指令碼和指令碼中使用的參數 (例如影格編號) 的組合。當所有步驟的所有工作都完成時，工作即完成。
- 環境 — 設定和拆卸由多個步驟或工作共用的指示。

您可以使用下列任何一種方式建立工作：

- 使用截止日期雲端提交者。
- 建立工作服務包並使用截止日期雲端命令列介面 (截止日期 Cloud CLI)。
- 使用 AWS 軟體開發套件。
- 使用 AWS Command Line Interface （AWS CLI）。

提交者是數位內容建立 (DCC) 軟體的外掛程式，可在 DCC 軟體介面中管理建立工作。建立工作之後，您可以使用提交者將工作傳送至截止日期雲端進行處理。在幕後，提交者會建立描述工作的 OpenJD 工作範本。同時，它會將您的資產檔案上傳到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。為了減少傳送檔案所需的時間，只會將自上次上傳檔案後變更的檔案傳送至 Amazon S3。

若要建立自己的指令碼和管道以將工作提交至截止日期 Cloud，您可以使用截止日期 Cloud CLI、AWS SDK 或呼叫作業以建立、取得、檢視和列出工作。 AWS CLI 下列主題說明如何使用期限雲端 CLI。

截止日期 Cloud CLI 會隨著截止日期雲端提交者一起安裝。如需詳細資訊，請參閱 設定截止日期雲端提交者。

主題

- 使用截止日期雲端 CLI 提交工作

- 在截止日期雲中排程工作
- 截止日期雲端 CLI 中的 Job 狀態
- 修改期限雲端中的工作
- 截止日期雲端如何處理工
- 疑難排解期限雲端工

使用截止日期雲端 CLI 提交工作

若要使用截止日期雲端命令列介面 (截止日期 Cloud CLI) 提交工作，請使用`deadline bundle submit`命令。

工作會提交至佇列。如果您尚未設定伺服器陣列和佇列，請使用 Dependpoint Cloud 主控台 (https://console.aws.amazon.com/https://console.aws.amazon.com/deadlinecloud/home) 來設定伺服器陣列和佇列，以及查看伺服器陣列和佇列識別碼。如需詳細資訊，請參閱定義伺服器陣列詳細資料和定義佇列詳

若要設定期限 Cloud CLI 的預設伺服器陣列和佇列，請使用下列命令。當您設定預設值時，您可以使用截止日期 Cloud CLI 命令，而無需指定伺服器陣列或佇列。在下列範例中，取代*farmId*和*queueId*使用您自己的資訊：

```
deadline config set defaults.farm_id farmId
deadline config set defaults.queue_id queueId
```

若要指定工作中的步驟和工作，請建立 OpenJD 工作範本。如需詳細資訊，請參閱開啟 Job 說明規格 GitHub 儲存庫中的範本結構描述 [版本：2023-09]。

下列範例是 YAML 工作範本。它定義了一個工作，每步兩個步驟和五個任務。

```
name: Sample Job
specificationVersion: jobtemplate-2023-09
steps:
- name: Sample Step 1
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
```

```
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
- name: Sample Step 2
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
```

若要建立工作，請建立名為的新資料夾sample_job，然後將範本檔案儲存在新資料夾中template.yaml。您可以使用下列截止日期 Cloud CLI 命令來提交工作：

```
deadline bundle submit path/to/sample_job
```

來自命令的響應包含作業的標識符。記住 ID，以便稍後可以檢查工作的狀態。

```
Submitting to Queue: test-queue
Waiting for Job to be created...
Submitted job bundle:
   sample_job
Job creation completed successfully
jobId
```

提交工作時，您還可以使用其他選項。如需詳細資訊，請參閱 使用截止日期 Cloud CLI 提交工作的更多選項。

使用截止日期 Cloud CLI 提交工作的更多選項

deadline bundle submit截止日期 Cloud CLI 命令提供的選項可讓您用來指定工作的其他資訊。下列範例向您示範如何：

- 指定處理工作樣板時使用的參數。
- 將共用環境中的檔案和資料夾附加至工作。

- 設定工作取消前的作業失敗次數上限。
- 設定工作的重試次數上限。

任務參數

parameters此選項會在您建立工作時設定工作參數的值。工作範本會定義欄位，而選parameters項會設定值。參數可以具有預設值。如果為參數指定了值，則指定的值會覆寫預設值。

下列工作範本定義TestParameter欄位：

```
name: Sample Job With Job Parameter
parameterDefinitions:
- default: test
  name: TestParameter
  type: STRING
specificationVersion: jobtemplate-2023-09
steps:
- description: step description
  name: MyStep
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
```

下面的命令設置TestParameter為「你好 AWS」的值：

```
deadline bundle submit sample_job --parameter "TestParameter=Hello AWS"
```

儲存設定檔

儲存設定檔有助於在不同作業系統的 Worker 之間共用檔案。使用截止日期雲端主控台建立儲存設定檔。然後，使用storage-profile-id參數來使用儲存裝置設定檔。如需詳細資訊，請參閱 截止日期雲中的共享存儲。

若要設定工作提交的儲存區設定檔，請使用截止日期 Cloud CLI，使用下列命令來設定storage-profile-id組態參數：

```
deadline config set settings.storage_profile_id storageProfileId
```

失敗的工作上限

此選max-failed-tasks-count項可設定在整個工作失敗且標記所有剩餘工作之前，可以失敗的工作數目上限CANCELED。預設值為 100.

```
deadline bundle submit sample_job --max-failed-tasks-count 10
```

失敗的工作重試次數上限

此選max-retries-per-task項可設定工作失敗前重試的次數上限。當一個任務被重試，它被放在狀READY態。預設值為 5。

```
deadline bundle submit sample_job --max-retries-per-task 10
```

在截止日期雲中排程工作

建立任務之後， AWS Dependpoint Cloud 會排程在與佇列相關聯的一或多個叢集上進行處理。處理特定作業的叢集是根據針對叢集設定的功能以及特定步驟的主機需求來選擇。

工作會以最佳優先順序排程，從最高到最低。當兩個工作具有相同的優先順序時，會先排定最舊的工作。

下列各節提供排定工作的程序詳細資訊。

判斷車隊相容性

建立工作後，截止日期 Cloud 會根據與工作提交至的佇列相關聯的叢集功能，檢查工作中每個步驟的主機需求。如果叢集符合主機需求，工作就會進入READY狀態。

如果作業中的任何步驟具有與佇列相關聯的叢集無法滿足的需求，則步驟的狀態會設定為NOT_COMPATIBLE。此外，工作中的其餘步驟也會取消。

叢集的功能是在車隊層級設定。即使叢集中的工作人員符合工作的需求，如果該工作的叢集不符合工作的需求，也不會從該工作指派工作的任務。

下列工作範本具有指定步驟主機需求的步驟：

```
name: Sample Job With Host Requirements
specificationVersion: jobtemplate-2023-09
steps:
- name: Step 1
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
  hostRequirements:
    amounts:
    # Capabilities starting with "amount." are amount capabilities. If they start with
 "amount.worker.",
    # they are defined by the OpenJD specification. Other names are free for custom
 usage.
    - name: amount.worker.vcpu
      min: 4
      max: 8
    attributes:
    - name: attr.worker.os.family
      anyOf:
      - linux
```

可將此工作排程至具有下列功能的叢集：

```
{
    "vCpuCount": {"min": 4, "max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
}
```

無法將此工作排程到具有下列任何功能的叢集：

```
{
    "vCpuCount": {"min": 4},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
```

```
}
    The vCpuCount has no maximum, so it exceeds the maximum vCPU host requirement.

{
    "vCpuCount": {"max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
}
    The vCpuCount has no minimum, so it doesn't satisfy the minimum vCPU host
 requirement.

{
    "vCpuCount": {"min": 4, "max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "windows",
    "cpuArchitectureType": "x86_64"
}
    The osFamily doesn't match.
```

機隊擴展

將任務指派給相容的服務受管理叢集時，會 auto 調整叢集。叢集中的工作者數量會根據可供叢集執行的工作數量而變動。

將任務指派給客戶管理的叢集時，Worker 可能已存在，或者可以使用事件型 auto 調整來建立。如需詳細資訊，請參閱 Amazon EC2 auto 擴展使用者指南中的使用 EventBridge來處理自動擴展事件。

工作階段

工作中的工作會分成一或多個工作階段。工作者會執行工作階段來設定環境、執行工作，然後拆除環境。每個工作階段都是由 Worker 必須執行的一或多個動作組成。

當 Worker 完成區段動作時，可將其他工作階段動作傳送給 Worker。Worker 會重複使用工作階段中的現有環境和工作附件，以更有效率地完成工作。

Job 附件是由您使用的提交者建立，做為截止日期 Cloud CLI 工作服務包的一部分。您也可以使用`create-job` AWS CLI 指令的`--attachments`選項來建立工作附件。環境分為兩個位置定義：附加至特定佇列的佇列環境，以及工作範本中定義的作業步驟環境。

有四種工作階段動作類型：

- syncInputJobAttachments— 將輸入工作附件下載至 Worker。
- envEnter— 針對環境執行onEnter動作。
- taskRun— 執onRun行任務的動作。
- envExit— 針對環境執行onExit動作。

下列工作範本具有步驟環境。它有一個定onEnter義來設置步驟環境，定onRun義要運行的任務的定onExit義，以及拆除步驟環境的定義。為此工作建立的工作階段將包括envEnter動作、一或多個taskRun動作，然後是envExit動作。

```
name: Sample Job with Maya Environment
specificationVersion: jobtemplate-2023-09
steps:
- name: Maya Step
  stepEnvironments:
  - name: Maya
    description: Runs Maya in the background.
    script:
      embeddedFiles:
      - name: initData
        filename: init-data.yaml
        type: TEXT
        data: |
          scene_file: MyAwesomeSceneFile
          renderer: arnold
          camera: persp
      actions:
        onEnter:
          command: MayaAdaptor
          args:
          - daemon
          - start
          - --init-data
          - file://{{Env.File.initData}}
        onExit:
          command: MayaAdaptor
          args:
          - daemon
          - stop
  parameterSpace:
    taskParameterDefinitions:
    - name: Frame
```

```
      range: 1-5
      type: INT
  script:
    embeddedFiles:
    - name: runData
      filename: run-data.yaml
      type: TEXT
      data: |
        frame: {{Task.Param.Frame}}
    actions:
      onRun:
        command: MayaAdaptor
        args:
        - daemon
        - run
        - --run-data
        - file://{{ Task.File.runData }}
```

步驟相依性

截止日期雲端支援定義步驟之間的相依性，讓一個步驟會等到另一個步驟完成後再開始。您可以為一個步驟定義多個相依性。在其所有相依性完成之前，不會排程具有相依性的步驟。

如果工作範本定義循環相依性，則會拒絕工作，並將工作狀態設定為CREATE_FAILED。

下列工作範本會建立具有兩個步驟的工作。 StepB取決於StepA. StepB只有在StepA成功完成後才會執行。

建立工作之後，處StepA於狀READY態且處StepB於狀PENDING態。StepA完成後，StepB移至狀READY態。如果StepA失敗或取消，則StepAStepB會移至狀CANCELED態。

您可以在多個步驟上設定相依性。例如，如果StepC取決於兩者 StepAStepB，StepC則在其他兩個步驟完成之前不會啟動。

```
name: Step-Step Dependency Test
specificationVersion: 'jobtemplate-2023-09'
steps:
- name: A
  script:
    actions:
      onRun:
        command: bash
        args: ['{{ Task.File.run }}']
```

```
    embeddedFiles:
      - name: run
        type: TEXT
        data: |
          #!/bin/env bash

          set -euo pipefail

          sleep 1
          echo Task A Done!
- name: B
  dependencies:
  - dependsOn: A # This means Step B depends on Step A
  script:
    actions:
      onRun:
        command: bash
        args: ['{{ Task.File.run }}']
    embeddedFiles:
      - name: run
        type: TEXT
        data: |
          #!/bin/env bash

          set -euo pipefail

          sleep 1
          echo Task B Done!
```

截止日期雲端 CLI 中的 Job 狀態

本主題說明如何使用 AWS 截止日期雲端命令列介面 (截止日期 Cloud CLI) 來檢視工作或步驟的狀態。如果您想要使用截止日期雲端監視器來檢視工作或步驟的狀態，請參閱在截止日期雲端中檢視和管理工作、步驟和工作。

您可以使用`deadline job get --job-id`截止日期 Cloud CLI 命令查看工作的狀態。對指令的回應包括工作或步驟的狀態，以及每個處理狀態中的作業數目。

當您第一次提交工作時，狀態為`CREATE_IN_PROGRESS`。如果工作通過驗證檢查，其狀態會變更為`CREATE_COMPLETE`。如果不是，狀態會變更為`CREATE_FAILED`。

工作可能會失敗驗證檢查的一些可能原因包括：

- 工作範本不遵循 OpenJD 規範。
- 工作包含太多步驟。
- 工作包含太多的工作總數。

若要查看工作中步驟和工作數目上限的配額，請使用 Service Quotas 主控台。如需詳細資訊，請參閱的配額 Deadline Cloud。

也可能存在內部服務錯誤，導致無法建立工作。如果發生這種情況，工作的狀態碼為INTERNAL_ERROR，狀態訊息欄位會提供更詳細的說明。

使用下列截止日期 Cloud CLI 命令來檢視工作的詳細資料。在下列範例中，請以您自己*jobID*的資訊取代：

```
deadline job get --job-id jobId
```

來自deadline job get命令的響應如下：

```
jobId: jobId
name: Sample Job
lifecycleStatus: CREATE_COMPLETE
lifecycleStatusMessage: Job creation completed successfully
priority: 50
createdAt: 2024-03-26 18:11:19.065000+00:00
createdBy: Test User
startedAt: 2024-03-26 18:12:50.710000+00:00
taskRunStatus: STARTING
taskRunStatusCounts:
  PENDING: 0
  READY: 5
  RUNNING: 0
  ASSIGNED: 0
  STARTING: 0
  SCHEDULED: 0
  INTERRUPTING: 0
  SUSPENDED: 0
  CANCELED: 0
  FAILED: 0
  SUCCEEDED: 0
  NOT_COMPATIBLE: 0
```

```
maxFailedTasksCount: 100
maxRetriesPerTask: 5
```

工作或步驟中的每個工作都有一個狀態。工作狀態會結合在一起，以提供工作和步驟的整體狀態。回應欄位中會報告每個狀態的taskRunStatusCounts工作數目。

工作或步驟的狀態取決於其工作的狀態。狀態由具有這些狀態的工作依序決定。步驟狀態與工作狀態相同。

下列清單說明這些狀態：

NOT_COMPATIBLE

這項工作與伺服器陣列不相容，因為沒有叢集可以完成工作中的其中一項工作。

RUNNING

一或多個 Worker 正在執行工作中的工作。只要至少有一個正在執行的工作，就會標示工作RUNNING。

ASSIGNED

一個或多個工作者被指派工作中的任務作為他們的下一個動作。已設定環境 (如果有的話)。

STARTING

一或多個 Worker 正在設定執行工作的環境。

SCHEDULED

工作的任務會排定在一或多個 Worker 上做為工作者的下一個動作。

READY

工作至少有一個作業已準備好可以處理。

INTERRUPTING

工作中至少有一個工作正在中斷。手動更新工作狀態時，可能會發生中斷。這也可能是因為亞馬遜彈性運算雲 (Amazon EC2) 現貨價格變化而發生中斷。

FAILED

工作中有一或多個工作未成功完成。

CANCELED

工作中的一個或多個任務已被取消。

SUSPENDED

工作中至少有一個工作已暫停。

PENDING

工作中的任務正在等待另一個資源的可用性。

SUCCEEDED

已成功處理工作中的所有工作。

修改期限雲端中的工作

您可以使用下列 AWS Command Line Interface (AWS CLI) update 指令來修改工作的組態，或設定工作、步驟或工作的目標狀態：

- aws deadline update-job
- aws deadline update-step
- aws deadline update-task

在下列update指令範例中，請*user input placeholder*使用您自己的資訊來取代每個指令。

您也可以使用截止日期雲端監視器來修改工作的組態。如需詳細資訊，請參閱 在截止日期雲端中檢視和管理工作、步驟和工作。

Example — 重新搜尋工作

除非有步驟相依性，否則工作中的所有工作都會切換到READY狀態。具有依賴關係的步驟切換到READY或PENDING恢復。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status PENDING
```

Example — 取消工作

工作中沒有狀態SUCCEEDED或已標記的所FAILED有工作CANCELED。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status CANCELED
```

Example — 將工作標記為失敗

工作中具有該狀態的所有工作SUCCEEDED都會保持不變。所有其他任務都會被標記FAILED。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status FAILED
```

Example — 標記工作成功

工作中的所有工作都會移至該SUCCEEDED狀態。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status SUCCEEDED
```

Example — 暫停工作

、或FAILED狀態中工作SUCCEEDED中CANCELED的工作不會變更。所有其他任務都會被標記SUSPENDED。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status SUSPENDED
```

Example — 變更工作的優先順序

更新工作的優先順序，以變更其排程的順序。優先順序較高的工作通常會先排定。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--priority 100
```

Example — 更改允許的失敗任務的數量

在取消剩餘工作之前，更新工作可以擁有的失敗工作數目上限。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--max-failed-tasks-count 200
```

Example — 變更允許的任務重試次數

在工作失敗之前，更新工作的重試次數上限。已達到重試次數上限的工作，除非此值增加，否則無法重新計算重試次數。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--max-retries-per-task 10
```

Example — 存檔工作

將工作的生命週期狀態更新為ARCHIVED。封存的工作無法排程或修改。您只能封存處於FAILED、CANCELEDSUCCEEDED、或SUSPENDED狀態的工作。

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--lifecycle-status ARCHIVED
```

Example — 重新查詢一個步驟

除非有步驟相依性，否則步驟中的所有工作都會切換到READY狀態。具有相依性的步驟中的工作會切換至READY或PENDING，且工作會還原。

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status PENDING
```

Example — 取消步驟

步驟中沒有狀態SUCCEEDED或已標記的所FAILED有工作CANCELED。

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status CANCELED
```

Example — 標記步驟失敗

步驟中具有狀態的所有工作SUCCEEDED都會保持不變。所有其他任務都會被標記FAILED。

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status FAILED
```

Example — 標記一個步驟成功

會標記步驟中的所有工作SUCCEEDED。

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
```

```
--step-id stepID \
--target-task-run-status SUCCEEDED
```

Example — 暫停步驟

、或FAILED狀態中步驟SUCCEEDED中CANCELED的工作不會變更。所有其他任務都會被標記SUSPENDED。

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status SUSPENDED
```

Example — 變更任務的狀態

當您使用update-task截止日期 Cloud CLI 命令時，工作會切換到指定的狀態。

```
aws deadline update-task \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--task-id taskID \
--target-task-run-status SUCCEEDED | SUSPENDED | CANCELED | FAILED | PENDING
```

截止日期雲端如何處理工

為了處理 Job， AWS 截止日期雲使用「打開職位描述」（OpenJD）工作模板來確定所需的資源。截止日期雲端會從與佇列相關聯的叢集中選取適合的工作者來執行某個步驟。所選 Worker 符合步驟所需的所有能力屬性。

接著，截止日期雲端會傳送指示給 Worker，以設定步驟的工作階段。步驟所需的軟體必須位於 Worker 執行個體上，才能執行工作。如果叢集的擴展設定有容量，服務可以在多個 Worker 上開啟工作階段。

您可以在 Amazon Machine Image (AMI) 中設定軟體，或者您的 Worker 可以在執行階段從儲存庫或套件管理員載入軟體。您可以使用佇列、工作或步驟環境來部署您偏好的軟體。

截止日期雲端服務會使用 OpenJD 範本來決定工作所需的步驟，以及每個步驟所需的工作。某些步驟與其他步驟有相依性，因此截止日期雲端會決定完成這些步驟的順序。然後，截止日期雲端會將每

個步驟的工作傳送給工作者處理。當工作完成時，服務會在相同的工作階段中傳送另一個工作，或者 Worker 可以啟動新的工作階段。

您可以在截止日期雲端監視器、截止日期雲端命令列介面 (截止日期 Cloud CLI) 或 AWS CLI. 如需使用監視器的詳細資訊，請參閱使用截止日期雲端監視器。如需有關使用期限雲端 CLI 的詳細資訊，請參閱截止日期雲端 CLI 中的 Job 狀態。

完成每個步驟中的所有工作之後，工作就會完成，輸出就可以下載到您的工作站。即使工作未完成，也可以下載每個步驟和已完成工作的輸出。

截止日期雲端會在工作提交後 120 天移除。移除工作時，也會移除與該工作相關聯的所有步驟和工作。如果您需要重新執行工作，請再次送出該工作的 OpenJD 範本。

疑難排解期限雲端工

如需「 AWS 截止日期雲端」中工作的常見問題的相關資訊，請參閱下列主題。

主題

為什麼我的工作建立失敗？

工作可能會失敗驗證檢查的一些可能原因包括：

- 工作範本不遵循 OpenJD 規範。
- 工作包含太多步驟。
- 工作包含太多的工作總數。
- 發生內部服務錯誤，造成工作無法建立。

若要查看工作中步驟和工作數目上限的配額，請使用 Service Quotas 主控台。如需詳細資訊，請參閱的配額 Deadline Cloud。

為什麼我的工作不兼容？

工作與佇列不相容的常見原因包括：

- 沒有任何叢集與提交工作的目標佇列相關聯。開啟截止日期雲端監視器，並檢查佇列是否有相關聯的叢集。如需如何檢視佇列的相關資訊，請參閱在期限雲端中檢視佇列和車隊詳細資料。
- 工作具有與佇列相關聯的任何叢集都不滿足的主機需求。若要檢查，請比較工作範本中的`hostRequirements`項目與伺服器陣列中叢集的組態。請確定其中一個叢集符合主機需求。如需叢集相容性的詳細資訊，請參閱判斷車隊相容性。若要檢視叢集組態，請參閱在期限雲端中檢視佇列和車隊詳細資料。

為什麼我的工作已經準備好了？

您的工作似乎停留在該`READY`狀態的可能原因包括：

- 與佇列相關聯之叢集的最大背景工作者計數設為零。若要檢查，請參閱在期限雲端中檢視佇列和車隊詳細資料。
- 佇列中有較高優先順序的工作。若要檢查，請參閱在期限雲端中檢視佇列和車隊詳細資料。
- 對於客戶管理的機隊，請檢查 auto 擴展配置。如需詳細資訊，請參閱 使用截止日期雲端擴展建議功能自動擴展 Amazon EC2 叢集。

為什麼我的工作失敗了？

工作失敗的原因有很多。若要搜尋問題，請開啟截止日期雲端監視器，然後選擇失敗的工作。選擇失敗的工作，然後檢視該工作的記錄檔。如需說明，請參閱在截止日期雲中查看日誌。

- 如果您看到授權錯誤，或是因為軟體沒有有效的授權而出現浮水印，請確定 Worker 可以連線至所需的授權伺服器。如需詳細資訊，請參閱 Connect 客戶管理的叢集連線到授權端點。

為什麼我的步驟是待處理的？

當一個或多個相依性尚未完成時，步驟可能會保持在`PENDING`狀態。您可以使用期限雲端監視器來檢查相依性的狀態。如需說明，請參閱檢視截止日期雲端中的步驟。

截止日期雲端的檔案儲存

Worker 必須能夠存取包含處理工作所需之輸入檔案的儲存位置，以及儲存輸出的位置。 AWS 截止日期雲端提供兩個儲存位置選項：

- 透過工作附件，截止日期雲端會在工作站和截止日期雲端工作者之間來回傳輸工作的輸入和輸出檔案。為了啟用文件傳輸，截止日期雲使用亞馬遜簡單存儲服務（Amazon S3）存儲桶在您的 AWS 帳戶.

  當您將工作附件與服務管理的叢集搭配使用時，您可以在虛擬私人網路 (VPN) 中設定虛擬檔案系統 (VFS)。然後 Worker 只能在需要時載入檔案。
- 使用共用儲存裝置時，您可以使用作業系統的檔案共用來提供檔案存取權。

  使用跨平台共用儲存裝置時，您可以建立儲存裝置設定檔，讓 Worker 可以將路徑對應到兩個不同作業系統之間的檔案。

主題

截止日期雲中的 Job 附件

Job 附件可讓您在工作站和 AWS 截止日期雲端之間來回傳輸檔案。使用任務附件，您無需為檔案手動設定 Amazon S3 儲存貯體。相反地，當您使用「截止日期雲端」主控台建立佇列時，您可以為工作附件選擇值區。

第一次將工作提交到期限雲端時，該工作的所有檔案都會傳輸到截止日期雲端。對於後續提交，只會傳輸已變更的檔案，以節省時間和頻寬。

處理完成後，您可以從工作詳細資訊頁面或使用「截止日期 Cloud CLI」`deadline job download-output` 指令下載結果。

您可以針對多個佇列使用相同的 S3 儲存貯體。為每個佇列設定不同的根前置詞，以組織值區中的附件。

使用主控台建立佇列時，您可以選擇現有的 AWS Identity and Access Management (IAM) 角色，也可以讓主控台建立新角色。如果主控台建立角色，它會設定存取為佇列指定之值區的權限。如果您選擇現有角色，則必須授與角色存取 S3 儲存貯體的權限。

工作附件 S3 儲存貯體的加密

依預設，Job 附件檔案會在 S3 儲存貯體中自動加密。這種方法有助於保護您的信息免受未經授權的訪您無需執行任何操作即可使用截止日期雲提供的密鑰對文件進行加密。如需詳細資訊，請參閱 Amazon S3 現在會自動加密 Amazon S3 使用者指南中的所有新物件。

您可以使用自己的客戶受管 AWS Key Management Service 金鑰來加密包含任務附件的 S3 儲存貯體。若要這麼做，您必須修改與值區相關聯之佇列的 IAM 角色，以允許存取 AWS KMS key.

開啟佇列角色的 IAM 政策編輯器

1. 登入 AWS Management Console 並開啟截止日期雲端主控台。從主頁面的 [開始使用] 區段中，選擇 [檢視伺服器陣列]。
2. 從伺服器陣列清單中，選擇包含要修改之佇列的伺服器陣列。
3. 從佇列清單中選擇要修改的佇列。
4. 在佇列詳細資料區段中，選擇服務角色以開啟服務角色的 IAM 主控台。

接下來，完成下列程序。

若要更新具有下列權限的角色原則 AWS KMS

1. 從權限原則清單中，選擇角色的策略。
2. 在 [此原則中定義的權限] 區段中，選擇 [編輯]。
3. 選擇 [新增陳述式]。
4. 將下列原則複製並貼到編輯器中。將*Region*、*accountID*和變更*keyID*為您自己的值。

```
{
   "Effect": "Allow",
   "Action": [
       "kms:Decrypt",
       "kms:DescribeKey",
       "kms:GenerateDataKey"
   ],
   "Resource": [
       "arn:aws:kms:Region:accountID:key/keyID"
```

```
    ]
}
```

5. 選擇下一步。
6. 檢閱原則的變更，然後在您滿意時選擇 [儲存變更]。

管理 S3 儲存貯體中的工作附件

截止日期雲端會將工作所需的工作附件檔案儲存在 S3 儲存貯體中。這些檔案會隨著時間累積，導致 Amazon S3 成本增加。若要降低成本，您可以將 S3 生命週期組態套用至 S3 儲存貯體。此設定可以自動刪除值區中的檔案。由於 S3 儲存貯體位於您的帳戶中，因此您可以隨時選擇修改或移除 S3 生命週期組態。如需詳細資訊，請參閱 Amazon S3 使用者指南中的 S3 生命週期組態範例。

如需更精細的 S3 儲存貯體管理解決方案，您 AWS 帳戶 可以根據上次存取 S3 儲存貯體中的物件設定為使物件過期。如需詳細資訊，請參閱 AWS 架構部落格上根據上次存取日期將 Amazon S3 物件過期，以降低成本。

截止日期雲虛擬文件系統

AWS 截止日期雲端中對任務附件進行虛擬檔案系統支援，可讓工作人員上的用戶端軟體直接與 Amazon 簡單儲存服務通訊。Worker 只能在需要時載入檔案，而不是在處理前下載所有檔案。檔案儲存在本機。這種方法可避免下載多次使用多次的資產。工作完成後，會移除所有檔案。

- 虛擬檔案系統可大幅提升特定工作設定檔的效能。一般而言，擁有較大員工叢集的總檔案較小的子集顯示最大的效益。少量檔案的工作程式較少，其處理時間大致相等。
- 虛擬檔案系統支援僅適用於服務管理叢集中的Linux工作者。
- 截止日期雲端虛擬檔案系統支援下列作業，但不符合 POSIX 標準：
  - 檔案`createdelete`、`open`、`close`、`read`、`writeappend`、`truncate`、`rename`、`move`、`copy`、`st`和 `falloc`
  - 目錄`createdeleterename`、`move`、`copy`、和 `stat`
- 虛擬檔案系統的設計目的是在您的任務只存取部分大型資料集時減少資料傳輸並改善效能，而且並未針對所有工作負載進行最佳化。您應該在執行生產作業之前測試工作負載。

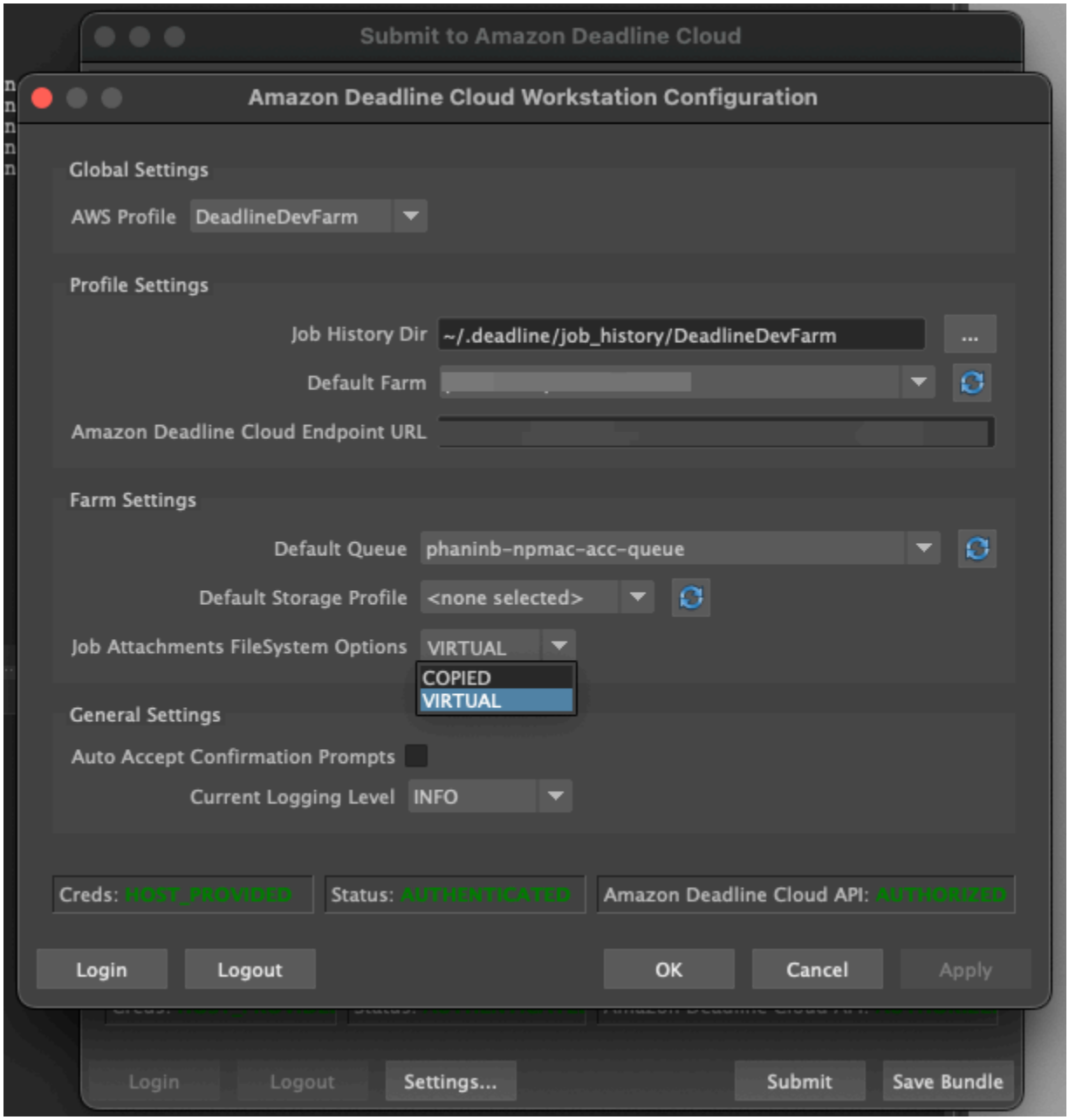
啟用 VFS 支援

每個工作都會啟用虛擬檔案系統支援 (VFS)。在下列情況下，工作會退回至預設的工作附件架構：

- Worker 執行個體設定檔不支援虛擬檔案系統。
- 問題阻止啟動虛擬文件系統進程。
- 虛擬檔案系統無法掛載。

使用提交者啟用虛擬檔案系統支援

1. 提交工作時，請選擇「設定」按鈕以開啟「AWS 截止日期雲端工作站」設定面板。
2. 從 Job 附件檔案系統選項下拉清單中，選擇虛擬。



3. 若要儲存變更，請選擇 [確定]。

若要啟用虛擬檔案系統支援 AWS CLI

- 當您送出儲存的工作時，請使用下列指令：

```
deadline bundle submit-job --job-attachments-file-system VIRTUAL
```

若要確認虛擬檔案系統是否已成功啟動特定任務，請在 Amazon Logs 中檢閱您的 CloudWatch 日誌。尋找下列訊息：

```
Using mount_point mount_point
Launching vfs with command command
Launched vfs as pid PID number
```

如果記錄檔包含下列訊息，則會停用虛擬檔案系統支援：

```
Virtual File System not found, falling back to COPIED for JobAttachmentsFileSystem.
```

虛擬檔案系統支援的疑難

您可以使用截止日期雲端監視器來檢視虛擬檔案系統的記錄。如需說明，請參閱在截止日期雲中查看日誌。

虛擬檔案系統記錄也會傳送至與 Worker 代理程式輸出共用之佇列相關聯的 CloudWatch 記錄群組。

截止日期雲中的共享存儲

若要使用共用儲存空間，Worker 會使用作業系統檔案共用系統來存取共用儲存空間，以便輸入和輸出工作。

您用來共用檔案的實際方法取決於您的作業系統，以及您在網路上實作共用儲存裝置的方式。您必須負責設定檔案共用的方式，並確保其符合您的需求。

如果您使用跨系統檔案共用解決方案，您可以使用儲存設定檔來對應Linux和檔案系統之間的Windows檔案位置。

期限雲端中的儲存設定檔

儲存區設定檔可讓您使用跨平台共用儲存區來設定伺服器陣列。儲存裝置設定檔會跨作業系統對應路徑，以便在 Worker 上處理的作業系統與提交工作站不同的作業系統。

當您在工作站與工作站之間混合使用作業系統的客戶管理叢集時，就需要儲存裝置設定檔。服務管理的叢集不支援儲存區設定檔。

建立儲存區設定檔之後，您必須授與使用該設定檔之佇列和叢集的存取權。

建立儲存裝置設定檔

1. 開啟截止日期雲端主控台。
2. 從 [開始使用] 中，選擇 [移至期限雲端儀表板]。
3. 選擇伺服器陣列，然後選擇 [儲存區設定檔] 索引標籤。
4. 選擇 [建立儲存設定檔]
5. 從下拉式清單中選擇作業系統。
6. 提供設定檔的「名稱」。清晰的名稱可協助您選擇送出工作時要使用的儲存裝置設定檔。
7. 針對「路徑」名稱，輸入您提交工作的工作站上工作資料的根位置。
8. 選擇儲存類型：
   - 「本端」是指 Worker 與工作站之間未共用的檔案位置。它們會作為工作附件上載。
   - 「共用」是指 Worker 與工作站之間共用的儲存區。共用儲存裝置中的檔案不會作為工作附件上傳。
9. 提供檔案系統位置路徑。這是工作資料的根目錄。
10. 選擇建立。

建立儲存區設定檔之後，您必須修改佇列和客戶管理的叢集，才能使用新的設定檔。若要允許存取儲存裝置設定檔，請在完成前一個程序後使用下列程序。

允許佇列和客戶管理的叢集使用儲存設定檔

1. 選擇 [佇列] 或 [叢集] 索引標籤。
2. 選擇要修改的佇列或叢集。
3. 選擇 [修改儲存設定檔]
4. 選取要允許的儲存裝置設定檔，以及該設定檔中的檔案系統位置。
5. 選擇儲存變更。

管理截止日期雲端的預算和用量

AWS 截止日期雲端預算管理員和用量總管是成本管理工具，可根據有關成本變數的可用資訊，提供使用截止日期雲端的大約成本。成本管理工具無法保證您實際使用截止日期雲端和其他 AWS 服務所欠的金額。

為了協助您管理截止日期雲端的成本，您可以使用下列功能：

- 預算管理員 — 使用截止日期雲端預算管理器，您可以建立和編輯預算以協助管理專案成本。
- 使用量總管 — 透過「截止日期雲端使用總管」，您可以檢視使用的 AWS 資源數量以及這些資源的預估成本。

成本假設

截止日期雲端成本管理工具所使用的基本計算方式為：

```
Cost per job =
    (CMF run time x CMF compute rate) +
    (SMF run time x SMF compute rate) +
    (License run time x license rate)
```

- 執行時間是工作中所有工作的總和，從開始時間到結束時間。
- 運算費率取決於服務受管理叢集的AWS 截止日期雲端定價。對於客戶管理的叢集，運算費率估計為每個工作人員小時 1 美元。
- 授權費率由截止日期雲端基本授權價格決定。不包括其他等級。如需授權定價的詳細資訊，請參閱AWS 截止日期雲端定價。

截止日期雲端成本管理工具的成本估算可能與您的實際成本有很多不同，原因有很多。常見原因包括：

- 客戶擁有的資源及其定價。您可以選擇從內部部署 AWS 或其他雲端供應商取得自己的資源，也可以從外部部署或外部取得資源。不會計算這些資源的實際成本。
- 閒置工人成本。對於執行個體計數下限大於零的叢集，閒置 Worker 不會計算在計算中。
- 促銷積分、折扣和自訂訂價協議。成本管理工具不會考慮促銷抵免額、私人定價協議或其他折扣。您可能有資格獲得不屬於預估的其他折扣。
- 資產儲存。資產儲存不包含在成本和使用量預估中。

- 價格變化。 AWS 提供大多數服務的 pay-as-you-go 定價。價格可能會隨著時間而變化。成本管理工具使用公共盟友可用的最多 up-to-date 價格，但更改後可能會有延遲。
- 稅收。成本管理工具不包括適用於我們購買服務的稅金。
- 四捨五入。成本管理工具執行定價資料的數學四捨五入。
- 貨幣。成本估算以美元計算。全球匯率隨時間而變化。如果您根據目前的匯率將估算值轉換為不同的貨幣，匯率的變更會影響估算值。
- 外部授權。如果您選擇使用預先購買的授權 (攜帶自己的授權)，則 Depate Cloud 成本管理工具無法將此費用列入考量。

使用截止日期雲端預算管理員

截止日期雲端預算管理員可協助您控制指定資源 (例如佇列、叢集或伺服器陣列) 的支出。您可以建立預算金額和限制，並設定自動化動作，協助減少或停止預算的額外支出。

以下各節提供使用截止日期雲端預算管理員的步驟。

主題

先決條件

若要使用截止日期雲端預算管理員，您必須具有OWNER存取層級。若要授與OWNER權限，請遵循中的步驟管理期限雲端中的使用者。

訪問預算管理器

若要存取截止日期雲端預算管理員，請遵循下列程序。

1. 登入 AWS Management Console 並開啟截止日期雲端主控台。

2. 選擇 [檢視農場]。
3. 找出您要取得相關資訊的伺服器陣列，然後選擇 [管理工作]。截止日期雲端監視器會在新標籤中開啟。
4. 在截止日期雲端監視器的左側導覽窗格中，選擇 [預算]。

預算管理程式彙總頁面會顯示有效與失效預算的清單：

- 作用中預算會根據選取的資源 (佇列) 追蹤。
- 失效預算已過期或由使用者取消，而且不再追蹤此預算限制的成本。

選擇預算後，預算摘要頁面會包含預算的基本資訊。提供的資訊包括預算名稱、狀態、資源、剩餘百分比、剩餘金額、總預算、開始日期及結束日期。

建立預算

若要建立預算，請遵循下列步驟。

1. 如果您尚未登入，請登入 AWS Management Console，開啟截止日期 Cloud 主控台，選擇伺服器陣列，然後選擇 [管理工作]。
2. 在「預算管理員」頁面中，選擇「建立預算」。
3. 在詳細資訊區段中，輸入預算的「預算」名稱。
4. (選擇性) 在說明欄位中，輸入預算的明確簡短說明。
5. 從資源中選擇佇列下拉式清單，以搜尋並選取您要建立預算的佇列。
6. 若為「期間」，請完成下列步驟來設定預算的開始與結束日期：

    a. 在開始日期中，以 YYYY/MM/DD 格式輸入預算追蹤的第一個日期，或選擇行事曆圖示並選取日期。

    預設開始日期為建立預算的日期。

    b. 在結束日期中，以 YYYY/MM/DD 格式輸入預算追蹤的最後日期，或選擇行事曆圖示並選取日期。

    預設結束日期為開始日期起 120 天。
7. 在「預算金額」中，輸入預算的金額。
8. (選擇性) 建議您建立限制警示。在「限制作業」區段中，您可以導入在預算中保留特定金額時所發生的自動化作業。若要執行此動作，請執行下列步驟。

a. 選擇 [新增動作]。

b. 在剩餘金額中，輸入您要開始動作的金額。

c. 在「動作」下拉式清單中，選擇您要的動作。動作包括：

- 完成目前工作後停止 — 當達到臨界值金額時，目前正在執行的所有工作都會繼續執行 (並產生成本)，直到完成為止。
- 立即停止工作 — 滿足閾值金額時，所有工作將立即取消。

d. 若要建立其他限制警示，請選擇 [新增動作]，然後重複前兩個步驟。

9. 選擇「建立預算」。便會顯示預算管理程式頁面。新建立的預算會顯示在「有效預算」頁標中。

檢視預算

建立預算後，您可以在「預算管理程式」頁面上檢視預算。從那裡，您可以查看預算的總金額和分配給特定預算的總成本。

若要檢視預算，請遵循下列步驟。

1. 如果您尚未登入，請登入 AWS Management Console，開啟截止日期 Cloud 主控台，選擇伺服器陣列，然後選擇 [管理工作]。
2. 從左側瀏覽窗格中選擇「預算」。便會顯示「預算管理程式」頁
3. 若要檢視有效預算，請選擇「有效預算」頁標，然後選擇您要檢視的預算名稱。預算詳細資訊頁面隨即出現。
4. 若要檢視到期預算的預算明細，請選擇「失效」預算頁標。然後，選擇您要查看的預算名稱。預算詳細資訊頁面隨即出現。

編輯預算

您可以編輯任何有效的預算。若要編輯有效預算，請遵循下列步驟。

1. 如果您尚未登入，請登入 AWS Management Console，開啟截止日期 Cloud 主控台，選擇伺服器陣列，然後選擇 [管理工作]。
2. 從「預算管理程式」頁面的「有效預算」頁標中，選擇您要編輯之預算旁邊的按鈕。
3. 從右上角的「作業」下拉式功能表中，選取「編輯預算」。
4. 進行您要的變更，然後選擇 [更新預算]。

停用預算

您可以停用任何有效的預算。停用預算會將其狀態從「有效」變更為「無效」。停用預算後，就不會再追蹤該預算金額的資源。

若要停用預算，請遵循下列步驟。

1. 如果您尚未登入，請登入 AWS Management Console，開啟截止日期 Cloud 主控台，選擇伺服器陣列，然後選擇 [管理工作]。
2. 從「預算管理程式」頁面的「有效預算」頁標中，選擇您要停用之預算旁邊的按鈕。
3. 從右上角的「作業」下拉式功能表中，選取「停用預算」。稍後，選取的預算會從「有效」變更為「失效」，並從「有效預算」頁標移至「失效預算」頁標。

使用期限雲端使用總管

使用截止日期雲端使用量總管，您可以查看每個伺服器陣列上發生的活動的即時指標。您可以透過不同的變數來查看伺服器陣列的成本，例如佇列、工作、授權產品或執行個體類型。選擇不同的時間範圍以查看特定時間段內的使用情況，並查看一段時間內的使用趨勢。您還可以查看所選數據點的詳細細分類，從而進一步了解指標。使用情況可以按時間（分鐘和小時）或費用（$ USD）顯示。

以下各節說明存取和使用期限雲端使用總管的步驟。

主題

- 先決條件
- 開啟使用情況總管
- 使用使用情況總管

先決條件

若要使用截止日期雲端使用總管，您必須擁有MANAGER或OWNER伺服器陣列權限。如需詳細資訊，請參閱 管理伺服器陣列、佇列和叢集的使用者和群組。

開啟使用情況總管

若要開啟截止日期雲端使用總管，請使用下列程序。

1. 登入 AWS Management Console 並開啟截止日期雲端主控台。

2. 若要查看所有可用的伺服器陣列，請選擇 [檢視
3. 找出您要取得相關資訊的伺服器陣列，然後選擇 [管理工作]。截止日期雲端監視器會在新標籤中開啟。
4. 在截止日期雲端監視器的左側功能表中，選取使用量總管。

使用使用情況總管

在使用情況總管頁面中，您可以選取可顯示資料的特定參數。根據預設，您會看到過去 7 天內的總使用量 (小時和分鐘)。您可以變更這些參數，顯示的資訊會根據參數設定動態變更。

您可以根據佇列、工作、計算使用量、執行個體類型或授權產品來分組結果。如果您選擇授權產品，則會計算特定授權的成本。對於所有其他組，時間是通過將每個任務運行所花費的時間加起來計算。

使用情況總管只會根據您設定的篩選條件傳回 100 個結果。結果會依建立時間戳記的日期遞減順序列出。如果結果超過 100 個，您會收到錯誤訊息。您可以細化查詢以減少結果數量：

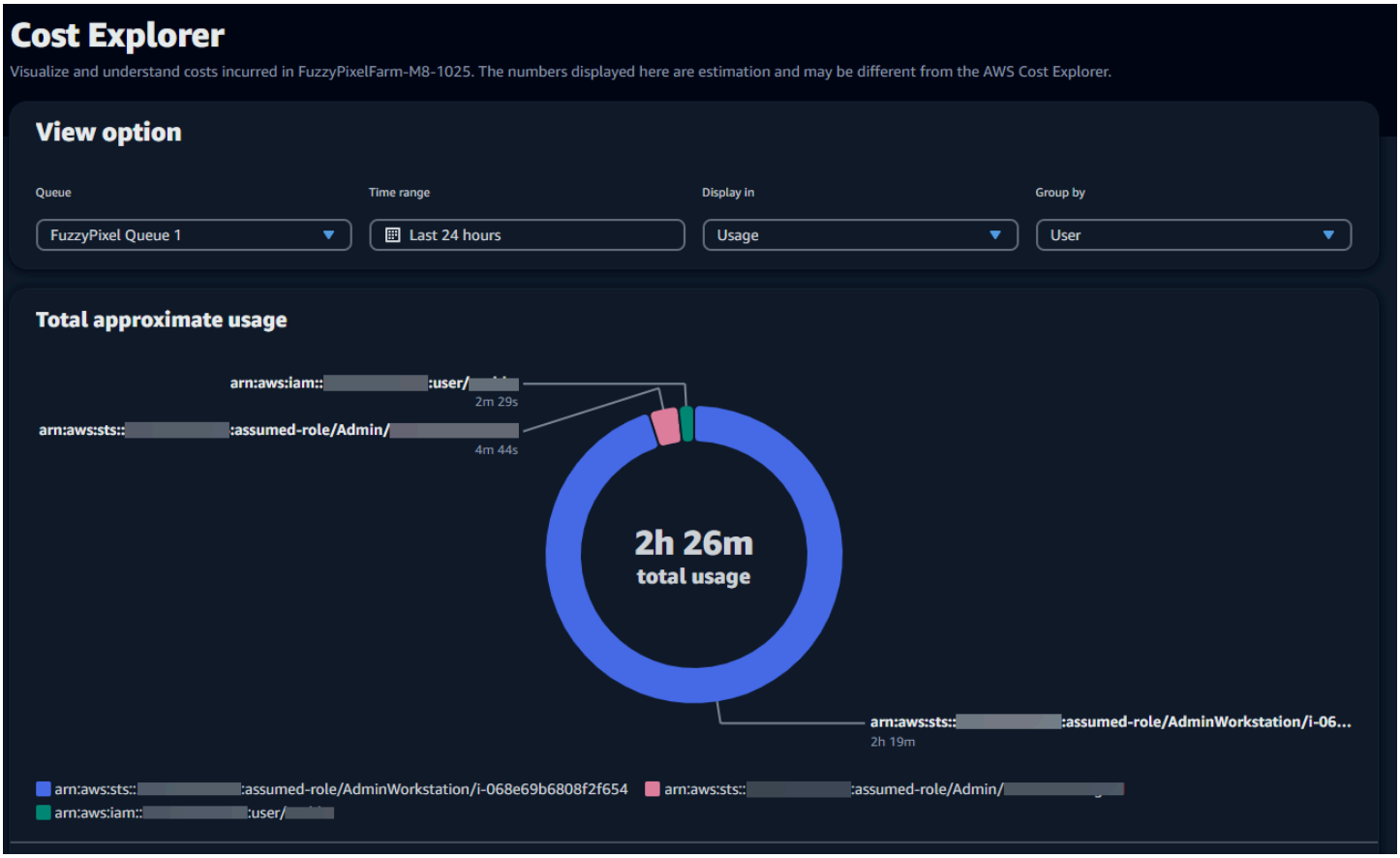
- 選擇較小的時間範圍
- 選取較少佇列
- 選取不同的群組，例如依佇列而非工作分組

主題

使用視覺化圖表檢閱資料

您可以使用視覺化格式檢閱資料，以識別可能需要更多分析或注意的趨勢和潛在區域。使用情況總管提供了一個圓形圖，可顯示整體使用情況和成本，並可選擇將總計分組為較小的小計。

> **Note**
>
> 圖表只會顯示前五個結果，其他結果合併在「其他」區段中。您可以在圖表下方的劃分區段中檢視所有結果。



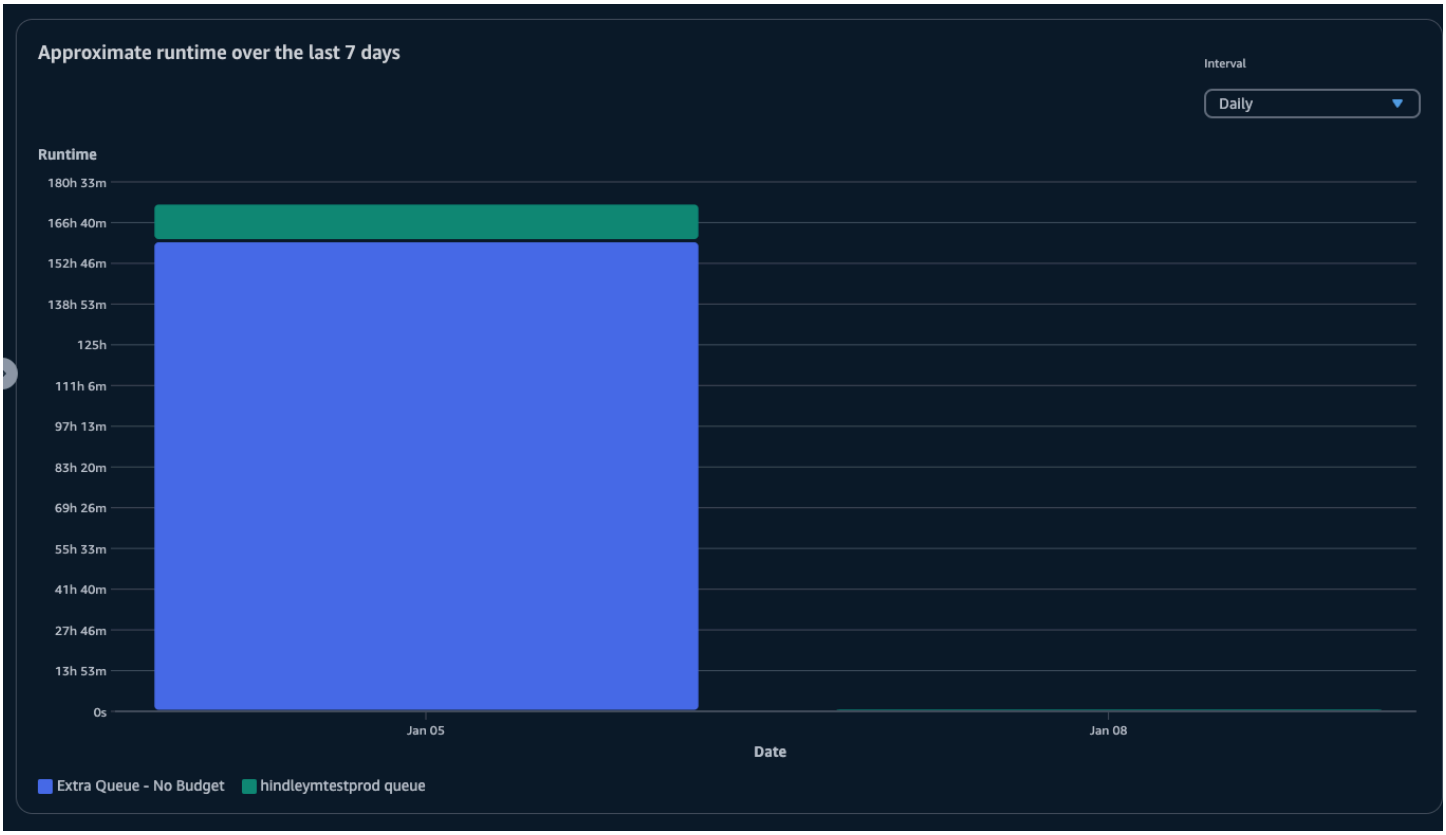
檢視指標明細

在圓餅圖下方，使用狀況總管會提供更詳細的特定量度劃分，這些指標會隨著參數變更而變更。依預設，五個結果會顯示在使用情況總管中。您可以使用劃分區段中的分頁箭頭來捲動結果。

依預設，劃分會最小化。若要展開並顯示結果，請選取檢視所有劃分箭頭。若要下載劃分，請選擇 [下載資料]。

檢視佇列的近似執行階段

您也可以根據指定的不同間隔檢視佇列的大約執行階段。間隔選項包括每小時、每天、每週和每月。選取間隔後，圖形會顯示佇列的大約執行階段。



成本管理

AWS 截止日期雲提供預算和用量總管，以幫助您控制和視覺化工作的成本。不過，截止日期雲端會使用其他 AWS 服務，例如 Amazon S3。這些服務的費用不會反映在截止日期雲端預算或使用量總管中，而是根據使用情況分別收費。視您設定截止日期雲端的方式而定，您可以使用下列 AWS 服務以及其他服務：

服務	定價頁面
Amazon CloudWatch 日誌	Amazon CloudWatch 日誌定價
Amazon Elastic Compute Cloud	Amazon 彈性運算雲定價
AWS Key Management Service	AWS Key Management Service 定價
AWS PrivateLink	AWS PrivateLink 定價
Amazon Simple Storage Service	Amazon Simple Storage Service 定價

服務	定價頁面
Amazon Virtual Private Cloud	Amazon Virtual Private Cloud 定價

成本管理最佳做法

使用以下最佳實務可協助您瞭解並控制使用 Dependpoint Cloud 時的成本，以及您可以在成本與效率之間進行的權衡。

> **Note**
>
> 使用截止日期雲端的最終成本取決於數個 AWS 服務之間的互動、處理的工作量以及執行工作的 AWS 區域 位置。下列最佳做法為準則，可能不會大幅降低成本。

CloudWatch 記錄檔的最佳做法

期限雲端會將工作者和工作記錄檔傳送至 CloudWatch 記錄檔 您需支付收集、儲存和分析這些記錄的費用。您可以只記錄監控任務所需的最少資料量，藉此降低成本。

當您建立佇列或叢集時，DependCloud 會建立具有下列名稱的 CloudWatch 記錄記錄檔群組：

- `aws/deadline/<FARM_ID>/<FLEET_ID>`
- `aws/deadline/<FARM_ID>/<QUEUE_ID>`

依預設，這些記錄永遠不會過期。您可以調整記錄群組的保留原則，以移除舊的記錄檔並協助降低儲存成本。您也可以將日誌匯出到 Amazon S3。Amazon S3 存儲成本低於 CloudWatch. 如需詳細資訊，請參閱將日誌資料匯出到 Amazon S3。

Amazon EC2 的最佳實務

您可以將 Amazon EC2 執行個體用於服務管理和客戶管理的叢集。有三個考慮因素：

- 對於服務管理的叢集，您可以設定叢集的最低背景工作者計數，選擇隨時提供一或多個執行個體。當您將最小背景工作者計數設定為 0 時，叢集一律會有許多 Worker 正在執行。這樣可以減少 Dependate Cloud 開始處理工作所需的時間，不過您需要支付執行個體閒置時間的費用。
- 對於服務管理的叢集，請設定叢集的大小上限。這會限制叢集可 auto 擴展至的執行個體數量。即使有更多工作等待處理，艦隊也不會超過這個規模。

- 對於服務管理和客戶管理的叢集，您可以在叢集中指定 Amazon EC2 執行個體類型。使用較小的執行個體每分鐘成本較低，但可能需要更長的時間才能完成工作。相反地，較大的執行個體每分鐘成本較高，但可以縮短完成工作的時間。瞭解您的工作對執行個體的需求有助於降低成本。
- 請盡可能為您的叢集選擇 Amazon EC2 競價型執行個體。Spot 執行個體可以降低價格使用，但可能會因隨需請求而中斷。隨需執行個體按秒計費，不會中斷。

的最佳做法 AWS KMS

根據預設，截止日期雲端會使用 AWS 擁有的金鑰對您的資料進行加密。您不需要支付此金鑰的費用。

您可以選擇使用客戶管理的金鑰來加密您的資料。當您使用自己的金鑰時，我們會根據金鑰的使用方式向您收費。如果您使用現有的金鑰，這將是額外使用的增量成本。

的最佳做法 AWS PrivateLink

您可以使 AWS PrivateLink 用介面端點在 VPC 和截止日期雲端之間建立連線。建立連線時，您可以呼叫所有截止日期 Cloud API 動作。您需要針對您建立的每個端點按小時計費。如果使用 PrivateLink，則必須至少建立三個端點，並且視您的組態而定，最多可能需要五個端點。

Amazon S3 的最佳實踐

截止日期雲端使用 Amazon S3 存放資產以進行處理、工作附件、輸出和日誌。若要降低與 Amazon S3 相關的成本，請減少存放的資料量。一些建議：

- 僅儲存目前使用中或即將使用的資產。
- 使用 S3 生命週期組態自動從 S3 儲存貯體刪除未使用的檔案。

Amazon VPC 的最佳實踐

當您針對客戶管理的叢集使用以使用為基礎的授權時，您會建立期限雲端授權端點，這是在您帳戶中建立的 Amazon VPC 端點。此端點按小時費率計費。若要降低成本，請在未使用以使用為基礎的授權時移除端點。

中的安全性 Deadline Cloud

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，這些架構是為了滿足對安全性最敏感的組織的需求而建置的。

安全是 AWS 與您之間共同承擔的責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 — AWS 負責保護 AWS 服務 中執行的基礎架構 AWS 雲端。 AWS 還為您提供可以安全使用的服務。若要瞭解適用的規範遵循方案 AWS Deadline Cloud，請參閱AWS 服務 遵循規範計劃。
- 雲端中的安全性 — 您的責任取決於您使用的資料。 AWS 服務 您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您瞭解如何在使用時套用共同責任模型 Deadline Cloud。下列主題說明如何設定 Deadline Cloud 以符合安全性與合規性目標。您還將學習如何使用其 AWS 服務 他幫助您監控和保護 Deadline Cloud 資源的其他方法。

主題

資料保護 Deadline Cloud

AWS 共用責任模型適用於中的資料保護 AWS Deadline Cloud。如此模型中所述， AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API Deadline Cloud 或 AWS SDK 時 AWS 服務 使用或其他使用時。 AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

主題

靜態加密

AWS Deadline Cloud 透過使用 AWS Key Management Service (AWS KMS) 中儲存的加密金鑰將其靜態加密，以保護敏感資料。靜態加密在所有可用的 AWS 區域 地方 Deadline Cloud 都可用。

加密資料表示沒有有效金鑰的使用者或應用程式無法讀取儲存在磁碟上的敏感資料。只有擁有有效受管理金鑰的對象才能解密資料。

若要取得有關如何 Deadline Cloud 使 AWS KMS 用靜態資料加密的資訊，請參閱金鑰管理。

傳輸中加密

對於傳輸中的資料，請 AWS Deadline Cloud 使用傳輸層安全性 (TLS) 1.2 或 1.3 來加密服務與背景工作之間傳送的資料。我們需要 TLS 1.2 並建議使用 TLS 1.3。此外，如果您使用虛擬私有雲（VPC），則可 AWS PrivateLink 以使用在 VPC 和. Deadline Cloud

金鑰管理

建立新的伺服器陣列時，您可以選擇下列其中一個金鑰來加密伺服器陣列資料：

- AWS 擁有的 KMS 金鑰 — 如果您在建立伺服器陣列時未指定金鑰，則為預設加密類型。KMS 金鑰擁有者 AWS Deadline Cloud。您無法檢視、管理或使用 AWS 擁有的金鑰。不過，您不需要採取任何動作來保護加密資料的金鑰。如需詳細資訊，請參閱AWS Key Management Service 開發人員指南中的AWS 擁有金鑰。
- 客戶受管 KMS 金鑰 — 您可以在建立伺服器陣列時指定客戶受管金鑰。伺服器陣列中的所有內容都會使用 KMS 金鑰加密。密鑰存儲在您的帳戶中，由您創建，擁有和管理，並 AWS KMS 收取費用。您可以完全控制 KMS 金鑰。您可以執行下列工作：
  - 建立和維護關鍵政策
  - 建立和維護 IAM 政策和授予操作
  - 啟用和停用金鑰政策
  - 新增標籤
  - 建立金鑰別名

  您無法手動輪換用於伺服 Deadline Cloud 器陣列的客戶擁有金鑰。支援自動旋轉金鑰。

  如需詳細資訊，請參閱AWS Key Management Service 開發人員指南中的客戶擁有的金鑰。

  若要建立客戶管理的金鑰，請依照AWS Key Management Service 開發人員指南中的建立對稱客戶管理金鑰的步驟進行。

如何 Deadline Cloud 使用 AWS KMS 補助

Deadline Cloud 需要授權才能使用您的客戶管理金鑰。當您建立使用客戶受管金鑰加密的伺服器陣列時， Deadline Cloud 會傳送`CreateGrant`要求以取得您指定之 KMS 金鑰存取權的要 AWS KMS 求，代表您建立授權。

Deadline Cloud 使用多個贈款。每個授權都被其中的不同部分使用 Deadline Cloud ，需要加密或解密您的數據。 Deadline Cloud 還使用授權來允許存取用於代表您存放資料的其他 AWS 服務，例如 Amazon 簡單儲存服務、Amazon 彈性區塊存放區或 OpenSearch。

允 Deadline Cloud 許管理服務管理叢集中機器的授權，包括 Deadline Cloud 帳戶號碼和角色，GranteePrincipal而不是服務主體。雖然並非典型，但若要使用為伺服器陣列指定的客戶受管 KMS 金鑰，為服務管理叢集中的員工加密 Amazon EBS 磁碟區是必要的。

客戶受管金鑰政策

金鑰政策會控制客戶受管金鑰的存取權限。每個金鑰都必須只有一個金鑰原則，其中包含判斷誰可以使用金鑰以及如何使用金鑰的陳述式。當您建立客戶受管金鑰時，您可以指定金鑰政策。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的管理客戶受管金鑰的存取。

適用的最低 IAM 政策 CreateFarm

若要使用客戶受管金鑰使用主控台或 CreateFarm API 作業建立伺服器陣列，必須允許下列 AWS KMS API 作業：

- kms:CreateGrant：新增客戶受管金鑰的授權。授與指定 AWS KMS 金鑰的主控台存取權。如需詳細資訊，請參閱AWS Key Management Service 開發人員指南中的使用補助金。
- kms:Decrypt— 允許 Deadline Cloud 解密伺服器陣列中的資料。
- kms:DescribeKey— 提供客戶管理的密鑰詳細信息， Deadline Cloud 以允許驗證密鑰。
- kms:GenerateDataKey— 允許 Deadline Cloud 使用唯一的數據密鑰加密數據。

下列原則陳述式會授與CreateFarm作業的必要權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineCreateGrants",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
```

```
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/1234567890abcdef0",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

唯讀作業的最低 IAM 政策

使用客戶管理的金鑰進 Deadline Cloud 行唯讀作業，例如取得伺服器陣列、佇列和叢集的相關資訊。必須允許下列 AWS KMS API 作業：

- kms:Decrypt— 允許 Deadline Cloud 解密伺服器陣列中的資料。
- kms:DescribeKey— 提供客戶管理的密鑰詳細信息， Deadline Cloud 以允許驗證密鑰。

下列原則陳述式會授與唯讀作業的必要權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadOnly",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

讀寫作業的最低 IAM 政策

使用客戶管理的金鑰進行讀寫 Deadline Cloud 作業，例如建立和更新伺服器陣列、佇列和叢集。必須允許下列 AWS KMS API 作業：

- kms:Decrypt— 允許 Deadline Cloud 解密伺服器陣列中的資料。
- kms:DescribeKey— 提供客戶管理的密鑰詳細信息， Deadline Cloud 以允許驗證密鑰。
- kms:GenerateDataKey— 允許 Deadline Cloud 使用唯一的數據密鑰加密數據。

下列原則陳述式會授與CreateFarm作業的必要權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadWrite",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey",
            ],
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

監控加密金鑰

將 AWS KMS 客戶受管金鑰與 Deadline Cloud 伺服器陣列搭配使用時，可以使用AWS CloudTrail或 Amazon CloudWatch Logs 追蹤 Deadline Cloud 傳送至的請求 AWS KMS。

CloudTrail 補助金事件

下列範例 CloudTrail 事件會在建立授權時發生，通常在您呼叫CreateFarmCreateMonitor、或CreateFleet作業時。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/Admin/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T02:05:26Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T02:05:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "operations": [
            "CreateGrant",
            "Decrypt",
            "DescribeKey",
            "Encrypt",
            "GenerateDataKey"
        ],
        "constraints": {
```

```
            "encryptionContextSubset": {
                "aws:deadline:farmId": "farm-abcdef1234567890987654321fedcba",
                "aws:deadline:accountId": "111122223333"
            }
        },
        "granteePrincipal": "deadline.amazonaws.com",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "retiringPrincipal": "deadline.amazonaws.com"
    },
    "responseElements": {
        "grantId": "6bbe819394822a400fe5e3a75d0e9ef16c1733143fff0c1fc00dc7ac282a18a0",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE44444"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail 用於解密的事件

使用客戶管理的 KMS 金鑰解密值時，會發生下列範例 CloudTrail 事件。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:51:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY
+p/5H+EuKd4Q=="
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "responseElements": null,
    "requestID": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeffffff",
    "eventID": "ffffffff-eeee-dddd-cccc-bbbbbbaaaaaa",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail 加密事件

使用客戶管理的 KMS 金鑰加密值時，會發生下列範例 CloudTrail 事件。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "numberOfBytes": 32,
        "encryptionContext": {
```

```
            "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY
+p/5H+EuKd4Q=="
        },
        "keyId": "arn:aws::kms:us-
west-2:111122223333:key/abcdef12-3456-7890-0987-654321fedcba"
    },
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


刪除客戶受管 KMS 金鑰

在 AWS Key Management Service (AWS KMS) 中刪除客戶管理的 KMS 金鑰具有破壞性且具有潛在危險性。它不可逆轉地刪除金鑰材料以及與金鑰相關聯的所有中繼資料。刪除客戶管理的 KMS 金鑰後，您就無法再解密該金鑰加密的資料。這意味著數據變得不可恢復。

這就是 AWS KMS 為什麼在刪除 KMS 金鑰之前提供客戶最多 30 天的等待期的原因。預設等待期間為 30 天。

關於等待期

由於刪除客戶管理的 KMS 金鑰具有破壞性且具有潛在危險性，因此我們要求您設定 7-30 天的等待期。預設等待期間為 30 天。

但是，實際等待時間可能比您排程的時間長達 24 小時。要獲取密鑰將被刪除的實際日期和時間，請使用該DescribeKey操作。您也可以在金鑰的詳細資料頁面的 [一般設定] 區段中，在AWS KMS 主控台中查看金鑰的排定刪除日期。請注意時區。

在等待期間，客戶管理的金鑰狀態和金鑰狀態為擱置刪除。

- 待刪除的客戶管理 KMS 金鑰無法用於任何密碼編譯作業。
- AWS KMS 不會輪替待刪除之客戶受管 KMS 金鑰的後備金鑰。

如需有關刪除客戶受管 KMS 金鑰的詳細資訊，請參閱AWS Key Management Service 開發人員指南中的刪除客戶主金鑰。

網際網路流量隱私權

AWS Deadline Cloud 支持 Amazon Virtual Private Cloud（Amazon VPC）以保護連接。Amazon VPC 提供的功能可讓您用來增加和監控虛擬私有雲 (VPC) 的安全性。

您可以使用在 VPC 內執行的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體來設定客戶管理叢集 (CMF)。透過部署要使用的 Amazon VPC 端點 AWS PrivateLink，CMF 中工作者與端 Deadline Cloud 點之間的流量會保留在您的 VPC 中。此外，您可以將 VPC 設定為限制執行個體的網際網路存取。

在服務管理的機隊中，工作人員無法從互聯網訪問，但他們確實可以訪問互聯網並通過互聯網連接到 Deadline Cloud 服務。

選擇退出

AWS Deadline Cloud 收集某些操作信息以幫助我們發展和改進 Deadline Cloud。收集的數據包括您的 AWS 帳戶 ID 和用戶 ID 之類的內容，以便我們可以在您遇到問題時正確識別您的身份 Deadline Cloud。我們也會收集 Deadline Cloud 特定資訊，例如資源 ID (適用時為 FarMid 或 queueID)、產品名稱 (例如 JobAttachments WorkerAgent、等) 及產品版本。

您可以選擇使用應用程式組態選擇退出此資料收集。每台與 Deadline Cloud客戶端工作站和車隊工作者互動的計算機都需要單獨選擇退出。

Deadline Cloud 監視器-桌面

Deadline Cloud monitor-桌面會收集操作資訊，例如當機發生時和應用程式開啟時，以協助我們瞭解您何時遇到應用程式問題。若要選擇退出此操作資訊的收集，請前往設定頁面並清除開啟資料收集，以測量截止日期 Cloud Monitor 的效能。

選擇退出之後，桌上型電腦監視器就不會再傳送作業資料。任何先前收集的數據將被保留，並可能仍然可以用於改善服務。如需更多資訊，請參閱 資料隱私權常見問答集。

AWS Deadline Cloud CLI 和工具

AWS Deadline Cloud CLI、提交者和 Worker Agent 都會收集操作資訊，例如發生當機的時間和提交工作的時間，以協助我們瞭解您何時遇到這些應用程式的問題。若要選擇退出此操作資訊的收集，請使用以下任何一種方法：

- 在終端機中，輸入**`deadline config set telemetry.opt_out true`**。

  這將在以當前用戶身份運行時選擇退出 CLI，提交者和 Worker 代理。
- 安裝 Deadline Cloud Worker 代理程式時，請新增**`--telemetry-opt-out`**命令列引數。例如 **`./install.sh --farm-id $FARM_ID --fleet-id $FLEET_ID --telemetry-opt-out`**。
- 在執行 Worker 代理程式、CLI 或提交者之前，請先設定環境變數：**`DEADLINE_CLOUD_TELEMETRY_OPT_OUT=true`**

選擇退出後，這些工 Deadline Cloud 具將不再發送操作數據。任何先前收集的數據將被保留，並可能仍然可以用於改善服務。如需更多資訊，請參閱 資料隱私權常見問答集。

期限雲端中的 Identity and Access Management

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員可控制哪些人可以通過驗證 (登入) 和授權 (具有權限) 來使用截止日期雲端資源。IAM 是您可以使用的 AWS 服務 ，無需額外付費。

主題

- 物件
- 使用身分驗證
- 使用政策管理存取權
- 截止日期雲端如何搭配 IAM 運作
- 截止日期雲端的身分識別原則範例
- AWS 截止日期雲端的受管政策
- 疑難排解 AWS 期限雲端身分和存取

物件

您使用 AWS Identity and Access Management (IAM) 的方式會因您在期限雲端中執行的工作而有所不同。

服務使用者 — 如果您使用 Definition Cloud 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多截止日期雲端功能來完成工作時，您可能需要其他權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取期限雲端中的功能，請參閱疑難排解 AWS 期限雲端身分和存取。

服務管理員 — 如果您負責公司的截止日期雲端資源，您可能擁有截止日期雲端的完整存取權。決定您的服務使用者應存取哪些截止日期雲端功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何將 IAM 與期限雲端搭配使用，請參閱截止日期雲端如何搭配 IAM 運作。

IAM 管理員 — 如果您是 IAM 管理員，可能需要瞭解如何撰寫政策以管理截止日期雲端存取權的詳細資訊。若要檢視可在 IAM 中使用的截止日期 Cloud 身分型政策範例，請參閱。截止日期雲端的身分識別原則範例

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identi ty Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。 AWS Directory Service同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI

或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的 IAM 中的跨帳戶資源存取。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務服務。
  - 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。如需了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需 Organizations 和 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

截止日期雲端如何搭配 IAM 運作

在您使用 IAM 管理截止日期雲端的存取權限之前，請先了解哪些 IAM 功能可用於截止日期雲端。

可搭配 AWS 期限雲端使用的 IAM 功能

IAM 功能	截止日期雲支持
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
轉送存取工作階段 (FAS)	是
服務角色	是
服務連結角色	否

若要深入瞭解截止日期雲端和其他使 AWS 服務 用大多數 IAM 功能的方式，請參閱 IAM 使用者指南中的與 IAM 搭配使用的AWS 服務。

截止日期雲端的身分識別原則

支援身分型政策	是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

截止日期雲端的身分識別原則範例

若要檢視截止日期雲端身分識別原則的範例，請參閱。截止日期雲端的身分識別原則範例

截止日期雲端內的資源型政策

支援以資源基礎的政策	否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM 中的跨帳戶資源存取。

截止日期雲端的政策動作

支援政策動作	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看截止日期雲端動作清單，請參閱服務授權參考資料中的AWS 截止日期雲端定義的動作。

截止日期雲端中的政策動作會在動作之前使用下列前置詞：

```
deadline
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "deadline:action1",
      "deadline:action2"
         ]
```

若要檢視截止日期雲端身分識別原則的範例，請參閱。截止日期雲端的身分識別原則範例

截止日期雲端的原則資源

支援政策資源	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看截止日期雲端資源類型及其 ARN 的清單，請參閱服務授權參考資料中的 AWS 截止日期雲端定義的資源。若要瞭解您可以使用哪些動作指定每個資源的 ARN，請參閱AWS 截止日期雲端定義的動作。

若要檢視截止日期雲端身分識別原則的範例，請參閱。截止日期雲端的身分識別原則範例

截止日期雲端的原則條件金鑰

支援服務特定政策條件金鑰	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯OR運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

若要查看截止日期雲端條件金鑰清單，請參閱服務授權參考資料中的AWS 截止日期雲端的條件金鑰。若要瞭解可以使用條件索引鍵的動作和資源，請參閱AWS 截止日期雲端定義的動作。

若要檢視截止日期雲端身分識別原則的範例，請參閱。截止日期雲端的身分識別原則範例

截止日期雲端中的 ACL

支援 ACL	否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 與截止日期雲

支援 ABAC (政策中的標籤)	是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱 IAM 使用者指南中的什麼是 ABAC?。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

使用臨時登入資料搭配期限雲

支援臨時憑證	是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料搭配AWS 服務 使用，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的切換至角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

期限雲端的轉寄存取工作階段

支援轉寄存取工作階段 (FAS)	是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求

AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。

截止日期雲端的服務角色

支援服務角色	是

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務服務。

> **Warning**
>
> 變更服務角色的權限可能會中斷期限雲端功能。只有在截止日期雲端提供指引時，才編輯服務角色。

截止日期雲端的服務連結角色

支援服務連結角色。	否

服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

截止日期雲端的身分識別原則範例

根據預設，使用者和角色沒有建立或修改截止日期雲端資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

有關 Deptionate Cloud 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱服務授權參考資料中 AWS Deptionate Cloud 的動作、資源和條件索引鍵。

主題

- 政策最佳實務
- 使用截止日期雲端主控台
- 將工作提交至佇列的原則
- 允許建立授權端點的策略
- 允許監視特定伺服器陣列佇列的原則

政策最佳實務

以身分識別為基礎的政策會決定某人是否可以在您的帳戶中建立、存取或刪除 Definition Cloud 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始授與使用者和工作負載的權限，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用截止日期雲端主控台

若要存取 AWS 截止日期雲端主控台，您必須擁有最少一組權限。這些權限必須允許您列出並檢視您的 AWS 帳戶. 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用截止日期雲端主控台，請同時將截止日期雲端*ConsoleAccess*或*ReadOnly* AWS 受管理的政策附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

將工作提交至佇列的原則

在此範例中，您會建立縮短原則，以授與將工作提交至特定伺服器陣列中特定佇列的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SubmitJobsFarmAndQueue",
            "Effect": "Allow",
            "Action": "deadline:CreateJob",
            "Resource": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_A/queue/QUEUE_B/
job/*"
        }
    ]
}
```

允許建立授權端點的策略

在此範例中，您會建立一個縮短範圍策略，以授與建立和管理授權端點所需的權限。使用此原則可為與伺服器陣列相關聯的 VPC 建立授權端點。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "SID": "CreateLicenseEndpoint",
        "Effect": "Allow",
```

```
        "Action": [
            "deadline:CreateLicenseEndpoint",
            "deadline:DeleteLicenseEndpoint",
            "deadline:GetLicenseEndpoint",
            "deadline:UpdateLicenseEndpoint",
            "deadline:ListLicenseEndpoints",
            "deadline:PutMeteredProduct",
            "deadline:DeleteMeteredProduct",
            "deadline:ListMeteredProducts",
            "deadline:ListAvailableMeteredProducts",
            "ec2:CreateVpcEndpoint",
            "ec2:DescribeVpcEndpoints",
            "ec2:DeleteVpcEndpoints"
        ],
        "Resource": "*"
    }]
}
```

允許監視特定伺服器陣列佇列的原則

在此範例中，您會建立一個縮短範圍原則，以授與監視特定伺服器陣列之特定佇列中工作的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "MonitorJobsFarmAndQueue",
        "Effect": "Allow",
        "Action": [
            "deadline:SearchJobs",
            "deadline:ListJobs",
            "deadline:GetJob",
            "deadline:SearchSteps",
            "deadline:ListSteps",
            "deadline:ListStepConsumers",
            "deadline:ListStepDependencies",
            "deadline:GetStep",
            "deadline:SearchTasks",
            "deadline:ListTasks",
            "deadline:GetTask",
            "deadline:ListSessions",
            "deadline:GetSession",
            "deadline:ListSessionActions",
            "deadline:GetSessionAction"
```

```
        ],
        "Resource": [
            "arn:aws:deadline:REGION:123456789012:farm/FARM_A/queue/QUEUE_B",
            "arn:aws:deadline:REGION:123456789012:farm/FARM_A/queue/QUEUE_B/*"
        ]
    }]
}
```

AWS 截止日期雲端的受管政策

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。 AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住， AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別 (使用者、群組和角色)。 AWS 當新的啟動或新 AWS 服務 的 API 操作可用於現有服務時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管理的策略：AWSDeadlineCloud-FleetWorker

您可以將`AWSDeadlineCloud-FleetWorker`政策附加到您的 AWS Identity and Access Management (IAM) 身分識別。

此原則會授與此叢集中的工作者連線至服務並從服務接收工作所需的權限。

許可詳細資訊

此政策包含以下許可：

- `deadline`— 允許主參與者管理叢集中的工作者。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南FleetWorker中的 AWSDeadlineCloud-。

AWS 受管理的策略：AWSDeadlineCloud-WorkerHost

您可將 AWSDeadlineCloud-WorkerHost 政策連接到 IAM 身分。

此原則會授與初始連線至服務所需的權限。它可以用作亞馬遜彈性運算雲端 (Amazon EC2) 執行個體設定檔。

許可詳細資訊

此政策包含以下許可：

- deadline-允許主參與者建立工作者。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南WorkerHost中的 AWSDeadlineCloud-。

AWS 受管理的策略：AWSDeadlineCloud-UserAccessFarms

您可將 AWSDeadlineCloud-UserAccessFarms 政策連接到 IAM 身分。

此原則可讓使用者根據所屬的伺服器陣列及其成員資格層級存取伺服器陣列資料。

許可詳細資訊

此政策包含以下許可：

- deadline— 允許使用者存取伺服器陣列資料。
- ec2— 允許使用者查看有關 Amazon EC2 執行個體類型的詳細資訊。
- identitystore— 允許使用者檢視使用者和群組名稱。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南中的 AWSDeadlineCloud-UserAccess 伺服器陣列。

AWS 受管理的策略：AWSDeadlineCloud-UserAccessFleets

您可將 AWSDeadlineCloud-UserAccessFleets 政策連接到 IAM 身分。

此原則可讓使用者根據所屬的伺服器陣列及其成員資格層級存取叢集資料。

許可詳細資訊

此政策包含以下許可：

- deadline— 允許使用者存取伺服器陣列資料。
- ec2— 允許使用者查看有關 Amazon EC2 執行個體類型的詳細資訊。
- identitystore— 允許使用者檢視使用者和群組名稱。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南中的 AWSDeadlineCloud-UserAccess 叢集。

AWS 受管理的策略：AWSDeadlineCloud-UserAccessJobs

您可將 AWSDeadlineCloud-UserAccessJobs 政策連接到 IAM 身分。

此原則可讓使用者根據所屬的伺服器陣列及其成員資格層級存取工作資料。

許可詳細資訊

此政策包含以下許可：

- deadline— 允許使用者存取伺服器陣列資料。
- ec2— 允許使用者查看有關 Amazon EC2 執行個體類型的詳細資訊。
- identitystore— 允許使用者檢視使用者和群組名稱。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南中的 AWSDeadlineCloud-UserAccess 任務。

AWS 受管理的策略：AWSDeadlineCloud-UserAccessQueues

您可將 AWSDeadlineCloud-UserAccessQueues 政策連接到 IAM 身分。

此原則可讓使用者根據所屬的伺服器陣列及其成員資格層級存取佇列資料。

許可詳細資訊

此政策包含以下許可：

- deadline— 允許使用者存取伺服器陣列資料。
- ec2— 允許使用者查看有關 Amazon EC2 執行個體類型的詳細資訊。

- identitystore— 允許使用者檢視使用者和群組名稱。

如需政策詳細資訊的 JSON 清單，請參閱 AWS 受管政策參考指南中的 AWSDeadlineCloud-UserAccessQueues。

截止日期雲端更新受 AWS 管理政策

檢視此服務開始追蹤這些變更後，截止日期雲端的 AWS 受管政策更新詳細資料。如需有關此頁面變更的自動警示，請訂閱 「截止日期雲端文件歷史記錄」頁面上的 RSS 摘要。

變更	描述	日期
期限雲端開始追蹤變更	截止日期 Cloud 開始追蹤其受 AWS 管理政策的變更。	2024年4月2日

疑難排解 AWS 期限雲端身分和存取

使用下列資訊可協助您診斷並修正使用期限雲端和 IAM 時可能會遇到的常見問題。

主題

- 我沒有在期限雲端中執行動作的授權
- 我沒有授權執行 iam：PassRole
- 我想允許我以外的人員存取我 AWS 帳戶 的截止日期雲端資源

我沒有在期限雲端中執行動作的授權

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 deadline:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 deadline:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 deadline:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam：PassRole

如果您收到未獲授權執行iam:PassRole動作的錯誤訊息，您必須更新原則，才能讓您將角色傳遞給 Dependate Cloud。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者marymajor嘗試使用主控台在截止日期 Cloud 中執行動作時，就會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人員存取我 AWS 帳戶 的截止日期雲端資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解截止日期雲端是否支援這些功能，請參閱截止日期雲端如何搭配 IAM 運作。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。

- 若要了解跨帳戶存取使用角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的 IAM 中的跨帳戶資源存取。

符合性驗證 Deadline Cloud

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

韌性 Deadline Cloud

AWS 全球基礎架構是圍繞 AWS 區域 和可用區域建立的。 AWS 區域 提供多個實體分離和隔離的可用區域，這些區域與低延遲、高輸送量和高冗餘網路相連。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱AWS 全域基礎結構。

AWS Deadline Cloud 不會備份存放在任務附件 S3 儲存貯體中的資料。您可以使用任何標準 Amazon S3 備份機制 (例如 S3 版本控制或 AWS Backup.

期限雲端中的基礎架構安

作為託管服務， AWS 截止日期雲受到 AWS 全球網絡安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫，透過網路存取期限雲端。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

截止日期雲端不支援使用 AWS PrivateLink 虛擬私有雲端 (VPC) 端點原則。它會使用 AWS PrivateLink 預設原則，授與端點的完整存取權。如需詳細資訊，請參閱AWS PrivateLink 使用指南中的預設端點策略。

期限雲中的配置和漏洞分析

AWS 處理基本安全性工作，例如客體作業系統 (OS) 和資料庫修補、防火牆組態和嚴重損壞修復。這些程序已由適當的第三方進行檢閱並認證。如需詳細資訊，請參閱以下 資源：

- 共同的責任模型
- Amazon Web Services：安全程序概觀 (白皮書)

AWS 截止日期 Cloud 會管理服務管理或客戶管理叢集上的工作：

- 對於服務管理的叢集，截止日期雲端會管理客體作業系統。
- 對於客戶管理的叢集，您必須負責管理作業系統。

如需有關 AWS 截止日期雲端的設定和弱點分析的其他資訊，請參閱

- 期限雲端的安全性最佳做法

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆的副問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

建議您在資源策略中使用aws:SourceArn和aws:SourceAccount全域條件前後關聯索引鍵，以限制將其他服務 AWS Deadline Cloud 提供給資源的權限。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，其中包含資源的完整 Amazon Resource Name (ARN)。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 `aws:SourceArn` 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 `arn:aws:deadline:*:123456789012:*`。

如果 `aws:SourceArn` 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須使用這兩個全域條件內容索引鍵來限制許可。

下列範例顯示如何在中使用`aws:SourceArn`和`aws:SourceAccount`全域條件前後關聯鍵字 Deadline Cloud 來避免混淆的副問題。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
```

```
      "Service": "deadline.amazonaws.com"
    },
    "Action": "deadline:ActionName",
    "Resource": [
      "*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:deadline:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

使 AWS Deadline Cloud 用介面端點存取 (AWS PrivateLink)

您可 AWS PrivateLink 以使用在 VPC 和 AWS Deadline Cloud. 您可以 Deadline Cloud 像在 VPC 中一樣進行存取，而無需使用網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公用 IP 位址即可存取 Deadline Cloud。

您可以建立由 AWS PrivateLink提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可作為目的地為 Deadline Cloud之流量的進入點。

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的透過 AWS PrivateLink存取 AWS 服務。

的注意事項 Deadline Cloud

設定的界面端點之前 Deadline Cloud，請參閱AWS PrivateLink 指南中的使用界面 VPC 端點存取 AWS 服務。

Deadline Cloud 支援透過介面端點呼叫其所有 API 動作。

預設情況下，允許透過介面端點進行完整存取。 Deadline Cloud 或者，您可以將安全群組與端點網路介面相關聯，以控制 Deadline Cloud 透過介面端點傳送到的流量。

Deadline Cloud 不支援 VPC 端點原則。如需詳細資訊，請參閱AWS PrivateLink 指南中的使用端點策略控制對 VPC 端點的存取。

Deadline Cloud 端點

Deadline Cloud 使用兩個端點存取服務 AWS PrivateLink。

Worker 會使用com.amazonaws.*region*.deadline.scheduling端點從佇列取得工作、將進度報告至 Deadline Cloud，以及將工作輸出傳回。如果您使用的是客戶管理的叢集，除非您正在使用管理作業，否則排程端點是您唯一需要建立的端點。例如，如果工作建立更多工作，您需要啟用管理端點來呼叫CreateJob作業。

Deadline Cloud 監視器會使用com.amazonaws.*region*.deadline.management來管理伺服器陣列中的資源，例如建立和修改佇列和叢集，或取得作業、步驟和工作的清單。

Deadline Cloud 下列 AWS 服務端點也需要端點：

- Deadline Cloud 用 AWS STS 於驗證 Worker，以便他們可以存取工作資產。如需詳細資訊 AWS STS，請參閱AWS Identity and Access Management 使用指南中 IAM 中的臨時安全登入資料。
- 如果您在沒有網際網路連線的子網路中設定客戶管理的叢集，則必須為 Amazon CloudWatch Logs 建立 VPC 端點，以便工作者可以寫入日誌。如需詳細資訊，請參閱使用監視 CloudWatch。
- 如果您使用任務附件，則必須為 Amazon Simple Storage Service (Amazon S3) 建立 VPC 端點，以便工作者可以存取附件。如需詳細資訊，請參閱中的 Job 附件 Deadline Cloud。

建立端點 Deadline Cloud

您可以建立介面端點，以便 Deadline Cloud 使用 Amazon VPC 主控台或 AWS Command Line Interface (AWS CLI)。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的建立介面端點。

建立使用下列服務名稱 Deadline Cloud 的管理和排程端點。將##替換為 AWS 區域 您部署的位置 Deadline Cloud。

```
com.amazonaws.region.deadline.management
```

```
com.amazonaws.region.deadline.scheduling
```

如果您為介面端點啟用私有 DNS，您可以 Deadline Cloud 使用其預設的區域 DNS 名稱向 API 要求。例如，worker.deadline.us-east-1.amazonaws.com針對 Worker 作業或所management.deadline.us-east-1.amazonaws.com有其他作業。

您也必須建立 AWS STS 使用下列服務名稱的端點：

```
com.amazonaws.region.sts
```

如果您的客戶管理叢集位於沒有網際網路連線的子網路上，您必須使用下列服務名稱建立 CloudWatch Logs 端點：

```
com.amazonaws.region.logs
```

如果您使用任務附件傳輸檔案，則必須使用下列服務名稱建立 Amazon S3 端點：

```
com.amazonaws.region.s3
```

期限雲端的安全性最佳做法

AWS 截止日期雲端 (截止日期雲端) 提供許多安全性功能，可在您開發和實作自己的安全性原則時考慮。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

> **Note**
>
> 如需有關許多安全性主題重要性的詳細資訊，請參閱共用的責任模型。

資料保護

基於資料保護目的，我們建議您使用 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別帳戶。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及其中的所有默認安全控制 AWS 服務。
- 使用 Amazon Macie 等進階受管安全服務，協助探索和保護存放在 Amazon 簡單儲存服務 (Amazon S3) 中的個人資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶帳戶號碼等敏感的識別資訊，放在自由格式的欄位中，例如名稱欄位。這包括當您使用 AWS 截止日期雲端或其他 AWS 服務 使用主控台 AWS CLI、API 或 AWS SDK 時。您輸入到 Definition Cloud 或其他服務的任何資料都可能會被拾取，以便包含在診斷記錄中。當您提供外部伺服器的 URL 時，請勿在驗證您對該伺服器請求的 URL 中包含登入資料資訊。

AWS Identity and Access Management 權限

使用使用者 AWS Identity and Access Management (IAM) 角色管理 AWS 資源存取權，並授予使用者最低權限。建立憑證管理原則和程序，以建立、散佈、輪換及撤銷 AWS 存取認證。如需詳細資訊，請參《IAM 使用者指南》中的 IAM 最佳實務。

以使用者和群組身分執行工作

在 Definition Cloud 中使用佇列功能時，最佳做法是指定作業系統 (OS) 使用者及其主要群組，讓作業系統使用者擁有佇列工作的最低權限。

當您指定「執行身分使用者」(和群組) 時，提交至佇列之作業的任何處理程序都會使用該 OS 使用者執行，並繼承該使用者的相關作業系統權限。

叢集和佇列組態結合在一起，以建立安全性狀態。在佇列端，可以指定「Job 業以使用者身分執行」和 IAM 角色來使用佇列作業的作業系統和 AWS 許可。叢集會定義基礎結構 (背景工作者主機、網路、掛接的共用儲存體)，當與特定佇列相關聯時，會執行佇列中的工作。Worker 主機上可用的資料必須由一或多個關聯佇列中的工作存取。指定使用者或群組有助於保護工作中的資料，不受其他佇列、其他已安裝的軟體或其他可存取 Worker 主機的使用者影響。當佇列沒有使用者時，它會以可模擬 (sudo) 任何佇列使用者的代理程式使用者身分執行。如此一來，沒有使用者的佇列就可以將權限提升到另一個佇列。

聯網

若要防止流量遭到攔截或重新導向，確保路由網路流量的方式和位置非常重要。

我們建議您以下列方式保護您的網路環境：

- 保護 Amazon Virtual Private Cloud (Amazon VPC) 子網路路由表，以控制 IP 層流量的路由方式。
- 如果您在伺服器陣列或工作站設定中使用 Amazon 路線 53 (Route 53) 做為 DNS 提供者，請安全存取 Route 53 API。
- 如果您在外部 (AWS 例如使用內部部署工作站或其他資料中心) 連線到 Dependate Cloud，請保護任何內部部署網路基礎結構。這包括路由器、交換器和其他網路裝置上的 DNS 伺服器和路由表。

工作和工作資料

截止日期雲端工作會在工作者主機上的工作階段 每個工作階段都會在 Worker 主機上執行一或多個處理序，這通常需要您輸入資料才能產生輸出。

若要保護這些資料，您可以使用佇列設定作業系統使用者。Worker 代理程式會使用佇列作業系統使用者執行工作階段子處理序。這些子程序會繼承佇列 OS 使用者的權限。

我們建議您遵循最佳做法，以確保存取這些子程序存取的資料安全。如需詳細資訊，請參閱共同責任模式。

農場結構

您可以通過多種方式安排截止日期雲艦隊和隊列。但是，某些安排存在安全隱患。

伺服器陣列具有最安全的界限之一，因為它無法與其他伺服器陣列 (包括叢集、佇列和儲存區設定檔) 共用 Dependate Cloud 資源。不過，您可以共用伺服器陣列內的外部 AWS 資源，這會損害安全性邊界。

您也可以使用適當的組態，在相同伺服器陣列內的佇列之間建立安全性界限。

請遵循下列最佳做法，在相同的伺服器陣列中建立安全佇列：

- 僅將叢集與相同安全性界限內的佇列產生關聯。注意下列事項：
  - 在 Worker 主機上執行工作後，資料可能會保留在後面，例如暫存目錄或佇列使用者的主目錄中。
  - 相同的作業系統使用者會在服務擁有的叢集 Worker 主機上執行所有作業，而不論您將工作提交至哪個佇列。
  - 工作可能會讓處理序在 Worker 主機上執行，讓其他佇列中的工作可以觀察其他執行中的處理序。
- 確保只有在相同安全邊界內的佇列共用 Amazon S3 儲存貯體以存放任務附件。
- 確定只有位於相同安全性界限內的佇列共用作業系統使用者。
- 保護整合至伺服器陣列邊界的任何其他 AWS 資源。

Job 附件佇列

Job 務附件與使用 Amazon S3 儲存貯體的佇列相關聯。

- 從 Amazon S3 儲存貯體中的根前綴寫入和讀取 Job 務附件。您可以在 `CreateQueue` API 呼叫中指定此根前置詞。

- 值區具有對應的角色Queue Role，可指定授與佇列使用者存取值區和根前置詞的角色。建立佇列時，您可以在任務附件儲存貯體和根前置詞旁邊指定 Queue Role Amazon 資源名稱 (ARN)。
- 對AssumeQueueRoleForRead、AssumeQueueRoleForUser和AssumeQueueRoleForWorker API 作業的授權呼叫會傳回一組的臨時安全登入資料Queue Role。

如果您建立佇列並重複使用 Amazon S3 儲存貯體和根前綴，則可能會向未經授權的方披露資訊。例如，QueEA 和 QueueB 共享相同的存儲桶和根前綴。在安全的工作流程中，藝術家可以存取佇列，但不能存取佇列 B。但是，當多個佇列共用一個值區時，ArtiStA 可以存取 QueueB 資料中的資料，因為它使用與 QueuEa 相同的儲存貯體和根前置詞。

主控台會設定預設安全的佇列。確保佇列具有不同的 Amazon S3 儲存貯體和根前綴組合，除非它們屬於通用安全邊界的一部分。

若要隔離佇列，您必須將設定Queue Role為僅允許佇列存取值區和根前置詞。在下列範例中，將每個####取代為您的資源特定資訊。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME",
        "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME/JOB_ATTACHMENTS_ROOT_PREFIX/*"
      ],
      "Condition": {
        "StringEquals": { "aws:ResourceAccount": "ACCOUNT_ID" }
      }
    },
    {
      "Action": ["logs:GetLogEvents"],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:REGION:ACCOUNT_ID:log-group:/aws/deadline/FARM_ID/*"
    }
```

```
  ]
}
```

您也必須在角色上設定信任原則。在下列範例中，以您的資源特定資訊取代*####*文字。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Effect": "Allow",
      "Principal": { "Service": "deadline.amazonaws.com" },
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "ACCOUNT_ID" },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_ID"
        }
      }
    },
    {
      "Action": ["sts:AssumeRole"],
      "Effect": "Allow",
      "Principal": { "Service": "credentials.deadline.amazonaws.com" },
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "ACCOUNT_ID" },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_ID"
        }
      }
    }
  ]
}
```

定制軟件 Amazon S3 存儲桶

您可以將下列陳述式新增`Queue Role`至您的，以存取 Amazon S3 儲存貯體中的自訂軟體。在下列範例中，將*############ S3 #######*。

```
"Statement": [
    {
        "Action": [
            "s3:GetObject",
```

```
            "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME",
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME/*"
        ]
    }
]
```

如需有關 Amazon S3 安全最佳實務的詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的 Amazon S3 安全最佳實務。

工作者主機

保護 Worker 主機，以協助確保每位使用者只能針對其指派的角色執行作業。

我們建議採用下列最佳做法來保護 Worker 主機：

- 除非提交至這些佇列的工作位於相同的安全性界限內，否則請勿在多個佇列中使用相同的`jobRunAsUser`值。
- 請勿`jobRunAsUser`將佇列設定為 Worker 代理程式執行的作業系統使用者名稱。
- 授與佇列使用者預定佇列工作負載所需的最低權限作業系統權限。確定他們沒有檔案系統寫入工作代理程式檔案或其他共用軟體的權限。
- 請確定只有 root 使用者Linux和擁有的帳號`Administrator`擁有，而且可以修改 Worker 代理程式程式檔案。Windows
- 在 Linux Worker 主機上，請考慮在中配置`umask`覆寫，`/etc/sudoers`以允許 Worker 代理程式使用者以佇列使用者身分啟動處理序。此設定有助於確保其他使用者無法存取寫入佇列的檔案。
- 授與受信任的個人對 Worker 主機的最低權限存取權。
- 限制本機 DNS 覆寫組態檔的權限 (開`/etc/hosts`啟Linux和開`C:\Windows\system32\etc\hosts`啟Windows，以及路由工作站和 Worker 主機作業系統上的表格。
- 限制工作站和背景工作者主機作業系統上 DNS 組態的權限。
- 定期修補操作系統和所有已安裝的軟件。這種方法包括專門與截止日期雲端配合使用的軟體，例如提交者、轉接器、背景工作者代理程式、OpenJD套件等。
- 在Windows佇列中使用強式密碼`jobRunAsUser`。
- 定期輪換佇列的密碼`jobRunAsUser`。
- 確保密碼的最低權限存取權限會隱藏並刪除未使用的Windows密碼。

- 不要授予隊列jobRunAsUser權限的計劃命令 future 運行：
  - 開啟Linux，拒絕這些帳戶存取cron和at。
  - 開啟時Windows，拒絕這些帳戶對Windows工作排程器的存取權。

> **Note**
>
> 如需有關定期修補作業系統和已安裝軟體之重要性的詳細資訊，請參閱共同的責任模型。

工作站

它是重要的是要保護工作站與訪問截止日期雲. 此方法有助於確保您提交到 Dependate Cloud 的任何工作都無法執行向您計費的任意工作負載 AWS 帳戶。

我們建議採用下列最佳作法來保護藝術家工作站。如需詳細資訊，請參閱 共同責任模型。

- 保護任何提供存取權的持續認證 AWS，包括截止日期雲端。如需詳細資訊，請參閱《 IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。
- 只安裝受信任、安全的軟體。
- 要求與身分識別提供者聯盟的使用者才能使用臨時認證進行存取 AWS 。
- 在截止日期雲端提交者程式檔案上使用安全權限，以防止竄改。
- 授予受信任的個人最低權限存取藝術家工作站。
- 只能使用您透過截止日期雲端監視器取得的提交者和介面卡。
- 限制工作站/etc/hosts和 Worker 主機作業系統上的資料表的權限和路由。
- 將工作站和 Worker 主機作業系統/etc/resolv.conf上的權限限制為。
- 定期修補操作系統和所有已安裝的軟件。這種方法包括專門與截止日期雲端配合使用的軟體，例如提交者、轉接器、背景工作者代理程式、OpenJD套件等。

監控 AWS 截止日期雲

監控是維護截止日期雲（ AWS 截止日期雲）和您的 AWS 解決方案的可靠性，可用性和性能的重要組成部分。從 AWS 解決方案的所有部分收集監控資料，以便在發生多點故障時，您可以更輕鬆地對多點失敗進行除錯。在您開始監視截止日期雲端之前，您應該建立一個監視計劃，其中包含下列問題的答案：

- 監控目標是什麼？
- 監控哪些資源？
- 監控這些資源的頻率為何？
- 將使用哪些監控工具？
- 誰將執行監控任務？
- 發生問題時應該通知誰？

AWS 截止日期雲端提供的工具可讓您用來監控資源並回應潛在事件。其中一些工具可以為您進行監視，某些工具需要手動干預。您應該盡可能自動化監控任務。

- Amazon 會即時 CloudWatch監控您的 AWS 資源和執行 AWS 的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以 CloudWatch 追蹤 Amazon EC2 執行個體的 CPU 使用率或其他指標，並在需要時自動啟動新執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

  截止日期雲有三個 CloudWatch 指標。
- Amazon CloudWatch 日誌可讓您從 Amazon EC2 執行個體和其他來源監控 CloudTrail、存放和存取日誌檔。 CloudWatch 記錄檔可以監控記錄檔中的資訊，並在符合特定臨界值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南。
- Amazon EventBridge 可用於自動化 AWS 服務並自動回應系統事件，例如應用程式可用性問題或資源變更。來自 AWS 服務的事件會以近乎即時 EventBridge 的方式傳送到。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。如需詳細資訊，請參閱 Amazon EventBridge 使用者指南。
- AWS CloudTrail擷取您帳戶或代表您 AWS 帳戶發出的 API 呼叫和相關事件，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 位址，以及呼叫發生的時間。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南。

主題

記錄呼叫 CloudTrail

AWS 截止日期雲與服務整合 AWS CloudTrail，可提供使用者、角色或 AWS 服務 在截止日期雲端中所採取的動作記錄的服務。 CloudTrail 將截止日期雲端的所有 API 呼叫擷取為事件。擷取的呼叫包括來自截止日期 Cloud 主控台的呼叫，以及對截止日期 Cloud API 作業的程式碼呼叫。

如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括截止日期雲端的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷向 Detection Cloud 提出的要求、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱使AWS CloudTrail 用者指南。

截止日期雲端資訊 CloudTrail

CloudTrail 在您創建帳戶 AWS 帳戶 時啟用。當活動在 Detection Cloud 中發生時，該活動會與事件歷史記錄中的其他 CloudTrail AWS 服務 事件一起記錄在事件中。您可以查看，搜索和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱檢視具有事 CloudTrail件記錄的事件。

CloudTrail 也會記錄使用者登入截止日期雲端監視並接收 AWS 認證時的事件。當使用者登入時，會出現包含來源`signin.amazonaws.com`和名稱的 CloudTrail 事件`UserAuthentication`。當從來源`sts.amazonaws.com`和名稱獲得登入使用者的 AWS 認證時，會發生第二個事件。`AssumeRole`使用者的 ID 會記錄在角色工作階段名稱中的第二個事件中。

如需正在進行的事件記錄 AWS 帳戶，包括截止日期雲端的事件，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤記錄來自 AWS 分區中所有區域的事件，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他， AWS 服務 以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。

如需詳細資訊，請參閱下列內容：

建立追蹤的概觀

CloudTrail 支援的服務與整合

設定 Amazon SNS 通知 CloudTrail

從多個區域接收 CloudTrail 記錄檔

從多個帳戶接收 CloudTrail 日誌文件

截止日期雲端支援將下列動作記錄為記 CloudTrail 錄檔中的事件：

- associate-member-to-farm
- associate-member-to-fleet
- associate-member-to-job
- associate-member-to-queue
- assume-fleet-role-for-讀
- assume-fleet-role-for-工人
- assume-queue-role-for-讀
- assume-queue-role-for-用戶
- assume-queue-role-for-工人
- 創建預算
- 創建農場
- create-fleet
- create-license-endpoint
- 創建監視器
- 創建隊列
- create-queue-environment
- create-queue-fleet-association
- create-storage-profile
- 創建工作者
- 刪除預算
- 刪除農場
- delete-fleet
- delete-license-endpoint

- delete-metered-product
- 刪除監視器
- 刪除佇列
- delete-queue-environment
- delete-queue-fleet-association
- delete-storage-profile
- 刪除工作者
- disassociate-member-from-farm
- disassociate-member-from-fleet
- disassociate-member-from-job
- disassociate-member-from-queue
- get-application-version
- 獲得預算
- 獲得農場
- get-feature-map
- 獲取艦隊
- get-license-endpoint
- 獲取監視器
- 獲取隊列
- get-queue-environment
- get-queue-fleet-association
- get-sessions-statistics-aggregation
- get-storage-profile
- get-storage-profile-for-隊列
- list-available-metered-products
- 列表預算
- list-farm-members
- 列表農場
- list-fleet-members
- 列表艦隊

- list-job-members
- list-license-endpoints
- list-metered-products
- 列表監視器
- list-queue-environments
- list-queue-fleet-associations
- list-queue-members
- 列表隊列
- list-storage-profiles
- list-storage-profiles-for-隊列
- list-tags-for-resource
- put-metered-product
- start-sessions-statistics-aggregation
- tag-resource
- untag-resource
- 更新預算
- 更新農場
- 更新艦隊
- 更新監視器
- 更新佇列
- update-queue-environment
- update-queue-fleet-association
- update-storage-profile
- 更新工作者

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 要求是使用根使用者登入資料還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項服務提出。

如需詳細資訊，請參閱使CloudTrail用者識別元素。

瞭解截止日期雲端記錄檔項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。 CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，包括有關請求的操作，動作的日期和時間，請求參數等信息。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

此 JSON 範例顯示呼叫 **`CreateFarm`** API 所產生的記錄檔：

```
{
    "eventVersion": "0",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
        "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE-PrincipalID",
                "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
                "accountId": "111122223333",
                "userName": "EXAMPLE-UserName"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-03-08T23:25:49Z"
            }
        }
    },
    "eventTime": "2021-03-08T23:25:49Z",
    "eventSource": "deadline.amazonaws.com",
    "eventName": "CreateFarm",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "EXAMPLE-userAgent",
    "requestParameters": {
        "displayName": "example-farm",
```

```
        "kmsKeyArn": "arn:aws:kms:us-west-2:111122223333:key/111122223333",
        "X-Amz-Client-Token": "12abc12a-1234-1abc-123a-1a11bc1111a",
        "description": "example-description",
        "tags": {
            "purpose_1": "e2e"
            "purpose_2": "tag_test"
        }
    },
    "responseElements": {
        "farmId": "EXAMPLE-farmID"
    },
    "requestID": "EXAMPLE-requestID",
    "eventID": "EXAMPLE-eventID",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
}
```


此範例顯示可協助您識別事件的地 AWS 區、IP 位址和其他 `requestParameters` "kmsKeyArn" (例如 "" 和 ")。`displayName`

使用監控 CloudWatch

Amazon CloudWatch (CloudWatch) 收集原始資料，並將其處理為可讀且近乎即時的指標。您可以在 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台，以檢視和篩選截止日期雲端指標。

- 在截止日期雲端客戶管理的叢集中， CloudWatch 會傳送兩個指標給您，`UnhealthyWorkerCount`並`RecommendedFleetSize`：
- 這些測量結果的命名空間為 `AWS/DeadlineCloud.`
- 您可以使用維度`farmID`和`fleetID`篩選量度。
- 這兩個量度都使用單位`count`。

這些統計資料會保留 15 個月，因此您可以存取歷史資訊，以更好地瞭解 Web 應用程式或服務的執行情況。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

截止日期雲端有兩種記錄檔 — 任務記錄檔和背景工作者記錄。任務日誌是當您以腳本或 DCC 運行時運行執行執行日誌。工作記錄可能會顯示事件，例如資產載入、拼貼彩現或找不到材質。

工作者記錄會顯示背景工作者代理程序 這些可能包括工作者代理程式啟動、自行註冊、報告進度、載入組態或完成工作等項目。

對於期限雲端，工作者會將這些記錄檔上傳至 CloudWatch 記錄。根據預設，記錄永遠不會過期。如果工作輸出大量資料，可能會產生額外費用。如需詳細資訊，請參閱 Amazon CloudWatch 定價。

您可以調整每個記錄群組的保留原則。較短的保留可移除舊的記錄檔，並有助於降低儲存成本。若要保留日誌，您可以在移除日誌之前將其存檔到 Amazon 簡單儲存服務。如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的使用主控台將日誌資料匯出到 Amazon S3。

> **Note**
>
> CloudWatch 記錄檔讀取受限於 AWS。如果您計劃邀請許多藝術家，我們建議您聯絡 AWS 客戶支援，並要求增加中的`GetLogEvents`配額 CloudWatch。此外，我們建議您在未偵錯時關閉記錄追蹤入口網站。

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的CloudWatch 日誌配額。

對事件採取行 EventBridge 動

截止日期雲端會將事件傳送 EventBridge 至 Amazon，以通知您服務狀態的變更。您可以使用 EventBridge 和這些事件來撰寫規則，以便在叢集發生變更時採取動作，例如通知您。有關更多信息，請參閱什麼是 Amazon EventBridge

車隊規模建議變更

當您將叢集設定為使用以事件為基礎的 auto 調整規模時，Deptionate Cloud 會傳送事件，供您用來管理叢集。這些事件中的每一個都包含叢集目前大小和要求大小的相關資訊。如需使用 EventBridge 事件和 Lambda 函數範例來處理事件的範例，請參閱使用截止日期雲端擴展建議功能自動擴展 Amazon EC2 叢集。

發生下列情況時，會傳送叢集大小建議變更事件：

- 建議的叢集大小發生變更`oldFleetSize`且與`newFleetSize`.

- 當服務偵測到實際的叢集大小與建議的叢集大小不符時。您可以從作GetFleet業回應中取得實際workerCount的叢集大小。當作用中的 Amazon EC2 執行個體無法註冊為截止日期雲端工作者時，可能會發生這種情況。

該事件的格式如下：

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "Fleet Size Recommendation Change",
    "source": "aws.deadline",
    "account": "111122223333",
    "time": "2017-12-22T18:43:48Z",
    "region": "us-west-1",
    "resources": [],
    "detail": {
        "farmId": "farm-12345678900000000000000000000000",
        "fleetId": "fleet-12345678900000000000000000000000",
        "oldFleetSize": 1,
        "newFleetSize": 5,
    }
}
```

下列欄位定義事件模式：

`"source": "aws.deadline"`

識別此事件的來源是截止日期雲端。

`"detail-type": "Fleet Size Recommendation Change"`

識別事件類型。

`"detail": { }`

提供叢集大小建議變更的相關資訊。

`"farmId": "farm-12345678900000000000000000000000"`

包含叢集之伺服器陣列的識別碼。

`"fleetId": "fleet-12345678900000000000000000000000"`

需要變更大小的叢集識別碼。

```
"oldFleetSize": 1
```

艦隊目前的規模。

```
"newFleetSize": 5
```

推薦新規模的艦隊。

的配額 Deadline Cloud

AWS Deadline Cloud 提供可用來處理工作的資源，例如伺服器陣列、叢集和佇列。當您建立時 AWS 帳戶，我們會為每個資源設定預設配額 AWS 區域。

Service Quotas 是一個集中的位置，您可以在其中查看和管理配額 AWS 服務。您也可以針對您使用的許多資源要求提高配額。

若要檢視的配額 Deadline Cloud，請開啟「Service Quotas」主控台。在導覽窗格中，選擇AWS 服務並選取Deadline Cloud。

若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。如果「Service Quotas」中尚未提供配額，請使用「增加服務配額」表單。

建立 AWS 截止日期雲端資源 AWS CloudFormation

AWS 截止日期雲端整合了這項服務 AWS CloudFormation，可協助您建立資源模型並設定 AWS 資源，以減少建立和管理資源和基礎架構的時間。您可以建立範本來描述所需的所有 AWS 資源 (例如伺服器陣列、佇列和叢集)，並為您 AWS CloudFormation 佈建和設定這些資源。

使用時 AWS CloudFormation，您可以重複使用範本，以一致且重複地設定您的截止日期雲端資源。描述您的資源一次，然後在多個區域中一遍又一遍地佈建相同 AWS 帳戶 的資源。

截止日期雲和 AWS CloudFormation 模板

若要佈建和設定截止日期雲端及相關服務的資源，您必須瞭解AWS CloudFormation 範本。範本是以 JSON 或 YAML 格式化的文本檔案。這些範本說明您要在 AWS CloudFormation 堆疊中佈建的資源。如果您不熟悉 JSON 或 YAML，可以使用 AWS CloudFormation 設計工具來協助您開始 AWS CloudFormation 使用範本。如需更多詳細資訊，請參閱 AWS CloudFormation 使用者指南 中的 什麼是 AWS CloudFormation 設計器？。

截止日期雲端支援在 AWS CloudFormation中建立伺服器陣列、佇列和叢集。如需詳細資訊，包括伺服器陣列、佇列和叢集的 JSON 和 YAML 範本範例，請參閱AWS CloudFormation 使用者指南中的AWS 截止日期雲端。

進一步了解 AWS CloudFormation

若要進一步了解 AWS CloudFormation，請參閱下列資源：

- AWS CloudFormation
- AWS CloudFormation 使用者指南
- AWS CloudFormation API 參考
- AWS CloudFormation 指令行介面使用者指南

截止日期 Cloud 使用者指南的文件歷程記錄

下表說明每個版本的AWS 截止日期雲端使用者指南中的重要變更。

變更	描述	日期
初始版本	這是截止日期雲端使用者指南的初始版本。	2024年4月2日

AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的AWS 詞彙表。

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。