



Red Hat Process Automation Manager 7.0

Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform

Red Hat Process Automation Manager 7.0 Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform

Red Hat Customer Content Services
brms-docs@redhat.com

法律上の通知

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書は、Red Hat OpenShift Container Platform に Red Hat Process Automation Manager 7.0 オペレーティング環境をデプロイする方法を説明します。

目次

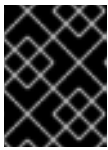
前書き	3
第1章 RED HAT OPENSIFT CONTAINER PLATFORM における RED HAT PROCESS AUTOMATION MANAGER の概要	4
第2章 OPENSIFT 環境に RED HAT PROCESS AUTOMATION MANAGER をデプロイする準備	6
2.1. イメージストリームが利用可能であることを確認	6
2.2. PROCESS SERVER にシークレットの作成	6
2.3. BUSINESS CENTRAL へのシークレットの作成	7
2.4. GLUSTERFS 設定の変更	7
第3章 オーサリング環境	9
3.1. 単一オーサリング環境のデプロイメント	9
3.2. 高可用性オーサリング環境のデプロイメント	12
3.3. 単一オーサリング環境のテンプレートの修正	15
3.4. 高可用性オーサリング環境のテンプレートの修正	17
3.5. 外部データベースへのカスタムの PROCESS SERVER イメージのビルド	18
付録A バージョン情報	21

前書き

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、プロセス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します。

前提条件

- OpenShift 環境で 4 ギガバイト以上のメモリーが利用できる。
- デプロイメントする OpenShift プロジェクトが作成されている。
- OpenShift Web コンソールと `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。
- Business Central または Business Central Monitoring の Pod をスケーリングする予定がある場合、OpenShift 環境では、ReadWriteMany モードで永続ボリュームがサポートされます。



重要

ReadWriteMany モードは、OpenShift Online および OpenShift Dedicated ではサポートされません。

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

OpenShift 環境をお持ちの場合は、その環境に Red Hat Process Automation Manager をデプロイできます。

この場合、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を提供できます。標準の OpenShift 方法を使用して Pod を管理し、負荷を分散できます。

以下の重要なコンポーネントが、OpenShift で実行する Pod として利用できます。

- **Process Server (Execution Server または KIE Server と呼ばれます)** は、プロセスまたはプロセスのグループを実行するインフラストラクチャー要素です。すべてのプロセスロジックが実行サーバーで実行します。

通常、Process Server にはデータベースサーバーが必要です。別の OpenShift Pod で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、Process Server は H2 データベースを使用できます。この場合、Pod をスケーリングすることはできません。

Process Server Pod は自由に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に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プロセスで実行し、同じデータベースサーバーを使用します。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。

異なるプロセスセットを実行する場合は、別の Process Server Pod をデプロイしますが、スケールアップまたはスケールダウンが可能になります。レプリケートした Process Server Pod を必要な数だけ持つことができます。

- **Business Central** は、オーサリングプロセスに対する 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 を使用してプロセスを開発して Process Server にプロセスをデプロイし、実行を監視します。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。その場合は複数の Pod が実行し、同じデータを共有します (現在のバージョンでは、高可用性機能はテクノロジープレビューです)。

Business Central には開発しているプロセスのソースを維持する Git リポジトリと、ビルドインの Maven リポジトリが含まれます。設定に応じて、Business Central はコンパイルしたプロセス (KJAR ファイル) をビルドイン Maven リポジトリに配置できます (設定した場合は外部 Maven リポジトリにも可能)。

- **Business Central Monitoring** は Web ベースの管理および監視コンソールです。Process Servers へのプロセ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、Process Server と、Process 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。Business Central または Business Central Monitoring を使用して異なる複数の Process Server と対話する場合は必要になります。また、環境に、複数の Process Server で実行するプロセ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、プロセスを要求する REST API コールを作成できます。Smart Router は、特定の要求に対してどの Process Server を呼び出す必要があるかを自動的に決定します。

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。Red Hat Process Automation Manager で提供されたテンプレートを使用して、最も一般的な組み合わせをデプロイできます。

以下の環境タイプが一般的です。

- **Authoring**: An environment for creating and modifying processes using Business Central. It consists of pods that provide Business Central for the authoring work and a Process Server for test execution of the processes. For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform](#).
- **Managed deployment**: An environment for running existing processes for staging and production purposes. This environment includes several groups of Process Server pods; you can deploy and undeploy processes on every such group and also scale the group up or down as necessary. Use Business Central Monitoring to deploy, run, and stop the processes and to monitor their execution. For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager managed server environment on Red Hat OpenShift Container Platform](#).
- **Deployment with immutable servers**: An alternate environment for running existing processes for staging and production purposes. In this environment, when you deploy a Process Server pod, it builds an image that loads and starts a process or group of processes. You cannot stop any process on the pod or add any new process to the pod. If you want to use another version of a process or modify the configuration in any other way, you deploy a new server image and displace the old one. In this system, you can use typical container-based integration workflows and do not need to use any other tools to manage the pods. Optionally, you can use Business Central Monitoring to monitor the performance of the environment and to stop and restart some of the process instances, but not to deploy additional processes to any Process Server or undeploy any existing ones (you can not add or remove containers). For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform](#).

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。環境が要件を満たすように、テンプレートを変更できます。

第2章 OPENSIFT 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、準備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が利用可能であることを確認

OpenShift 環境で、デプロイメントに必要なイメージストリームが利用可能であることを確認する必要があります。一部の OpenShift 環境には、必要なイメージストリームが含まれているため、それが利用可能かどうかを確認する必要があります。利用できない場合は、`rhcam70-image-streams.yaml` ファイルをインストールする必要があります。

手順

1. 以下のコマンドを実行します。

```
$ oc get imagestreamtag -n openshift | grep rhcam70-businesscentral
$ oc get imagestreamtag -n openshift | grep rhcam70-kieserver
```

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利用できるため、これ以外の操作は必要ありません。

2. 1 つまたは両方のコマンドの結果が空の場合は、Red Hat Process Automation Manager 7.0 の [Software Downloads](#) ページから、製品の配信可能ファイル `rhcam-7.0.0-openshift-templates.zip` をダウンロードし、`rhcam70-image-streams.yaml` ファイルを展開します。以下のいずれかを行います。

- 以下のコマンドを実行します。

```
$ oc create -f rhcam70-image-streams.yaml
```

- OpenShift Web UI で **Add to Project** → **Import YAML / JSON** を選択し、ファイルを選択するか、ファイルの内容を貼り付けます。

2.2. PROCESS SERVER に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。詳細は、OpenShift ドキュメントの [「Secrets」](#) の章を参照してください。

Process Server に SSL 証明書を作成し、それをシークレットとして OpenShift 環境に提供します。

手順

1. Process Server 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。プロダクション環境で、期待する Process Server の URL に一致する有効な署名付き証明書を生成します。キーストアを `keystore.jks` ファイルに保存します。証明書の名前と、キーストアファイルのパスワードを記録します。
自己署名型または購入する SSL 証明書を使用してキーストアを作成する方法は [「Generate a SSL Encryption Key and Certificate」](#) を参照してください。

2. `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

OpenShift 環境に Business Central または Business Central Monitoring をデプロイする場合は、Business Central に SSL 証明書を作成し、OpenShift 環境にシークレットとして提供します。Business Central と Process Server には、同じ証明書とキーストアを使用しないでください。

手順

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。実稼働環境で、期待する Business Central の URL に一致する有効な署名付き証明書を生成します。キーストアを `keystore.jks` ファイルに保存します。証明書の名前と、キーストアファイルのパスワードを登録します。
自己署名型または購入する SSL 証明書を使用してキーストアを作成する方法は「[Generate a SSL Encryption Key and Certificate](#)」を参照してください。
2. `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. GLUSTERFS 設定の変更

OpenShift 環境が GlusterFS を使用して永続的なストレージボリュームを提供するかどうかを確認します。GlusterFS を使用している場合は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します。

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
```

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、変更する必要はありません。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の名前です。以下は例になります。

```
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. **volumeoptions** キーのある行に、**features.cache-invalidation on** および **performance.nl-cache on** の 2 つのパラメーターを追加します。以下は例になります。

```
volumeoptions: client.ssl off, server.ssl off, features.cache-invalidation on, performance.nl-cache 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の名前です。以下は例になります。

```
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
```

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 Process Server で構成されます。

要件に応じて、単一オーサリング環境、または高可用性 (HA) オーサリング環境にデプロイできます。

単一オーサリング環境には 2 つの Pod が含まれます。Pod の 1 つが Business Central を実行し、別の Pod が Process Server を実行します。Process Server には、組み込みインメモリー H2 データベースエンジンが含まれます。このような環境では、可能な限り最小のリソースを使用します。ただし、インメモリーデータベースであるため、Process Server の Pod を再起動すると、プロセス情報をすべて失います。

HA オーサリング環境には複数の Pod が含まれます。Business Central および Process Server は、並行で実行でき、永続ストレージを共有するスケラブルな Pod に提供されます。データベースは、個別の高可用性サービスにより提供されます。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。



重要

現在のバージョンでは、高可用性機能はテクノロジープレビューです。

3.1. 単一オーサリング環境のデプロイメント

単一オーサリング環境をデプロイするには、`rhcam-70-authoring.yaml` テンプレートファイルを使用します。製品の配信可能ファイル `rhcam-7.0.0-openshift-templates.zip` からこのファイルを展開できます。このファイルは、Red Hat Process Automation Manager 7.0 の [Software Downloads](#) ページからダウンロードできます。

テンプレートファイルで定義した環境を修正する場合は「[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。

手順

1. 以下の方法を使用してテンプレートをデプロイします。

- OpenShift Web UI では、**Add to Project** → **Import YAML / JSON** を選択し、`rhcam70-authoring.yaml` ファイルを選択またはその内容を貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
- OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/rhcam70-authoring.yaml -p  
BUSINESS_CENTRAL_HTTPS_SECRET=businesscentral-app-secret -p  
KIE_SERVER_HTTPS_SECRET=kieserver-app-secret
```

このコマンドラインで以下を行います。

- `<template-path>` を、ダウンロードしたテンプレートファイルのパスに置き換えます。

- 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。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。

2. 必要に応じて以下のパラメーターを設定します。

- **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 Keystore Secret Name (KIE_SERVER_HTTPS_SECRET)**: 「[Process Server にシークレットの作成](#)」で作成した Process Server のシークレットの名前。
- **Application Name (APPLICATION_NAME)**: OpenShift アプリケーションの名前。Business Central および Process Server のデフォルト URL で使用されます。また、OpenShift は、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各セットを作成するためにそのアプリケーション名を使用します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名は同じにすることはできません。
- **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)**: 「[Process Server にシークレットの作成](#)」で作成したキーストアの証明書名。
- **KIE Server Keystore Password (KIE_SERVER_HTTPS_PASSWORD)**: 「[Process Server に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 ID (KIE_SERVER_ID)**: Process Server が参加する Business Central のサーバーテンプレートの名前。このパラメーターには値を自由に設定できます。値を設定しないと、Process Server で Pod の再起動またはスケーリングが行われ、サーバーの各新規インスタンスを新しいテンプレートに追加できるようになります。
- **ImageStream Namespace (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が利用可能であることを確認](#)」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。
以下のユーザー名およびパスワードを設定することもできます。
- **KIE Admin User (KIE_ADMIN_USER)** および **KIE Admin Password (KIE_ADMIN_PWD)**: Business Central の管理ユーザーのユーザー名およびパスワード。
- **KIE Server User (KIE_SERVER_USER)** および **KIE Server Password (KIE_SERVER_PWD)**: Process Server に接続するのにクライアントアプリケーションが使用するユーザー名およびパスワード。
ビルドした KJAR ファイルを外部の Maven リポジトリに置く場合は、以下のパラメーターを設定します。
- **Maven repository URL (MAVEN_REPO_URL)**: Maven リポジトリの URL。

- **Maven repository username (MAVEN_REPO_USERNAME)**: Maven リポジトリのユーザー名。
 - **Maven repository password (MAVEN_REPO_PASSWORD)**: Maven リポジトリのユーザー名。
3. Process Server で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修正したら、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 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.MySQL5Dialect** (MySQL および MariaDB で使用される)
 - **org.hibernate.dialect.PostgreSQLDialect**
 - **org.hibernate.dialect.SQLServer2012Dialect** (MS SQL で使用される)
 - **org.hibernate.dialect.DB2Dialect**
 - **org.hibernate.dialect.Oracle12cDialect**
 - **org.hibernate.dialect.SybaseASE15Dialect**
 - **KIE Server External Database Host (KIE_SERVER_EXTERNALDB_HOST)**: 外部データベースサーバーのホスト名。
 - **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。
4. 「[外部データベースへのカスタムの Process Server イメージのビルド](#)」の説明通りに、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する場合は、KIE Server Image Stream Name (**KIE_SERVER_IMAGE_STREAM_NAME**)

パラメーターを、以下の値に設定します。

- Microsoft SQL Server の場合: **rhpm70-kieserver-mssql-openshift**
 - MariaDB の場合: **rhpm70-kieserver-mariadb-openshift**
 - IBM DB2 の場合: **rhpm70-kieserver-db2-openshift**
 - Oracle データベースの場合: **rhpm70-kieserver-oracle-openshift**
 - Sybase の場合: **rhpm70-kieserver-sybase-openshift**
5. 使用している方法に応じて、環境の作成を終了します。
- OpenShift Web UI の場合は **Create** をクリックします。
 - 完了し、コマンドラインを実行します。

3.2. 高可用性オーサリング環境のデプロイメント

高可用性オーサリング環境をデプロイするには、**rhpm-70-authoring-ha.yaml** テンプレートファイルを使用します。製品の配信可能ファイル **rhpm-7.0.0-openshift-templates.zip** からこのファイルを展開できます。このファイルは、Red Hat Process Automation Manager 7.0 の [Software Downloads](#) ページからダウンロードできます。

テンプレートファイルで定義した環境を修正する場合は「[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。



重要

現在のバージョンでは、高可用性機能はテクノロジープレビューです。

手順

1. 以下の方法を使用してテンプレートをデプロイします。
 - OpenShift Web UI では、**Add to Project** → **Import YAML / JSON** を選択し、**rhpm70-authoring-ha.yaml** ファイルを選択またはその内容を貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
 - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/rhpm70-authoring-ha.yaml -p
BUSINESS_CENTRAL_HTTPS_SECRET=businesscentral-app-secret -p
KIE_SERVER_HTTPS_SECRET=kieserver-app-secret
```

このコマンドラインで以下を行います。

- **<template-path>** を、ダウンロードしたテンプレートファイルのパスに置き換えます。

- 必要なパラメーターに設定するために必要な数だけ **-p PARAMETER=value** ペアを使用します。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。
2. 必要に応じて以下のパラメーターを設定します。
- **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 Keystore Secret Name (KIE_SERVER_HTTPS_SECRET)**: 「[Process Server にシークレットの作成](#)」で作成した Process Server のシークレットの名前。
 - **Application Name (APPLICATION_NAME)**: OpenShift アプリケーションの名前。Business Central および Process Server のデフォルト URL で使用されます。また、OpenShift は、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各セットを作成するためにそのアプリケーション名を使用します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名は同じにすることはできません。
 - **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)**: 「[Process Server にシークレットの作成](#)」で作成したキーストアの証明書名。
 - **KIE Server Keystore Password (KIE_SERVER_HTTPS_PASSWORD)**: 「[Process Server に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 ID (KIE_SERVER_ID)**: Process Server が参加する Business Central のサーバーテンプレートの名前。このパラメーターには値を自由に設定できます。値を設定しないと、Process Server で Pod の再起動またはスケリングが行われ、サーバーの各新規インスタンスを新しいテンプレートに追加できるようになります。
 - **ImageStream Namespace (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が利用可能であることを確認](#)」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。
以下のユーザー名およびパスワードを設定することもできます。
 - **KIE Admin User (KIE_ADMIN_USER)** および **KIE Admin Password (KIE_ADMIN_PWD)**: Business Central の管理ユーザーのユーザー名およびパスワード。
 - **KIE Server User (KIE_SERVER_USER)** および **KIE Server Password (KIE_SERVER_PWD)**: Process Server に接続するのにクライアントアプリケーションが使用するユーザー名およびパスワード。
ビルドした KJAR ファイルを外部の Maven リポジトリに置く場合は、以下のパラメーターを設定します。
 - **Maven repository URL (MAVEN_REPO_URL)**: Maven リポジトリの URL。

- **Maven repository username (MAVEN_REPO_USERNAME):** Maven リポジトリのユーザー名。
 - **Maven repository password (MAVEN_REPO_PASSWORD):** Maven リポジトリのユーザー名。
3. Process Server で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修正したら、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 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.MySQL5Dialect** (MySQL および MariaDB で使用される)
 - **org.hibernate.dialect.PostgreSQLDialect**
 - **org.hibernate.dialect.SQLServer2012Dialect** (MS SQL で使用される)
 - **org.hibernate.dialect.DB2Dialect**
 - **org.hibernate.dialect.Oracle12cDialect**
 - **org.hibernate.dialect.SybaseASE15Dialect**
 - **KIE Server External Database Host (KIE_SERVER_EXTERNALDB_HOST):** 外部データベースサーバーのホスト名。
 - **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。
4. 「外部データベースへのカスタムの Process Server イメージのビルド」の説明通りに、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する場合は、KIE Server Image Stream Name (**KIE_SERVER_IMAGE_STREAM_NAME**)

パラメーターを、以下の値に設定します。

- Microsoft SQL Server の場合: **rhcam70-kieserver-mssql-openshift**
 - MariaDB の場合: **rhcam70-kieserver-mariadb-openshift**
 - IBM DB2 の場合: **rhcam70-kieserver-db2-openshift**
 - Oracle データベースの場合: **rhcam70-kieserver-oracle-openshift**
 - Sybase の場合: **rhcam70-kieserver-sybase-openshift**
5. 使用している方法に応じて、環境の作成を終了します。
- OpenShift Web UI の場合は **Create** をクリックします。
 - 完了し、コマンドラインを実行します。

3.3. 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 の Pod を作成したり、外部データベースサーバー (OpenShift プロジェクト外) を使用する場合は、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。

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。

手順

必要に応じて、**rhcam70-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 **rhcam70-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** ブロックを、**MySQL database parameters** ブロックに置き換えます (**rhcam70-kieserver-mysql.yaml** 内のこのブロックと、その後の置換ブロックを使用します)。

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 **rhpm70-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があります。
 1. **H2 database parameters** ブロックを、**PostgreSQL database parameters** ブロックに置き換えます (**rhpm70-kieserver-postgresql.yaml** 内のこのブロックと、その後の置換ブロックを使用します)。
 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、**rhpm70-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** ブロックを **External database parameters** ブロックに置き換えます (**rhpm70-kieserver-externaldb.yaml** 内のこのブロックと、その後の置換ブロックを使用します)。
 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 Process 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 Process Server イメージをビルドする必要があります。手順は「[外部データベースへのカスタムの Process Server イメージのビルド](#)」を参照してください。

3.4. 高可用性オーサリング環境のテンプレートの修正

デフォルトでは、高可用性オーサリングテンプレートは、Process Server にデータベースサーバーを提供するために MySQL Pod を作成します。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
```

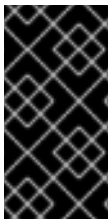
変更内容によっては、1つのテンプレートファイルのブロックを、Red Hat Process Automation Manager で提供されている別のテンプレートファイルのブロックに置き換える必要があります。その場合は、ブロックを削除して新しいブロックを正しい場所に貼り付けます。

手順

必要に応じて、**rhpan70-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、**rhpan70-kieserver-postgresql.yaml** ファイルのブロックに置き換えます。
 1. **MySQL database parameters** ブロックを、**PosgreSQL database parameters** ブロックに置き換えます (**rhpan70-kieserver-postgresql.yaml** 内のこのブロックと、その後の置換ブロックを使用します)。
 2. **MySQL service** ブロックを **PosgreSQL service** ブロックに置き換えます。
 3. **MySQL driver settings** ブロックを **PosgreSQL driver settings** ブロックに置き換えます。
 4. **MySQL deployment config** ブロックを **PosgreSQL deployment config** ブロックに置き換えます。
 5. **MySQL persistent volume claim** ブロックを **PosgreSQL 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、**rhpm70-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** ブロックを **External database parameters** ブロックに置き換えます (**rhpm70-kieserver-externaldb.yaml** 内のこのブロックと、その後の置換ブロックを使用します)。
 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 Process 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 Process Server イメージをビルドする必要があります。手順は「[外部データベースへのカスタムの Process Server 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、レプリカ数を希望する値に変更します。

3.5. 外部データベースへのカスタムの **PROCESS SERVER** イメージのビルド

Process Server に外部データベースサーバーを使用し、そのサーバーが 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 Process Server イメージをビルドする必要があります。

このビルド手順を使用して、以下のデータベースサーバーのドライバーを提供できます。

- 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テスト済みバージョンは、『[Red Hat JBoss Data Virtualization 7 でサポートされる構成](#)』を参照してください。

ビルド手順は、既存の Process Server イメージを拡張するカスタムイメージを作成します。ここでは、オリジナルイメージと同じバージョンタグを持つ **openshift** 名前空間の新しい **ImageStream** にこのカスタムイメージをプッシュします。

前提条件

- **cluster-admin** ロールを持つユーザーとして **oc** コマンドを実行し、OpenShift 環境のプロジェクトにログインしている。
- IBM DB2、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。

手順

1. IBM DB2、Oracle データベース、または Sybase の場合は、HTTP サーバーのローカルディレクトリーに JDBC ドライバーの JAR を提供します。ローカルディレクトリーまたは HTTP サーバーでは、以下のパスが期待されます。
 - IBM DB2 の場合:
<local_path_or_url>/com/ibm/db2/jcc/db2jcc4/10.5/db2jcc4-10.5.jar
 - Oracle データベースの場合:
<local_path_or_url>/com/oracle/ojdbc7/12.1.0.1/ojdbc7-12.1.0.1.jar
 - Sybase の場合: **<local_path_or_url>/com/sybase/jconn4/16.0_PL05/jconn4-16.0_PL05.jar**
<local_path_or_url> は、ローカルディレクトリーへのパス、またはドライバーを提供した HTTP サーバーの URL です。
2. カスタムビルドのソースコードをインストールするには、Red Hat Process Automation Manager 7.0 の [Software Downloads](#) ページから、製品の配信可能ファイル **rhpmam-7.0.0-openshift-templates.zip** をダウンロードします。ファイルを展開し、コマンドラインで、展開したファイルの **templates/contrib/jdbc** ディレクトリーに変更します。
3. 以下のサブディレクトリーを変更します。
 - Microsoft SQL Server の場合: **mssql-driver-image**
 - MariaDB の場合: **mariadb-driver-image**
 - IBM DB2 の場合: **db2-driver-image**
 - Oracle データベースの場合: **oracle-driver-image**
 - Sybase の場合: **sybase-driver-image**
4. 以下のコマンドを実行します。
 - Microsoft SQL Server または MariaDB の場合:

```
./build.sh
```
 - IBM DB2、Oracle Database、または Sybase の場合:

```
./build.sh --artifact-repo=<local_path_or_url>
```


<local_path_or_url> は、ローカルディレクトリーへのパス、またはドライバーを提供した HTTP サーバーの URL です。以下は例となります。

```
../build.sh --artifact-repo=/home/builder/drivers  
../build.sh --artifact-  
repo=http://nexus.example.com/nexus/content/groups/public
```

プロセスに OpenShift の Docker レジストリーを設定する場合は、**--registry=<registry_name.domain_name:port>** パラメーターも build コマンドに 追加します。

例:

```
../build.sh --registry=docker-registry.custom-domain:80  
  
../build.sh --artifact-repo=/home/builder/drivers --registry=docker-  
registry.custom-domain:80
```


付録A バージョン情報

Documentation last updated on: Monday, October 1, 2018.