Control-M ワークフローオーケストレーターと AWS Mainframe Modernization の統合を使用する

AWS 規範ガイダンス

AWS 規範ガイダンス: Control-M ワークフローオーケストレーターと AWS Mainframe Modernization の統合を使用する

Table of Contents

Control-M ワークフローオーケストレーターと の統合の使用 AWS Mainframe Modernization

Amazon Web Services and BMC Software, Inc. (寄稿者)

2024 年 2 月 (ドキュメント履歴)

ビジネスとデジタルトランスフォーメーションのニーズに対応し、俊敏性の向上、コストの削減、イノベーションの迅速化を実現するため、お客様はメインフレームアプリケーションをモダナイズしています。re:Invent 2021 で、Amazon Web Services (AWS) は、お客様がメインフレームワークロードをモダナイズするのに役立つ AWS Mainframe Modernizationサービスを発表しました。 AWS Mainframe Modernization は、クラウドネイティブなマネージド型の高可用性ランタイム環境を提供します AWS。

このガイドでは、メインフレームのモダナイゼーションジャーニー中にバッチアプリケーションをモダナイズするための追加オプションを提供するために、BMC Helix Control-M ワークフローオーケストレーションを AWS Mainframe Modernization サービスと統合する方法について詳しく説明します。Control-M の機能を利用することで、組織はエンドツーエンドのビューを提供する単一のツールを使用して、既存のワークロードと移行されたワークロードで構成される環境を合理化できます。

概要

Control-M と AWS Mainframe Modernization Replatform を Micro Focus と統合すると、次の利点があります。

- 人材間のギャップの低減に役立つ
- 俊敏性の高い DevOps の手法で迅速なイノベーションを支援する
- 大幅な変更を加えることなく、アプリケーションやデータへのアクセスが簡単になる
- アプリケーションの実行または拡張のコストを最適化する
- ビジネスの俊敏性を最大限に高め、コストを削減する

AWS Replatform with Micro Focus サービスは、メインフレームアプリケーションを AWS クラウドネイティブのマネージドランタイム環境にモダナイズするのに役立ちます。移行とモダナイズの計画および実装に役立つツールとリソースを提供します。ユーザーはバッチジョブの送信またはキャンセル、バッチジョブ実行の詳細の確認ができます。ユーザーがバッチジョブを送信するたびに、 AWS

Replatform with Micro Focus サービスはモニタリングできる個別のバッチジョブ実行を作成します。AWS Mainframe Modernization サービスウェブコンソールを使用して、バッチジョブを名前で検索し、ジョブコントロール言語 (JCL)、スクリプトファイル、パラメータをバッチジョブに提供できます。

Control-M を使用すると、アプリケーションとデータのワークフローの定義、スケジュール設定、管理、監視ができます。こうすることで可視性と信頼性が向上し、サービスレベルアグリーメント (SLA) の改善に役立ちます。Control-M は、オンプレミスと のワークフローを統合し、自動化し、オーケストレーション AWS クラウド して、ビジネスサービスを時間どおりに提供できるようにします。ユーザーは単一の統合グラフィカルビューで、豊富なプラグインライブラリを使用して、ファイル転送、アプリケーション、データソース、インフラストラクチャを含むすべてのワークフローのオーケストレーションができます。クラウドでプロビジョニングされた Control-M は、 のエフェメラル機能を使用します AWS クラウド。継続的インテグレーションと継続的デリバリー (CI/CD) ツールチェーン内の REST API、JSON、Python などによる「コードとしてのジョブ」手法を採用しているため、Control-M のワークフローではバージョン管理、テスト、保守を実行し、開発者と DevOps エンジニアが協力して作業することができます。

> **Note**
>
> AWS Mainframe Modernization AWS Blu Age によるリファクタリングもサポートされています。詳細については、[AWS 「 規範ガイダンスガイド」]を参照してください。

目標とするビジネス成果

このソリューションは、次の目標を達成するのに役立ちます。

ビジネス上の利点

- シームレスな移行とワークフローオーケストレーション – Control-M は、メインフレームワークロードのスムーズな移行を容易にします AWS。複雑なワークフローをオーケストレーションし、移行中と移行後のバッチ処理の継続性を確保します。
- ハイブリッド運用管理 – Control-M を使用すると、移行中のレガシーメインフレームプロセスと新しいクラウドベースのアプリケーションの両方を効率的に管理できます AWS。
- コスト削減とリソース使用の最適化 — Control-M AWS を使用して に移行すると、クラウドリソースの使用とスケーリングが最適化され、運用コストが削減されます。
- 俊敏性とイノベーションの強化 – への移行 AWS は、企業が市場の変化に迅速に適応し、イノベーションにクラウドネイティブサービスを使用するのに役立ちます。
- コンプライアンス、セキュリティ、効率的なモニタリング — Control-M は、継続的なコンプライアンスとセキュリティを確保するのに役立ちます。Control-M は、クラウド環境で強化されたリアルタイムモニタリングおよびレポート機能も提供します。
- 人材ギャップの軽減 – への移行 AWS は、レガシープラットフォームでのスキルのあるメインフレームプロフェッショナルの検索と維持に関連する課題に対処します。

技術的な利点

このソリューションは次の目標達成に役立ちます。

- Control-M 環境を拡張して、 AWS Mainframe Modernization サービスのワークロードを管理します。
- ウィザードを使用して Control-M REST APIsにすばやく組み込むことで、迅速なイノベーションをサポートします。
- Control-M の統合ビューで、すべてのアプリケーションワークロードジョブのオーケストレーション、スケジュール設定、監視を行う。
- 変化する市場状況や顧客の需要により効果的に対応できるように、組織のスケーラビリティとビジネスの俊敏性を高めます。
- メインフレームワークロードをモダナイズし、クラウドネイティブサービスの利点を利用することで、アプリケーションを実行または拡張するためのコストを最適化します。

本ガイドの用語

- メインフレームアプリケーションは、一連のビジネスプロセスを達成、促進するメインフレーム·プログラムおよびサブプログラムのセットを指します。メインフレームアプリケーションは、バッチ処理システムまたはオンライントランザクション処理 (OLTP) システムです。
- バッチジョブは、ユーザーによる操作を必要とせずに実行するように設定された、スケジュールされたプログラムを指します。「Micro Focus を使用した AWS リプラットフォーム」では、バッチジョブ JCL ファイルとバッチジョブバイナリの両方が Amazon Simple Storage Service (Amazon S3) バケットに準備され、両方の場所がアプリケーション定義ファイルに表示されます。
- AWS Mainframe Modernization は、メインフレームアプリケーションを移行、モダナイズ、実行、運用するためのクラウドネイティブサービスです。
- Amazon S3 は、高い耐久性、可用性、パフォーマンスを備えたスケーラブルなオブジェクトストレージです。
- Amazon CloudWatch は、DevOps エンジニア、デベロッパー、サイト信頼性エンジニア (SREs)、IT マネージャー、製品所有者向けに構築されたモニタリングおよびオブザーバビリティサービスです。
- Control-M Web は、ビジネスアプリケーションワークフローの構築、テスト、デプロイ、スケジューリング、監視のライフサイクル全体を通じて、メインフレームのバッチジョブを含むエンタープライズワークロードを管理するためのソリューションです。グラフィカルな機能とプログラム機能が多数用意されているため、すべてのユーザーが最も快適に感じる方法で Control-M の機能にアクセスできます。
- Control-M の Application Integrator は、アプリケーションまたはクラウドサービスと Control-M の統合を可能にするジョブタイプを作成するために使用されるウェブベースのローコードデザイナーです。 Application Integrator ジョブタイプが構築されると、他のすべての Control-M ジョブとまったく同じように動作し、ターゲット統合のすべての Control-M 機能と関数が公開されます。このガイドでは、 と AWS リプラットフォームと Micro Focus の統合について説明します。
- Planning Domain は、ジョブフローの作成と更新を管理するための機能セットです。
- Monitoring Domain は、ジョブフローを管理するための機能セットです。実行中のすべてのジョブとそのステータスがここに表示されます。エラーが発生すると、アラートが生成されます。運用上のアクションを実行して、ジョブのステータス表示、遅延や障害への対応、問題の分析、是正措置の実施ができます。
- Control-M エージェントは、Control-M サーバーによって管理されるさまざまなコンピュータにインストールされます。ジョブはホストを識別するエージェント名に割り当てられます。エージェントは HTTPS プロトコル (ポート 443) を使用してサーバーと通信します。

前提

このガイドの例と図は、次の仮定を反映しています。

- 移行されるメインフレーム･ アプリケーションは、単一のプログラムまたは複数のプログラムを実行することがあります。このガイドの図には、わかりやすくするため、アプリケーションごとに 1 つのプログラムと複数のサブプログラムが表示されています。
- メインフレームアプリケーションは、複数のバッチジョブが定義された Micro Focus マネージドランタイム環境の AWS Replatform に移行され、実行されます。このパイロット版では、「チュートリアル: Micro Focus のマネージドランタイム」の手順に従って、「Micro Focus を使用した AWS リプラットフォーム」で BankDemo サンプルアプリケーションを設定します。 https://docs.aws.amazon.com/m2/latest/userguide/tutorial-runtime.html
- Control-M のフルインストールでは、アプリケーションプラグインやアドオンを含むすべての Control-M コンポーネントが使用できます。
- Control-M エージェントはジョブの管理を担当します。ワークロードを分散するため、エージェントを複数のコンピュータにインストールできます。これにより、パフォーマンスと耐障害性を向上できます。
- Control-M Automation API は、RESTful Web サービス (REST API) を通じて Control-M の機能を公開します。ジョブ、接続プロファイル、ユーザーとロール、サイト標準などのアーティファクトは JSON で記述し、さまざまなサービスへの入力として提供したり、さまざまなサービスによる出力として生成したりできます。サービスには、curl などの機能、または提供されている ctm コマンドラインインターフェイス (CLI) を使用して、HTTPS リクエストから直接アクセスできます。

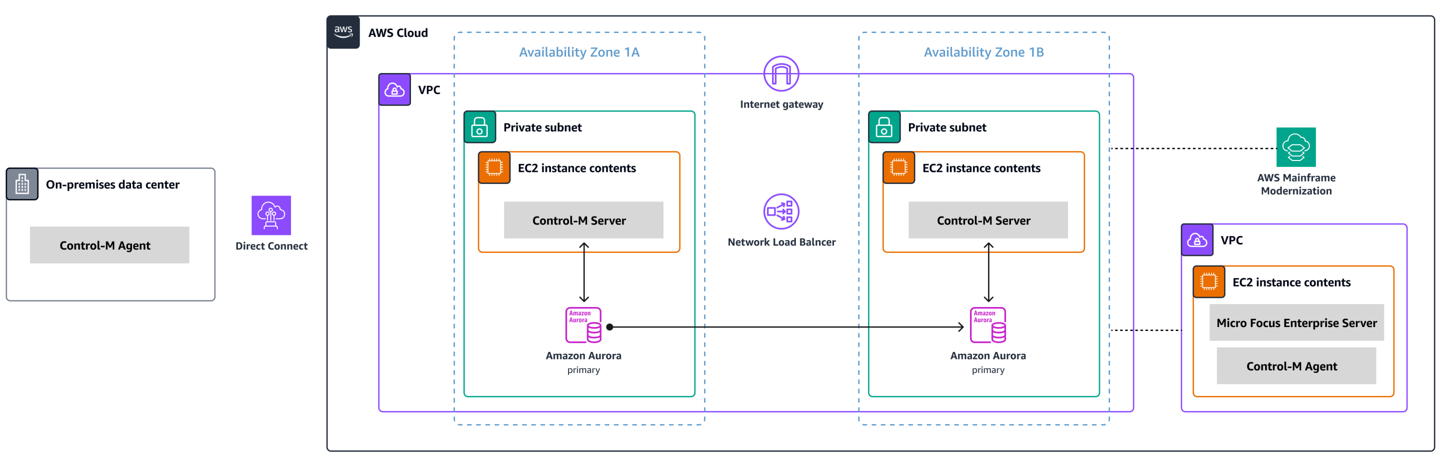
アーキテクチャ

AWS Mainframe Modernization Micro Focus を使用したリプラットフォームは、次の 2 つのモードで使用できます。

- AWS Micro Focus とリプラットフォームは、Micro Focus バックエンドで動的にデプロイされ、によって完全に管理されるサーバーレスマネージドランタイム環境です AWS。 AWS Micro Focus を使用したリプラットフォームは、Micro Focus とやり取りするためのクラウドネイティブ API レイヤーを提供します。このマネージド型アプローチでは、再プラットフォームに使用できるのは Micro Focus のみです。UniKix ソリューションは利用できません。
- AWS Amazon Elastic Compute Cloud (Amazon EC2) での Micro Focus を使用したリプラットフォームは、選択した EC2 インスタンスタイプで起動するプリインストールされた Micro Focus 環境の Amazon マシンイメージ (AMI) として提供されます。このカスタムデプロイでは、ネイティブ Micro Focus を直接公開します。

どちらのモードにも、トランザクションマネージャー、データマッピングツール、スクリーンリーダーとマップリーダー、バッチジョブ実行環境が含まれます。どちらのモードでも、ソースコードの変更を最小限に抑えながら、分散サーバーでメインフレームアプリケーションを実行できます。

次の図は、Control-M が Amazon EC2 インスタンスでホストされているワークフロー統合を示しています。Amazon Aurora データベースは、バッチジョブの管理と実行に必要なデータを維持するために使用されます。アーキテクチャは、高可用性を実現するマルチアベイラビリティーゾーン (マルチ AZ) デプロイです。アプリケーションのバッチジョブとデータは、 AWS Micro Focus ランタイム環境のリプラットフォームでオーケストレーションされます。この図は、Amazon EC2 でのフルマネージドモードとカスタムモードの両方の AWS リプラットフォームを示しています。



この図は、次のリソースを示しています。

1. オンプレミス環境では、Control-M エージェントがインストールされ、IBM Z/OS または他のワークロードで実行されているワークロードを制御できます。x86 で実行されているワークロードは、AWS Direct Connect を介して AWS 環境に接続します。
2. Control-M Server は、高可用性とディザスタリカバリのために、マルチ AZ 環境のアクティブ/パッシブモードで EC2 インスタンスのペアにインストールされます。
3. Control-M (EC2 インスタンスで実行) で使用される Amazon Aurora バックエンドデータベースは、高可用性とディザスタリカバリのためにセカンダリアベイラビリティーゾーンにレプリカと共にデプロイされます。
4. 別の VPC には、プリインストールされた Micro Focus 環境の AMI として配信された Micro Focus を使用した AWS リプラットフォームを持つ EC2 インスタンスが含まれています。Control-M エージェントは、拡張されたジョブ管理機能を提供する Micro Focus ユーティリティとやり取りするためにこのインスタンスにインストールされます。

移行プロジェクト中も、メインフレームサーバーと分散サーバーの両方の非 AWS ロケーションでワークロードを管理している可能性があります。示されているアーキテクチャは規範的なものではなく、一般的な方向性を提供することを目的としています。ディザスタリカバリオプションを含む詳細な設定は、Control-M 実装の一部として構築することをお勧めします。

Control-M とのマネージド AWS Mainframe Modernization 型統合

このセクションでは、Micro Focus ランタイムエンジンでデプロイされたマネージド AWS Mainframe Modernization 環境で実行されるバッチジョブを Control-M と統合し、サポートする方法について説明します。Amazon EC2 で Micro Focus 環境を備えたカスタム AWS リプラットフォームを実装する場合は、AWS Mainframe Modernization Amazon EC2 での Control-M との統合」セクションを参照してください。

このセクションでは、以下の前提条件を前提としています。

- アクティブ AWS アカウント。
- メインフレームアプリケーションは、複数のバッチジョブが定義された Micro Focus マネージドランタイム環境の AWS Replatform に移行され、実行されます。
- このパイロット版では、BankDemo サンプルアプリケーションが でセットアップされています AWS Mainframe Modernization。セットアップ手順については、「チュートリアル: Micro Focus のマネージドランタイム」を参照してください。

以下のトピックでは、さまざまなタイプの統合ワークフローの Control-M スケジューラと AWS Mainframe Modernization 環境の統合に必要なstep-by-stepのセットアップについて説明します。

- Control-M リソースをデプロイする
- AWS Mainframe Modernization 用の Control-M 接続プロファイルを作成する
- Control-M Planning でジョブとスケジュールを作成する
- ジョブのモニタリング

Control-M リソースをデプロイする

AWS Mainframe Modernization を Control-M と統合する最初の 2 つのステップは、プラグインとジョブタイプをデプロイすることです。

AWS Mainframe Modernization Control-M プラグインをデプロイする

プラグインは、Control-M がオーケストレーションするアプリケーションとサービスの統合とサポートを提供します。マネージド AWS Mainframe Modernization サービスの場合は、プラグインをデプロイします AWS Mainframe Modernization 。

プラグインのデプロイは、アクティビティの頻度が低くなります。プラグインを初めてインストールする場合は、Control-M ドキュメントの手順に従ってください。使用する既存のプラグインがある場合は、このステップをスキップして接続プロファイルを作成します。

Control-M Application Integrator で AWS Mainframe Modernization ジョブタイプをデプロイする

ジョブタイプのデプロイは、通常 1 回限りのアクティビティです。使用する既存のジョブタイプがある場合は、この手順をスキップし、次の接続プロファイルの作成に移ります。

サンプルジョブタイプ AIJOB.ctmai は、Git リポジトリで提供されています。ジョブタイプをデプロイするには、Application Integrator を使用して次の手順を実行する必要があります。

1. aws-mainframe-modernization-controlm-integration GitHub リポジトリのクローンを作成し、Application Integrator がアクセスできるファイルシステムの場所に`AIJOB.ctmai`ファイルをダウンロードします。
2. Application Integrator にログインします。
3. [ホーム] タブで [ファイルからジョブタイプをインポート] を選択し、`AIJOB.ctmai` の場所を選択します。
4. 提供されているサンプルに変更を加える場合は、Application Integrator の知識が必要です。
5. Control-M のドキュメントの指示に従い、ジョブタイプをデプロイします。

の Control-M 接続プロファイルを作成する AWS Mainframe Modernization

接続プロファイルは、アプリケーションの特定のインスタンスの接続属性とセキュリティ認証情報を定義します。各接続プロファイルは複数のジョブから参照できます。アプリケーションと認証情報の組み合わせごとに個別のプロファイルを作成できます。

接続プロファイルを定義するには、Control-M Web インターフェイスの設定ドメインで使用可能なグラフィカルユーザーインターフェイス (GUI) を使用するか、JSON を使用できます。 AWS

Mainframe Modernization プラグインの接続プロファイルの詳細については、[Control-M のドキュメント](#)を参照してください。

次のコードは、JSON の使用例です。

```
{
  "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
  }
}
```

この例に似た JSON ファイルを作成し、[Control-M Automation API デプロイサービスを使用してデプロイ](#)します。たとえば、JSON コードが という名前のファイルに保存されている場合`cp-MANAGED-M2-REPLATFORM.json`、この接続プロファイルをデプロイする `ctm` CLI 構文は次のとおりです。

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

Control-M Automation API からのレスポンスは、次のようになります。

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

Control-M Planning でジョブとスケジュールを作成する

ジョブタイプがデプロイされ、AWS 接続の接続プロファイルが作成されたので、ジョブの作成と実行を開始できます。

各 AWS Mainframe Modernization サービスジョブは、4 つのセクションの属性のセットで構成されます。各セクションには多数の属性があります。次のリストは、より一般的に使用される属性の一部を示しています。

- 全般:
  - ジョブの名前
  - ジョブが属するアプリケーションおよびサブアプリケーション
  - 送信する JCL
  - ドキュメントへのリンク
- スケジューリング:
  - このジョブを実行できる月日
  - ビジネス会計期間、祝日、またはアルゴリズムで定義できないその他の特別な日付などのカレンダー
  - 時間枠
  - 周期的行動
- 前提条件:
  - アップストリームの依存関係 (通常は、正常に完了しなければ実行できないジョブ)
  - 必要になる可能性のあるリソース
  - 必要になる可能性のあるユーザーアクション
- ジョブの完了時に Control-M が実行するアクション:
  - ジョブの成功または失敗の判断 (通常はジョブの完了コードに基づくが、出力テキストを使用するか、特定のステータスをチェックするようにその設定を上書きできます)
  - 失敗または成功の通知 (E メールなど)
  - ダウンストリームの依存関係の公開状況

接続プロファイルと同様、ジョブも GUI または JSON で Control-M Automation API を使用して作成およびデプロイできます。

以下のセクションでは、いくつかの一般的なワークフローシナリオについて説明します。

- 以前のジョブのステータスコードに基づいてジョブを開始する
- スケジュールされた頻度でジョブ実行を自動化する
- 基本ジョブはイベントに基づいて実行される

以前のジョブのステータスに基づいてジョブを開始する

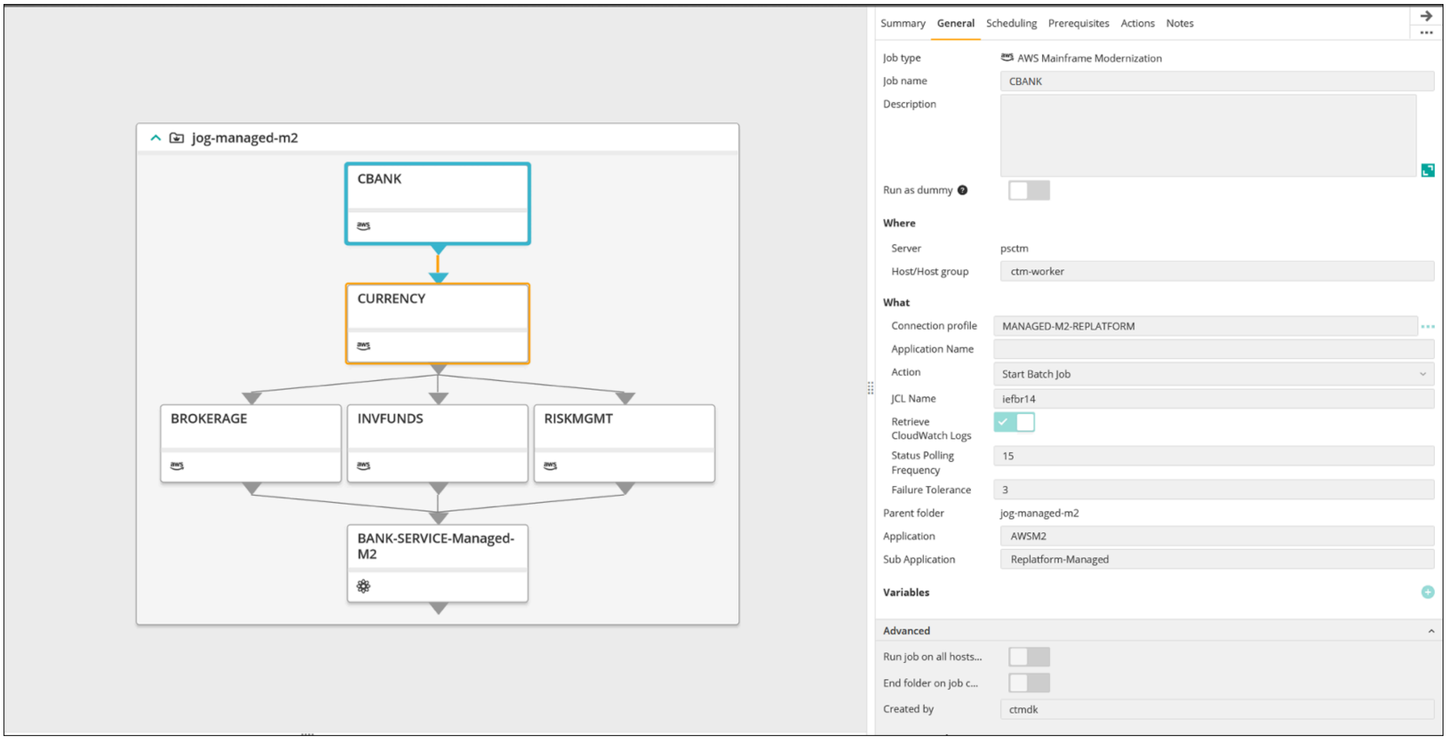
ワークフローと呼ばれるジョブのフローを作成します。ワークフロー内のジョブは、前のジョブが正常に完了すると依存関係と相互に連結されます。

Control-M ウェブ GUI の使用

Control-M ユーザーインターフェイスからジョブを開始するには、次の手順を実行します。

1. 計画ドメインで、新しい Workspace を追加します。これにより、空のフォルダーオブジェクトを含むキャンバスが開きます。
2. AWS Mainframe Modernization ジョブタイプ (指定されたジョブタイプテンプレートを使用している場合は M2JOB) を選択し、フォルダにドラッグします。
3. ジョブタイプの色が緑色に変わったら、ドロップします。右側のペインには、全般、スケジュール、前提条件、アクションのセクションがあります。ジョブを作成するには、Control-M ドキュメントの標準手順を参照してください。
4. 次に、ジョブタイプを設定するには、ジョブ名の値が必要です。ジョブ名の値は、コンソールの AWS Mainframe Modernization アプリケーション定義画面で、または ListBatchJobDefinitions API を実行して確認できます。このパイロットでは、複数のジョブに対してステップ 2～4 を繰り返し、それぞれに任意の名前を付けます。名前の例は、CBANK、CURRENCY、INVFUNDS、BROKERAGE、RISKMGMT、および ですBANK-SERVICE-Managed-M2。
5. これらのジョブを必要なフローに接続するには、ジョブオブジェクトの下にある条件三角形を選択し、次のジョブにドラッグします。例えば、 の下にある条件三角形を選択しCBANK、 にドラッグしますCURRENCY。このアクションは CBANK の前身になりますCURRENCY。デフォルトでは、CURRENCY の実行資格を得る前に、 が正常に完了CBANKする必要があります。

次のスクリーンショットは、基本的なジョブフローの Control-M Planning ビューを示しています。



画像提供: BMC Software, Inc. ©2022

JSON の使用

同じフローを JSON でコーディングできます。

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

このフローをデプロイするには、デプロイサービスを使用します。

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

スケジュールされた頻度でジョブ実行を自動化する

前のステップで作成したフローを使用して、基本スケジュールとランタイムスケジュールを追加できます。

- 基本スケジュールでは、ジョブを実行できる日 (平日、稼働日のみ、月末、四半期末など) を定義します。
- ランタイムスケジューリングは、ジョブが実行可能な日にいつ実行されるかを決定します (たとえば、1 時間ごと、指定されたリソースが利用可能になった後、または手動確認の後のみ)。

基本スケジュールとランタイムスケジュールは、[スケジューリング] タブで設定できます。

基本ジョブはイベントに基づいて実行される

Control-M Managed File Transfer (MFT) は、FTP/SFTP クライアントおよびサーバーとして機能し、ローカルホストとリモートホスト間のファイルの監視や転送に使用できます。File Transfer ジョブの定義に関する詳細は、Control-M のドキュメントを参照してください。

このパイロットでは、File Transfer ジョブを使用して、`bmc-poc-bucket` という名前の S3 バケット内の `/bmcfile` フォルダーにある、拡張子が .poc のファイルのファイル作成イベントを監視します。このイベントが発生すると、Control-M ジョブが開始され、次のジョブが実行されます。オプションで、バケット名を含むフルパスを渡すことができます。

ジョブのモニタリング

Control-M Monitoring ドメイン内および を通じてジョブの処理をモニタリングおよび検証できるため AWS Management Console、これらの両方のプラットフォームで包括的な監視と検証を行うことができます。

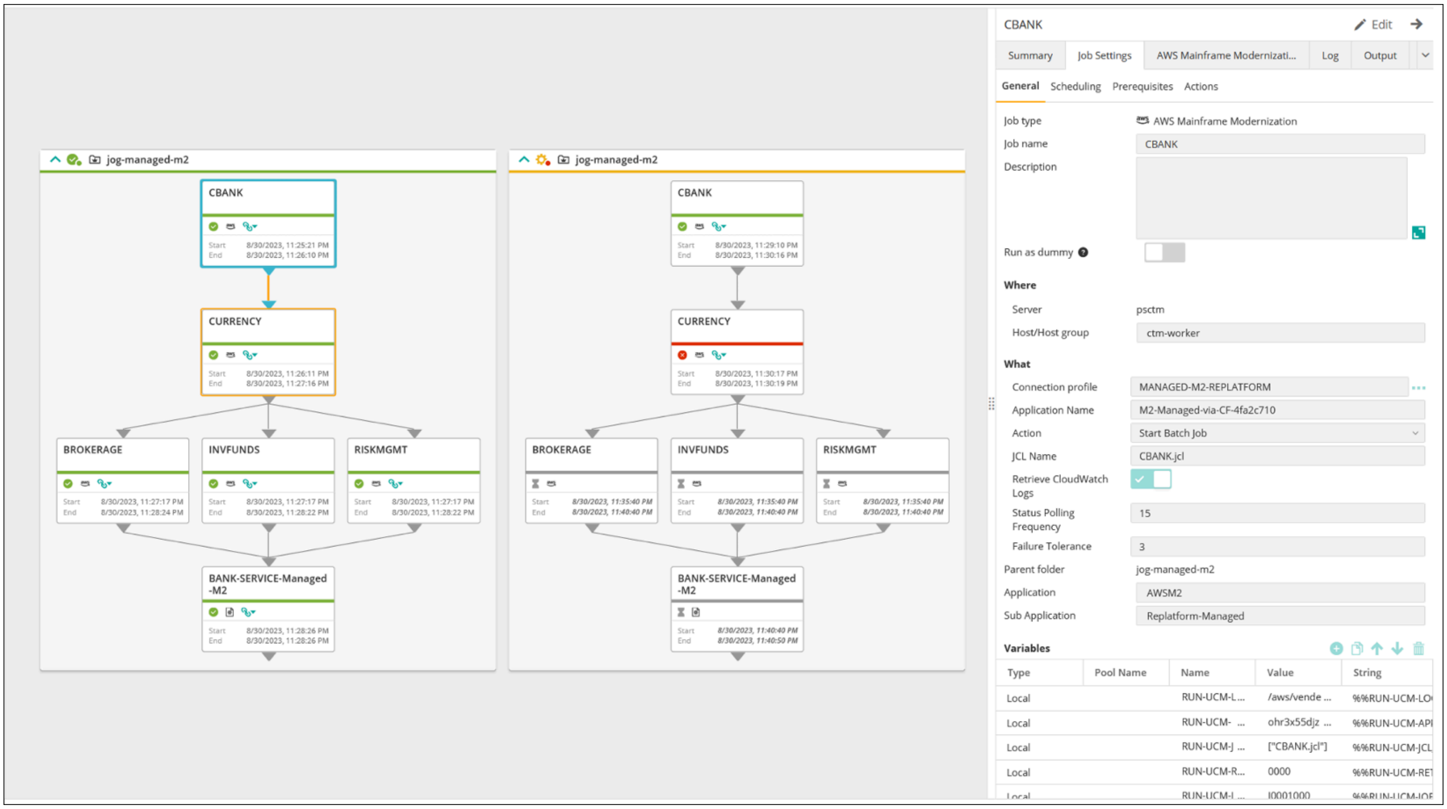
Control-M Monitoring

ジョブの送信と実行は、Control-M Monitoring Domain で監視できます。デフォルトでは、 AWS Mainframe Modernization サービスジョブは他のすべての Control-M 作業とともに表示されます。他のワークロード (またはその他のフィルタリング要件) なしで AWS Mainframe Modernization サービスジョブのみを表示する場合は、Viewpoint を作成できます。

Viewpoint には、ジョブ情報だけではなく、上流と下流の依存関係も表示されます。さらに、ワークフローに AWS Mainframe Modernization およびその他の Control-M ジョブタイプが含まれている場合は、モニタリングドメインでフロー全体を表示および管理できます。

詳細な手順については、Control-M ドキュメントの「モニタリング」の「Viewpoints」セクションを参照してください。

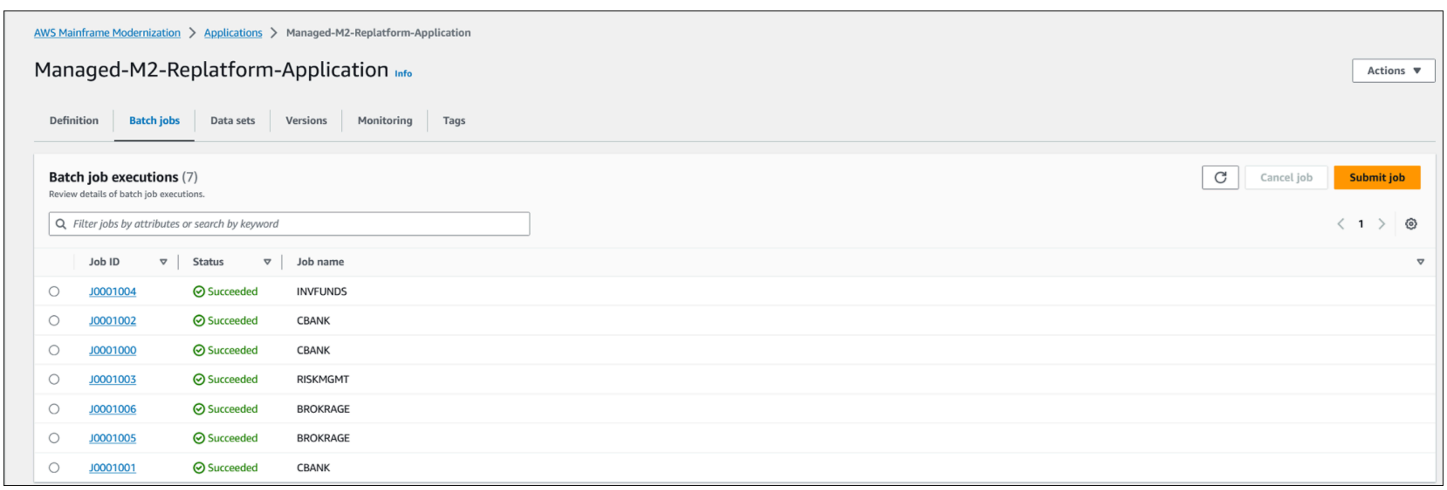
次のスクリーンショットは、2 つのワークフローの出力を示しています。左側では、ワークフローが正常に完了し、すべてのジョブが緑色で表示されます。右側では、ジョブが赤色で示されている失敗ステータスをCURRRENCY返したため、ワークフローは部分的にしか成功しません。ワークフローはそこで停止し、残りのジョブは Wait Schedule 状態のままになります。



画像提供: BMC Software, Inc. ©2022

コンソールでのモニタリング

ジョブとログ情報を表示するには AWS、 にサインインし AWS Management Console、 AWS Mainframe Modernization コンソールに移動します。



このビューには、依存関係や、 AWS Mainframe Modernization サービスによって管理されていないワークロードは含まれません。

AWS Mainframe Modernization Amazon EC2 での Control-M との統合

このセクションでは、EC2 インスタンスにデプロイされたカスタム AWS Mainframe Modernization ランタイム環境で実行されるバッチジョブを Control-M と統合し、サポートする方法について説明します。Micro Focus ランタイム環境でフルマネージド AWS リプラットフォームを実装する場合は、「Managed AWS Mainframe Modernization integration with Control-M」セクションを参照してください。

このセクションでは、以下の前提条件を前提としています。

- アクティブなAWS アカウント
- EC2 インスタンスが作成される Virtual Private Cloud (VPC)。
- メインフレームアプリケーションは、EC2 インスタンスの Micro Focus 環境を備えた AWS Replatform で移行および実行され、複数の定義済みバッチジョブを持つ Micro Focus ランタイムエンジンをサポートしています。このパイロット版については、「Micro Focus を使用したアプリケーションのリプラットフォーム」の手順に従います。ドキュメントには、Amazon EC2 での Micro Focus ランタイム環境を使用した AWS Replatform の設定と運用に関するすべてのタスクと追加情報が含まれています。

以下のトピックでは、Control-M と AWS Replatform と Micro Focus 環境の統合に必要なセットアップについて説明します。

- Control-M および Micro Focus リソースをデプロイする
- Control-M 接続プロファイルを作成する
- Control-M Planning でジョブとスケジュールを作成する
- モニタリングを使用して Control-M でジョブ実行を管理する

Control-M および Micro Focus リソースをデプロイする

AWS Mainframe Modernization を Control-M と統合する最初の 2 つのステップは、Control-M エージェントをデプロイし、エージェントにプラグインをデプロイすることです。3 番目のステップは、Control-M Application Integrator にジョブタイプをデプロイすることです。

EC2 インスタンスに Control-M エージェントをデプロイする

Micro Focus on Amazon EC2 ランタイム環境でカスタム AWS リプラットフォームを使用する場合、 `MFBSIFJCL`ユーティリティが呼び出されます。ユーティリティは、 の Micro Focus Enterprise Server オファーを使用して起動されたホストで動作します AWS Marketplace。`MFBSIFJCL` ユーティリティを実行するには、そのホストに Control-M エージェントもデプロイする必要があります。手順については、[Control-M のドキュメント](#)を参照してください。

> **Note**
>
> 必要なインストールメディアは、 [BMC 電子製品ダウンロードサイト](#)からダウンロードできます。

Control-M エージェントに Micro Focus プラグインをデプロイする

プラグインは、Control-M がオーケストレーションするアプリケーションとサービスの統合とサポートを提供します。

プラグインのデプロイは、アクティビティの頻度が低くなります。使用する既存のプラグインがすでにある場合は、このステップをスキップして接続プロファイルを作成します。

Amazon EC2 では、Micro Focus 付き AWS Mainframe Modernization リプラットフォームサービスは Micro Focus エンジンを公開します。統合するために AWS Mainframe Modernization、Control-M は Micro Focus プラグインを使用します。詳細については、[Control-M のドキュメント](#)を参照してください。

このプラグインは、Micro Focus Enterprise Server が実行されているホストにインストールされている エージェントにデプロイする必要があります。

Control-M Application Integrator で AWS Mainframe Modernization ジョブタイプをデプロイする

ジョブタイプのデプロイは、通常 1 回限りのアクティビティです。使用する既存のジョブタイプがある場合は、この手順をスキップし、次の[接続プロファイルの作成](#)に移ります。

サンプルジョブタイプ [AIJOB.ctmai](#) は、Git リポジトリで提供されています。ジョブタイプをデプロイするには、[Application Integrator](#) を使用して次の手順を実行する必要があります。

- aws-mainframe-modernization-controlm-integration GitHub リポジトリのクローンを作成し、Application Integrator がアクセスできるファイルシステムの場所にAIJOB.ctmaiファイルをダウンロードします。
- Application Integrator にログインします。
- [ホーム] タブで [ファイルからジョブタイプをインポート] を選択し、AIM2JOB.ctmai の場所を選択します。
- 提供されているサンプルに変更を加える場合は、Application Integrator の知識が必要です。
- Control-M のドキュメントの指示に従い、ジョブタイプをデプロイします。

Control-M 接続プロファイルを作成する

接続プロファイルは、アプリケーションの特定のインスタンスの接続属性とセキュリティ認証情報を定義します。各接続プロファイルは複数のジョブから参照できます。アプリケーションと認証情報の組み合わせごとに個別のプロファイルを作成できます。

接続プロファイルを定義するには、Control-M ウェブインターフェイスの設定ドメインを使用するか、JSON を使用できます。次のコードは、JSON の使用例です。

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

サンプルコードは GitHub リポジトリの ConnectionProfile-Custom-M2-Replatform.json ファイルにあります。コードをデプロイするには、Control-M Automation API デプロイサービスを使用します。

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

Control-M Automation API からのレスポンスは、次のようになります。


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


Control-M Planning でジョブとスケジュールを作成する

これで、Micro Focus 接続用のプラグインと接続プロファイルがデプロイされたので、ジョブの作成と実行を開始できます。

Control-M for Micro Focus の各ジョブは、4 つのセクションの一連の属性で構成されます。各セクションには多数の属性があります。次のリストは、より一般的に使用される属性の一部を示しています。

- 全般:
  - ジョブの名前
  - ジョブが属するアプリケーションおよびサブアプリケーション
  - 送信する JCL
  - サイトが提供する運用ドキュメントへのリンク
- スケジューリング:
  - このジョブを実行できる月日
  - ビジネス会計期間、祝日、またはアルゴリズムで定義できないその他の特別な日付などのカレンダー
  - 時間枠
  - 1 時間ごとの実行などの周期的な動作

- 前提条件
  - アップストリームの依存関係 (通常は、正常に完了しなければ実行できないジョブ)
  - 必要になる可能性のあるリソース
  - 必要になる可能性のあるユーザーアクション
- ジョブの完了時に Control-M が実行するアクション:
  - ジョブの成功または失敗の確認 (通常はジョブの完了コードに基づくが、出力テキストを使用するか、特定のステータスをチェックするようにその設定を上書きできます)
  - 失敗または成功の通知 (E メールなど)
  - ダウンストリームの依存関係の公開状況

接続プロファイルと同様に、ジョブは Control-M Web を使用して作成およびデプロイすることも、JSON で記述して Control-M Automation API を使用してデプロイすることもできます。

以下のセクションでは、いくつかの一般的なワークフローシナリオについて説明します。

- ワークフローを使用してジョブを開始する
- スケジュールされた頻度でジョブ実行を自動化する

ワークフローを使用してジョブを開始する

ワークフローと呼ばれるジョブのフローを作成します。ワークフロー内のジョブは、前のジョブが正常に完了すると依存関係と相互に連結されます。

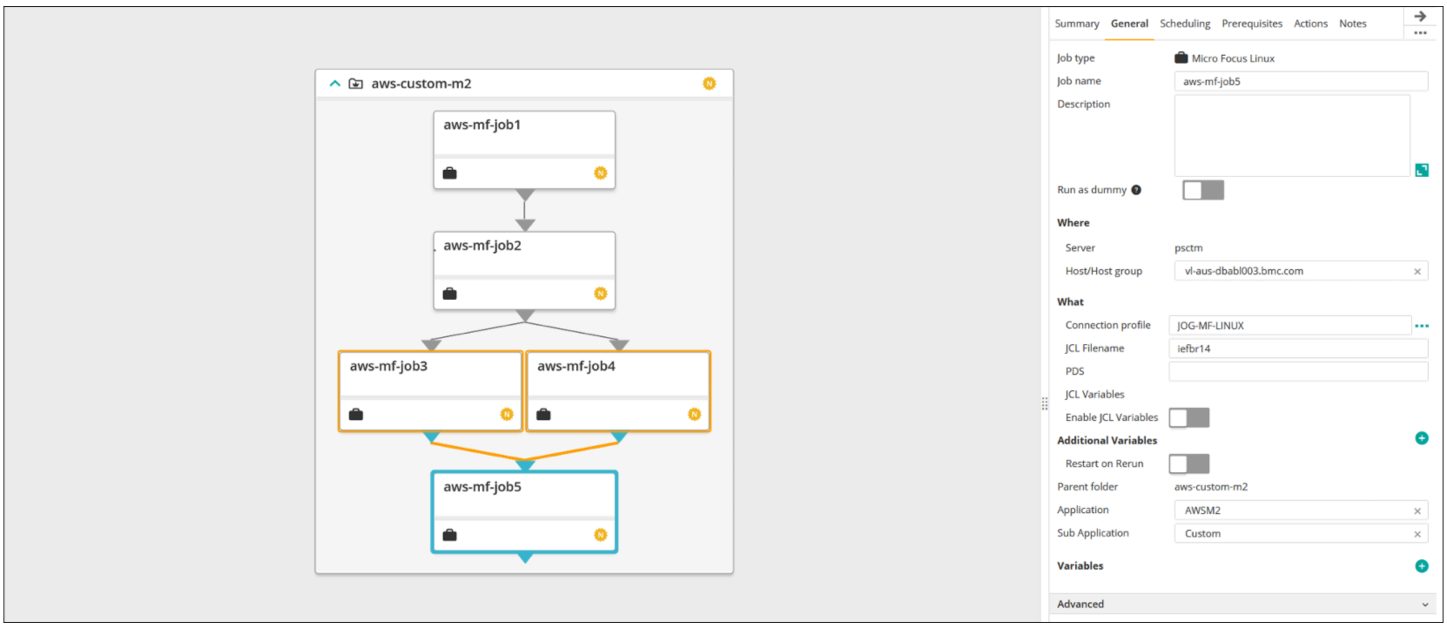
Control-M Web の使用

Control-M ユーザーインターフェイスからジョブを開始するには、次の手順を実行します。

1. Planning Domain に新しいワークスペースを追加します。これにより、空のフォルダーオブジェクトを含むキャンバスが開きます。
2. メニューバーで、ジョブの追加を選択し、Micro Focus Windows または Micro Focus Linux ジョブを選択して、フォルダの面にドラッグします。
3. ジョブの色が緑色に変わったら、ドロップします。右側のペインには、全般、スケジュール、前提条件、アクションのセクションがあります。ジョブを作成するには、Control-M ドキュメントの標準手順を参照してください。

4. Control-M for Micro Focus ジョブを構築する手順については、Control-M ドキュメントを参照してください。このパイロットでは、5 つのジョブに対してステップ 2～4 を繰り返し、それぞれを から aws-mf-job1に命名しますaws-mf-job5。
5. これらのジョブを必要なフローに接続するには、ジョブオブジェクトの下にある条件三角形を選択し、次のジョブにドラッグします。例えば、 の下にある条件三角形を選択しaws-mf-job1、にドラッグしますaws-mf-job2。このアクションは aws-mf-job1 の前身になりますaws-mf-job2。デフォルトでは、 の実行資格aws-mf-job2を得る前に、 が正常に完了aws-mf-job1する必要があります。

次の図は、基本的なジョブフローの Control-M Planning ビューを示しています。



画像提供: BMC Software, Inc. ©2022

JSON の使用

同じフローを JSON でコーディングできます。GitHub リポジトリでは、サンプルコードは ファイルにあります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"
      ]
    }
  }
}
```

このフローをデプロイするには、Control-M Automation API を使用します。

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

スケジュールされた頻度でジョブ実行を自動化する

前のステップで作成したフローを使用して、基本スケジュールとランタイムスケジュールを追加できます。

- 基本スケジューリングは、ジョブの実行対象となる日数を定義します (例えば、平日、営業日、月末、四半期末のみ）。基本的なスケジューリングは、[スケジューリング](#)タブで設定できます。
- ランタイムスケジューリングは、実行可能な日にジョブを実行するタイミングを決定します (たとえば、前のジョブが完了した後、1 時間ごと、指定されたリソースが利用可能になった後、または手動確認の後のみ）。

ランタイムスケジューリングは、「スケジュール」セクションで定義し、「前提条件」セクションで定義できます。

ジョブのモニタリング

Control-M Monitoring ドメインと [Micro Focus Enterprise Server Common Web Administration ユーザーインターフェイス](#)でジョブをモニタリングおよび検証できます。

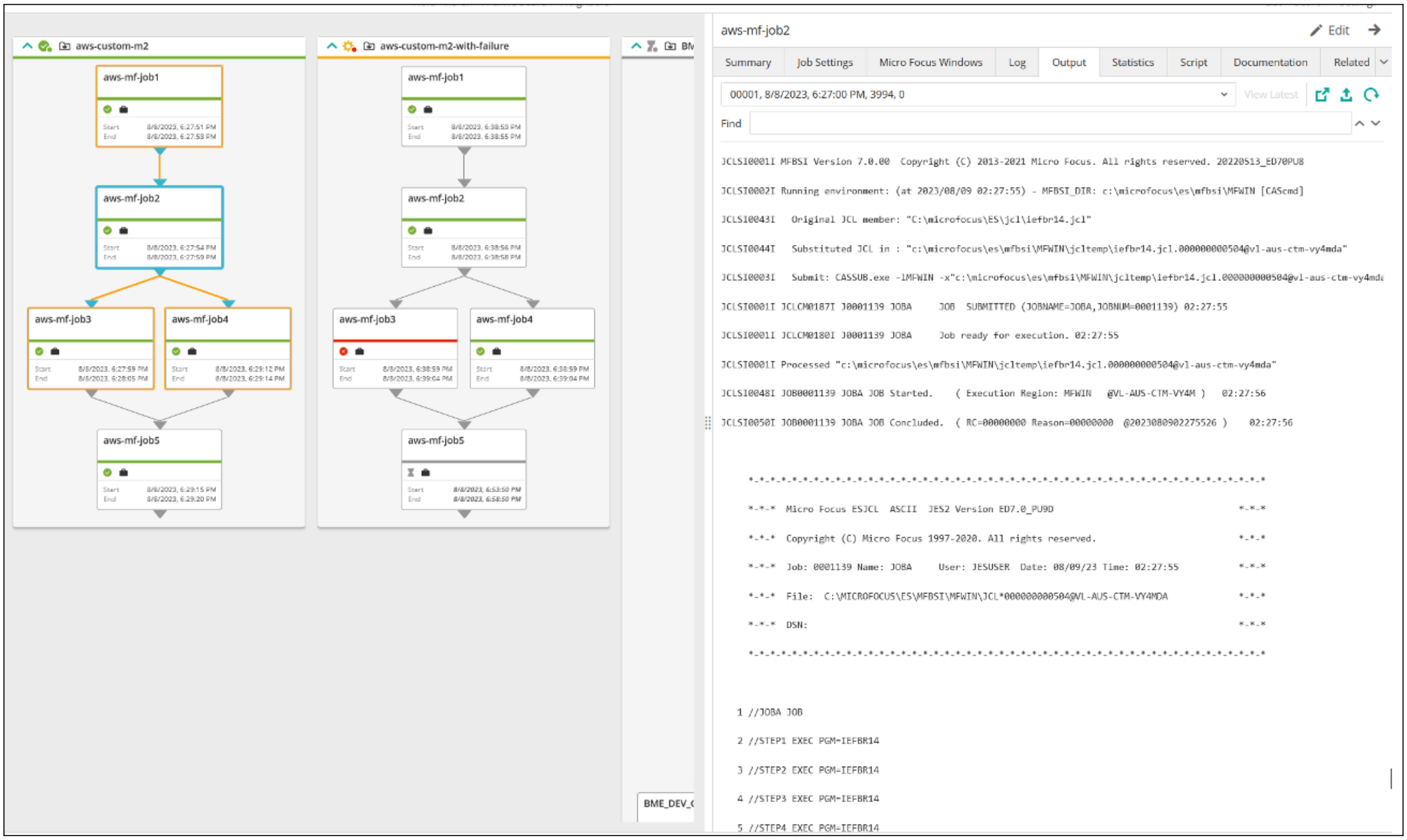
Control-M Monitoring

ジョブの送信と実行は、Control-M Monitoring Domain で監視できます。デフォルトでは、 AWS Mainframe Modernization サービスジョブは他のすべての Control-M 作業とともに表示されます。他のワークロード (またはその他のフィルタリング要件) なしで AWS Mainframe Modernization サービスジョブのみを表示する場合は、Viewpoint を作成できます。

Viewpoint には、ジョブ情報だけではなく、上流と下流の依存関係も表示されます。さらに、ワークフローに AWS Mainframe Modernization およびその他のタイプの Control-M ジョブが含まれている場合は、モニタリングドメインでフロー全体を表示および管理できます。

Control-M ドキュメントの「モニタリング」の「Viewpoints」セクションにアクセスして、詳細なステップを実行できます。

次のスクリーンショットは、2 つのワークフローの出力を示しています。左側では、ワークフローは正常に完了し、5 つのジョブすべてが緑色で表示されています。右側では、ワークフローが部分的にしか成功しません。 は失敗ステータスを`aws-mf-job3`返し、ワークフローはそこで停止し、Wait Schedule 状態のまま`aws-mf-job5`になります。



画像提供: BMC Software, Inc. ©2022

ベストプラクティス

初期計画と統合の段階では、以下のベストプラクティスに従うことをお勧めします。

- 統合する前に、移行または自動化する必要があるワークロードとプロセスを十分に理解してください。これにより、移行に最も重要なジョブを特定し、Control-M を使用してスケジューリングと自動化を計画できます。
- メインフレームワークロードを に移行するときは AWS、Control-M を使用して最初から自動化を計画します。クラウド環境でジョブとワークフローをスケジュール、管理、モニタリングする方法を検討します。
- 一元化された接続プロファイルを使用して、管理するオブジェクトの数を減らし、Control-M エージェントの伸縮自在なデプロイを簡素化することをお勧めします。
- 可能であれば、メインフレームの移行を段階的に実行して、複雑さとリスクを軽減します。段階的な移行を行うことによって、移行チームは移行の進捗状況に関するフィードバックを迅速に提供することができます。企業は、そのフィードバックを使用して内部プロセスを最適化し、移行のペースを加速できます。
- 不要な作業を避けるため、初期段階では提供されているジョブタイプと接続プロファイル用のテンプレートを使用することを検討してください。

関連リソース

リファレンス

- Micro Focus
- Control-M
- Control-M トライアル
- Control-M アプリケーションインテグレーター
- Control-M ドキュメント
- メインフレームのモダナイゼーション: DevOps on AWS with Micro Focus （AWS 規範ガイダンスパターン）

Code

- aws-mainframe-modernization-controlm-integration GitHub リポジトリ

寄稿者

寄稿者

本ドキュメントの寄稿者は次のとおりです。

- Sunil Bemarkar、シニアパートナーソリューションアーキテクト – DevOps、 AWS
- Joe Goldenberg、 BMC Software, Inc。
- Pablo Alonso Prieto、シニアメインフレームアーキテクト、 AWS
- Vaidy Sankaran、シニアモダナイゼーションアーキテクト、 AWS
- Vij Balakrishn、CloudOps シニアパートナー開発マネージャー、 AWS

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
セクションを追加しました。	新しいセクションではAWS Mainframe Modernization 、Amazon EC2 と Control-M の統合について説明します。	2024 年 2 月 19 日
初版発行	—	2022 年 11 月 16 日