Red Hat Decision Manager 7.13

Red Hat OpenShift Container Platform への Red Hat Decision Manager のデプロイメント

Last Updated: 2023-10-14

Red Hat Decision Manager 7.13 Red Hat OpenShift Container Platform への Red Hat Decision Manager のデプロイメント

法律上の通知

概要

本書では、オーサリング環境、管理サーバー環境、イミュータブルサーバー環境、その他のサポートされる環境など、Red Hat OpenShift Container Platform でさまざまな Red Hat Decision Manager 環境をデプロイする方法を説明します。

目次

はじめに

開発者またはシステム管理者は、オーサリング環境、管理サーバー環境、イミュータブルサーバー環境、その他のサポートされる環境など、Red Hat OpenShift Container Platform でさまざまな Red Hat Decision Manager 環境をデプロイできます。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みにより、これらの変更は今後の複数のリリースに対して段階的に実施されます。詳細は、**Red Hat CTO である Chris Wright のメッセージ** をご覧ください。

パート I. OPERATOR を使用した RED HAT OPENSHIFT CONTAINER PLATFORM 4 への RED HAT DECISION MANAGER 環境のデプロイメント

システムエンジニアは、Red Hat OpenShift Container Platform 4 に Red Hat Decision Manager 環境をデプロイしてサービスや他のビジネスアセットを開発または実行するインフラストラクチャーを提供します。OpenShift Operator を使用して、構造化された YAML ファイルに定義された環境をデプロイして、必要に応じてこの環境を維持して変更できます。

前提条件

- Red Hat OpenShift Container Platform 4 の環境を利用できる。現在のリリースがサポートする OpenShift Container Platform の正確なバージョンについては、Red Hat Decision Manager 7 でサポートされる設定 を参照してください。
- デプロイメントする OpenShift プロジェクトが作成されている。
- OpenShift Web コンソールを使用してプロジェクトにログインしている。
- 以下のリソースが OpenShift クラスターで利用できる。アプリケーションの負荷によっては、許容可能なパフォーマンスのために、より多くのリソース割り当てが必要になることがあります。
  - オーサリング環境の場合は、Business Central Pod 用に 4 ギガバイトのメモリーと 2 つの仮想 CPU コアが必要です。高可用性のデプロイメントでは、レプリカごとにこれらのリソースが必要で、2 つのレプリカがデフォルトで作成されます。
  - 各 KIE Server Pod の各レプリカについて、2 ギガバイトのメモリーと 1 つの仮想 CPU コア。
  - 高可用性オーサリングのデプロイメントでは、Red Hat AMQ および Red Hat Data Grid の Pod に、設定されたデフォルトに応じて追加のリソースが必要になります。

注記

MaxMetaspaceSize のデフォルト値は以下の通りです。

- Business Central イメージ: 1024m
- KIE Server イメージ: 512m
- その他のイメージ: 256m

- 動的永続ボリューム (PV) のプロビジョニングが有効になっている。または、動的 PV プロビジョニングが有効でない場合は、十分な永続ボリュームが利用できる状態でなければなりません。デフォルトでは、デプロイされるコンポーネントには以下の PV サイズが必要です。
  - デフォルトでは、Business Central は 1 Gi 分の PV が必要です。Business Central 永続ストレージの PV サイズを変更できます。
- 高可用性オーサリング環境をデプロイする場合は、OpenShift 環境が **ReadWriteMany** モードの永続ボリュームをサポートている。ご使用の環境がこのモードに対応していない場合は、NFS を使用してボリュームをプロビジョニングできます。OpenShift のパブリックおよび専用クラウドでのアクセスモードのサポートに関する情報は、Red Hat OpenShift Container Platform ドキュメントの アクセスモード を参照してください。

第1章 RED HAT OPENSHIFT CONTAINER PLATFORM における RED HAT DECISION MANAGER の概要

Red Hat Decision Manager は、Red Hat OpenShift Container Platform 環境にデプロイすることができます。

この場合、Red Hat Decision Manager のコンポーネントは、別の OpenShift Pod としてデプロイされます。各 Pod のスケールアップおよびスケールダウンを個別に行い、特定のコンポーネントに必要な数だけコンテナーを提供できます。標準の OpenShift の手法を使用して Pod を管理し、負荷を分散できます。

以下の Red Hat Decision Manager の主要コンポーネントが OpenShift で利用できます。

- KIE Server (**実行サーバー (Execution Server)** とも呼ばれる) は、デシジョンサービスおよびその他のデプロイ可能なアセット (**サービス** と総称される) を実行するインフラストラクチャー要素です。サービスのすべてのロジックは実行サーバーで実行されます。
  一部のテンプレートでは、KIE Server Pod をスケールアップして、同一または異なるホストで実行するコピーを必要な数だけ提供できます。Pod のスケールアップまたはスケールダウンを行うと、そのコピーはすべて同じサービスを実行します。OpenShift は負荷分散を提供しているため、要求はどの Pod でも処理できます。

  KIE Server Pod を個別にデプロイし、サービスの異なるグループを実行することができます。この Pod もスケールアップやスケールダウンが可能です。複製された個別の KIE Server Pod を必要な数だけ設定することができます。

- Business Central は、オーサリングサービスに対する Web ベースのインタラクティブ環境です。Business Central は管理コンソールも提供します。Business Central を使用してサービスを開発し、それらを KIE Server にデプロイできます。
  Business Central は一元化アプリケーションです。複数の Pod を実行し、同じデータを共有する高可用性用に設定できます。

  Business Central には開発するサービスのソースを保管する Git リポジトリーが含まれます。また、ビルトインの Maven リポジトリーも含まれます。設定に応じて、Business Central はコンパイルしたサービス (KJAR ファイル) をビルドイン Maven リポジトリーに配置できます (設定した場合は外部 Maven リポジトリーにも可能)。

OpenShift 内でさまざまな環境設定にこのコンポーネントおよびその他のコンポーネントを配置できます。

1.1. オーサリング環境のアーキテクチャー

Red Hat Decision Manager では、Business Central のコンポーネントに、オーサリングサービス用の Web ベースの対話型ユーザーインターフェイスが含まれています。KIE Server のコンポーネントでこれらのサービスを実行します。

Business Central を使用して、KIE Server 上でサービスをデプロイすることもできます。複数の KIE Server を使用して異なるサービスを実行して同じ Business Central から複数のサーバーを制御できます。

単一のオーサリング環境
単一のオーサリング環境では、Business Central のインスタンスが 1 つだけ実行されます。複数のユーザーが同時に Web インターフェイスにアクセスできますが、パフォーマンスが制限される可能性があり、フェイルオーバー機能はありません。

Business Central には、開発したサービスの各種ビルドバージョン (KJAR ファイル/アーティファクト)

を格納する、ビルトイン Maven リポジトリーが含まれています。継続的インテグレーション/継続的デプロイメント (CICD) ツールを使用して、リポジトリーからこのようなアーティファクトを取得し、必要に応じて移動できます。

Business Central は、ビルトインの Git リポジトリーにソースコードを保存します (**.niogit** ディレクトリーに保存)。組み込まれたインデックスメカニズムを使用して、サービス内でアセットをインデックス化します。

Business Central では、Maven リポジトリーと Git リポジトリーに永続ストレージを使用します。

単一のオーサリング環境には、デフォルトで KIE Server インスタンスが 1 台含まれています。

単一のオーサリング環境では、**コントローラーストラテジー** を使用できます。Business Central には、KIE Server を管理できるコンポーネントである **コントローラー** が含まれています。Business Central に接続するように KIE Server を設定した場合、KIE Server は REST API を使用してコントローラーに接続します。この接続を使用すると、WebSocket が永続的に解放されます。コントローラーストラテジーを使用する OpenShift デプロイメントでは、KIE Server インスタンスはそれぞれ、Business Central コントローラーに接続するように初期設定されます。

Business Central ユーザーインターフェイスを使用して KIE Server でサービスをデプロイしたり管理したりする場合、KIE Server はコントローラー接続の WebSocket を使用して要求を受け取ります。サービスをデプロイする場合は、KIE Server が Business Central の一部である Maven リポジトリーから必要なアーティファクトを要求します。

クライアントアプリケーションは、REST API 経由で、KIE Server で実行されるサービスを使用します。

図1.1 単一のオーサリング環境のアーキテクチャー図

KIE Server のクラスタリングと複数の KIE Server の使用

KIE Server Pod をスケーリングして、KIE Server のクラスター環境を実行できます。

クラスターデプロイメントでは、複数の KIE Server インスタンスが同じサービスを実行します。このようなサーバーは、Business Central コントローラーから同じ要求を受信できるように、同じサーバー ID を使用して Business Central コントローラーに接続します。Red Hat OpenShift Container Platform ではサーバー間の負荷分散が可能です。同じクライアントからの要求が別のインスタンスで処理される可能性があるため、クラスター化された KIE Server インスタンスで実行するサービスは、ステートレスでなければなりません。

独立した KIE Server を複数デプロイして、異なるサービスを実行することも可能です。このような場合、サーバーは異なるサーバー ID 値を指定して Business Central コントローラーに接続します。各サーバーにサービスをデプロイする場合は、Business Central UI を使用できます。

Smart Router

任意の Smart Router コンポーネントは、クライアントアプリケーションと KIE Server インスタンスの間にレイヤーを提供します。独立した KIE Server インスタンスを複数使用する場合に役立ちます。

クライアントアプリケーションは、異なる KIE Server インスタンスで実行されるサービスを使用できますが、常に Smart Router に接続されます。Smart Router は自動的に、必要なサービスを実行する KIE Server インスタンスに要求を渡します。また、Smart Router では、サービスのバージョン管理も可能で、追加の負荷分散レイヤーも提供されます。

高可用性オーサリング環境

高可用性 (HA) のオーサリング環境では Business Central Pod がスケーリングされるため、複数の Business Central インスタンスが実行されます。Red Hat OpenShift Container Platform は、ユーザー要求の負荷分散を提供します。この環境は、複数のユーザーに最適なパフォーマンスを提供し、フェイルオーバーをサポートします。

Business Central の各インスタンスには、構築されたアーティファクト用の Maven リポジトリーが含まれており、ソースコードには **.niogit** の Git リポジトリーを使用します。このインスタンスは、リポジトリー用に共有の永続ストレージを使用します。このストレージには、**ReadWriteMany** アクセス権のある永続ボリュームが必要です。

Red Hat DataGrid のインスタンスは、Business Central で開発されたすべてのプロジェクトとアセットをインデックス化します。

Red Hat AMQ インスタンスは、Business Central のすべてのインスタンス間に、Java CDI メッセージを伝播します。たとえば、新規プロジェクトが作成された場合、アセットがインスタンスの 1 つでロックまたは変更された場合に、その情報が即座に他の全インスタンスで反映されます。

コントローラーストラテジーは、クラスターデプロイメントには適していません。OpenShift デプロイメントの場合は、高可用性の Business Central は**OpenShift スタートアップストラテジー** を使用して KIE Server を管理する必要があります。

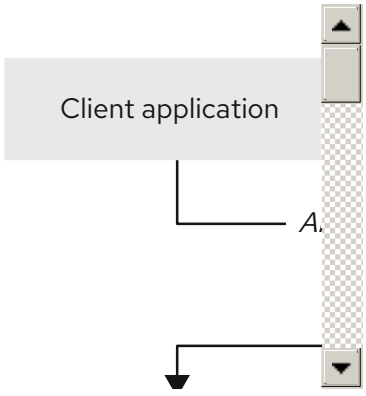
KIE Server デプロイメント (スケーリング可能) ごとに、現在の状態を反映する ConfigMap を作成します。Business Central は、ConfigMap を読み込むことで全 KIE Server を検出します。

ユーザーが KIE Server 設定 (例: サービスのデプロイまたはアンデプロイ) で変更を要求した場合に、Business Central は KIE Server への接続を開始し、REST API 要求を送信します。KIE Server は、すべてのインスタンスが再デプロイされ、新規設定が反映されるように、ConfigMap を変更して新しい設定の状態を反映してから、その再デプロイをトリガーします。ConfigMaps の詳細は、KIE configuration and ConfigMaps を参照してください。

OpenShift 環境で、独立した KIE Server を複数デプロイできます。KIE Server にはそれぞれ、必要な設定が指定された個別の ConfigMap が設定されます。KIE Server は個別にスケーリングできます。

OpenShift デプロイメントに、Smart Router を追加できます。

図1.2 高可用性オーサリング環境のアーキテクチャー図



第2章 OPENSHIFT 環境への RED HAT DECISION MANAGER のデプロイメントの準備

OpenShift 環境に Red Hat Decision Manager をデプロイする前に、準備手順をいくつか完了する必要があります。追加イメージ (たとえば、デシジョンサービスの新しいバージョン、または別のデシジョンサービス) をデプロイする場合は、この手順を繰り返す必要はありません。

注記

トライアル環境をデプロイする場合は、「Red Hat レジストリーに対してお使いの環境が認証されていることを確認する方法」 で説明されている手順を完了し、その他の準備手順は行わないでください。

2.1. RED HAT レジストリーに対してお使いの環境が認証されていることを確認する方法

Red Hat OpenShift Container Platform で Red Hat Decision Manager コンポーネントをデプロイするには、OpenShift が Red Hat レジストリーから正しいイメージをダウンロードできるようにする必要があります。

OpenShift は、お使いのサービスアカウントのユーザー名とパスワードを使用して Red Hat レジストリーへの認証が行われるように設定する必要があります。この設定は namespace ごとに固有であり、Operator が機能している場合は、**openshift** namespace に対する設定がすでに完了しています。

ただし、Red Hat Decision Manager のイメージストリームが **openshift** namespace にない場合や、Red Hat Decision Manager を新規バージョンに自動更新するように設定されている場合、Operator はこのプロジェクトの namespace にイメージをダウンロードする必要があります。対象の namespace の認証設定を完了する必要があります。

手順

1. **oc** コマンドで OpenShift にログインして、プロジェクトがアクティブであることを確認します。
2. Registry Service Accounts for Shared Environments で説明されている手順を実行します。Red Hat カスタマーポータルにログインして、このドキュメントにアクセスし、レジストリーサービスアカウントを作成する手順を実行します。
3. **OpenShift Secret** タブを選択し、**Download secret** のリンクをクリックして、YAML シークレットファイルをダウンロードします。
4. ダウンロードしたファイルを確認して、**name:** エントリーに記載の名前をメモします。
5. 以下のコマンドを実行します。

```
oc create -f <file_name>.yaml
oc secrets link default <secret_name> --for=pull
oc secrets link builder <secret_name> --for=pull
```

<file_name> はダウンロードしたファイルに、**<secret_name>** はファイルの **name:** のエントリーに記載されている名前に置き換えてください。

2.2. KIE SERVER のシークレットの作成

OpenShift は **シークレット** と呼ばれるオブジェクトを使用してパスワードやキーストアなどの機密情報を保持します。OpenShift のシークレットに関する詳細は、Red hat OpenShift Container Platform ドキュメントの What is a secret を参照してください。

KIE Server では HTTPS でアクセスできるように SSL 証明書を使用します。このデプロイメントでは、サンプルシークレットを自動的に作成できます。ただし、実稼働環境では、KIE Server の SSL 証明書を作成し、これをシークレットとして OpenShift 環境に提供する必要があります。

手順

1. KIE Server の SSL 暗号化向けの秘密鍵と公開鍵で **keystore.jks** という名前の SSL キーストアを生成します。キーストアの作成と証明書の使用の詳細については、How to Configure Server Security を参照してください。

注記

実稼働環境で、想定されている KIE Server の URL と一致する、有効な署名済み証明書を生成します。

2. 証明書の名前をメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **jboss** です。

3. キーストアファイルのパスワードをメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **mykeystorepass** です。

4. **oc** コマンドを使用して、新しいキーストアファイルからシークレット **kieserver-app-secret** を生成します。

```
$ oc create secret generic kieserver-app-secret --from-file=keystore.jks
```

2.3. BUSINESS CENTRAL へのシークレットの作成

HTTPS アクセスを提供するために、Business Central では SSL 証明書を使用します。このデプロイメントでは、サンプルシークレットを自動的に作成できます。ただし、実稼働環境では、Business Central の SSL 証明書を作成し、これをシークレットとして OpenShift 環境に提供する必要があります。

Business Central と KIE Server に同じ証明書およびキーストアを使用しないでください。

手順

1. KIE Server の SSL 暗号化向けの秘密鍵と公開鍵で **keystore.jks** という名前の SSL キーストアを生成します。キーストアの作成と証明書の使用の詳細については、How to Configure Server Security を参照してください。

注記

実稼働環境で、Business Central の予想される URL と一致する有効な署名済み証明書を生成します。

2. 証明書の名前をメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **jboss** です。

3. キーストアファイルのパスワードをメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **mykeystorepass** です。

4. **oc** コマンドを使用して、新しいキーストアファイルからシークレット **businesscentral-app-secret** を生成します。

```
$ oc create secret generic businesscentral-app-secret --from-file=keystore.jks
```

2.4. AMQ ブローカー接続のシークレットの作成

KIE Server を AMQ ブローカーに接続し、AMQ ブローカー接続に SSL を使用する場合は、接続の SSL 証明書を作成し、これを OpenShift 環境にシークレットとして指定する必要があります。

手順

1. KIE Server の SSL 暗号化向けの秘密鍵と公開鍵で **keystore.jks** という名前の SSL キーストアを生成します。キーストアの作成と証明書の使用の詳細については、How to Configure Server Security を 参照してください。

注記

実稼働環境で、AMQ ブローカー接続の予想される URL に一致する有効な署名済みの証明書を生成します。

2. 証明書の名前をメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **jboss** です。

3. キーストアファイルのパスワードをメモします。Red Hat Decision Manager 設定におけるこのデフォルト名は **mykeystorepass** です。

4. **oc** コマンドを使用して、新しいキーストアファイルから **broker-app-secret** という名前のシークレットを生成します。

```
$ oc create secret generic broker-app-secret --from-file=keystore.jks
```

2.5. GIT フックの準備

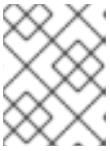
オーサリング環境では、Business Central のプロジェクトのソースコードが変更された場合に Git フックを使用してカスタムの操作を実行できます。Git フックは一般的に、アップストリームのリポジトリーを操作する時に使用します。

Git フックが SSH 認証を使用してアップストリームのリポジトリーを操作できるようにするには、リポジトリーに、認証用の秘密鍵と既知のホストファイルも指定する必要があります。

Git フックを設定しない場合は、この手順を飛ばして次に進んでください。

手順

1. Git フックファイルを作成します。方法は、Git hooks reference documentation を参照してください。



注記

Business Central では **pre-commit** スクリプトはサポートされません。**post-commit** スクリプトを使用してください。

2. 設定マップ (ConfigMap)、またはこれらのファイルを含む永続ボリュームを作成します。ConfigMaps の詳細は、KIE configuration and ConfigMaps を参照してください。

   - Git フックが 1 つまたは複数の固定スクリプトファイルで設定される場合は、**oc** コマンドを使用して設定アップを作成します。以下に例を示します。

     ```
     oc create configmap git-hooks --from-file=post-commit=post-commit
     ```

   - Git フックはロングファイルで設定されるか、実行可能ファイルや JAR ファイルなどのバイナリーに依存する場合は、永続ボリュームを使用します。永続ボリュームと永続ボリューム要求を作成し、ボリュームと要求を関連付けて、このファイルをボリュームに転送する必要があります。
     永続ボリュームおよび永続ボリューム要求の手順については、Red Hat OpenShift Container Platform ドキュメントの Storage を参照してください。永続ボリュームへのファイルのコピー方法は、Transferring files in and out of containers を参照してください。

3. Git フックスクリプトが SSH 認証を使用してアップストリームのリポジトリーと対話する必要がある場合は、必要なファイルでシークレットを作成します。

   a. リポジトリーに格納されている公開鍵に一致する秘密鍵を使用して、**id_rsa** ファイルを作成します。

   b. リポジトリーの正しい名前、アドレス、公開鍵で **known_hosts** ファイルを作成します。

   c. 以下のように **oc** コマンドを使用して、2 つのファイルでシークレットを作成します。

      ```
      oc create secret git-hooks-secret --from-file=id_rsa=id_rsa --from-
      file=known_hosts=known_hosts
      ```

注記

デプロイメントでこのシークレットを使用する場合は、**id_rsa** と **known_hosts** ファイルを、Business Central の Pod にある **/home/jboss/.ssh** ディレクトリーにマウントします。

2.6. NFS を使用した READWRITEMANY アクセスモードの永続ボリュームのプロビジョニング

高可用性 Business Central をデプロイする場合、ご使用の環境は **ReadWriteMany** アクセスモードで永続ボリュームをプロビジョニングする必要があります。高可用性 Business Central をデプロイする場合、ご使用の環境は **ReadWriteMany** アクセスモードで永続ボリュームをプロビジョニングする必要があります。

お使いの設定で **ReadWriteMany** アクセスモードの永続ボリュームのプロビジョニングが必要であるものの、環境がそのようなプロビジョニングに対応しない場合は、NFS を使用してボリュームをプロビジョニングします。それ以外の場合、この手順は省略します。

手順

NFS サーバーをデプロイし、NFS を使用して永続ボリュームをプロビジョニングします。NFS を使用して永続ボリュームをプロビジョニングする方法の詳細は、 OpenShift Container Platform Storage ガイドの "Persistent storage using NFS" のセクションを参照してください。

2.7. S2I ビルドに使用する BUSINESS CENTRAL からのソースコードのデプロイメント

Source-to-Image (S2I) プロセスを使用してイミュータブル KIE Server を作成する予定がある場合は、Git リポジトリーにサービスのソースコードを提供する必要があります。オーサリングサービスに Business Central を使用する場合は、サービスのソースコードをデプロイメントして、S2I ビルドを使用する別の Git リポジトリー (GitHub や GitLab のオンプレミスインストールなど) に配置できます。

S2I プロセスを使用する予定がない場合や、サービスのオーサリングに Business Central を使用していない場合は、この手順を飛ばして次に進んでください。

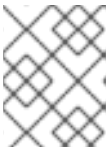
手順

1. 以下のコマンドを使用してソースコードをデプロイメントします。

```
git clone https://<business-central-host>:443/git/<MySpace>/<MyProject>
```

このコマンドでは、以下の変数を置き換えてください。

- **<business-central-host>**: Business Central を実行しているホスト
- **<MySpace>**: プロジェクトが配置された Business Central 領域の名前
- **<MyProject>**: プロジェクトの名前



注記

Business Central でプロジェクトの完全な URL を表示するには、**Menu → Design → <MyProject> → Settings** の順にクリックします。

注記

HTTPS 通信に自己署名証明書を使用している場合にこのコマンドを実行すると、エラーメッセージ **SSL certificate problem** が表示され失敗する可能性があります。このような場合は、**GIT_SSL_NO_VERIFY** 環境変数を使用するなど、**git** で SSL 証明書の検証を無効にします。

```
env GIT_SSL_NO_VERIFY=true git clone https://<business-central-host>:443/git/<MySpace>/<MyProject>
```

2. S2I ビルドの別の Git リポジトリー (GitHub または GitLab など) へのソースコードのアップロード

2.8. ネットワークが制限された環境でのデプロイメントの準備

公開インターネットに接続されていないネットワークが制限された環境に Red Hat Decision Manager をデプロイできます。ネットワークが制限された環境での Operator のデプロイメント方法については、Red Hat OpenShift Container Platform ドキュメントの Using Operator Lifecycle Manager on restricted networks を参照してください。

重要

Red Hat Decision Manager 7.13 では、制限されたネットワークへのデプロイメントはテクノロジープレビュー機能となっています。Red Hat のテクノロジープレビュー機能のサポートの詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

公開インターネットへの送信アクセスが設定されていないデプロイメントを使用するには、必要なすべてのアーティファクトのミラーが含まれる Maven リポジトリーを用意する必要もあります。このリポジトリーを作成する方法は、「オフラインで使用する Maven ミラーリポジトリーの用意」 を参照してください。

2.9. オフラインで使用する MAVEN ミラーリポジトリーの用意

Red Hat OpenShift Container Platform 環境に公開インターネットへの送信アクセスが設定されていない場合には、必要なアーティファクトすべてのミラーが含まれる Maven リポジトリーを用意して、このリポジトリーを使用できるようにする必要があります。

注記

Red Hat OpenShift Container Platform 環境がインターネットに接続されている場合は、この手順を飛ばして次に進むことができます。

前提条件

- 公開インターネットへの送信アクセスが設定されているコンピューターが利用できる。

手順

1. 書き込みアクセス権がある Maven リリースリポジトリーを設定します。リポジトリーは認証なしで読み取りアクセスを許可する必要があり、OpenShift 環境にはこのリポジトリーへのネットワークアクセスが必要です。
   OpenShift 環境に、Nexus リポジトリーマネージャーをデプロイできます。OpenShift への Nexus の設定方法は、Red Hat OpenShift Container Platform 3.11 ドキュメントの Nexus の設定 を参照してください。記載の手順は、OpenShift Container Platform 4 にも該当します。

   このリポジトリーをミラーとして使用し、公開されている Maven アーティファクトをホストします。イミュータブルなサーバーにこれらのサービスをデプロイするため、このリポジトリーで独自のサービスを提供することもできます。

2. 公開インターネットに送信アクセスができるコンピューターで、以下のアクションを実行します。

3. Red Hat カスタマーポータルの Software Downloads ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。

   - **製品:** Process Automation Manager
   - **バージョン:** 7.13.4

     a. **Red Hat Process Automation Manager 7.13.4 Offliner Content List**(**rhpam-7.13.4-offliner.zip**) の製品配信可能ファイルをダウンロードして展開します。

     b. **rhpam-7.13.4-offliner.zip** ファイルの内容を任意のディレクトリーに展開します。

     c. ディレクトリーに移動し、以下のコマンドを入力します。

```
./offline-repo-builder.sh offliner.txt
```

このコマンドは、**repository** サブディレクトリーを作成し、必要なアーティファクトをこのサブディレクトリーにダウンロードします。これはミラーリポジトリーです。

一部のダウンロードが失敗したことを示すメッセージが表示された場合は、同じコマンドを再度実行してください。ダウンロードが再び失敗する場合は、Red Hat サポートに連絡してください。

d. **repository** サブディレクトリーのすべてのアーティファクトを、作成した Maven ミラーリポジトリーにアップロードします。アーティファクトをアップロードするには、Git リポジトリー Maven リポジトリー tools から利用できる Maven リポジトリー Provisioner ユーティリティーを使用できます。

4. Business Central 外でサービスを開発し、追加の依存関係がある場合は、ミラーリポジトリーにその依存関係を追加します。サービスを Maven プロジェクトとして開発した場合は、以下の手順を使用し、これらの依存関係を自動的に用意します。公開インターネットへに送信接続できるコンピューターで、この手順を実行します。

a. ローカルの Maven キャッシュディレクトリー (~/**.m2/repository**) のバックアップを作成して、ディレクトリーを削除します。

b. **mvn clean install** コマンドを使用してプロジェクトのソースをビルドします。

c. すべてのプロジェクトで以下のコマンドを入力し、Maven を使用してプロジェクトで生成したすべてのアーティファクトのランタイムの依存関係をすべてダウンロードするようにします。

```
mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -
Djava.net.preferIPv4Stack=true
```

/path/to/project/pom.xml を、プロジェクトの **pom.xml** ファイルのパスに置き換えます。

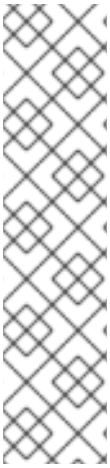
d. ローカルの Maven キャッシュディレクトリー (~/**.m2/repository**) から作成した Maven ミラーリポジトリーにすべてのアーティファクトをアップロードします。アーティファクトをアップロードするには、Git リポジトリー Maven リポジトリー tools から利用できる Maven リポジトリー Provisioner ユーティリティーを使用できます。

第3章 OPENSHIFT OPERATOR を使用した RED HAT DECISION MANAGER 環境のデプロイおよび管理

OpenShift Operator は、環境を記述する YAML ソースを使用して Red Hat Decision Manager 環境をデプロイします。Red Hat Decision Manager は、YAML ソースの作成、環境のデプロイに使用できるインストーラーを提供します。

Business Automation Operator で環境をデプロイする場合は、環境の YAML 記述を作成し、環境が常にこの記述と一致していることを確認します。記述を編集して環境を変更することができます。

Red Hat OpenShift Container Platform で Operator アプリケーションを削除することで、環境を削除できます。



注記

高可用性の Business Central で環境を削除すると、Operator は、JBoss Datagrid および JBoss AMQ StatefulSet の生成時に作成された永続ボリューム要求 (PVC) を削除しません。この動作は、Kubernetes 設計の一部で、永続ボリューム要求を削除するとデータが損失される可能性があります。StatefulSet の削除時における永続ボリューム要求の取り扱いは、Kubernetes documentation を参照してください。

同じ namespace とアプリケーション名を使用する新規環境を構築すると、その環境ではパフォーマンスが向上されるように永続ボリュームを再利用します。

新規デプロイメントで古いデータを使用しないようにするには、Persistent Volume Claim を手動で削除します。

3.1. BUSINESS AUTOMATION OPERATOR のサブスクライブ

Operator を使用して Red Hat Decision Manager をデプロイできるようにするには、OpenShift のビジネス自動化のオペレーターにサブスクリプション登録する必要があります。

手順

1. OpenShift Web クラスターコンソールでプロジェクトに移動します。
2. OpenShift Web コンソールのナビゲーションパネルで、**Catalog → OperatorHub** または **Operators → OperatorHub** を選択します。
3. **Business Automation** を検索し、これを選択してから **Install** をクリックします。
4. **Create Operator Subscription** ページで、ターゲットの名前空間および承認ストラテジーを選択します。
   必要に応じて、**承認ストラテジー** を **Automatic** に設定して、Operator の自動更新を有効にします。Operator の更新は直ちに製品を更新しませんが、製品を更新する前に必要になります。特定のすべての製品デプロイメントの設定を使用して、自動または手動の製品更新を設定します。
5. **Subscribe** をクリックしてサブスクリプションを作成します。

3.2. OPERATOR を使用した RED HAT DECISION MANAGER 環境のデプロイ

Business Automation Operator にサブスクライブした後に、インストーラーウィザードを使用して Red Hat Decision Manager 環境を設定し、デプロイできます。

重要

Red Hat Decision Manager 7.13 では、Operator インストーラーウィザードはテクノロジープレビュー機能となっています。Red Hat のテクノロジープレビュー機能の詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

3.2.1. Business Automation Operator の使用による Red Hat Decision Manager 環境のデプロイメントの開始

Business Automation Operator を使用して Red Hat Decision Manager 環境のデプロイメントを開始するには、インストーラーウィザードにアクセスします。インストーラーウィザードは Operator にサブスクライブするとデプロイされます。

前提条件

- Business Automation Operator にサブスクライブしている。Operator にサブスクライブする方法は、「Business Automation Operator のサブスクライブ」 を参照してください。

手順

1. Red Hat OpenShift Container Platform Web クラスターコンソールメニューで、**Catalog → Installed operators** または **Operators → Installed operators** を選択します。
2. **businessautomation** が含まれる Operator の名前をクリックします。この Operator の情報が表示されます。
3. ウィンドウの右側にある **Installer** リンクをクリックします。
4. プロンプトが出されたら、OpenShift 認証情報でログインします。

結果

ウィザードの **Installation** タブが表示されます。

3.2.2. 環境の基本設定の設定

Business Automation Operator を使用して Red Hat Decision Manager 環境のデプロイを開始した後に、環境のタイプを選択し、他の基本的な設定を行う必要があります。

前提条件

- 「Business Automation Operator の使用による Red Hat Decision Manager 環境のデプロイメントの開始」 の説明に従って、Business Automation Operator を使用して Red Hat Decision Manager 環境のデプロイを開始し、インストーラーウィザードにアクセスしている。

手順

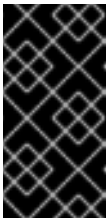
1. **Application Name** フィールドに、OpenShift アプリケーションの名前を入力します。この名前は、すべてのコンポーネントのデフォルト URL で使用されます。

2. **Environment** リストで、環境のタイプを選択します。このタイプは、デフォルトの設定を定めるものです。この設定を必要に応じて変更することができます。以下のタイプは Red Hat Decision Manager で利用できます。

   - **rhdm-trial**: すばやく設定して、アセットの開発や実行を評価またはデモで確認するのに使用できる試用版の環境。Business Central と KIE Server 1 台が含まれています。この環境では永続ストレージを使用しないため、この環境で実行した作業内容は保存されません。

   - **rhdm-authoring**: Business Central を使用してサービスを作成し、変更する環境。これは、オーサリング作業用に Business Central を提供する Pod およびサービスのテスト実行用に KIE Server を提供する Pod で設定されます。この環境を使用して、ステージングおよび実稼働の目的でサービスを実行することも可能です。環境に KIE Server を追加して、同じ Business Central で管理できます。

   - **rhdm-authoring-ha**: Business Central を使用してサービスを作成し、変更する環境。これは、オーサリング作業用に Business Central を提供する Pod およびサービスのテスト実行用に KIE Server を提供する Pod で設定されます。このバージョンのオーサリング環境は、高可用性が確保されるように Business Central Pod のスケーリングをサポートします。

     重要

     Red Hat Decision Manager 7.13 では、高可用性 Business Central 機能はテクノロジープレビュー機能となっています。Red Hat Technology Preview 機能の詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

   - **rhpam-production-immutable**: ステージングおよび実稼働の目的で既存のサービスを実行するための別の環境。ソースからサービスをビルドしたり、Maven リポジトリーからサービスをプルする KIE Server Pod を 1 つ以上設定できます。その後、必要に応じて各 Pod を複製できます。
     Pod からサービスを削除したり、新しいサービスを Pod に追加したりすることはできません。サービスの別のバージョンを使用するか、他の方法で設定を変更する場合は、新規のサーバーイメージをデプロイして、以前のイメージを置き換えます。コンテナーベースの統合ワークフローを使用して、Pod を管理できます。

     この環境を設定する場合は、**KIE Servers** タブで KIE Server をカスタマイズし、**Set immutable server configuration** ボタンをクリックするか、**KIE_SERVER_CONTAINER_DEPLOYMENT** 環境変数を設定します。KIE Server の設定手順は、「環境のカスタム KIE Server 設定の設定」 を参照してください。

3. 新しいバージョンへの自動アップグレードを有効にするには、**Enable Upgrades** ボックスを選択します。このボックスを選択すると、Red Hat Decision Manager 7.13 の新しいパッチバージョンが利用可能になると、Operator は自動的にこのバージョンにデプロイメントをアップグレードします。サービスはすべて確保され、アップグレードプロセス全体で通常通り利用できます。
   Red Hat Decision Manager 7.x の新規マイナーバージョンが利用できる場合にも、同じ自動アップグレードプロセスを有効にする場合は、**Include minor version upgrade** のチェックボックスを選択します。

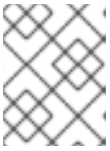
   

   注記

   Red Hat Decision Manager のコンポーネントにカスタムイメージを使用する場合は、自動更新を無効にします。

4. イメージのダウンロードにイメージタグを使用する場合は、**Use Image Tags** ボックスを選択します。この設定は、カスタムレジストリーを使用する場合や、Red Hat サポートがダイレクトされる場合に役立ちます。

5. デプロイメントへの SSL 接続を無効にする場合は、**Disable SSL Routes** ボックスを選択します。この場合、外部に公開されるすべてのルートはクリアテキスト (HTTP) 接続を使用します。

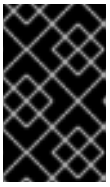


注記

このボックスを選択しないと、セキュアな (HTTPS) ルートのみが外部に公開されます。

6. **Custom registry** のカスタムイメージレジストリーを使用する場合は、**Image registry** フィールドにレジストリーの URL を入力します。このレジストリーに適切に署名され、認識された SSL 証明書がない場合は、**Insecure** ボックスを選択します。
特定のイメージを使用するためのイメージレジストリーの設定方法については、「特定のイメージを使用するためのイメージレジストリーの設定」 を参照してください。

7. **Admin user** の下の、**Username** フィードおよび **Password** フィールドに、Red Hat Decision Manager の管理者ユーザーのユーザー名とパスワードを入力します。



重要

RH-SSO または LDAP 認証を使用する場合は、Red Hat Decision Manager の **kie-server,rest-all,admin** ロールを使用して、認証システムで同じユーザーを設定する必要があります。

8. オプション: 起動ストラテジーを選択します。**OpenShiftStartupStrategy** 設定はデフォルトで有効にされます。
一部のオーサリング環境では、複数のユーザーが同じ KIE Server に同時にサービスをデプロイできることを確認する必要があります。デフォルトでは、Business Central を使用して KIE Server にサービスをデプロイした後に、ユーザーは数秒待ってから追加のサービスをデプロイする必要があります。**OpenShiftStartupStrategy** 設定はデフォルトで有効になり、この制限が発生します。制限を削除するには、**Startup Strategy** リストから **ControllerBasedStartupStrategy** 設定を選択します。

注記

高可用性の Business Central を使用する環境でコントローラーベースのストラテジーを有効にしないでください。

9. オプション: HTTPS 通信のトラストストアとして OpenShift CA バンドルを使用する場合は、**Use OpenShift CA Bundle** ボックスを選択します。

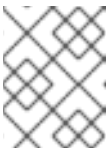
10. オプション: Admin ユーザーの認証情報を含むシークレットを使用する場合は、次のタスクを実行します。

    a. **Admin user configuration** リストから、**Secret configuration** を選択します。

    b. **OpenShift admin credentials secret** の **Secret** フィールドに、シークレットの名前を入力します。**Secret** フィールドを空白にした場合、**kie-admin-credentials** のデフォルトのシークレットが使用されます。

c. **OpenShift admin credentials secret** の **Username** フィールドに、シークレットで使用する管理者ユーザーのユーザー名を入力します。**Username** フィールドを空白にした場合、**kie-admin-credentials** のデフォルトのユーザー名が使用されます。

d. **OpenShift admin credentials secret** の **Password** フィールドに、シークレットで使用する管理者ユーザーのパスワードを入力します。**Password** フィールドを空白にすると、**kie-admin-credentials** のデフォルトパスワードが使用されます。



注記

kie-admin-credentials がない場合、デフォルトのユーザー名とパスワードで **kie-admin-credentials** が生成されます。

次のステップ

デフォルト設定で環境をデプロイする必要がある場合は、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合は、引き続き他の設定パラメーターの設定を行います。

3.2.2.1. 特定のイメージを使用するためのイメージレジストリーの設定

環境の基本設定中に、特定のイメージを使用するようにカスタムレジストリーを設定できます。基本的な環境設定の詳細については、「環境の基本設定の設定」 を参照してください。

前提条件

- 「環境の基本設定の設定」 の説明に従って、Business Automation Operator を使用して Red Hat Decision Manager 環境の設定を開始し、インストーラーウィザードにアクセスしている。
- 「環境の基本設定の設定」 の手順に従って、Red Hat Decision Manager 環境設定の **カスタムレジストリー** の下にある **イメージレジストリーフィールドにイメージレジストリー** の URL を設定しました。

手順

- Red Hat Decision Manager 環境の設定中に、カスタムレジストリーから特定のイメージを使用するには、以下のいずれかの手順を実行します。
  - インストーラウィザードを使用してイメージを指定する場合は、**Console** タブおよび **KIE Server** タブでイメージコンテキスト、イメージ名、イメージタグの各パラメーターを設定します。

表3.1 パラメーター

名前	説明
Image context	レジストリー内のイメージのコンテキスト。
Image	イメージの名前。
Image tag	イメージのタグ。このフィールドを設定しない場合、インストールは最新のタグを使用します。

- KieApp CR の YAML ファイルを使用してイメージを指定する場合は、イメージレジストリーとイメージの詳細をファイルに追加してください。以下に例を示します。

例

```
apiVersion: app.kiegroup.org/v2
kind: KieApp
spec:
 ...
 useImageTags: true
 imageRegistry:
  registry: registry.example.com:5000
  ...
 objects:
  ...
  servers:
   - id:  ...
     ...
     image: YOUR_IMAGE_NAME
     imageContext: YOUR_IMAGE_CONTEXT
     imageTag: YOUR_IMAGE_TAG
     ...
```

注記

設定された名前を持つ ImageStream が使用している名前空間または Red Hat OpenShift Container Platform の名前空間に存在する場合は、Operator はこの ImageStream を使用し、新しい ImageStream を作成しません。

ImageStream を自動更新するように設定するには、KieApp CR の YAML ファイルで、ImageStream プロパティー **scheduledImportPolicy** を **true** に設定する必要があります。以下に例を示します。

scheduledImportPolicy の例

```
apiVersion: app.kiegroup.org/v2
kind: KieApp
spec:
  ...
  useImageTags: true
  scheduledImportPolicy: true
  imageRegistry:
   registry: registry.example.com:5000
  ...
  objects:
   ...
   servers:
    - id:  ...
      ...
      image: YOUR_IMAGE_NAME
      imageContext: YOUR_IMAGE_CONTEXT
      imageTag: YOUR_IMAGE_TAG
      ...
```

3.2.3. 環境のセキュリティー設定の設定

Business Automation Operator を使用して Red Hat Decision Manager 環境の基本的な設定を行った後に、必要に応じて環境の認証 (セキュリティー) 設定を実行できます。

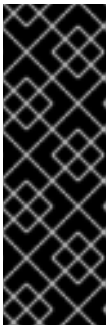
前提条件

- 「環境の基本設定の設定」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Decision Manager 環境の基本設定を行っている。
- 認証に RH-SSO または LDAP を使用する必要がある場合には、認証システムに適切なロールを持つユーザーを作成していること。**kie-server,rest-all,admin** ロールを持つ少なくとも 1 人の管理ユーザー (たとえば、**adminUser**) を作成する必要があります。このユーザーには、**Installation** タブで設定したユーザー名とパスワードが必要です。
- RH-SSO 認証を使用する必要がある場合は、環境のすべてのコンポーネントの RH-SSO システムでクライアントを作成しており、正しい URL を指定している。この動作により、最大限の制御が確保されます。他の方法として、デプロイメントでクライアントを作成できます。

手順

1. **Installation** タブが開いている場合は、**Next** をクリックして **Security** タブを表示します。
2. **Authentication mode** リストで、以下のモードのいずれかを選択します。
   - **Internal**: 環境のデプロイ時に初期ユーザーを設定します。設定後スクリプトを作成して、Elytron security サブシステムにユーザーを追加できます。設定後スクリプトの作成方法は、「Elytron ユーザー設定やその他の設定の指定」 を参照してください。
   - **RH-SSO**: Red Hat Decision Manager は認証に Red Hat Single Sign-On を使用します。
   - **LDAP**: Red Hat Decision Manager は認証に LDAP を使用します。
3. 選択した **Authentication mode** に基づいてセキュリティー設定を完了します。
   RH-SSO を選択してい場合は、RH-SSO 認証を設定します。
   a. **RH-SSO URL** フィールドに、RH-SSO URL を入力します。
   b. **Realm** フィールドに、RH-SSO レルム名を入力します。
   c. 環境のコンポーネントに RH-SSO クライアントを作成していない場合は、**SSO admin user** フィールドおよび **SSO admin password** フィールドに、RH-SSO システムの管理者ユーザーの認証情報を入力します。
   d. RH-SSO システムに適切な署名済みの SSL 証明書がない場合は、**Disable SSL cert validation** ボックスを選択します。
   e. **Principal attribute** フィールドで、ユーザー名に使用される RH-SSO プリンシパル属性を変更する必要がある場合は、新規属性の名前を入力します。

   LDAP を選択した場合は、LDAP 認証を設定します。



重要

baseFilter フィールドは、認証するユーザーのコンテキストを見つけるために使用される従来の LDAP 検索フィルターです。ログインモジュールコールバックから取得した入力 **username** または **userDN** は、**{0}** 式が使用されるいずれの場所でもフィルターに置き換えられます。検索フィルターの一般的な例は **(uid={0})** です。Elytron ベースのサブシステムの場合には、このプロパティーは、検索式を使用せずに検索フィルターパラメーターだけで設定する必要があります。たとえば、**(uid={0})** の代わりに **uid** を検索します。

a. **LDAP URL** フィールドに、LDAP URL を入力します。

b. LDAP パラメーターを設定します。これらのパラメーターは、Red Hat JBoss EAP の Elytron サブシステムを使用して LDAP 認証を設定します。Red Hat JBoss EAP の Elytron サブシステムを LDAP と共に使用する方法は、Configure Authentication with an LDAP-Based Identity Store を参照してください。

注記

LDAP フェイルオーバーを有効にする場合は、**AUTH_LDAP_URL** パラメーターに、2 つ以上の LDAP サーバーアドレスをスペースで区切って設定できます。

4. **RH-SSO** または **LDAP** を選択した場合や、RH-SSO システムまたは LDAP システムがデプロイメントに必要なすべてのロールを定義していない場合は、認証システムのロールを Red Hat Decision Manager のロールにマップできます。
ロールマッピングを有効にするには、単一の設定文字列またはロールマッピング設定ファイルとしてロールマッピングを指定する必要があります。ロールマッピング設定にファイルを使用する場合は、ファイルをプロジェクト namespace の OpenShift 設定マップまたはシークレットオブジェクトに指定する必要があります。

文字列は **role=role1,role2;another-role=role2** パターンを使用する必要があります。例えば、**admins=kie-server,rest-all,admin;developers=kie-server,rest-all** のようになります。

ファイルには、次の形式のエントリーが含まれている必要があります。

```
ldap_role=product_role1, product_role2...
```

以下に例を示します。

```
admins=kie-server,rest-all,admin
```

この文字列またはファイルを使用するには、以下の変更を加えます。

a. **RoleMapper**の下の**Roles properties file**フィールドに、ロール設定文字列、またはロールマッピング設定ファイルの完全修飾パス名 (例 **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**) を入力します。

b. オプション: **Roles keep mapped**ボックスまたは **Roles keep non mapped**ボックスを選択します。ロールマッピングを定義する場合、デフォルトではマッピングで定義するロールのみが利用できます。認証システムで定義された元のロールを保持し、マッピングが他の

ロールにマップする場合は、**Roles keep mapped** ボックスを選択します。認証システムで定義されていて、マッピングに記載されていないオリジナルのロールを保持する場合は、**Roles keep non mapped** ボックスを選択します。

c. ロール設定ファイルを使用している場合は、**RoleMapper Configuration object** の下のフィールドを設定します。

- **Kind** ラベルで、ファイルを提供するオブジェクトの kind (**ConfigMap** または **Secret**) を選択します。
- **Name** フィールドにオブジェクトの名前を入力します。このオブジェクトは、ロールマッピング設定ファイルに指定したパスで Business Central および KIE Server Pod に自動的にマウントされます。

5. 他のパスワードを設定します (必要な場合)。

- **AMQ password** および **AMQ cluster password** は、JMS API を使用した ActiveMQ との対話に使用するパスワードです。
- **Keystore password** は、HTTPS 通信のシークレットで使用されるキーストアファイルのパスワードです。「KIE Server のシークレットの作成」 または 「Business Central へのシークレットの作成」 の説明にしたがってシークレットを作成した場合は、このパスワードを設定します。
- **Database password** は、環境の一部であるデータベースサーバー Pod のパスワードです。

次のステップ

すべてのコンポーネントのデフォルト設定で環境をデプロイする必要がある場合は、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合は、引き続き Business Central および KIE Server の設定パラメーターを設定します。

3.2.4. 環境の Business Central 設定の設定

Business Automation Operator を使用して Red Hat Decision Manager 環境の基本的なセキュリティー設定を行ってから、環境の Business Central コンポーネントの設定を任意で実行することができます。

rhdm-production-immutable 以外のすべての環境タイプには、このコンポーネントが含まれます。

rhpam-production-immutable 環境には Business Central または Business Central Monitoring が含まれていないため、この環境の設定は変更しないでください。

前提条件

- 「環境の基本設定の設定」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Decision Manager 環境の基本設定を行っている。
- 認証に RH-SSO または LDAP を使用する必要がある場合は、「環境のセキュリティー設定の設定」 の説明に従ってセキュリティー設定を完了している。

手順

1. **Installation** または **Security** タブが開いている場合は、**Console** タブが表示されるまで **Next** をクリックします。
2. 「Business Central へのシークレットの作成」 の説明に従って Business Central のシークレットを作成している場合は、**Keystore secret** フィールドにシークレットの名前を入力します。

3. オプション: Business Central のデプロイメントにカスタムイメージを使用する場合は、次の追加手順を実行します。

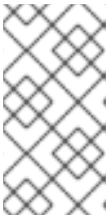
   a. **Installation** タブでカスタムレジストリーを設定します。カスタムレジストリーを設定しない場合、インストールはデフォルトの Red Hat レジストリーを使用します。カスタムレジストリー値の設定に関する詳細は、「環境の基本設定の設定」 を参照してください。

   b. **Console** タブで、以下のフィールドを設定します。

      - **Image context**: レジストリー内のイメージのコンテキスト。
      - **Image**: イメージの名前。
      - **Image tag**: イメージのタグ。このフィールドを設定しない場合、インストールは **latest** タグを使用します。
        たとえば、イメージの完全なアドレスが **registry.example.com/mycontext/mycentral:1.0-SNAPSHOT** の場合、カスタムレジストリーを **registry.example.com** に、**Image context** フィールドを **mycontext** に、**Image** フィールドを **mycentral** に、そして **Image tag** フィールドを **1.0-SNAPSHOT** に設定します。

4. オプション: 外部ルートにカスタムホスト名を設定するには、**Custom hostname to be used on the Business Central external Route** フィールドに、次の例のような書式でドメインを入力します。

   ```
   `businesscentral.example.com`
   ```

   注記

   カスタムホスト名は有効で解決可能である必要があります。

   カスタムホスト名を変更するには、**routeHostname** プロパティーを変更します。

5. オプション: Edge の終端ルートを有効にして設定するには、次の手順を実行します。

   a. **Change route termination** で **Enable Edge termination** を選択します。

   b. オプション: **Key** フィールドに、秘密鍵を入力します。

   c. オプション: **Certificate** フィールドに、証明書を入力します。

   d. オプション: **CaCertificate** フィールドに、CaCertificate を入力します。

6. 必要に応じて、Git フックを設定します。
   オーサリング環境では、Git フックを使用して、Business Central の内部 Git リポジトリーと外部 Git リポジトリー間の操作を容易化できます。Git フックを使用する場合は、プロジェクト namespace の OpenShift 設定マップ、シークレット、または Persistent Volume Claim (PVC: 永続ボリューム要求) オブジェクトに Git フックディレクトリーを準備する必要があります。Git の SSH 認証用の SSH キーと既知のホストファイルでシークレットを作成することもできます。Git フックの作成に関する詳細は、「Git フックの準備」 を参照してください。

   Git フックディレクトリーを使用するには、以下の変更を加えます。

   a. **Mount path** フィールドの **GitHooks** の下に、ディレクトリーの完全修飾名を入力します (例: **/opt/kie/data/git/hooks**)。

b. **GitHooks Configuration object** の下のフィールドで、ファイルを提供するオブジェクトの Kind (**ConfigMap**、**Secret**、または **PersistentVolumeClaim**) を選択し、オブジェクトの **Name** を入力します。このオブジェクトは、Git フックディレクトリーの指定したパスで Business Central Pod に自動的にマウントされます。

7. 必要に応じて、**SSH secret** フィールドに、SSH キーと既知のホストファイルを含む、シークレットを入力します。

8. オプション: jBPM およびケース管理機能が無効になるようにデシジョン管理のみの機能用に KIE サーバーを設定するには、**Execute Kie Server only with Decisions capabilities**を選択します。

9. 必要に応じて、Business Central または Business Central monitoring のレプリカ数を **Replicas** フィールドに入力します。この数は **rhpam-authoring** 環境では変更しません。

10. 必要に応じて、**Console コンポーネント** ページで Business Central 永続ボリュームサイズ **pvSize** を設定するには、**Persistent Volume Size** フィールドに必要なサイズを入力します。デフォルトのサイズは、Business Central の場合は 1Gi、Business Central Monitoring の場合は 64Mb です。

11. 必要に応じて、**Resource quotas** 下のフィールドに必要な CPU およびメモリーの上限値を入力します。

12. Business Central Pod の Java 仮想マシンの設定をカスタマイズする必要がある場合は、**Enable JVM configuration** ボックスを選択してから、**Enable JVM configuration** の下のフィールドに情報を入力します。すべてのフィールドは任意です。設定可能な JVM パラメーターについては、「JVM 設定パラメーター」 を参照してください。

13. RH-SSO 認証を選択している場合は、Business Central の RH-SSO を設定します。

    a. **Client name** フィールドにクライアント名を入力し、**Client secret** フィールドにクライアントシークレットを入力します。この名前を持つクライアントが存在しない場合は、デプロイメントでこの名前およびシークレットを持つ新規クライアントの作成を試行します。

    b. デプロイメントで新規クライアントを作成する場合は、Business Central へのアクセスに使用する HTTP および HTTPS URL を **SSO HTTP URL** フィールドおよび **SSO HTTPS URL** フィールドに入力します。この情報は、クライアントに記録されます。

14. オプション: 高可用性環境を設定している場合、DataGrid コンポーネントのユーザー名とパスワードを**DataGrid username**と**DataGrid password**フィールドに設定します。デフォルトでは、ユーザー名は **infinispan** で、パスワードは自動的に生成されます。

15. 必要に応じて、環境変数を随時設定します。環境変数を設定するには、**Add new Environment variable** をクリックしてから、変数の名前および値を **Name** フィールドおよび **Value** フィールドに入力します。

    - オプション: プロキシー設定を行う場合は、以下の環境変数を使用します。

      - **https_proxy**: https プロキシーの場所。これは、**HTTPS_PROXY**、**http_proxy**、および **HTTP_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **myuser:mypass@127.0.0.1:8080**

      - **HTTPS_PROXY**: https プロキシーの場所。これは **http_proxy** および **HTTP_PROXY** に優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **myuser@127.0.0.1:8080**

      - **http_proxy**: http プロキシーの場所。これは **HTTP_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **http://127.0.0.1:8080**

- **HTTP_PROXY**: http プロキシーの場所。これは、Maven ビルドと Java ランタイムの両方に使用されます。例: **127.0.0.1:8080**
- **no_proxy**: 直接アクセスできるホスト、IP アドレス、またはドメインのコンマ区切りリスト。これは **NO_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: ***.example.com**
- **NO_PROXY**: 直接アクセスできるホスト、IP アドレス、またはドメインのコンマ区切りリスト。これは、Maven ビルドと Java ランタイムの両方に使用されます。例: **foo.example.com,bar.example.com**

- 外部 Maven リポジトリーを使用する必要がある場合は、以下の変数を設定します。
  - **MAVEN_REPO_URL**: Maven リポジトリーの URL
  - **MAVEN_REPO_ID**: Maven リポジトリーの ID (例: **repo-custom**)
  - **MAVEN_REPO_USERNAME**: Maven リポジトリーのユーザー名
  - **MAVEN_REPO_PASSWORD** Maven リポジトリーのパスワード

重要

オーサリング環境で、Business Central を使用して外部の Maven リポジトリーにプロジェクトをプッシュする場合は、デプロイメント時にこのリポジトリーを設定して、全プロジェクトのリポジトリーへのエクスポートを設定する必要があります。外部の Maven リポジトリーへの Business Central プロジェクトのエクスポートに関する情報は、**Red Hat Decision Manager プロジェクトのパッケージ化およびデプロイ** を参照してください。

- OpenShift 環境が公開インターネットに接続されていない場合は、「オフラインで使用する Maven ミラーリポジトリーの用意」 に従って設定した Maven ミラーにアクセスできるように設定します。以下の変数を設定してください。
  - **MAVEN_MIRROR_URL**: 「オフラインで使用する Maven ミラーリポジトリーの用意」でセットアップした Maven ミラーリポジトリーの URL。この URL は、OpenShift 環境の Pod からアクセスできるようにする必要があります。
  - **MAVEN_MIRROR_OF**: ミラーから取得されるアーティファクトを定める値。**mirrorOf** 値の設定方法は、Apache Maven ドキュメントの Mirror Settings を参照してください。デフォルト値は **external:*** です。この値の場合、Maven はミラーから必要なアーティファクトをすべて取得し、他のリポジトリーにクエリーを送信しません。
    外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合は、ミラーからこのリポジトリー内のアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

    オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合は、ミラーからこのリポジトリーのアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhpamcentr**)。

- 場合によっては、Business Central の Maven リポジトリーキャッシュの永続化が必要です。デフォルトでは、キャッシュは永続化されないため、Business Central Pod を再起動またはスケーリングすると、すべての Maven アーティファクトが再度ダウンロードされ、Business Central 内のすべてのプロジェクトが再度ビルドされる必要があります。キャッ

シュの永続性を有効にした場合は、ダウンロードは必要なく、状況によっては起動にかかる時間が改善される可能性があります。ただし、Business Central 永続ボリュームには、大きな追加領域が必要です。
Maven リポジトリーキャッシュの永続性を有効にするには、**KIE_PERSIST_MAVEN_REPO** 環境変数を **true** に設定します。

KIE_PERSIST_MAVEN_REPO を **true** に設定した場合には、オプションで **KIE_M2_REPO_DIR** 変数を使用してキャッシュのカスタムパスを設定できます。デフォルトのパスは **/opt/kie/data/m2** です。**/opt/kie/data** ディレクトリーツリー内のファイルは永続化されます。

次のステップ

KIE Server のデフォルト設定で環境をデプロイする必要がある場合は、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合は、引き続き KIE Server の設定パラメーターを設定します。

3.2.5. 環境のカスタム KIE Server 設定の設定

Business Automation Operator のすべての環境タイプには、デフォルトで 1 つまたは複数の KIE Server が含まれます。

必要に応じて、KIE Server のカスタム設定を設定できます。この場合、デフォルトの KIE Server は作成されず、設定する KIE Server のみがデプロイされます。

前提条件

- 「環境の基本設定の設定」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Decision Manager 環境の基本設定を行っている。

手順

1. **Installation**、**Security**、または **Console** タブが開いている場合は、**KIE Servers** タブが表示されるまで **Next** をクリックします。
2. **Add new KIE Server** をクリックして、新規の KIE Server 設定を追加します。
3. **Id** フィールドに、この KIE Server インスタンスの ID を入力します。KIE Server インスタンスが Business Central または Business Central Monitoring インスタンスに接続される場合、この ID はサーバーが加わるサーバーグループを決めるものとなります。
4. **Name** フィールドに、KIE Server の名前を入力します。
5. **Deployments** フィールドに、デプロイする同様の KIE Server の数を入力します。インストーラーは、同じ設定で複数の KIE Server をデプロイできます。KIE Server の ID および名前は自動的に変更され、一意な状態に保たれます。
6. 「KIE Server のシークレットの作成」 の説明に従って KIE Server のシークレットを作成している場合は、**Keystore secret** フィールドにシークレットの名前を入力します。
7. オプション: jBPM およびケース管理機能が無効になるようにデシジョン管理のみの機能用に KIE サーバーを設定するには、**Execute Kie Server only with Decisions capabilities**を選択します。
8. オプション: KIE Server のレプリカ数を **Replicas** フィールドに入力します。

9. オプション: 外部ルートにカスタムホスト名を設定するには、**Custom hostname to be used on the KIE Server external Route** フィールドに、次の例のような書式でドメインを入力します。

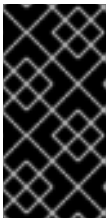
```
`kieserver.example.com`
```

注記

カスタムホスト名は有効で解決可能である必要があります。

カスタムホスト名を変更するには、**routeHostname** プロパティーを変更します。

10. オプション: Edge の終端ルートを有効にして設定するには、次の手順を実行します。
    a. **Change route termination** で **Enable Edge termination** を選択します。
    b. オプション: **Key** フィールドに、秘密鍵を入力します。
    c. オプション: **Certificate** フィールドに、証明書を入力します。
    d. オプション: **CaCertificate** フィールドに、CaCertificate を入力します。
11. オプション: カスタムの KIE Server イメージを使用する場合は、以下の追加手順を実行します。
    a. **Set KIE Server image**をクリックします。
    b. OpenShift イメージストリームからイメージをプルする場合は **Kind** リストから **ImageStreamTag** を選択し、Docker レジストリーからイメージをプルする場合は **DockerImage** を選択します。
    c. 次のいずれかの手順を実行して、イメージ名を設定します。
        - **ImageStreamTag** の kind を選択した場合は、**Name** フィールドにイメージのイメージストリームタグ名を入力します (例: **my-custom-is-tag:1.0**)。
          対応するイメージストリームがご使用の環境に存在しない場合、Operator はデフォルトの Red Hat レジストリーとタグを使用して、このイメージストリームを作成します。**Installation** タブでカスタムレジストリーを設定した場合、Operator はこのレジストリーを使用してイメージストリームを作成します。
        - **DockerImage** の kind を選択した場合は、イメージの完全修飾イメージ名を **Name** フィールドに入力します (例: **registry.io/test/testing:1.0**)。
          ご使用の環境がアクセスできる任意のレジストリーからイメージを設定できます。
    d. **openshift** 名前空間にないイメージストリームを使用する場合は、**Namespace** フィールドに名前空間を入力します。
       カスタムイメージの作成手順については、「KIE Server のカスタムイメージの作成」 を参照してください。
12. オプション: Source to Image (S2I) ビルドを使用して、イミュータブル KIE Server を設定する必要がある場合は、以下の追加の手順を実行します。



重要

Maven リポジトリーからサービスをプルするイミュータブル KIE Server を設定する必要がある場合は、**Set Immutable server configuration** をクリックせず、この手順も実行しないでください。代わりに、**KIE_SERVER_CONTAINER_DEPLOYMENT** 環境変数を設定します。

a. **Set Immutable server configuration** をクリックします。

b. **KIE Server コンテナーデプロイメント** フィールドに、デプロイメントが Source to Image (S2I) ビルドの結果からデプロイメントする必要があるサービスの識別情報 (KJAR ファイル) を入力します。形式は **<containerId>=<groupId>:<artifactId>:<version>** になります。また、コンテナーのエイリアス名で指定する場合には、形式は **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>** になります。以下の例に示されるように、区切り文字 | を使用して 2 つ以上の KJAR ファイルを指定できます (例: **containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2**)。

c. OpenShift 環境に公開インターネットへの接続がない場合は、「オフラインで使用する Maven ミラーリポジトリーの用意」 に従って、**Maven mirror URL** フィールドに設定した Maven ミラーの URL を入力します。

d. **Artifact directory** フィールドで、Maven が正常にビルドされた後に、必要なバイナリーファイル (KJAR ファイルおよびその他の必要なファイル) が含まれるプロジェクト内のパスを入力します。通常、このディレクトリーはビルドのターゲットディレクトリーです。ただし、Git リポジトリーのこのディレクトリーにビルド済みのバイナリーを提供できます。

e. S2I ビルドにカスタムベース KIE Server イメージを使用する必要がある場合は、**Set Base build image** をクリックして、**Name** フィールドにイメージストリームの名前を入力します。イメージストリームが **openshift** 名前空間にない場合は、名前空間を **Namespace** フィールドに入力します。OpenShift イメージストリームタグではなく Docker イメージ名を使用する必要がある場合は、**Kind** の値を **DockerImage** に変更します。

f. **Set Git source** をクリックし、以下のフィールドに情報を入力します。

- **S2I Git URI** サービスのソースが含まれる Git リポジトリーの URI。
- **Reference**: Git リポジトリーのブランチ。
- **コンテキストディレクトリー**: (オプション) Git リポジトリーからダウンロードされたプロジェクト内のソースへのパス。デフォルトで、ダウンロードされたプロジェクトのルートディレクトリーはソースディレクトリーです。

注記

Git ソースを設定しない場合、イミュータブルな KIE Server は S2I ビルドを使用しません。その代わりに、設定済みの Maven リポジトリーから **KIE Server コンテナーデプロイメント** フィールドで定義したアーティファクトをプルします。

g. S2I を使用し、Git Webhook を設定して Git リポジトリーの変更が KIE Server の自動リビルドをトリガーするように設定する必要がある場合は、**Add new Webhook** をクリックします。次に、**Type** フィールドで Webhook のタイプを選択し、**Secret** フィールドで Webhook のシークレット文字列を入力します。

h. S2I ビルドのビルド環境変数を設定するには、**Add new Build Config Environment variable** をクリックしてから、変数の名前および値を **Name** フィールドおよび **Value** フィールドに入力します。

13. 必要に応じて、**Resource quotas** 下のフィールドに必要な CPU およびメモリーの上限値を入力します。複数の KIE Server を設定している場合は、制限値はそれぞれのサーバーに別々に適用されます。

14. RH-SSO 認証を選択している場合は、KIE Server の RH-SSO を設定します。

    a. **Client name** フィールドにクライアント名を入力し、**Client secret** フィールドにクライアントシークレットを入力します。この名前を持つクライアントが存在しない場合は、デプロイメントでこの名前およびシークレットを持つ新規クライアントの作成を試行します。

    b. デプロイメントで新規クライアントを作成する場合は、KIE Server インスタンスへのアクセスに使用する HTTP および HTTPS URL を **SSO HTTP URL** フィールドおよび **SSO HTTPS URL** フィールドに入力します。この情報は、クライアントに記録されます。

15. 外部 AMQ メッセージブローカーを使用して JMS API から KIE Server と対話する場合は、**Enable JMS Integration** 設定を有効にします。JMS 統合を設定するための追加のフィールドが表示され、必要に応じて値を入力する必要があります。

    - **User name**、**Password**: ブローカーのユーザー認証が環境で必要な場合の、標準ブローカーユーザーのユーザー名およびパスワード。
    - **Executor**: この設定を選択して JMS executor を無効にします。Executor はデフォルトで有効になります。
    - **Executor transacted**: この設定を選択して、Executor キューで JMS トランザクションを有効にします。
    - **Enable signal**: この設定を選択して JMS 経由でシグナルの設定を有効にします。
    - **Enable audit** この設定を選択して JMS 経由で監査ロギングを有効にします。
    - **Audit transacted**: この設定を選択して、監査キューで JMS トランザクションを有効にします。
    - **Queue executor**、**Queue request**、**Queue response**、**Queue signal**、**Queue audit** 使用するキューのカスタム JNDI 名。これらの値のいずれかを設定する場合は、**AMQ キュー** パラメーターも設定する必要があります。
    - **AMQ Queues**: AMQ キュー名はコンマで区切られます。これらのキューはブローカーの起動時に自動的に作成され、JBoss EAP サーバーの JNDI リソースとしてアクセスできます。カスタムキュー名を使用する場合は、このフィールドでサーバーが使用するすべてのキューの名前を入力する必要があります。
    - **Enable SSL integration**: AMQ ブローカーへの SSL 接続を使用する場合は、この設定を選択します。この場合は、「AMQ ブローカー接続のシークレットの作成」 で作成したシークレットの名前や、シークレットに使用したキーストアおよび信頼ストアの名前およびパスワードも指定する必要があります。

16. KIE Server Pod の Java 仮想マシンの設定をカスタマイズする必要がある場合は、**Enable JVM configuration** ボックスを選択してから、**Enable JVM configuration** の下のフィールドに情報を入力します。すべてのフィールドは任意です。設定可能な JVM パラメーターについては、「JVM 設定パラメーター」 を参照してください。

17. オプション: プロキシー設定を行う場合は、以下の環境変数を使用します。

- **https_proxy**: https プロキシーの場所。これは、**HTTPS_PROXY**、**http_proxy**、および **HTTP_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **myuser:mypass@127.0.0.1:8080**
- **HTTPS_PROXY**: https プロキシーの場所。これは **http_proxy** および **HTTP_PROXY** に優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **myuser@127.0.0.1:8080**
- **http_proxy**: http プロキシーの場所。これは **HTTP_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: **http://127.0.0.1:8080**
- **HTTP_PROXY**: http プロキシーの場所。これは、Maven ビルドと Java ランタイムの両方に使用されます。例: **127.0.0.1:8080**
- **no_proxy**: 直接アクセスできるホスト、IP アドレス、またはドメインのコンマ区切りリスト。これは **NO_PROXY** よりも優先され、Maven ビルドと Java ランタイムの両方に使用されます。例: ***.example.com**
- **NO_PROXY**: 直接アクセスできるホスト、IP アドレス、またはドメインのコンマ区切りリスト。これは、Maven ビルドと Java ランタイムの両方に使用されます。例: **foo.example.com,bar.example.com**

18. 必要に応じて、環境変数を随時設定します。環境変数を設定するには、**Add new Environment variable** をクリックしてから、変数の名前および値を **Name** フィールドおよび **Value** フィールドに入力します。

- 設定した Maven リポジトリーからサービスをプルするイミュータブル KIE Server を設定する必要がある場合は、以下の設定を入力します。

  i. **KIE_SERVER_CONTAINER_DEPLOYMENT** 環境変数を設定します。変数には、デプロイメントが Maven リポジトリーからプルする必要のあるサービス (KJAR ファイル) の ID 情報が含まれている必要があります。形式は **<containerId>=<groupId>:<artifactId>:<version>** になります。また、コンテナーのエイリアス名で指定する場合には、形式は **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>** になります。以下の例に示されるように、区切り文字 | を使用して 2 つ以上の KJAR ファイルを指定できます (例: **containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2**)。

  ii. 外部 Maven リポジトリーの設定

- 外部 Maven リポジトリーを設定する必要がある場合には、以下の変数を設定します。
  - **MAVEN_REPO_URL**: Maven リポジトリーの URL
  - **MAVEN_REPO_ID**: Maven リポジトリーの ID (例: **repo-custom**)
  - **MAVEN_REPO_USERNAME**: Maven リポジトリーのユーザー名
  - **MAVEN_REPO_PASSWORD**: Maven リポジトリーのパスワード
- OpenShift 環境が公開インターネットに接続されていない場合は、「オフラインで使用する Maven ミラーリポジトリーの用意」 に従って設定した Maven ミラーにアクセスできるように設定します。以下の変数を設定してください。
  - **MAVEN_MIRROR_URL**: 「オフラインで使用する Maven ミラーリポジトリーの用意」でセットアップした Maven ミラーリポジトリーの URL。この URL は、OpenShift 環境の Pod からアクセスできるようにする必要があります。この KIE Server を S2I として設定している場合は、この URL をすでに入力されています。

- **MAVEN_MIRROR_OF**: ミラーから取得されるアーティファクトを定める値。この KIE Server を S2I として設定している場合は、この値を設定しません。**mirrorOf** 値の設定方法は、Apache Maven ドキュメントの Mirror Settings を参照してください。デフォルト値は **external:*** です。この値の場合、Maven はミラーから必要なアーティファクトをすべて取得し、他のリポジトリーにクエリーを送信しません。
  外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合は、ミラーからこのリポジトリー内のアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

  オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合は、ミラーからこのリポジトリーのアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhpamcentr**)。

- Prometheus を使用してメトリックを収集し、保存するように KIE Server デプロイメントを設定する必要がある場合は、**PROMETHEUS_SERVER_EXT_DISABLED** 環境変数を **false** に設定します。Prometheus メトリクス収集の設定手順については、Managing and monitoring KIE Server を参照してください。

- Red Hat Single Sign-On 認証を使用していて、アプリケーションと Red Hat Single Sign-On の相互作用に CORS (cross-origin resource sharing) のサポートが必要な場合は、**CORS フィルター設定**を設定します。

  - デフォルトの設定で CORS を使用するには、**CORS Filters configuration**リストで**Default configuration**が選択されていることを確認し、**Enable CORS with Default values**を選択します。

  - カスタム設定で CORS を使用するには、**CORS Filters configuration**リストから**Custom configuration**を選択し、CORS フィルターに関連する値を入力します。

次のステップ

追加の KIE Server を設定するには、**Add new KIE Server**を再びクリックし、新規サーバー設定の手順を繰り返します。

Finish をクリックしてから **Deploy** をクリックし、環境をデプロイします。

3.3. OPERATOR を使用してデプロイした環境の変更

環境を Operator を使用してデプロイした場合は、通常の OpenShift の手法を使用して環境を変更することはできません。たとえば、デプロイメント設定またはサービスを削除しても、これは同じパラメーターで自動的に再作成されます。

環境を変更するには、環境の YAML の記述を変更する必要があります。パスワードなどの一般的な設定を変更し、KIE Server を追加してスケーリングできます。

手順

1. OpenShift Web クラスターコンソールでプロジェクトに移動します。
2. OpenShift Web コンソールナビゲーションパネルで **Catalog → Installed operators** または **Operators → Installed operators** を選択します。
3. 表で **Business Automation** Operator 行を見つけ、その行で **KieApp** をクリックします。この Operator を使用してデプロイした環境の情報が表示されます。
4. デプロイした環境の名前をクリックします。

5. **YAML** タブを選択します。
YAML ソースが表示されます。YAML ソースの **spec:** でコンテンツを編集して、環境の設定を変更できます。

6. Red Hat Decision Manager のデプロイバージョンを変更する場合は、**spec:** に以下の行を追加します。

```
version: 7.13.4
```

7.13.4 は、必要な別のバージョンに置き換えることができます。カスタムイメージを使用する場合など、自動更新が無効になっている場合は、この設定を使用して Red Hat Decision Manager を新規バージョンにアップグレードしてください。

7. パスワードなどの共通の設定を変更するには、**commonConfig:** の値を編集します。

8. 新しい KIE Server を追加する場合は、以下の例に示されているように、**servers:** のブロックの最後にそれらの記述を追加します。

- 名前が **server-a** と **server-a-2** のサーバー 2 台を追加するには、以下の行を追加します。

```
- deployments: 2
  name: server-a
```

- S2I プロセスのソースからビルドされるサービスを含む、イミュータブルな KIE Server を追加するには、以下の行を追加します。

```
- build:
    kieServerContainerDeployment: <deployment>
    gitSource:
      uri: <url>
      reference: <branch>
      contextDir: <directory>
```

以下の値を置き換えます。

  - **<deployment>**: ソースからビルドしたデシジョンサービス (KJAR ファイル) の識別情報。形式は **<containerId>=<groupId>:<artifactId>:<version>** になります。区切り記号 | を使用して 2 つ以上の KJAR ファイルを指定できます (例: **containerId=groupId:artifactId:version|c2=g2:a2:v2**)。Maven ビルドプロセスは、Git リポジトリーのソースからこのようなファイルをすべて生成する必要があります。

  - **<url>**: デシジョンサービスのソースを含む Git リポジトリーの URL。

  - **<branch>**: Git リポジトリーのブランチ。

  - **<directory>**: Git リポジトリーからダウンロードしたプロジェクトのソースへのパス。

9. KIE Server をスケーリングする場合は、**servers:** のブロックに含まれるサーバーの記述を検索して、その記述の下に **replicas:** 設定を追加します。たとえば、**replicas: 3** はサーバーを Pod 3 つにスケーリングします。

10. 他に変更を加える場合は、利用可能な設定の CRD ソースを確認します。CRD ソースを表示するには、管理ユーザーで **oc** コマンドを使用して Red Hat OpenShift Container Platform 環境にログインし、以下のコマンドを入力します。

```
oc get crd kieapps.app.kiegroup.org -o yaml
```

11. **Save** をクリックしてから **has been updated** ポップアップメッセージを待機します。
12. **Reload** をクリックして、環境の新しい YAML の記述を表示します。

3.4. ELYTRON ユーザー設定やその他の設定の指定

LDAP または RH-SSO 認証を使用しない場合は、Red Hat Decision Manager は Red Hat JBoss EAP の Elytron サブシステムの内部ユーザーに依存します。デフォルトでは、管理ユーザーのみが作成されます。その他のユーザーを Red Hat JBoss EAP の Elytron security サブシステムに追加する必要がある場合があります。これには、Red Hat JBoss EAP 設定後スクリプトを実行する必要があります。

この設定後スクリプト、またはその他の Red Hat JBoss EAP 設定後スクリプトは、Red Hat OpenShift Container Platform への Red Hat Decision Manager のデプロイメントで設定できます。

手順

1. GitHub リポジトリー からサンプルファイルをダウンロードします。
2. サンプルファイルに基づいて以下のファイルを準備します。
   - **postconfigure.sh**: Red Hat JBoss EAP の実行に必要な設定後シェルスクリプト。この例では、このスクリプトは **add-users.cli** スクリプトを使用して Elytron ユーザーを追加します。CLI スクリプト以外の設定後タスクを実行する場合は、このスクリプトを変更します。
   - **latepostconfigure.sh**: Red Hat Decision Manager バージョン 7.13.4 で必要な空のファイル。
   - **add-users.cli**: Elytron ユーザーまたは他の CLI タスクを設定するための Red Hat JBoss EAP コマンドラインインターフェイススクリプト。以下の行の間にコマンドを追加します。

     ```
     embed-server --std-out=echo --server-config=standalone-openshift.xml batch

       <your jboss-cli commands>

     run-batch quit
     ```

3. **oc** コマンドを使用して Red Hat OpenShift Container Platform クラスターにログインし、デプロイメントの namespace に移動します。
4. 以下のコマンドを使用して、準備したファイルで ConfigMap を作成します。

   ```
   oc create configmap postconfigure \
      --from-file=add-users.cli=add-users.cli \
      --from-file=delayedpostconfigure.sh=delayedpostconfigure.sh \
      --from-file=postconfigure.sh=postconfigure.sh
   ```

5. 以下のコマンドを入力して、**kieconfigs-7.13.4** Config Map を編集します。

   ```
   oc edit cm kieconfigs-7.13.4
   ```

6. このファイルでは、**console:** セクションの配置設定を変更して Business Central に設定を追加し、**servers:** セクションのすべての配置設定を変更して KIE Server インスタンスに設定を追加します。

各デプロイメント設定で、以下の変更を加えます。

- **deploymentConfigs.metadata.spec.template.spec.containers.volumeMounts** の下に以下の行を追加します。

```
- name: postconfigure-mount
  mountPath: /opt/eap/extensions
```

- **deploymentConfigs.metadata.spec.template.spec.containers.volumeMounts** の下に以下の行を追加します。

```
- name: "postconfigure-mount"
  configMap:
    name: "postconfigure"
    defaultMode: 0555
```

7. ファイルを保存します。この時点以降、新規 Operator のデプロイメントには設定後の設定が含まれます。

既存のデプロイメントでは、設定後設定が自動的に追加されない場合には、Business Central および KIE Server Pod を削除できます。Operator は、設定後設定で更新バージョンを自動的に開始します。

3.5. JVM 設定パラメーター

Operator を使用して Red Hat Decision Manager をデプロイする場合は、必要に応じて Business Central および KIE Server の多数の JVM 設定パラメーターを設定できます。これらのパラメーターは、対応するコンテナーの環境変数を設定します。

以下の表では、Operator を使用して Red Hat Decision Manager をデプロイする際に設定できるすべての JVM 設定パラメーターのリストを表示しています。

デフォルト設定は、ほとんどのユースケースに最適です。必要な場合にのみ変更を行ってください。

表3.2 JVM 設定パラメーター

設定フィールド	環境変数	説明	例
Java Opts の追加	JAVA_OPTS_APPEND	JAVA_OPTS で生成されたオプションに追加されるユーザー指定の Java オプション。	**-Dsome.property=foo**
Java 最大メモリー比	JAVA_MAX_MEM_RATIO	Java 仮想マシンに使用できるコンテナーメモリーの最大パーセンテージ。残りのメモリーはオペレーティングシステムに使用されます。デフォルト値は **50** であり、50% の制限があります。**-Xmx** JVM オプションを設定します。**0** の値を入力した場合、**-Xmx** オプションは設定されません。	**40**

設定フィールド	環境変数	説明	例
Java 初期メモリー比	JAVA_INITIAL_MEM_RATIO	Java 仮想マシンに最初に使用されるコンテナーメモリーの割合。デフォルト値は **25** であるため、この値が **Java Max Initial Memory** 値を超えない場合は、Pod メモリーの 25% が最初に JVM に割り当てられます。**-Xms** JVM オプションを設定します。**0** の値を入力すると、**-Xms** オプションは設定されません。	**25**
Java 最大初期メモリー	JAVA_MAX_INITIAL_MEM	Java 仮想マシンで最初に使用できるメモリーの最大量 (メガバイト単位)。**Java initial memory ratio** パラメーターで設定されるように初期の割り当てメモリーがこの値よりも大きい場合、この値で設定されたメモリー量は **-Xms** JVM オプションを使用して割り当てられます。デフォルト値は **4096** です。	**4096**
Java 診断	JAVA_DIAGNOSTICS	この設定を有効にすると、追加の JVM 診断情報の標準出力への出力が有効になります。デフォルトでは無効にされています。	**true**
Java デバッグ	JAVA_DEBUG	この設定を有効にして、リモートデバッグをオンに切り替えます。デフォルトでは無効にされています。JVM オプション **-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=${debug_port}** を追加します。ここで、**${debug_port}** はデフォルトで **5005** に設定されます。	**true**
Java デバッグポート	JAVA_DEBUG_PORT	リモートデバッグに使用されるポート。デフォルト値は **5005** です。	**8787**
GC の最小ヒープ解放比率	GC_MIN_HEAP_FREE_RATIO	拡張を回避するためのガベージコレクション (GC) 後のヒープ解放の最小パーセンテージ。JVM オプション **-XX:MinHeapFreeRatio** を設定します。	**20**
GC の最大ヒープ解放比率	GC_MAX_HEAP_FREE_RATIO	縮小を回避するための GC 後のヒープ解放の最大パーセンテージ。JVM オプション **-XX:MaxHeapFreeRatio** を設定します。	**40**

設定フィールド	環境変数	説明	例
GC 時間比率	GC_TIME_RATIO	ガベージコレクションに費やした時間に対する、ガベージコレクション外で費やした時間 (アプリケーションの実行に費やした時間など) の比率を指定します。JVM オプション **-XX:GCTimeRatio** を設定します。	**4**
GC 適応サイズポリシーの重み	GC_ADAPTIVE_SIZE_POLICY_WEIGHT	以前の GC 時間に対する現在の GC 時間の重み付け。JVM オプション **-XX:AdaptiveSizePolicyWeight** を設定します。	**90**
GC の最大メタスペースサイズ	GC_MAX_METASPACE_SIZE	メタスペースの最大サイズ。JVM オプション **-XX:MaxMetaspaceSize** を設定します。	**100**

3.6. KIE 設定と CONFIGMAPS

Red Hat Decision Manager Operator をインストールすると、Operator は、現在の名前空間に対して **kieconfig-$VERSION** という接頭辞が付いた YAML ファイルである ConfigMaps を作成します。ConfigMap は、その名前空間の Red Hat Decision Manager コンポーネントの **DeploymentConfigs**、シークレット、ルート、およびサービスなどの機能の設定を含む YAML ファイルです。

ConfigMap の例

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mysql.yaml: |
    ## KIE Databases BEGIN
    databases:
      ## RANGE BEGINS
      #[[ range $index, $Map := .Databases ]]
…
```

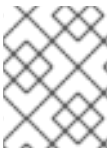
Operator は、ConfigMaps を使用してコンポーネントを設定およびデプロイします。これには、KIE Server、Smart Router、Business Central などのサポートされているすべての Red Hat Decision Manager コンポーネントと、永続ボリューム、ビルド設定、ルートなどのサービス関連の設定が含まれます。ConfigMap を手動で編集する場合、Operator は、環境が調整されるときに新しい値を使用してデプロイメントを作成します。

Red Hat Decision Manager Operator は、現在のバージョンと以前のバージョンの Red Hat Decision Manager コンポーネントを同時に使用でき、バージョンごとに ConfigMaps (7.13.0 と 7.12.1 など) を使用できます。

kieconfigs-7.13.4

これには、**common.yaml** 設定ファイルが含まれています。詳細は、GitHub の **common.yaml** を参照してください。この設定ファイルを使用して、次のコンポーネントを設定できます。

- サーバーオブジェクトである KIE Server は、**## KIE Servers BEGIN** プレースホルダーで識別されます。
- コンソールオブジェクトである BusinessCentral と Business Central Monitoring は、**common.yaml** の最初の行で定義されています。
- smartrouter オブジェクトである Smartrouter は、**## KIE smartrouter BEGIN** プレースホルダーによって識別されます。



注記

kieconfigs-7.13.4 は、これら 3 つのコンポーネントに関連するルートとサービスも保持します。

kieconfigs-7.13.4-dashbuilder

これには、Dashbuilder コンポーネントの設定 YAML ファイルが含まれています。
詳細は、GitHub の **rhpam-standalone-dashbuilder.yaml** を参照してください。

kieconfigs-7.13.4-dbs

これには、MySQL および PostgreSQL データベースの基本的な **DeploymentConfig** が含まれています。
MySQL 設定の詳細は、GitHub の **mysql.yaml** を参照してください。

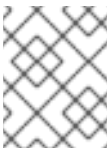
PostgreSQL 設定の詳細は、GitHub の **postgresql.yaml** を参照してください。

kieconfigs-7.13.4-dbs-pim

これには、プロセスインスタンス移行 (PIM) がサポートするデータベース (external、MySQL、PostgreSQL) 用のスニペット設定が含まれています。
PIM 外部設定の詳細は、GitHub の **external.yaml** を参照してください。

PIM MySQL 設定の詳細は、GitHub の **mysql.yaml** を参照してください。

PIM PostgreSQL 設定の詳細は、GitHub の **postgresql.yaml** を参照してください。



注記

これらの YAML ファイルは、この configMap の **application.properties** を使用して設定された PIM データベースの特定の設定のみを保持します。

kieconfigs-7.13.4-dbs-servers

これには、サポートされているデータベース設定 (external、h2、MySQL、PostgreSQL) のスニペット設定が含まれています。
外部設定の詳細は、GitHub の **external.yaml** を参照してください。

h2 設定の詳細は、GitHub の h2 **.yaml を** 参照してください。h2 設定は、実稼働環境ではサポートされていないことに注意してください。

MySQL 設定の詳細は、GitHub の **mysql.yaml** を参照してください。

PostgreSQL 設定の詳細は、GitHub の **postgresql.yaml** を参照してください。

kieconfigs-7.13.4-envs

これには、オーサリング環境やトライアル環境などの各 Red Hat Decision Manager 環境の特定の設定が含まれています。この ConfigMap には、次の YAML ファイルが含まれています。

- rhdm-authoring-ha.yaml
- rhdm-authoring.yaml
- rhdm-production-immutable.yaml
- rhdm-trial.yaml
- rhpam-authoring-ha.yaml
- rhpam-authoring.yaml
- rhpam-production-immutable.yaml
- rhpam-production.yaml
- rhpam-standalone-dashbuilder.yaml
- rhpam-trial.yaml
  各 Red Hat Decision Manager 環境の特定の設定の詳細については、GitHub の Red Hat Decision Manager 環境の ConfigMaps を 参照してください。

kieconfigs-7.13.4-jms

これには、JMS Executor が有効な場合の KIE Server の ActiveMQ 設定が含まれています。JMS Executor 設定の詳細は、GitHub の **activemq-jms-config** を参照してください。

kieconfigs-7.13.4-pim

これには、プロセスインスタンス移行 (PIM) DeploymentConfig と関連する PIM 設定が含まれています。MySQL またはその他のデータベースを PIM で使用している場合は、**kieconfigs-7.13.4-dbs-pim** configMap を使用し、**mysql.yaml** ファイルを編集する必要があります。

3.6.1. ConfigMaps の使用

ConfigMap を使用して Red Hat Decision Manager Operator をカスタマイズし、関連する設定を適用できます。ConfigMaps に変更を加えるには、**oc** コマンドツールまたは Red Hat OpenShift Container Platform コンソールを使用します。

前提条件

- 現在の名前空間に Operator がインストールされている。
- KieApp が利用できる。
- **kieconfig-$VERSION-*** が利用可能です。
  kieconfig-$VERSION-* が利用可能かどうかを確認するには、次のコマンドを実行します。

```
$ oc get cm | grep kieconfigs
```

oc get cm | grep kieconfigs の出力例。

```
kieconfigs-7.13.0              1       64m
kieconfigs-7.13.0-dashbuilder  1       64m
kieconfigs-7.13.0-dbs          2       64m
kieconfigs-7.13.0-dbs-pim      3       64m
kieconfigs-7.13.0-dbs-servers  4       64m
kieconfigs-7.13.0-envs         10      64m
kieconfigs-7.13.0-jms          1       64m
kieconfigs-7.13.0-pim          1       64m
kieconfigs-7.12.1              1       64m
kieconfigs-7.12.1-dashbuilder  1       64m
kieconfigs-7.12.1-dbs          2       64m
kieconfigs-7.12.1-dbs-pim      3       64m
kieconfigs-7.12.1-dbs-servers  4       64m
kieconfigs-7.12.1-envs         10      64m
kieconfigs-7.12.1-jms          1       64m
kieconfigs-7.12.1-pim          1       64m
```

手順

1. Business Central と KIE Server 用にレプリカが 1 つある **rhpam-authoring environment** を作成します。

   rhpam-authoring の例:

```
apiVersion: app.kiegroup.org/v2
kind: KieApp
metadata:
  name: rhpam-authoring
  annotations:
    consoleName: rhpam-authoring
    consoleTitle: PAM Authoring
    consoleDesc: Deploys a PAM Authoring environment
spec:
  environment: rhpam-authoring
objects:
  servers:
    - replicas: 1
  console:
    replicas: 1
```

2. 次のいずれかの手順を実行します。

   - **oc** ツールで特定の ConfigMap を開くには、以下のコマンドを実行します。

```
$ oc edit cm/<CONFIGMAP_NAME>
```

注記

oc ツールは、テキスト編集ツールの **vim** に似ています。

   - Red Hat OpenShift Container Platform コンソールを使用して特定の ConfigMap を開くには、**ConfigMaps** ページで **kieconfigs-7.13.4** に移動し、その YAML バージョンを開いて編集します。

3. YAML ファイルを変更するには、変更内容を含む **annotations** フィールドを追加します。以下に例を示します。
Console.deploymentConfigs.metadata に以下を追加します。

```
annotations:
  my.custom.annotation/v1: v1-rhpam-app-console
```

KIE Server を更新するには、**## KIE Servers Start** プレースホルダー識別子に以下を追加します。

```
annotations:
  my.custom.annotation/v1: v1-rhpam-app-kieserver
```

注記

また、この ConfigMap を使用して、Smart Router の設定を編集することもできます。

4. 変更を保存するには、次のいずれかの手順を実行します。
   - **oc** ツールエディターで、変更を保存して終了するには、**:wq!** と入力します。
   - Red Hat OpenShift Container Platform コンソールで、Red Hat OpenShift Container Platform コンソールを使用して変更を保存する場合は、**Save** をクリックします。
5. 環境が実行されていて、Operator が変更されたコンポーネントの再デプロイメントを自動的に開始しなかった場合は、以下のステップを実行して、**oc** コマンドツールを使用してターゲットコンポーネントの DeploymentConfig を手動で削除する必要があります。
   a. DeploymentConfig を返すには、以下のコマンドを実行する。

```
$ oc get dc
```

oc get dc の出力が返されます。

oc get dc の出力例です。

```
$ oc get dc
NAME                        REVISION   DESIRED   CURRENT   TRIGGERED BY
rhpam-authoring-kieserver           1         1         1         config
rhpam-authoring-rhpamcentr          1         1         1         config
```

   b. 対象コンポーネントの DeploymentConfig を削除する場合は、以下のコマンドを実行します。

```
$ oc delete dc/rhpam-authoring-kieserver
```

```
$ oc delete dc/rhpam-authoring-rhpamcentr
```

デプロイメントは、ConfigMap で適用された変更内容で再デプロイされます。

6. KIE Server DeploymentConfig のアノテーションを確認することで、変更が適用されたことを確認するには、次のコマンドを実行します。

```
$ oc describe dc/rhpam-authoring-kieserver
```

oc describe dc/rhpam-authoring-kieserver の出力例です。

```
Name:  rhpam-authoring-kieserver
Namespace: examplenamespace
Created: 15 minutes ago
Labels:  app=rhpam-authoring
  application=rhpam-authoring
  service=rhpam-authoring-kieserver
  services.server.kie.org/kie-server-id=rhpam-authoring-kieserver
Annotations: my.custom.annotation/v1=v1-rhpam-app-kieserver
```

3.7. KIE SERVER のカスタムイメージの作成

カスタムイメージを作成して、KIE Server のデプロイメントにファイルを追加できます。次に、イメージを独自のコンテナーレジストリーにプッシュする必要があります。Red Hat Decision Manager をデプロイする場合は、カスタムイメージを使用するように Operator を設定できます。

カスタムイメージを使用する場合は、自動のバージョンアップグレードを無効にする必要があります。新規バージョンをインストールする場合は、以前と同じ名前と、新規バージョンタグを指定してイメージをビルドし、レジストリーにそのイメージをプッシュします。その後にバージョンを変更すると、Operator が自動的に新規イメージをプルします。Operator での製品バージョンの変更に関する説明は、「Operator を使用してデプロイした環境の変更」 を参照してください。

特に、次のタイプのカスタムイメージを作成できます。

- 追加の RPM パッケージを含めた KIE Server のカスタムイメージ
- 追加の JAR クラスライブラリーを含めた KIE Server のカスタムイメージ

3.7.1. 追加の RPM パッケージを含めたカスタムの KIE Server イメージの作成

追加の RPM パッケージのインストール先のカスタム KIE Server イメージを作成できます。このイメージをカスタムレジストリーにプッシュして、KIE Server のデプロイに使用できます。

Red Hat Enterprise Linux 8 リポジトリーから任意のパッケージをインストールできます。以下の例では、**ps** ユーティリティーが含まれる **procps-ng** パッケージをインストールしていますが、変更して他のパッケージをインストールすることができます。

手順

1. **podman login** コマンドを使用して **registry.redhat.io** レジストリーの認証を行います。レジストリーの認証に関する詳細は、Red Hat コンテナーレジストリーの認証 を参照してください。
2. サポートされている KIE Server のベースイメージをダウンロードするには、次のコマンドを入力します。

```
podman pull registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.13.4
```

3. ベースイメージをもとにカスタムイメージを定義する **Dockerfile** を作成します。このファイルで、現在のユーザーを **root** に変更して、**yum** コマンドで RPM パッケージをインストールしてから **USER 185** (Red Hat JBoss EAP ユーザー) に戻します。以下の例では、 **Dockerfile** ファイルの内容を示します。

```
FROM registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.13.4
USER root
RUN yum -y install procps-ng
USER 185
```

必要に応じて RPM ファイルの名前を置き換えます。**yum** コマンドは自動的に Red Hat Enterprise Linux 8 リポジトリーからの全依存関係を自動的にインストールします。複数の RMP ファイルをインストールする必要がある場合があります。今回は、**RUN** コマンドを複数回使用します。

4. **Dockerfile** を使用してカスタムイメージをビルドします。レジストリー名など、イメージの完全修飾名を指定します。ベースイメージと同じバージョンタグを使用する必要があります。イメージをビルドするには、以下のコマンドを入力します。

```
podman build . --tag registry_address/image_name:7.13.4
```

以下に例を示します。

```
podman build . --tag registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
```

5. ビルドが完了したら、イメージを実行してログインし、カスタマイズが成功したことを確認します。以下のコマンドを入力します。

```
podman run -it --rm registry_address/image_name:7.13.4 /bin/bash
```

以下に例を示します。

```
podman run -it --rm registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4 /bin/bash
```

イメージのシェルプロンプトで、コマンドを入力して RPM がインストールされていることをテストし、**exit** と入力します。たとえば、**procps-ng** の場合は **ps** コマンドを実行します。

```
[jboss@c2fab36b778e ~]$ ps
PID TTY          TIME CMD
  1 pts/0    00:00:00 bash
 13 pts/0    00:00:00 ps
[jboss@c2fab36b778e ~]$ exit
```

6. カスタムイメージをレジストリーにプッシュするには、次のコマンドを入力します。

```
podman push registry_address/image_name:7.13.4
docker://registry_address/image_name:7.13.4
```

以下に例を示します。

```
podman push registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
docker://registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
```

次のステップ

KIE Server をデプロイする場合は、イメージ名と namespace を設定してレジストリーにカスタムイメージを指定します。**Set KIE Server image**をクリックして、**Kind** の値を **DockerImage** に変更してから、バージョンタグがないレジストリー名など、イメージ名を指定します。以下に例を示します。

```
registry.example.com/custom/rhpam-kieserver-rhel8
```

Operator を使用した KIE Server のデプロイに関する詳細は、「環境のカスタム KIE Server 設定の設定」 を参照してください。

3.7.2. 追加の JAR ファイルを使用したカスタム KIE Server イメージの作成

追加の JAR ファイル (単数、複数問わず) のインストール先のカスタムの KIE Server イメージを作成してサーバーの機能を拡張できます。このイメージをカスタムレジストリーにプッシュして、KIE Server のデプロイに使用できます。

たとえば、カスタムクラス JAR を作成して、カスタムの Prometheus メトリックを KIE Server に提供できます。カスタムクラスの作成手順については、**Managing and monitoring KIE Server**の Extending Prometheus metrics monitoring in KIE Server with custom metrics を参照してください。

手順

1. KIE Server で動作するカスタムライブラリーを開発します。以下のドキュメントと例を使用して、ライブラリーを開発できます。
   - **Managing and monitoring KIE Server**の KIE Server capabilities and extensions
   - Domain-specific Prometheus metrics with Red Hat Process Automation Manager and Decision Manager
   - Extend KIE Server with additional transport
2. JAR ファイルが **target** ディレクトリーに配置されるように Maven を使用してライブラリーをビルドします。この例では、**custom-kieserver-ext-1.0.0.Final.jar** のファイル名を使用します。
3. **podman login** コマンドを使用して **registry.redhat.io** レジストリーの認証を行います。レジストリーの認証に関する詳細は、Red Hat コンテナーレジストリーの認証 を参照してください。
4. サポートされている KIE Server のベースイメージをダウンロードするには、次のコマンドを入力します。

   ```
   podman pull registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.13.4
   ```

5. ベースイメージをもとにカスタムイメージを定義する **Dockerfile** を作成します。このファイルは JAR ファイル (単数、複数を問わず) を **/opt/eap/standalone/deployments/ROOT.war/WEB-INF/lib/** ディレクトリーにコピーする必要があります。以下の例では、**Dockerfile** ファイルの内容を示します。

   ```
   FROM registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.13.4
   COPY target/custom-kieserver-ext-1.0.0.Final.jar
   /opt/eap/standalone/deployments/ROOT.war/WEB-INF/lib/
   ```

6. **Dockerfile** を使用してカスタムイメージをビルドします。レジストリー名など、イメージの完全修飾名を指定します。ベースイメージと同じバージョンタグを使用する必要があります。イメージをビルドするには、以下のコマンドを入力します。

   ```
   podman build . --tag registry_address/image_name:7.13.4
   ```

   以下に例を示します。

```
podman build . --tag registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
```

7. カスタムイメージをレジストリーにプッシュするには、次のコマンドを入力します。

```
podman push registry_address/image_name:7.13.4
docker://registry_address/image_name:7.13.4
```

以下に例を示します。

```
podman push registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
docker://registry.example.com/custom/rhpam-kieserver-rhel8:7.13.4
```

次のステップ

KIE Server をデプロイする場合は、イメージ名と namespace を設定してレジストリーにカスタムイメージを指定します。**Set KIE Server image**をクリックして、**Kind** の値を **DockerImage** に変更してから、バージョンタグがないレジストリー名など、イメージ名を指定します。以下に例を示します。

```
registry.example.com/custom/rhpam-kieserver-rhel8
```

Operator を使用した KIE Server のデプロイに関する詳細は、「環境のカスタム KIE Server 設定の設定」 を参照してください。

第4章 RED HAT OPENSHIFT CONTAINER PLATFORM 3 のデプロイメントからの情報の移行

以前に Red Hat OpenShift Container Platform 3 で Red Hat Decision Manager デプロイメントを使用していた場合は、バージョン 3 のデプロイメントから Red Hat OpenShift Container Platform 4 の新しいデプロイメントに情報を移行できます。

情報を移行する前に、Operator を使用して、新しい Red Hat Decision Manager インフラストラクチャーを Red Hat OpenShift Container Platform 4 にデプロイする必要があります。以前のインフラストラクチャーのデプロイメントに存在する要素を、新しいデプロイメントにも追加します。以下に例を示します。

- 既存のオーサリングデプロイメントの場合は、Business Central と最低でも KIE Server 1 台を含めて新しいオーサリングインフラストラクチャーを作成します。
- 既存のイミュータブル KIE Server の場合は、同じアーティファクトで新しいイミュータブル KIE Server をデプロイします。

4.1. BUSINESS CENTRAL での情報の移行

Red Hat OpenShift Container Platform 3 に、既存のオーサリング環境がある場合は、この環境の Business Central から **.niogit** リポジトリーと Maven リポジトリーを Red Hat OpenShift Container Platform 4 の新規デプロイメントにある Business Central にコピーします。このアクションで、新しいデプロイメントにすべて同じプロジェクトとアーティファクトが作成されます。

前提条件

- Red Hat OpenShift Container Platform 3 および Red Hat OpenShift Container Platform 4 のインフラストラクチャーの両方に、ネットワークでアクセスできるマシンが必要です。
- 対象のマシンに Red Hat OpenShift Container Platform 4 からの **oc** コマンドラインクライアントをインストールしておく必要があります。コマンドラインクライアントのインストール方法については、Red Hat OpenShift Container Platform ドキュメントの CLI tools を参照してください。

手順

1. Business Central や KIE Server など、以前のデプロイメントや新しいデプロイメントの要素に接続されている Web クライアントやクライアントアプリケーションがないことを確認します。
2. 空の一時ディレクトリーを作成して、そのディレクトリーに移動します。
3. **oc** コマンドを使用して、Red Hat OpenShift Container Platform 3 インフラストラクチャーにログインし、以前のデプロイメントが含まれるプロジェクトに切り替えます。
4. 以前のデプロイメントにある Pod 名を表示するには、以下のコマンドを実行します。

   ```
   oc get pods
   ```

   Business Central の Pod を検索します。この Pod の名前には **rhpamcentr** が含まれます。高可用性のデプロイメントでは、Business Central Pod はどれでも使用できます。
5. 以下の例のように、**oc** コマンドを使用して、**.niogit** リポジトリーと Maven リポジトリーを Pod からローカルマシンにコピーします。

```
oc cp myapp-rhpamcentr-5-689mw:/opt/kie/data/.niogit .niogit
oc cp myapp-rhpamcentr-5-689mw:/opt/kie/data/maven-repository maven-repository
```

6. **oc** コマンドを使用して、Red Hat OpenShift Container Platform 4 インフラストラクチャーにログインし、新しいデプロイメントが含まれるプロジェクトに切り替えます。

7. 新しいデプロイメントにある Pod 名を表示するには、以下のコマンドを実行します。

```
oc get pods
```

Business Central の Pod を検索します。この Pod の名前には **rhpamcentr** が含まれます。高可用性のデプロイメントでは、Business Central Pod はどれでも使用できます。

8. 以下の例のように、**oc** コマンドを使用して、**.niogit** リポジトリーと Maven リポジトリーをローカルマシンから Pod にコピーします。

```
oc cp .niogit myappnew-rhpamcentr-abd24:/opt/kie/data/.niogit
oc cp maven-repository myappnew-rhpamcentr-abd24:/opt/kie/data/maven-repository
```

付録A バージョン情報

本書の最終更新日: 2023 年 9 月 5 日 (火)

付録B お問い合わせ先

Red Hat Decision Manager ドキュメントチーム: brms-docs@redhat.com