aws

Using Control-M workflow orchestrator integration with AWS Mainframe Modernization

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Using Control-M workflow orchestrator integration with AWS Mainframe Modernization

Table of Contents

Using Control-M workflow orchestrator integration with AWS Mainframe Modernization

Amazon Web Services and BMC Software, Inc. ([contributors](#))

February 2024 ([document history](#))

To meet the needs of a business and digital transformation, customers are modernizing mainframe applications to achieve increased agility, reduce costs, and innovate faster. At re:Invent 2021, Amazon Web Services (AWS) announced the [AWS Mainframe Modernization](#) service to help customers modernize their mainframe workloads. AWS Mainframe Modernization provides a cloud-native managed and highly available runtime environment on AWS.

This guide provides detailed information on how [BMC Helix Control-M](#) workflow orchestration can be integrated with the AWS Mainframe Modernization service to give you an additional option for modernizing your batch applications during your mainframe modernization journey. By using the capabilities of Control-M, organizations can rationalize an environment that consists of existing and migrated workloads using a single tool that delivers an end-to-end view.

Overview

The integration of Control-M and AWS Mainframe Modernization Replatform with Micro Focus offers the following benefits:

- Helps with reducing talent gaps

- Supports rapid innovation with an agile DevOps approach

- Provides easier access to applications and data without significant changes

- Optimizes the costs of running or extending applications

- Maximizes business agility and reduces costs.

The AWS Replatform with Micro Focus service helps you modernize your mainframe applications to AWS cloud-native managed runtime environments. It provides tools and resources to help you plan and implement migration and modernization. Users can submit or cancel batch jobs and review the details of batch job runs. Each time a user submits a batch job, the AWS Replatform with Micro Focus service creates a separate batch job run, which can be monitored. Using the AWS Mainframe

Modernization service web console, you can search for batch jobs by name, provide job control language (JCL), script files, and parameters to batch jobs.

Using Control-M, you can define, schedule, manage, and monitor application and data workflows. This enhances visibility and reliability, which helps with improving service-level agreements (SLAs). Control-M integrates, automates, and orchestrates workflows on premises and in AWS Cloud to help you deliver business services on time. With a single unified graphical view, users can orchestrate all your workflows, including file transfers, applications, data sources and infrastructure, with a rich library of plug-ins. Provisioned in a cloud, Control-M uses the ephemeral capabilities of the AWS Cloud. Because they use a jobs-as-code approach, with REST APIs, JSON, and Python within the continuous integration and continuous delivery (CI/CD) toolchain, Control-M workflows can be versioned, tested, and maintained, so that developers and DevOps engineers can work collaboratively.

> **ⓘ Note**
>
> AWS Mainframe Modernization Refactor with AWS Blu Age is also supported. For more information, see the AWS Prescriptive Guidance guide

Targeted business outcomes

This solution helps you achieve the following objectives.

Business benefits

- **Seamless migration and workflow orchestration** – Control-M facilitates the smooth transition of mainframe workloads to AWS. It orchestrates complex workflows and ensuring continuity in batch processing during and after migration.

- **Hybrid operations management** – Control-M enables efficient management of both legacy mainframe processes and new cloud-based applications during the transition to AWS.

- **Cost reduction and optimized resource use** – Migrating to AWS with Control-M reduces operational costs through optimized cloud-resource use and scaling.

- **Enhanced agility and innovation** – The move to AWS helps your company to adapt rapidly to market changes and use cloud-native services for innovation.

- **Compliance, security, and efficient monitoring** – Control-M helps ensure ongoing compliance and security. Control-M also provides enhanced real-time monitoring and reporting capabilities in the cloud environment.

- **Reducing talent gaps** – The move to AWS addresses the challenges associated with finding and retaining skilled mainframe professionals on a legacy platform.

Technical benefits

This solution helps you achieve the following objectives:

- Extend your Control-M environment to manage workloads on the AWS Mainframe Modernization service.

- Support rapid innovations by using wizards to incorporate Control-M REST APIs into processes quickly.

- Orchestrate, schedule and monitor all your application workload jobs in the Control-M unified view.

- Increase your organization's scalability and business agility, so that you can respond more effectively to changing market conditions and customer demands.

- Optimize the cost of running or extending applications by modernizing their mainframe workloads and using the benefits of cloud-native services.

Terms in this guide

- A *mainframe application* refers to a set of related mainframe programs and subprograms that accomplish and facilitate a set of business processes. Mainframe applications can be batch processing systems or online transaction processing (OLTP) systems.

- A *batch job* refers to a scheduled program that is configured to run without requiring user interaction. In AWS Replatform with Micro Focus, both batch job JCL files and batch job binaries are prepared in an Amazon Simple Storage Service (Amazon S3) bucket, and the location of both is provided in the application definition file.

- AWS Mainframe Modernization is a cloud-native service for migrating, modernizing, running, and operating mainframe applications.

- Amazon S3 is scalable object storage with high durability, availability, and performance.

- Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners.

- *Control-M Web* is a solution for managing enterprise workloads, including mainframe batch jobs, through the entire lifecycle of building, testing, deploying, scheduling, and monitoring business application workflows. An extensive set of graphical and programmatic facilities provide all users with access to Control-M facilities in the way they feel most comfortable.

- Control-M's *Application Integrator* is a web-based, low-code designer used to create job types that enable integration between applications or cloud services with Control-M. After Application Integrator job types are built, they operate exactly like all other Control-M jobs, exposing all Control-M features and functions for the target integration. In this guide, the integration is with AWS Replatform with Micro Focus.

- The *Planning domain* is a set of capabilities for managing the creation and updating of job flows.

- The *Monitoring domain* is a set of capabilities for managing the job flows. This is where you view all running jobs and their status. If errors occur, alerts are generated. You can perform operational actions to view job status, react to delays or failures, analyze problems, and take corrective actions.

- *Control-M Agents* are installed on different computers that are managed by Control-M Server. Jobs are assigned to an agent name that identifies a host. The Agent communicates with the server using HTTPS protocol (port 443).

Assumptions

The examples and diagrams in this guide reflect the following assumptions:

- The mainframe application that is being migrated might run a single program or multiple programs. For simplicity, the diagrams in this guide display a single program and multiple subprograms for each application.

- The mainframe application is migrated and running in an AWS Replatform with Micro Focus managed runtime environment with multiple defined batch jobs. For this pilot, set up the BankDemo example application in AWS Replatform with Micro Focus by following the instructions in Tutorial: Managed Runtime for Micro Focus.

- A full Control-M installation is available, with all Control-M components, including Application plug-ins and add-ons.

- The Control-M Agent is responsible for managing jobs. To distribute the workload, Agents can be installed on multiple computers. This can enhance performance and resiliency.

- The Control-M Automation API exposes Control-M functionality through Restful Web Services (REST API). Artifacts such as jobs, connection profiles, users and roles, and site standards can be written in JSON and provided as input to or are generated as output by various services. Services can be accessed directly through HTTPS requests by using curl or similar facilities or the provided ctm command line interface (CLI).
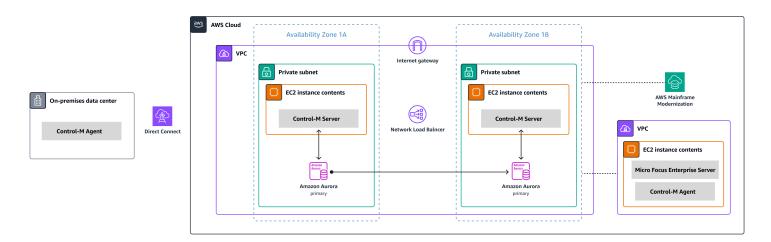
Architecture

AWS Mainframe Modernization Replatform with Micro Focus is available in two modes:

- AWS Replatform with Micro Focus is a serverless managed runtime environment that is dynamically deployed with a Micro Focus backend and is fully managed by AWS. AWS Replatform with Micro Focus provides a cloud-native API layer for interacting with Micro Focus. In this managed approach, only Micro Focus is available for replatforming. The UniKix solution isn't available.

- AWS Replatform with Micro Focus on Amazon Elastic Compute Cloud (Amazon EC2) is delivered as an Amazon Machine Image (AMI) of a preinstalled Micro Focus environment that you launch on the EC2 instance type that you choose. This custom deployment exposes native Micro Focus directly.

Both modes include transaction managers, data mapping tools, screen and maps readers, and batch job run environments. You can use either mode to run mainframe applications on distributed servers with minimal changes to the source code.

The following diagram shows workflow integration where Control-M is hosted on an Amazon EC2 instance. An Amazon Aurora database is used for maintaining the data required to manage and run batch jobs. The architecture is a Multi-Availability Zone (Multi-AZ) deployment for high availability. Applications' batch jobs and data are orchestrated in the AWS Replatform with Micro Focus runtime environment. The diagram shows both AWS Replatform with Micro Focus modes: fully managed and custom on Amazon EC2.

The diagram shows the following resources:

1. In the on-premises environment, the Control-M Agent is installed to control workloads still running on IBM Z/OS or other workload. The workloads that are running on x86 connect to the AWS environment through AWS Direct Connect.

2. Control-M Server is installed on a pair of EC2 instances in an active-passive mode in a Multi-AZ environment for high availability and disaster recovery.

3. The Amazon Aurora backend database used by Control-M (running on an EC2 instance) is deployed with a replica in the secondary Availability Zone for high availability and disaster recovery.

4. A separate VPC contains an EC2 instance that has AWS Replatform with Micro Focus delivered as an AMI of a preinstalled Micro Focus environment. Control-M Agent is installed on this instance to interact with Micro Focus utilities that provide extended job management capabilities.

During the migration project, you might still be managing workloads in non AWS locations on both mainframe and distributed servers. The architecture shown isn't intended to be prescriptive but to provide a general direction. We recommend that a detailed configuration, including disaster recovery options, is constructed as part of the Control-M implementation.

Managed AWS Mainframe Modernization integration with Control-M

This section describes how Control-M integrates with and supports batch jobs that run in a managed AWS Mainframe Modernization environment deployed with a Micro Focus runtime engine. If you are implementing a custom AWS Replatform with Micro Focus environment on Amazon EC2, see the AWS Mainframe Modernization on Amazon EC2 integration with Control-M section.

This section assumes the following prerequisites:

- An active AWS account.

- The mainframe application is migrated and running in an AWS Replatform with Micro Focus managed runtime environment with multiple defined batch jobs.

- For this pilot, the BankDemo example application is set up in AWS Mainframe Modernization. For setup instructions, see Tutorial: Managed Runtime for Micro Focus.

The following topics describe step-by-step setup required for integration between Control-M Scheduler and the AWS Mainframe Modernization environment for different types of integration workflows:

- Deploy Control-M resources

- Create a Control-M connection profile for AWS Mainframe Modernization

- Create jobs and schedules in Control-M Planning

- Monitor jobs

Deploy Control-M resources

The first two steps in integrating AWS Mainframe Modernization with Control-M are to deploy the plug-in and the job type.

Deploy the AWS Mainframe Modernization Control-M plug-in

Plug-ins provide integration and support for the applications and services that Control-M orchestrates. For the managed AWS Mainframe Modernization service, deploy the AWS Mainframe Modernization plug-in.

The plug-in deployment is an infrequent activity. If installing the plug-in for the first time, follow the steps in the Control-M documentation. If you have an existing plug-in that you want to use, skip this step, and create a connection profile.

Deploy an AWS Mainframe Modernization job type in Control-M Application Integrator

The job type deployment is usually a one-time activity. If you already have an existing job type that you want to use, skip this step and move on to the next step of creating a connection profile.

The example job type AIJOB.ctmai is supplied in the git repository. To deploy a job type, you must perform the following steps using Application Integrator:

1. Clone the aws-mainframe-modernization-controlm-integration GitHub repository and download the `AIJOB.ctmai` file to a file system location that can be accessed by Application Integrator.
2. Log in to Application Integrator.
3. On the **Home** tab, select **Import job type from file**, and select the location of `AIJOB.ctmai`.
4. If you want to make any modifications to the supplied example, familiarize yourself with Application Integrator.
5. Deploy the job type by following the instructions in the Control-M documentation.

Create a Control-M connection profile for AWS Mainframe Modernization

Connection profiles define connection attributes and security credentials for a specific instance of an application. Each connection profile can be referenced by multiple jobs. You can have separate profiles for each unique combination of an application and credentials.

To define connection profiles, you can use the graphical user interface (GUI) available in the **Configuration** domain of the Control-M Web interface, or you can use JSON. For information

about the connection profile of the AWS Mainframe Modernization plug-in, see the Control-M
documentation.

The following code is an example of using JSON:

```
{
  "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
  }
}
```

Create a JSON file that is similar to the example, and deploy it by using the Control-M Automation
API deploy service. For example, if the JSON code is saved to a file named cp-MANAGED-M2-
REPLATFORM.json, the ctm CLI syntax to deploy this connection profile is the following:

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

The response from Control-M Automation API will be similar to the following:

```
[
  {
    "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"
```

```
        ]
    }
]
```

Create jobs and schedules in Control-M Planning

Now that you have the job type deployed and a connection profile for AWS connectivity, you can start creating and running jobs.

Each AWS Mainframe Modernization service job consists of a set of attributes, in four sections. Each section can have many attributes. The following list shows some of the more commonly used attributes.

- General:
  - The job name
  - The application and sub-application that the job belongs to
  - The JCL to submit
  - A link to documentation
- Scheduling:
  - Months and Days this job is eligible to run
  - Calendars such as a business accounting periods, holidays, or other special dates that cannot be defined algorithmically
  - Time windows
  - Cyclic behavior
- Prerequisites:
  - Upstream dependencies (usually jobs, which must be completed successfully before this job becomes eligible to run)
  - Resources that might be required
  - User action that might be required
- Actions performed by Control-M upon completion of the job:
  - Determining job success or failure (usually based on the job's completion codes, but you can override that setting to use output text or checks for a specific status)
  - Notifications, such as email, for failure or success
  - Publishing status for downstream dependencies

As with connection profiles, jobs can be created and deployed in the GUI or in JSON and deployed by using Control-M Automation API.

The following sections examine some common workflow scenarios:

- Initiate jobs based on the status code of previous jobs
- Automate job runs on a scheduled frequency
- Base job runs on events

Initiate jobs based on the status of previous jobs

Create a flow of jobs called a workflow. Jobs in the workflow are interconnected with dependencies on successful completion of the previous jobs.

Using the Control-M Web GUI

To initiate a job from the Control-M user interface, do the following:

1. In the **Planning** domain, add a new **Workspace**. This opens a canvas with an empty folder object.

2. Select the AWS Mainframe Modernization job type (if you are using the supplied job type template, this is called M2JOB), and drag it onto the folder.

3. After the color of the job type changes to green, drop it. The pane on the right contains the **General**, **Scheduling**, **Prerequisites**, and **Actions** sections. To create the job, see the standard instructions in the Control-M documentation.

4. Next, you need the **Job Name** values to configure the job types. You can find the **Job Name** values on the **Application** definition screen of the AWS Mainframe Modernization console or by running the ListBatchJobDefinitions API. For this pilot, repeat steps 2–4 for several jobs, naming each one with any names you choose. Example names are CBANK, CURRENCY, INVFUNDS, BROKERAGE, RISKMGMT, and BANK-SERVICE-Managed-M2.

5. To connect these jobs into the flow that you want, choose the condition triangle under the job object, and drag it onto the next job. For example, choose the condition triangle under CBANK, and drag it onto CURRENCY. This action makes CBANK the predecessor of CURRENCY. By default, CBANK must now be completed successfully before CURRENCY becomes eligible to run.

The following screenshot shows the Control-M Planning view of the basic job flow.
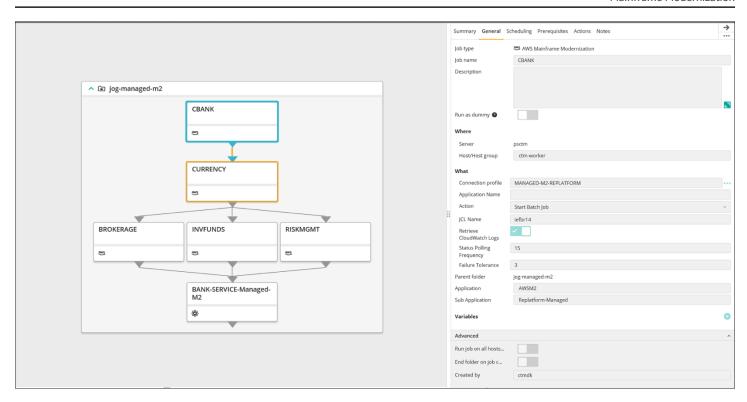
Image provided courtesy of BMC Software, Inc. ©2022

Using JSON

The same flow can be coded in JSON:

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

To deploy this flow, use the deploy service:

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

Automate job runs on a scheduled frequency

Using the flow that you created in the previous steps, you can add basic and runtime scheduling.

- Basic scheduling defines which days a job can run (for example, every weekday, only on working days, month end, or quarter end).

- Runtime scheduling determines when a job will run on the day that it's eligible to run (for example, every hour, after specified resources are available, or only after manual confirmation).

You can set the basic and runtime scheduling on the **Scheduling** tab.

Base job runs on events

Control-M Managed File Transfer (MFT) is an FTP/SFTP client and server that you can use to watch and transfer files between a local host and a remote host. For more information about defining a File Transfer job, see the [Control-M documentation](#).

This pilot uses the File Transfer job to watch for a file-creation event of a file with the .poc extension in the `/bmcfile` folder in an S3 bucket named `bmc-poc-bucket`. When that event occurs, the Control-M job is initiated to run the next job. You can optionally pass the full path, including the bucket name.

Monitor jobs

You can monitor and validate the processing of jobs within the Control-M Monitoring domain and through the AWS Management Console, ensuring comprehensive observation and verification across both these platforms.

Control-M Monitoring

Job submissions and runs can be monitored in the *Control-M Monitoring domain.* By default, AWS Mainframe Modernization service jobs will appear together with all other Control-M work. If you want to see only the AWS Mainframe Modernization service jobs without any other workload (or any other filtering requirements), you can create a *Viewpoint*.

Viewpoints show not only job information but also relationships with upstream and downstream dependencies. Additionally, if your workflow includes AWS Mainframe Modernization and other Control-M job types, you can see and manage the entire flow in the Monitoring domain.

To follow detailed steps, see the [Viewpoints section of Monitoring](#) in the Control-M documentation.

The following screenshot shows the outputs of two workflows. On the left side, the workflow is completed successfully with all jobs in green. On the right side, the workflow is only partially

successful because job CURRRENCY returned the **Failed** status, which is indicated by the red color. The workflow stopped there, leaving the remaining jobs in the **Wait Schedule** state.
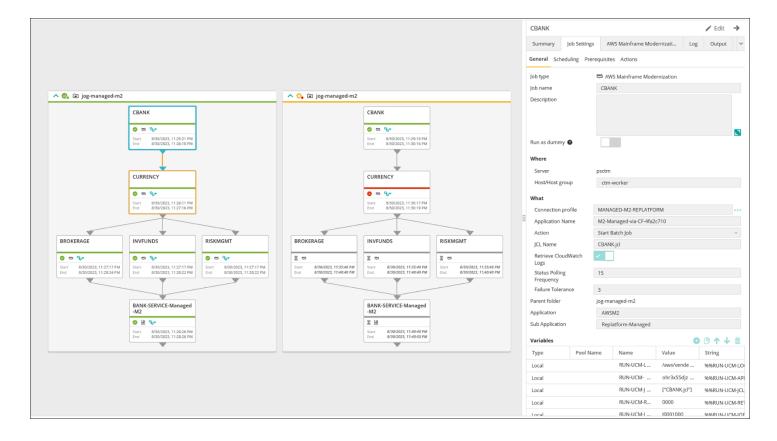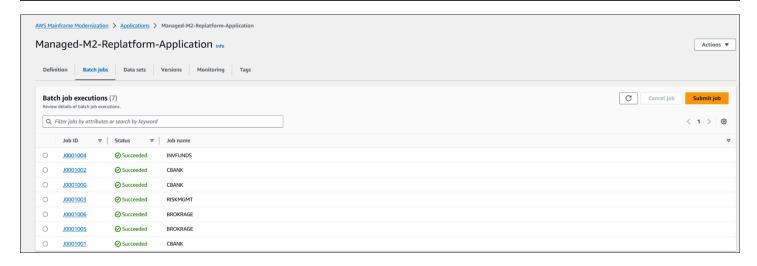

Image provided courtesy of BMC Software, Inc. ©2022

Monitoring on the console

To view job and log information on AWS, sign in to the AWS Management Console, and then navigate to the AWS Mainframe Modernization console.

This view doesn't include dependencies nor any workload that isn't managed by the AWS
Mainframe Modernization service.

AWS Mainframe Modernization on Amazon EC2 integration with Control-M

This section describes how Control-M integrates with and supports batch jobs that run in a custom AWS Mainframe Modernization runtime environment deployed on an EC2 instance. If you are implementing the fully managed AWS Replatform with Micro Focus runtime environment, see the Managed AWS Mainframe Modernization integration with Control-M section.

This section assumes the following prerequisites:

- An active AWS account.

- A virtual private cloud (VPC) where the EC2 instances will be created.

- The mainframe application is migrated and running in an AWS Replatform with Micro Focus environment on an EC2 instance and is supporting the Micro Focus runtime engine with multiple defined batch jobs. For this pilot, follow the instructions at Replatforming applications with Micro Focus. The documentation includes all tasks and additional information on configuring and operating the AWS Replatform with Micro Focus runtime environment on Amazon EC2.

This following topics cover the setup required for integration between Control-M and the AWS Replatform with Micro Focus environment:

- Deploy Control-M and Micro Focus resources

- Create a Control-M connection profile

- Create jobs and schedules in Control-M Planning

- Manage job runs in Control-M by using Monitoring

Deploy Control-M and Micro Focus resources

The first two steps in integrating AWS Mainframe Modernization with Control-M are to deploy a Control-M Agent and then deploy a plug-in on the Agent. The third step is to deploy the job type in Control-M Application Integrator.

Deploy a Control-M Agent on an EC2 instance

When using a custom AWS Replatform with Micro Focus on Amazon EC2 runtime environment, the `MFBSIFJCL` utility is invoked. The utility operates on the host that was launched using the Micro Focus Enterprise Server offer from the AWS Marketplace. To run the `MFBSIFJCL` utility, you must also deploy a Control-M Agent on that host. For instructions, see the [Control-M documentation](#).

> ⓘ **Note**
>
> You can download required installation media from the [BMC Electronic Product Download site](#).

Deploy the Micro Focus plug-in on the Control-M Agent

Plug-ins provide integration and support for the applications and services that Control-M orchestrates.

The plug-in deployment is an infrequent activity. If you already have an existing plug-in that you want to use, skip this step, and create a connection profile.

On Amazon EC2, the AWS Mainframe Modernization Replatform with Micro Focus service exposes the Micro Focus engine. To integrate AWS Mainframe Modernization, Control-M uses the Micro Focus plug-in. For more information, see the [Control-M documentation](#).

You must deploy this plug-in to an Agent that is installed on a host where Micro Focus Enterprise Server is running.

Deploy an AWS Mainframe Modernization job type in Control-M Application Integrator

The job type deployment is usually a one-time activity. If you already have an existing job type that you want to use, skip this step and move on to the next step of [creating a connection profile](#).

The example job type [AIJOB.ctmai](#) is supplied in the git repository. To deploy a job type, you must perform the following steps using [Application Integrator](#):

- Clone the [aws-mainframe-modernization-controlm-integration](#) GitHub repository and download the `AIJOB.ctmai` file to a file system location that can be accessed by Application Integrator.

- Log in to Application Integrator.

- On the **Home** tab, select **Import job type from file**, and select the location of AIM2JOB.ctmai.

- If you want to make any modifications to the supplied example, familiarize yourself with Application Integrator.

- Deploy the job type by following the instructions in the Control-M documentation.

Create a Control-M connection profile

Connection profiles define connection attributes and security credentials for a specific instance of an application. Each connection profile can be referenced by multiple jobs. You can have separate profiles for each unique combination of an application and credentials.

To define connection profiles, you can use the Configuration domain of the Control-M Web interface, or you can use JSON. The following code is an example of using JSON:

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

The example code is in the file ConnectionProfile-Custom-M2-Replatform.json in the GitHub repo. To deploy the code, use the Control-M Automation API deploy service:

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

The response from the Control-M Automation API will be similar to the following:

```
[
```

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

Create jobs and schedules in Control-M Planning

Now that you have the plug-in deployed and a connection profile for Micro Focus connectivity, you can start creating and running jobs.

Each Control-M for Micro Focus job consists of a set of attributes, in four sections. Each section can have many attributes. The following list shows some of the more commonly used attributes.

- General:
  - The job name
  - The application and sub-application that the job belongs to
  - The JCL to submit
  - A link to site-provided operational documentation
- Scheduling:
  - Months and Days this job is eligible to run
  - Calendars such as a business accounting periods, holidays, or other special dates that cannot be defined algorithmically
  - Time windows
  - Cyclic behavior such as running every hour
- Prerequisites
  - Upstream dependencies (usually jobs, which must be completed successfully before this job becomes eligible to run)

- Resources that might be required

- User action that might be required

- Actions performed by Control-M upon completion of the job:

  - Determining job success or failure (usually based on the job's completion codes, but you can override that setting to use output text or checks for a specific status)

  - Notifications, such as email, for failure or success

  - Publishing status for downstream dependencies

As with connection profiles, jobs can be created and deployed by using Control-M Web or written in JSON and deployed by using Control-M Automation API.

The following sections examine some common workflow scenarios:

- Initiate jobs by using a workflow

- Automate job runs on a scheduled frequency

Initiate jobs by using a workflow

Create a flow of jobs called a workflow. Jobs in the workflow are interconnected with dependencies on successful completion of the previous jobs.

Using Control-M Web

To initiate a job from the Control-M user interface, do the following:

1. In the Planning domain, add a new Workspace. This opens a canvas with an empty folder object.

2. On the Menu bar, choose **Add Job**, choose the Micro Focus Windows or Micro Focus Linux job, and drag it onto the face of the folder.

3. After the color of the job changes to green, drop it. The pane on the right contains the **General**, **Scheduling**, **Prerequisites**, and **Actions** sections. To create the job, see the standard instructions in the Control-M documentation.

4. For instructions on building a Control-M for Micro Focus job, see the Control-M documentation. For this pilot, repeat steps 2–4 for five jobs, naming each one from `aws-mf-job1` to `aws-mf-job5`.

5. To connect these jobs into the flow that you want, choose the condition triangle under the job
   object, and drag it onto the next job. For example, choose the condition triangle under `aws-`
   `mf-job1`, and drag it onto `aws-mf-job2`. This action makes `aws-mf-job1` the predecessor of
   `aws-mf-job2`. By default, `aws-mf-job1` must now be completed successfully before `aws-mf-`
   `job2` becomes eligible to run.

The following diagram shows the Control-M Planning view of the basic job flow.
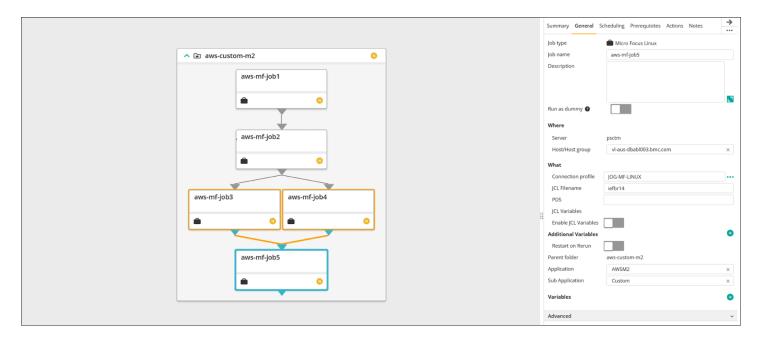

Image provided courtesy of BMC Software, Inc. ©2022

Using JSON

The same flow can be coded in JSON. In the GitHub repo, the example code is in the file `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"
    ]
  }
 }
}
```

To deploy this flow, use the Control-M Automation API:

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

Automate job runs on a scheduled frequency

Using the flow that you created in the previous steps, you can add basic and runtime scheduling.

- Basic scheduling defines which days a job is eligible to run (for example, every weekday, only on working days, month end, or quarter end). You can set the basic scheduling on the Scheduling tab.

- Runtime scheduling determines when a job runs on the day that it's eligible to run (for example, after predecessor jobs complete, every hour, after specified resources are available, or only after manual confirmation).

You can define some runtime scheduling in the **Scheduling** section and others in **Prerequisites** section.

Monitor jobs

You can monitor and verify jobs in the Control-M Monitoring domain and in the Micro Focus Enterprise Server Common Web Administration user interface.

Control-M Monitoring

Job submissions and runs can be monitored in the Control-M Monitoring domain. By default, AWS Mainframe Modernization service jobs will appear together with all other Control-M work. If you want to see only the AWS Mainframe Modernization service jobs without any other workload (or any other filtering requirements), you can create a Viewpoint.

Viewpoints show not only job information but also relationships with upstream and downstream dependencies. Additionally, if your workflow includes AWS Mainframe Modernization and other types of Control-M jobs, you can see and manage the entire flow in the Monitoring domain.

You can follow detailed steps by visiting [Viewpoints section of Monitoring](#) in the Control-M documentation.

The following screenshot shows the outputs of two workflows. On the left side, the workflow is completed successfully with all five jobs in green. On the right side, the workflow is only partially successful because `aws-mf-job3` returned the **Failed** status, and the workflow stopped there, leaving `aws-mf-job5` in **Wait Schedule** state.
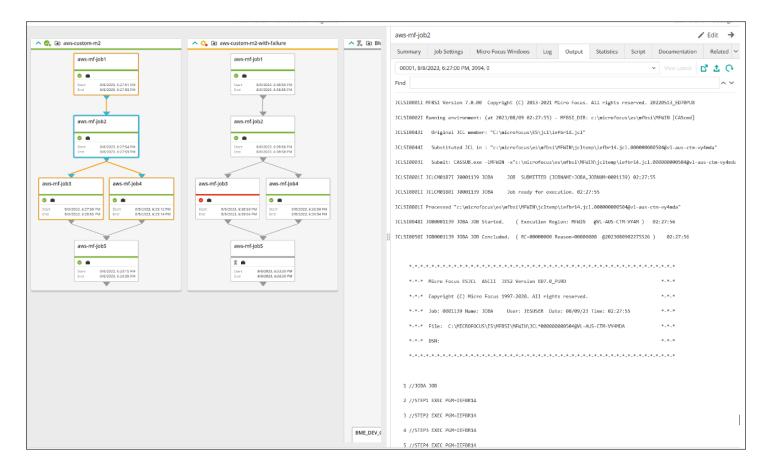

Image provided courtesy of BMC Software, Inc. ©2022

Best practices

During the initial planning and integration stages, we recommend the following best practices:

- Before integration, thoroughly understand the workload and processes that need to be migrated or automated. This helps in identifying the most critical jobs for migration and in planning their scheduling and automation using Control-M.

- When migrating mainframe workloads to AWS, plan for their automation with Control-M from the start. Consider how jobs and workflows will be scheduled, managed, and monitored in the cloud environment.

- We recommend using centralized connection profiles because this approach reduces the number of objects to manage and simplifies elastic deployment of Control-M Agents.

- When possible, perform mainframe migration incrementally to reduce complexity and risk. By doing incremental migration, migration teams can provide faster feedback regarding the migration progress. Businesses can use that feedback to optimize internal processes to accelerate the pace of migration.

- To avoid unnecessary work, consider using the provided templates for job type and connection profile for initial stages.

Related resources

References

- [Micro Focus](#)
- [Control-M](#)
- [Control-M trial](#)
- [Control-M Application Integrator](#)
- [Control-M documentation](#)
- [Mainframe modernization: DevOps on AWS with Micro Focus](#) (AWS Prescriptive Guidance pattern)

Code

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

Contributors

Contributors

Contributors to this document include:

- Sunil Bemarkar, Senior Partner Solutions Architect – DevOps, AWS
- Joe Goldberg, BMC Software, Inc.
- Pablo Alonso Prieto, Senior Mainframe Architect, AWS
- Vaidy Sankaran, Senior Modernization Architect, AWS
- Vij Balakrishn, Senior Partner Development Manager – CloudOps, AWS

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an [RSS feed](#).

Change	Description	Date
[Section added.](#)	A new section covers [AWS Mainframe Modernization on Amazon EC2 integration with Control-M](#).	February 19, 2024
[Initial publication](#)	—	November 16, 2022