



使用 Control-M 工作流程协调器与大型机现代化集成 AWS

AWS 规范性指导



AWS 规范性指导: 使用 Control-M 工作流程协调器与大型机现代化集成 AWS

Table of Contents

简介	1
概述	1
目标成果	3
本指南中的术语	3
假设	4
架构	6
与 Control- AWS Mainframe Modernization M 的托管集成	8
部署 Control-M 资源	8
部署 AWS Mainframe Modernization Control-M 插件	8
在 Control-M 应用程序集成器中部署 AWS Mainframe Modernization 作业类型	9
创建连接配置文件	9
创建作业和计划	10
使用工作流程启动作业	11
按计划自动运行作业	15
基础作业根据事件运行	15
监控作业	15
Control-M 监控	15
在控制台上监控	16
AWS Mainframe Modernization 在 Amazon EC2 上与 Control-M 集成	18
部署资源	18
部署 Control-M 代理	18
部署微焦插件	8
部署作业类型	19
创建连接配置文件	19
创建作业和计划	21
使用工作流程启动作业	22
按计划自动运行作业	25
监控作业	25
Control-M 监控	26
最佳实践	27
资源	28
贡献者	29
文档历史记录	30
.....	xxxi

使用 Control-M 工作流程协调器与 AWS Mainframe Modernization

亚马逊 Web BMC Services and Software, Inc. ([贡献者](#))

2024 年 2 月 ([文档历史记录](#))

为了满足业务和数字化转型的需求，客户正在对大型机应用程序进行现代化改造，以提高敏捷性、降低成本并加快创新速度。在 re: Invent 2021 上，Amazon Web Services (AWS) 宣布了这项 [AWS Mainframe Modernization](#) 服务，旨在帮助客户实现大型机工作负载的现代化。AWS Mainframe Modernization 在上提供了云原生托管且高度可用的运行时环境。AWS

本指南提供了有关如何将 [BMCHelix Control-M](#) 工作流程编排与该 AWS Mainframe Modernization 服务集成的详细信息，以便在大型机现代化之旅中为批处理应用程序提供更多现代化选项。通过使用 Control-M 的功能，组织可以使用提供视图的单一工具合理化由现有工作负载和迁移的工作负载组成的环境。end-to-end

概述


Control-M 和 AWS Mainframe Modernization Replatform 与 Micro Focus 的集成具有以下好处：

- 有助于缩小人才缺口
- 通过敏捷 DevOps 方法支持快速创新
- 无需进行重大更改即可轻松访问应用程序和数据
- 优化运行或扩展应用程序的成本
- 最大限度地提高业务敏捷性并降低成本。

带有 Micro Focus 的 AWS Replatform 服务可帮助您将大型机应用程序现代化为 AWS 云原生托管运行时环境。它提供了工具和资源来帮助您规划和实施迁移与现代化。用户可以提交或取消批处理作业，查看批处理作业运行的详细信息。每次用户提交批处理作业时，带有 Micro Focus 的 AWS 的 Replatform 服务都会创建一个单独的批处理作业运行，可以对其进行监控。使用 AWS Mainframe Modernization 服务 Web 控制台，您可以按名称搜索批处理作业，为批处理作业提供作业控制语言 (JCL)、脚本文件和参数。

使用 Control-M，您可以定义、调度、管理和监控应用程序和数据 workflow。这增强了可见性和可靠性，从而有助于改善服务级别协议 (SLAs)。Control-M 集成、自动化和协调内部和内部的工作流程，AWS

Cloud 帮助您按时交付业务服务。通过一个统一的图形视图，用户可以使用丰富的插件库编排所有工作流，包括文件传输、应用程序、数据来源和基础设施。Control-M 在云中配置，使用了的临时功能。AWS Cloud由于他们在持续集成和持续交付 (CI/CD) 工具链中使用RESTAPIsJSON、和 Python jobs-as-code 的方法，因此可以对 Control-M 工作流程进行版本控制、测试和维护，以便开发人员和工程师可以协同工作。 DevOps

 Note

AWS Mainframe Modernization 还支持使用 AWS Blu Age 进行重构。有关更多信息，请参阅 [《AWS 规范性指导》](#) 指南

目标业务成果

此解决方案可帮助您实现以下目标。

商业利益

- 无缝迁移和工作流程编排 — Control-M 有助于将大型机工作负载平稳过渡到 AWS。它可以协调复杂的工作流程，并确保迁移期间和迁移后批处理的连续性。
- 混合运营管理 — Control-M 可在向混合运营过渡期间高效管理传统的大型机流程和基于云的新应用程序。AWS
- 降低成本和优化资源使用 — 使用 Control-M 迁移到 AWS 可通过优化云资源使用和扩展来降低运营成本。
- 增强敏捷性和创新性 — 此举 AWS 可帮助您的公司快速适应市场变化并使用云原生服务进行创新。
- 合规性、安全性和高效监控 — Control-M 有助于确保持续的合规性和安全性。Control-M 还在云环境中提供增强的实时监控和报告功能。
- 缩小人才缺口 — 此举旨在 AWS 应对与在传统平台上寻找和留住熟练的大型机专业人员相关的挑战。

技术优势

该解决方案可帮助您实现以下目标：

- 扩展您的 Control-M 环境以管理服务上的工作负载。AWS Mainframe Modernization
- 使用向导将 Control-M REST API 快速整合到流程中，从而支持快速创新。
- 在 Control-M 统一视图中编排、调度和监控所有应用程序工作负载作业。
- 提高组织的可扩展性和业务灵活性，以便您可以更有效地响应不断变化的市场条件和客户需求。
- 通过对大型机工作负载进行现代化改造并利用云原生服务的优势，优化运行或扩展应用程序的成本。

本指南中的术语

- 大型机应用程序是指一套相关的大型机程序和子程序，用于完成和促进一系列业务流程。大型机应用程序可以是批处理系统，也可以是联机事务处理 (OLTP) 系统。
- 批处理作业是指配置为无需用户交互即可运行的计划程序。在 AWS Replatform with Micro Focus 中，批处理作业 JCL 文件和批处理作业二进制文件都是在亚马逊简单存储服务 (Amazon S3) Service 存储桶中准备的，应用程序定义文件中提供了两者的位置。

- [AWS Mainframe Modernization](#) 是一项云原生服务，用于迁移、现代化、运行和操作大型机应用程序。
- [Amazon S3](#) 是可扩展的对象存储，具有高持久性、可用性和性能。
- [Amazon CloudWatch](#) 是一项监控和可观察性服务，专为 DevOps 工程师、开发人员、站点可靠性工程师 (SRE)、IT 经理和产品负责人打造。
- Control-M Web 是一款解决方案，用于在构建、测试、部署、调度和监控业务应用程序工作流的整个生命周期中管理企业工作负载，包括大型机批处理作业。一套图形和编程功能，让所有用户都能以最舒适的方式访问 Control-M 工具。
- Control-M 的应用程序集成器是一款基于 Web 的低代码设计器，用于创建任务类型，以便通过 Control-M 在应用程序或云服务之间实现集成。Application Integrator 作业类型构建后，它们的运行方式与所有其他 Control-M 作业完全相同，为目标集成公开了所有 Control-M 特性和功能。在本指南中，集成了带有 Micro Focus 的 AWS Replatform。
- 规划域是一套管理作业流创建和更新的功能。
- 监控域是一套管理作业流的功能。在这里，您可以查看所有正在运行的作业及其状态。如果发生错误，则会生成警报。您可以执行操作来查看作业状态、对延误或故障做出反应、分析问题并采取纠正措施。
- Control-M 代理安装在由 Control-M 服务器管理的不同计算机上。作业被分配一个代理名称，用来识别主机。代理使用 HTTPS 协议（端口 443）与服务器通信。

假设

本指南中的示例和图表反映了以下假设：

- 正在迁移的大型机应用程序可能运行单个程序或多个程序。为简单起见，本指南中的图表显示了每个应用程序的单个程序和多个子程序。
- 大型机应用程序将在带有 Micro Focus 的 AWS Replatform 托管运行时环境中迁移并运行，该环境包含多个已定义的批处理作业。在本试点中，按照[教程：Micro Focus 的托管运行时中的说明在 AWS Replatform with Micro Focus](#) 中设置 BankDemo 示例应用程序。
- 一整套 [Control-M 安装](#) 可用，包含所有 Control-M 组件，包括应用程序插件和附加组件。
- [Control-M 代理](#) 负责管理作业。为了分发工作负载，可以在多台计算机上安装代理。这可以提高性能和弹性。
- [Control-M Automation API](#) 通过 Restful Web Services (REST API) 公开 Control-M 功能。作业、连接配置文件、用户和角色以及站点标准之类的构件可以用 JSON 编写，并作为输入提供给各种服

务，或者由各种服务生成为输出。使用 curl 或类似工具或提供的 [ctm 命令行接口 \(CLI\)](#)，可以直接通过 HTTPS 请求访问服务。

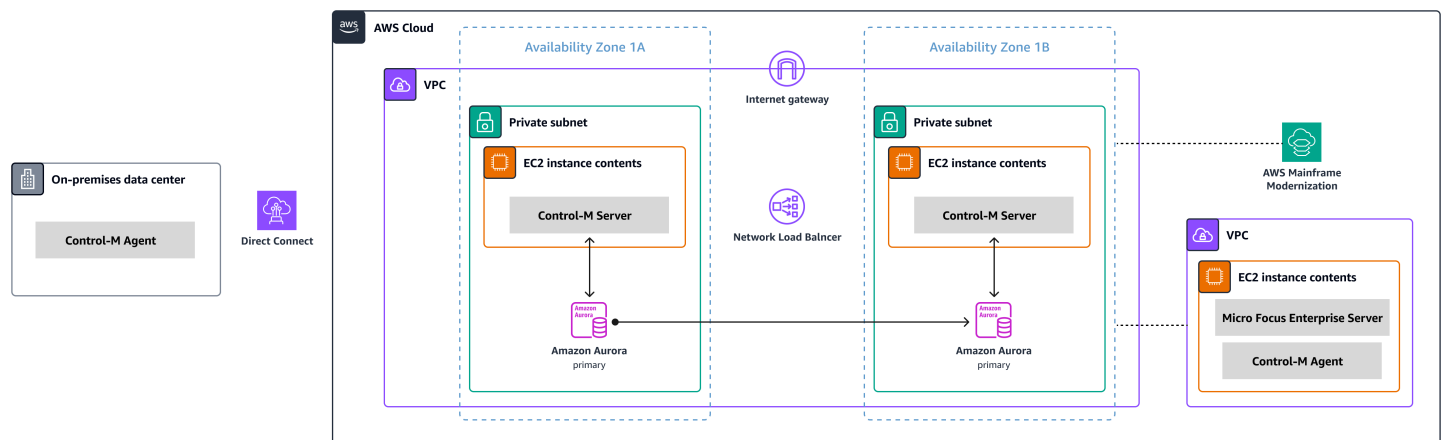
架构

AWS Mainframe Modernization 使用 Micro Focus 进行平台重置有两种模式可供选择：

- AWS 带有 Micro Focus 的 Replatform 是一种无服务器托管运行时环境，通过 Micro Focus 后端动态部署，完全由管理。AWS 使用 Micro Focus 进行平台重构提供了云原生 API 层，用于与 Micro Focus 进行交互。在这种托管方法中，只有 Micro Focus 可用于平台重组。该 UniKix 解决方案不可用。
- AWS 在亚马逊弹性计算云 (Amazon EC2) 上使用 Micro Focus 进行平台重建 Amazon EC2 以预装 Micro Focus 环境的亚马逊系统映像 (AMI) 的形式交付，该环境是在你选择的 EC2 实例类型上启动的。此自定义部署会直接公开原生 Micro Focus。

这两种模式都包括事务管理器、数据映射工具、屏幕和地图阅读器以及批处理作业运行环境。您可以使用任一模式在分布式服务器上运行大型机应用程序，只需对源代码进行最少的更改。

下图显示了在 Amazon EC2 实例上托管 Control-M 的工作流程集成。Amazon Aurora 数据库用于维护管理和运行批处理任务所需的数据。该架构采用多可用区（多可用区）部署，可实现高可用性。应用程序的批处理作业和数据是在带有 Micro Focus 的 AWS Replatform 运行时环境中编排的。该图显示了两种带有 Micro Focus 模式的重定向 AWS 平台：在 Amazon EC2 上完全托管和自定义。



该图显示了以下资源：

1. 在本地环境中，安装 Control-M 代理是为了控制仍在 IBM Z/OS 上运行的工作负载或其他工作负载。在 x86 上运行的工作负载通过 AWS Direct Connect 连接到 AWS 环境。
2. Control-M Server 以主动-被动模式安装在多可用区环境中的两个 EC2 实例上，以实现高可用性和灾难恢复。

3. Control-M 使用的 Amazon Aurora 后端数据库（在 EC2 实例上运行）与副本一起部署在辅助可用区，以实现高可用性和灾难恢复。
4. 单独的 VPC 包含一个 EC2 实例，该实例具有带有 Micro Focus 的 AWS Replatform，作为预安装的 Micro Focus 环境的 AMI 提供。此实例上安装了 Control-M 代理，用于与提供扩展作业管理功能的 Micro Focus 实用程序进行交互。

在迁移项目期间，您可能仍在大型机和分布式服务器上管理非 AWS 位置的工作负载。所示架构的目的不是规范性的，而是为了提供一个总体方向。我们建议在 Control-M 实现中构建包括灾难恢复选项在内的详细配置。

与 Control- AWS Mainframe Modernization M 的托管集成

本节介绍 Control-M 如何集成并支持在使用 Micro Focus 运行时引擎部署的托管 AWS Mainframe Modernization 环境中运行的批处理作业。如果您要在 Amazon EC2 上实现带有 Micro Focus 环境的自定义 AWS Replatform，请参阅“[AWS Mainframe Modernization 在 Amazon EC2 上与 Control-M 集成](#)”部分。

本节假设以下先决条件：

- 活跃 AWS 账户的。
- 大型机应用程序将在带有 Micro Focus 的 AWS Replatform 托管运行时环境中迁移并运行，该环境包含多个已定义的批处理作业。
- 在本试点中，BankDemo 示例应用程序是在中设置的 AWS Mainframe Modernization。有关设置说明，请参阅[教程：Micro Focus 的托管运行时](#)。

以下主题介绍了 Control-M Scheduler 与不同类型集成工作流程的 AWS Mainframe Modernization 环境之间的集成所需的 step-by-step 设置：

- [部署 Control-M 资源](#)
- [为 AWS Mainframe Modernization 创建 Control-M 连接配置文件](#)
- [在 Control-M 规划中创建作业和计划](#)
- [监控作业](#)

部署 Control-M 资源

将 AWS 大型机现代化与 Control-M 集成的前两个步骤是部署插件和任务类型。

部署 AWS Mainframe Modernization Control-M 插件

插件为 Control-M 编排的应用程序和服务提供集成和支持。对于托管 AWS Mainframe Modernization 服务，请部署 AWS Mainframe Modernization 插件。

插件部署是一种很少发生的活动。如果是首次安装插件，请按照 [Control-M](#) 文档中的步骤进行操作。如果您有要使用的现有插件，请跳过此步骤并[创建连接配置文件](#)。

在 Control-M 应用程序集成器中部署 AWS Mainframe Modernization 作业类型

作业类型部署通常是一次性活动。如果您已有要使用的现有作业类型，请跳过此步骤，并继续[创建连接配置文件](#)的下一步。

git 存储库中提供了示例作业类型 [AIJOB.ctmai](#)。要部署作业类型，必须使用[应用程序集成器](#)执行以下步骤：

1. 克隆 [aws-mainframe-modernization-controlm-in](#) tegrati AIJOB.ctmai on GitHub 存储库并将文件下载到应用程序集成商可以访问的文件系统位置。
2. 登录 Application Integrator。
3. 在主页选项卡上，选择从文件导入作业类型，然后选择 AIJOB.ctmai 的位置。
4. 如果要对提供的示例进行任何修改，请先熟悉 Application Integrator。
5. 按照 [Control-M 文档](#)中的说明部署作业类型。

为创建 Control-M 连接配置文件 AWS Mainframe Modernization

连接配置文件定义了应用程序特定实例的连接属性和安全凭证。每个连接配置文件可以被多个作业引用。您可以为应用程序和凭证的每种独特组合创建单独的配置文件。

[要定义连接配置文件](#)，您可以使用 [Control-M Web 界面配置域中提供的图形用户界面 \(GUI\)](#)，也可以使用 [JSON](#)。有关 AWS Mainframe Modernization 插件连接配置文件的信息，请参阅 [Control-M 文档](#)。

以下代码是使用 JSON 的示例：

```
{
  "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
  }
}
```

创建与示例类似的 JSON 文件，然后使用 [Control-M Automation API 部署](#) 服务进行部署。例如，如果 JSON 代码保存到名为的文件中 `cp-MANAGED-M2-REPLATFORM.json`，则部署此连接配置文件的 `ctm` CLI 语法如下：

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

来自 Control-M Automation API 的响应将类似于以下内容：

```
[
  {
    "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 规划中创建作业和计划

现在，您已经部署了任务类型和 AWS 连接的连接配置文件，就可以开始创建和运行任务了。

每个 AWS Mainframe Modernization 服务作业都由一组属性组成，分为四个部分。每个部分可以有許多属性。以下列表显示了一些比较常用的属性。

- 常规：
 - 作业名称
 - 作业所属的应用程序和子应用程序
 - 要提交的 JCL
 - 文档链接
- 调度：

- 该作业符合运行资格的月份和天数
- 日历，例如企业会计期、节假日或其他无法通过算法定义的特殊日期
- 时间窗口
- 循环行为
- 先决条件：
 - 上游依赖项（通常是作业，必须成功完成此作业后才能运行）
 - 可能需要的资源
 - 可能需要的用户操作
- Control-M 在作业完成后执行的操作：
 - 确定作业成功还是失败（通常基于作业的完成代码，但您可以覆盖该设置以使用输出文本或检查特定状态）
 - 失败或成功通知，如电子邮件
 - 下游依赖项的发布状态

与连接配置文件一样，可以在 [GUI](#) 或 [JSON](#) 中创建和部署作业，并使用 Control-M Automation API 进行部署。

以下部分介绍了一些常见的工作流场景：

- [根据先前作业的状态代码启动作业](#)
- [按计划频率自动运行作业](#)
- [基础作业根据事件运行](#)

根据先前作业的状态启动作业

创建名为工作流的作业流。工作流中的作业与成功完成先前作业的依赖项相互关联。

使用 Control-M Web GUI

要从 Control-M 用户界面启动任务，请执行以下操作：

1. 在“规划”域中，添加一个新的工作空间。这将打开一个包含空文件夹对象的画布。
2. 选择 AWS Mainframe Modernization 作业类型（如果您使用的是提供的作业类型模板，则该模板称为 [M2JOB](#)），然后将其拖到该文件夹中。

- 任务类型的颜色变为绿色后，将其丢掉。右侧的窗格包含“常规”、“计划”、“先决条件”和“操作”部分。要创建作业，请参阅 [Control-M](#) 文档中的标准说明。
- 接下来，您需要使用 Job Name 值来配置作业类型。您可以在 AWS Mainframe Modernization 控制台的“应用程序定义”屏幕上或通过运行 [ListBatchJobDefinitions API](#) 来找到 Job Name 值。在本试点中，对多个作业重复步骤 2-4，使用您选择的任何名称命名每个作业。示例名称有 CBANK、CURRENCY、INVFUNDS BROKERAGE、RISKMGMT、和 BANK-SERVICE-Managed-M2。
- 要将这些作业连接到所需的流程中，请选择作业对象下方的条件三角形，然后将其拖动到下一个作业上。例如，选择下面的条件三角形 CBANK，然后将其拖动到上 CURRENCY。此操作使 CBANK 成为 CURRENCY 的前身。默认情况下，现在 CBANK 必须成功完成，CURRENCY 才有资格运行。

以下屏幕截图显示了基本任务流程的 Control-M 计划视图。

The screenshot displays the Control-M interface. On the left, a job plan titled 'jog-managed-m2' is shown. It features a flowchart where 'CBANK' is connected to 'CURRENCY', which then branches into 'BROKERAGE', 'INVFUNDS', and 'RISKMGMT', all of which lead to '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), Server (psctm), Host/Host group (ctm-worker), and various scheduling and execution parameters.

图片由 BMC Software, Inc.©2022 提供

使用 JSON

同样的流程可以用 JSON 编码：

```
{
  "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:SLAManagement",
  "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"
  ]
}
}
```

```
}
```

要部署此流程，请使用部署服务：

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

按计划频率自动运行作业

使用您在前面步骤中创建的流，可以添加基本调度和运行时调度。

- 基本调度定义了作业可以运行的天数（例如，每个工作日、仅工作日、月末或季度末）。
- 运行时间调度决定了作业在符合运行条件的当天何时运行（例如，每小时、在指定的资源可用之后，或者仅在手动确认之后）。

您可以在调度选项卡上设置基本调度和运行时调度。

基础作业根据事件运行

Control-M 托管文件传输（MFT）是一个 FTP/SFTP 客户端和服务端，可用于在本地主机和远程主机之间监视和传输文件。有关定义文件传输作业的更多信息，请参阅 [Control-M documentation](#)。

本次试验使用文件传输作业来监视 S3 存储桶（名为 bmc-poc-bucket）中 /bmcfile 文件夹中文件（扩展名为 .poc）的文件创建事件。当该事件发生时，将启动 Control-M 作业以运行下一个作业。您可以选择传递完整路径，包括存储桶名称。

监控作业

您可以在 Control-M Monitoring 域内以及通过监控和验证作业的处理情况 AWS Management Console，从而确保在这两个平台上进行全面的观察和验证。

Control-M 监控

可以在 Control-M 监控域中监控作业提交和运行。默认情况下，AWS Mainframe Modernization 服务作业将与所有其他 Control-M 工作一起显示。如果您只想查看没有任何其他工作负载（或任何其他筛选要求）的 AWS Mainframe Modernization 服务作业，则可以创建 Viewpoint。

视点不仅显示作业信息，还显示与上游和下游依赖项的关系。此外，如果您的工作流程包括 AWS Mainframe Modernization 和其他 Control-M 作业类型，则可以在监控域中查看和管理整个流程。

要执行详细步骤，请参阅 Control-M 文档中[监控的“视点”部分](#)。

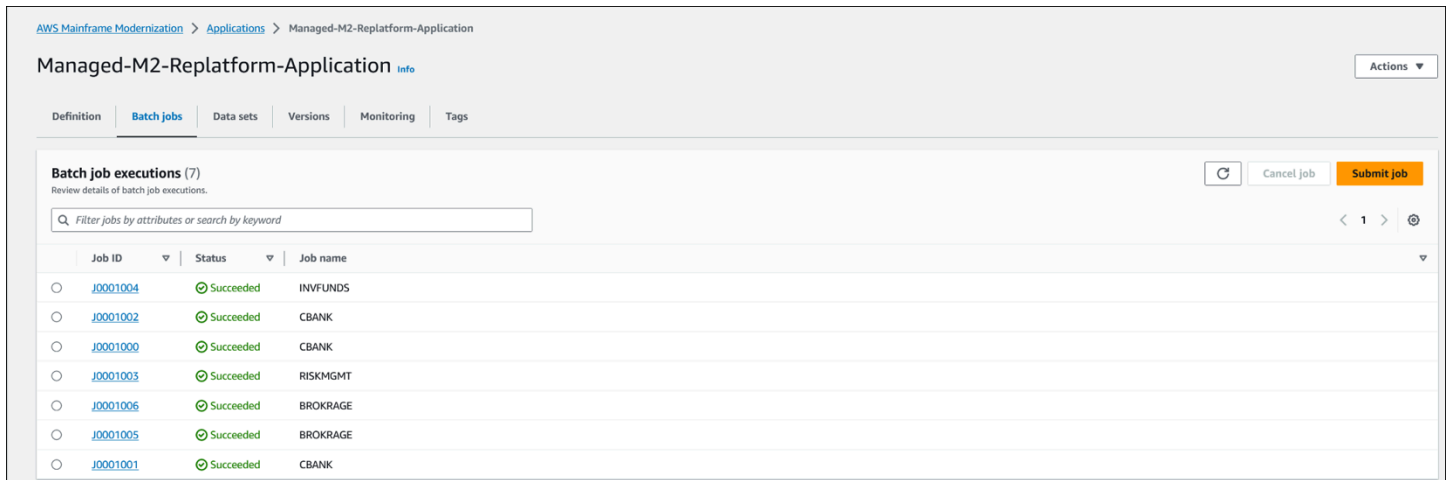
以下屏幕截图显示了两个工作流的输出。在左侧，工作流程已成功完成，所有作业均显示为绿色。在右侧，工作流程仅部分成功，因为作业CURRENCY返回了“失败”状态，该状态由红色表示。工作流程就此停止，剩余的作业仍处于“等待计划”状态。

The screenshot displays two workflow instances for 'jog-managed-m2'. The left instance shows a successful execution where all jobs (CBANK, CURRENCY, BROKERAGE, INVFUNDS, RISKMGMT, and BANK-SERVICE-Managed-M2) are in a green state. The right instance shows a failure in the CURRENCY job, which is highlighted in red, leading to the remaining jobs (BROKERAGE, INVFUNDS, RISKMGMT, and BANK-SERVICE-Managed-M2) being in a grey 'Waiting for plan' state. On the right side of the screenshot, a configuration panel for the 'CBANK' job is visible, showing details such as Job type (AWS Mainframe Modernization), Job name (CBANK), and various settings like Where (Server: psctm, Host/Host group: ctm-worker) and What (Connection profile: MANAGED-M2-REPLATFORM, Application Name: M2-Managed-via-CF-4fa2c710, Action: Start Batch Job, JCL Name: CBANKjcl).

图片由 BMC Software, Inc.©2022 提供

在控制台上监控

要在此查看作业和日志信息 AWS，请登录 AWS Management Console，然后导航到[AWS Mainframe Modernization 控制台](#)。



The screenshot displays the AWS Managed-M2-Replatform-Application console. The breadcrumb navigation shows the path: AWS Mainframe Modernization > Applications > Managed-M2-Replatform-Application. The main heading is "Managed-M2-Replatform-Application" with an "Info" link and an "Actions" dropdown menu. Below the heading 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". To the right of the search bar are buttons for "Cancel job" and "Submit job". Below the search bar is a table with the following data:

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

此视图不包括依赖关系，也不包括任何不由 AWS Mainframe Modernization 服务管理的工作负载。

AWS Mainframe Modernization 在 Amazon EC2 上与 Control-M 集成

本节介绍 Control-M 如何集成并支持在 EC2 实例上部署的自定义 AWS 大型机现代化运行时环境中运行的批处理作业。如果您正在实施带有 Micro Focus 的完全托管的 AWS Replatform 运行时环境，请参阅[与 Control-M 的托管 AWS Mainframe Modernization 集成](#)部分。

本节假设以下先决条件：

- 一个有效的 Amazon Web Services account。
- 将在其中创建 EC2 实例的虚拟私有云 (VPC)。
- 大型机应用程序在 EC2 实例上的 AWS Replatform 和 Micro Focus 环境中迁移和运行，并且支持具有多个已定义批处理任务的 Micro Focus 运行时引擎。在本试点中，请按照使用[Micro Focus 重塑平台应用程序中的说明](#)进行操作。该文档包括有关在 Amazon EC2 上配置和操作带有 Micro Focus 的 AWS Replatform 运行时环境的所有任务和其他信息。

以下主题涵盖了 Control-M 和带有 Micro Focus 的 AWS Replatform 环境之间集成所需的设置：

- [部署 Control-M和Micro Focus资源](#)
- [创建 Control-M 连接配置文件](#)
- [在 Control-M 规划中创建作业和计划](#)
- [使用“监控”在 Control-M 中管理作业运行](#)

部署 Control-M和Micro Focus资源

将 AWS 大型机现代化与 Control-M 集成的前两个步骤是部署 Control-M 代理，然后在代理上部署插件。第三步是在 Control-M 应用程序集成器中部署作业类型。

在 EC2 实例上部署 Control-M 代理

在 Amazon EC2 运行时环境中使用带有 Micro Focus 的自定义 AWS Replatform 时，将调MFBSIFJCL用该实用程序。该实用程序在使用的 Micro Focus Enterprise Server 产品启动的主机上运行 AWS Marketplace。要运行该MFBSIFJCL实用程序，还必须在该主机上部署 Control-M 代理。有关说明，请参阅[Control-M 文档](#)。

Note

您可以从 [BMC 电子产品下载网站下载](#) 所需的安装媒体。

在 Control-M 代理上部署 Micro Focus 插件

插件为 Control-M 编排的应用程序和服务提供集成和支持。

插件部署是一种很少发生的活动。如果您已经有要使用的现有插件，请跳过此步骤并创建连接配置文件。

在 Amazon EC2 上，带有 Micro Focus 的 AWS Mainframe Modernization Replatform 服务公开了 Micro Focus 引擎。为了进行集成 AWS Mainframe Modernization，Control-M 使用 Micro Focus 插件。有关更多信息，请参阅 [Control-M 文档](#)。

必须将此插件部署到安装在运行 Micro Focus Enterprise Server 的主机上的代理上。

在 Control-M 应用程序集成器中部署 AWS Mainframe Modernization 作业类型

作业类型部署通常是一次性活动。如果您已有要使用的现有作业类型，请跳过此步骤，并继续[创建连接配置文件](#)的下一步。

git 存储库中提供了示例作业类型 [AIJOB.ctmai](#)。要部署作业类型，必须使用[应用程序集成器](#)执行以下步骤：

- 克隆 [aws-mainframe-modernization-controlm-in](#) tegrati AIJOB.ctmai on GitHub 存储库并将文件下载到应用程序集成商可以访问的文件系统位置。
- 登录 Application Integrator。
- 在主页选项卡上，选择从文件导入作业类型，然后选择 AIM2JOB.ctmai 的位置。
- 如果要对提供的示例进行任何修改，请先熟悉 Application Integrator。
- 按照 [Control-M 文档](#) 中的说明部署作业类型。

创建 Control-M 连接配置文件

连接配置文件定义了应用程序特定实例的连接属性和安全凭证。每个连接配置文件可以被多个作业引用。您可以为应用程序和凭证的每种独特组合创建单独的配置文件。

要定义连接配置文件，您可以使用 Control-M Web 界面的[配置域](#)，也可以使用 JSON。以下代码是使用 JSON 的示例：

```
{
  "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": "*****"
  }
}
```

示例代码位于 GitHub repo 的文件 `ConnectionProfile-Custom-M2-Replatform.json` 中。要部署代码，请使用 Control-M Automation API 部署服务：

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

来自 Control-M Automation API 的响应将类似于以下内容：

```
[
  {
    "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 规划中创建作业和计划

现在，您已经部署了插件和 Micro Focus 连接的连接配置文件，就可以开始创建和运行作业了。

每个 Control-M for Micro Focus 作业都由一组属性组成，分为四个部分。每个部分可以有許多属性。以下列表显示了一些比较常用的属性。

- 常规：
 - 作业名称
 - 作业所属的应用程序和子应用程序
 - 要提交的 JCL
 - 指向现场提供的操作文档的链接
- 调度：
 - 该作业符合运行资格的月份和天数
 - 日历，例如企业会计期、节假日或其他无法通过算法定义的特殊日期
 - 时间窗口
 - 循环行为，例如每小时跑一次
- 先决条件
 - 上游依赖项（通常是作业，必须成功完成此作业后才能运行）
 - 可能需要的资源
 - 可能需要的用户操作
- Control-M 在作业完成后执行的操作：
 - 确定作业成功还是失败（通常基于作业的完成代码，但您可以覆盖该设置以使用输出文本或检查特定状态）
 - 失败或成功通知，如电子邮件
 - 下游依赖项的发布状态

与连接配置文件一样，可以[使用 Control-M Web 创建和部署作业](#)，也可以使用 JSON 编写并使用 [Control-M Automation API](#) 进行部署。

以下部分介绍了一些常见的工作流场景：

- [使用工作流程启动作业](#)
- [按计划频率自动运行作业](#)

使用工作流程启动作业

创建名为工作流的作业流。工作流中的作业与成功完成先前作业的依赖项相互关联。

使用 Control-M Web

要从 Control-M 用户界面启动作业，请执行以下操作：

1. 在“规划域”中，添加一个新的工作区。这将打开一个包含空文件夹对象的画布。
2. 在菜单栏上，选择“添加作业”，选择 Micro Focus Windows 或 Micro Focus Linux 作业，然后将其拖到文件夹的正面。
3. 作业颜色变为绿色后，将其丢掉。右侧的窗格包含“常规”、“计划”、“先决条件”和“操作”部分。要创建作业，请参阅 [Control-M](#) 文档中的标准说明。
4. 有关构建 Control-M for Micro Focus 作业的说明，请参阅 [Control-M](#) 文档。在本试点中，对五个作业重复步骤 2—4，将每个作业命名为“”。aws-mf-job1 aws-mf-job5
5. 要将这些作业连接到所需的流程中，请选择作业对象下方的条件三角形，然后将其拖动到下一个作业上。例如，选择下面的条件三角形aws-mf-job1，然后将其拖动到上aws-mf-job2。此操作使aws-mf-job1之成为的前身aws-mf-job2。默认情况下，现在aws-mf-job1必须成功完成aws-mf-job2才有资格运行。

下图显示了基本作业流的 Control-M 规划视图。



图片由 BMC Software, Inc.©2022 提供

使用 JSON

同样的流程可以用 JSON 编码。在 GitHub repo 中，示例代码位于文件 `Folder-Custom-M2-Replatform.json` 中：

```
{
  "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"
  ]
}
}
```

要部署此流程，请使用 Control-M 自动化 API：

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

按计划频率自动运行作业

使用您在前面步骤中创建的流，可以添加基本调度和运行时调度。

- 基本计划定义了作业有资格运行的日期（例如，每个工作日，仅在工作日、月末或季度末）。您可以在“计划”选项卡上设置基本日[程安排](#)。
- 运行时调度决定了作业在符合运行条件的当天何时运行（例如，在前置任务完成之后、每小时、在指定的资源可用之后，或者仅在手动确认之后）。

您可以在“调度”部分定义一些运行时调度，在“先决条件”部分定义其他运行时计划。

监控作业

您可以在 Control-M Monitoring 域和 [Micro Focus Enterprise Server Common Web 管理用户界面](#) 中监控和验证作业。

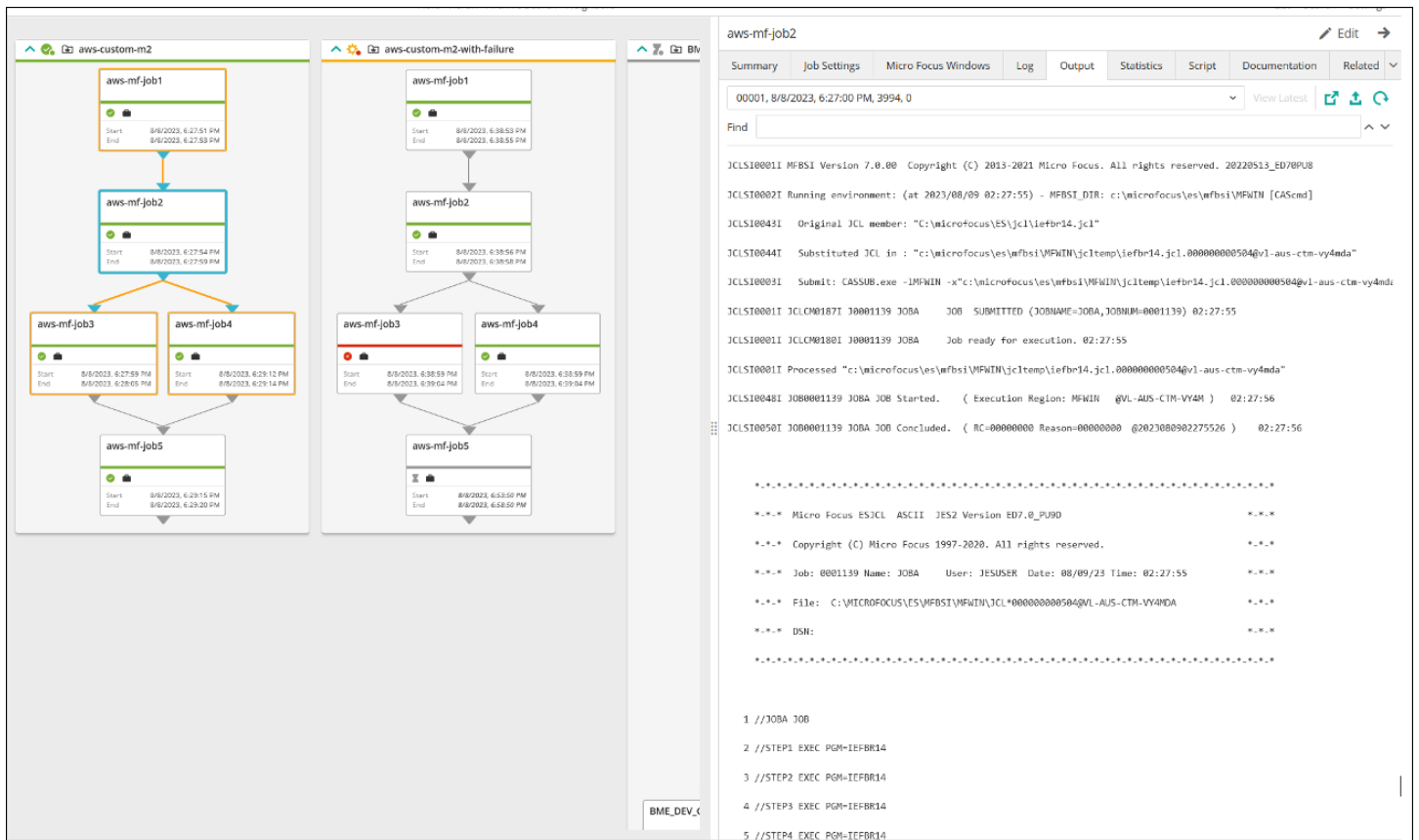
Control-M 监控

可以在 Control-M 监控域中监控作业提交和运行。默认情况下，AWS Mainframe Modernization 服务作业将与所有其他 Control-M 工作一起显示。如果您只想查看没有任何其他工作负载 (或任何其他筛选要求) 的 AWS Mainframe Modernization 服务作业，则可以创建 Viewpoint。

视点不仅显示作业信息，还显示与上游和下游依赖项的关系。此外，如果您的工作流程包括 AWS Mainframe Modernization 和其他类型的 Control-M 作业，则可以在监控域中查看和管理整个流程。

您可以通过访问 Control-M 文档中[监控的“视点”部分](#)来执行详细步骤。

以下屏幕截图显示了两个工作流的输出。在左侧，工作流已成功完成，五个作业均为绿色。在右侧，工作流仅部分成功，因为aws-mf-job3返回了“失败”状态，而工作流则停aws-mf-job5在那里，处于“等待计划”状态。



图片由 BMC Software, Inc.©2022 提供

最佳实践

在初始规划和整合阶段，我们建议采用以下最佳实践：

- 在集成之前，请彻底了解需要迁移或自动化的工作负载和流程。这有助于确定最关键的迁移任务，并使用 Control-M 规划其日程安排和自动化。
- 将大型机工作负载迁移到时 AWS，请从一开始就使用 Control-M 规划其自动化。考虑如何在云环境中安排、管理和监控作业和工作流程。
- 我们建议使用集中式连接配置文件，因为这种方法可以减少要管理的对象数量，并简化 Control-M 代理的弹性部署。
- 在可能的情况下，以增量方式执行大型机迁移，以降低复杂性和风险。通过增量迁移，迁移团队可以更快地提供有关迁移进度的反馈。企业可以利用这些反馈来优化内部流程，加快迁移速度。
- 为避免不必要的工作，考虑在初始阶段使用提供的作业类型和连接配置文件模板。

相关资源

参考

- [Micro Focus](#)
- [Control-M](#)
- [Control-M 试用](#)
- [Control-M Application Integrator](#)
- [Control-M 文档](#)
- [大型机现代化：DevOps 在 AWS 上使用 Micro Focus \(AWS 规范性指导模式 \)](#)

代码

- [aws-mainframe-modernization-controlm-集成存储库](#) GitHub

贡献者

贡献者

本文档的贡献者包括：

- Sunil Bemarker，高级合作伙伴解决方案架构师 —， DevOps AWS
- 乔·戈德堡，BMC软件公司
- 巴勃罗·阿隆索·普列托，高级主机架构师， AWS
- Vaidy Sankaran，高级现代化建筑师， AWS
- Vij Balakrishn，高级合伙人开发经理 —， CloudOps AWS

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 [RSS 源](#)。

变更	说明	日期
已添加章节。	一个新章节介绍了 Amaz AWS Mainframe Modernization on EC2 与 Control-M 的集成 。	2024年2月19日
初次发布	—	2022 年 11 月 16 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。