Red Hat Decision Manager 7.9

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

ガイド

Last Updated: 2022-11-20

Red Hat Decision Manager 7.9 Red Hat OpenShift Container Platform への Red Hat Decision Manager のデプロイメント

ガイド

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書では、オーソリング環境、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

前書き

開発者またはシステム管理者は、オーソリング環境、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、**弊社の CTO である Chris Wright のメッセージ**を参照してください。

パート I. OPERATOR を使用した RED HAT OPENSHIFT CONTAINER PLATFORM 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 バージョン 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 バージョン 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 Process Automation 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。
  - 高可用性オーソ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。
- 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 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 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 のクラスターリングと複数の 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 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 の間にレイヤーを提供します。独立した KIE Server 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 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 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。

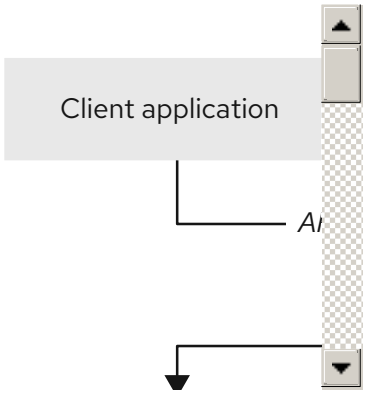
KIE Server デプロイメント (スケーリング可能) ごとに、現在の状態を反映する ConfigMap を作成します。Business Central は、ConfigMap を読み込むことで全 KIE Server を検出します。

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、全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、独自の再デプロイをトリガーします。

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 シークレットの概要 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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 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 **decisioncentral-app-secret** を生成します。

```
$ oc create secret generic decisioncentral-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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 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

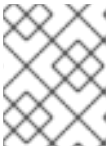
オーサリング環境では、Business Central のプロジェクトのソースコードが変更された場合に Git フックを使用してカスタムの操作を実行できます。Git フックは一般的に、アップストリームのリポジトリーを操作する時に使用します。

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

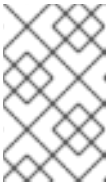
Business Central では **pre-commit** スクリプトはサポートされません。**post-commit** スクリプトを使用してください。

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。
   - 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説明は、Red Hat OpenShift Container Platform ドキュメントの ストレージ 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は、OpenShift Container Platform ストレージ の NFS を使用した永続ストレージ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展開

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展開して、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。

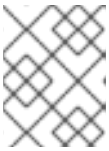
手順

1. 以下のコマンドを使用してソースコードを展開します。

```
git clone https://<decision-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。

- **<decision-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://<decision-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は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

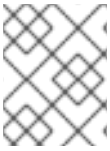
重要

Red Hat Decision Manager 7.9 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 バージョン 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。

   a. **Red Hat Process Automation Manager 7.9.1 Offliner Content List**をクリックして、Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-offliner.zip** をダウンロードします。

   b. **rhdm-7.9.1-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。

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 repository tools から利用できる Maven Repository Provisioner ユーティリティーを使用できます。

3. 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 は、プロジェクトの **pom.xml** ファイルへの正しいパスに置き換えます。

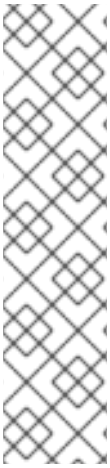
d. ローカルの Maven キャッシュディレクトリー (~/**.m2/repository**) から作成した Maven ミラーリポジトリーにすべてのアーティファクトをアップロードします。アーティファクトをアップロードするには、Git リポジトリー Maven repository tools から利用できる Maven Repository Provisioner ユーティリティーを使用できます。

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 7.9 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

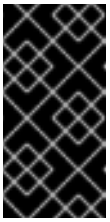
1. **Application Name** フィールドに、OpenShift アプリケーションの名前を入力します。この名前は、すべてのコンポーネントのデフォルト URL で使用されます。

2. **Environment** 一覧で、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。

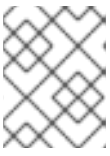
     重要

     Red Hat Decision Manager 7.9 では、Business Central の高可用性機能はテクノロジープレビューとしてのみの提供となっています。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。

   - **rhdm-production-immutable**: ステージングおよび実稼働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 7.9 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。

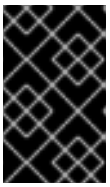
4. オプション: イメージのダウンロードにイメージタグを使用する場合は、**Use Image Tags** ボックスを選択します。この設定は、カスタムレジストリーを使用する場合や、Red Hat サポートがダイレクトされる場合に役立ちます。

5. **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。

注記

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、**Console** および **KIE Server** タブでイメージのコンテキスト、名前、およびタグを設定します。

6. **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。

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。

3.2.3. 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** 一覧で、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。このユーザーは Business Central を使用して他のユーザーを随時セットアップできます。

- **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。

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

   b. Red Hat JBoss EAP の LdapExtended ログインモジュールの設定に対応する LDAP パラメーターを設定します。これらの設定に関する説明は、LdapExtended ログインモジュール 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、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ロールマッピング設定ファイルを指定する必要があります。
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
```

このファイルの使用を有効にするには、以下の変更を行います。

   a. **Roles properties file** フィールドの **RoleMapper** の下に、ルールマッピング設定ファイルの完全修飾パス名を入力します (例: **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**)。

b. 認証システムで定義されているロールをマッピングファイルで定義されているロールに置き換える場合は、**Replace roles** ボックスを選択します。それ以外の場合は、RH-SSO または LDAP で定義されたロールと設定ファイルで定義されたロールの両方が利用可能です。

c. **RoleMapper 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** または **Secret**) 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、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。

rhdm-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. 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。

   c. 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。

5. 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。**rhdm-authoring** 環境では、この数を変更しません。

6. 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。

7. 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。

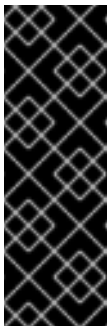
8. RH-SSO 認証を選択している場合は、Business Central の RH-SSO を設定します。

   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、KIE Server 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。

9. 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。

- 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。

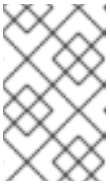
    オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合は、ミラーからこのリポジトリーのアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhdmcentr**)。

- 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。

- 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、ユーザーは、Business Central を使用して KIE Server にサービスをデプロイして数秒待ってから追加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、**コントローラーストラテジー** を使用するように **rhdm-authoring** 環境を設定します。これについての特定のニーズがない限り、この変更を行わないでください。コントローラーストラテジーを有効にする場合は、Business Central と同じ環境内のすべての KIE Server でこの変更を行ってください。



注記

高可用性の Business Central を使用する環境でコントローラーストラテジーを有効にしないでください。この環境では、コントローラーストラテジーは正しく機能しません。

Business Central でコントローラーストラテジーを有効にするには、**KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED** 環境変数を **false** に設定します。

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。
- 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**、**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、KIE Server の ID を入力します。KIE Server 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: KIE Server のレプリカ数を **Replicas** フィールドに入力します。

8. オプション: KIE Server デプロイメントにカスタムイメージを使用する場合は、以下の一連の追加手順のいずれかを完了します。

    a. レジストリーでイメージを指定して Docker イメージを使用する場合:

        i. **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。

        ii. **KIE Server** 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/myserver:1.0-SNAPSHOT** の場合は、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 **myserver** に、そして **Image tag** フィールドを **1.0-SNAPSHOT** に設定します。

    b. 既存の OpenShift イメージストリームのイメージを使用する場合:

        i. **Set KIE Server image**をクリックします。

        ii. イメージストリームタグの名前を **Name** フィールドに入力します。

        iii. 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。
        イメージストリームタグが OpenShift 環境ですでに設定されている場合、インストールではこのタグが使用されます。タグが設定されていない場合は、インストールにより、デフォルトのイメージ名およびタグを使用してイメージストリームタグが作成されます。

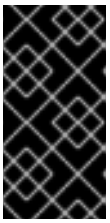
        

        注記

        Kind 値を **DockerImage** に変更しないでください。このオプションは、Red Hat Decision Manager 7.9.1 では機能しません。

        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。

9. Source to Image (S2I) ビルドを使用して、イミュータフル 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_REPLOYMENT** 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) ビルドの結果から展開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 URl** サービスのソースが含まれる Git リポジトリーの URl。
- **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。

10. 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。

11. 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 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。

12. 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。

13. 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。

14. 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。

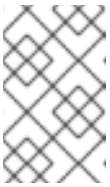
    オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合は、ミラーからこのリポジトリーのアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhdmcentr**)。

- Prometheus を使用してメトリクスを収集し、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 メトリクス収集の方法は、**KIE Server の管理および監視** を参照してください。

- Red Hat Single Sign-On 認証を使用し... Red Hat Single Sign-On を使用したアプリケー...

- Red Hat Single Sign-On 認証を使用し、Red Hat Single Sign-On を使用したアプリケーションの対話で CORS のサポートが必要な場合は、**SSO_ENABLE_CORS** 変数を **true** に設定します。

- 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、ユーザーは、Business Central を使用して KIE Server にサービスをデプロイして数秒待ってから追加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、**コントローラーストラテジー** を使用するように **rhdm-authoring** 環境を設定します。これについての特定のニーズがない限り、この変更を行わないでください。コントローラーストラテジーを有効にする場合は、Business Central と同じ環境内のすべての KIE Server でこの変更を行ってください。



注記

高可用性の Business Central を使用する環境でコントローラーストラテジーを有効にしないでください。この環境では、コントローラーストラテジーは正しく機能しません。

KIE Server でコントローラーストラテジーを有効にするには、**KIE_SERVER_STARTUP_STRATEGY** 環境変数を **ControllerBasedStartupStrategy** に設定し、**KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED** 環境変数を **false** に設定します。

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: 7.9.1
```

7.9.1 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. 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一覧を表示しています。

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3.1 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**
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**

設定フィールド	環境変数	説明	例
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**
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**

3.5. 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

3.5.1. 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/rhdm-7/rhdm-kieserver-rhel8:7.9.1
```

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/rhdm-7/rhdm-kieserver-rhel8:7.9.1
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:7.9.1
```

以下に例を示します。

```
podman build . --tag registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

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:7.9.1 /bin/bash
```

以下に例を示します。

```
podman run -it --rm registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1 /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:7.9.1
docker://registry_address/image_name:7.9.1
```

以下に例を示します。

```
podman push registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
docker://registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

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/rhdm-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。

3.5.2. 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 Prometheus メトリクスを KIE Server に提供できます。カスタムクラスの作成手順は、**KIE Server の管理とモニターリング**の カスタムのメトリクスを使用した KIE Server の Prometheus メトリクスモニターリングの拡張 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。

- **KIE Server の管理およびモニターリング**の KIE Server 機能および拡張
- 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/rhdm-7/rhdm-kieserver-rhel8:7.9.1
```

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/rhdm-7/rhdm-kieserver-rhel8:7.9.1
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:7.9.1
```

以下に例を示します。

```
podman build . --tag registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

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:7.9.1
docker://registry_address/image_name:7.9.1
```

以下に例を示します。

```
podman push registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
docker://registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

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/rhdm-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 バージョン 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 バージョン 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 バージョン 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 バージョン 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 バージョン 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 バージョン 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 バージョン 3 および Red Hat OpenShift Container Platform バージョン 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 バージョン 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は、Red Hat OpenShift Container Platform ドキュメントの CLI ツール 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 バージョン 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 **rhdmcentr** 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-rhdmcentr-5-689mw:/opt/kie/data/.niogit .niogit
oc cp myapp-rhdmcentr-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 バージョン 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 **rhdmcentr** 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。

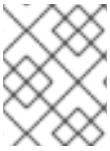
```
oc cp .niogit myappnew-rhdmcentr-abd24:/opt/kie/data/.niogit
oc cp maven-repository myappnew-rhdmcentr-abd24:/opt/kie/data/maven-repository
```

パート II. テンプレートを使用した RED HAT OPENSHIFT CONTAINER PLATFORM 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 バージョン 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。提供されたテンプレートを 1 つ使用して、特定のニーズに合わせて事前定義された Red Hat Decision Manager 環境をデプロイすることができます。

前提条件

- Red Hat OpenShift Container Platform バージョン 3.11 がデプロイされている。
- 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。
  - 高可用性オーソ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。
- 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。



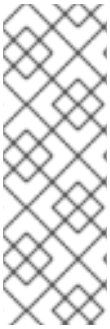
注記

クラスターの容量を確認する方法は、Red Hat OpenShift Container Platform 3.11 製品ドキュメントの クラスター容量の分析 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。
- **oc** コマンドを使用してプロジェクトにログインしている。**oc** コマンドランツールに関する詳細は、OpenShift の CLI リファレンス を参照してください。OpenShift Web コンソールを使用してテンプレートをデプロイするには、Web コンソールを使用してログインしている必要もあります。
- 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。
  - 複製された KIE Server Pod のセットには、デフォルトでデータベースに 1 つの 1Gi PV が必要になります。テンプレートパラメーターの PV サイズを変更できます。この要件は、外部データベースサーバーを使用する場合には適用されません。

- Business Central にはデフォルトで 1 Gi PV が必要です。テンプレートパラメーターで、Business Central 永続ストレージの PV サイズを変更することができます。

- Business Central の Pod をスケーリングする予定がある場合、OpenShift 環境では、**ReadWriteMany** モードで永続ボリュームがサポートされている。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。ただし、パフォーマンスと信頼性を最大化するには、GlusterFS を使用して、高可用性オーサーリング環境用に永続ボリュームをプロビジョニングします。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、アクセスモード を参照してください。



注記

Red Hat Decision Manager バージョン 7.5 以降では、Red Hat OpenShift Container Platform 3.x 向けのイメージおよびテンプレートが非推奨になりました。上記のイメージとテンプレートには新機能が追加されませんが、Red Hat OpenShift Container Platform バージョン 3.x の完全サポートが終了するまでサポートは継続されます。Red Hat OpenShift Container Platform バージョン 3.x の完全なサポートライフサイクルフェーズに関する詳細は、Red Hat OpenShift Container Platform のライフサイクルポリシー (最新バージョン以外) を参照してください。

注記

Red Hat Decision Manager テンプレートを Red Hat OpenShift Container Platform 4.x と一緒に使用しないでください。Red Hat Decision Manager を Red Hat OpenShift Container Platform 4.x にデプロイするには、**Operator を使用した Red Hat OpenShift Container Platform への Red Hat Decision Manager 環境のデプロイ** の説明を参照してください。

第5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。

以下の環境タイプが一般的です。

- **トライアル**: Red Hat Decision Manager のデモおよび評価のための環境です。この環境には、Business Central と KIE Server が含まれます。この環境はすばやく設定でき、これを使用して、アセットの開発や実行を評価し、体験できます。ただし、この環境では永続ストレージを使用せず、この環境でのいずれの作業も保存されません。

- **オーサリングまたは管理環境**: Business Central を使用してサービスを作成および変更し、サービスを KIE Server で実行するために使用できる環境アーキテクチャーです。これは、オーサリング作業用の Business Central を提供する Pod およびサービス実行用の 1 つ以上の KIE Server を提供する Pod で設定されます。それぞれの KIE Server が 1 つの Pod となり、Pod はスケールアップまたはスケールダウンを随時実行して複製できます。Business Central を使用して、それぞれの KIE Server でサービスをデプロイしたり、デプロイ解除したりすることができます。

- **イミュータブルサーバーを使用するデプロイメント**: ステージングおよび実稼働目的で既存のサービスを実行するための代替の環境です。この環境では、KIE Server Pod のデプロイ時に、

サービスまたはサービスのグループを読み込み、起動するイメージをビルドします。この Pod でサービスを停止したり、新しいサービスを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したり、別の方法で設定を変更する必要がある場合は、新規のサーバーイメージをデプロイして、古いサーバーと入れ替えます。このシステムでは、KIE Server は OpenShift 環境の Pod のように実行されるため、任意のコンテナーベースの統合ワークフローを使用することができ、他のツールを使用して Pod を管理する必要はありません。

OpenShift に Red Hat Decision Manager 環境をデプロイするには、Red Hat Decision Manager で用意した OpenShift テンプレートを使用します。

5.1. 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 が 1 台含まれています。

単一のオーサリング環境ではデフォルトで、**コントローラーストラテジー** を使用します。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 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5.1 単一のオーサリング環境のアーキテクチャー図

KIE Server のクラスターリングと複数の 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 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 の間にレイヤーを提供します。独立した KIE Server 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 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 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。

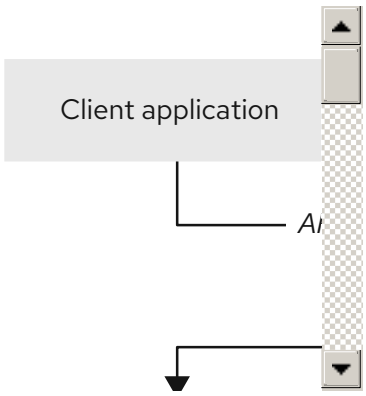
KIE Server デプロイメント (スケーリング可能) ごとに、現在の状態を反映する ConfigMap を作成します。Business Central は、ConfigMap を読み込むことで全 KIE Server を検出します。

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、全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、独自の再デプロイをトリガーします。

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5.2 高可用性オーサリング環境のアーキテクチャー図



第6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、「イメージストリームとイメージレジストリーの可用性確認」 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。

6.1. イメージストリームとイメージレジストリーの可用性確認

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。これらのイメージをダウンロードするために、OpenShift ではイメージの場所情報が含まれる **イメージストリーム** が必要になります。また、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。

OpenShift 環境のバージョンによっては、必要なイメージストリームが含まれている場合があります。イメージストリームが提供されているかどうかを確認する必要があります。デフォルトでイメージストリームが OpenShift に含まれている場合は、OpenShift インフラストラクチャーがレジストリー認証サーバー用に設定されているのであれば、使用できます。管理者は、OpenShift 環境のインストール時に、レジストリーの認証設定を完了する必要があります。

それ以外の方法として、レジストリー認証を独自のプロジェクトで設定し、イメージストリームをそのプロジェクトにインストールすることができます。

手順

1. Red Hat OpenShift Container Platform が Red Hat レジストリーへのアクセス用に、ユーザー名とパスワードで設定されているかを判断します。必須の設定に関する詳細は、レジストリーの場所の設定 を参照してください。OpenShift オンラインサブスクリプションを使用する場合は、Red Hat レジストリー用のアクセスはすでに設定されています。

2. Red Hat OpenShift Container Platform が Red Hat レジストリーへのアクセス用のユーザー名とパスワードで設定されている場合は、以下のコマンドを実行します。

```
$ oc get imagestreamtag -n openshift | grep -F rhdm79-decisioncentral-openshift
$ oc get imagestreamtag -n openshift | grep -F rhdm79-kieserver-openshift
```

   両コマンドの出力が空でない場合は、必要なイメージストリームが **openshift** namespace にあるため、これ以外の操作は必要ありません。

3. コマンドの 1 つまたは複数の出力が空白の場合や、Red Hat レジストリーにアクセスするために、OpenShift をユーザー名およびパスワードで設定していない場合は、以下の手順を実行してください。

   a. **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。

b. 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、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する必要があります。

c. **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。

d. 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。

e. 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。

f. Software Downloads ページから製品の配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードし、**rhdm79-image-streams.yaml** ファイルを展開してください。

g. 以下のコマンドを入力します。

```
$ oc apply -f rhdm79-image-streams.yaml
```

注記

上記の手順を完了したら、イメージストリームを独自のプロジェクトの名前空間にインストールします。今回の例では、テンプレートのデプロイ時に **IMAGE_STREAM_NAMESPACE** パラメーターをこのプロジェクトの名前に設定する必要があります。

6.2. 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 シークレット の章を参照してください。

KIE Server への HTTP アクセス用に SSL 証明書を作成し、これをシークレットとして OpenShift 環境に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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 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
```

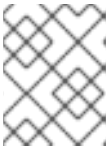
6.3. BUSINESS CENTRAL へのシークレットの作成

Business Central が含まれている環境では、Business Central への HTTP アクセス用の 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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 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 **decisioncentral-app-secret** を生成します。

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

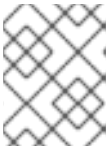
6.4. SMART ROUTER のシークレットの作成

Smart Router が含まれている環境では、Smart Router への HTTP アクセス用の 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。

Smart Router の証明書およびキーストアに、KIE Server または Business Central で使用されているものと同じものを指定しないでください。

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。自己署名または購入した SSL 証明書でキーストアを作成する方法は、SSL 暗号化キーおよび証明書 を参照してください。



注記

実稼働環境で、Smart Router 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 **smartrouter-app-secret** を生成します。

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

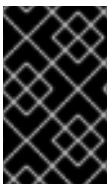
6.5. 管理ユーザーのシークレットの作成

Red Hat Decision Manager 管理ユーザーアカウントのユーザー名とパスワードを含む汎用シークレットを作成する必要があります。このシークレットは、試行版テンプレート以外のテンプレートを使用して Red Hat Decision Manager をデプロイするのに必要です。

シークレットには、リテラルのユーザー名とパスワードが含まれている必要があります。ユーザー名のキー名は **KIE_ADMIN_USER** です。パスワードのキー名は **KIE_ADMIN_PWD** です。

複数のテンプレートを使用して Red Hat Decision Manager のコンポーネントをデプロイする場合は、これらのすべてのデプロイメントに同じシークレットを使用します。コンポーネントは、このユーザーアカウントを利用して相互に通信します。

Business Central が含まれている環境で、このユーザーアカウントを使用して Business Central にログインすることもできます。



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パスワードを持つ同じユーザーを設定する必要があります。

手順

oc コマンドを使用し、ユーザー名およびパスワードの **kie-admin-user-secret** という汎用シークレットを生成します。

```
$ oc create secret generic rhpam-credentials --from-literal=KIE_ADMIN_USER=adminUser --from-literal=KIE_ADMIN_PWD=adminPassword
```

このコマンドで、**adminPassword** を管理ユーザーのパスワードに置き換えます。必要に応じて、**adminUser** を管理ユーザーの別のユーザー名に置き換えることができます。

6.6. GLUSTERFS 設定の変更

オーサリング環境をデプロイする場合は、OpenShift 環境が GlusterFS を使用して永続ストレージボリュームを提供するかどうかを確認する必要があります。GlusterFS を使用している場合は、Business Central の最適なパフォーマンスを確保するために、ストレージクラスの設定を変更して GlusterFS ス

トレージをチューニングする必要があります。

手順

1. お使いの環境で GlusterFS が使用されているかどうかを確認するには、以下のコマンドを実行します。

```
oc get storageclass
```

この結果で、**(default)** マーカーが、**glusterfs** をリストするストレージクラスにあるかどうかを確認します。たとえば、以下の結果では、デフォルトのストレージクラスが **gluster-container** であり、**glusterfs** をリストします。

```
NAME              PROVISIONER                    AGE
gluster-block     gluster.org/glusterblock       8d
gluster-container (default) kubernetes.io/glusterfs 8d
```

結果に、**glusterfs** をリストしないデフォルトストレージクラスが含まれる場合、または結果が空の場合は、変更する必要がありません。変更しない場合は、残りの手順を省略します。

2. デフォルトストレージクラスの設定を YAML ファイルに保存するには、以下のコマンドを実行します。

```
oc get storageclass <class-name> -o yaml >storage_config.yaml
```

<class-name> はデフォルトのストレージクラス名に置き換えます。以下に例を示します。

```
oc get storageclass gluster-container -o yaml >storage_config.yaml
```

3. **storage_config.yaml** ファイルを編集します。

   a. 以下のキーがある行を削除します。

   - **creationTimestamp**
   - **resourceVersion**
   - **selfLink**
   - **uid**

   b. Business Central を、高可用性設定がない単一の Pod としてのみ使用する予定の場合は、**volumeoptions** キーが含まれる行に、以下のオプションを追加します。

```
features.cache-invalidation on
performance.nl-cache on
```

以下に例を示します。

volumeoptions: client.ssl off, server.ssl off, features.cache-invalidation on, performance.nl-cache on

   c. Business Central を高可用性設定で使用する予定の場合は、**volumeoptions** キーが含まれる行に、以下のオプションを追加します。

```
features.cache-invalidation on
nfs.trusted-write on
nfs.trusted-sync on
performance.nl-cache on
performance.stat-prefetch off
performance.read-ahead off
performance.write-behind off
performance.readdir-ahead off
performance.io-cache off
performance.quick-read off
performance.open-behind off
locks.mandatory-locking off
performance.strict-o-direct on
```

以下に例を示します。

volumeoptions: client.ssl off, server.ssl off, features.cache-invalidation on, nfs.trusted-write on, nfs.trusted-sync on, performance.nl-cache on, performance.stat-prefetch off, performance.read-ahead off, performance.write-behind off, performance.readdir-ahead off, performance.io-cache off, performance.quick-read off, performance.open-behind off, locks.mandatory-locking off, performance.strict-o-direct on

4. 既存のデフォルトストレージクラスを削除するには、以下のコマンドを実行します。

```
oc delete storageclass <class-name>
```

<class-name> はデフォルトのストレージクラス名に置き換えます。以下に例を示します。

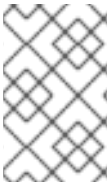
```
oc delete storageclass gluster-container
```

5. 新しい設定を使用してストレージクラスを再作成するには、以下のコマンドを実行します。

```
oc create -f storage_config.yaml
```

6.7. 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。



注記

高可用性オーサリング環境をデプロイする場合、パフォーマンスと信頼性を最大化するには、GlusterFS を使用して永続ボリュームをプロビジョニングします。「GlusterFS 設定の変更」 の説明に従って GlusterFS ストレージクラスを設定します。

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については、Red Hat OpenShift Container Platform 3.11 ドキュメントの **クラスターの設定** の NFS を使用した永続ストレージを参照してください。

6.8. S2I ビルドに使用する BUSINESS CENTRAL からのソースコードの展開

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展開して、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展開します。

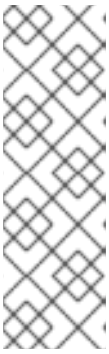
```
git clone https://<decision-central-host>:443/git/<MySpace>/<MyProject>
```

このコマンドでは、以下の変数を置き換えてください。

- **<decision-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://<decision-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

6.9. 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。

   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。

   a. **Red Hat Process Automation Manager 7.9.1 Offliner Content List**をクリックして、Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-offliner.zip** をダウンロードします。

   b. **rhdm-7.9.1-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。

   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 repository tools から利用できる Maven Repository Provisioner ユーティリティーを使用できます。

3. 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 は、プロジェクトの **pom.xml** ファイルへの正しい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 repository tools から利用できる Maven Repository Provisioner ユーティリティーを使用できます。

第7章 トライアル環境

試用版 (評価版) の Red Hat Decision Manager 環境をデプロイできます。この環境は、サービスのオーサリングと管理を行う Business Central と、サービスのテスト実行を行う KIE Server で設定されます。

この環境には、永続ストレージが含まれません。トライアル環境で作成または変更するアセットは保存されません。

この環境は、テストおよびデモ用のアクセスを前提として設計されています。CORS (Cross-Origin Resource Sharing) をサポートします。これは、ページの他のリソースが他のサーバーによって提供される場合に、ブラウザーを使用して KIE Server エンドポイントにアクセスできることを意味します。KIE Server エンドポイントは通常 REST 呼び出しを対象としていますが、一部のデモ設定でブラウザーアクセスが必要になることがあります。

7.1. 試用環境のデプロイ

トライアル環境をデプロイする手順は最小限です。必要な設定はなく、すべてのパスワードが単一の値に設定されます。デフォルトのパスワードは、**RedHat** です。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードします。
2. **rhdm79-trial-ephemeral.yaml** テンプレートファイルを抽出します。
3. 以下の方法を使用してテンプレートをデプロイします。
   - OpenShift Web UI では、**Add to Project → Import YAML / JSON**を選択し、**rhdm79-trial-ephemeral.yaml** ファイルを選択するか、その内容を貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
   - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

     ```
     oc new-app -f <template-path>/rhdm79-trial-ephemeral.yaml
     ```

     このコマンドラインでは、**<template-path>** は、ダウンロードしたテンプレートファイルのパスに置き換えます。
4. 必要に応じて、このテンプレートに記載されているようにパラメーターを設定します。通常の試用版の開発では、以下のパラメーターのみが必要です。
   - **ImageStream 名前空間 (IMAGE_STREAM_NAMESPACE)**: イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「イメージストリームとイメージレジストリーの可用性確認」 を参照) は、名前空間が **openshift** になります。イメージストリームファイルをインストールした場合は、名前空間が OpenShift プロジェクトの名前になります。
5. 使用している方法に応じて、環境の作成を終了します。
   - OpenShift Web UI の場合は **Create** をクリックします。

    - **This will create resources that may have security or project behavior implications** のポップアップメッセージが表示される可能性があります。このメッセージが表示された場合は、**Create Anyway** をクリックします。

- 完了し、コマンドラインを実行します。

第8章 オーサリングまたは管理サーバー環境

Business Central を使用してサービスの作成や変更を行う環境や、Business Central が管理する KIE Server でサービスを実行する環境をデプロイできます。この環境は、Business Central と 1 つまたは複数の KIE Server で設定されます。

Business Central を使用するとサービスの開発や KIE Server へのデプロイを実行できます。複数の KIE Server を Business Central に接続して、各サーバーへのサービスのデプロイを管理することができます。

必要な場合は、別の環境を作成して Business Central の 1 つのデプロイメントを使用してサービスのオーサリングを行い (**オーサリング環境**)、Business Central のもう 1 つのデプロイメントを使用して複数 KIE Server のステージングまたは実稼働サーバーのデプロイメントを管理できます (**管理サーバー環境**)。通常は、1 つの専用オーサリング環境には 1 つの KIE Server があれば十分です。外部 Maven リポジトリーを使用してオーサリング環境のサービスを保存し、それらを別の管理サーバー環境にデプロイできます。

Red Hat Decision Manager では、オーサリング環境と管理サーバー環境のデプロイの手順は同じです。最初に、Business Central と 1 つの KIE Server で設定されるオーサリング環境テンプレートをデプロイする必要があります。

必要な場合は、追加の KIE Server テンプレートを同じ名前空間にデプロイし、複数の KIE Server を含む環境を作成できます。この環境は、サービスのステージングおよび実稼働のデプロイメント用の管理サーバー環境にすることができます。

必要に応じて、単一のオーサリング環境テンプレートまたは高可用性 (HA) オーサリング環境テンプレートのいずれかをデプロイできます。

単一オーサリング環境には 2 つの Pod が含まれます。それらの Pod の 1 つは Business Central を実行し、もう 1 つは KIE Server を実行します。この環境は、単一ユーザーのオーサリングや、OpenShift インフラストラクチャーのリソースが制限されている場合に最も適しています。これには、**ReadWriteMany** アクセスモードをサポートする永続ボリュームは不要です。

単一のオーサリング環境では、Business Central をスケーリングすることはできません。KIE Server はスケーリングできます。

HA オーサリング環境では、Business Central と KIE Server の両方がスケーリング可能な Pod で提供されます。Pod をスケーリングすると、永続ストレージはコピー間で共有されます。

Business Central で高可用性機能を有効にするには、AMQ および Data Grid を含む追加の Pod が必要です。これらの Pod は高可用性オーサリングテンプレートで設定され、デプロイされます。高可用性オーサリング環境を使用して、特に複数のユーザーが同時にオーサリングに関与する場合に、信頼性と応答性を最大限提供します。

Red Hat Decision Manager の現行バージョンでは、HA オーサリング環境は特定の制限付きでサポートされています。

- Business Central Pod がユーザーがそれを使用している間にクラッシュすると、ユーザーにはエラーメッセージが送られ、ユーザーは別の Pod にリダイレクトされます。この場合、再度ログインする必要はありません。
- ユーザーの操作時に Business Central Pod がクラッシュする場合は、コミット (保存) されていないデータが失われる可能性があります。
- プロジェクトの作成時に Business Central Pod がクラッシュする場合は、使用できないプロジェクトが作成される可能性があります。

- アセットの作成時に Business Central Pod がクラッシュする場合は、アセットが作成されるものの、インデックス化されないため使用できない可能性があります。ユーザーは Business Central でアセットを開き、再度保存してインデックス化することができます。

- サービスを KIE Server にデプロイすると、KIE Server デプロイメントが再度ロールアウトされます。ロールアウトが完了するまで、同じ KIE Server に別のサービスをデプロイできません。

高可用性オーサリング環境では、必要に応じて、別の管理対象またはイミュータブル KIE Server を追加でデプロイすることも可能です。Business Central は、イミュータブル KIE Server や管理対象 KIE Server など、同じ namespace 内の KIE Server を自動検出できます。

単一のオーサリング環境で管理対象またはイミュータブルな KIE Server を追加でデプロイする場合は、「追加の KIE Server を Business Central に接続するための **OpenShiftStartupStrategy** 設定の有効化」 に記載されているように、環境内の **OpenShiftStartupStrategy** 設定を手作業で有効にする手順が別途必要になります。この設定により、他の KIE Server の検出が可能になります。

管理対象の KIE Server のデプロイ方法は、「オーサリング環境または管理環境向けの追加の管理 KIE Server のデプロイ」 を参照してください。

イミュータブルな KIE Server をデプロイする方法は、「S2I ビルドの使用によるイミュータブル KIE Server のデプロイ」 および 「KJAR サービスからのイミュータブル KIE Server のデプロイ」 を参照してください。

8.1. オーサリング環境のデプロイメント

OpenShift テンプレートを使用し、単一または高可用性オーサリング環境をデプロイできます。この環境は、Business Central および単一の KIE Server で設定されます。

8.1.1. オーサリング環境用のテンプレートの設定開始

単一オーサリング環境をデプロイする必要がある場合は、**rhdm79-authoring.yaml** テンプレートファイルを使用します。

高可用性オーサリング環境をデプロイする必要がある場合は、**rhdm79-authoring-ha.yaml** テンプレートファイルを使用します。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードします。

2. 必要なテンプレートファイルを展開します。

3. 以下のいずれかの方法を使用してテンプレートのデプロイを開始します。

   - OpenShift Web UI を使用するには、OpenShift アプリケーションコンソールで **Add to Project → Import YAML / JSON** を選択してから **<template-file-name>.yaml** ファイルを選択または貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。

   - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

     ```
     oc new-app -f <template-path>/<template-file-name>.yaml -p
     DECISION_CENTRAL_HTTPS_SECRET=decisioncentral-app-secret -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
     ```

-

このコマンドラインで、以下のように変更します。

- **<template-path>** を、ダウンロードしたテンプレートファイルのパスに置き換えます。
- **<template-file-name>** は、テンプレート名に置き換えます。
- 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。

次のステップ

テンプレートのパラメーターを設定します。「オーサリング環境に必要なパラメーターの設定」の手順に従い、共通のパラメーターを設定します。テンプレートファイルを表示して、すべてのパラメーターの説明を確認します。

8.1.2. オーサリング環境に必要なパラメーターの設定

テンプレートをオーサリング環境をデプロイするように設定する場合は、いずれの場合でも以下のパラメーターを設定する必要があります。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - Credentials secret (**CREDENTIALS_SECRET**): 「管理ユーザーのシークレットの作成」で作成される管理ユーザーの認証情報を含むシークレットの名前。
   - **Business Central サーバーキーストアのシークレット名** (**DECISION_CENTRAL_HTTPS_SECRET**): 「Business Central へのシークレットの作成」で作成した Business Central のシークレットの名前。
   - KIE Server Keystore Secret Name(**KIE_SERVER_HTTPS_SECRET**): 「KIE Server のシークレットの作成」で作成した KIE Server のシークレットの名前。
   - **Business Central サーバーの証明署名** (**DECISION_CENTRAL_HTTPS_NAME**): 「Business Central へのシークレットの作成」で作成したキーストアの証明書の名前。
   - **Business Central サーバーキーストアのパスワード** (**DECISION_CENTRAL_HTTPS_PASSWORD**): 「Business Central へのシークレットの作成」で作成したキーストアのパスワード。
   - KIE Server Certificate Name(**KIE_SERVER_HTTPS_NAME**): 「KIE Server のシークレットの作成」で作成したキーストアの証明書名。
   - KIE Server Keystore Password (**KIE_SERVER_HTTPS_PASSWORD**): 「KIE Server のシークレットの作成」で作成したキーストアのパスワード。
   - **アプリケーション名** (**APPLICATION_NAME**): OpenShift アプリケーションの名前。これは、Business Central Monitoring および KIE Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。

- ImageStream 名前空間 (**IMAGE_STREAM_NAMESPACE**): イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「イメージストリームとイメージレジストリーの可用性確認」 を参照) は、名前空間が **openshift** になります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

8.1.3. オーサリング環境用のイメージストリーム namespace の設定

openshift ではない名前空間でイメージストリームを作成した場合は、テンプレートで名前空間を設定する必要があります。

すべてのイメージストリームが Red Hat OpenShift Container Platform 環境ですでに利用可能な場合は、この手順を省略できます。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

「イメージストリームとイメージレジストリーの可用性確認」 の説明に従ってイメージストリームファイルをインストールした場合は、**ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) パラメーターを OpenShift プロジェクトの名前に設定します。

8.1.4. オーサリング環境用のオプションの Maven リポジトリーの設定

テンプレートをオーサリング環境をデプロイするように設定する際、ビルドされた KJAR ファイルを外部の Maven リポジトリーに配置する必要がある場合は、リポジトリーにアクセスするためにパラメーターを設定する必要があります。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

カスタム Maven リポジトリーへのアクセスを設定するには、以下のパラメーターを設定します。

- **Maven リポジトリーの URL** (**MAVEN_REPO_URL**): Maven リポジトリーの URL。
- **Maven リポジトリーの ID** (**MAVEN_REPO_ID**): Maven リポジトリーの ID。デフォルト値は **repo-custom** です。
- **Maven リポジトリーのユーザー名** (**MAVEN_REPO_USERNAME**): Maven リポジトリーのユーザー名。
- **Maven リポジトリーのパスワード** (**MAVEN_REPO_PASSWORD**): Maven リポジトリーのパスワード。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

重要

Business Central プロジェクトを KJAR アーティファクトとして外部の Maven リポジトリーにエクスポートまたはプッシュするには、全プロジェクトの **pom.xml** ファイルにもリポジトリー情報を追加する必要があります。Business Central プロジェクトの外部リポジトリーへ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。

8.1.5. オーサリング環境の公開インターネットへの接続のない環境に Maven ミラーへのアクセスを設定する

テンプレートをオーサリング環境をデプロイするように設定する際に、 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」 に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

Maven ミラーへのアクセスを設定するには、以下のパラメーターを設定します。

- Maven ミラー URL (**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」 で設定した 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 (**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:*,!repo-rhdmcentr** です。この値で、Maven は Business Central のビルトイン Maven リポジトリーからアーティファクトを直接取得し、ミラーから他の必要なアーティファクトを取得します。外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合は、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。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。

  - ビルトイン Business Central Maven リポジトリー (**DECISION_CENTRAL_MAVEN_SERVICE**) を設定する場合は、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-rhdmcentr**)。

- 両リポジトリーを設定した場合は、ミラーから両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-rhdmcentr,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

8.1.6. 高可用性オーサリング環境用の Business Central と KIE Server のレプリカの設定

高可用性オーサリング環境をデプロイする場合に、デフォルトでは、Business Central のレプリカと KIE Server のレプリカが 2 つずつ最初に作成されます。

必要に応じて、レプリカの数を変更できます。

単一のオーサリング環境では、この手順を飛ばして次に進んでください。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

レプリカの数を変更するには、次のパラメーターを設定します。

- **Business Central Container レプリカ** (**DECISION_CENTRAL_CONTAINER_REPLICAS**): デプロイメントで Business Central に最初に作成するレプリカ数。
- **KIE Server コンテナーのレプリカ** (**KIE_SERVER_CONTAINER_REPLICAS**): デプロイメントで KIE Server に最初に作成するレプリカ数。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

8.1.7. オーサリング環境用の Git フックディレクトリーの指定

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 リポジトリーの対話を容易にすることができます。

Git フックを使用する必要がある場合は、Git フックディレクトリーを設定する必要があります。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

Git フックディレクトリーを設定するには、以下のパラメーターを設定します。

- **Git フックディレクトリー** (**GIT_HOOKS_DIR**): Git フックディレクトリーへの完全修飾パス (例: **/opt/kie/data/git/hooks**)。ディレクトリーの内容を指定し、これを指定されたパスにマウ

ントする必要があります。設定マップまたは永続ボリュームを使用して Git フックディレクトリーを指定し、マウントする方法は、「(オプション) Git フックディレクトリーの指定」 を参照してください。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

8.1.8. 高可用性デプロイメントのリソース使用状況の設定

高可用性テンプレート (**rhdm79-authoring-ha.yaml**) をデプロイしている場合は、要件に合わせてパフォーマンスを最適化するためにリソースの使用を任意で設定することができます。

単一オーサリング環境テンプレート (**rhdm79-authoring.yaml**) をデプロイしている場合は、この手順を省略してください。

リソースのサイジングの詳細は、Red Hat OpenShift Container Platform 3.11 の製品ドキュメントの以下のセクションを参照してください。

- アプリケーションメモリーのサイジング
- コンピュートリソース

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

以下のパラメーターをテンプレートに設定します (該当する場合)。

- **Business Central コンテナーのメモリー制限** (**DECISION_CENTRAL_MEMORY_LIMIT**): Business Central コンテナーについて OpenShift 環境で必要とされるメモリー量。デフォルト値は **8Gi** です。
- **Business Central の JVM 最大メモリー割合** (**DECISION_CENTRAL_JAVA_MAX_MEM_RATIO**): Business Central の Java Virtual Machine に使用されるコンテナーメモリーのパーセンテージ。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 80% を制限値として **80** になります。
- **Business Central コンテナーの CPU 制限**(**DECISION_CENTRAL_CPU_LIMIT**): Business Central の CPU 使用の最大値。デフォルト値は **2000m** です。
- **KIE Server コンテナーのメモリー制限** (**KIE_SERVER_MEMORY_LIMIT**): KIE Server コンテナーについて OpenShift 環境で必要とされるメモリー量。デフォルト値は **1Gi** です。
- **KIE Server コンテナーの CPU 制限**(**KIE_SERVER_CPU_LIMIT**): KIE Server の CPU 使用の最大値。デフォルト値は **1000m** です。
- **DataGrid Container のメモリー制限** (**DATAGRID_MEMORY_LIMIT**): Red Hat Data Grid コンテナーについて OpenShift 環境で必要とされるメモリー量。デフォルト値は **2Gi** です。
- **DataGrid Container CPU 制限**(**DATAGRID_CPU_LIMIT**): Red Hat Data Grid の CPU 使用の最大値。デフォルト値は **1000m** です。

8.1.9. オーサリング環境用の RH-SSO 認証パラメーターの設定

RH-SSO 認証を使用する必要がある場合、テンプレートをオーサリング環境をデプロイするように設定する際に追加の設定を実行します。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- Red Hat Decision Manager のレルムが RH-SSO 認証システムに作成されている。
- Red Hat Decision Manager のユーザー名およびパスワードが RH-SSO 認証システムに作成されている。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- デプロイしている Red Hat Decision Manager 環境の全コンポーネントに対して、クライアントが RH-SSO 認証システムに作成されている。クライアントのセットアップには、コンポーネントの URL が含まれます。環境のデプロイ後に URL を確認し、編集できます。または、Red Hat Decision Manager のデプロイメントでクライアントを作成できます。ただし、このオプションの環境に対する制御の詳細度合はより低くなります。
- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
    - **RH-SSO URL** (**SSO_URL**): RH-SSO の URL。
    - **RH-SSO レルム名** (**SSO_REALM**): Red Hat Decision Manager の RH-SSO レルム。
    - **RH-SSO が無効な SSL 証明書の検証** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): RH-SSO インストールで有効な HTTPS 証明書を使用していない場合は **true** に設定します。
2. 以下の手順のいずれかを実行します。
    a. RH-SSO で Red Hat Decision Manager のクライアントを作成した場合は、テンプレートで以下のパラメーターを設定します。
        - **Business Central RH-SSO クライアント名** (**DECISION_CENTRAL_SSO_CLIENT**): Business Central の RH-SSO クライアント名。
        - **Business Central RH-SSO クライアントのシークレット** (**DECISION_CENTRAL_SSO_SECRET**): Business Central のクライアント向けに RH-SSO で設定するシークレット文字列。
        - **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server の RH-SSO クライアント名。

- **KIE Server RH-SSO クライアントのシークレット (KIE_SERVER_SSO_SECRET)**: KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。

b. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。

- **Business Central RH-SSO クライアント名 (DECISION_CENTRAL_SSO_CLIENT)**: Business Central 向けに RH-SSO に作成するクライアント名。
- **Business Central RH-SSO クライアントのシークレット (DECISION_CENTRAL_SSO_SECRET)**: Business Central のクライアント向けに RH-SSO で設定するシークレット文字列。
- **KIE Server RH-SSO クライアント名 (KIE_SERVER_SSO_CLIENT)**: KIE Server 向けに RH-SSO に作成するクライアント名。
- **KIE Server RH-SSO クライアントのシークレット (KIE_SERVER_SSO_SECRET)**: KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
- **RH-SSO レルムの管理者のユーザー名 (SSO_USERNAME)** および **RH-SSO レルムの管理者のパスワード (SSO_PASSWORD)**: Red Hat Decision Manager の RH-SSO レルムの管理者ユーザーに指定するユーザー名とパスワード必要なクライアントを作成するためにこのユーザー名およびパスワードを指定する必要があります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

デプロイの完了後に、RH-SSO 認証システムで Red Hat Decision Manager のコンポーネントの URL が正しいことを確認してください。

8.1.10. オーサリング環境用の LDAP 認証パラメーターの設定

LDAP 認証を使用する必要がある場合は、テンプレートをオーサリング環境をデプロイするように設定する際に追加の設定を実行します。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- LDAP システムに Red Hat Decision Manager のユーザー名およびパスワードを作成している。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

1. テンプレートの **AUTH_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 **LdapExtended** ログインモジュールの設定に対応します。これらの設定に関する説明は、LdapExtended login module 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。

LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。

- RoleMapping rolesProperties ファイルパス (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties** など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「(任意) LDAP ロールマッピングファイルの指定」を参照してください。

- RoleMapping replaceRole プロパティー (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): **true** に設定した場合、マッピングしたロールは、LDAP サーバーに定義したロールに置き換えられます。**false** に設定した場合は、LDAP サーバーに定義したロールと、マッピングしたロールの両方がユーザーアプリケーションロールとして設定されます。デフォルトの設定は **false** です。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

8.1.11. オーサリング環境用の Prometheus メトリクス収集の有効化

KIE Server デプロイメントを Prometheus を使用してメトリクスを収集し、保存するように設定する必要がある場合、デプロイ時に KIE Server でこの機能のサポートを有効にします。

前提条件

- 「オーサリング環境用のテンプレートの設定開始」 に説明されているテンプレートの設定を開始していること。

手順

Prometheus メトリクス収集のサポートを有効にするには、**Prometheus Server 拡張無効** (**PROMETHEUS_SERVER_EXT_DISABLED**) パラメーターを **false** に設定します。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「オーサリング環境用テンプレートのデプロイの実行」 の手順に従います。

Prometheus メトリクス収集の方法は、**KIE Server の管理および監視** を参照してください。

8.1.12. オーサリング環境用テンプレートのデプロイの実行

OpenShift Web UI またはコマンドラインで必要なすべてのパラメーターを設定した後に、テンプレートのデプロイを実行します。

手順

使用している方法に応じて、以下の手順を実行します。

- OpenShift Web UI の場合は **Create** をクリックします。
  - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。
- コマンドラインに入力して、Enter キーを押します。

次のステップ

環境の要件に応じて、任意で 10章 *環境をデプロイした後の任意の手順* で説明されている手順を完了します。

8.2. 追加の KIE SERVER を BUSINESS CENTRAL に接続するための OPENSHIFTSTARTUPSTRATEGY 設定の有効化

Red Hat Decision Manager オーサリングテンプレートを使用してデプロイされた環境では、Business Central は 1 つの KIE Server を管理します。KIE Server Pod をスケーリングすることができますが、すべてのコピーが同じサービスを実行します。

Business Central に追加で KIE Server を接続できます。ただし、**rhdm79-authoring.yaml** を使用して単一のオーサリング環境をデプロイした場合は、環境で **OpenShiftStartupStrategy** 設定を有効にする必要があります。**OpenShiftStartupStrategy** を有効にすると、Business Central は同じ名前空間にある KIE Server を検出し、これらの KIE Server は Business Central に接続するように設定できます。

OpenShiftStartupStrategy 設定では、KIE Server にサービスをデプロイすると、KIE Server デプロイメントが再度ロールアウトされます。ロールアウトが完了するまで、同じ KIE Server に別のサービスをデプロイできません。ロールアウトにはかなり時間が掛かる可能性があるため、**OpenShiftStartupStrategy** 設定によっては、オーサリング環境には適さない場合があります。

rhdm79-authoring-ha.yaml テンプレートを使用して高可用性オーサリング環境をデプロイした場合は、この手順を実行しないでください。この環境では、デフォルトで **OpenShiftStartupStrategy** 設定が有効です。

追加の KIE Server を Business Central に接続する場合を除き、この手順を実行しないでください。

前提条件

- **rhdm79-authoring.yaml** テンプレートを使用してオーサリング環境をデプロイしている。
- **oc** ツールを使用して環境がデプロイされている OpenShift プロジェクトにログインしている。

手順

1. 以下のコマンドを入力して、プロジェクトにデプロイされているデプロイメント設定を表示します。

```
$ oc get dc
```

2. コマンドの出力で、Business Central Pod と KIE Server Pod のデプロイメント設定名を見つけます。

   - Business Central のデプロイメント設定の名前は、**myapp-rhdmcentr** です。**myapp** を、テンプレートの **APPLICATION_NAME** パラメーターに設定される環境のアプリケーション名に置き換えます。

   - KIE Server のデプロイメント設定の名前は **myapp-kieserver** です。**myapp** をアプリケーション名に置き換えます。

3. 以下のコマンドを入力し、Pod で **OpenShiftStartupStrategy** 設定を有効にします。

```
$ oc env myapp-rhdmcentr KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED=true
$ oc env myapp-kieserver
KIE_SERVER_STARTUP_STRATEGY=OpenShiftStartupStrategy
```

   これらのコマンドで、**myapp-rhdmcentr** を Business Central デプロイメント設定名に、**myapp-kieserver** を KIE Server デプロイメント設定名に置き換えます。

4. **OpenShiftStartupStrategy** 設定を有効にする場合、デフォルトで Business Central は、オーサリングテンプレートと同じ値の **APPLICATION_NAME** パラメーターでデプロイされる KIE Server のみを検出します。その他のアプリケーション名を持つ KIE Server を Business Central に接続する必要がある場合は、以下のコマンドを入力します。

```
$ oc env myapp-rhdmcentr
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED=true
```

   このコマンドで、**myapp-rhdmcentr** を Business Central デプロイメント設定名に置き換えます。

8.3. オーサリング環境または管理環境向けの追加の管理 KIE SERVER のデプロイ

追加の管理 KIE Server をオーサリング環境または管理環境にデプロイできます。サーバーを Business Central デプロイメントと同じプロジェクトにデプロイします。

rhdm79-authoring.yaml テンプレートを使用して単一のオーサリング環境をデプロイした場合には、環境内の **OpenShiftStartupStrategy** 設定を有効にして、Business Central が KIE Server に接続できるようにします。**OpenShiftStartupStrategy** 設定を有効にする方法は、「追加の KIE Server を Business Central に接続するための **OpenShiftStartupStrategy** 設定の有効化」 を参照してください。高可用性オーサリング環境の場合は、この手順を実行する必要はありません。

KIE Server は、Maven リポジトリーからサービスを読み込みます。サーバーを Business Central ビルトインリポジトリーまたは外部リポジトリーのいずれかを使用するように設定する必要があります。

サーバーは、サービスが読み込まれていない状態で起動します。Business Central または KIE Server の REST API を使用してサーバー上にサービスをデプロイまたはデプロイ解除します。

8.3.1. 追加の管理 KIE Server テンプレート設定の開始

追加の管理 KIE Server をデプロイするには、**rhdm79-kieserver.yaml** テンプレートファイルを使用します。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードします。
2. **rhdm79-kieserver.yaml** テンプレートファイルを展開します。
3. 以下のいずれかの方法を使用してテンプレートのデプロイを開始します。
   - OpenShift Web UI を使用するには、OpenShift アプリケーションコンソールで **Add to Project → Import YAML / JSON** を選択してから、**rhdm79-kieserver.yaml** ファイルを選択するか、またはこれを貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
   - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

     ```
     oc new-app -f <template-path>/rhdm79-kieserver.yaml -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
     ```

     このコマンドラインで、以下のように変更します。
     - **<template-path>** を、ダウンロードしたテンプレートファイルのパスに置き換えます。
     - 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。

次のステップ

テンプレートのパラメーターを設定します。「追加の管理 KIE Server に必要なパラメーターの設定」の手順に従い、共通のパラメーターを設定します。テンプレートファイルを表示して、すべてのパラメーターの説明を確認します。

8.3.2. 追加の管理 KIE Server に必要なパラメーターの設定

テンプレートを追加の管理 KIE Server をデプロイするように設定する際、いずれの場合でも以下のパラメーターを設定する必要があります。

前提条件

- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - Credentials secret (**CREDENTIALS_SECRET**): 「管理ユーザーのシークレットの作成」で作成される管理ユーザーの認証情報を含むシークレットの名前。
   - KIE Server Keystore Secret Name(**KIE_SERVER_HTTPS_SECRET**): 「KIE Server のシークレットの作成」で作成した KIE Server のシークレットの名前。
   - KIE Server Certificate Name(**KIE_SERVER_HTTPS_NAME**): 「KIE Server のシークレットの作成」で作成したキーストアの証明書名。
   - KIE Server Keystore Password (**KIE_SERVER_HTTPS_PASSWORD**): 「KIE Server のシークレットの作成」で作成したキーストアのパスワード。

- **アプリケーション名** (**APPLICATION_NAME**): OpenShift アプリケーションの名前。これは、Business Central Monitoring および KIE Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名を同じにすることはできません。また、アプリケーション名は、KIE Server が Business Central で参加するサーバーの設定 (サーバーテンプレート) の名前を決定するものとなります。複数の KIE Server をデプロイしている場合は、それぞれのサーバーに異なるアプリケーション名があることを確認する必要があります。

- **KIE Server モード** (**KIE_SERVER_MODE**): **rhdm79-kieserver.yaml** テンプレートで、デフォルト値は **PRODUCTION** です。**PRODUCTION** モードでは、**SNAPSHOT** バージョンの KJAR アーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。**PRODUCTION** モードで新規バージョンをデプロイするには、同じ KIE Server で新規コンテナーを作成します。**SNAPSHOT** バージョンをデプロイするか、または既存コンテナーのアーティファクトのバージョンを変更するには、このパラメーターを **DEVELOPMENT** に設定します。

- **ImageStream 名前空間** (**IMAGE_STREAM_NAMESPACE**): イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「イメージストリームとイメージレジストリーの可用性確認」 を参照) は、名前空間が **openshift** になります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

8.3.3. 追加の管理 KIE Server のイメージストリーム namespace の設定

openshift ではない名前空間でイメージストリームを作成した場合は、テンプレートで名前空間を設定する必要があります。

すべてのイメージストリームが Red Hat OpenShift Container Platform 環境ですでに利用可能な場合は、この手順を省略できます。

前提条件

- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

「イメージストリームとイメージレジストリーの可用性確認」 の説明に従ってイメージストリームファイルをインストールした場合は、**ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) パラメーターを OpenShift プロジェクトの名前に設定します。

8.3.4. 追加の管理 KIE Server 用の Business Central インスタンスについての情報の設定

同じ名前空間で Business Central インスタンスから KIE Server への接続を有効にする場合は、Business Central インスタンスに関する情報を設定する必要があります。

Business Central インスタンスは、KIE Server と同じ認証情報シークレット (**CREDENTIALS_SECRET**) を使用して設定する必要があります。

前提条件

- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - **Business Central サービスの名前** (**DECISION_CENTRAL_SERVICE**): Business Central の OpenShift サービス名。
2. サーバーがサービスを読み込むに使用する Maven リポジトリーへのアクセスを設定します。Business Central が使用するものと同じリポジトリーを設定する必要があります。
   - Business Central が独自のビルトインリポジトリーを使用する場合は、以下のパラメーターを設定します。
     - **Business Central の Maven サービスの名前** (**DECISION_CENTRAL_MAVEN_SERVICE**): Business Central の OpenShift サービス名。
   - Business Central を外部 Maven リポジトリーを使用するように設定している場合は、以下のパラメーターを設定します。
     - **Maven リポジトリーの URL** (**MAVEN_REPO_URL**): Business Central が使用する外部 Maven リポジトリーの URL。
     - **Maven リポジトリーの ID** (**MAVEN_REPO_ID**): Maven リポジトリーの ID。デフォルト値は **repo-custom** です。
     - **Maven リポジトリーのユーザー名** (**MAVEN_REPO_USERNAME**): Maven リポジトリーのユーザー名。
     - **Maven リポジトリーのパスワード** (**MAVEN_REPO_PASSWORD**): Maven リポジトリーのパスワード。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

8.3.5. 追加の管理 KIE Server の公開インターネットへの接続のない環境に Maven ミラーへのアクセスを設定する

テンプレートを追加の管理 KIE Server をデプロイするように設定する際に、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」 に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Maven ミラーへのアクセスを設定するには、以下のパラメーターを設定します。

- **Maven ミラー URL** (**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」 で設定した 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** (**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。

  - ビルトイン Business Central Maven リポジトリー (**DECISION_CENTRAL_MAVEN_SERVICE**) を設定する場合は、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-rhdmcentr**)。

  - 両リポジトリーを設定した場合は、ミラーから両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-rhdmcentr,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

8.3.6. 追加の管理 KIE Server の RH-SSO 認証パラメーターの設定

RH-SSO 認証を使用する必要がある場合は、管理 KIE Server をデプロイするようにテンプレートを設定する際に追加の設定を実行します。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- Red Hat Decision Manager のレルムが RH-SSO 認証システムに作成されている。

- Red Hat Decision Manager のユーザー名およびパスワードが RH-SSO 認証システムに作成されている。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。

- デプロイしている Red Hat Decision Manager 環境の全コンポーネントに対して、クライアントが RH-SSO 認証システムに作成されている。クライアントのセットアップには、コンポーネントの URL が含まれます。環境のデプロイ後に URL を確認し、編集できます。または、Red Hat Decision Manager のデプロイメントでクライアントを作成できます。ただし、このオプションの環境に対する制御の詳細度合はより低くなります。
- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - **RH-SSO URL** (**SSO_URL**): RH-SSO の URL。
   - **RH-SSO レルム名** (**SSO_REALM**): Red Hat Decision Manager の RH-SSO レルム。
   - **RH-SSO が無効な SSL 証明書の検証** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): RH-SSO インストールで有効な HTTPS 証明書を使用していない場合は **true** に設定します。
2. 以下の手順のいずれかを実行します。
   a. RH-SSO で Red Hat Decision Manager のクライアントを作成した場合は、テンプレートで以下のパラメーターを設定します。
      - **Business Central RH-SSO クライアント名** (**DECISION_CENTRAL_SSO_CLIENT**): Business Central の RH-SSO クライアント名。
      - **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server の RH-SSO クライアント名。
      - **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
   b. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。
      - **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server 向けに RH-SSO に作成するクライアント名。
      - **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
      - **RH-SSO レルムの管理者のユーザー名** (**SSO_USERNAME**) および **RH-SSO レルムの管理者のパスワード** (**SSO_PASSWORD**): Red Hat Decision Manager の RH-SSO レルムの管理者ユーザーに指定するユーザー名とパスワード必要なクライアントを作成するためにこのユーザー名およびパスワードを指定する必要があります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

デプロイの完了後に、RH-SSO 認証システムで Red Hat Decision Manager のコンポーネントの URL が正しいことを確認してください。

8.3.7. 追加の管理 KIE Server の LDAP 認証パラメーターの設定

LDAP 認証を使用する必要がある場合は、テンプレートを追加の管理 KIE Server をデプロイするように設定する際に追加の設定を実行します。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- LDAP システムに Red Hat Decision Manager のユーザー名およびパスワードを作成している。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
  「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. テンプレートの **AUTH_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 **LdapExtended** ログインモジュールの設定に対応します。これらの設定に関する説明は、LdapExtended login module 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。

   LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。

   - **RoleMapping rolesProperties ファイルパス** (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties** など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「(任意) LDAP ロールマッピングファイルの指定」を参照してください。
   - **RoleMapping replaceRole プロパティー** (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): **true** に設定した場合、マッピングしたロールは、LDAP サーバーに定義したロールに置き換えられます。**false** に設定した場合は、LDAP サーバーに定義したロールと、マッピングしたロールの両方がユーザーアプリケーションロールとして設定されます。デフォルトの設定は **false** です。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

8.3.8. 追加の管理 KIE Server の Prometheus メトリクス収集の有効化

KIE Server デプロイメントを Prometheus を使用してメトリクスを収集し、保存するように設定する必要がある場合、デプロイ時に KIE Server でこの機能のサポートを有効にします。

前提条件

- 「追加の管理 KIE Server テンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Prometheus メトリクス収集のサポートを有効にするには、**Prometheus Server 拡張無効** (**PROMETHEUS_SERVER_EXT_DISABLED**) パラメーターを **false** に設定します。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「追加の管理 KIE Server テンプレートデプロイの実行」 の手順に従います。

Prometheus メトリクス収集の方法は、**KIE Server の管理および監視** を参照してください。

8.3.9. 追加の管理 KIE Server テンプレートデプロイの実行

OpenShift Web UI またはコマンドラインで必要なすべてのパラメーターを設定した後に、テンプレートのデプロイを実行します。

手順

使用している方法に応じて、以下の手順を実行します。

- OpenShift Web UI の場合は **Create** をクリックします。
  - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。
- コマンドラインに入力して、Enter キーを押します。

次のステップ

環境の要件に応じて、任意で 10章 *環境をデプロイした後の任意の手順* で説明されている手順を完了します。

第9章 イミュータブルサーバーを使用した環境

事前定義プロセスを使用して **イミュータブル** KIE Server を実行する 1 つ以上の Pod を含む環境をデプロイできます。KIE Server の各 Pod は、必要に応じて個別にスケーリングできます。

イメージの作成時に、イミュータブル KIE Server ですべてのサービスをサーバーに読み込む必要があります。実行中のイミュータブル KIE Server でサービスのデプロイまたはデプロイ解除を行うことはできません。このアプローチの利点は、サービスが含まれる KIE Server はコンテナー化されたサービスのように実行され、特別な管理を必要としない点にあります。KIE Server は OpenShift 環境で 1 つの Pod のように実行されます。必要に応じて、コンテナーベースの統合ワークフローを使用できます。

KIE Server イメージを作成する場合は、S2I (Source to Image) を使用してサービスをビルドする必要があります。サービスのソースおよびその他のビジネスアセットを使用して Git リポジトリーを提供します。Business Central でサービスまたはアセットを開発する場合は、S2I ビルドの個別のリポジトリーにソースをコピーします。OpenShift は自動的にソースをビルドし、KIE Server イメージにサービスをインストールして、このサービスでコンテナーを起動します。

オーサリングサービスに Business Central を使用する場合は、プロセスのソースを展開して、S2I ビルドで使用する別の 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。

または、KJAR ファイルとしてすでにビルドされているサービスを使用して同様の KIE Server デプロイメントを作成できます。この場合、サービスを Maven リポジトリーに指定する必要があります。Business Central のビルトインリポジトリーまたは独自のリポジトリーを使用できます (例: Nexus デプロイメント)。サーバー Pod が起動すると、これは KJAR サービスを Maven リポジトリーから取得します。Pod 上のサービスが更新したり、変更することはありません。Pod の毎回の再起動またはスケーリング時に、サーバーはリポジトリーからファイルを取得するため、デプロイメントをイミュータブルに保つには、それらのファイルが Maven リポジトリーで変更されないようにする必要があります。

イミュータブルのイメージを作成する方法はいずれも、イメージの管理が必要ありません。サービスの新規バージョンを使用する場合は、新規イメージをビルドできます。

9.1. S2I ビルドの使用によるイミュータブル KIE SERVER のデプロイ

S2I ビルドを使用してイミュータブル KIE Server をデプロイできます。サーバーをデプロイする際、デプロイメント手順ではこのサーバーで実行される必要のあるすべてのサービスのソースコードを取得し、サービスをビルドし、それらをサービスイメージに組み込みます。

実行中のイミュータブル KIE Server でサービスのデプロイまたはデプロイ解除を行うことはできません。Business Central を使用すると、モニター情報を表示できます。KIE Server は OpenShift 環境で 1 つの Pod のように実行されます。必要に応じて、コンテナーベースの統合ワークフローを使用できます。

イミュータブル KIE Server の JMS 機能を有効にできます。JMS 機能を使用すると、外部 AMQ メッセージブローカーを使用し、JMS API 経由でサーバーと対話できます。

Business Central が同じ名前空間にデプロイされる場合、これはイミュータブル KIE Server を自動的に検出します。Business Central を使用してイミュータブル KIE Server でサービスの起動や停止が可能です (ただしデプロイはできません)。

9.1.1. S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始

S2I ビルドを使用してイミュータブル KIE Server をデプロイするには、JMS 機能を有効にする必要がある場合は **rhdm79-prod-immutable-kieserver-amq.yaml** テンプレートファイルを使用します。そうでない場合は、**rhdm79-prod-immutable-kieserver.yaml** テンプレートファイルを使用します。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードします。
2. 必要なテンプレートファイルを展開します。
3. 以下のいずれかの方法を使用してテンプレートのデプロイを開始します。
   - OpenShift Web UI を使用するには、OpenShift アプリケーションコンソールで **Add to Project → Import YAML / JSON** を選択してから **<template-file-name>.yaml** ファイルを選択または貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
   - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

     ```
     oc new-app -f <template-path>/<template-file-name>.yaml -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
     ```

     このコマンドラインで、以下のように変更します。
     - **<template-path>** を、ダウンロードしたテンプレートファイルのパスに置き換えます。
     - **<template-file-name>** は、テンプレート名に置き換えます。
     - 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。

次のステップ

テンプレートのパラメーターを設定します。「S2I の使用によるイミュータブル KIE Server に必要なパラメーターの設定」 の手順に従い、共通のパラメーターを設定します。テンプレートファイルを表示して、すべてのパラメーターの説明を確認します。

9.1.2. S2I の使用によるイミュータブル KIE Server に必要なパラメーターの設定

テンプレートをイミュータブル KIE Server を S2I ビルドを使用してデプロイするように設定する際、いずれの場合でも以下のパラメーターを設定する必要があります。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - Credentials secret (**CREDENTIALS_SECRET**): 「管理ユーザーのシークレットの作成」で作成される管理ユーザーの認証情報を含むシークレットの名前。
   - KIE Server Keystore Secret Name(**KIE_SERVER_HTTPS_SECRET**): 「KIE Server のシークレットの作成」で作成した KIE Server のシークレットの名前。
   - KIE Server Certificate Name(**KIE_SERVER_HTTPS_NAME**): 「KIE Server のシークレットの作成」で作成したキーストアの証明書名。

- KIE Server Keystore Password (**KIE_SERVER_HTTPS_PASSWORD**): 「KIE Server のシークレットの作成」で作成したキーストアのパスワード。

- **アプリケーション名** (**APPLICATION_NAME**): OpenShift アプリケーションの名前。これは、Business Central Monitoring および KIE Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名を同じにすることはできません。また、アプリケーション名は、KIE Server が Business Central で参加するサーバーの設定 (サーバーテンプレート) の名前を決定するものとなります。複数の KIE Server をデプロイしている場合は、それぞれのサーバーに異なるアプリケーション名があることを確認する必要があります。

- **KIE Server コンテナーのデプロイメント** (**KIE_SERVER_CONTAINER_DEPLOYMENT**): ソースのビルド後にデプロイメントでローカルまたは外部リポジトリーからプルする必要のあるデシジョンサービス (KJAR ファイル) の ID 情報。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。

  ```
  containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2
  ```

  コンテナー ID の重複を防ぐには、アーティファクトビルドごとに、またはプロジェクト内で、一意のアーティファクト ID を指定する必要があります。

- **Git リポジトリー URL** (**SOURCE_REPOSITORY_URL**): サービスのソースを含む Git リポジトリーの URL。

- **Git 参照** (**SOURCE_REPOSITORY_REF**): 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** (**CONTEXT_DIR**): Git リポジトリーからダウンロードしたプロジェクト内のソースへのパス。

- **アーティファクトディレクトリー** (**ARTIFACT_DIR**): Maven のビルドに成功したあと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) を含むプロジェクトのパス。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。

- **ImageStream 名前空間** (**IMAGE_STREAM_NAMESPACE**): イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「イメージストリームとイメージレジストリーの可用性確認」 を参照) は、名前空間が **openshift** になります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.3. S2I の使用によるイミュータブル KIE Server のイメージストリーム namespace の設定

openshift ではない名前空間でイメージストリームを作成した場合は、テンプレートで名前空間を設定する必要があります。

すべてのイメージストリームが Red Hat OpenShift Container Platform 環境ですでに利用可能な場合は、この手順を省略できます。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

「イメージストリームとイメージレジストリーの可用性確認」 の説明に従ってイメージストリームファイルをインストールした場合は、**ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) パラメーターを OpenShift プロジェクトの名前に設定します。

9.1.4. S2I の使用によるイミュータブル KIE Server 用の Business Central インスタンスに関する情報の設定

同じ名前空間で Business Central インスタンスから KIE Server への接続を有効にする場合は、Business Central インスタンスに関する情報を設定する必要があります。

Business Central インスタンスは、KIE Server と同じ認証情報シークレット (**CREDENTIALS_SECRET**) を使用して設定する必要があります。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
    - **Business Central サービスの名前** (**DECISION_CENTRAL_SERVICE**): Business Central の OpenShift サービス名。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.5. S2I の使用によるイミュータブル KIE Server のオプションの Maven リポジトリーの設定

S2I ビルドを使用してテンプレートをイミュータブル KIE Server をデプロイするように設定する際に、ソースビルドに公開 Maven ツリーで利用可能ではない依存関係が含まれ、個別のカスタム Maven リポジトリーが必要な場合は、リポジトリーにアクセスできるようにパラメーターを設定する必要があります。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

カスタム Maven リポジトリーへのアクセスを設定するには、以下のパラメーターを設定します。

- **Maven リポジトリーの URL** (**MAVEN_REPO_URL**): Maven リポジトリーの URL。
- **Maven リポジトリーの ID** (**MAVEN_REPO_ID**): Maven リポジトリーの ID。デフォルト値は **repo-custom** です。
- **Maven リポジトリーのユーザー名** (**MAVEN_REPO_USERNAME**): Maven リポジトリーのユーザー名。
- **Maven リポジトリーのパスワード** (**MAVEN_REPO_PASSWORD**): Maven リポジトリーのパスワード。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.6. S2I の使用によるイミュータブル KIE Server の公開インターネットへの接続のない環境での Maven ミラーへのアクセスの設定

S2I ビルドを使用してテンプレートをイミュータブル KIE Server をデプロイするように設定する際に、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」 に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Maven ミラーへのアクセスを設定するには、以下のパラメーターを設定します。

- **Maven ミラー URL** (**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」 で設定した 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** (**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。
  - ビルトイン Business Central Maven リポジトリー (**DECISION_CENTRAL_MAVEN_SERVICE**) を設定する場合は、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-rhdmcentr)。

- 両リポジトリーを設定した場合は、ミラーから両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-rhdmcentr,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.7. S2I の使用によるイミュータブル KIE Server 用の AMQ サーバーとの通信の設定

rhdm79-prod-immutable-kieserver-amq.yaml テンプレートファイルを使用する場合は、KIE Server の JMS 機能が有効にされます。外部の AMQ メッセージブローカーを使用して、JMS API 経由でサーバーと対話できます。

環境に必要な場合は、JMS 設定を変更できます。

前提条件

- **rhdm79-prod-immutable-kieserver-amq.yaml** テンプレートファイルを使用して 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始している。

手順

必要に応じて、お使いの環境に以下のパラメーターのいずれかを設定します。

- **AMQ ユーザー名** (**AMQ_USERNAME**) および **AMQ パスワード** (**AMQ_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。
- **AMQ ロール** (**AMQ_ROLE**): 標準ブローカーユーザーのユーザーロール。デフォルトロールは **admin** です。
- **AMQ キュー** (**AMQ_QUEUES**): コンマで区切られた AMQ キュー名。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。カスタムのキュー名を使用する場合は、同じキュー名を **KIE_SERVER_JMS_QUEUE_RESPONSE** パラメーター、**KIE_SERVER_JMS_QUEUE_REQUEST** パラメーター、**KIE_SERVER_JMS_QUEUE_SIGNAL** パラメーター、**KIE_SERVER_JMS_QUEUE_AUDIT** パラメーター、および **KIE_SERVER_JMS_QUEUE_EXECUTOR** パラメーターに設定する必要もあります。
- **AMQ グローバル最大サイズ** (**AMQ_GLOBAL_MAX_SIZE**): メッセージデータが消費できるメモリーの最大量。値が指定されない場合は、Pod で利用可能なメモリーの半分が割り当てられます。
- **AMQ プロトコル** (**AMQ_PROTOCOL**): コンマで区切られた、KIE Server が AMQ サーバーとの通信に使用できるブローカーのプロトコル。許可される値は、**openwire**、**amqp**、**stomp**、および **mqtt** です。**openwire** のみが JBoss EAP でサポートされます。デフォルト値は **openwire** です。

- **AMQ ブローカーイメージ** (**AMQ_BROKER_IMAGESTREAM_NAME**): AMQ ブローカーイメージのイメージストリーム名。

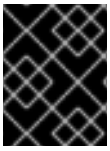
次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.8. S2I の使用によるイミュータブル KIE Server の RH-SSO 認証パラメーターの設定

RH-SSO 認証を使用する必要がある場合は、テンプレートを S2I ビルドを使用してイミュータブル KIE Server をデプロイするように設定する際に追加の設定を実行します。



重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- Red Hat Decision Manager のレルムが RH-SSO 認証システムに作成されている。
- Red Hat Decision Manager のユーザー名およびパスワードが RH-SSO 認証システムに作成されている。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- デプロイしている Red Hat Decision Manager 環境の全コンポーネントに対して、クライアントが RH-SSO 認証システムに作成されている。クライアントのセットアップには、コンポーネントの URL が含まれます。環境のデプロイ後に URL を確認し、編集できます。または、Red Hat Decision Manager のデプロイメントでクライアントを作成できます。ただし、このオプションの環境に対する制御の詳細度合はより低くなります。
- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
    - **RH-SSO URL** (**SSO_URL**): RH-SSO の URL。
    - **RH-SSO レルム名** (**SSO_REALM**): Red Hat Decision Manager の RH-SSO レルム。
    - **RH-SSO が無効な SSL 証明書の検証** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): RH-SSO インストールで有効な HTTPS 証明書を使用していない場合は **true** に設定します。
2. 以下の手順のいずれかを実行します。
    a. RH-SSO で Red Hat Decision Manager のクライアントを作成した場合は、テンプレートで以下のパラメーターを設定します。

- **Business Central RH-SSO クライアント名** (**DECISION_CENTRAL_SSO_CLIENT**): Business Central の RH-SSO クライアント名。
- **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server の RH-SSO クライアント名。
- **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。

b. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。

- **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server 向けに RH-SSO に作成するクライアント名。
- **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
- **RH-SSO レルムの管理者のユーザー名** (**SSO_USERNAME**) および **RH-SSO レルムの管理者のパスワード** (**SSO_PASSWORD**): Red Hat Decision Manager の RH-SSO レルムの管理者ユーザーに指定するユーザー名とパスワード必要なクライアントを作成するためにこのユーザー名およびパスワードを指定する必要があります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

デプロイの完了後に、RH-SSO 認証システムで Red Hat Decision Manager のコンポーネントの URL が正しいことを確認してください。

9.1.9. S2I の使用によるイミュータブル KIE Server の LDAP 認証パラメーターの設定

LDAP 認証を使用する場合は、S2I ビルドを使用してイミュータブル KIE Server をデプロイするテンプレートを設定する時に、以下の追加設定を行います。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- LDAP システムに Red Hat Decision Manager のユーザー名およびパスワードを作成している。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. テンプレートの **AUTH_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 **LdapExtended** ログインモジュールの設定に対応します。これらの設定に関する説明は、LdapExtended login module 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。

LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。

- RoleMapping rolesProperties ファイルパス (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties** など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「(任意) LDAP ロールマッピングファイルの指定」を参照してください。

- RoleMapping replaceRole プロパティー (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): **true** に設定した場合、マッピングしたロールは、LDAP サーバーに定義したロールに置き換えられます。**false** に設定した場合は、LDAP サーバーに定義したロールと、マッピングしたロールの両方がユーザーアプリケーションロールとして設定されます。デフォルトの設定は **false** です。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

9.1.10. S2I の使用によるイミュータブル KIE Server の Prometheus メトリクス収集の有効化

KIE Server デプロイメントを Prometheus を使用してメトリクスを収集し、保存するように設定する必要がある場合、デプロイ時に KIE Server でこの機能のサポートを有効にします。

前提条件

- 「S2I の使用によるイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Prometheus メトリクス収集のサポートを有効にするには、**Prometheus Server 拡張無効** (**PROMETHEUS_SERVER_EXT_DISABLED**) パラメーターを **false** に設定します。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行」 の手順に従います。

Prometheus メトリクス収集の方法は、**KIE Server の管理および監視** を参照してください。

9.1.11. S2I の使用によるイミュータブル KIE Server テンプレートのデプロイの実行

OpenShift Web UI またはコマンドラインで必要なすべてのパラメーターを設定した後に、テンプレートのデプロイを実行します。

手順

使用している方法に応じて、以下の手順を実行します。

- OpenShift Web UI の場合は **Create** をクリックします。
  - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。
- コマンドラインに入力して、Enter キーを押します。

次のステップ

環境の要件に応じて、任意で 10章 *環境をデプロイした後の任意の手順* で説明されている手順を完了します。

9.2. KJAR サービスからのイミュータブル KIE SERVER のデプロイ

KJAR ファイルとしてすでにビルドされているサービスを使用して、イミュータブル KIE Server をデプロイできます。

サービスを Maven リポジトリーに指定する必要があります。Business Central のビルトインリポジトリーまたは独自のリポジトリーを使用できます (例: Nexus デプロイメント)。サーバー Pod が起動すると、これは KJAR サービスを Maven リポジトリーから取得します。Pod 上のサービスが更新したり、変更することはありません。Pod の毎回の再起動またはスケーリング時に、サーバーはリポジトリーからファイルを取得するため、デプロイメントをイミュータブルに保つには、それらのファイルが Maven リポジトリーで変更されないようにする必要があります。

実行中のイミュータブル KIE Server でサービスのデプロイまたはデプロイ解除を行うことはできません。Business Central を使用すると、モニター情報を表示できます。KIE Server は OpenShift 環境で 1 つの Pod のように実行されます。必要に応じて、コンテナーベースの統合ワークフローを使用できます。

Business Central が同じ名前空間にデプロイされる場合、これはイミュータブル KIE Server を自動的に検出します。Business Central を使用してイミュータブル KIE Server でサービスを起動および停止を実行でき (ただしデプロイはできません)、モニターデータを表示できます。

9.2.1. KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始

KJAR サービスからイミュータブル KIE Server をデプロイするには、**rhdm79-kieserver.yaml** テンプレートファイルを使用します。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードします。
2. **rhdm79-kieserver.yaml** テンプレートファイルを展開します。

3. 以下のいずれかの方法を使用してテンプレートのデプロイを開始します。

- OpenShift Web UI を使用するには、OpenShift アプリケーションコンソールで **Add to Project → Import YAML / JSON** を選択してから、**rhdm79-kieserver.yaml** ファイルを選択するか、またはこれを貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
- OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/rhdm79-kieserver.yaml -p
KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
```

このコマンドラインで、以下のように変更します。

- **<template-path>** を、ダウンロードしたテンプレートファイルのパスに置き換えます。
- 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。

次のステップ

テンプレートのパラメーターを設定します。「KJAR サービスからのイミュータブル KIE Server の必須パラメーターの設定」の手順に従い、共通のパラメーターを設定します。テンプレートファイルを表示して、すべてのパラメーターの説明を確認します。

9.2.2. KJAR サービスからのイミュータブル KIE Server の必須パラメーターの設定

テンプレートをイミュータブル KIE Server を KJAR サービスからデプロイするように設定する際、いずれの場合でも以下のパラメーターを設定する必要があります。

前提条件

- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。

- **Credentials secret** (**CREDENTIALS_SECRET**): 「管理ユーザーのシークレットの作成」で作成される管理ユーザーの認証情報を含むシークレットの名前。
- **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): 「KIE Server のシークレットの作成」で作成した KIE Server のシークレットの名前。
- **KIE Server Certificate Name**(**KIE_SERVER_HTTPS_NAME**): 「KIE Server のシークレットの作成」で作成したキーストアの証明書名。
- **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): 「KIE Server のシークレットの作成」で作成したキーストアのパスワード。
- **アプリケーション名** (**APPLICATION_NAME**): OpenShift アプリケーションの名前。これは、Business Central Monitoring および KIE Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。同じテンプレートを同じ

プロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名を同じにすることはできません。また、アプリケーション名は、KIE Server が Business Central で参加するサーバーの設定 (サーバーテンプレート) の名前を決定するものとなります。複数の KIE Server をデプロイしている場合は、それぞれのサーバーに異なるアプリケーション名があることを確認する必要があります。

- **Maven repository URL** (**MAVEN_REPO_URL**): Maven リポジトリーの URL。KIE Server にデプロイするすべてのプロセス (KJAR ファイル) をこのリポジトリーにアップロードする必要があります。
- **Maven リポジトリーの ID** (**MAVEN_REPO_ID**): Maven リポジトリーの ID。デフォルト値は **repo-custom** です。
- **Maven リポジトリーのユーザー名** (**MAVEN_REPO_USERNAME**): Maven リポジトリーのユーザー名。
- **Maven リポジトリーのパスワード** (**MAVEN_REPO_PASSWORD**): Maven リポジトリーのパスワード。
- **KIE Server コンテナーのデプロイメント** (**KIE_SERVER_CONTAINER_DEPLOYMENT**): 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。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。

  ```
  containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2
  ```

- **KIE Server モード** (**KIE_SERVER_MODE**): **rhdm79-kieserver-*.yaml** テンプレートで、デフォルト値は **PRODUCTION** です。**PRODUCTION** モードでは、**SNAPSHOT** バージョンの KJAR アーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。**PRODUCTION** モードで新規バージョンをデプロイするには、同じ KIE Server で新規コンテナーを作成します。**SNAPSHOT** バージョンをデプロイするか、または既存コンテナーのアーティファクトのバージョンを変更するには、このパラメーターを **DEVELOPMENT** に設定します。
- **ImageStream 名前空間** (**IMAGE_STREAM_NAMESPACE**): イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「イメージストリームとイメージレジストリーの可用性確認」 を参照) は、名前空間が **openshift** になります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」 の手順に従います。

9.2.3. イミュータブル KIE Server のイメージストリーム namespace の設定

openshift ではない名前空間でイメージストリームを作成した場合は、テンプレートで名前空間を設定する必要があります。

すべてのイメージストリームが Red Hat OpenShift Container Platform 環境ですでに利用可能な場合は、この手順を省略できます。

前提条件

- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

「イメージストリームとイメージレジストリーの可用性確認」 の説明に従ってイメージストリームファイルをインストールした場合は、**ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) パラメーターを OpenShift プロジェクトの名前に設定します。

9.2.4. KJAR サービスを使用したイミュータブル KIE Server 用の Business Central インスタンスに関する情報の設定

同じ名前空間で Business Central インスタンスから KIE Server への接続を有効にする場合は、Business Central インスタンスに関する情報を設定する必要があります。

Business Central インスタンスは、KIE Server と同じ認証情報シークレット (**CREDENTIALS_SECRET**) を使用して設定する必要があります。

前提条件

- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - **Business Central サービスの名前** (**DECISION_CENTRAL_SERVICE**): Business Central の OpenShift サービス名。
2. 以下の設定が Business Central の同じ設定と同じ値に設定されていることを確認します。
   - **Maven リポジトリーの URL** (**MAVEN_REPO_URL**): サービスのデプロイに使用する必要のある外部 Maven リポジトリーの URL。
   - **Maven リポジトリーのユーザー名** (**MAVEN_REPO_USERNAME**): Maven リポジトリーのユーザー名。
   - **Maven リポジトリーのパスワード** (**MAVEN_REPO_PASSWORD**): Maven リポジトリーのパスワード。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」 の手順に従います。

9.2.5. KJAR サービスを使用したイミュータブル KIE Server の公開インターネットへの接続のない環境での Maven ミラーへのアクセスの設定

KJAR サービスを使用してテンプレートをイミュータブル KIE Server をデプロイするように設定する際に、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」 に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Maven ミラーへのアクセスを設定するには、以下のパラメーターを設定します。

- **Maven ミラー URL** (**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」 で設定した 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** (**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。
  - ビルトイン Business Central Maven リポジトリー (**DECISION_CENTRAL_MAVEN_SERVICE**) を設定する場合は、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-rhdmcentr**)。
  - 両リポジトリーを設定した場合は、ミラーから両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-rhdmcentr,!repo-custom**)。**repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」 の手順に従います。

9.2.6. KJAR サービスの使用によるイミュータブル KIE Server の RH-SSO 認証パラメーターの設定

RH-SSO 認証を使用する必要がある場合は、テンプレートを KJAR サービスを使用してイミュータブル KIE Server をデプロイするように設定する際に追加の設定を実行します。

重要

LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- Red Hat Decision Manager のレルムが RH-SSO 認証システムに作成されている。
- Red Hat Decision Manager のユーザー名およびパスワードが RH-SSO 認証システムに作成されている。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。

「管理ユーザーのシークレットの作成」 で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。

- デプロイしている Red Hat Decision Manager 環境の全コンポーネントに対して、クライアントが RH-SSO 認証システムに作成されている。クライアントのセットアップには、コンポーネントの URL が含まれます。環境のデプロイ後に URL を確認し、編集できます。または、Red Hat Decision Manager のデプロイメントでクライアントを作成できます。ただし、このオプションの環境に対する制御の詳細度合はより低くなります。
- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。
   - **RH-SSO URL** (**SSO_URL**): RH-SSO の URL。
   - **RH-SSO レルム名** (**SSO_REALM**): Red Hat Decision Manager の RH-SSO レルム。
   - **RH-SSO が無効な SSL 証明書の検証** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): RH-SSO インストールで有効な HTTPS 証明書を使用していない場合は **true** に設定します。
2. 以下の手順のいずれかを実行します。
   a. RH-SSO で Red Hat Decision Manager のクライアントを作成した場合は、テンプレートで以下のパラメーターを設定します。
      - **Business Central RH-SSO クライアント名** (**DECISION_CENTRAL_SSO_CLIENT**): Business Central の RH-SSO クライアント名。
      - **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server の RH-SSO クライアント名。
      - **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
   b. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。
      - **KIE Server RH-SSO クライアント名** (**KIE_SERVER_SSO_CLIENT**): KIE Server 向けに RH-SSO に作成するクライアント名。
      - **KIE Server RH-SSO クライアントのシークレット** (**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
      - **RH-SSO レルムの管理者のユーザー名** (**SSO_USERNAME**) および **RH-SSO レルムの管理者のパスワード** (**SSO_PASSWORD**): Red Hat Decision Manager の RH-SSO レルムの管理者ユーザーに指定するユーザー名とパスワード必要なクライアントを作成するためにこのユーザー名およびパスワードを指定する必要があります。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」の手順に従います。

デプロイの完了後に、RH-SSO 認証システムで Red Hat Decision Manager のコンポーネントの URL が正しいことを確認してください。

9.2.7. KJAR サービスの使用によるイミュータブル KIE Server の LDAP 認証パラメーターの設定

LDAP 認証を使用する必要がある場合は、KJAR サービスからイミュータブル KIE Server をデプロイするようにテンプレートを設定するには、以下のように追加で設定を行います。

重要

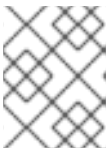
LDAP 認証および RH-SSO 認証を同じデプロイメントに設定しないようにしてください。

前提条件

- LDAP システムに Red Hat Decision Manager のユーザー名およびパスワードを作成している。利用可能なロールの一覧については、11章*Red Hat Decision Manager ロールおよびユーザー* を参照してください。
「管理ユーザーのシークレットの作成」で説明されているように、管理ユーザーのシークレットで設定されたユーザー名およびパスワードを使用してユーザーを作成する必要があります。このユーザーには **kie-server,rest-all,admin** ロールが必要です。
- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」に説明されているテンプレートの設定を開始していること。

手順

1. テンプレートの **AUTH_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 **LdapExtended** ログインモジュールの設定に対応します。これらの設定に関する説明は、LdapExtended login module 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。

LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。

- **RoleMapping rolesProperties ファイルパス** (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties** など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「(任意) LDAP ロールマッピングファイルの指定」を参照してください。
- **RoleMapping replaceRole プロパティー** (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): **true** に設定した場合、マッピングしたロールは、LDAP サーバーに定義したロールに置き換えられます。**false** に設定した場合は、LDAP サーバーに定義したロールと、マッピング

したロールの両方がユーザーアプリケーションロールとして設定されます。デフォルトの設定は **false** です。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」 の手順に従います。

9.2.8. KJAR サービスの使用によるイミュータブル KIE Server からの Prometheus メトリクス収集の有効化

KIE Server デプロイメントを Prometheus を使用してメトリクスを収集し、保存するように設定する必要がある場合、デプロイ時に KIE Server でこの機能のサポートを有効にします。

前提条件

- 「KJAR サービスでのイミュータブル KIE Server のテンプレート設定の開始」 に説明されているテンプレートの設定を開始していること。

手順

Prometheus メトリクス収集のサポートを有効にするには、**Prometheus Server 拡張無効** (**PROMETHEUS_SERVER_EXT_DISABLED**) パラメーターを **false** に設定します。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行」 の手順に従います。

Prometheus メトリクス収集の方法は、**KIE Server の管理および監視** を参照してください。

9.2.9. KJAR サービスの使用によるイミュータブル KIE Server テンプレートデプロイの実行

OpenShift Web UI またはコマンドラインで必要なすべてのパラメーターを設定した後に、テンプレートのデプロイを実行します。

手順

使用している方法に応じて、以下の手順を実行します。

- OpenShift Web UI の場合は **Create** をクリックします。
  - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。
- コマンドラインに入力して、Enter キーを押します。

次のステップ

環境の要件に応じて、任意で *10章 環境をデプロイした後の任意の手順* で説明されている手順を完了します。

第10章 環境をデプロイした後の任意の手順

環境の要件によっては、デプロイ後に任意の手順を完了しないといけない場合があります。

10.1. (オプション) GIT フックディレクトリーの指定

オーサリング環境をデプロイして **GIT_HOOKS_DIR** パラメーターを設定した場合は、Git フックのディレクトリーを指定して、Business Central デプロイメントにこのディレクトリーをマウントする必要があります。

Git フックは一般的に、アップストリームのリポジトリーとの対話に使用します。Git フックを使用して、アップストリームのリポジトリーにコミットをプッシュできるようにするには、アップストリームのリポジトリーで設定した公開鍵に対応する秘密鍵を指定する必要があります。

前提条件

- テンプレートを使用して Red Hat Decision Manager のオーサリング環境をデプロイしている。
- デプロイメントに **GIT_HOOKS_DIR** パラメーターを設定している。

手順

1. SSH 認証を使用してアップストリームリポジトリーを操作する必要がある場合は、次の手順を実行して、必要なファイルを含むシークレットを作成してマウントします。
   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
   ```

   d. 以下の例では、Business Central デプロイメントの ssh キーパスにこのシークレットをマウントします。

   ```
   oc set volume dc/<myapp>-rhdmcentr --add --type secret --secret-name git-hooks-secret --mount-path=/home/jboss/.ssh --name=ssh-key
   ```

   <myapp> をテンプレートの設定時に設定したアプリケーション名に置き換えます。
2. Git フックディレクトリーを作成します。方法は、Git hooks reference documentation を参照してください。
   たとえば、単純な Git フックディレクトリーで、変更をアップストリームにプッシュする post-commit フックを指定できます。プロジェクトがリポジトリーから Business Central にインポートされた場合、このリポジトリーはアップストリームリポジトリーとして設定されたままになります。パーミッションを **755** の値に指定し、以下の内容を含めて **post-commit** という名前のファイルを作成します。

   ```
   git push
   ```

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。

3. Git フックディレクトリーを Business Central デプロイメントに指定します。設定マップまたは永続ボリュームを使用できます。

   a. Git フックに 1 つまたは複数の固定スクリプトファイルが含まれる場合は、設定マップを使用します。以下の手順を実行してください。

      i. 作成した Git フックディレクトリーに移動します。

      ii. ディレクトリーのファイルから OpenShift 設定マップを作成します。次のコマンドを実行します。

      ```
      oc create configmap git-hooks --from-file=<file_1>=<file_1> --from-file=<file_2>=<file_2> ...
      ```

      file_1、**file_2** などは、Git フックのスクリプトファイル名に置き換えます。以下に例を示します。

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

      iii. Business Central デプロイメントの設定したパスに設定マップをマウントします。

      ```
      oc set volume dc/<myapp>-rhdmcentr --add --type configmap --configmap-name git-hooks  --mount-path=<git_hooks_dir> --name=git-hooks
      ```

      <myapp> をテンプレートの設定時に設定したアプリケーション名に、**<git_hooks_dir>** はテンプレート設定時に設定した **GIT_HOOKS_DIR** の値に置き換えます。

   b. Git フックが長いファイルで設定されているか、または実行可能なファイルや KJAR 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。永続ボリュームを作成し、永続ボリューム要求を作成してボリュームを要求に関連付け、ファイルをボリュームに転送し、ボリュームを **myapp-rhdmcentr** デプロイメント設定にマウントする必要があります (**myapp**をアプリケーション名に置き換えます)。永続ボリュームの作成およびマウント方法は、永続ボリュームの使用 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。

4. 数分待機してから、プロジェクト内の Pod の一覧およびステータスを確認します。Business Central は Git フックディレクトリーが指定されるまで開始されないため、KIE Server は全く起動されない可能性があります。Process Server が起動しているかどうかを確認するには、以下のコマンドの出力で確認します。

   ```
   oc get pods
   ```

   稼働中の KIE Server Pod がない場合には、これを起動します。

   ```
   oc rollout latest dc/<myapp>-kieserver
   ```

   <myapp> を、テンプレートの設定時に設定されたアプリケーション名に置き換えます。

10.2. (オプション) 自己署名証明書で HTTPS サーバーにアクセスするためのトラストストアの提供

Red Hat Decision Manager インフラストラクチャーのコンポーネントは、自己署名の HTTPS 証明書を使用するサーバーにアクセスするのに、HTTPS アクセスを使用する必要がある場合があります。たとえば、Business Central および KIE Server は、自己署名の HTTPS サーバー証明書を使用する内部の Nexus リポジトリーと対話する必要がある場合があります。

このような場合は、HTTPS 接続が正常に完了するようにするには、トラストストアを使用してこれらのサービスのクライアント証明書を指定する必要があります。

Red Hat Decision Manager のコンポーネントが自己署名の HTTPS サーバー証明書を使用するサーバーと通信する必要がない場合は、この手順を飛ばして次に進んでください。

前提条件

- テンプレートを使用して Red Hat Decision Manager 環境をデプロイしている。
- デプロイメントに追加するクライアント証明書がある。

手順

1. 対象の証明書を使用してトラストストアを準備します。次のコマンドを使用して、トラストストアを作成するか、証明書を既存のトラストストアに追加します。必要なすべての証明書を 1 つのトラストストアに追加します。

   ```
   keytool -importcert -file certificate-file -alias alias -keyalg algorithm -keysize size -
   trustcacerts -noprompt -storetype JKS -keypass truststore-password -storepass
   truststore-password -keystore keystore-file
   ```

   以下の値を置き換えます。

   - **certificate-file**: トラストストアに追加する証明書のパス名。
   - **alias**: トラストストアの証明書のエイリアス。トラストストアに複数の証明書を追加する場合は、全証明書に一意のエイリアスが必要です。
   - **algorithm**: 証明書に使用する暗号化アルゴリズム。通常は **RSA** です。
   - **size**: バイト単位での証明書キーの単位 (例: **2048**)。
   - **truststore-password**: トラストストアのパスワード。
   - **keystore-file**: トラストストアファイルのパス名。ファイルが存在しない場合には、このコマンドにより、新規トラストストアが作成されます。
     次のコマンド例は、**/var/certs/nexus.cer** ファイルから **/var/keystores/custom-trustore.jks** ファイルのトラストストアに証明書を追加します。トラストストアのパスワードは **mykeystorepass** です。

     ```
     keytool -importcert -file /var/certs/nexus.cer -alias nexus-cert -keyalg RSA -keysize 2048
     -trustcacerts -noprompt -storetype JKS -keypass mykeystorepass -storepass
     mykeystorepass -keystore /var/keystores/custom-trustore.jks
     ```

2. 以下のように **oc** コマンドを使用して、トラストストアファイルでシークレットを作成します。

```
oc create secret generic truststore-secret --from-file=/var/keystores/custom-trustore.jks
```

3. お使いのインフラストラクチャーに必要なコンポーネントをデプロイする場合は、以下の例のように、シークレットをマウントしてから **JAVA_OPTS_APPEND** オプションを設定して Java アプリケーションのインフラストラクチャーがトラストストアを使用できるようにします。

```
oc set volume dc/myapp-rhdmcentr --add --overwrite --name=custom-trustore-volume --
mount-path /etc/custom-secret-volume --secret-name=custom-secret

oc set env dc/myapp-rhdmcentr JAVA_OPTS_APPEND='-
Djavax.net.ssl.trustStore=/etc/custom-secret-volume/custom-trustore.jks -
Djavax.net.ssl.trustStoreType=jks -Djavax.net.ssl.trustStorePassword=mykeystorepass'
```

```
oc set volume dc/myapp-kieserver --add --overwrite --name=custom-trustore-volume --
mount-path /etc/custom-secret-volume --secret-name=custom-secret

oc set env dc/myapp-kieserver JAVA_OPTS_APPEND='-
Djavax.net.ssl.trustStore=/etc/custom-secret-volume/custom-trustore.jks -
Djavax.net.ssl.trustStoreType=jks -Djavax.net.ssl.trustStorePassword=mykeystorepass'
```

myapp をテンプレートの設定時に指定したアプリケーション名に置き換えます。

10.3. (任意) LDAP ロールマッピングファイルの指定

AUTH_ROLE_MAPPER_ROLES_PROPERTIES パラメーターを設定する場合は、ロールマッピングを定義するファイルを指定する必要があります。影響を受けるすべてのデプロイメント設定にこのファイルをマウントしてください。

前提条件

- テンプレートを使用して Red Hat Decision Manager 環境をデプロイしている。
- デプロイメントに **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** パラメーターを設定している。

手順

1. **my-role-map** など、ロールマッピングのプロパティーファイルを作成します。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
```

2. 以下のコマンドを入力して、このファイルから OpenShift 設定ファイルのマッピングを作成します。

```
oc create configmap ldap-role-mapping --from-file=<new_name>=<existing_name>
```

<new_name> は、Pod に指定するファイルの名前 (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES** ファイルで指定した名前と同じである必要が

あります) に置き換えます。また、**<existing_name>** は、作成したファイル名に置き換えます。以下に例を示します。

```
oc create configmap ldap-role-mapping --from-file=rolemapping.properties=my-role-map
```

3. ロールマッピング用に指定した全デプロイメント設定に設定マップをマウントします。以下のデプロイメント設定は、この環境で影響を受ける可能性があります。

   myapp はアプリケーション名に置き換えます。複数の KIE Server デプロイメントが異なるアプリケーション名で存在する場合があります。

   すべてのデプロイメント設定について、以下のコマンドを実行します。

```
oc set volume dc/<deployment_config_name> --add --type configmap --configmap-name
ldap-role-mapping --mount-path=<mapping_dir> --name=ldap-role-mapping
```

   <mapping_dir> は、**/opt/eap/standalone/configuration/rolemapping** など、**AUTH_ROLE_MAPPER_ROLES_PROPERTIES** で設定したディレクトリー名 (ファイル名なし) に置き換えます。

第11章 RED HAT DECISION MANAGER ロールおよびユーザー

Business Central または KIE Server にアクセスするには、サーバーを起動する前にユーザーを作成して適切なロールを割り当てます。

Business Central と KIE Server は、JAVA 認証承認サービス (JAAS) ログインモジュールを使用してユーザーを認証します。Business Central と KIE Server の両方が単一のインスタンスで実行されている場合は、同じ JAAS サブジェクトとセキュリティードメインを共有します。したがって、Business Central に対して認証されたユーザーは、KIE Server にもアクセスできます。

ただし、Business Central と KIE Server が異なるインスタンスで実行されている場合、JAAS ログインモジュールは両方に対して個別にトリガーされます。したがって、Business Central で認証されたユーザーは、KIE Server にアクセス (Business Central でプロセス定義を表示または管理など) するための個別認証が必要となります。ユーザーが KIE Server で認証されていない場合は、ログファイルに 401 エラーが記録され、Business Central に **Invalid credentials to load data from remote server.Contact your system administrator.** メッセージが表示されます。

本セクションでは、利用可能な Red Hat Decision Manager のユーザーロールを説明します。

注記

admin、**analyst**、および **rest-all** のロールは Business Central 用に予約されています。**kie-server** ロールは KIE Server 用に予約されています。このため、Business Central または KIE Server のいずれか、またはそれら両方がインストールされているかどうかによって、利用可能なロールは異なります。

- **admin**: **admin** ロールを持つユーザーは Business Central 管理者です。管理者は、ユーザーの管理や、リポジトリーの作成、クローン作成、および管理ができます。アプリケーションで必要な変更をすべて利用できます。**admin** ロールを持つユーザーは、Red Hat Decision Manager の全領域にアクセスできます。

- **analyst**: **analyst** ロールを持つユーザーには、すべてのハイレベル機能へのアクセスがあります。プロジェクトのモデル化が可能です。ただし、このユーザーは、**Design → Projects** ビューでスペースに貢献者を追加したり、スペースを削除したりできません。**analyst** ロールを持つユーザーは、管理者向けの **Deploy → Execution Servers** ビューにアクセスできません。ただし、これらのユーザーは、ライブラリーパースペクティブにアクセスするときに **Deploy** ボタンを使用できます。

- **rest-all**: **rest-all** ロールを持つユーザーは、Business Central REST 機能にアクセスできます。

- **kie-server**: **kie-server** ロールを持つユーザーは、KIE Server REST 機能へのアクセスがあります。

第12章 OPENSHIFT テンプレートの参考資料

Red Hat Decision Manager には、以下の OpenShift テンプレートが含まれています。このテンプレートにアクセスするには、Red Hat カスタマーポータルの Software Downloads ページから、製品の配信可能ファイル **rhdm-7.9.1-openshift-templates.zip** をダウンロードして展開します。

- **rhdm79-trial-ephemeral.yaml** は、Business Central および Business Central に接続された KIE Server を提供します。この環境では、永続ストレージのない一時的な設定を使用します。このテンプレートの詳細は、「rhdm79-trial-ephemeral.yaml template」 を参照してください。
- 「rhdm79-authoring.yaml template」 は、Business Central と Business Central に接続された KIE Server を提供します。この環境を使用して、サービスや他のビジネスアセットをオーサリングしたり、ステージングまたは実稼働環境でこれらのサービスを実行できます。このテンプレートの詳細は、「rhdm79-authoring.yaml template」 を参照してください。
- 「rhdm79-authoring-ha.yaml template」 は、高可用性 Business Central と Business Central に接続された KIE Server を提供します。この環境を使用して、サービスや他のビジネスアセットをオーサリングしたり、ステージングまたは実稼働環境でこれらのサービスを実行できます。このテンプレートの詳細は、「rhdm79-authoring-ha.yaml template」 を参照してください。
- 「rhdm79-kieserver.yaml template」 は KIE Server を提供します。KIE Server を Business Central に接続するように設定できます。これにより、1 つの Business Central が複数の別個の KIE Server を管理するステージングまたは実稼働環境をセットアップできます。このテンプレートの詳細は、「rhdm79-kieserver.yaml template」 を参照してください。
- 「rhdm79-prod-immutable-kieserver.yaml template」 で、イミュータブル KIE Server が設定されます。このテンプレートのデプロイメントには、KIE Server 上で実行予定の 1 つまたは複数サービスの source-to-image (S2I) ビルドが含まれます。このテンプレートの詳細は、「rhdm79-prod-immutable-kieserver.yaml template」 を参照してください。
- **rhdm79-prod-immutable-kieserver-amq.yaml** で、イミュータブル KIE Server が設定されます。このテンプレートのデプロイメントには、KIE Server 上で実行予定の 1 つまたは複数サービスの source-to-image (S2I) ビルドが含まれます。このバージョンのテンプレートには、JMS 統合が含まれます。このテンプレートの詳細は、「rhdm79-prod-immutable-kieserver-amq.yaml template」 を参照してください。

12.1. RHDM79-TRIAL-EPHEMERAL.YAML TEMPLATE

Red Hat Decision Manager 7.9 の一時オーサリングおよびテスト環境向けのアプリケーションテンプレート (非推奨)

12.1.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True

変数名	イメージの環境変数	説明	値の例	必須
DEFAULT_PASSWORD	**KIE_ADMIN_PWD**	試用版環境でユーザーが簡単に使用できるように用意された、複数コンポーネントに使用されるデフォルトのパスワード。	RedHat	True
KIE_ADMIN_USER	**KIE_ADMIN_USER**	KIE 管理者のユーザー名。	adminUser	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False
KIE_SERVER_MODE	**KIE_SERVER_MODE**	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**DEVELOPMENT**	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN	**AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE**	KIE Server の Access-Control-Allow-Origin 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	*	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS	**AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE**	KIE Server の Access-Control-Allow-Methods 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	GET、POST、OPTIONS、PUT	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS	**AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE**	KIE Server の Access-Control-Allow-Headers 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	Accept、Authorization、Content-Type、X-Requested-With	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS	**AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE**	KIE Server の Access-Control-Allow-Credentials 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	true	False
KIE_SERVER_ACCESS_CONTROL_MAX_AGE	**AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE**	KIE Server の Access-Control-Max-Age 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	1	False
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	false	False
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。 (org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	true	False
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL (ミリ秒単位)。 (org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	60000	False

変数名	イメージの環境変数	説明	値の例	必須
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	–	False
MAVEN_REPO_ID	**MAVEN_REPO_ID**	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	repo-custom	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_URL	**MAVEN_REPO_URL**	Maven リポジトリーまたはサービスへの完全修飾 URL。	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	git フックに使用するディレクトリー (必要な場合)。	**/opt/kie/data/git/hooks**	False
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central コンテナーのメモリー制限。	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限。	1Gi	False
SSO_URL	**SSO_URL**	RH-SSO URL。	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO クライアント名。	–	False
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報	パスワード	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール。	user	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.1.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.1.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-rhdmcentr	8080	http	Decision Central のすべての Web サーバーのポート。

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver	8080	–	すべての KIE Server Web サーバーのポート。

12.1.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-rhdmcentr-http	なし	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**

12.1.2.3. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.1.2.3.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-rhdmcentr	ImageChange
${APPLICATION_NAME}-kieserver	ImageChange

12.1.2.3.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-rhdmcentr	1
${APPLICATION_NAME}-kieserver	1

12.1.2.3.3. Pod テンプレート

12.1.2.3.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-rhdmcentr	**${APPLICATION_NAME}-rhdmsvc**
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-rhdmsvc**

12.1.2.3.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-rhdmcentr	rhdm-decisioncentral-rhel8
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.1.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

12.1.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.1.2.3.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-rhdmcentr	jolokia	8778	**TCP**
	http	8080	**TCP**
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**

12.1.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-rhdmcentr	**KIE_ADMIN_USER**	KIE 管理者のユーザー名。	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	試用版環境でユーザーが簡単に使用できるように用意された、複数コンポーネントに使用されるデフォルトのパスワード。	**${DEFAULT_PASSWORD}**
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	true
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	WORKBENCH_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-rhdmcentr
	MAVEN_REPO_ID	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	GIT_HOOKS_DIR	git フックに使用するディレクトリー (必要な場合)。	**${GIT_HOOKS_DIR}**
	KUBERNETES_NAMESPACE	–	–

デプロイメント	変数名	説明	値の例
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO クライアントシークレット。	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO クライアント名。	**${DECISION_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール。	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	KIE 管理者のユーザー名。	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	試用版環境でユーザーが簡単に使用できるように用意された、複数コンポーネントに使用されるデフォルトのパスワード。	**${DEFAULT_PASSWORD}**
	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**${KIE_SERVER_MODE}**

デプロイメント	変数名	説明	値の例
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL

デプロイメント	変数名	説明	値の例
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	KIE 管理者のユーザー名。	**${KIE_ADMIN_USER}**
	RHDMCENTR_MAVEN_REPO_PASSWORD	試用版環境でユーザーが簡単に使用できるように用意された、複数コンポーネントに使用されるデフォルトのパスワード。	**${DEFAULT_PASSWORD}**
	EXTERNAL_MAVEN_REPO_ID	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**

デプロイメント	変数名	説明	値の例
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット。	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール。	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
	FILTERS	–	AC_ALLOW_ORIGIN,AC_ALLOW_METHODS,AC_ALLOW_HEADERS,AC_ALLOW_CREDENTIALS,AC_MAX_AGE
	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Origin
	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE	KIE Server の Access-Control-Allow-Origin 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN}**
	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Methods
	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE	KIE Server の Access-Control-Allow-Methods 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS}**

デプロイメント	変数名	説明	値の例
	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Headers
	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE	KIE Server の Access-Control-Allow-Headers 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS}**
	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Credentials
	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE	KIE Server の Access-Control-Allow-Credentials 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS}**
	AC_MAX_AGE_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Max-Age
	AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE	KIE Server の Access-Control-Max-Age 応答ヘッダーの値を設定します (CORS サポートに役立ちます)。	**${KIE_SERVER_ACCESS_CONTROL_MAX_AGE}**
	KUBERNETES_NAMESPACE	–	–

12.1.2.4. 外部の依存関係

12.1.2.4.1. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

12.2. RHDM79-AUTHORING.YAML TEMPLATE

Red Hat Decision Manager 7.9 の HA 以外の永続的なオーサリング環境向けのアプリケーションテンプレート (非推奨)

12.2.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
CREDENTIALS_SECRET	–	KIE_ADMIN_USER 値および KIE_ADMIN_PWD 値を含むシークレット。	rhpam-credentials	True
KIE_SERVER_CONTROLLER_TOKEN	**KIE_SERVER_CONTROLLER_TOKEN**	ベアラー認証用の KIE Server コントローラートークン。 (org.kie.server.controller.token システムプロパティーを設定)	–	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。 (org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_MODE	**KIE_SERVER_MODE**	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**DEVELOPMENT**	False
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans の有効化/無効化 (システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)。	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Decision Central の http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	–	False
DECISION_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Decision Central の https サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合には空白にします (例: <application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	KIE Server の http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	KIE Server の https サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	–	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_HTTPS_SECRET	–	Decision Central のキーストアファイルが含まれるシークレットの名前。	decisioncentral-app-secret	True
DECISION_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイルの名前。	keystore.jks	False
DECISION_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	サーバー証明書に関連付けられている名前	jboss	False
DECISION_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	キーストアおよび証明書のパスワード。	mykeystorepass	False
KIE_SERVER_HTTPS_SECRET	–	キーストアファイルを含むシークレット名	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	キーストアおよび証明書のパスワード。	mykeystorepass	False
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	false	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	true	False
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	60000	False
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	–	False
MAVEN_MIRROR_OF	**MAVEN_MIRROR_OF**	KIE Server の Maven ミラー設定。	external:*,!repo-rhdmcentr	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	**MAVEN_REPO_ID**	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False
MAVEN_REPO_URL	**MAVEN_REPO_URL**	Maven リポジトリーまたはサービスへの完全修飾 URL。	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	git フックに使用するディレクトリー (必要な場合)。	**/opt/kie/data/git/hooks**	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_VOLUME_CAPACITY	–	Decision Central のランタイムデータに向けた永続ストレージのサイズ。	1Gi	True
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central コンテナーのメモリー制限。	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限。	1Gi	False
SSO_URL	**SSO_URL**	RH-SSO URL。	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO クライアント名。	–	False
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.2.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.2.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-rhdmcentr	8080	http	Decision Central のすべての Web サーバーのポート。
	8443	https	

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver	8080	http	すべての KIE Server Web サーバーのポート。
	8443	https	

12.2.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-rhdmcentr-http	なし	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
${APPLICATION_NAME}-rhdmcentr-https	TLS パススルー	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS パススルー	**${KIE_SERVER_HOSTNAME_HTTPS}**

12.2.2.3. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.2.2.3.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-rhdmcentr	ImageChange
${APPLICATION_NAME}-kieserver	ImageChange

12.2.2.3.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-rhdmcentr	1
${APPLICATION_NAME}-kieserver	1

12.2.2.3.3. Pod テンプレート

12.2.2.3.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-rhdmcentr	**${APPLICATION_NAME}-rhdmsvc**
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-rhdmsvc**

12.2.2.3.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-rhdmcentr	rhdm-decisioncentral-rhel8
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.2.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

12.2.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

- ${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.2.2.3.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-rhdmcentr	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**

12.2.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-rhdmcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_MBEANS	KIE Server の mbeans の有効化/無効化 (システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)。	**${KIE_MBEANS}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	false
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	KIE_SERVER_CONTROLLER_TOKEN	ベアラー認証用の KIE Server コントローラートークン。(org.kie.server.controller.token システムプロパティーを設定)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhdmcentr**

デプロイメント	変数名	説明	値の例
	MAVEN_MIRROR_URL	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	GIT_HOOKS_DIR	git フックに使用するディレクトリー (必要な場合)。	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/decisioncentral-secret-volume**

デプロイメント	変数名	説明	値の例
	HTTPS_KEYSTORE	シークレット内のキーストアファイル名	**${DECISION_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${DECISION_CENTRAL_HTTPS_NAME}**
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	**${DECISION_CENTRAL_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO クライアントシークレット。	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO クライアント名。	**${DECISION_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	HOSTNAME_HTTP	Decision Central の http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Decision Central の https サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合には空白にします (例: <application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**${KIE_SERVER_MODE}**

デプロイメント	変数名	説明	値の例
	KIE_MBEANS	KIE Server の mbeans の有効化/無効化 (システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)。	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	ControllerBasedStartupStrategy

デプロイメント	変数名	説明	値の例
	MAVEN_MIRROR_URL	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定。	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	認証情報のシークレットに合わせて設定
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	認証情報のシークレットに合わせて設定

デプロイメント	変数名	説明	値の例
	EXTERNAL_MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード。	**${KIE_SERVER_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–
	SSO_URL	RH-SSO URL。	**${SSO_URL}**

デプロイメント	変数名	説明	値の例
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット。	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	KIE Server の http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	KIE Server の https サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	**${KIE_SERVER_HOSTNAME_HTTPS}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	**${AUTH_LDAP_DEFAULT_ROLE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**

12.2.2.3.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-rhdmcentr	decisioncentral-keystore-volume	**/etc/decisioncentral-secret-volume**	ssl certs	True
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True

12.2.2.4. 外部の依存関係

12.2.2.4.1. ボリューム要求

PersistentVolume オブジェクトは、OpenShift クラスターのストレージリソースです。管理者が GCE Persistent Disks、AWS Elastic Block Store (EBS)、NFS マウントなどのソースから **PersistentVolume** オブジェクトを作成して、ストレージをプロビジョニングします。詳細は、Openshift ドキュメント を参照してください。

名前	アクセスモード
${APPLICATION_NAME}-rhdmcentr-claim	ReadWriteOnce

12.2.2.4.2. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

decisioncentral-app-secret kieserver-app-secret

12.3. RHDM79-AUTHORING-HA.YAML TEMPLATE

Red Hat Decision Manager 7.9 の HA の永続的なオーサリング環境向けのアプリケーションテンプレート (非推奨)

12.3.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True

変数名	イメージの環境変数	説明	値の例	必須
CREDENTIALS_SECRET	–	KIE_ADMIN_USER 値および KIE_ADMIN_PWD 値を含むシークレット。	rhpam-credentials	True
KIE_SERVER_CONTROLLER_TOKEN	**KIE_SERVER_CONTROLLER_TOKEN**	ベアラー認証用の KIE Server コントローラートークン。(org.kie.server.controller.token システムプロパティーを設定)	–	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False
KIE_SERVER_MODE	**KIE_SERVER_MODE**	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**DEVELOPMENT**	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Decision Central の http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	–	False
DECISION_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Decision Central の https サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合には空白にします (例: <application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	KIE Server の http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	KIE Server の https サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	–	False
DECISION_CENTRAL_HTTPS_SECRET	–	Decision Central のキーストアファイルが含まれるシークレットの名前。	decisioncentral-app-secret	True
DECISION_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	Decision Central のシークレット内のキーストアファイルの名前。	keystore.jks	False
DECISION_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	Decision Central のサーバー証明書に関連付けられている名前。	jboss	False
DECISION_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	Decision Central のキーストアおよび証明書のパスワード。	mykeystorepass	False
KIE_SERVER_HTTPS_SECRET	–	KIE Server のキーストアファイルが含まれるシークレットの名前。	kieserver-app-secret	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	KIE Server のシークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	KIE Server のサーバー証明書に関連付けられている名前。	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	KIE Server のキーストアおよび証明書のパスワード。	mykeystorepass	False
APPFORMER_JMS_BROKER_USER	**APPFORMER_JMS_BROKER_USER**	JMS ブローカーに接続するためのユーザー名。	jmsBrokerUser	True
APPFORMER_JMS_BROKER_PASSWORD	**APPFORMER_JMS_BROKER_PASSWORD**	JMS ブローカーに接続するためのパスワード。	–	True
DATAGRID_IMAGE	–	DataGrid イメージ。	registry.redhat.io/jboss-datagrid-7/datagrid73-openshift:1.6	True
DATAGRID_CPU_LIMIT	–	DataGrid Container の CPU 制限。	1000m	True
DATAGRID_MEMORY_LIMIT	–	DataGrid コンテナーのメモリー制限。	2Gi	True
DATAGRID_VOLUME_CAPACITY	–	DataGrid のランタイムデータの永続ストレージのサイズ。	1Gi	True
AMQ_BROKER_IMAGE	–	AMQ ブローカーイメージ。	registry.redhat.io/amq7/amq-broker:7.7	True
AMQ_ROLE	–	標準ブローカーユーザーのユーザーロール。	admin	True

変数名	イメージの環境変数	説明	値の例	必須
AMQ_NAME	–	ブローカーの名前。	broker	True
AMQ_GLOBAL_MAX_SIZE	–	メッセージデータが使用可能な最大メモリー量を指定します。値が指定されていない場合は、システムのメモリーの半分が割り当てられます。	10 gb	False
AMQ_VOLUME_CAPACITY	–	AMQ ブローカーボリュームの永続ストレージのサイズ。	1Gi	True
AMQ_REPLICAS	–	クラスターのブローカーレプリカ数。	2	True
DECISION_CENTRAL_CONTAINER_REPLICAS	–	Decision Central Container Replicas は、起動する Decision Central のコンテナー数を定義します。	2	True
KIE_SERVER_CONTAINER_REPLICAS	–	KIE Server Container Replicas は、起動する KIE Server のコンテナー数を定義します。	2	True
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	false	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	true	False
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	60000	False
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_IMAGE_STREAM_NAME	–	Decision Central に使用するイメージストリームの名前。デフォルトは rhdm-decisioncentral-rhel8 です。	rhdm-decisioncentral-rhel8	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	–	False
MAVEN_MIRROR_OF	**MAVEN_MIRROR_OF**	KIE Server の Maven ミラー設定。	external:*,!repo-rhdmcentr	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	**MAVEN_REPO_ID**	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False
MAVEN_REPO_URL	**MAVEN_REPO_URL**	Maven リポジトリーまたはサービスへの完全修飾 URL。	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	git フックに使用するディレクトリー (必要な場合)。	**/opt/kie/data/git/hooks**	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_VOLUME_CAPACITY	–	Decision Central のランタイムデータに向けた永続ストレージのサイズ。	1Gi	True
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central コンテナーのメモリー制限。	8Gi	True
DECISION_CENTRAL_JAVA_MAX_MEM_RATIO	**JAVA_MAX_MEM_RATIO**	Decision Central コンテナー JVM の最大メモリー比率。**-Xmx** がコンテナーで利用可能なメモリーの比率に設定されます。デフォルトは 80 です。これは、利用可能なメモリーの範囲の上限が 80% であることを意味します。**-Xmx** オプションの追加を省略するには、この値を 0 に設定します。	80	True
DECISION_CENTRAL_CPU_LIMIT	–	Decision Central コンテナーの CPU 制限。	2000m	True
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限。	1Gi	True
KIE_SERVER_CPU_LIMIT	–	KIE Server コンテナーの CPU 制限。	1000m	True
SSO_URL	**SSO_URL**	RH-SSO URL。	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO クライアント名。	–	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール	user	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.3.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.3.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-rhdmcentr	8080	http	Decision Central のすべての Web サーバーのポート。
	8443	https	

サービス	ポート	名前	説明
${APPLICATION_NAME}-rhdmcentr-ping	8888	ping	rhdmcentr クラスターリングの JGroups ping ポート。
${APPLICATION_NAME}-datagrid-ping	8888	ping	クラスター化されたアプリケーションの ping サービスを提供します。
${APPLICATION_NAME}-datagrid	11222	hotrod	Hot Rod プロトコルでアプリケーションにアクセスするためのサービスを提供します。
${APPLICATION_NAME}-kieserver	8080	http	すべての KIE Server Web サーバーのポート。
	8443	https	
${APPLICATION_NAME}-amq-tcp	61616	–	ブローカーの OpenWire ポート。
ping	8888	–	amq クラスターリングの JGroups ping ポート。

12.3.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-rhdmcentr-http	なし	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
${APPLICATION_NAME}-rhdmcentr-https	TLS パススルー	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS パススルー	**${KIE_SERVER_HOSTNAME_HTTPS}**

12.3.2.3. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.3.2.3.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-rhdmcentr	ImageChange
${APPLICATION_NAME}-kieserver	ImageChange

12.3.2.3.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-rhdmcentr	2
${APPLICATION_NAME}-kieserver	2

12.3.2.3.3. Pod テンプレート

12.3.2.3.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-rhdmcentr	**${APPLICATION_NAME}-rhdmsvc**
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-rhdmsvc**

12.3.2.3.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-rhdmcentr	**${DECISION_CENTRAL_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.3.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

12.3.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.3.2.3.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-rhdmcentr	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**

12.3.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-rhdmcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	true
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	true に設定すると、KIE Server のグローバル検出機能はオンになります (org.kie.server.controller.openshift.global.discovery.enabled システムプロパティーを設定)。	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	Business Central の OpenShift 統合がオンの場合は、このパラメーターを true に設定すると、OpenShift 内部サービスエンドポイント経由での KIE Server への接続が有効になります。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	KIE_SERVER_CONTROLLER_TOKEN	ベアラー認証用の KIE Server コントローラートークン。(org.kie.server.controller.token システムプロパティーを設定)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhdmcentr**
	MAVEN_MIRROR_URL	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**

デプロイメント	変数名	説明	値の例
	MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	GIT_HOOKS_DIR	git フックに使用するディレクトリー (必要な場合)。	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/decisioncentral-secret-volume**
	HTTPS_KEYSTORE	Decision Central のシークレット内のキーストアファイルの名前。	**${DECISION_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	Decision Central のサーバー証明書に関連付けられている名前。	**${DECISION_CENTRAL_HTTPS_NAME}**
	HTTPS_PASSWORD	Decision Central のキーストアおよび証明書のパスワード。	**${DECISION_CENTRAL_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-rhdmcentr-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	APPFORMER_INFINISPAN_SERVICE_NAME	–	**${APPLICATION_NAME}-datagrid**
	APPFORMER_INFINISPAN_PORT	–	11222

デプロイメント	変数名	説明	値の例
	APPFORMER_JMS_BROKER_ADDRESS	–	**${APPLICATION_NAME}-amq-tcp**
	APPFORMER_JMS_BROKER_PORT	–	61616
	APPFORMER_JMS_BROKER_USER	JMS ブローカーに接続するためのユーザー名。	**${APPFORMER_JMS_BROKER_USER}**
	APPFORMER_JMS_BROKER_PASSWORD	JMS ブローカーに接続するためのパスワード。	**${APPFORMER_JMS_BROKER_PASSWORD}**
	JAVA_MAX_MEM_RATIO	Decision Central コンテナー JVM の最大メモリー比率。**-Xmx** がコンテナーで利用可能なメモリーの比率に設定されます。デフォルトは 80 です。これは、利用可能なメモリーの範囲の上限が 80% であることを意味します。**-Xmx** オプションの追加を省略するには、この値を 0 に設定します。	**${DECISION_CENTRAL_JAVA_MAX_MEM_RATIO}**
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO クライアントシークレット。	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO クライアント名。	**${DECISION_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**

デプロイメント	変数名	説明	値の例
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Decision Central の http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Decision Central の https サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合には空白にします (例: <application-name>-rhdmcentr-<project>.<default-domain-suffix>)。	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**${KIE_SERVER_MODE}**

デプロイメント	変数名	説明	値の例
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy

デプロイメント	変数名	説明	値の例
	MAVEN_MIRROR_URL	Decision Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定。	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	認証情報のシークレットに合わせて設定
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	認証情報のシークレットに合わせて設定

デプロイメント	変数名	説明	値の例
	EXTERNAL_MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	KIE Server のシークレット内のキーストアファイルの名前。	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	KIE Server のサーバー証明書に関連付けられている名前。	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	KIE Server のキーストアおよび証明書のパスワード。	**${KIE_SERVER_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–

デプロイメント	変数名	説明	値の例
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット。	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	KIE Server の http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**

デプロイメント	変数名	説明	値の例
	HOSTNAME_HTTPS	KIE Server の https サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**

12.3.2.3.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-rhdmcentr	decisioncentral-keystore-volume	**/etc/decisioncentral-secret-volume**	ssl certs	True
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True

12.3.2.4. 外部の依存関係

12.3.2.4.1. ボリューム要求

PersistentVolume オブジェクトは、OpenShift クラスターのストレージリソースです。管理者が GCE Persistent Disks、AWS Elastic Block Store (EBS)、NFS マウントなどのソースから **PersistentVolume** オブジェクトを作成して、ストレージをプロビジョニングします。詳細は、Openshift ドキュメント を参照してください。

名前	アクセスモード
${APPLICATION_NAME}-rhdmcentr-claim	ReadWriteMany

12.3.2.4.2. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

decisioncentral-app-secret kieserver-app-secret

12.3.2.4.3. クラスターリング

OpenShift EAP では、Kubernetes または DNS の検出メカニズム 2 つの内 1 つを使用してクラスターリングを実現できます。これには、standalone-openshift.xml で **<openshift.KUBE_PING/>** 要素または **<openshift.DNS_PING/>** 要素のいずれかを指定して JGroups プロトコルスタックを設定します。テンプレートは、**DNS_PING** を使用するように設定しますが、イメージで使用するデフォルトは `KUBE_PING` となっています。

使用される検出メカニズムは、**JGROUPS_PING_PROTOCOL** 環境変数によって指定されます。これは **openshift.DNS_PING** または **openshift.KUBE_PING** のいずれかに設定できます。**OpenShift.KUBE_PING** は、**JGROUPS_PING_PROTOCOL** に値が指定されていない場合は、イメージによって使用されるデフォルトです。

DNS_PING を機能させるには、以下の手順を実行する必要があります。

1. **OPENSHIFT_DNS_PING_SERVICE_NAME** 環境変数は、クラスターの ping サービス名に設定する必要があります (上記の表を参照)。設定していない場合には、サーバーは単一ノードのクラスター (ノードが 1 つのクラスター) のように機能します。

2. **OPENSHIFT_DNS_PING_SERVICE_PORT** 環境変数は、サービスを公開するポート番号

2. **OPENSHIFT_DNS_PING_SERVICE_PORT** 環境変数は、ping サービスを公開するポート番号に設定する必要があります (上記の表を参照)。**DNS_PING** プロトコルは可能な場合には SRV レコードからのポートを識別しようとします。デフォルト値は 8888 です。

3. ping ポートを公開する ping サービスは定義する必要があります。このサービスはヘッドレス (ClusterIP=None) で、以下の条件を満たす必要があります。

    a. ポートは、ポート検出が機能するように、名前を指定する必要があります。

    b. **service.alpha.kubernetes.io/tolerate-unready-endpoints** を **"true"** に指定してアノテーションを設定する必要があります。このアノテーションを省略すると、起動時にノードごとに独自の単一ノードのクラスターが形成され、(起動後でないと他のノードが検出されないので) 起動後にこのクラスターが他のノードのクラスターにマージされます。

DNS_PING で使用する ping サービスの例

```
kind: Service
apiVersion: v1
spec:
   clusterIP: None
   ports:
   - name: ping
     port: 8888
   selector:
      deploymentConfig: eap-app
metadata:
   name: eap-app-ping
   annotations:
      service.alpha.kubernetes.io/tolerate-unready-endpoints: "true"
      description: "The JGroups ping port for clustering."
```

KUBE_PING を機能させるには以下の手順を実行する必要があります。

1. **OPENSHIFT_KUBE_PING_NAMESPACE** 環境変数を設定する必要があります (上記の表を参照)。設定していない場合には、サーバーは単一ノードのクラスター (ノードが 1 つのクラスター) のように機能します。

2. **OPENSHIFT_KUBE_PING_LABELS** 環境変数を設定する必要があります (上記の表を参照)。設定されていない場合には、アプリケーション外の Pod (namespace に関係なく) が参加しようとします。

3. Kubernetes の REST API にアクセスできるようにするには、Pod が実行されているサービスアカウントに対して承認を行う必要があります。これはコマンドラインで行います。

例12.1 policy コマンド

myproject の namespace におけるデフォルトのサービスアカウントの使用:

```
oc policy add-role-to-user view system:serviceaccount:myproject:default -n myproject
```

myproject の namespace における eap-service-account の使用:

```
oc policy add-role-to-user view system:serviceaccount:myproject:eap-service-account -n
myproject
```

12.4. RHDM79-KIESERVER.YAML TEMPLATE

Red Hat Decision Manager 7.9 での管理 KIE Server 向けのアプリケーションテンプレート (非推奨)

12.4.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	–	False
MAVEN_MIRROR_OF	**MAVEN_MIRROR_OF**	KIE Server の Maven ミラー設定。	external:*	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	**EXTERNAL_MAVEN_REPO_ID**	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False
MAVEN_REPO_URL	**EXTERNAL_MAVEN_REPO_URL**	Maven リポジトリーまたはサービスへの完全修飾 URL。	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**EXTERNAL_MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_SERVICE	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	myapp-rhdmcentr	False
CREDENTIALS_SECRET	–	KIE_ADMIN_USER 値および KIE_ADMIN_PWD 値を含むシークレット。	rhpam-credentials	True
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_MODE	**KIE_SERVER_MODE**	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**PRODUCTION**	False
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	–	False
KIE_SERVER_HTTPS_SECRET	–	キーストアファイルを含むシークレット名	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	キーストアおよび証明書のパスワード。	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限。	1Gi	False
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。オプションでエイリアスあり (形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2)	rhdm-kieserver-library=org.openshift.quickstarts:rhdm-kieserver-library:1.6.0-SNAPSHOT	False
KIE_SERVER_MGMT_DISABLED	**KIE_SERVER_MGMT_DISABLED**	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	true	False
SSO_URL	**SSO_URL**	RH-SSO URL。	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット	252793ed-7118-4ca8-8dab-5622fa97d892	False

変数名	イメージの環境変数	説明	値の例	必須
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール。	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このプロパティーは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.4.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.4.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver	8080	http	すべての KIE Server Web サーバーのポート。
	8443	https	

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver-ping	8888	ping	クラスターリング向けの JGroups ping ポート。

12.4.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS パススルー	**${KIE_SERVER_HOSTNAME_HTTPS}**

12.4.2.3. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.4.2.3.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-kieserver	ImageChange

12.4.2.3.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-kieserver	1

デプロイメント	レプリカ

12.4.2.3.3. Pod テンプレート

12.4.2.3.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

12.4.2.3.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.4.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

12.4.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.4.2.3.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**

12.4.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**${KIE_SERVER_MODE}**
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**

デプロイメント	変数名	説明	値の例
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_MIRROR_URL	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定。	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr

デプロイメント	変数名	説明	値の例
	RHDMCENTR_MAVEN_REPO_SERVICE	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	認証情報のシークレットに合わせて設定
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	認証情報のシークレットに合わせて設定
	EXTERNAL_MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーまたはサービスへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_MGMT_DISABLED	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード。	**${KIE_SERVER_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**

デプロイメント	変数名	説明	値の例
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール。	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このプロパティーは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**

12.4.2.3.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True

12.4.2.4. 外部の依存関係

12.4.2.4.1. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

kieserver-app-secret

12.5. RHDM79-PROD-IMMUTABLE-KIESERVER.YAML TEMPLATE

Red Hat Decision Manager 7.9 での実稼働環境におけるイミュータブル KIE Server 向けのアプリケーションテンプレート (非推奨)

12.5.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
CREDENTIALS_SECRET	–	KIE_ADMIN_USER 値および KIE_ADMIN_PWD 値を含むシークレット。	rhpam-credentials	True
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	–	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HTTPS_SECRET	–	キーストアファイルを含むシークレット名	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	キーストアおよび証明書のパスワード。	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。オプションでエイリアスあり (形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2)	rhdm-kieserver-hellorules=org.openshift.quickstarts:rhdm-kieserver-hellorules:1.6.0-SNAPSHOT	True
SOURCE_REPOSITORY_URL	–	アプリケーションの Git ソース URI。	https://github.com/jboss-container-images/rhdm-7-openshift-image.git	True
SOURCE_REPOSITORY_REF	–	Git ブランチ/タグ参照。	master	False

変数名	イメージの環境変数	説明	値の例	必須
CONTEXT_DIR	–	ビルドする Git プロジェクト内のパス。ルートプロジェクトディレクトリーの場合は空になります。	quickstarts/hello-rules/hellorules	False
GITHUB_WEBHOOK_SECRET	–	GitHub トリガーシークレット。	–	True
GENERIC_WEBHOOK_SECRET	–	汎用ビルドのトリガーシークレット。	–	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	–	False
MAVEN_MIRROR_OF	**MAVEN_MIRROR_OF**	KIE Server の Maven ミラー設定。	external:*	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	**EXTERNAL_MAVEN_REPO_ID**	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False
MAVEN_REPO_URL	**EXTERNAL_MAVEN_REPO_URL**	Maven リポジトリーへの完全修飾 URL。	–	False
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**EXTERNAL_MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False
DECISION_CENTRAL_SERVICE	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	myapp-rhdmcentr	False

変数名	イメージの環境変数	説明	値の例	必須
ARTIFACT_DIR	–	deployment フォルダーにコピーするアーカイブ取得元のディレクトリー一覧。指定されていない場合は、全アーカイブまたはターゲットがコピーされます。	–	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限。	1Gi	False
KIE_SERVER_MGMT_DISABLED	**KIE_SERVER_MGMT_DISABLED**	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	true	True
SSO_URL	**SSO_URL**	RH-SSO URL	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False

変数名	イメージの環境変数	説明	値の例	必須
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール。	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.5.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.5.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver	8080	http	すべての KIE Server Web サーバーのポート。
	8443	https	
${APPLICATION_NAME}-kieserver-ping	8888	ping	クラスターリング向けの JGroups ping ポート。

12.5.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS パススルー	**${KIE_SERVER_HOSTNAME_HTTPS}**

12.5.2.3. ビルド設定

buildConfig は、単一のビルド定義と、新規ビルドを作成する必要のあるタイミングについての一連のトリガーを記述します。**buildConfig** は REST オブジェクトで、API サーバーへの POST で使用して新規インスタンスを作成できます。詳細は、Openshift ドキュメント を参照してください。

S2I イメージ	リンク	ビルドの出力	BuildTriggers および設定
rhdm-kieserver-rhel8:7.9.0	**rhpam-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub、Generic、ImageChange、ConfigChange

12.5.2.4. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.5.2.4.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-kieserver	ImageChange

12.5.2.4.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照

してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-kieserver	2

12.5.2.4.3. Pod テンプレート

12.5.2.4.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

12.5.2.4.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

12.5.2.4.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

12.5.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.5.2.4.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**

デプロイメント	名前	ポート	プロトコル
	ping	8888	**TCP**

12.5.2.4.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_SERVER_MODE	–	**DEVELOPMENT**
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。(org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_MIRROR_URL	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定。	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**

デプロイメント	変数名	説明	値の例
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	認証情報のシークレットに合わせて設定
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	認証情報のシークレットに合わせて設定
	EXTERNAL_MAVEN_REPO_ID	Maven リポジトリーに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhdmcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${KIE_SERVER_HTTPS_NAME}**

デプロイメント	変数名	説明	値の例
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード。	**${KIE_SERVER_HTTPS_PASSWORD}**
	KIE_SERVER_MGMT_DISABLED	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット。	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**

デプロイメント	変数名	説明	値の例
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	**${AUTH_LDAP_BIND_CREDENTIAL}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール。	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

デプロイメント	変数名	説明	値の例
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**

12.5.2.4.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True

12.5.2.5. 外部の依存関係

12.5.2.5.1. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

kieserver-app-secret

12.6. RHDM79-PROD-IMMUTABLE-KIESERVER-AMQ.YAML TEMPLATE

Red Hat Decision Manager 7.9 の ActiveMQ と統合された実稼働環境におけるイミュータブル KIE Server 向けのアプリケーションテンプレート (非推奨)

12.6.1. パラメーター

テンプレートを使用すると、値を引き継ぐパラメーターを定義できます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、Openshift ドキュメント を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
CREDENTIALS_SECRET	–	KIE_ADMIN_USER 値および KIE_ADMIN_PWD 値を含むシークレット。	rhpam-credentials	True
IMAGE_STREAM_NAMESPACE	–	Red Hat Decision Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは rhdm-kieserver-rhel8 です。	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 7.9.0 です。	7.9.0	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_MBEANS	**KIE_MBEANS**	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティーを設定)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	false	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	–	False
KIE_SERVER_HTTPS_SECRET	–	キーストアファイルを含むシークレット名	kieserver-app-secret	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	キーストアおよび証明書のパスワード	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	false	False
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。オプションでエイリアスあり (形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2)	rhdm-kieserver-hellorules=org.openshift.quickstarts:rhdm-kieserver-hellorules:1.6.0-SNAPSHOT	True
SOURCE_REPOSITORY_URL	–	アプリケーションの Git ソース URI。	https://github.com/jboss-container-images/rhdm-7-openshift-image.git	True
SOURCE_REPOSITORY_REF	–	Git ブランチ/タグ参照。	master	False

変数名	イメージの環境変数	説明	値の例	必須
CONTEXT_DIR	–	ビルドする Git プロジェクト内のパス。ルートプロジェクトディレクトリーの場合は空になります。	quickstarts/hello-rules/hellorules	False
GITHUB_WEBHOOK_SECRET	–	GitHub トリガーシークレット。	–	True
GENERIC_WEBHOOK_SECRET	–	汎用ビルドのトリガーシークレット。	–	True
MAVEN_MIRROR_URL	–	S2I ビルドに使用する Maven ミラー	–	False
MAVEN_REPO_ID	**EXTERNAL_MAVEN_REPO_ID**	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	my-repo-id	False
MAVEN_REPO_URL	**EXTERNAL_MAVEN_REPO_URL**	Maven リポジトリーへの完全修飾 URL。	–	False
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	–	False
MAVEN_REPO_PASSWORD	**EXTERNAL_MAVEN_REPO_PASSWORD**	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_SERVICE	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	myapp-rhdmcentr	False
ARTIFACT_DIR	–	deploymento フォルダーにコピーするアーカイブ取得元のディレクトリー一覧。指定されていない場合は、全アーカイブまたはターゲットがコピーされます。	–	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナーのメモリー制限	1Gi	False
KIE_SERVER_MGMT_DISABLED	**KIE_SERVER_MGMT_DISABLED**	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	true	True
KIE_SERVER_JMS_QUEUE_REQUEST	**KIE_SERVER_JMS_QUEUE_REQUEST**	JMS の要求キューの JNDI 名。デフォルト値は queue/KIE.SERVER.REQUEST です。	queue/KIE.SERVER.REQUEST	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_JMS_QUEUE_RESPONSE	**KIE_SERVER_JMS_QUEUE_RESPONSE**	JMS の応答キューの JNDI 名。デフォルト値は queue/KIE.SERVER.RESPONSE です。	queue/KIE.SERVER.RESPONSE	False
AMQ_USERNAME	**AMQ_USERNAME**	標準ブローカーユーザーのユーザー名。ブローカーに接続するために必要です。空白の場合は生成されます。	–	False
AMQ_PASSWORD	**AMQ_PASSWORD**	標準ブローカーユーザーのパスワード。ブローカーに接続するために必要です。空白の場合は生成されます。	–	False
AMQ_ROLE	**AMQ_ROLE**	標準ブローカーユーザーのユーザーロール。	admin	True

変数名	イメージの環境変数	説明	値の例	必須
AMQ_QUEUES	**AMQ_QUEUES**	コンマで区切られたキュー名。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ます。さらに、これらは EAP で JNDI リソースとしてアクセス可能になります。これらのキューは KIE Server が必要とするデフォルトキューです。カスタムキューを使用する場合は、KIE_SERVER_JMS_QUEUE_RESPONSE パラメーターおよび KIE_SERVER_JMS_QUEUE_REQUEST パラメーターと同じ値を使用します。	queue/KIE.SERVER.REQUEST,queue/KIE.SERVER.RESPONSE	False
AMQ_GLOBAL_MAX_SIZE	**AMQ_GLOBAL_MAX_SIZE**	メッセージデータが使用可能な最大メモリー量を指定します。値が指定されていない場合は、システムのメモリーの半分が割り当てられます。	10 gb	False
AMQ_SECRET	–	AMQ SSL 関連のファイルが含まれるシークレット名。	broker-app-secret	True
AMQ_TRUSTSTORE	**AMQ_TRUSTSTORE**	AMQ SSL トラストストアファイル名。	broker.ts	False
AMQ_TRUSTSTORE_PASSWORD	**AMQ_TRUSTSTORE_PASSWORD**	AMQ トラストストアのパスワード。	changeit	False
AMQ_KEYSTORE	**AMQ_KEYSTORE**	AMQ キーストアのファイル名。	broker.ks	False

変数名	イメージの環境変数	説明	値の例	必須
AMQ_KEYSTORE_PASSWORD	**AMQ_KEYSTORE_PASSWORD**	AMQ キーストアおよび証明書のパスワード。	changeit	False
AMQ_PROTOCOL	**AMQ_PROTOCOL**	コンマで区切られた、設定するブローカーのプロトコル。許可される値は、**openwire**、**amqp**、**stomp**、および **mqtt** です。**openwire** のみが EAP でサポートされます。	openwire	False
AMQ_BROKER_IMAGESTREAM_NAME	–	AMQ ブローカーイメージストリーム名。	amq-broker:7.7	True
AMQ_IMAGE_STREAM_NAMESPACE	–	Red Hat AMQ イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStream を別の名前空間/プロジェクトにインストールしている場合に限りこのパラメーターを変更する必要があります。	openshift	True
SSO_URL	**SSO_URL**	RH-SSO URL	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO レルム名。	–	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server の RH-SSO クライアント名。	–	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server の RH-SSO クライアントシークレット	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	–	False
SSO_PASSWORD	**SSO_PASSWORD**	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	ユーザー名として使用する RH-SSO プリンシパル属性。	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	認証に使用するバインド DN。	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	認証に使用する LDAP の認証情報。	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	使用する検索範囲。	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	distinguishedName	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	ユーザーロールを含む属性の名前。	memberOf	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	ou=groups,ou=example,ou=com	False
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	(memberOf={1})	False
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	認証された全ユーザーに対して含まれるロール	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	false	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	–	False

変数名	イメージの環境変数	説明	値の例	必須
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このプロパティーは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

12.6.2. オブジェクト

CLI はさまざまなオブジェクトタイプをサポートします。これらのオブジェクトタイプの一覧や略語については、Openshift ドキュメント を参照してください。

12.6.2.1. サービス

サービスは、Pod の論理セットや、Pod にアクセスするためのポリシーを定義する抽象概念です。詳細は、コンテナーエンジンのドキュメント を参照してください。

サービス	ポート	名前	説明
${APPLICATION_NAME}-kieserver	8080	http	すべての KIE Server Web サーバーのポート。
	8443	https	
${APPLICATION_NAME}-kieserver-ping	8888	ping	クラスターリング向けの JGroups ping ポート。

サービス	ポート	名前	説明
${APPLICATION_NAME}-amq-jolokia	8161	amq-jolokia-console	ブローカーのコンソールおよび Jolokia ポート。
${APPLICATION_NAME}-amq-amqp	5672	amq-amqp	ブローカーの AMQP ポート。
${APPLICATION_NAME}-amq-amqp-ssl	5671	amq-amqp-ssl	ブローカーの AMQP SSL ポート。
${APPLICATION_NAME}-amq-mqtt	1883	amq-mqtt	ブローカーの MQTT ポート。
${APPLICATION_NAME}-amq-mqtt-ssl	8883	amq-mqtt-ssl	ブローカーの MQTT SSL ポート。
${APPLICATION_NAME}-amq-stomp	61613	amq-stomp	ブローカーの STOMP ポート。
${APPLICATION_NAME}-amq-stomp-ssl	61612	amq-stomp-ssl	ブローカーの STOMP SSL ポート。
${APPLICATION_NAME}-amq-tcp	61616	amq-tcp	ブローカーの OpenWire ポート。
${APPLICATION_NAME}-amq-tcp-ssl	61617	amq-tcp-ssl	ブローカーの OpenWire (SSL) ポート。

12.6.2.2. ルート

ルートは、**www.example.com** などの外部から到達可能なホスト名を指定してサービスを公開する 1 つの手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、Openshift ドキュメント を参照してください。

サービス	セキュリティー	ホスト名
insecure-${APPLICATION_NAME}-kieserver-http	なし	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS パススルー	**${KIE_SERVER_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-amq-jolokia-console	TLS パススルー	<default>

サービス	セキュリティー	ホスト名
${APPLICATION_NAME}-amq-tcp-ssl	TLS パススルー	<default>

12.6.2.3. ビルド設定

buildConfig は、単一のビルド定義と、新規ビルドを作成する必要のあるタイミングについての一連のトリガーを記述します。**buildConfig** は REST オブジェクトで、API サーバーへの POST で使用して新規インスタンスを作成できます。詳細は、Openshift ドキュメント を参照してください。

S2I イメージ	リンク	ビルドの出力	BuildTriggers および設定
rhdm-kieserver-rhel8:7.9.0	**rhpam-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub、Generic、ImageChange、ConfigChange

12.6.2.4. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートをベースとするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、Openshift ドキュメント を参照してください。

12.6.2.4.1. トリガー

トリガーは、OpenShift 内外を問わず、イベントが発生すると新規デプロイメントを作成するように促します。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	トリガー
${APPLICATION_NAME}-kieserver	ImageChange
${APPLICATION_NAME}-amq	ImageChange

12.6.2.4.2. レプリカ

レプリケーションコントローラーを使用すると、指定した数だけ、Pod のレプリカを一度に実行させることができます。レプリカが増えると、レプリケーションコントローラーが Pod の一部を終了させます。レプリカが足りない場合には、起動させます。詳細は、コンテナーエンジンのドキュメント を参照してください。

デプロイメント	レプリカ
${APPLICATION_NAME}-kieserver	2
${APPLICATION_NAME}-amq	1

12.6.2.4.3. Pod テンプレート

12.6.2.4.3.1. サービスアカウント

サービスアカウントは、各プロジェクト内に存在する API オブジェクトです。他の API オブジェクトのように作成し、削除できます。詳細は、Openshift ドキュメント を参照してください。

デプロイメント	サービスアカウント
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

12.6.2.4.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**
${APPLICATION_NAME}-amq	**${AMQ_BROKER_IMAGESTREAM_NAME}**

12.6.2.4.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

${APPLICATION_NAME}-amq

> /bin/bash -c /opt/amq/bin/readinessProbe.sh

12.6.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

12.6.2.4.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**

デプロイメント	名前	ポート	プロトコル
${APPLICATION_NAME}-amq	console-jolokia	8161	**TCP**
	amq-amqp	5672	**TCP**
	amqp-ssl	5671	**TCP**
	amq-mqtt	1883	**TCP**
	mqtt-ssl	8883	**TCP**
	amq-stomp	61613	**TCP**
	stomp-ssl	61612	**TCP**
	amq-tcp	61616	**TCP**
	amq-tcp-ssl	61617	**TCP**

12.6.2.4.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**
	KIE_ADMIN_USER	管理ユーザー名。	認証情報のシークレットに合わせて設定
	KIE_ADMIN_PWD	管理ユーザーのパスワード。	認証情報のシークレットに合わせて設定
	KIE_SERVER_MODE	–	**DEVELOPMENT**
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**

デプロイメント	変数名	説明	値の例
	DROOLS_SERVER_FILTER_CLASSES	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティーを設定)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	false に設定すると、prometheus サーバー拡張が有効になります。(org.kie.prometheus.server.ext.disabled システムプロパティーを設定)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティーを設定)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server コンテナーのデプロイメント設定。任意でエイリアスあり。形式: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_SERVICE	必要かつ到達可能な場合にサービスルックアップ (maven リポジトリーの使用など) を許可するのに使用される任意の Decision Central のサービス名。	**${DECISION_CENTRAL_SERVICE}**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**

デプロイメント	変数名	説明	値の例
	RHDMCENTR_MAVEN_REPO_USERNAME	–	認証情報のシークレットに合わせて設定
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	認証情報のシークレットに合わせて設定
	EXTERNAL_MAVEN_REPO_ID	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーへの完全修飾 URL。	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするためのユーザー名 (必要な場合)。	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_JMS_QUEUE_REQUEST	JMS の要求キューの JNDI 名。デフォルト値は queue/KIE.SERVER.REQUEST です。	**${KIE_SERVER_JMS_QUEUE_REQUEST}**
	KIE_SERVER_JMS_QUEUE_RESPONSE	JMS の応答キューの JNDI 名。デフォルト値は queue/KIE.SERVER.RESPONSE です。	**${KIE_SERVER_JMS_QUEUE_RESPONSE}**
	MQ_SERVICE_PREFIX_MAPPING	–	**${APPLICATION_NAME}-amq7=AMQ**
	AMQ_USERNAME	標準ブローカーユーザーのユーザー名。ブローカーに接続するために必要です。空白の場合は生成されます。	**${AMQ_USERNAME}**

デプロイメント	変数名	説明	値の例
	AMQ_PASSWORD	標準ブローカーユーザーのパスワード。ブローカーに接続するために必要です。空白の場合は生成されます。	**${AMQ_PASSWORD}**
	AMQ_PROTOCOL	コンマで区切られた、設定するブローカーのプロトコル。許可される値は、**openwire**、**amqp**、**stomp**、および **mqtt** です。**openwire** のみが EAP でサポートされます。	tcp
	AMQ_QUEUES	コンマで区切られたキュー名。これらのキューは、ブローカーの起動時に自動的に作成されます。さらに、これらは EAP で JNDI リソースとしてアクセス可能になります。これらのキューは KIE Server が必要とするデフォルトキューです。カスタムキューを使用する場合は、KIE_SERVER_JMS_QUEUE_RESPONSE パラメーターおよび KIE_SERVER_JMS_QUEUE_REQUEST パラメーターと同じ値を使用します。	**${AMQ_QUEUES}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	**${KIE_SERVER_HTTPS_PASSWORD}**

デプロイメント	変数名	説明	値の例
	KIE_SERVER_MGMT_DISABLED	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。org.kie.server.mgmt.api.disabled プロパティーを true に、org.kie.server.startup.strategy プロパティー を LocalContainersStartupStrategy に設定します。	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)。	**${SSO_USERNAME}**

デプロイメント	変数名	説明	値の例
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>。	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。フェイルオーバーの場合は、2 つ以上の LDAP エンドポイントをスペースで区切って設定します。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN。	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報。	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定されている場合、DN はユーザー名に対して解析されます。false に設定されている場合、DN はユーザー名に対して解析されません。このオプションは、usernameBeginString および usernameEndString とともに使用されます。	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	ユーザー名を公開するため、DN の最初から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	ユーザー名を公開するため、DN の最後から削除される文字列を定義します。このオプションは usernameEndString と合わせて使用し、parseUsername が true に設定されている場合にのみ考慮されます。	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${AUTH_LDAP_ROLES_CTX_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_ROLE_FILTER	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (member={0}) です。認証済み userDN に一致する他の例は (member={1}) です。	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	クエリーによって返された DN に roleNameAttributeID が含まれるかどうかを示すフラグ。true に設定した場合には、DN は roleNameATtributeID に対してチェックされます。false に設定すると、DN は roleNameATtributeID に対して確認されません。このフラグは LDAP クエリーのパフォーマンスを向上できます。	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このプロパティーは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-amq	**AMQ_USER**	標準ブローカーユーザーのユーザー名。ブローカーに接続するために必要です。空白の場合は生成されます。	**${AMQ_USERNAME}**

デプロイメント	変数名	説明	値の例
	AMQ_PASSWORD	標準ブローカーユーザーのパスワード。ブローカーに接続するために必要です。空白の場合は生成されます。	**${AMQ_PASSWORD}**
	AMQ_ROLE	標準ブローカーユーザーのユーザーロール。	**${AMQ_ROLE}**
	AMQ_NAME	–	**${APPLICATION_NAME}-broker**
	AMQ_TRANSPORTS	コンマで区切られた、設定するブローカーのプロトコル。許可される値は、**openwire**、**amqp**、**stomp**、および **mqtt** です。**openwire** のみが EAP でサポートされます。	**${AMQ_PROTOCOL}**
	AMQ_QUEUES	コンマで区切られたキュー名。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ます。さらに、これらは EAP で JNDI リソースとしてアクセス可能になります。これらのキューは KIE Server が必要とするデフォルトキューです。カスタムキューを使用する場合は、KIE_SERVER_JMS_QUEUE_RESPONSE パラメーターおよび KIE_SERVER_JMS_QUEUE_REQUEST パラメーターと同じ値を使用します。	**${AMQ_QUEUES}**
	AMQ_GLOBAL_MAX_SIZE	メッセージデータが使用可能な最大メモリー量を指定します。値が指定されていない場合は、システムのメモリーの半分が割り当てられます。	**${AMQ_GLOBAL_MAX_SIZE}**

デプロイメント	変数名	説明	値の例
	AMQ_REQUIRE_LOGIN	–	true
	AMQ_ANYCAST_PREFIX	–	–
	AMQ_MULTICAST_PREFIX	–	–
	AMQ_KEYSTORE_TRUSTSTORE_DIR	–	**/etc/amq-secret-volume**
	AMQ_TRUSTSTORE	AMQ SSL トラストストアファイル名。	**${AMQ_TRUSTSTORE}**
	AMQ_TRUSTSTORE_PASSWORD	AMQ トラストストアのパスワード。	**${AMQ_TRUSTSTORE_PASSWORD}**
	AMQ_KEYSTORE	AMQ キーストアのファイル名。	**${AMQ_KEYSTORE}**
	AMQ_KEYSTORE_PASSWORD	AMQ キーストアおよび証明書のパスワード。	**${AMQ_KEYSTORE_PASSWORD}**

12.6.2.4.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True
${APPLICATION_NAME}-amq	broker-secret-volume	**/etc/amq-secret-volume**	ssl certs	True

12.6.2.5. 外部の依存関係

12.6.2.5.1. シークレット

このテンプレートでは、アプリケーションを実行するために以下のシークレットをインストールする必要があります。

kieserver-app-secret broker-app-secret

12.7. OPENSHIFT の使用に関するクイックリファレンス

Red Hat OpenShift Container Platform で Red Hat Decision Manager テンプレートのデプロイ、モニターリング、管理、デプロイ解除するには、OpenShift Web コンソールまたは **oc** コマンドを使用できます。

Web コンソールの使用に関する説明は、Web コンソールを使用したイメージの作成およびビルド を参照してください。

oc コマンドの使用方法に関する詳細は、CLI リファレンス を参照してください。次のコマンドが必要になる可能性があります。

- プロジェクトを作成するには、以下のコマンドを使用します。

```
$ oc new-project <project-name>
```

  詳細は、CLI を使用したプロジェクトの作成 を参照してください。

- テンプレートをデプロイするには (またはテンプレートからアプリケーションを作成するには)、以下のコマンドを実行します。

```
$ oc new-app -f <template-name> -p <parameter>=<value> -p <parameter>=<value> ...
```

  詳細は、CLI を使用したアプリケーションの作成 を参照してください。

- プロジェクト内のアクティブな Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods
```

- Pod のデプロイメントが完了し、実行中の状態になっているかどうかなど、Pod の現在のステータスを表示するには、以下のコマンドを使用します。

```
$ oc describe pod <pod-name>
```

  oc describe コマンドを使用して、他のオブジェクトの現在のステータスを表示できます。詳細は、アプリケーションの変更操作 を参照してください。

- Pod のログを表示するには、以下のコマンドを使用します。

```
$ oc logs <pod-name>
```

- デプロイメントログを表示するには、テンプレート参照で **DeploymentConfig** 名を検索し、以下のコマンドを入力します。

```
$ oc logs -f dc/<deployment-config-name>
```

  詳細は、デプロイメントログの表示 を参照してください。

- ビルドログを表示するには、テンプレート参照で **BuildConfig** 名を検索し、以下のコマンドを入力します。

```
$ oc logs -f bc/<build-config-name>
```

  詳細は、ビルドログのアクセス を参照してください。

- アプリケーションの Pod をスケーリングするには、テンプレート参照で **DeploymentConfig** 名を検索し、以下のコマンドを入力します。

```
$ oc scale dc/<deployment-config-name> --replicas=<number>
```

詳細は、手動スケーリング を参照してください。

- アプリケーションのデプロイメントを解除するには、以下のコマンドを使用してプロジェクトを削除します。

```
$ oc delete project <project-name>
```

または、**oc delete** コマンドを使用して、Pod またはレプリケーションコントローラーなど、アプリケーションの一部を削除できます。詳細は、アプリケーションの修正操作 を参照してください。

パート III. デシジョンエンジンを使用した高可用性イベント駆動型デシジョン機能の RED HAT OPENSHIFT CONTAINER PLATFORM への実装

ビジネスルール開発者は、デシジョンエンジンを使用するコードで、複合イベント処理 (CAP: Complex Event Processing) など、高可用性イベント駆動型デシジョン機能を使用できます。高可用性イベント駆動型デシジョン機能は、Red Hat OpenShift Container Platform に実装できます。

Operator を使用した Red Hat OpenShift Container Platform への Red Hat Decision Manager 環境のデプロイメント の記載のとおり、Red Hat OpenShift Container Platform では、Red Hat Decision Manager の標準デプロイメントを使用して、高可用性イベント駆動型デシジョン機能を実装することができません。理由は、標準デプロイメントは、ステートレス処理しかサポートしないためです。そのため、指定の参照実装を使用して、カスタム実装を作成する必要があります。

前提条件

- Red Hat OpenShift Container Platform バージョン 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 Process Automation Manager 7 でサポートされる設定 を参照してください。
- Red Hat AMQ Streams を含む OpenShift 環境に、Kafka Cluster がデプロイされている。
- OpenJDK Java 開発環境がインストールされている。
- Maven、Docker、および kubectl がインストールされている。
- OpenShift コマンドラインツール **oc** がインストールされている。

第13章 RED HAT OPENSHIFT CONTAINER PLATFORM での高可用性イベント駆動型デシジョン機能

デシジョンエンジンを使用して、Red Hat OpenShift Container Platform に高可用性イベント駆動型デシジョン機能を実装します。

イベント は、特定の時点で発生するファクトをモデル化します。デシジョンエンジンは、一時オペレーターが豊富にあり、イベントの比較、相関、累積ができます。イベント駆動型のデシジョン機能では、デシジョンエンジンがイベントをもとに一連の複雑なデシジョンを処理します。イベントはすべて、エンジンの状態を変更でき、後続のイベントのデシジョンに影響を与えます。

パートI「Operator を使用した Red Hat OpenShift Container Platform への Red Hat Decision Manager 環境のデプロイメント」 で説明されているように、Red Hat OpenShift Container Platform での Red Hat Decision Manager の標準デプロイメントを使用して、高可用性イベント駆動型の決定を実行することはできません。デプロイメントには、KIE Server Pod が含まれており、スケーリング時も Pod ごとに独立したままになります。Pod の状態は同期されません。そのため、ステートレス呼び出しのみを確実に処理できます。

複合イベント処理 (CEP) API は、デシジョンエンジンを含むイベント駆動型デシジョン機能で便利です。デシジョンエンジンは、CEP を使用してイベントコレクションにある複数のイベントを検出して処理し、イベント間に存在する関係を明確にして、このようなイベントや関係をもとに新規データを推測します。デシジョンエンジンでの CEP に関する情報は、**Red Hat Decision Manager のデシジョンエンジン** を参照してください。

Red Hat Decision Manager が提供する参照実装をもとに、Red Hat OpenShift Container Platform に高可用性イベント駆動型デシジョン機能を実装します。この実装を使用すると、安全にフェイルオーバーできる環境が実現できます。

この参照実装では、処理コードを使用して Pod をスケーリングできます。Pod のレプリカは独立していません。レプリカの 1 つが自動的に **リーダー** として指定されます。リーダーが機能を停止した場合には、別のリーダーが自動的にリーダーになり、中断やデータの損失なしに、処理が続行されます。

リーダーの選択は、Kubernetes ConfigMaps で実装されます。リーダーと他のレプリカは、Kafka を介してメッセージを交換することで連携します。リーダーが必ず、最初にイベントを処理します。処理が完了したら、リーダーは他のレプリカに通知します。リーダーではないレプリカは、リーダーでの処理が完了してからでないとイベントは実行されません。

新規レプリカがクラスターに参加すると、このレプリカは、リーダーから、現在の Drools セッションのスナップショットを要求します。Kafka トピックで利用可能なスナップショットがある場合に、リーダーは既存で最新のスナップショットを使用できます。最新のスナップショットがない場合はリーダーがオンデマンドで新しいスナップショットを生成します。スナップショットを受信後に、新しいレプリカはそのスナップショットをデシリアライズし、最終的に、スナップショットに含まれていない最後のイベントを実行し、その後にリーダーと連携して新規イベントの処理は開始されません。

デフォルトの実装方法では、このサービスは HA CEP サーバーに KJAR として組み込まれています。このような場合は、サーバーをもう一度ビルドしてデプロイし、サービスのバージョンを変更します。新規バージョンに切り替えると、作業メモリーの内容は失われます。デフォルトの実装方法に関する詳細は、*14章HA CEP サーバーの実装* を参照してください。

作業メモリーの内容を失わずにサービスのバージョンをアップグレードする場合には、別の実装方法を使用して、KJAR と全依存関係を Maven リポジトリーに用意します。この実装方法では、クライアントコードから **UpdateKJarGAV** 呼び出しを使用して、新規 KJAR バージョンのデプロイメントをトリガーします。この呼び出しは、リーダーと他のレプリカが処理し、各 Pod が新しい KJAR を読み込みます。作業メモリーの内容は、そのまま残ります。この実装方法に関する詳細は、*15章Mave リポジトリーを使用した HA CEP サーバーを実装して KJAR サービスを更新する手順* を参照してください。

第14章 HA CEP サーバーの実装

高可用性 (HA) CEP サーバーは、Red Hat OpenShift Container Platform 環境で実行します。このサーバーには、必要なすべての Drools ルールと、イベント処理に必要なその他のコードが含まれています。

ソースを準備して、ビルドし、Red Hat OpenShift Container Platform にデプロイします。

または、別のプロセスを使用して、いつでも KJAR サービスを更新できる HA CEP サーバーをデプロイします。このプロセスに関する詳細は、15章*Mave リポジトリーを使用した HA CEP サーバーを実装して KJAR サービスを更新する手順* を参照してください。

前提条件

- **oc** コマンドラインツールを使用して、管理者権限があるプロジェクトにログインしている。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-reference-implementation.zip** をダウンロードします。
2. ファイルの内容を展開してから、**rhdm-7.9.1-openshift-drools-hacep-distribution.zip** ファイルを展開します。
3. **openshift-drools-hacep-distribution/sources** ディレクトリーに移動します。
4. **sample-hacep-project/sample-hacep-project-kjar** ディレクトリー内のサンプルプロジェクトをもとに、サーバーのコードを確認して変更します。複合イベント処理のロジックは、**src/main/resources/org/drools/cep** サブディレクトリーの DRL ルールで定義します。
5. 標準の Maven コマンドを使用してプロジェクトをビルドします。

```
mvn clean install -DskipTests
```

6. Red Hat AMQ Streams 向けの OpenShift operator を有効にして、プロジェクトで AMQ Streams (kafka) クラスターを作成します。Red Hat AMQ Streams のインストールに関する情報は、**Using AMQ Streams on OpenShift** を参照してください。
7. サーバーの操作に必要な kafka のトピックを作成するには、**openshift-drools-hacep-distribution/sources** ディレクトリーで、以下のコマンドを実行します。

```
oc apply -f kafka-topics/control.yaml
oc apply -f kafka-topics/events.yaml
oc apply -f kafka-topics/kiesessioninfos.yaml
oc apply -f kafka-topics/snapshot.yaml
```

8. アプリケーションが、リーダーの選択に使用する ConfigMap にアクセスできるように、ロールベースのアクセス制御を設定します。**springboot** ディレクトリーに移動して、以下のコマンドを入力します。

```
oc create -f kubernetes/service-account.yaml
oc create -f kubernetes/role.yaml
oc create -f kubernetes/role-binding.yaml
```

Red Hat OpenShift Container Platform のロールベースのアクセス制御に関する詳細は、Red Hat OpenShift Container Platform 製品ドキュメントの RBAC の使用によるパーミッションの定義および適用 を参照してください。

9. **springboot** ディレクトリーで、以下のコマンドを実行してデプロイメント用のイメージを作成し、OpenShift 環境用に設定したリポジトリーにプッシュします。

```
oc new-build --binary --strategy=docker --name openshift-kie-springboot
oc start-build openshift-kie-springboot --from-dir=. --follow
```

10. 以下のコマンドを実行して、ビルドしたイメージの名前を検出します。

```
oc get is/openshift-kie-springboot -o template --template='{{range .status.tags}}{{range
.items}}{{.dockerImageReference}}{{end}}{{end}}'
```

11. テキストエディターで **kubernetes/deployment.yaml** ファイルを開きます。
12. 既存のイメージ URL を直前のコマンドの結果に置き換えます。
13. 文頭に @ の記号がある行の最後にある文字をすべて削除し、その行に **:latest** を追加します。以下に例を示します。

```
image: image-registry.openshift-image-registry.svc:5000/hacep/openshift-kie-
springboot:latest
```

14. ファイルを保存します。
15. 以下のコマンドを実行してイメージをデプロイします。

```
oc apply -f kubernetes/deployment.yaml
```

第15章 MAVE リポジトリーを使用した HA CEP サーバーを実装して KJAR サービスを更新する手順

HA CEP サーバーを実装して、KJAR サービスと全依存関係を指定の Maven リポジトリーから取得できます。このような場合、Maven リポジトリーでサービスを更新して、クライアントコードから呼び出しを行うことで、いつでも KJAR サービスを更新できます。

ソースを準備して、ビルドし、Red Hat OpenShift Container Platform にデプロイします。サーバーをデプロイする前に、**deployment.yaml** ファイルに特定の環境変数を設定します。Maven リポジトリーを使用するには、**UPDATABLEKJAR** 変数を **true** に設定する必要があります。

前提条件

- **oc** コマンドラインツールを使用して、管理者権限があるプロジェクトにログインしている。
- Red Hat OpenShift Container Platform 環境からアクセス可能な Maven リポジトリーを設定している。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-reference-implementation.zip** をダウンロードします。
2. ファイルの内容を展開してから、**rhdm-7.9.1-openshift-drools-hacep-distribution.zip** ファイルを展開します。
3. **openshift-drools-hacep-distribution/sources** ディレクトリーに移動します。
4. **sample-hacep-project/sample-hacep-project-kjar** ディレクトリー内のサンプルプロジェクトをもとに、サーバーのコードを確認して変更します。複合イベント処理のロジックは、**src/main/resources/org/drools/cep** サブディレクトリーの DRL ルールで定義します。
5. 標準の Maven コマンドを使用してプロジェクトをビルドします。

   ```
   mvn clean install -DskipTests
   ```

   作成した KJAR と必要な依存関係を Maven リポジトリーにアップロードします。
6. Red Hat AMQ Streams 向けの OpenShift operator を有効にして、プロジェクトで AMQ Streams (kafka) クラスターを作成します。Red Hat AMQ Streams のインストールに関する情報は、Using AMQ Streams on OpenShift を参照してください。
7. サーバーの操作に必要な kafka のトピックを作成するには、**openshift-drools-hacep-distribution/sources** ディレクトリーで、以下のコマンドを実行します。

   ```
   oc apply -f kafka-topics/control.yaml
   oc apply -f kafka-topics/events.yaml
   oc apply -f kafka-topics/kiesessioninfos.yaml
   oc apply -f kafka-topics/snapshot.yaml
   ```
8. アプリケーションが、リーダーの選択に使用する ConfigMap にアクセスできるように、ロールベースのアクセス制御を設定します。**springboot** ディレクトリーに移動して、以下のコマンドを入力します。

```
oc create -f kubernetes/service-account.yaml
oc create -f kubernetes/role.yaml
oc create -f kubernetes/role-binding.yaml
```

Red Hat OpenShift Container Platform のロールベースのアクセス制御に関する詳細は、Red Hat OpenShift Container Platform 製品ドキュメントの RBAC の使用によるパーミッションの定義および適用 を参照してください。

9. **springboot** ディレクトリーで **pom.xml** ファイルを編集して、以下の依存関係を削除します。

```
<dependency>
    <groupId>org.kie</groupId>
    <artifactId>sample-hacep-project-kjar</artifactId>
</dependency>
```

10. **springboot** ディレクトリーで、以下のコマンドを実行してデプロイメント用のイメージを作成し、OpenShift 環境用に設定したリポジトリーにプッシュします。

```
oc new-build --binary --strategy=docker --name openshift-kie-springboot
oc start-build openshift-kie-springboot --from-dir=. --follow
```

11. 以下のコマンドを実行して、ビルドしたイメージの名前を検出します。

```
oc get is/openshift-kie-springboot -o template --template='{{range .status.tags}}{{range
.items}}{{.dockerImageReference}}{{end}}{{end}}'
```

12. テキストエディターで **kubernetes/deployment.yaml** ファイルを開きます。

13. 既存のイメージ URL を直前のコマンドの結果に置き換えます。

14. 文頭に @ の記号がある行の最後にある文字をすべて削除し、その行に **:latest** を追加します。以下に例を示します。

```
image: image-registry.openshift-image-registry.svc:5000/hacep/openshift-kie-
springboot:latest
```

15. **containers:** 行と **env:** 行の下で、以下の例のように環境変数を設定します。

```
containers:
  - env:
    - name: UPDATABLEKJAR
      value: "true"
    - name: KJARGAV
      value: <GroupID>:<ArtifactID>:<Version>
    - name: MAVEN_LOCAL_REPO
      value: /app/.m2/repository
    - name: MAVEN_MIRROR_URL
      value: http://<nexus_url>/repository/maven-releases/
    - name: MAVEN_SETTINGS_XML
      value: /app/.m2/settings.xml
```

この例では **KJARGAV** 変数は、KJAR サービスのグループ、アーティファクト、バージョン (GAV) に置き換え、**MAVEN_MIRROR_URL** 変数の値は、KJAR サービスを含む Maven リポジトリーの URL に置き換えます。

必要に応じて他の変数を設定します。サポート対象の環境変数の一覧は、「HA CEP サーバーがサポートする環境変数 (オプション)」 を参照してください。

16. ファイルを保存します。

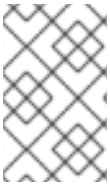
17. 以下のコマンドを実行してイメージをデプロイします。

```
oc apply -f kubernetes/deployment.yaml
```

クライアントコードから KJAR の更新をトリガーする方法は、16章*HA CEP クライアントの作成* を参照してください。

15.1. HA CEP サーバーがサポートする環境変数 (オプション)

以下の表には、Maven リポジトリーを使用するように設定された HA CEP サーバーに設定できる任意の環境変数をまとめています。これらの変数を **deployment.yaml** ファイルに追加し、それらをデプロイメント時に設定します。



注記

Maven リポジトリーを使用するには、15章*Mave リポジトリーを使用した HA CEP サーバーを実装して KJAR サービスを更新する手順* の説明に従って、サーバーの **UPDATABLEKJAR** 環境変数および **KJARGAV** 環境変数を設定してください。

表15.1 HA CEP サーバーがサポートする環境変数 (オプション)

名前	説明	例
MAVEN_LOCAL_REPO	ローカルの Maven リポジトリーとして使用するディレクトリー。	**/root/.m2/repository**
MAVEN_MIRROR_URL	アーティファクトの取得に使用可能な Maven ミラーのベース URL。	**http://nexus3-my-kafka-project.192.168.99.133.nip.io/repository/maven-public/**
MAVEN_MIRRORS	設定すると、マルチミラーサポートが有効になります。この値には、コンマで区切れたミラーのプリフィックス一覧が含まれます。この変数を設定した場合には、他の **MAVEN_MIRROR_*** 変数の名前に **DEV_MAVEN_MIRROR_URL** や **QE_MAVEN_MIRROR_URL** などのプリフィックスを含める必要があります。	**DEV,QE**
MAVEN_REPOS	設定すると、マルチリポジトリーサポートが有効になります。この値には、コンマで区切れたリポジトリーのプリフィックス一覧が含まれます。この変数を設定した場合には、他の **MAVEN_REPO_*** 変数の名前に **DEV_MAVEN_REPO_URL** や **QE_MAVEN_REPO_URL** などのプリフィックスを含める必要があります。	**DEV,QE**

名前	説明	例
MAVEN_SETTINGS_XML	使用するカスタムの Maven ファイル **settings.xml** の場所。	**/root/.m2/settings.xml**
prefix_MAVEN_MIRROR_ID	指定のミラーに使用する ID。省略する場合には、一意の ID が生成されます。	**internal-mirror**
prefix_MAVEN_MIRROR_OF	このミラーでミラリングされるリポジトリー ID。デフォルト値は **external:*** です。	**external:*,!my-repo**
prefix_MAVEN_MIRROR_URL	ミラーの URL。	**http://10.0.0.1:8080/repository/internal**
prefix_MAVEN_REPO_HOST	Maven リポジトリーのホスト名。	**repo.example.com**
prefix_MAVEN_REPO_ID	Maven リポジトリー ID。	**my-repo**
prefix_MAVEN_REPO_LAYOUT	Maven リポジトリーのレイアウト。	**default**
prefix_MAVEN_REPO_USERNAME	Maven リポジトリーのユーザー名。	**mavenUser**
prefix_MAVEN_REPO_PASSPHRASE	Maven リポジトリーのパスフレーズ。	**maven1!**
prefix_MAVEN_REPO_PASSWORD	Maven リポジトリーのパスワード。	**maven1!**
prefix_MAVEN_REPO_PATH	Maven リポジトリーのパス。	**/maven2/**
prefix_MAVEN_REPO_PORT	Maven リポジトリーのポート。	**8080**
prefix_MAVEN_REPO_PRIVATE_KEY	Maven リポジトリーに接続するのに使用する秘密鍵へのローカルパス。	**${user.home}/.ssh/id_dsa**
prefix_MAVEN_REPO_PROTOCOL	Maven リポジトリーのプロトコル。	**http**
prefix_MAVEN_REPO_RELEASES_ENABLED	Maven リポジトリーのリリースが有効。	**true**

名前	説明	例
prefix_MAVEN_REPO_RELEASES_UPDATE_POLICY	Maven リポジトリーリリース更新ポリシー。	**always**
prefix_MAVEN_REPO_SERVICE	Maven リポジトリーの OpenShift サービス。この値は、URL または host/port/protocol が指定されていない場合に使用します。	**buscentr-myapp**
prefix_MAVEN_REPO_SNAPSHOTS_ENABLED	Maven リポジトリーのスナップショットが有効。	**true**
prefix_MAVEN_REPO_SNAPSHOTS_UPDATE_POLICY	Maven リポジトリースナップショット更新ポリシー。	**always**
prefix_MAVEN_REPO_URL	Maven リポジトリーの完全修飾 URL	**http://repo.example.com:8080/maven2/**

第16章 HA CEP クライアントの作成

CEP クライアントコードを HA CEP サーバーイメージと通信できるように、適応する必要があります。お使いのクライアントコード向けの参照実装に含まれるサンプルプロジェクトを使用してください。また、OpenShift 環境内外を問わず、クライアントコードを実行できます。

手順

1. Red Hat カスタマーポータルの Software Downloads ページから製品配信可能ファイル **rhdm-7.9.1-reference-implementation.zip** をダウンロードします。

2. ファイルの内容を展開してから、**rhdm-7.9.1-openshift-drools-hacep-distribution.zip** ファイルを展開します。

3. **openshift-drools-hacep-distribution/sources** ディレクトリーに移動します。

4. **sample-hacep-project/sample-hacep-project-client** ディレクトリーのサンプルプロジェクトをもとにクライアントコードをレビューし、変更します。このコードが *17章HA CEP クライアントおよびサーバーコードの要件* に記載の追加要件を満たしていることを確認します。

5. *15章Mave リポジトリーを使用した HA CEP サーバーを実装して KJAR サービスを更新する手順* で説明した方法を使用する実装で KJAR バージョンを更新するには、以下のコードのように、**UpdateKJarGAV** 呼び出しをクライアントに追加します。

```
        TopicsConfig envConfig = TopicsConfig.getDefaultTopicsConfig();
        Properties props = getProperties();
        try (RemoteStreamingKieSession producer =
RemoteStreamingKieSession.create(props, envConfig)){
            producer.updateKJarGAV("org.kie:fake-jar:0.1");
        }
```

   この呼び出しの実行時に、GAV を指定した KJAR が Maven リポジトリーで利用できるようにしてください。

6. **sample-hacep-project/sample-hacep-project-client** ディレクトリーで、パスワードに **password** と指定してキーストアを生成します。以下のコマンドを入力します。

```
keytool -genkeypair -keyalg RSA -keystore src/main/resources/keystore.jks
```

7. OpenShift 環境から HTTPS 証明書を展開して、キーストアーに追加します。以下のコマンドを実行します。

```
oc extract secret/my-cluster-cluster-ca-cert --keys=ca.crt --to=- > src/main/resources/ca.crt
keytool -import -trustcacerts -alias root -file src/main/resources/ca.crt -keystore
src/main/resources/keystore.jks -storepass password -noprompt
```

8. プロジェクトの **src/main/resources** サブディレクトリーで、**configuration.properties** ファイルを開き、**<bootstrap-hostname>** を Kafka サーバーのルートが提供するアドレスに置き換えます。

9. 標準の Maven コマンドを使用してプロジェクトをビルドします。

```
mvn clean install
```

10. **sample-hacep-project-client** プロジェクトのディレクトリーに移動して、以下のコマンドを入力し、クライアントを実行します。

```
mvn exec:java -Dexec.mainClass="org.kie.hacep.sample.client.ClientProducerDemo"
```

このコマンドは、**ClientProducerDemo** クラスの **main** メソッドを実行します。

第17章 HA CEP クライアントおよびサーバーコードの要件

高可用性 CEP のクライアントおよびサーバーコードを開発する場合には、以下のような特定の追加要件に準拠します。

kie-remote API

クライアントコードは、**kie** API ではなく **kie-remote** API を使用する必要があります。**kie-remote** API は、**org.kie:kie-remote** Maven アーティファクトに指定します。また、ソースコードは、Maven モジュール **kie-remote** にあります。

明示的なタイムスタンプ

デシジョンエンジンは、イベントの発生する順番を決定する必要があります。このような理由から、イベントには必ず、タイムスタンプを割り当てます。高可用性環境では、イベントをモデル化する JavaBean のプロパティーに、このタイムスタンプを指定します。次に、イベントクラスに **@Timestamp** アノテーションをつける必要があります。以下の例のように、ここではタイムスタンプ属性自体の名前がパラメーターとなります。

```
@Role(Role.Type.EVENT)
@Timestamp("myTime")
public class StockTickEvent implements Serializable {

    private String company;
    private double price;
    private long myTime;
}
```

タイムスタンプ属性を指定しない場合には、クライアントがリモートセッションにイベントを挿入するタイミングをもとに、Drools が全イベントにタイムスタンプを割り当てます。ただし、このメカニズムは、クライアントマシンのクロックにより異なります。異なるクライアント間でクロックにずれがある場合は、このようなホストが挿入したイベント間で不整合が発生する可能性があります。

メモリー以外のアクションの Lambda 式

作業メモリーアクション (デシジョンエンジンの作業メモリー内の情報を挿入、変更、または削除するアクション) は、クラスターの全ノードで処理する必要があります。メモリーアクションではないアクションは、リーダーでのみ実行する必要があります。

たとえば、今回のコードには、以下のルールが含まれます。

```
rule FindAdult when
  $p : Person(age >= 18)
then
  modify($p) { setAdult(true) }; // working memory action
  sendEmailTo($p); // side effect
end
```

このルールがトリガーされると、対象となる人は、すべてのノードで大人としてマークする必要があります。ただし、送信されるメール数が 1 通だけとなるように、リーダーだけがメールを送信できます。

そのため、以下の例のように、lambda 式のメールアクション (**副作用** と呼ばれる) をラップします。

```
rule FindAdult when
  $p : Person(age >= 18)
then
```

```
  modify($p) { setAdult(true) };
  DroolsExecutor.getInstance().execute( () -> sendEmailTo($p) );
end
```

付録A バージョン情報

本書の最終更新日: 2022 年 3 月 8 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