



使用 Control-M 工作流程協調器與 AWS Mainframe Modernization 的整合

AWS 方案指引



AWS 方案指引: 使用 Control-M 工作流程協調器與 AWS Mainframe Modernization 的整合

Table of Contents

簡介	1
概觀	1
目標成果	3
本指南中的術語	3
前提	4
架構	6
與 Control-M 的受管 AWS Mainframe Modernization 整合	8
部署 Control-M 資源	8
部署 AWS Mainframe Modernization Control-M 外掛程式	8
在 Control-M 應用程式整合器中部署 AWS Mainframe Modernization 任務類型	9
建立連線設定檔	9
建立作業和排程	10
使用工作流程啟動任務	11
依排程自動執行作業	15
基本作業根據事件執行	15
監控作業	15
Control-M 監控	15
在 主控台上監控	16
AWS Mainframe Modernization 在 Amazon EC2 上與 Control-M 整合	18
部署資源	18
部署 Control-M 代理程式	18
部署 Micro Focus 外掛程式	8
部署作業類型	19
建立連線設定檔	19
建立作業和排程	21
使用工作流程啟動任務	22
依排程自動執行作業	25
監控作業	25
Control-M 監控	26
最佳實務	27
資源	28
貢獻者	29
文件歷史紀錄	30
.....	xxxix

使用 Control-M 工作流程協調器與 整合 AWS Mainframe Modernization

Amazon Web Services 和 BMC Software , Inc. ([貢獻者](#))

2024 年 2 月 ([文件歷史記錄](#))

Note

AWS Mainframe Modernization 服務 (受管執行期環境體驗) 不再開放給新客戶。對於類似 AWS Mainframe Modernization Service (受管執行期環境體驗) 的功能，請探索 AWS Mainframe Modernization Service (自我管理體驗)。現有客戶可以繼續正常使用該服務。如需詳細資訊，請參閱[AWS Mainframe Modernization 可用性變更](#)。

為了滿足業務和數位轉型的需求，客戶正在對大型主機應用程式進行現代化，以提高敏捷性、降低成本並加快創新速度。在 re：Invent 2021，Amazon Web Services (AWS) 宣布[AWS Mainframe Modernization](#)此服務，協助客戶現代化大型主機工作負載。AWS Mainframe Modernization 提供雲端原生的受管且高可用性的執行期環境 AWS。

本指南提供如何將 [BMC Helix Control-M](#) 工作流程協同運作與 AWS Mainframe Modernization 服務整合的詳細資訊，讓您在大型主機現代化旅程中為批次應用程式提供額外的現代化選項。透過使用 Control-M 的功能，組織可以使用提供端對端檢視的單一工具來合理化由現有和遷移的工作負載組成的環境。


概觀

Control-M 和 AWS Mainframe Modernization Replatform 與 Micro Focus 整合可提供下列優點：

- 有助於縮小人才差距
- 透過敏捷的 DevOps 方法支援快速創新
- 無需進行重大變更即可更輕鬆地存取應用程式和資料
- 優化執行或延伸應用程式的成本
- 最大化業務敏捷性並降低成本。

Replatform with Micro Focus AWS 服務可協助您將大型主機應用程式現代化為 AWS 雲端原生受管執行期環境。它提供工具和資源來協助您規劃和實作遷移和現代化。使用者可以提交或取消批次作業並檢閱批次作業執行的詳細資訊。每次使用者提交批次任務時，Replatform with Micro Focus AWS 服務都會建立個別的批次任務執行，以供監控。使用 AWS Mainframe Modernization 服務 Web 主控台，您可以依名稱搜尋批次任務、提供任務控制語言 (JCL)、指令碼檔案和參數給批次任務。

使用 Control-M，您可以定義、排程、管理和監控應用程式和資料工作流程。這增強了可見性和可靠性，有助於改進服務水準協議 (SLA)。Control-M 整合、自動化和協調內部部署和中的工作流程 AWS 雲端，以協助您按時交付商業服務。透過單一統一的圖形檢視，使用者可以使用豐富的外掛程式庫來協調所有工作流程，包括檔案傳輸、應用程式、資料來源和基礎設施。Control-M 在雲端中佈建，使用的暫時性功能 AWS 雲端。由於 Control-M 使用作業即程式碼方法，並在持續整合和持續交付 (CI/CD) 工具鏈中使用 REST API、JSON 和 Python，因此可以對 Control-M 工作流程進行版本控制、測試和維護，以便開發人員和 DevOps 工程師可以協同工作。

 Note

AWS Mainframe Modernization 也支援重構 AWS Blu Age。如需詳細資訊，請參閱 [AWS 方案指引指南](#)

目標業務成果

此解決方案可協助您達成下列目標。

商業利益

- 無縫遷移和工作流程協調 – Control-M 有助於大型主機工作負載順利轉移 AWS。它協調複雜的工作流程，並確保遷移期間和之後的批次處理持續性。
- 混合式操作管理 – Control-M 可在轉換至 期間有效管理舊版大型主機程序和新的雲端應用程式 AWS。
- 降低成本和最佳化資源使用 – AWS 使用 Control-M 遷移到 ，透過最佳化雲端資源使用和擴展來降低營運成本。
- 增強的敏捷性和創新能力 – 的遷移 AWS 可協助貴公司快速適應市場變化，並使用雲端原生服務進行創新。
- 合規、安全及高效監控 – Control-M 有助於確保持續合規和安全性。Control-M 也在雲端環境中提供增強的即時監控和報告功能。
- 減少人才差距 – 為 AWS 解決在舊版平台上尋找和保留熟練大型主機專業人員的相關挑戰而進行的遷移。

技術優勢

此解決方案可協助您實現下列目標：

- 擴展 Control-M 環境以管理 AWS Mainframe Modernization 服務的工作負載。
- 使用精靈將 Control-M REST APIs 快速整合到程序中，以支援快速創新。
- 在 Control-M 統一檢視中協調、排程和監控所有應用程式工作負載作業。
- 提高組織的可擴展性和業務敏捷性，讓您可以更有效地回應不斷變化的市場條件和客戶需求。
- 透過現代化大型主機工作負載並使用雲端原生服務的優勢，最佳化執行或擴展應用程式的成本。

本指南中的術語

- 大型主機應用程式是指完成和促進一組業務程序的一組相關的大型主機程式和子程式。大型主機應用程式可以是批次處理系統或線上交易處理 (OLTP) 系統。

- 批次作業是指設定為無需使用者互動即可執行的排定程式。在具有 Micro Focus AWS 的 Replatform 中，批次任務 JCL 檔案和批次任務二進位檔都準備在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中，而應用程式定義檔案中提供了兩者的位置。
- [AWS Mainframe Modernization](#) 是一種雲端原生服務，用於遷移、現代化、執行和操作大型主機應用程式。
- [Amazon S3](#) 是具有高耐用性、可用性和效能的可擴展物件儲存。
- [Amazon CloudWatch](#) 是一項為 DevOps 工程師、開發人員、網站可靠性工程 (SRE)、IT 經理和產品擁有者建置的監控和可觀測性服務。
- Control-M Web 是一個用於在建置、測試、部署、排程和監控業務應用程式工作流程的整個生命週期中管理企業工作負載 (包括大型主機批次作業) 的解決方案。一組廣泛的圖形和程式設計設施可讓所有使用者以他們感覺最舒服的方式存取 Control-M 設施。
- Control-M 的應用程式整合器是一種以 Web 為基礎的低程式碼設計工具，用於建立任務類型，以啟用應用程式或雲端服務與 Control-M 之間的整合。建置應用程式整合器任務類型後，它們的運作方式會與所有其他 Control-M 任務完全相同，公開目標整合的所有 Control-M 功能。在本指南中，整合是使用 AWS Replatform 搭配 Micro Focus。
- 規劃域是一組用於管理作業流程的建立和更新的功能。
- 監控域是一組用於管理作業流程的功能。您可以在此處檢視所有正在執行的作業及其狀態。如果發生錯誤，則會產生提醒。您可以執行操作動作來檢視作業狀態、對延遲或故障做出反應、分析問題並採取更正動作。
- Control-M 代理程式安裝在由 Control-M Server 管理的不同電腦上。作業指派給識別主機的代理程式名稱。代理程式使用 HTTPS 協定 (連接埠 443) 與伺服器通訊。

前提

本指南中的範例和圖表反映了下列假設：

- 正在遷移的大型主機應用程式可能會執行單一程式或多個程式。為簡單起見，本指南中的圖表顯示了每個應用程式的單一程式和多個子程式。
- 大型主機應用程式會在具有多個定義批次任務的 Replatform with Micro Focus AWS 受管執行期環境中遷移和執行。在此試行中，請依照[教學課程：Managed Runtime for Micro Focus](#) 中的指示，在 AWS Replatform with Micro Focus 中設定 BankDemo 範例應用程式。
- 完整的 [Control-M 安裝](#) 可用，具有包括應用程式外掛程式和附加元件在內的所有 Control-M 元件。
- [Control-M 代理程式](#) 負責管理作業。為了分配工作負載，可以將代理程式安裝在多台電腦上。這可以增強效能和恢復能力。

- [Control-M Automation API](#) 透過 Restful Web 服務 (REST API) 公開 Control-M 功能。作業、連線設定檔、使用者和角色以及網站標準等成品可以用 JSON 編寫，並作為各種服務的輸入提供或由各種服務產生為輸出。您可以使用 curl 或類似設施或提供的 ctm [命令列界面 \(CLI\)](#)，透過 HTTPS 請求直接存取服務。

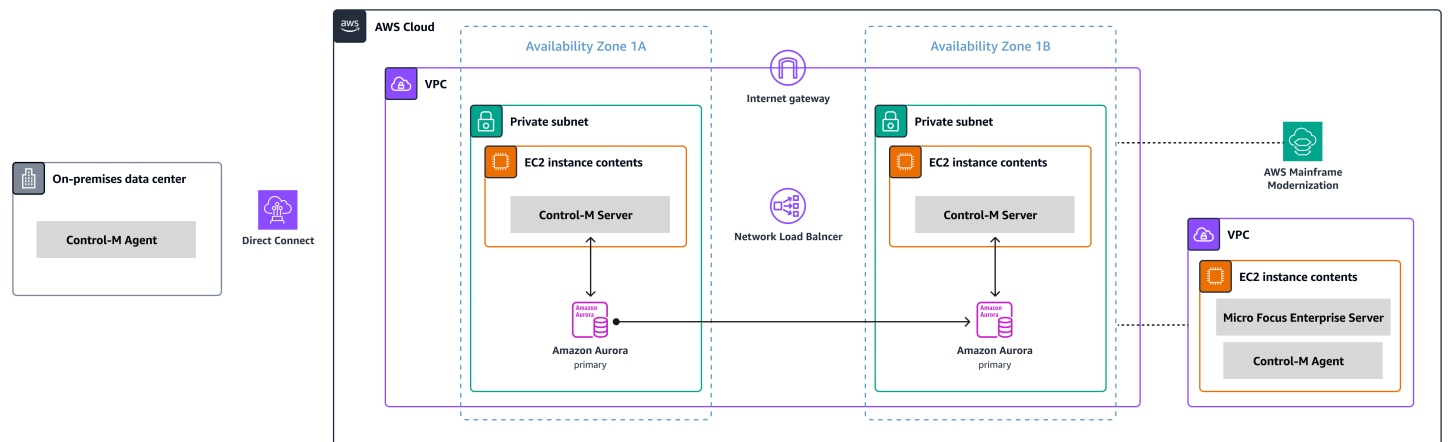
架構

AWS Mainframe Modernization 具有 Micro Focus 的 Replatform 有兩種模式：

- AWS Replatform with Micro Focus 是一種無伺服器受管執行期環境，使用 Micro Focus 後端動態部署，並由完全管理 AWS。AWS Replatform with Micro Focus 提供雲端原生 API 層，用於與 Micro Focus 互動。在此受管方法中，只有 Micro Focus 可供修改。UniKix 解決方案無法使用。
- AWS 在 Amazon Elastic Compute Cloud (Amazon EC2) 上使用 Micro Focus 的 Replatform 會以預先安裝 Micro Focus 環境的 Amazon Machine Image (AMI) 的形式交付，而您在您選擇的 EC2 執行個體類型上啟動。此自訂部署會直接公開原生 Micro Focus。

這兩種模式都包含交易管理員、資料映射工具、螢幕和映射讀取器，以及批次任務執行環境。您可以使用任一模式在分散式伺服器上執行大型主機應用程式，且原始碼的變更最少。

下圖顯示 Amazon EC2 執行個體上託管 Control-M 的工作流程整合。Amazon Aurora 資料庫用於維護管理和執行批次任務所需的資料。架構是多可用區域 (Multi-AZ) 部署，可提供高可用性。應用程式批次任務和資料會在 Replatform 中 AWS 與 Micro Focus 執行期環境協調。圖表顯示具有 Micro Focus 模式的兩種 Replatform：在 AWS Amazon EC2 上完全管理和自訂。



圖表顯示下列資源：

1. 在內部部署環境中，會安裝 Control-M Agent 來控制仍在 IBM Z/OS 或其他工作負載上執行的工作負載。在 x86 上執行的工作負載會透過 AWS Direct Connect 連線至 AWS 環境。
2. Control-M Server 安裝在多可用區域環境中主動-被動模式下的一對 EC2 執行個體上，以實現高可用性和災難復原。

3. Control-M 使用的 Amazon Aurora 後端資料庫（在 EC2 執行個體上執行）會在次要可用區域中使用複本部署，以實現高可用性和災難復原。
4. 單獨的 VPC 包含 EC2 AWS 執行個體，其具有以預先安裝 Micro Focus 環境的 AMI 形式交付的 Replatform with Micro Focus。Control-M Agent 安裝在此執行個體上，以與提供延伸任務管理功能的 Micro Focus 公用程式互動。

在遷移專案期間，您可能仍在管理大型主機和分散式伺服器上非 AWS 位置的工作負載。顯示的架構並非做為規範，而是提供一般方向。我們建議您在 Control-M 實作中建構詳細的組態，包括災難復原選項。

與 Control-M 的受管 AWS Mainframe Modernization 整合

本節說明 Control-M 如何與 整合，並支援在部署 Micro Focus 執行期引擎的受管 AWS Mainframe Modernization 環境中執行的批次任務。如果您要在 Amazon EC2 AWS 上實作具有 Micro Focus 環境的自訂 Replatform，請參閱 [AWS Mainframe Modernization Amazon EC2 上的 與 Control-M 整合](#) 一節。

本節假設下列先決條件：

- 作用中 AWS 帳戶。
- 大型主機應用程式會在具有多個定義批次任務的 AWS Replatform with Micro Focus 受管執行期環境中遷移和執行。
- 在此試行中，會設定 BankDemo 範例應用程式 AWS Mainframe Modernization。如需設定說明，請參閱 [教學課程：Micro Focus 的受管執行期](#)。

下列主題說明 Control-M 排程器與不同類型整合工作流程 AWS Mainframe Modernization 的環境之間整合所需的step-by-step設定：

- [部署 Control-M 資源](#)
- [為 AWS Mainframe Modernization 建立 Control-M 連線設定檔](#)
- [在 Control-M Planning 中建立作業和排程](#)
- [監控作業](#)

部署 Control-M 資源

整合 AWS Mainframe Modernization 與 Control-M 的前兩個步驟是部署外掛程式和任務類型。

部署 AWS Mainframe Modernization Control-M 外掛程式

外掛程式為 Control-M 協調的應用程式和服務提供整合和支援。針對 受管 AWS Mainframe Modernization 服務，部署 AWS Mainframe Modernization 外掛程式。

外掛程式部署是不常發生的活動。如果是第一次安裝外掛程式，請遵循 [Control-M 文件](#) 中的步驟。如果您有要使用的現有外掛程式，請略過此步驟，並 [建立連線設定檔](#)。

在 Control-M 應用程式整合器中部署 AWS Mainframe Modernization 任務類型

作業類型部署通常是一次性活動。如果您已具有要使用的現有作業類型，請略過此步驟，並繼續[建立連線設定檔](#)的下一步。

git 儲存庫中提供了作業類型範例 [AIJOB.ctmai](#)。若要部署任務類型，您必須使用[應用程式整合器](#)執行下列步驟：

1. 複製 [aws-mainframe-modernization-controlm-integration](#) GitHub 儲存庫，並將 AIJOB.ctmai 檔案下載到應用程式整合器可存取的檔案系統位置。
2. 登入 Application Integrator。
3. 在首頁標籤上，選取從檔案匯入作業類型，然後選取 AIJOB.ctmai 的位置。
4. 如果您要對提供的範例進行任何修改，請熟悉 Application Integrator。
5. 遵循 [Control-M 文件](#) 中的指示部署作業類型。

建立的 Control-M 連線設定檔 AWS Mainframe Modernization

連線設定檔定義應用程式特定執行個體的連線屬性和安全憑證。每個連線設定檔都可以被多個作業參考。您可以為應用程式和憑證的每個唯一組合建立不同的設定檔。

若要[定義連線設定檔](#)，您可以使用 Control-M Web 介面的組態網域中可用的圖形使用者介面 (GUI)，也可以使用 [JSON](#)。如需 AWS Mainframe Modernization 外掛程式連線設定檔的相關資訊，請參閱 [Control-M 文件](#)。

下列程式碼是使用 JSON 的範例：

```
{
  "MANAGED-M2-REPLATFORM": {
    "Type": "ConnectionProfile:AWS Mainframe Modernization",
    "Mainframe Modernization URL": "https://m2.{{AwsRegion}}.amazonaws.com",
    "Connection Timeout": "30",
    "AWS Region": "us-west-2",
    "Authentication": "NoSecret",
    "IAM Role": "--- IAM Role name ---",
    "AWS Logs URL": "https://logs.{{AwsRegion}}.amazonaws.com",
    "Description": "",
    "Centralized": true
  }
}
```

```
}
```

建立類似範例的 JSON 檔案，並使用 [Control-M Automation API 部署服務進行部署](#)。例如，如果 JSON 程式碼儲存至名為 `cp-MANAGED-M2-REPLATFORM.json` 的檔案，則部署此連線設定檔的 `ctm` CLI 語法如下：

```
ctm deploy cp-MANAGED-M2-REPLATFORM.json
```

Control-M Automation API 的回應類似下列內容：

```
[
  {
    "deploymentFile": "cp-Managed-M2-REPLATFORM.json",
    "deploymentState": "DEPLOYED_CONNECTION_PROFILES",
    "deploymentStatus": "ENDED_OK",
    "successfulFoldersCount": 0,
    "successfulSmartFoldersCount": 0,
    "successfulSubFoldersCount": 0,
    "successfulJobsCount": 0,
    "successfulConnectionProfilesCount": 1,
    "successfulDriversCount": 0,
    "isDeployDescriptorValid": false,
    "deployedConnectionProfiles": [
      "MANAGED-M2-REPLATFORM"
    ]
  }
]
```

在 Control-M Planning 中建立作業和排程

現在您已部署任務類型和 AWS 連線的連線設定檔，您可以開始建立和執行任務。

每個 AWS Mainframe Modernization 服務任務由一組屬性組成，分為四個部分。每個部分都可以具有許多屬性。下列清單顯示一些較常用的屬性。

- 一般：
 - 作業名稱
 - 作業所屬的應用程式和子應用程式
 - 要提交的 JCL
 - 文件的連結

- 排程：
 - 此作業有資格執行的月份和天數
 - 行事曆，例如商業會計期間、假日或其他無法以演算法定義的特殊日期
 - 時段
 - 循環行為
- 事前準備：
 - 上游相依性 (通常為作業，必須先成功完成此作業才有資格執行)
 - 可能需要的資源
 - 可能需要的使用者動作
- Control-M 完成作業後執行的動作：
 - 確定任務成功或失敗 (通常根據任務的完成代碼，但您可以覆寫該設定，以使用輸出文字或檢查特定狀態)
 - 失敗或成功的通知，例如電子郵件
 - 下游相依性的發佈狀態

與連線設定檔一樣，您可以在[在 GUI 中](#)或[在 JSON 中](#)建立和部署作業，並使用 Control-M Automation API 進行部署。

下列各節會檢查一些常見的工作流程案例：

- [根據先前作業的狀態碼啟動作業](#)
- [以排定的頻率自動執行作業](#)
- [基本作業根據事件執行](#)

根據先前作業的狀態啟動作業

建立稱為工作流程的作業流程。在成功完成先前作業後，工作流程中的作業與相依性相互關聯。

使用 Control-M Web GUI

若要從 Control-M 使用者介面啟動任務，請執行下列動作：

1. 在規劃網域中，新增工作區。這將開啟一個具有空資料夾物件的畫布。
2. 選取 AWS Mainframe Modernization 任務類型 (如果您使用提供的任務類型範本，這稱為 [M2JOB](#))，並將其拖曳到資料夾。

3. 在任務類型的顏色變更為綠色後，將其捨棄。右側的窗格包含一般、排程、先決條件和動作區段。若要建立任務，請參閱 [Control-M 文件](#) 中的標準說明。
4. 接著，您需要任務名稱值來設定任務類型。您可以在主控台的應用程式定義畫面上 AWS Mainframe Modernization 或執行 [ListBatchJobDefinitions API](#) 來尋找任務名稱值。在此試行中，針對數個任務重複步驟 2–4，以您選擇的任何名稱命名每個任務。範例名稱為 CBANK、CURRENCY、INVFUNDS、BROKERAGE、RISKMGMT 和 BANK-SERVICE-Managed-M2。
5. 若要將這些任務連接到您想要的流程，請選擇任務物件下方的條件三角形，然後將其拖曳至下一個任務。例如，選擇下方的條件三角形 CBANK，然後將其拖曳至 CURRENCY。此動作會成為的前 CBANK 身 CURRENCY。根據預設，CBANK 現在必須成功完成，CURRENCY 才有資格執行。

下列螢幕擷取畫面顯示基本任務流程的 Control-M Planning 檢視。

The screenshot displays the Control-M Planning interface. On the left, a job flow diagram for 'jog-managed-m2' shows a sequence of jobs: CBANK (highlighted in blue) flows into CURRENCY (highlighted in orange), which then branches into three parallel jobs: BROKERAGE, INVFUNDS, and RISKMGMT. These three jobs then converge into a final job: BANK-SERVICE-Managed-M2. On the right, the configuration panel for the 'CBANK' job is visible, showing details such as Job type (AWS Mainframe Modernization), Job name (CBANK), Description, Run as dummy (unchecked), Where (Server: psctm, Host/Host group: ctm-worker), What (Connection profile: MANAGED-M2-REPLATFORM, Application Name, Action: Start Batch Job, JCL Name: iefbr14, Retrieve CloudWatch Logs: checked, Status Polling Frequency: 15, Failure Tolerance: 3, Parent folder: jog-managed-m2, Application: AWSM2, Sub Application: Replatform-Managed), Variables, and Advanced settings (Run job on all hosts..., End folder on job c..., Created by: ctmck).

影像由 BMC Software, Inc. 提供 ©2022

使用 JSON

相同的流程可以用 JSON 編碼：

```
{
  "Defaults": {
    "Application": "AWSM2",
```

```
"SubApplication": "Replatform-Managed",
"Job": {
  "Host": "ctm-worker",
  "Output": {}
}
},
"jog-managed-m2": {
  "Type": "Folder",
  "ControlmServer": "psctm",
  "OrderMethod": "Manual",
  "SiteStandard": "_z_DemoBusinessFlows",
  "CBANK": {
    "Type": "Job:AWS Mainframe Modernization",
    "ConnectionProfile": "MANAGED-M2-REPLATFORM",
    "JCL Name": "iefbr14",
    "Retrieve CloudWatch Logs": "checked",
    "Action": "Start Batch Job",
    "Application Version": "1"
  },
  "CURRENCY": {
    "Type": "Job:AWS Mainframe Modernization",
    "ConnectionProfile": "MANAGED-M2-REPLATFORM",
    "JCL Name": "iefbr14",
    "Retrieve CloudWatch Logs": "checked",
    "Action": "Start Batch Job",
    "Application Version": "1"
  },
  "BROKERAGE": {
    "Type": "Job:AWS Mainframe Modernization",
    "ConnectionProfile": "MANAGED-M2-REPLATFORM",
    "JCL Name": "iefbr14",
    "Retrieve CloudWatch Logs": "checked",
    "Action": "Start Batch Job",
    "Application Version": "1"
  },
  "INVFUNDS": {
    "Type": "Job:AWS Mainframe Modernization",
    "ConnectionProfile": "MANAGED-M2-REPLATFORM",
    "JCL Name": "iefbr14",
    "Retrieve CloudWatch Logs": "checked",
    "Action": "Start Batch Job",
    "Application Version": "1"
  },
  "RISKMGMT": {
```

```
"Type": "Job:AWS Mainframe Modernization",
"ConnectionProfile": "MANAGED-M2-REPLATFORM",
"JCL Name": "iefbr14",
"Retrieve CloudWatch Logs": "checked",
"Action": "Start Batch Job",
"Application Version": "1"
},
"BANK-SERVICE-Managed-M2": {
  "Type": "Job:SLAMangement",
  "ServiceName": "Bank Service - Managed M2",
  "RunAs": "ctmagent",
  "CompleteBy": {
    "Time": "12:00",
    "Days": "0"
  }
},
"leftbranch": {
  "Type": "Flow",
  "Sequence": [
    "CURRENCY",
    "RISKMGMT",
    "BANK-SERVICE-Managed-M2"
  ]
},
"middlebranch": {
  "Type": "Flow",
  "Sequence": [
    "CBANK",
    "CURRENCY",
    "INVFUNDS",
    "BANK-SERVICE-Managed-M2"
  ]
},
"rightbranch": {
  "Type": "Flow",
  "Sequence": [
    "CURRENCY",
    "BROKERAGE",
    "BANK-SERVICE-Managed-M2"
  ]
}
}
```

若要部署此流程，請使用 部署服務：

```
ctm deploy folder-MANAGED-M2-REPLATFORM.json
```

以排定的頻率自動執行作業

使用您在先前步驟中建立的流程，您可以新增基本和執行期排程。

- 基本排程定義作業可以執行的天數 (例如，每個工作日、僅在工作日、月底或季末)。
- 執行時間排程會決定任務何時會在其有資格執行的日期執行 (例如，每小時、指定資源可用後，或僅在手動確認後)。

您可以在排程標籤上設定基本和執行期排程。

基本作業根據事件執行

Control-M Managed File Transfer (MFT) 是一種 FTP/SFTP 用戶端和伺服器，可用於在本機主機與遠端主機之間監看和傳輸檔案。如需有關定義檔案傳輸作業的詳細資訊，請參閱 [Control-M 文件](#)。

此試驗使用檔案傳輸作業來監看名為 bmc-poc-bucket 的 S3 儲存貯體的 /bmcfile 資料夾中副檔名為 .poc 的檔案的檔案建立事件。在該事件發生時，系統會啟動 Control-M 作業以執行下一個作業。您可以選擇性地傳遞完整路徑，包括儲存貯體名稱。

監控作業

您可以監控和驗證 Control-M Monitoring 網域和內任務的處理 AWS 管理主控台，確保在這些平台上進行全面的觀察和驗證。

Control-M 監控

您可以在 Control-M 監控域中監控作業提交和執行。根據預設，AWS Mainframe Modernization 服務任務將與所有其他 Control-M 工作一起顯示。如果只想查看沒有任何其他工作負載 (或任何其他篩選需求) AWS Mainframe Modernization 的服務任務，您可以建立檢視點。

視點不僅顯示作業資訊，還顯示與上游和下游相依性的關係。此外，如果您的工作流程包含 AWS Mainframe Modernization 和其他 Control-M 任務類型，您可以在監控網域中查看和管理整個流程。

若要遵循詳細步驟，請參閱 Control-M 文件中的 [監控的檢視點一節](#)。

下列螢幕擷取畫面顯示兩個工作流程的輸出。在左側，工作流程會以綠色成功完成所有任務。在右側，工作流程僅部分成功，因為任務CURRENCY傳回失敗狀態，以紅色表示。工作流程在那裡停止，將剩餘的任務保留在等待排程狀態。

The screenshot displays two workflow execution instances for 'jog-managed-m2'. The left instance shows a successful execution path: CBANK (Start: 8/30/2023, 11:25:21 PM; End: 8/30/2023, 11:26:10 PM) → CURRENCY (Start: 8/30/2023, 11:26:11 PM; End: 8/30/2023, 11:27:16 PM) → BROKERAGE (Start: 8/30/2023, 11:27:17 PM; End: 8/30/2023, 11:28:24 PM), INVFUNDS (Start: 8/30/2023, 11:27:17 PM; End: 8/30/2023, 11:28:22 PM), and RISKMGMT (Start: 8/30/2023, 11:27:17 PM; End: 8/30/2023, 11:28:22 PM) → BANK-SERVICE-Managed -M2 (Start: 8/30/2023, 11:28:26 PM; End: 8/30/2023, 11:28:26 PM). The right instance shows a failed execution path: CBANK (Start: 8/30/2023, 11:29:10 PM; End: 8/30/2023, 11:30:16 PM) → CURRENCY (Start: 8/30/2023, 11:30:17 PM; End: 8/30/2023, 11:30:19 PM) → BROKERAGE (Start: 8/30/2023, 11:35:40 PM; End: 8/30/2023, 11:40:40 PM), INVFUNDS (Start: 8/30/2023, 11:35:40 PM; End: 8/30/2023, 11:40:40 PM), and RISKMGMT (Start: 8/30/2023, 11:35:40 PM; End: 8/30/2023, 11:40:40 PM) → BANK-SERVICE-Managed -M2 (Start: 8/30/2023, 11:40:40 PM; End: 8/30/2023, 11:40:59 PM). The CURRENCY task in the right instance is highlighted in red, indicating a failure.

The configuration panel for the 'CBANK' job is shown on the right. It includes the following details:

- Job type: AWS Mainframe Modernization
- Job name: CBANK
- Description: [Redacted]
- Run as dummy: [Unchecked]
- Where: Server: psctm, Host/Host group: ctm-worker
- What: Connection profile: MANAGED-M2-REPLATFORM, Application Name: M2-Managed-via-CF-4fa2c710, Action: Start Batch Job, JCL Name: CBANKjcl, Retrieve CloudWatch Logs: [Checked], Status Polling Frequency: 15, Failure Tolerance: 3
- Parent folder: jog-managed-m2
- Application: AWSM2
- Sub Application: Replatform-Managed

The Variables section is also visible, showing a table of local variables:

Type	Pool Name	Name	Value	String
Local		RUN-UCM-L...	/aws/vende ...	%%RUN-UCM-LO
Local		RUN-UCM- ...	ohr3x55djz ...	%%RUN-UCM-API
Local		RUN-UCMj ...	["CBANKjcl"]	%%RUN-UCM-JCL
Local		RUN-UCM-R...	0000	%%RUN-UCM-RE
Local		RUN-UCM-I ...	00001000	%%RUN-UCM-INF

影像由 BMC Software, Inc. 提供 ©2022

在 主控台上監控

若要檢視任務和登入資訊 AWS，請登入 AWS 管理主控台，然後導覽至 [AWS Mainframe Modernization 主控台](#)。

The screenshot displays the AWS Mainframe Modernization console interface for the 'Managed-M2-Replatform-Application'. The breadcrumb navigation shows 'AWS Mainframe Modernization > Applications > Managed-M2-Replatform-Application'. The page title is 'Managed-M2-Replatform-Application' with an 'Info' link and an 'Actions' dropdown menu. Below the title, there are tabs for 'Definition', 'Batch jobs', 'Data sets', 'Versions', 'Monitoring', and 'Tags'. The 'Batch jobs' tab is active, showing 'Batch job executions (7)'. A search bar is present with the placeholder text 'Filter jobs by attributes or search by keyword'. Below the search bar is a table with columns for 'Job ID', 'Status', and 'Job name'. All jobs listed have a status of 'Succeeded'.

Job ID	Status	Job name
J0001004	Succeeded	INVFUNDS
J0001002	Succeeded	CBANK
J0001000	Succeeded	CBANK
J0001003	Succeeded	RISKMGMT
J0001006	Succeeded	BROKRAGE
J0001005	Succeeded	BROKRAGE
J0001001	Succeeded	CBANK

此檢視不包含相依性，也不包含非由 AWS Mainframe Modernization 服務管理的任何工作負載。

AWS Mainframe Modernization 在 Amazon EC2 上與 Control-M 整合

本節說明 Control-M 如何與部署在 EC2 執行個體上的自訂 AWS Mainframe Modernization 執行期環境中執行的批次任務整合並提供支援。如果您要使用 Micro Focus 執行時間環境實作全受管 AWS Replatform，請參閱[受管與 Control-M 的 AWS Mainframe Modernization 整合](#)一節。

本節假設下列先決條件：

- 作用中的 AWS 帳戶
- 虛擬私有雲端 (VPC)，其中將建立 EC2 執行個體。
- 大型主機應用程式會在 EC2 執行個體上具有 Micro Focus 的 AWS Replatform 環境中遷移和執行，並支援具有多個定義批次任務的 Micro Focus 執行期引擎。在此試行中，請遵循[使用 Micro Focus 複寫應用程式](#)的指示。文件包含有關在 Amazon EC2 上使用 Micro Focus 執行期環境設定和操作 AWS Replatform 的所有任務和其他資訊。

下列主題涵蓋 Control-M 與 AWS Replatform 與 Micro Focus 環境整合所需的設定：

- [部署 Control-M 和 Micro Focus 資源](#)
- [建立 Control-M 連線設定檔](#)
- [在 Control-M Planning 中建立作業和排程](#)
- [使用監控在 Control-M 中管理任務執行](#)

部署 Control-M 和 Micro Focus 資源

整合 AWS Mainframe Modernization 與 Control-M 的前兩個步驟是部署 Control-M Agent，然後在 Agent 上部署外掛程式。第三個步驟是在 Control-M 應用程式整合器中部署任務類型。

在 EC2 執行個體上部署 Control-M 代理程式

搭配 Micro Focus on Amazon EC2 執行期環境使用自訂 AWS Replatform 時，會叫用 MFBSIFJCL 公用程式。公用程式會在使用 Micro Focus Enterprise Server 優惠啟動的主機上運作 AWS Marketplace。若要執行 MFBSIFJCL 公用程式，您還必須在該主機上部署 Control-M 代理程式。如需說明，請參閱[Control-M 文件](#)。

Note

您可以從 [BMC Electronic Product Download 網站](#) 下載必要的安裝媒體。

在 Control-M Agent 上部署 Micro Focus 外掛程式

外掛程式為 Control-M 協調的應用程式和服務提供整合和支援。

外掛程式部署是不常發生的活動。如果您已經有要使用的現有外掛程式，請略過此步驟，並建立連線設定檔。

在 Amazon EC2 上，使用 Micro Focus AWS Mainframe Modernization 的 Replatform 服務會公開 Micro Focus 引擎。為了整合 AWS Mainframe Modernization，Control-M 使用 Micro Focus 外掛程式。如需詳細資訊，請參閱 [Control-M 文件](#)。

您必須將此外掛程式部署到安裝在執行 Micro Focus Enterprise Server 之主機上的代理程式。

在 Control-M 應用程式整合器中部署 AWS Mainframe Modernization 任務類型

作業類型部署通常是一次性活動。如果您已具有要使用的現有作業類型，請略過此步驟，並繼續 [建立連線設定檔](#) 的下一步。

git 儲存庫中提供了作業類型範例 [AIJOB.ctmai](#)。若要部署任務類型，您必須使用 [應用程式整合器](#) 執行下列步驟：

- 複製 [aws-mainframe-modernization-controlm-integration](#) GitHub 儲存庫，並將 AIJOB.ctmai 檔案下載到應用程式整合器可存取的檔案系統位置。
- 登入 Application Integrator。
- 在首頁標籤上，選取從檔案匯入作業類型，然後選取 AIM2JOB.ctmai 的位置。
- 如果您要對提供的範例進行任何修改，請熟悉 Application Integrator。
- 遵循 [Control-M 文件](#) 中的指示部署作業類型。

建立 Control-M 連線設定檔

連線設定檔定義應用程式特定執行個體的連線屬性和安全憑證。每個連線設定檔都可以被多個作業參考。您可以為應用程式和憑證的每個唯一組合建立不同的設定檔。

若要定義連線設定檔，您可以使用 Control-M Web 介面的[組態網域](#)，也可以使用 JSON。下列程式碼是使用 JSON 的範例：

```
{
  "MICROFOCUS-WINDOWS": {
    "Type": "ConnectionProfile:Micro Focus Windows",
    "Centralized": true,
    "Description": "Micro Focus on Windows Connection Profile - file locations refer to the Enterprise Server host", "MFBSI Config Path": "C:\\microfocus\\ES\\mfbsi\\MFWIN\\mfbsi.cfg",
    "MFBSI Directory Path": "c:\\microfocus\\es\\mfbsi\\MFWIN",
    "Runtime Environment": "\\\"C:\\Program Files (x86)\\Micro Focus\\Enterprise Developer\\createenv.bat\\\"", "Run As": "dbauser",
    "RunAs-Pass": "*****"
  }
}
```

範例程式碼位於 GitHub 儲存庫 `ConnectionProfile-Custom-M2-Replatform.json` 的檔案中。若要部署程式碼，請使用 Control-M Automation API 部署服務：

```
ctm deploy ConnectionProfile-Custom-M2-Replatform.json
```

Control-M Automation API 的回應將類似於以下內容：

```
[
  {
    "deploymentFile": "cp-JOG-MF-WINDOWS.json",
    "deploymentState": "DEPLOYED_CONNECTION_PROFILES",
    "deploymentStatus": "ENDED_OK",
    "successfulFoldersCount": 0,
    "successfulSmartFoldersCount": 0,
    "successfulSubFoldersCount": 0,
    "successfulJobsCount": 0,
    "successfulConnectionProfilesCount": 1,
    "successfulDriversCount": 0,
    "isDeployDescriptorValid": false,
    "deployedConnectionProfiles": [ " MICROFOCUS-WINDOWS " ]
  }
]
```

在 Control-M Planning 中建立作業和排程

現在您已部署 外掛程式和 Micro Focus 連線的連線設定檔，您可以開始建立和執行任務。

每個 Control-M for Micro Focus 任務都由一組屬性組成，分為四個部分。每個部分都可以具有許多屬性。下列清單顯示一些較常用的屬性。

- 一般：
 - 作業名稱
 - 作業所屬的應用程式和子應用程式
 - 要提交的 JCL
 - 網站提供的操作文件連結
- 排程：
 - 此作業有資格執行的月份和天數
 - 行事曆，例如商業會計期間、假日或其他無法以演算法定義的特殊日期
 - 時段
 - 循環行為，例如每小時執行一次
- 先決條件
 - 上游相依性 (通常為作業，必須先成功完成此作業才有資格執行)
 - 可能需要的資源
 - 可能需要的使用者動作
- Control-M 完成作業後執行的動作：
 - 確定任務成功或失敗 (通常根據任務的完成代碼，但您可以覆寫該設定，以使用輸出文字或檢查特定狀態)
 - 失敗或成功的通知，例如電子郵件
 - 下游相依性的發佈狀態

如同連線設定檔，任務可以使用 [Control-M Web](#) 建立和部署，或以 [JSON 撰寫](#)，並使用 Control-M Automation API 部署。

下列各節會檢查一些常見的工作流程案例：

- [使用工作流程啟動任務](#)
- [以排定的頻率自動執行作業](#)

使用工作流程啟動任務

建立稱為工作流程的作業流程。在成功完成先前作業後，工作流程中的作業與相依性相互關聯。

使用 Control-M Web

若要從 Control-M 使用者介面啟動任務，請執行下列動作：

1. 在規劃域中，新增一個新的工作區。這將開啟一個具有空資料夾物件的畫布。
2. 在選單列上，選擇新增任務，選擇 Micro Focus Windows 或 Micro Focus Linux 任務，然後將其拖曳到資料夾的臉部。
3. 在任務的顏色變更為綠色後，將其捨棄。右側的窗格包含一般、排程、先決條件和動作區段。若要建立任務，請參閱 [Control-M 文件](#) 中的標準說明。
4. 如需建置 Control-M for Micro Focus 任務的說明，請參閱 [Control-M 文件](#)。對於此試驗，請對五個任務重複步驟 2–4，將每個任務命名aws-mf-job1為 aws-mf-job5。
5. 若要將這些任務連接到您想要的流程，請選擇任務物件下方的條件三角形，然後將其拖曳至下一個任務。例如，選擇 下方的條件三角形aws-mf-job1，然後將其拖曳至 aws-mf-job2。此動作會成為 的前aws-mf-job1身aws-mf-job2。根據預設，現在aws-mf-job1必須成功完成，aws-mf-job2才有資格執行。

下圖顯示了基本作業流程的 Control-M Planning 檢視。



影像由 BMC Software, Inc. 提供 ©2022

使用 JSON

相同的流程可以 JSON 編碼。在 GitHub 儲存庫中，範例程式碼位於 檔案 Folder-Custom-M2-Replatform.json：

```
{
  "aws-custom-m2": {
    "Type": "Folder",
    "ControlmServer": "psctm",
    "OrderMethod": "Manual",
    "Application": "AWSM2",
    "SubApplication": "Replatform-Custom",
    "aws-mf-job1": {
      "Type": "Job:Micro Focus Windows",
      "ConnectionProfile": "MICROFOCUS-WINDOWS",
      "Enable JCL Variables": "unchecked",
      "Restart on Rerun": "unchecked",
      "Recapture ABEND Codes": "Ignore",
      "Recapture COND Codes": "Ignore",
      "Auto Adjust Restart": "Ignore",
      "Set MF_UCC11": "Ignore",
      "Restart with Modified JCL": "No",
      "Application": "AWSM2",
      "SubApplication": "Replatform-Custom",
      "Host": "microfocus-es-host",
      "Output": {}
    },
    "aws-mf-job2": {
      "Type": "Job:Micro Focus Windows",
      "ConnectionProfile": "MICROFOCUS-WINDOWS",
      "Enable JCL Variables": "unchecked",
      "Restart on Rerun": "unchecked",
      "Recapture ABEND Codes": "Ignore",
      "Recapture COND Codes": "Ignore",
      "Auto Adjust Restart": "Ignore",
      "Set MF_UCC11": "Ignore",
      "Restart with Modified JCL": "No",
      "Application": "AWSM2",
      "SubApplication": "Replatform-Custom",
      "Host": "microfocus-es-host",
      "Output": {}
    },
    "aws-mf-job3": {
```

```
"Type": "Job:Micro Focus Windows",
"ConnectionProfile": "MICROFOCUS-WINDOWS",
"Enable JCL Variables": "unchecked",
"Restart on Rerun": "unchecked",
"Recapture ABEND Codes": "Ignore",
"Recapture COND Codes": "Ignore",
"Auto Adjust Restart": "Ignore",
"Set MF_UCC11": "Ignore",
"Restart with Modified JCL": "No",
"Application": "AWSM2",
"SubApplication": "Replatform-Custom",
"Host": "microfocus-es-host",
"Output": {}
},
"aws-mf-job4": {
  "Type": "Job:Micro Focus Windows",
  "ConnectionProfile": "MICROFOCUS-WINDOWS",
  "Enable JCL Variables": "unchecked",
  "Restart on Rerun": "unchecked",
  "Recapture ABEND Codes": "Ignore",
  "Recapture COND Codes": "Ignore",
  "Auto Adjust Restart": "Ignore",
  "Set MF_UCC11": "Ignore",
  "Restart with Modified JCL": "No",
  "Application": "AWSM2",
  "SubApplication": "Replatform-Custom",
  "Host": "microfocus-es-host",
  "Output": {}
},
"aws-mf-job5": {
  "Type": "Job:Micro Focus Windows",
  "ConnectionProfile": "MICROFOCUS-WINDOWS",
  "Enable JCL Variables": "unchecked",
  "Restart on Rerun": "unchecked",
  "Recapture ABEND Codes": "Ignore",
  "Recapture COND Codes": "Ignore",
  "Auto Adjust Restart": "Ignore",
  "Set MF_UCC11": "Ignore",
  "Restart with Modified JCL": "No",
  "Application": "AWSM2",
  "SubApplication": "Replatform-Custom",
  "Host": "microfocus-es-host",
  "Output": {}
},
```

```
"leftbranch": {
  "Type": "Flow",
  "Sequence": [
    "aws-mf-job1",
    "aws-mf-job2",
    "aws-mf-job3",
    "aws-mf-job5"
  ]
},
"rightbranch": {
  "Type": "Flow",
  "Sequence": [
    "aws-mf-job2",
    "aws-mf-job4",
    "aws-mf-job5"
  ]
}
}
```

若要部署此流程，請使用 Control-M Automation API：

```
ctm deploy Folder-Custom-M2-Replatform.json
```

以排定的頻率自動執行作業

使用您在先前步驟中建立的流程，您可以新增基本和執行期排程。

- 基本排程會定義任務符合執行資格的天數（例如，每個工作日、僅限工作日、月底或季末）。您可以在排程索引標籤上設定基本[排程](#)。
- 執行時間排程會決定任務在符合執行資格的當天執行的時間（例如，前一任務完成後、每小時、指定資源可用後，或僅在手動確認後）。

您可以在排程區段中定義一些執行時間排程，在先決條件區段中定義其他執行時間排程。

監控作業

您可以在 Control-M 監控網域和 [Micro Focus Enterprise Server Common Web Administration 使用者介面](#) 中監控和驗證任務。

Control-M 監控

您可以在 Control-M 監控域中監控作業提交和執行。根據預設，AWS Mainframe Modernization 服務任務將與所有其他 Control-M 工作一起顯示。如果只想查看沒有任何其他工作負載（或任何其他篩選需求）AWS Mainframe Modernization 的服務任務，您可以建立檢視點。

視點不僅顯示作業資訊，還顯示與上游和下游相依性的關係。此外，如果您的工作流程包含 AWS Mainframe Modernization 和其他類型的 Control-M 任務，您可以在監控網域中查看和管理整個流程。

您可以前往 Control-M 文件中的[監控的檢視點區段](#)，以遵循詳細步驟。

下列螢幕擷取畫面顯示兩個工作流程的輸出。在左側，工作流程已成功完成，所有五個作業都呈現綠色。在右側，工作流程僅部分成功，因為 aws-mf-job3 傳回失敗狀態，而工作流程停止了該狀態，並保持 aws-mf-job5 等待排程狀態。

The screenshot displays the Control-M monitoring interface. On the left, two workflow diagrams are shown:

- aws-custom-m2:** A successful workflow where all five jobs (aws-mf-job1 to aws-mf-job5) are shown with green status icons and completed timestamps.
- aws-custom-m2-with-failure:** A workflow where aws-mf-job3 is shown with a red status icon and a failed timestamp, while aws-mf-job5 is shown with a grey status icon and a pending timestamp.

On the right, the log output for 'aws-mf-job2' is displayed, showing job execution details and a list of steps:

```

aws-mf-job2
Summary Job Settings Micro Focus Windows Log Output Statistics Script Documentation Related
00001, 8/8/2023, 6:27:00 PM, 3994, 0
Find
JCLS100011 MFBSI Version 7.0.00 Copyright (C) 2013-2021 Micro Focus. All rights reserved. 20220513_ED70PU8
JCLS100021 Running environment: (at 2023/08/09 02:27:55) - MFBSI_DIR: c:\microfocus\es\mfbsi\MFBSI [CAScmd]
JCLS100431 Original JCL member: "C:\microfocus\es\jcl\iefbr14.jcl"
JCLS100441 Substituted JCL in : "c:\microfocus\es\mfbsi\MFBSI\jcltemp\iefbr14.jcl.000000000504@v1-aus-ctm-vy4nda"
JCLS100031 Submit: CASSUB.exe -IMFWIN -x"c:\microfocus\es\mfbsi\MFBSI\jcltemp\iefbr14.jcl.000000000504@v1-aus-ctm-vy4nda"
JCLS100011 JCLCM01871 10001139 JOBA JOB SUBMITTED (JOBNAME=JOBA,JOBNUM=0001139) 02:27:55
JCLS100011 JCLCM1801 10001139 JOBA Job ready for execution. 02:27:55
JCLS100011 Processed "c:\microfocus\es\mfbsi\MFBSI\jcltemp\iefbr14.jcl.000000000504@v1-aus-ctm-vy4nda"
JCLS100481 J0B0001139 JOBA JOB Started. ( Execution Region: MFWIN @VL-AUS-CTM-VY4M ) 02:27:56
JCLS100501 J0B0001139 JOBA JOB Concluded. ( RC=00000000 Reason=00000000 @2023080902275526 ) 02:27:56

.....
*.*.* Micro Focus ESJCL ASCII JES2 Version ED7_0_PU9D *.*.*
*.*.* Copyright (C) Micro Focus 1997-2020. All rights reserved. *.*.*
*.*.* Job: 0001139 Name: JOBA User: JESUSER Date: 08/09/23 Time: 02:27:55 *.*.*
*.*.* File: C:\MICROFOCUS\ES\MFBSI\MFBSI\JCL*000000000504@V1-AUS-CTM-VY4NDA *.*.*
*.*.* DSN: *.*.*
.....

1 //JOBA JOB
2 //STEP1 EXEC PGM=IEFBR14
3 //STEP2 EXEC PGM=IEFBR14
4 //STEP3 EXEC PGM=IEFBR14
5 //STEP4 EXEC PGM=IEFBR14
  
```

影像由 BMC Software, Inc. 提供 ©2022

最佳實務

在初始規劃和整合階段，我們建議使用下列最佳實務：

- 在整合之前，請徹底了解需要遷移或自動化的工作負載和程序。這有助於識別最關鍵的遷移任務，以及使用 Control-M 規劃排程和自動化。
- 將大型主機工作負載遷移至時 AWS，請從頭開始規劃使用 Control-M 進行自動化。考慮如何在雲端環境中排程、管理和監控任務和工作流程。
- 我們建議使用集中式連線設定檔，因為此方法可減少要管理的物件數並簡化 Control-M 代理程式的彈性部署。
- 如果可能，逐步執行大型主機遷移以降低複雜性和風險。透過進行增量遷移，遷移團隊可以更快地提供有關遷移進度的意見回饋。企業可以使用該意見回饋來優化內部程序，以加快遷移速度。
- 為了避免不必要的工作，請考慮在初始階段使用提供的作業類型和連線設定檔範本。

相關資源

參考

- [Micro Focus](#)
- [Control-M](#)
- [Control-M 試用](#)
- [Control-M Application Integrator](#)
- [Control-M 文件](#)
- [大型主機現代化 : DevOps on AWS with Micro Focus \(AWS 方案指引模式 \)](#)

Code

- [aws-mainframe-modernization-controlm-integration](#) GitHub 儲存庫

貢獻者

貢獻者

本文件的貢獻者包括：

- Sunil Bemarker , DevOps 資深合作夥伴解決方案架構師 AWS
- Joe Goldberg , BMC Software , Inc.
- Pablo Alonso Prieto , 資深大型主機架構師 , AWS
- Vaidy Sankaran , 資深現代化架構師 , AWS
- Vij Balakrishn , CloudOps 資深合作夥伴開發經理 , AWS

文件歷史紀錄

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

變更	描述	日期
已新增區段。	新的章節涵蓋 AWS Mainframe Modernization Amazon EC2 與 Control-M 整合 。	2024 年 2 月 19 日
初次出版	—	2022 年 11 月 16 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。