Red Hat Process Automation Manager 7.7

Red Hat OpenShift Container Platform への Red Hat Process Automation Manager オーサリング環境のデプロイ

ガイド

Last Updated: 2021-07-18

Red Hat Process Automation Manager 7.7 Red Hat OpenShift Container Platform への Red Hat Process Automation Manager オーサリング環境のデプロイ

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

本書は、Red Hat OpenShift Container Platform に Red Hat Process Automation Manager 7.7 オーサリング環境をデプロイする方法を説明します。

目次

はじめに

システムエンジニアは Red Hat OpenShift Container Platform に Red Hat Process Automation Manager オーサリング環境をデプロイして、サービス、プロセスアプリケーションおよびその他のビジネスアセットを開発するプラットフォームを提供できます。

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。
- OpenShift クラスター/namespace で 4 ギガバイト以上のメモリーが利用可能である。
- 高可用性のデプロイメントでは、以下のリソースが OpenShift クラスターで利用可能である。
  - Business Central で複製された Pod の場合、8 ギガバイトのメモリーと 2 CPU コアが各レプリカに必要です。デフォルトで 2 つのレプリカが作成されます。
  - KIE Server で複製された Pod の場合、1 ギガバイトのメモリーと 1 CPU コアが各レプリカに必要です。デフォルトで 2 つのレプリカが作成されます。
  - Red Hat Data Grid で複製された Pod の場合、2 ギガバイトのメモリーと 1 CPU コアが各レプリカに必要です。デフォルトで 2 つのレプリカが作成されます。
  - Red Hat AMQ で複製された Pod は、クラスターに設定されたデフォルトのリソース制限を使用します。
  - MySQL で複製された Pod は、クラスターに設定されたデフォルトのリソース制限を使用します。

注記

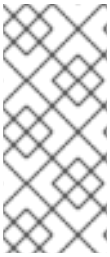
クラスターの容量を確認する方法は、Red Hat OpenShift Container Platform 3.11 製品ドキュメントの「クラスター容量の分析」を参照してください。

- 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 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化されている。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。
- 高可用性 Business Central を含む高可用性オーサリング環境をデプロイする場合、OpenShift 環境は **ReadWriteMany** モードの永続ボリュームをサポートします。ご使用の環境がこのモードに対応していない場合、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 Process Automation Manager バージョン 7.5 以降、Red Hat OpenShift Container Platform 3.x のサポートは非推奨となっています。この機能は今後のリリースで削除されます。



注記

Red Hat Process Automation Manager テンプレートを Red Hat OpenShift Container Platform 4.x と共に使用しないでください。Red Hat Process Automation 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 Process Automation Manager 環境のデプロイ』の説明を参照してください**。

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

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

この場合に、Red Hat Process Automation 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 Process Automation 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、プロセスアプリケーション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。
  通常、KIE Server にはデータベースサーバーが必要です。別の OpenShift Pod にデータベースサーバーを提供したり、別のデータベースサーバーを使用するように OpenShift で実行サーバーを設定したりできます。また、KIE Server では H2 データベースを使用できますが、使用する場合は、Pod をスケーリングできません。

  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 をスケールアップまたはスケールダウンすると、そのコピーはすべて同じデータベースサーバーサービスを使用し、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、管理および監視コンソールを提供します。Business Central を使用してサービスを開発して Process Server にそれらのサービスをデプロイできます。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。また、Business Central を使用してプロセスの実行を監視することもできます。
  Business Central は一元化アプリケーションですが、高可用性用に設定できます。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)。

- Business Central Monitoring は Web ベースの管理および監視コンソールです。KIE Server へのサービスのデプロイメントを管理し、監視情報を提供しますが、オーサリング機能は含まれません。このコンポーネントを使用して、ステージングおよび実稼働環境を管理できます。

- Smart Router は、KIE Server と、KIE Server と対話するその他のコンポーネントとの間の任意のレイヤーです。環境に、複数の KIE Server で実行するサービスが多数含まれる場合、Smart Router はすべてのクライアントアプリケーションに対応するエンドポイントを 1 つ提供します。クライアントアプリケーションは、サービスを要求する REST API 呼び出しを実行できます。Smart Router は、特定の要求を処理できる KIE Server を自動的に呼び出します。

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。

- **オーサリング**: Business Central を使用してサービスを作成し、変更するために使用する環境です。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 で構成されます。この環境のデプロイメント手順については、『Red HatOpenShift Container Platform への Red Hat Process Automation Manager オーサリング環境のデプロイ』を参照してください。

- **管理対象のデプロイメント**: ステージングおよび実稼働用として既存のサービスを実行するために使用する環境。この環境には、KIE Server Pod のいくつかのグループが含まれます。このようなすべてのグループに対してサービスのデプロイおよびデプロイ解除を実行します。必要に応じてこれらのグループのスケールアップおよびスケールダウンを実行できます。Business Central Monitoring を使用してサービスをデプロイし、実行し、停止し、またそれらの実行を監視します。
  2 種類の管理環境をデプロイすることができます。**自由形式** のサーバー環境では、最初に Business Central Monitoring と KIE Server を 1 つデプロイします。自由形式 のサーバー環境では、最初に Business Central Monitoring と 1 つの KIE Server をデプロイします。Business Central Monitoring は、同じ名前空間内のすべてのサーバーに接続できます。**この環境のデプロイメント手順については、『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager フリーフォーム環境のデプロイ』**を参照してください。

  または、**固定の** 管理サーバー環境をデプロイすることもできます。単一デプロイメントには、Business Central Monitoring、 Smart Router、および事前に設定された数の KIE Server (デフォルトでは 2 サーバーですが、テンプレートを変更して数を変更することができます) が含まれます。サーバーの追加や削除は後のプロセスでは容易に行えなくなります。この環境のデプロイメント手順については、『Red HatOpenShift Container Platform への Red Hat Process Automation Manager 固定管理サーバー環境のデプロイ』を参照してください。

- **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ので、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。オプションとして、Business Central Monitoring を使用して環境のパフォーマンスを監視したり、サービスインスタンスの一部を停止および再起動したりできますが、追加のサービスを KIE Server にデプロイしたり、既存のサービスのデプロイを解除したりすることはできません (コンテナーの追加または削除はできません)。この環境のデプロイメント手順については、『Red HatOpenShift Container Platform への Red Hat Process Automation Manager イミュータブルサーバー環境のデプロイ』を参照してください。

試用 **または評価環境をデプロイすることも可能です**。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しないので、この環境で実行した作業内容は保存されません。この環境のデプロイメント手順については、『Red HatOpenShift Container Platform への Red Hat Process Automation Manager 試用環境のデプロイ』を参照してください。

OpenShift に Red Hat Process Automation Manager 環境をデプロイするには、Red Hat Process Automation Manager で提供されるテンプレートを使用できます。設定が環境に適したものになるようにテンプレートを変更できます。

第2章 OPENSHIFT 環境に RED HAT PROCESS AUTOMATION MANAGER をデプロイする準備

OpenShift 環境に Red Hat Process Automation Manager をデプロイする前に、タスク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、プロセスの新しいバージョン、または別のプロセス）をデプロイする場合は、このタスクを繰り返す必要はありません。

2.1. 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 Process Automation Manager コンポーネントを Red Hat OpenShift Container Platform にデプロイするには、OpenShift が Red Hat レジストリーから適切なイメージをダウンロードできることを確認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 rhpam-businesscentral | grep -F 7.7
$ oc get imagestreamtag -n openshift | grep -F rhpam-kieserver | grep -F 7.7
```

   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。

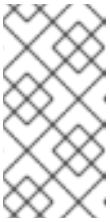
```
oc create -f <file_name>.yaml
oc secrets link default <secret_name> --for=pull
oc secrets link builder <secret_name> --for=pull
```

<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 **rhpam-7.7.0-openshift-templates.zip の製品配信可能ファイルをダウンロードし、rhpam77-image-streams.yaml** ファイルを展開します。

g. ターミナルで以下のコマンドを入力します。

```
$ oc apply -f rhpam77-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。

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、OpenShift 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 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. キーストアを **keystore.jks** ファイルに保存します。

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

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

5. **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
```

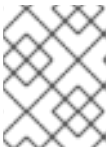
2.3. BUSINESS CENTRAL へのシークレットの作成

Business Central 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。

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。

Procedure

1. Business Central の SSL 暗号化の秘密鍵および公開鍵を使用して 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. キーストアを **keystore.jks** ファイルに保存します。
3. 証明書の名前をメモします。Red Hat Process Automation Manager 設定におけるこのデフォルト名は **jboss** です。
4. キーストアファイルのパスワードをメモします。Red Hat Process Automation Manager 設定におけるこのデフォルトの値は **mykeystorepass** です。
5. **oc** コマンドを使用して、新しいキーストアファイルからシークレット **businesscentral-app-secret** を生成します。

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

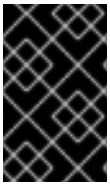
2.4. 管理ユーザーのシークレットの作成

Red Hat Process Automation 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、試用版テンプレート以外のテンプレートを使用して Red Hat Process Automation 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 Process Automation Manager のコンポーネントをデプロイする場合、これら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。

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、同じパスワードを持つ同じユーザーを Red Hat Process Automation Manager の **kie-server,rest-all,admin** ロールを使用して認証システムで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。

2.5. GLUSTERFS 設定の変更

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。たとえば、以下のようになります。

   ```
   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。たとえば、以下のようになります。

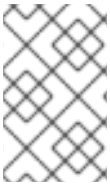
```
oc delete storageclass gluster-container
```

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
```

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。



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、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、『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。

2.7. 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別個のミラーリポジトリーとして使用します。

   または、サービスにカスタムの外部リポジトリー (Nexus など) を使用する場合、同じリポジトリーをミラーリポジトリーとして使用できます。

2. 公共の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、以下の手順を実行します。

   a. Red Hat カスタマーポータルの Software Downloads ページから **rhpam-7.7.0-offliner.zip** の製品配信可能ファイルをダウンロードします。

   b. **rhpam-7.7.0-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。アーティファクトのアップロードには、Maven repository tools Git リポジトリーから利用できる 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。アーティファクトのアップロードには、Maven repository tools Git リポジトリーから利用できる Maven Repository Provisioner ユーティリティーを使用できます。

2.8. 外部データベースのカスタム KIE SERVER 拡張イメージのビルド

KIE Server に外部データベースサーバーを使用し、そのデータベースサーバーが MySQL または PostgreSQL 以外の場合は、環境をデプロイする前にこのサーバー用のドライバーを使用するカスタムの KIE Server 拡張イメージをビルドする必要があります。

このビルド手順を実行して、以下のデータベースサーバーのドライバーを指定できます。

- Microsoft SQL Server
- MariaDB
- IBM DB2
- Oracle データベース
- Sybase

データベースサーバーのサポートされるバージョンについては、「Red Hat Process Automation Manager 7 でサポートされる構成」を参照してください。

ビルド手順では、既存の KIE Server イメージを拡張するカスタム拡張イメージを作成します。このカスタム拡張イメージは OpenShift 環境にインポートしてから、**EXTENSIONS_IMAGE** パラメーターで参照する必要があります。

前提条件

- **oc** コマンドを使用して OpenShift 環境にログインしている。OpenShift ユーザーには **registry-editor** ロールが必要です。
- Oracle Database または Sybase の場合は、データベースサーバーベンダーから JDBC ドライバーをダウンロードしている。
- 以下の必要なソフトウェアをインストールしている。
  - Docker
  - Cekit バージョン 3.2
  - Cekit の以下のライブラリーおよび拡張機能:
    - **odcs-client**: **python3-odcs-client** パッケージまたは同様のパッケージで提供される。
    - **docker**: **python3-docker** パッケージまたは同様のパッケージで提供される。
    - **docker-squash**: **python3-docker-squash** または同様のパッケージで提供される。
    - **behave**: **python3-behave** パッケージまたは同様のパッケージで提供される。
    - **S2i**: **Source-to-image** パッケージまたは同様のパッケージで提供される。

Procedure

1. IBM DB2、Oracle Database または Sybase の場合、JDBC ドライバー JAR ファイルをローカルディレクトリーに指定します。
2. Red Hat カスタマーポータルの Software Downloads ページから利用可能な **rhpam-7.7.0-openshift-templates.zip** の製品配信可能ファイルをダウンロードします。
3. ファイルを展開し、コマンドラインを使用して展開されたファイルの **templates/contrib/jdbc** ディレクトリーに移動します。このディレクトリーには、カスタムビルドのソースコードが含まれます。
4. データベースサーバーのタイプに応じて、以下のコマンドのいずれかを実行します。
   - Microsoft SQL Server の場合:

```
make build mssql
```

- MariaDB の場合:

```
make build mariadb
```

- IBM DB2 の場合:

```
make build db2
```

- Oracle Database の場合:

```
make build oracle artifact=/tmp/ojdbc7.jar version=7.0
```

このコマンドで、**/tmp/ojdbc7.jar** をダウンロードされた Oracle Database ドライバーのパス名に、**7.0** をドライバーのバージョンに置き換えます。

- Sybase の場合:

```
make build sybase artifact=/tmp/jconn4-16.0_PL05.jar version=16.0_PL05
```

このコマンドで、**/tmp/jconn4-16.0_PL05.jar** をダウンロードされた Sybase ドライバーのパス名に、 **16.0_PL05** をドライバーのバージョンに置き換えます。

5. 以下のコマンドを実行して、ローカルで利用可能な Docker イメージを一覧表示します。

```
docker images
```

ビルドされたイメージの名前 (例: **jboss-kie-db2-extension-openshift-image**) およびイメージのバージョンタグ (**11.1.4.4** など。**latest** タグではない) をメモします。

6. OpenShift 環境のレジストリーに直接アクセスし、イメージをレジストリーにプッシュします。ユーザーパーミッションに応じて、イメージを **openshift** namespace またはプロジェクト namespace にプッシュできます。レジストリーへのアクセスおよびイメージのプッシュの手順については、Red Hat OpenShift Container Platform 製品ドキュメントの「Accessing the Registry Directly 」を参照してください。

7. 外部データベースサーバーをサポートするテンプレートを使って KIE Server デプロイメントを設定する場合、以下のパラメーターを設定します。

- **Drivers Extension Image** (**EXTENSIONS_IMAGE**): 拡張イメージの ImageStreamTag 定義 (例: **jboss-kie-db2-extension-openshift-image:11.1.4.4**)

- **Drivers ImageStream Namespace** (**EXTENSIONS_IMAGE_NAMESPACE**): 拡張イメージのアップロード先の namespace (例: **openshift** またはプロジェクト namespace)

第3章 オーサリング環境

Business Central を使用してプロセスを作成および修正する環境をデプロイできます。オーサリング作業に使用する Business Central と、プロセスのテスト実行を行う KIE Server で構成されます。必要な場合は、追加の KIE Server を Business Central に接続できます。

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。KIE Server にはデフォルトで、組み込み済みの H2 データベースエンジンが含まれます。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。H2 データベースエンジンはスケーリングをサポートしていないため、デフォルトでは KIE Server をスケーリングすることもできません。ただし、別の MySQL またはPostgreSQL データベースサーバー Pod を使用するようにテンプレートを変更できます。この場合は、KIE Server をスケーリングできます。単一のオーサリング環境テンプレートを変更する手順については、「単一オーサリング環境のテンプレートの修正」 を参照してください。

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。データベースは別の Pod で提供されます。

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 Process Automation 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場合、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 KIE Server を追加でデプロイする場合には、「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 **のデプロイの手順については、『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager フリーフォーム環境のデプロイ 』**を参照してください。イミュータブル KIE Server **のデプロイ手順については、『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager イミュータブルサーバー環境のデプロイ 』**を参照してください。

3.1. 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 で構成されます。

3.1.1. オーサリング環境用のテンプレートの設定を開始する

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、rhpam77-authoring.yaml テンプレートファイルを使用します。デフォルトでは、単一オーサリングテンプレートは、永続的なストレージを持つ H2 データベースを使用します。MySQL または PostgreSQL Pod を作成するか、または外部データベースサーバー (OpenShift プロジェクト外) を使用することを選択する場合、環境をデプロイする前にテンプレートを変更します。テンプレートの変更に関する説明は、「単一オーサリング環境のテンプレートの修正」 を参照してください。

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、rhpam77-authoring-ha.yaml テンプレートファイルを使用します。デフォルトで、高可用性オーサリングテンプレートは MySQL Pod を使用して KIE Server のデータベースサーバーを提供します。PostgreSQL を使用するか、または外部サーバー (OpenShift プロジェクト外) を使用することを選択する場合、環境をデプロイする前にテンプレートを変更する必要があります。また、テンプレートを変更して Business Central 用に最初に作成されたレプリカの数を変更することもできます。テンプレートの変更に関する説明は、「高可用性オーサリング環境のテンプレートの修正」 を参照してください。

Procedure

1. Red Hat カスタマーポータルの Software Downloads ページから利用可能な **rhpam-7.7.0-openshift-templates.zip** の製品配信可能ファイル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** ファイルを選択するか、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
     BUSINESS_CENTRAL_HTTPS_SECRET=businesscentral-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。

3.1.2. 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を開始する」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。

Procedure

1. 以下のパラメーターを設定します。
   - **認証情報シークレット (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 Server Keystore Secret Name (BUSINESS_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 キーストアのシークレット名 (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 Server Certificate Name(BUSINESS_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 Server Keystore Password (BUSINESS_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 キーストアのパスワード (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別個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すでに利用可能な場合（「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）、名前空間は **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。

3.1.3. 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 namespace でイメージストリームを作成した場合、テンプレートで namespace 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を開始する」 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。

Procedure

「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。

3.1.4. 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を開始する」 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。

Procedure

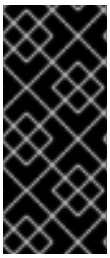
カスタム Maven リポジトリーへのアクセスを設定するには、以下のパラメーターを設定します。

- **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 Process Automation 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。

3.1.5. 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を開始する」 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。

Procedure

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 mirror 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** 値の設定方法については、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-rhpamcentr** です。外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合には、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。

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。

3.1.6. 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を開始する」 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。

Procedure

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、マウントする方法については、「(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。

3.1.7. 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(rhpam77-authoring-ha.yaml)をデプロイしている場合、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オプション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(rhpam77-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についての詳細は、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を開始する」 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。

Procedure

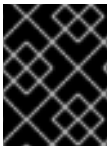
以下のパラメーターをテンプレートに設定します (該当する場合)。

- **Business Central Container Memory Limit**(**BUSINESS_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 JVM Max Memory Ratio** (**BUSINESS_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 Container CPU Limit**(**BUSINESS_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。

3.1.8. 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 Process Automation Manager のレルムが RH-SSO 認証システムに作成されていること。
- Red Hat Process Automation 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こと。利用可能なロールの一覧は、*4章Red Hat Process Automation 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。
- クライアントが、デプロイしている Red Hat Process Automation Manager 環境のすべてのコンポーネントについて RH-SSO 認証システムに作成されていること。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 Process Automation Manager デプロイメントは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を開始する」 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。

Procedure

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 Realm name** (**SSO_REALM**): Red Hat Process Automation 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には **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 Process Automation 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 Client name**(**BUSINESS_CENTRAL_SSO_CLIENT**): Business Central の RH-SSO クライアント名。
- **Business Central RH-SSO Client Secret**(**BUSINESS_CENTRAL_SSO_SECRET**): Business Central のクライアント向けに RH-SSO に設定されているシークレット文字列。
- **KIE Server RH-SSO Client name**(**KIE_SERVER_SSO_CLIENT**): KIE Server の RH-SSO クライアント名。
- **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): KIE Server のクライアントに対して RH-SSO に設定されているシークレットの文字列。

b. RH-SSO で Red Hat Process Automation Manager のクライアントを作成するには、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 Client name**(**BUSINESS_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 Client Secret**(**BUSINESS_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 Client name**(**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 Client Secret**(**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 Realm Admin Username**(**SSO_USERNAME**) および **RH-SSO Realm Admin Password** (**SSO_PASSWORD**): Red Hat Process Automation Manager の RH-SSO レルムのレルム管理者ユーザーのユーザー名およびパスワード。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 Process Automation 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。

3.1.9. 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 Process Automation Manager のユーザー名およびパスワードを作成していること。利用可能なロールの一覧は、*4章Red Hat Process Automation 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を開始する」 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。

Procedure

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 ログインモジュール」を参照してください。
LDAP サーバーでデプロイメントに必要なすべてのロールが定義されていない場合には、Red Hat Process Automation Manager ロールに LDAP グループをマップできます。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、「(オプション) 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。

3.1.10. オーサリング環境用に外部データベースサーバーを使用するためのパラメーターの設定

「単一オーサリング環境のテンプレートの修正」 または 「高可用性オーサリング環境のテンプレートの修正」 に説明されているように、KIE Server 用に外部データベースサーバーを使用するようにテンプレートを変更した場合、オーサリング環境をデプロイするようにテンプレートを設定する際に、以下の追加の設定を行います。

前提条件

- 「オーサリング環境用のテンプレートの設定を開始する」 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。

Procedure

1. 以下のパラメーターを設定します。

   - KIE Server External Database Driver(**KIE_SERVER_EXTERNALDB_DRIVER**): サーバーの種類に応じたサーバーのドライバー。

  - **mysql**
  - **postgresql**
  - **mariadb**
  - **mssql**
  - **db2**
  - **oracle**
  - **sybase**

- KIE Server External Database User(**KIE_SERVER_EXTERNALDB_USER**) および KIE Server External Database Password (**KIE_SERVER_EXTERNALDB_PWD**): 外部データベースサーバーのユーザー名およびパスワード。
- KIE Server External Database URL(**KIE_SERVER_EXTERNALDB_URL**): 外部データベースサーバーの JDBC URL。
- KIE Server External Database Host(**KIE_SERVER_EXTERNALDB_SERVICE_HOST**) および KIE Server External Database Port (**KIE_SERVER_EXTERNALDB_SERVICE_PORT**): 外部データベースサーバーのホスト名およびポート番号。これらのパラメーターを、**KIE_SERVER_EXTERNALDB_URL** パラメーターを設定する代わりに設定できます。
- KIE Server External Database Dialect(**KIE_SERVER_EXTERNALDB_DIALECT**): サーバーの種類に応じたサーバーの Hibernate ダイアレクト。
  - **org.hibernate.dialect.MySQL5InnoDBDialect** （MySQL および MariaDB で使用される）
  - **org.hibernate.dialect.PostgreSQL82Dialect**
  - **org.hibernate.dialect.SQLServer2012Dialect** (MS SQL で使用される)
  - **org.hibernate.dialect.DB2Dialect**
  - **org.hibernate.dialect.Oracle10gDialect**
  - **org.hibernate.dialect.SybaseASE157Dialect**
- KIE Server External Database name(**KIE_SERVER_EXTERNALDB_DB**): 外部データベースサーバーで使用するデータベース名。
- JDBC Connection Checker class (**KIE_SERVER_EXTERNALDB_CONNECTION_CHECKER**): データベースサーバーの JDBC connection checker class の名前。この情報がないと、データベースサーバー接続は、データベースサーバーの再起動時などで接続が失われた後に復元することができません。
- JDBC Exception Sorter class (**KIE_SERVER_EXTERNALDB_EXCEPTION_SORTER**): データベースサーバーの JDBC exception sorter class の名前。この情報がないと、データベースサーバー接続は、データベースサーバーの再起動時などで接続が失われた後に復元することができません。

2. 「外部データベースのカスタム KIE Server 拡張イメージのビルド」 で説明されているように、MySQL または PostgreSQL 以外の外部データベースサーバーを使用するためにカスタムイメージを作成している場合は、以下のパラメーターを設定します。

    - **Drivers Extension Image** (**EXTENSIONS_IMAGE**): 拡張イメージの ImageStreamTag 定義 (例: **jboss-kie-db2-extension-openshift-image:11.1.4.4**)

    - **Drivers ImageStream Namespace** (**EXTENSIONS_IMAGE_NAMESPACE**): 拡張イメージのアップロード先の namespace (例: **openshift** またはプロジェクト namespace)

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。

3.1.11. 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を開始する」 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。

Procedure

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については、「KIE Server の管理および監視」**を参照してください。

3.1.12. 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。

3.2. (オプション) 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。

Procedure

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。たとえば、以下のようになります。

```
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-rhpamcentr**: Business Central
- **myapp-kieserver**: KIE Server

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。

3.3. (オプション) GIT フックディレクトリーの指定

GIT_HOOKS_DIR パラメーターを設定した場合には、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。

Procedure

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>-rhpamcentr --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
```

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。たとえば、以下のようになります。

```
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>-rhpamcentr --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構成されているか、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、そのボリュームを要求に関連付けてから、ファイルをそのボリュームに転送し、ボリュームを **myapp-rhpamcentr** デプロイメント設定にマウントします (**myapp** は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については、「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については、「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。

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

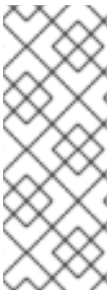
Red Hat Process Automation Manager オーサリングテンプレートを使用してデプロイされた環境では、Business Central が 1 つの KIE Server を管理します。高可用性オーサリングテンプレートを使用している場合や、単一のオーサリングテンプレートを変更して組み込み H2 データベース以外のデータベースサーバーを使用する場合、KIE Server Pod をスケーリングすることができますが、すべてのコピーが同じサービスを実行します。

Business Central に追加で KIE Server を接続できます。**ただし、rhpam77-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には**、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 namespace 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ので、**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。

rhpam77-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。



注記

Red Hat Process Automation Manager バージョン 7.7.0 **でのみ、rhpam77-authoring-ha.yaml** テンプレートで **OpenShiftStartupStrategy** 設定が有効化されていません。**このバージョンをデプロイした場合には、この手順を実行して、OpenShiftStartupStrategy 設定を有効にする必要があります**。高可用性オーサリング環境では、KIE Server 1 台または複数の KIE Server の安定した操作に、この設定が必要です。

前提条件

- **rhpam77-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こと。または、Red Hat Process Automation Manager バージョン 7.7.0（このバージョンのみ）で **rhpam77-authoring-ha.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こと。
- **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-rhpamcentr** 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-rhpamcentr 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-rhpamcentr** 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ていえる 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-rhpamcentr
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED=true
```

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

3.5. 単一オーサリング環境のテンプレートの修正

デフォルトでは、単一オーサリングテンプレートは、永続的なストレージを持つ H2 データベースを使用します。MySQL または PostgreSQL Pod を作成するか、または外部データベースサーバー (OpenShift プロジェクト外) を使用することを選択する場合、環境をデプロイする前にテンプレートを変更します。

KIE Server Pod をスケーリングする必要がある場合は、MySQL または PostgreSQL Pod または外部データベースサーバーを使用する必要があります。OpenShift テンプレートは、OpenShift が作成できる一連のオブジェクトを定義します。環境設定を変更するには、このオブジェクトの修正、追加、または削除が必要になります。このタスクを簡単にするために、Red Hat Process Automation Manager テンプレートにコメントが提供されます。

コメントの中には、テンプレート内のブロックを表すもの (**BEGIN** から **END** まで) があります。たとえば、以下のブロックの名前は **Sample block** です。

```
## Sample block BEGIN
sample line 1
sample line 2
sample line 3
## Sample block END
```

変更内容によっては、1 つのテンプレートファイルのブロックを、Red Hat Process Automation Manager で提供されている別のテンプレートファイルのブロックに置き換える必要があります。その場合は、ブロックを削除して新しいブロックを正しい場所に貼り付けます。

Procedure

必要に応じて、rhpam77-authoring.yaml テンプレートファイルを以下のように変更します。

- H2 データベースの代わりに MySQL を使用する場合は、ファイル内で、BEGIN コメントから **END コメントまでの数ブロックを、同じようにコメントがある rhpam77-kieserver-mysql.yaml** ファイルのブロックに置き換えます。その他のブロックを削除して、希望する場所にブロックを追加する必要もあります。

  1. **H2 database parameters** という名前のブロックを、**MySQL database parameters** に置き換えます。（このブロックと後続のすべての置換ブロックを **rhpam77-kieserver-mysql.yaml** ファイルから取得します）
  2. **H2 driver settings** ブロックを **MySQL driver settings** ブロックに置き換えます。
  3. **H2 persistent volume claim** ブロックを **MySQL persistent volume claim** ブロックに置き換えます。
  4. **H2 volume mount** ブロックと **H2 volume settings** ブロックを削除します。
  5. **Place to add database service** コメントの下に **MySQL service** ブロックを追加します。
  6. **Place to add database deployment config** コメントの下に **MySQL deployment config** ブロックを追加します。

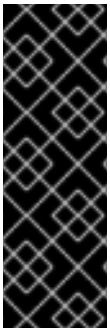
- H2 データベースの代わりに PostgreSQL を使用する場合は、ファイル内で、BEGIN コメントから **END コメントまでの数ブロックを、同じようにコメントがある rhpam77-kieserver-postgresql.yaml** ファイルのブロックに置き換えます。その他のブロックを削除して、希望す

る場所にブロックを追加する必要もあります。

1. **H2 database parameters** という名前のブロックを、**PostgreSQL database parameters** に置き換えます。（このブロックと後続のすべての置換ブロックを **rhpam77-kieserver-postgresql.yaml** ファイルから取得します）
2. **H2 driver settings** ブロックを **PostgreSQL driver settings** ブロックに置き換えます。
3. **H2 persistent volume claim** ブロックを **PostgreSQL persistent volume claim** ブロックに置き換えます。
4. **H2 volume mount** ブロックと **H2 volume settings** ブロックを削除します。
5. **Place to add database service** コメントの下に **PostgreSQL service** ブロックを追加します。
6. **Place to add database deployment config** コメントの下に **PostgreSQL deployment config** ブロックを追加します。

- 外部データベースサーバーを使用する場合は、ファイル内で、BEGIN コメントから **END コメントまでの数ブロックを、rhpam 77-kieserver-externaldb.yaml ファイルのブロックに置き換え、いくつかのブロックを削除します。**

1. **H2 database parameters という名前のブロックを、External database parameters という名前のブロックに置き換えます。（このブロックと後続のすべての置換ブロックを rhpam77-kieserver-externaldb.yaml ファイルから取得します）**
2. **H2 driver settings ブロックを、External database driver settings ブロックに置き換えます。**
3. **ファイルから、以下のブロックの BEGIN コメントから END コメントまでを削除します。**
   - **H2 persistent volume claim**
   - **H2 volume mount**
   - **H2 volume settings**



重要

標準の KIE Server イメージに外部データベースサーバー MySQL 用および PostgreSQL 用のドライバーが含まれます。別のデータベースサーバーを使用する場合は、カスタムの KIE Server イメージをビルドする必要があります。手順については、「外部データベースのカスタム KIE Server 拡張イメージのビルド」 を参照してください。

3.6. 高可用性オーサリング環境のテンプレートの修正

デフォルトで、高可用性オーサリングテンプレートは MySQL Pod を使用して KIE Server のデータベースサーバーを提供します。代わりに PostgreSQL、または (OpenShift プロジェクト外の) 外部サーバーを使用する場合は、環境をデプロイする前にテンプレートを修正する必要があります。

また、高可用性オーサリングテンプレートを変更して、Business Central に最初に作成したレプリカの数も変更できます。

OpenShift テンプレートは、OpenShift が作成できる一連のオブジェクトを定義します。環境設定を変更するには、このオブジェクトの修正、追加、または削除が必要になります。このタスクを簡単にするために、Red Hat Process Automation Manager テンプレートにコメントが提供されます。

コメントの中には、テンプレート内のブロックを表すもの (BEGIN から END まで) があります。たとえば、以下のブロックの名前は Sample block です。

```
## Sample block BEGIN
sample line 1
sample line 2
sample line 3
## Sample block END
```

変更内容によっては、1 つのテンプレートファイルのブロックを、Red Hat Process Automation Manager で提供されている別のテンプレートファイルのブロックに置き換える必要があります。その場合は、ブロックを削除して新しいブロックを正しい場所に貼り付けます。

Procedure

必要に応じて、rhpam77-authoring-ha.yaml テンプレートファイルを以下のように変更します。

- MySQL の代わりに PostgreSQL を使用する場合は、ファイル内で、BEGIN コメントから END コメントまでの数ブロックを、rhpam 77-kieserver-postgresql.yaml ファイルのブロッ

クに置き換えます。

1. MySQL database parameters という名前のブロックを、PostgreSQL database parameters に置き換えます。（このブロックと後続のすべての置換ブロックを rhpam77-kieserver-postgresql.yaml ファイルから取得します）

2. MySQL service ブロックを PostgreSQL service ブロックに置き換えます。

3. MySQL driver settings ブロックを PostgreSQL driver settings ブロックに置き換えます。

4. MySQL deployment config ブロックを PostgreSQL deployment config ブロックに置き換えます。

5. MySQL persistent volume claim ブロックを PostgreSQL persistent volume claim ブロックに置き換えます。

- 外部データベースサーバーを使用する場合は、ファイル内で、BEGIN コメントから END コメントまでの数ブロックを、rhpam 77-kieserver-externaldb.yaml ファイルのブロックに置き換え、いくつかのブロックを削除します。

1. MySQL database parameters という名前のブロックを、External database parameters という名前のブロックに置き換えます。（このブロックと後続のすべての置換ブロックを rhpam77-kieserver-externaldb.yaml ファイルから取得します）

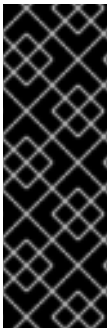
2. MySQL driver settings ブロックを External database driver settings ブロックに置き換えます。

3. ファイルから、以下のブロックの BEGIN コメントから END コメントまでを削除します。

   - MySQL service

   - MySQL deployment config

- **MySQL persistent volume claim**



重要

標準の KIE Server イメージに外部データベースサーバー MySQL 用および PostgreSQL 用のドライバーが含まれます。別のデータベースサーバーを使用する場合は、カスタムの KIE Server イメージをビルドする必要があります。手順については、「外部データベースのカスタム KIE Server 拡張イメージのビルド」 を参照してください。

- ## Replicas for Business Central コメントの下の行に、Business Central に最初に作成したレプリカの数を変更する場合は、レプリカの数を希望する値に変更します。

第4章 RED HAT PROCESS AUTOMATION 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これらのユーザーに適切なロールを割り当てる必要があります。

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. のメッセージが表示されます。

このセクションでは、利用可能な Red Hat Process Automation Manager ユーザーロールについて説明します。

注記

admin、analyst、developer、manager、process-admin、user、および rest-all のロールは Business Central 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にすべてアクセスできます。admin ロールを持つユーザーは、Red Hat Process Automation 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。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。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。

- **developer: developer ロールを持つユーザーは、ほぼすべての機能にアクセスができ、ルール、モデル、プロセスフロー、フォーム、およびダッシュボードを管理できます。アセットリポジトリーを管理し、プロジェクトを作成、ビルド、およびデプロイでき、Red Hat CodeReady Studio を使用してプロセスを表示できます。developer ロールが割り当てられているユーザーには、新規リポジトリーの作成やクローン作成などの、特定の管理機能は表示されません。**

- **manager: manager ロールを持つユーザーはレポートを表示できます。このユーザーは通常、ビジネスプロセス、そのパフォーマンス、ビジネスインジケーター、その他のビジネス関連のレポートに関する統計に関心があります。このルールを持つユーザーがアクセスできるのはプロセスおよびタスクのレポートに限られます。**

- **process-admin: process-admin ロールを持つユーザーは、ビジネスプロセス管理者です。ビジネスプロセス、ビジネスタスク、および実行エラーへの完全アクセスがあります。このユーザーは、ビジネスレポートを表示でき、タスク受信箱リストにアクセスできます。**

- **user: user ロールを持つユーザーは、タスクの受信箱リストで有効です。これには、現在実行しているプロセスの一部であるビジネスタスクも含まれます。このルールを持つユーザーはプロセスとタスクのレポートを確認して、プロセスを管理できます。**

- **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。このロールは、Business Central で Manage ビューおよび Track ビューにアクセスするユーザーにとって必須となります。**

第5章 OPENSHIFT テンプレートの参考資料

Red Hat Process Automation Manager は以下の OpenShift テンプレートを提供します。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、rhpam-7.7.0-openshift-templates.zip の製品配信可能ファイルをダウンロードし、これを展開します。

- rhpam77-authoring.yaml は Business Central および Business Central に接続された KIE Server を提供します。KIE Server は永続ストレージを持つ H2 データベースを使用します。この環境を使用してプロセス、サービス、およびその他のビジネスアセットのオーサリングを実行できます。このテンプレートの詳細は、「rhpam77-authoring.yaml template」 を参照してください。

- rhpam77-authoring-ha.yaml は高可用性 Business Central、Business Central に接続された KIE Server、および KIE Server が使用する MySQL インスタンスを提供します。この環境を使用してプロセス、サービス、およびその他のビジネスアセットのオーサリングを実行できます。このテンプレートの詳細は、「rhpam77-authoring-ha.yaml template」 を参照してください。

5.1. RHPAM77-AUTHORING.YAML TEMPLATE

Red Hat Process Automation Manager 7.7 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

5.1.1. 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。詳細情報は、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

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_TOKEN	**KIE_SERVER_CONTROLLER_TOKEN**	KIE Server コントローラートークン: ベアラー認証用 (org.kie.server.controller.token システムプロパティーを設定します)	–	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。例: クエリー (org.kie.server.bypass.auth.user システムプロパティーを設定します)	false	False
KIE_SERVER_PERSISTENCE_DS	**RHPAM_JNDI**	KIE Server 永続データソース。 (org.kie.server.persistence.ds システムプロパティーを設定)	java:/jboss/datasources/rhpam	False
KIE_SERVER_H2_USER	**RHPAM_USERNAME**	KIE Server H2 データベースユーザー名。	sa	False
KIE_SERVER_H2_PWD	**RHPAM_PASSWORD**	KIE Server H2 データベースパスワード。	–	False

変数名	イメージの環境変数	説明	値の例	必須
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 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
BUSINESS_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Business 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>-rhpamcentr-<project>.<default-domain-suffix>)。	–	False
BUSINESS_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Business Central 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は空白にします (例: <application-name>-rhpamcentr-<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
BUSINESS_CENTRAL_HTTPS_SECRET	–	Business 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	businesscentral-app-secret	True
BUSINESS_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイル名	keystore.jks	False
BUSINESS_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
BUSINESS_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	–	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
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	シークレット内のキーストアファイル名	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
DB_VOLUME_CAPACITY	–	データベースボリュームの永続ストレージのサイズ。	1Gi	True

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します)	5000	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 Process Automation Manager イメージの ImageStream がインストールされている namespace。これらの ImageStreams は通常 OpenShift の namespace にインストールされています。ImageStream を異なる namespace/プロジェクトにインストールしている場合にのみこれを変更する必要があります。デフォルトは「openshift」です。	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「rhpam-kieserver-rhel8」です。	rhpam-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「7.7.0」です。	7.7.0	True

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Business 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には、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-rhpamcentr	False
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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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

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
BUSINESS_CENTRAL_VOLUME_CAPACITY	–	Business Central のランタイムデータ向けの永続ストレージのサイズ	1Gi	True
BUSINESS_CENTRAL_MEMORY_LIMIT	–	Business 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
BUSINESS_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Business Central RH-SSO クライアント名	–	False
BUSINESS_CENTRAL_SSO_SECRET	**SSO_SECRET**	Business 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	ldap://myldap.example.com	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、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれてい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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある 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

5.1.2. 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。

5.1.2.1. 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}-rhpamcentr	8080	http	すべての Business Central Web サーバーのポート

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

5.1.2.2. Routes

ルートとは、**www.example.com** など、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) で構成されます。詳細情報は、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-${APPLICATION_NAME}-rhpamcentr-http	なし	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**
${APPLICATION_NAME}-rhpamcentr-https	TLS パススルー	**${BUSINESS_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}**

5.1.2.3. 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。詳細情報は、Openshift ドキュメントを参照してください。

5.1.2.3.1. 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。詳細情報は、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}-rhpamcentr	ImageChange
${APPLICATION_NAME}-kieserver	ImageChange

5.1.2.3.2. 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}-rhpamcentr	1
${APPLICATION_NAME}-kieserver	1

5.1.2.3.3. Pod テンプレート

5.1.2.3.3.1. 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。詳細情報は、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
${APPLICATION_NAME}-rhpamcentr	**${APPLICATION_NAME}-rhpamsvc**
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-rhpamsvc**

5.1.2.3.3.2. イメージ

デプロイメント	イメージ
${APPLICATION_NAME}-rhpamcentr	rhpam-businesscentral-rhel8
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

5.1.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhpamcentr

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

${APPLICATION_NAME}-kieserver

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

5.1.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhpamcentr

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

${APPLICATION_NAME}-kieserver

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

5.1.2.3.3.5. 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}-rhpamcentr	jolokia	8778	**TCP**
	http	8080	**TCP**

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

5.1.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-rhpamcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	–	–
	KIE_ADMIN_PWD	–	–
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	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}**

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)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL (ミリ秒単位)。(org.kie.server.controller.template.cache.ttl システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE Server コントローラートークン: ベアラー認証用 (org.kie.server.controller.token システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	MAVEN_MIRROR_URL	Business Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**

デプロイメント	変数名	説明	値の例
	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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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)。	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/businesscentral-secret-volume**
	HTTPS_KEYSTORE	シークレット内のキーストアファイル名	**${BUSINESS_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${BUSINESS_CENTRAL_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	**${BUSINESS_CENTRAL_HTTPS_PASSWORD}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhpamcentr**
	SSO_URL	RH-SSO URL。	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	**${SSO_REALM}**
	SSO_SECRET	Business Central RH-SSO クライアントシークレット	**${BUSINESS_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Business Central RH-SSO クライアント名	**${BUSINESS_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	Business 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>-rhpamcentr-<project>.<default-domain-suffix>)。	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Business Central の https サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: <application-name>-rhpamcentr-<project>.<default-domain-suffix>)。	**${BUSINESS_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${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}**

デプロイメント	変数名	説明	値の例
	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、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれてい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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある 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}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhpamcentr**
	DATASOURCES	–	**RHPAM**
	RHPAM_DATABASE	–	rhpam7

デプロイメント	変数名	説明	値の例
	RHPAM_JNDI	KIE Server 永続データソース。(org.kie.server.persistence.ds システムプロパティーを設定)	**${KIE_SERVER_PERSISTENCE_DS}**
	RHPAM_JTA	–	true
	RHPAM_DRIVER	–	h2
	RHPAM_USERNAME	KIE Server H2 データベースユーザー名。	**${KIE_SERVER_H2_USER}**
	RHPAM_PASSWORD	KIE Server H2 データベースパスワード。	**${KIE_SERVER_H2_PWD}**
	RHPAM_NONXA	–	false
	RHPAM_XA_CONNECTION_PROPERTY_URL	–	jdbc:h2:/opt/kie/data/h2/rhpam;AUTO_SERVER=TRUE
	KIE_SERVER_PERSISTENCE_DIALECT	–	org.hibernate.dialect.H2Dialect
	KIE_ADMIN_USER	–	–
	KIE_ADMIN_PWD	–	–
	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナーでアーティファクトのバージョンを変更することはできません。(org.kie.server.mode システムプロパティーを設定)	**${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)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。例: クエリー (org.kie.server.bypass.auth.user システムプロパティーを設定します)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhpamcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_PERSISTENCE_DS	KIE Server 永続データソース。(org.kie.server.persistence.ds システムプロパティーを設定)	**${KIE_SERVER_PERSISTENCE_DS}**
	KIE_SERVER_STARTUP_STRATEGY	–	ControllerBasedStartupStrategy

デプロイメント	変数名	説明	値の例
	MAVEN_MIRROR_URL	Business 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には、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	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_ID	–	repo-rhpamcentr
	RHPAMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhpamcentr**
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHPAMCENTR_MAVEN_REPO_USERNAME	–	–
	RHPAMCENTR_MAVEN_REPO_PASSWORD	–	–

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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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名	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	サーバー証明書に関連付けられている名前	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	**${KIE_SERVER_HTTPS_PASSWORD}**
	SSO_URL	RH-SSO URL。	**${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	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${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、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれてい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。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${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}**

5.1.2.3.3.7. ボリューム

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

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

5.1.2.4. 外部の依存関係

5.1.2.4.1. 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。詳細情報は、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}-rhpamcentr-claim	ReadWriteOnce
${APPLICATION_NAME}-kie-claim	ReadWriteOnce

5.1.2.4.2. 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。

businesscentral-app-secret kieserver-app-secret

5.2. RHPAM77-AUTHORING-HA.YAML TEMPLATE

Red Hat Process Automation Manager 7.7 向けの、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)

5.2.1. 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。詳細情報は、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**	KIE Server コントローラートークン: ベアラー認証用 (org.kie.server.controller.token システムプロパティーを設定します)	–	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。例: クエリー (org.kie.server.bypass.auth.user システムプロパティーを設定します)	false	False
KIE_SERVER_PERSISTENCE_DS	**KIE_SERVER_PERSISTENCE_DS**	KIE Server 永続データソース。(org.kie.server.persistence.ds システムプロパティーを設定)	java:/jboss/datasources/rhpam	False
MYSQL_USER	**RHPAM_USERNAME**	MySQL データベースユーザー名。	rhpam	False
MYSQL_PWD	**RHPAM_PASSWORD**	MySQL データベースパスワード。	–	False
MYSQL_DB	**RHPAM_DATABASE**	MySQL データベース名。	rhpam7	False

変数名	イメージの環境変数	説明	値の例	必須
MYSQL_DB_VOLUME_CAPACITY	–	KIE Server データベースボリュームの永続ストレージのサイズ。	1Gi	True
MYSQL_IMAGE_STREAM_NAMESPACE	–	MySQL イメージの ImageStream がインストールされている namespace。ImageStream は openshift namespace にすでにインストールされています。ImageStream を異なる namespace/プロジェクトにインストールしている場合にのみこれを変更する必要があります。デフォルトは「openshift」です。	openshift	False
MYSQL_IMAGE_STREAM_TAG	–	MySQL イメージのバージョン。デフォルトは「5.7」です。	5.7	False
KIE_SERVER_MYSQL_DIALECT	**KIE_SERVER_PERSISTENCE_DIALECT**	KIE Server MySQL Hibernate 方言。	org.hibernate.dialect.MySQL57Dialect	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)	**DEVELOPMENT**	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
BUSINESS_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Business 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>-rhpamcentr-<project>.<default-domain-suffix>)。	–	False
BUSINESS_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Business Central 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は空白にします (例: <application-name>-rhpamcentr-<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
BUSINESS_CENTRAL_HTTPS_SECRET	–	Business 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	businesscentral-app-secret	True
BUSINESS_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	Business 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
BUSINESS_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	Business Central のサーバー証明書に関連付けられている名前	jboss	False
BUSINESS_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	Business 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
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

変数名	イメージの環境変数	説明	値の例	必須
DATAGRID_IMAGE	–	DataGrid イメージ	registry.redhat.io/jboss-datagrid-7/datagrid73-openshift:1.4	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:7.5	True
AMQ_ROLE	–	標準ブローカーユーザーのユーザーロール。	admin	True
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

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**	OpenShift 内部サービスエンドポイント経由で KIE Server への接続を有効にします。 (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します)	5000	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 Process Automation Manager イメージの ImageStream がインストールされている namespace。これらの ImageStreams は通常 OpenShift の namespace 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
BUSINESS_CENTRAL_IMAGE_STREAM_NAME	–	Business Central に使用するイメージストリームの名前。デフォルトは「rhpam-businesscentral-rhel8」です。	rhpam-businesscentral-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「rhpam-kieserver-rhel8」です。	rhpam-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「7.7.0」です。	7.7.0	True

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Business 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には、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-rhpamcentr	False
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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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

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
TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	**TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL**	EJB タイマーのデータベースデータストアサービスの更新間隔を設定します。	60000	True
BUSINESS_CENTRAL_VOLUME_CAPACITY	–	Business Central のランタイムデータ向けの永続ストレージのサイズ	1Gi	True
BUSINESS_CENTRAL_MEMORY_LIMIT	–	Business Central コンテナーのメモリー制限	8Gi	True

変数名	イメージの環境変数	説明	値の例	必須
BUSINESS_CENTRAL_JAVA_MAX_MEM_RATIO	**JAVA_MAX_MEM_RATIO**	Business 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
BUSINESS_CENTRAL_CPU_LIMIT	–	Business 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
BUSINESS_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Business Central RH-SSO クライアント名	–	False
BUSINESS_CENTRAL_SSO_SECRET	**SSO_SECRET**	Business 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

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。	ldap://myldap.example.com	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、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれてい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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある 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。ファイルのすべてのエントリーの形式は 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

5.2.2. 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。

5.2.2.1. 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}-rhpamcentr	8080	http	すべての Business Central Web サーバーのポート
	8443	https	
${APPLICATION_NAME}-rhpamcentr-ping	8888	ping	rhpamcentr クラスタリングの JGroups ping ポート
${APPLICATION_NAME}-datagrid-ping	8888	ping	クラスタリング向けの JGroups ping ポート
${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 クラスタリングの JGroups ping ポート
${APPLICATION_NAME}-mysql	3306	–	MySQL サーバーのポート。

5.2.2.2. Routes

ルートとは、www.example.com など、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) で構成されます。詳細情報は、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-${APPLICATION_NAME}-rhpamcentr-http	なし	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**

サービス	セキュリティー	ホスト名
${APPLICATION_NAME}-rhpamcentr-https	TLS パススルー	**${BUSINESS_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}**

5.2.2.3. 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。詳細情報は、Openshift ドキュメントを参照してください。

5.2.2.3.1. 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。詳細情報は、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}-rhpamcentr	ImageChange
${APPLICATION_NAME}-kieserver	ImageChange
${APPLICATION_NAME}-mysql	ImageChange

5.2.2.3.2. 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}-rhpamcentr	2
${APPLICATION_NAME}-kieserver	2
${APPLICATION_NAME}-mysql	1

5.2.2.3.3. Pod テンプレート

5.2.2.3.3.1. 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。詳細情報は、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
${APPLICATION_NAME}-rhpamcentr	**${APPLICATION_NAME}-rhpamsvc**
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-rhpamsvc**

5.2.2.3.3.2. イメージ

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

5.2.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhpamcentr

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

${APPLICATION_NAME}-kieserver

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

${APPLICATION_NAME}-mysql

/bin/sh -i -c MYSQL_PWD="$MYSQL_PASSWORD" mysql -h 127.0.0.1 -u $MYSQL_USER -D $MYSQL_DATABASE -e 'SELECT 1'

5.2.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhpamcentr

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

${APPLICATION_NAME}-kieserver

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

${APPLICATION_NAME}-mysql

tcpSocket on port 3306

5.2.2.3.3.5. 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}-rhpamcentr	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
${APPLICATION_NAME}-mysql	–	3306	**TCP**

5.2.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
${APPLICATION_NAME}-rhpamcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	–	–
	KIE_ADMIN_PWD	–	–
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	false

デプロイメント	変数名	説明	値の例
	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)。	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	OpenShift 内部サービスエンドポイント経由で KIE Server への接続を有効にします。(org.kie.server.controller.openshift.prefer.kieserver.service システムプロパティーを設定します)	**${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します)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE Server コントローラートークン: ベアラー認証用 (org.kie.server.controller.token システムプロパティーを設定します)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhpamcentr**
	MAVEN_MIRROR_URL	Business Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**

デプロイメント	変数名	説明	値の例
	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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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)。	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/businesscentral-secret-volume**
	HTTPS_KEYSTORE	Business Central のシークレット内のキーストアファイルの名前	**${BUSINESS_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	Business Central のサーバー証明書に関連付けられている名前	**${BUSINESS_CENTRAL_HTTPS_NAME}**

デプロイメント	変数名	説明	値の例
	HTTPS_PASSWORD	Business Central のキーストアおよび証明書のパスワード	**${BUSINESS_CENTRAL_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-rhpamcentr-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	APPFORMER_INFINISPAN_SERVICE_NAME	–	**${APPLICATION_NAME}-datagrid**
	APPFORMER_INFINISPAN_PORT	–	11222
	APPFORMER_JMS_BROKER_ADDRESS	–	**${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	Business 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。	**${BUSINESS_CENTRAL_JAVA_MAX_MEM_RATIO}**
	SSO_URL	RH-SSO URL。	**${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。	**${SSO_REALM}**
	SSO_SECRET	Business Central RH-SSO クライアントシークレット	**${BUSINESS_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Business Central RH-SSO クライアント名	**${BUSINESS_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者ユーザー名 (存在しない場合)	**${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。	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Business Central の http サービスルートのカスタムホスト名。デフォルトホスト名は空白にします (例: insecure-<application-name>-rhpamcentr-<project>.<default-domain-suffix>)。	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**

デプロイメント	変数名	説明	値の例
	HOSTNAME_HTTPS	Business Central 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は空白にします (例: <application-name>-rhpamcentr-<project>.<default-domain-suffix>)。	**${BUSINESS_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。	**${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、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれてい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}**
	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}**

デプロイメント	変数名	説明	値の例
	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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある 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}**
	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。ファイルのすべてのエントリーの形式は 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}-rhpamcentr**
	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	EJB タイマーのデータベースデータストアサービスの更新間隔を設定します。	**${TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL}**
	DATASOURCES	–	**RHPAM**
	RHPAM_DATABASE	MySQL データベース名。	**${MYSQL_DB}**
	RHPAM_DRIVER	–	mariadb
	RHPAM_USERNAME	MySQL データベースユーザー名。	**${MYSQL_USER}**
	RHPAM_PASSWORD	MySQL データベースパスワード。	**${MYSQL_PWD}**
	RHPAM_SERVICE_HOST	–	**${APPLICATION_NAME}-mysql**

デプロイメント	変数名	説明	値の例
	RHPAM_SERVICE_PORT	–	3306
	KIE_SERVER_PERSISTENCE_DIALECT	KIE Server MySQL Hibernate 方言。	**${KIE_SERVER_MYSQL_DIALECT}**
	KIE_SERVER_PERSISTENCE_DS	KIE Server 永続データソース。(org.kie.server.persistence.ds システムプロパティーを設定)	**${KIE_SERVER_PERSISTENCE_DS}**
	RHPAM_JNDI	KIE Server 永続データソース。(org.kie.server.persistence.ds システムプロパティーを設定)	**${KIE_SERVER_PERSISTENCE_DS}**
	RHPAM_JTA	–	true
	KIE_ADMIN_USER	–	–
	KIE_ADMIN_PWD	–	–
	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)	**${KIE_MBEANS}**
	KIE_SERVER_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
	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルタリング。(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。例: クエリー (org.kie.server.bypass.auth.user システムプロパティーを設定します)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhpamcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_STARTUP_STRATEGY	–	ControllerBasedStartupStrategy
	MAVEN_MIRROR_URL	Business Central および KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	**${MAVEN_MIRROR_URL}**

デプロイメント	変数名	説明	値の例
	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_ID	–	repo-rhpamcentr
	RHPAMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhpamcentr**
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHPAMCENTR_MAVEN_REPO_USERNAME	–	–
	RHPAMCENTR_MAVEN_REPO_PASSWORD	–	–
	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には、MAVEN_MIRROR_OF に追加して、オプションで設定したミラーから除外できます (例: external:*,!repo-rhpamcentr,!repo-custom)。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL が設定されているが MAVEN_MIRROR_ID が設定されていない場合には、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}**

デプロイメント	変数名	説明	値の例
	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。	**${KIE_SERVER_HTTPS_PASSWORD}**
	SSO_URL	RH-SSO URL。	**${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}**

デプロイメント	変数名	説明	値の例
	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}**
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	**${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)。	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれている場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	**${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。	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	**${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。ファイルのすべてのエントリーの形式は 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}**
${APPLICATION_NAME}-mysql	**MYSQL_USER**	MySQL データベースユーザー名。	**${MYSQL_USER}**
	MYSQL_PASSWORD	MySQL データベースパスワード。	**${MYSQL_PWD}**
	MYSQL_DATABASE	MySQL データベース名。	**${MYSQL_DB}**

5.2.2.3.3.7. ボリューム

デプロイメント	名前	mountPath	目的	readOnly
${APPLICATION_NAME}-rhpamcentr	businesscentral-keystore-volume	**/etc/businesscentral-secret-volume**	ssl certs	True
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True
${APPLICATION_NAME}-mysql	**${APPLICATION_NAME}-mysql-pvol**	**/var/lib/mysql/data**	mysql	false

5.2.2.4. 外部の依存関係

5.2.2.4.1. 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。詳細情報は、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}-rhpamcentr-claim	ReadWriteMany
${APPLICATION_NAME}-mysql-claim	ReadWriteOnce

5.2.2.4.2. 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。

businesscentral-app-secret kieserver-app-secret

5.2.2.4.3. クラスタリング

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、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5.1 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
```

5.3. 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 Process Automation Manager テンプレートをデプロイし、モニターし、管理し、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変更操作」を参照してください。

付録A バージョン情報

本書の最終更新日： 2021 年 6 月 25 日（金）