



Red Hat Process Automation Manager 7.4

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

Red Hat Process Automation Manager 7.4 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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Deploying_a_Red_Hat_Process_Automation_Manager_authoring_environment_on_Red_Hat_Opfile |.

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, the Red Hat 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 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 7.4 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	4
第1章 RED HAT OPENSIFT CONTAINER PLATFORM における RED HAT PROCESS AUTOMATION MANAGER の概要	5
第2章 OPENSIFT 環境に RED HAT PROCESS AUTOMATION MANAGER をデプロイする準備	8
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	8
2.2. 高可用性デプロイメント用に AMQ スケールダウンコントローラーイメージストリームが利用可能であるかど うかの確認	9
2.3. PROCESS SERVER にシークレットの作成	10
2.4. BUSINESS CENTRAL へのシークレットの作成	11
2.5. GLUSTERFS 設定の変更	11
2.6. オフラインで使用する MAVEN ミラーリポジトリーの用意	13
2.7. 外部データベースのカスタム PROCESS SERVER 拡張イメージのビルド	14
第3章 オーサリング環境	17
3.1. オーサリング環境のデプロイメント	17
3.1.1. オーサリング環境用のテンプレートの設定開始	17
3.1.2. オーサリング環境に必要なパラメーターの設定	18
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	19
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	20
3.1.5. オーサリング環境のビルドイン Maven リポジトリーにアクセスするための認証情報の指定	21
3.1.6. 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	21
3.1.7. オーサリング環境用の Git フックディレクトリーの指定	22
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	22
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	24
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	25
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	27
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	27
3.2. (任意) LDAP ロールマッピングファイルの指定	28
3.3. (オプション) GIT フックディレクトリーの指定	29
3.4. 単一オーサリング環境のテンプレートの修正	30
3.5. 高可用性オーサリング環境のテンプレートの修正	32
第4章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー	34
第5章 OPENSIFT テンプレートの参考資料	36
5.1. RHPAM74-AUTHORING.YAML テンプレート	36
5.1.1. パラメーター	36
5.1.2. オブジェクト	51
5.1.2.1. サービス	51
5.1.2.2. ルート	52
5.1.2.3. デプロイメント設定	52
5.1.2.3.1. トリガー	52
5.1.2.3.2. レプリカ	53
5.1.2.3.3. Pod テンプレート	53
5.1.2.4. 外部の依存関係	73
5.1.2.4.1. ボリューム要求	73
5.1.2.4.2. シークレット	74
5.2. RHPAM74-AUTHORING-HA.YAML テンプレート	74
5.2.1. パラメーター	74
5.2.2. オブジェクト	92

5.2.2.1. サービス	92
5.2.2.2. ルート	93
5.2.2.3. デプロイメント設定	94
5.2.2.3.1. トリガー	94
5.2.2.3.2. レプリカ	94
5.2.2.3.3. Pod テンプレート	95
5.2.2.4. 外部の依存関係	116
5.2.2.4.1. ボリューム要求	117
5.2.2.4.2. シークレット	117
5.2.2.4.3. クラスターリング	117
5.3. OPENSIFT の使用に関するクイックリファレンス	118
付録A バージョン情報	121

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、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

- 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。
- デプロイメントに使用する OpenShift プロジェクトが作成されている。
- **oc** コマンドを使用してプロジェクトにログインしている。**oc** コマンドランツールに関する詳細は、OpenShift の [CLI リファレンス](#) を参照してください。OpenShift Web コンソールを使用してテンプレートをデプロイするには、Web コンソールを使用してログインしている必要もあります。
- 動的永続ボリューム (PV) のプロビジョニングが有効になっている。または、動的 PV プロビジョニングが有効でない場合は、十分な永続ボリュームが利用できる状態でなければなりません。デフォルトでは、以下のサイズが必要です。
 - 複製された Process 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 1Gi 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。
- Authoring High Availability テンプレートを使用して Business Central Pod をスケーリングする予定の場合には、お使いの OpenShift 環境で **ReadWriteMany** モードを使用した永続ボリュームをサポートする必要があります。OpenShift Online ボリュームプラグイン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。



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

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。

- **Process Server (実行サーバー (Execution Server) または KIE Server と呼ばれる)** は、意思決定サービス、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、Process 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、Process 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 をスケールアップできません。

Process 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。

Process 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 Process Server Pod を必要な数だけ設定することができます。

- **Business Central** は、オーサリングサービスに対する Web ベースのインタラクティブ環境です。また、管理および監視コンソールも提供します。Business Central を使用してサービスを開発し、Process 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。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 リポジトリが含まれます。また、ビルトインの Maven リポジトリも含まれます。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)。



重要

現在のバージョンでは、高可用性の Business Central 機能はテクノロジープレビュー機能となっています。Red Hat のテクノロジープレビュー機能のサポートの詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

- **Business Central Monitoring** は Web ベースの管理および監視コンソールです。Process 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、ステージング環境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。環境に、複数の Process 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 Process 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。この環境は、オーサリングの作業用の Business Central と、サービスのテスト実行用の Process Server 1 台で設定されます。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 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のに使用する環境。この環境には、Process Server Pod のいくつかのグループが含まれます。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できます。**自由形式**のサーバー環境では、最初に Business 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 1 つの Process Server をデプロイします。その後、追加として任意の数の KIE Server をデプロイできます。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 フリーフォーム管理サーバー環境のデプロイ』](#)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 Process 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。後のプロセスでは、サーバーを簡単に追加または削除することはできません。この環境のデプロイメント手順については、[『Red Hat OpenShift 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、Process 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、起動するイメージがビルドされます。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、Process 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できますが、追加のサービスを Process 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 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。

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 Process Server が含まれます。この環境はすばやく設定でき、これを使用して、アセットの開発や実行を評価し、体験できます。ただし、この環境では永続ストレージを使用せず、この環境でのいずれの作業も保存されません。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 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 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、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 rhpam74-businesscentral
$ oc get imagestreamtag -n openshift | grep rhpam74-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 **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 **rhpm-7.4.0-openshift-templates.zip** の製品配信可能ファイルをダウンロードし、**rhpm74-image-streams.yaml** ファイルを展開します。
- g. 以下のアクションの1つを完了します。
 - 以下のコマンドを入力します。

```
$ oc create -f rhpm74-image-streams.yaml
```

- OpenShift Web UI で、**Add to Project** → **Import YAML / JSON**を選択してからファイルを選択するか、またはその内容を貼り付けます。



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. 高可用性デプロイメント用に AMQ スケールダウンコントローラーイメージストリームが利用可能かどうかの確認

Red Hat Process Automation Manager 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、AMQ スケールダウンコントローラーの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。



重要

Red Hat Process Automation Manager 7.4 では、高可用性 Business Central 機能はテクノロジープレビュー機能となっています。AMQ スケールダウンコントローラーイメージもテクノロジープレビュー機能としてのみ利用可能です。Red Hat のテクノロジープレビュー機能のサポートの詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

前提条件

- 「[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および Red Hat Process Automation Manager のイメージレジストリーが利用可能であることを確認している。

手順

AMQ スケールダウンコントローラーのイメージストリームが利用可能であることを確認するには、以下の追加の手順を実行します。

1. AMQ スケールダウンコントローラーのイメージストリームが存在するかを確認します。以下のコマンドを入力します。

```
$ oc get imagestreamtag -n openshift | grep amq-broker-72-scaledown-controller-openshift
```

コマンドの出力が空白でない場合、必要なイメージストリームは **openshift** namespace で利用可能な状態であるため、これ以外の操作は必要ありません。

2. コマンドの出力が空の場合は、以下の手順を実行してください。
 - a. [amq-broker-7-scaledown-controller-image-streams.yaml](#) ファイルをダウンロードします。
 - b. 以下のアクションの1つを完了します。

- 以下のコマンドを入力します。

```
$ oc create -f amq-broker-7-scaledown-controller-image-streams.yaml
```

- OpenShift Web UI で **Add to Project** → **Import YAML / JSON** を選択し、**amq-broker-7-scaledown-controller-image-streams.yaml** ファイルを選択するか、このファイルの内容を貼り付けます。



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。この場合、テンプレートをデプロイする際に

AMQ_SCALEDOWN_CONTROLLER_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。

2.3. 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。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。

Process 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. 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。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、Process Server 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 **kieserver-app-secret** を生成します。

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

2.4. 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 Process Server に同じ証明書およびキーストアを使用しないでください。

手順

1. 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
```

2.5. 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、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. **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。以下に例を示します。

```
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. オフラインで使用する MAVEN ミラーリポジトリの用意

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 リポジトリを用意して、このリポジトリ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

- 書き込み可能な Maven リリースリポジトリを準備します。このリポジトリは、認証なしに読み込みアクセスを許可する必要があります。OpenShift 環境は、このリポジトリへのアクセスが必要です。OpenShift 環境に、Nexus リポジトリ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、「Nexus の [設定](#)」を参照してください。このリポジトリを別個のミラーリポジトリとして使用します。
または、サービスにカスタムの外部リポジトリ (Nexus など) を使用する場合、同じリポジトリをミラーリポジトリとして使用できます。
- 公開インターネットに送信アクセスができるコンピューターで、以下のアクションを実行します。
 - 最新バージョンの [Offliner ツール](#) をダウンロードします。
 - 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-7.4.0-offliner.txt** の製品配信可能ファイルをダウンロードします。
 - 以下のコマンドを入力して、Offliner ツールを使用し、必要なアーティファクトをダウンロードします。

```
java -jar offliner-<version>.jar -r https://maven.repository.redhat.com/ga/ -r https://repo1.maven.org/maven2/ -d /home/user/temp rhpam-7.4.0-offliner.txt
```

/home/user/temp は空の一時ディレクトリーに、**<version>** はダウンロードした Offliner ツールのバージョンに置き換えます。ダウンロードにはかなり時間がかかる可能性があります。

- このツールにより、ダウンロードの失敗が報告された場合には、以下のコマンドを入力して初回にダウンロードに失敗したアーティファクトをダウンロードします。

```
grep Path: errors.log | sed -n -e 's/^.*/Path: //p' > failed-downloads.txt
java -jar offliner-<version>.jar -r https://maven.repository.redhat.com/ga/ -r https://repo1.maven.org/maven2/ -d /home/user/temp failed-downloads.txt
```

もう一度ダウンロードの失敗が報告され、失敗数が初回のダウンロード合計数と比較して少しの場合には、続行してください。

- e. 一時ディレクトリーから作成した Maven 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 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 Provisioner](#) ユーティリティーを使用できます。

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

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 Process Server 拡張イメージをビルドする必要があります。

このビルド手順の手順を完了して、次のデータベースサーバーのいずれかにドライバーを提供します。

- 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でサポートされる設定](#) を参照してください。

ビルド手順では、既存の Process 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。

手順

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 **rhpm-7.4.0-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。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置き換え、**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置き換え、`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` 名前空間またはプロジェクト名前空間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、「[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 Process 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名前空間 (例: `openshift` またはプロジェクト名前空間)

第3章 オーサリング環境

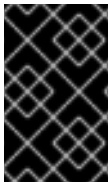
Business Central を使用してプロセスを作成および修正する環境をデプロイできます。オーサリング作業に使用する Business Central と、プロセスのテスト実行を行う Process Server で設定されます。

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。データベースは別の Pod で提供されます。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参与する場合に、信頼性と応答性を最大限提供します。

必要な場合は、追加の管理またはイミュータブル Process Server をデプロイすることもできます。Business Central は、イミュータブル Process Server および管理 Process Server を含む、同じ namespace の Process Server を自動的に検出します。この機能には、固定された管理インフラストラクチャーにデプロイされたものを除くすべての Process Server に対して有効にされた **OpenShiftStartupStrategy** 設定が必要です。**OpenShiftStartupStrategy** 設定が有効にされた状態で管理 Process Server をデプロイする手順については、『[Deploying a Red Hat Process Automation Manager freeform managed server environment on Red Hat OpenShift Container Platform](#)』を参照してください。イミュータブル Process Server のデプロイ手順については、『[Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform](#)』を参照してください。



重要

Red Hat Process Automation Manager 7.4 では、高可用性 Business Central 機能はテクノロジープレビュー機能となっています。Red Hat のテクノロジープレビュー機能のサポートの詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

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 Process Server で設定されます。

3.1.1. オーサリング環境用のテンプレートの設定開始

単一オーサリング環境をデプロイする必要がある場合は、**rhpam74-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は、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、**rhpam74-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。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。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。

手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページから利用可能な **rhcam-7.4.0-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** ファイルを選択または貼り付けます。Add Template ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
 - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/&lt;template-file-name&gt;.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](#)」の手順を実行し、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開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。

- **Business Central 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。
 - **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。
 - **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 Process Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。
 - **ImageStream 名前空間 (IMAGE_STREAM_NAMESPACE):** イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが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。
2. 以下のユーザー名とパスワードを設定できます。デフォルトでは、デプロイメントはパスワードを自動的に生成します。
- **KIE Admin User(KIE_ADMIN_USER) および KIE Admin Password(KIE_ADMIN_PWD):** 管理者ユーザーのユーザー名およびパスワード。Business Central を使用して同じテンプレートでデプロイされる Process Server 以外の Process Server を制御するか、またはモニターする場合、ユーザー名およびパスワードを設定し、これらを記録する必要があります。
 - **KIE Server User(KIE_SERVER_USER) および KIE Server Password (KIE_SERVER_PWD):** いずれかの 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できるユーザー名およびパスワード。

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。

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

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

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

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](#)」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 名前空間 (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。
2. 高可用性オーサリング環境をデプロイしており、「[高可用性デプロイメント用に AMQ スケールダウンコントローラーイメージストリームが利用可能であるかどうかの確認](#)」の説明に従って AMQ スケールダウンコントローラーイメージストリームの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、**AMQ Scaledown Controller ImageStream Namespace (AMQ_SCALEDOWN_CONTROLLER_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 リポジトリに配置する必要がある場合は、リポジトリにアクセスするためにパラメーターを設定する必要があります。

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

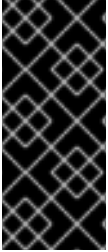
カスタム Maven リポジトリへのアクセスを設定するには、以下のパラメーターを設定します。

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

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ビルドイン Maven リポジトリにアクセスするための認証情報の指定

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に、Business Central に組み込まれている Maven リポジトリを使用し、追加の Process Server を Business Central に接続する必要がある場合、この Maven リポジトリにアクセスするための認証情報を設定する必要があります。次に、これらの認証情報を使用して Process Server を設定できます。

また、RH-SSO または LDAP 認証を設定している場合、ビルトイン Maven リポジトリの認証情報を、RH-SSO または LDAP で設定されるユーザー名およびパスワードに設定する必要があります。この設定は、Process Server が Maven リポジトリにアクセスできるようにするために必要です。

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

ビルトイン Maven リポジトリの認証情報を設定するには、以下のパラメーターを設定します。

- **Username for the Maven service hosted by Business Central**
(**BUSINESS_CENTRAL_MAVEN_USERNAME**): ビルドインの Maven リポジトリのユーザー名。
- **Password for the Maven service hosted by Business Central**
(**BUSINESS_CENTRAL_MAVEN_PASSWORD**): ビルドインの Maven リポジトリのパスワード。

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。

3.1.6. 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 ミラーリポジトリの用意](#)』に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

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

- **Maven 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。この値で、Maven は Business Central のビルトイン Maven リポジトリーからアーティファクトを直接取得し、ミラーから他の必要なアーティファクトを取得します。外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合は、このリポジトリー内のアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*;!repo-custom**)。 **repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。

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. オーサリング環境用の Git フックディレクトリーの指定

Git フックを使用して Business Central の内部 Git リポジトリーと外部 Git リポジトリーの対話を容易にすることができます。

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

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」 に説明されているようにテンプレートの設定を開始していること。

手順

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

- **Git フックディレクトリー (GIT_HOOKS_DIR)**: Git フックディレクトリーへの完全修飾パス (例: `/opt/kie/data/git/hooks`)。ディレクトリーの内容を指定し、これを指定されたパスにマウントする必要があります。設定マップまたは永続ボリュームを使用して Git フックディレクトリーを指定し、マウントする方法については、「[\(オプション\) 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。

3.1.8. 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。
- 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。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。以下のユーザーは、環境のパラメーターを設定するために必要です。
 - **kie-server,rest-all,admin** ロールを持つ管理者ユーザー。このユーザーは環境を管理し、これを使用できます。Process Server はこのユーザーを使用して Business Central で認証します。
 - **kie-server,rest-all,user** ロールを持つサーバーユーザー。このユーザーは、Process Server に対する REST API 呼び出しを実行できます。Business Central はこのユーザーを使用して Process Server で認証します。
- 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。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開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

1. テンプレートの **KIE_ADMIN_USER** および **KIE_ADMIN_PASSWORD** パラメーターを、RH-SSO 認証システムで作成したユーザー名およびパスワードに設定します。
2. テンプレートの **KIE_SERVER_USER** および **KIE_SERVER_PASSWORD** パラメーターを、RH-SSO 認証システムで作成したサーバーユーザーのユーザー名およびパスワードに設定します。
3. 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は **true** に設定します。
4. 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):** Process Server の RH-SSO クライアント名。
 - **KIE Server RH-SSO Client Secret(KIE_SERVER_SSO_SECRET):** Process Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
- b. RH-SSO 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):** Process 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):** Process 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almのレalm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。この環境のパラメーターを設定するために、少なくとも以

下のユーザーを作成している必要があります。

- **kie-server,rest-all,admin** ロールを持つ管理者ユーザー。このユーザーは環境を管理し、これを使用できます。
 - **kie-server,rest-all,user** ロールを持つサーバーユーザー。このユーザーは、Process Server に対する REST API 呼び出しを実行できます。
- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

1. LDAP サービスでは、デプロイメントパラメーターですべてのユーザー名を作成します。パラメーターを設定しない場合には、デフォルトのユーザー名を使用してユーザーを作成します。作成したユーザーにはロールに割り当てる必要もあります。
 - **KIE_ADMIN_USER**: デフォルトのユーザー名 **adminUser**、ロール: **kie-server,rest-all,admin**
 - **KIE_SERVER_USER**: デフォルトのユーザー名 **executionUser**、ロール **kie-server,rest-all,guest**
LDAP で設定可能なユーザーロールについては、「[ロール およびユーザー](#)」を参照してください。
2. テンプレートの **AUTH_LDAP*** パラメーターを設定します。これらのパラメーターは、Red Hat JBoss EAP の **LdapExtended** ログインモジュールの設定に対応します。これらの設定に関する説明は、[LdapExtended login module](#) を参照してください。
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、「[\(任意\) LDAP ロールマッピングファイルの指定](#)」を参照してください。
 - **RoleMapping replaceRole** プロパティ (**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、Process 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開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

1. 以下のパラメーターを設定します。

- **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.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 Host(KIE_SERVER_EXTERNALDB_SERVICE_HOST)**: 外部データベースサーバーのホスト名。
- **KIE Server External Database Port(KIE_SERVER_EXTERNALDB_SERVICE_PORT)**: 外部データベースサーバーのポート番号。

- **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 Process 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名前空間 (例: `openshift` またはプロジェクト名前空間)

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

Process 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 Process Server でこの機能のサポートを有効にします。

前提条件

- 「[オーサリング環境用のテンプレートの設定開始](#)」に説明されているようにテンプレートの設定を開始していること。

手順

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

次のステップ

必要な場合は、追加のパラメーターを設定します。

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

Prometheus メトリクス収集の方法については、「[Process 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. (任意) LDAP ロールマッピングファイルの指定

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

手順

1. **my-role-map** など、ロールマッピングのプロパティ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**: Process Server

myapp はアプリケーション名に置き換えます。複数の Process 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。

手順

1. SSH 認証を使用してアップストリームリポジトリーを操作する必要がある場合は、次の手順を実行して、必要なファイルを含むシークレットを作成してマウントします。
 - a. リポジトリーに格納されている公開鍵に一致する秘密鍵を使用して、**id_rsa** ファイルを作成します。
 - b. リポジトリーの正しい名前、アドレス、公開鍵で **known_hosts** ファイルを作成します。
 - c. 以下のように **oc** コマンドを使用して、2つのファイルでシークレットを作成します。

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

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

```
oc set volume dc/<myapp>-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設定されているか、または実行可能なファイルや KJAR ファイルなどのバイナリーに依存する場合は、永続ボリュームを使用します。永続ボリュームを作成し、永続ボリューム要求を作成して、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は、[永続ボリュームの使用](#) を参照してください。永続ボリュームへのファイルのコピー方法は、[Transferring files in and out of containers](#) を参照してください。
4. 数分待機してから、プロジェクト内の Pod の一覧およびステータスを確認します。Business Central は Git フックディレクトリーが指定されるまで開始されないの、Process 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 Process 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. 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、**rhpm74-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 **rhpm74-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 **rhpm74-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 **rhpm74-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** ブロックを、**PostgreSQL database parameters** 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 **rhpm74-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、**rhpm74-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 **rhpm74-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. このファイルの以下のブロックから、**BEGIN** から **END** までのコメントを削除します。
 - **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、「[外部データベースのカスタム Process Server 拡張イメージのビルド](#)」を参照してください。

3.5. 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。

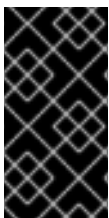
```
## Sample block BEGIN
sample line 1
sample line 2
sample line 3
## Sample block END
```

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、`rhpm74-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、`rhpm74-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。（このブロックと後続のすべての置換ブロックを `rhpm74-kieserver-postgresql.yaml` ファイルから取得します）
 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、`rhpm74-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。（このブロックと後続のすべての置換ブロックを `rhpm74-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. このファイルの以下のブロックから、**BEGIN** から **END** までのコメントを削除します。
 - **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、「[外部データベースのカスタム Process 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 Process Server にアクセスするには、サーバーを起動する前にユーザーを作成して適切なロールを割り当てます。

Business Central および Process 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 Process 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、Process Server にもアクセスできます。

ただし、Business Central と Process 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 に対して認証されたユーザーは、Process Server にアクセスするために個別に認証する必要があります（例：Business Central でプロセス定義を表示または管理する場合）。ユーザーが Process Server で認証されていない場合は、ログファイルに 401 エラーが記録され、Business Central に **Invalid credentials to load data from remote server.Contact your system administrator.** メッセージが表示されます。

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 Process Server 用に予約されています。このため、Business Central または Process 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、リポジトリの作成、クローン作成、および管理ができます。アプリケーションで必要な変更をすべて利用できます。**admin** ロールを持つユーザーは、Red Hat 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。これらは、プロジェクトのモデリングと実行を行うことができます。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。**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 JBoss Developer 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、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 Process Server (KIE サーバー) REST 機能へのアクセスがあります。このルールは、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必要になります。

第5章 OPENSIFT 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、**rhpmam-7.4.0-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。

- **rhpmam74-authoring.yaml** は Business Central および Business Central に接続された Process Server を提供します。Process 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を作成できます。このテンプレートの詳細は、「[rhpmam74-authoring.yaml テンプレート](#)」を参照してください。
- **rhpmam74-authoring-ha.yaml** は高可用性 Business Central および Business Central に接続された Process Server、および Process 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を作成できます。高可用性機能はテクノロジープレビューとして利用できます。このテンプレートの詳細は、「[rhpmam74-authoring-ha.yaml テンプレート](#)」を参照してください。

5.1. RHPAM74-AUTHORING.YAML テンプレート

Red Hat Process Automation Manager 7.4 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、パラメーターの参照時には、この値が代入されます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、[Openshift ドキュメント](#) を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
KIE_ADMIN_USER	KIE_ADMIN_USER	KIE 管理者のユーザー名	adminUser	False
KIE_ADMIN_PASSWORD	KIE_ADMIN_PASSWORD	KIE 管理者のパスワード	–	False
KIE_SERVER_CONTROLLER_USER	KIE_SERVER_CONTROLLER_USER	KIE サーバーコントローラーのユーザー名。 (org.kie.server.controller.user システムプロパティを設定)	controllerUser	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_PWD	KIE_SERVER_CONTROLLER_PWD	KIE サーバーコントローラーのパスワード。 (org.kie.server.controller.pwd システムプロパティを設定)	–	False
KIE_SERVER_CONTROLLER_TOKEN	KIE_SERVER_CONTROLLER_TOKEN	ベアラー認証用の KIE Server コントローラートークン。 (org.kie.server.controller.token システムプロパティを設定)	–	False
KIE_SERVER_USER	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティを設定)	executionUser	False
KIE_SERVER_PASSWORD	KIE_SERVER_PASSWORD	KIE サーバーのパスワード。 (org.kie.server.password システムプロパティを設定)	–	False
KIE_SERVER_BYPASS_AUTH_USER	KIE_SERVER_BYPASS_AUTH_USER	KIE サーバーはタスク関連の操作は認証済みユーザーをスキップできる。 (org.kie.server.bypass.auth.user システムプロパティを設定)	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を設定)	java:/jboss/datasources/rhpam	False
KIE_SERVER_H2_USER	RHPAM_USERNAME	KIE サーバー H2 データベースユーザー名。	sa	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_H2_PWD	RHPAM_PASSWORD	KIE Server H2 データベースパスワード。	–	False
KIE_SERVER_MODE	KIE_SERVER_MODE	KIE Server モード。有効な値は 'DEVELOPMENT' または 'PRODUCTION' です。実稼働モードでは、SNAPSHOT バージョンのアーティファクトは KIE Server にデプロイできず、既存のコンテナ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 kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。 (org.drools.server.filter.classes システムプロパティを設定)	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の名前。	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の名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	HTTPS_NAME	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	HTTPS_PASSWORD	キーストアおよび証明書のパスワード。	mykeystorepass	False
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を設定)	60000	False

変数名	イメージの環境変数	説明	値の例	必須
IMAGE_STREAM_NAMESPACE	–	Red Hat ミドルウェアイメージの ImageStreams がインストールされている namespace これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStreams を別の namespace/プロジェクトにインストールしている場合には、これを変更するだけで結構です。	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 rhpam74-kieserver-openshift です。	rhpam74-kieserver-openshift	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 1.0 です。	1.0	True
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、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

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	MAVEN_REPO_ID	Maven リポジトリに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False
MAVEN_REPO_URL	MAVEN_REPO_URL	Maven リポジトリ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 リポジトリにアクセスするユーザー名 (必要な場合)	—	False
MAVEN_REPO_PASSWORD	MAVEN_REPO_PASSWORD	Maven リポジトリにアクセスするパスワード (必要な場合)。	—	False
BUSINESS_CENTRAL_MAVEN_USERNAME	KIE_MAVEN_USER	EAP 内の Business Central がホストする Maven サービスにアクセスするためのユーザー名	mavenUser	True

変数名	イメージの環境変数	説明	値の例	必須
BUSINESS_CENTRAL_MAVEN_PASSWORD	KIE_MAVEN_PASSWORD	EAP 内の Business Central がホストする Maven サービスにアクセスするためのパスワード	–	True
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 コンテナのメモリー制限	2 Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナのメモリー制限	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
KIE_SERVER_SSO_CLIENT	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False

変数名	イメージの環境変数	説明	値の例	必須
SSO_USERNAME	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者のユーザー名 (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
AUTH_LDAP_BASE_CTX_DN	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	ou=users,ou=example,ou=com	False

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

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

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

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_RECURSION	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。 roleAttributelsDN 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_USE_ROLE_ATTRIBUTE_ID_TO_CHECK	AUTH_LDAP_REFERRAL_USE_ROLE_ATTRIBUTE_ID_TO_CHECK	リファール (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。リファールを使用し、ロールオブジェクトがリファール内部にあると、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	
	8001	git-ssh	

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

5.1.2.2. ルート

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、セキュリティ設定 (任意) で設定されます。詳細は、[OpenShift ドキュメント](#) を参照してください。

サービス	セキュリティ	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もとにするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、[OpenShift ドキュメント](#) を参照してください。

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。詳細は、[OpenShift ドキュメント](#) を参照してください。

Deployment	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、[コンテナエンジンのドキュメント](#)を参照してください。

Deployment	レプリカ
<code>\${APPLICATION_NAME}-rhpamcentr</code>	1
<code>\${APPLICATION_NAME}-kieserver</code>	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。詳細は、[Openshift ドキュメント](#)を参照してください。

Deployment	サービスアカウント
<code>\${APPLICATION_NAME}-rhpamcentr</code>	<code>\${APPLICATION_NAME}-rhpamsvc</code>
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${APPLICATION_NAME}-rhpamsvc</code>

5.1.2.3.3.2. イメージ

デプロイメント	イメージ
<code>\${APPLICATION_NAME}-rhpamcentr</code>	<code>rhpam74-businesscentral-openshift</code>
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${KIE_SERVER_IMAGE_STREAM_NAME}</code>

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
<code>\${APPLICATION_NAME}-rhpamcentr</code>	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
	git-ssh	8001	TCP
<code>\${APPLICATION_NAME}-kieserver</code>	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
<code>\${APPLICATION_NAME}-rhpamcentr</code>	KIE_ADMIN_USER	KIE 管理者のユーザー名	<code>\${KIE_ADMIN_USER}</code>
	KIE_ADMIN_PWD	KIE 管理者のパスワード	<code>\${KIE_ADMIN_PWD}</code>
	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 <code>kie.mbeans</code> および <code>kie.scanner.mbeans</code> を設定)	<code>\${KIE_MBEANS}</code>
	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 (<code>org.kie.server.controller.openshift.global.discovery.enabled</code> システムプロパティを設定)。	<code>\${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}</code>

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_OPENSIFT_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_OPENSIFT_PREFER_KIESERVER_SERVICE}`
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL (ミリ秒単位)。 (org.kie.server.controller.template.cache.ttl システムプロパティーを設定)	`\${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}`
	KIE_WORKBENCH_CONTROLLER_OPENSIFT_ENABLED	–	true
	KIE_SERVER_CONTROLLER_USER	KIE サーバーコントローラーのユーザー名。 (org.kie.server.controller.user システムプロパティーを設定)	`\${KIE_SERVER_CONTROLLER_USER}`
	KIE_SERVER_CONTROLLER_PWD	KIE サーバーコントローラーのパスワード。 (org.kie.server.controller.pwd システムプロパティーを設定)	`\${KIE_SERVER_CONTROLLER_PWD}`
	KIE_SERVER_CONTROLLER_TOKEN	ベアラー認証用の KIE Server コントローラートークン。 (org.kie.server.controller.token システムプロパティーを設定)	`\${KIE_SERVER_CONTROLLER_TOKEN}`
	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティーを設定)	`\${KIE_SERVER_USER}`

デプロイメント	変数名	説明	値の例
	KIE_SERVER_PWD	KIE サーバーのパスワード。 (org.kie.server.pwd システムプロパティを設定)	`\${KIE_SERVER_PWD}`
	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、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	`\${MAVEN_MIRROR_URL}`
	MAVEN_REPO_ID	Maven リポジトリに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*;!repo-rhpamcentr;!repo-custom などがあります。 MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	`\${MAVEN_REPO_ID}`
	MAVEN_REPO_URL	Maven リポジトリまたはサービスへの完全修飾 URL。	`\${MAVEN_REPO_URL}`
	MAVEN_REPO_USERNAME	Maven リポジトリにアクセスするユーザー名 (必要な場合)	`\${MAVEN_REPO_USERNAME}`
	MAVEN_REPO_PASSWORD	Maven リポジトリにアクセスするパスワード (必要な場合)。	`\${MAVEN_REPO_PASSWORD}`

デプロイメント	変数名	説明	値の例
	KIE_MAVEN_USER	EAP 内の Business Central がホストする Maven サービスにアクセスするためのユーザー名	`\${BUSINESS_CENTRAL_MAVEN_USERNAME}`
	KIE_MAVEN_PWD	EAP 内の Business Central がホストする Maven サービスにアクセスするためのパスワード	`\${BUSINESS_CENTRAL_MAVEN_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の名前。	`\${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}`
	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}`

デプロイメント	変数名	説明	値の例
	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者のユーザー名 (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}`
	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}`

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

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

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

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	リファーラル (referral) を使用しない場合はこのオプションを使用する必要はありません。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: member) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。	`\${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}`
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMapping のログインモジュールで、指定したファイルを使用するように設定します。このパラメーターは、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を設定)	`\${KIE_SERVER_PERSISTENCE_DS}`
	RHPAM_JTA	–	true
	RHPAM_DRIVER	–	h2
	RHPAM_USERNAME	KIE サーバー 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 管理者のユーザー名	`\${KIE_ADMIN_USER}`
	KIE_ADMIN_PWD	KIE 管理者のパスワード	`\${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 サーバーはタスク関連の操作は認証済みユーザーをスキップできる。 (org.kie.server.bypass.auth.user システムプロパティーを設定)	`\${KIE_SERVER_BYPASS_AUTH_USER}`
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_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	–	OpenShiftStartupStrategy
	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティーを設定)	`\${KIE_SERVER_USER}`

デプロイメント	変数名	説明	値の例
	KIE_SERVER_PWD	KIE サーバーのパスワード。 (org.kie.server.pwd システムプロパティを設定)	\${KIE_SERVER_PWD}
	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、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_OFF	KIE Server の Maven ミラー設定。	\${MAVEN_MIRROR_OFF}
	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	EAP 内の Business Central がホストする Maven サービスにアクセスするためのユーザー名	\${BUSINESS_CENTRAL_MAVEN_USERNAME}
	RHPAMCENTR_MAVEN_REPO_PASSWORD	EAP 内の Business Central がホストする Maven サービスにアクセスするためのパスワード	\${BUSINESS_CENTRAL_MAVEN_PASSWORD}

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

デプロイメント	変数名	説明	値の例
	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 レルムの管理者のユーザー名 (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、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定した場合には、DN はユーザー名に対して解析されます。false に設定した場合には、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_ROLE_S_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	`\${AUTH_LDAP_ROLE_S_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。roleAttributesDN プロパティを true に設定すると、このプロパティ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。リファールを使用し、ロールオブジェクトがリファール内部にあると、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、置き換えられません。	\${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}-rhpmcentr	businesscentral-keystore-volume	/etc/businesscentral-secret-volume	ssl certs	True
\${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}-rhpmcentr-claim	ReadWriteOnce

名前	アクセスモード
<code>\${APPLICATION_NAME}-kie-claim</code>	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. RHPAM74-AUTHORING-HA.YAML テンプレート

Red Hat Process Automation Manager 7.4 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

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、パラメーターの参照時には、この値が代入されます。この値は、パラメーターの参照時には、この値が代入されます。参照はオブジェクト一覧フィールドの任意のテキストフィールドで定義できます。詳細は、[Openshift ドキュメント](#) を参照してください。

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
KIE_ADMIN_USER	KIE_ADMIN_USER	KIE 管理者のユーザー名	adminUser	False
KIE_ADMIN_PASSWORD	KIE_ADMIN_PASSWORD	KIE 管理者のパスワード	–	False
KIE_SERVER_CONTROLLER_USER	KIE_SERVER_CONTROLLER_USER	KIE サーバーコントローラーのユーザー名。 (org.kie.server.controller.user システムプロパティーを設定)	controllerUser	False
KIE_SERVER_CONTROLLER_PASSWORD	KIE_SERVER_CONTROLLER_PASSWORD	KIE サーバーコントローラーのパスワード。 (org.kie.server.controller.pwd システムプロパティーを設定)	–	False

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	ベアラー認証用の KIE Server コントローラートークン。 (org.kie.server.controller.token システムプロパティを設定)	–	False
KIE_SERVER_USER	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティを設定)	executionUser	False
KIE_SERVER_PASSWORD	KIE_SERVER_PASSWORD	KIE サーバーのパスワード。 (org.kie.server.pwd システムプロパティを設定)	–	False
KIE_SERVER_BYPASS_AUTH_USER	KIE_SERVER_BYPASS_AUTH_USER	KIE サーバーはタスク関連の操作は認証済みユーザーをスキップできる。 (org.kie.server.bypass.auth.user システムプロパティを設定)	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名前空間。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。これは MySQL バージョンに対応するように意図されています。これは MySQL バージョンに対応するように意図されており、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 kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルターリング。 (org.drools.server.filter.classes システムプロパティを設定)	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:1.1	True
DATAGRID_MEMORY_LIMIT	–	DataGrid コンテナのメモリー制限	512 Mi	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_SCALEDOWN_CONTROLLER_IMAGE_STREAM_NAMESPACE	–	AMQ スケールダウンコントローラーイメージの ImageStream がインストールされている namespace。デフォルトは openshift です。	openshift	True
AMQ_SCALEDOWN_CONTROLLER_IMAGE_STREAM_NAME	–	AMQ スケールダウンコントローラーに使用するイメージストリーム名。デフォルトは、amq-broker-72-scaledown-controller-openshift です。	amq-broker-72-scaledown-controller-openshift	True
AMQ_SCALEDOWN_CONTROLLER_IMAGE_STREAM_TAG	–	AMQ スケールダウンコントローラーイメージタグ。デフォルトは 1.0 です。	1.0	True
AMQ_BROKER_IMAGE	–	AMQ ブローカーイメージ。	registry.redhat.io/amq-broker-7/amq-broker-73-openshift:7.3	True
AMQ_ROLE	–	標準ブローカーユーザーのユーザーロール。	admin	True

変数名	イメージの環境変数	説明	値の例	必須
AMQ_NAME	–	ブローカーの名前	broker	True
AMQ_GLOBAL_MAX_SIZE	–	メッセージデータが使用可能な最大メモリ量を指定します。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
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の KIE サービス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を設定)	60000	False

変数名	イメージの環境変数	説明	値の例	必須
IMAGE_STREAM_NAMESPACE	–	Red Hat ミドルウェアイメージの ImageStreams がインストールされている namespace これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStreams を別の namespace/プロジェクトにインストールしている場合には、これを変更するだけで結構です。	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 rhpam74-businesscentral-openshift です。	rhpam74-businesscentral-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 rhpam74-kieserver-openshift です。	rhpam74-kieserver-openshift	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 1.0 です。	1.0	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、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 リポジトリに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*;!repo-rhpamcentr;!repo-custom などがあります。MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	repo-custom	False

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 リポジトリにアクセスするユーザー名 (必要な場合)	–	False
MAVEN_REPO_PASSWORD	MAVEN_REPO_PASSWORD	Maven リポジトリにアクセスするパスワード (必要な場合)。	–	False
BUSINESS_CENTRAL_MAVEN_USERNAME	KIE_MAVEN_USER	EAP 内の Business Central がホストする Maven サービスにアクセスするためのユーザー名	mavenUser	True
BUSINESS_CENTRAL_MAVEN_PASSWORD	KIE_MAVEN_PASSWORD	EAP 内の Business Central がホストする Maven サービスにアクセスするためのパスワード	–	True
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 コンテナのメモリー制限。	2Gi	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナのメモリー制限。	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
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 レルムの管理者のユーザー名 (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

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

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

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

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

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。 roleAttributeIsDN プロパティを true に設定すると、このプロパティ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 roleNameAttribute Id 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_USE_R_ATTRIBUTE_ID_TO_CHECK	AUTH_LDAP_REFERRAL_USE_R_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

5.2.2. 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	

サービス	ポート	名前	説明
	8001	git-ssh	
\${APPLICATION_NAME}-rhpamcentr-ping	8888	ping	rhpamcentr クラスターリングの JGroups ping ポート
\${APPLICATION_NAME}-datagrid-ping	8888	ping	クラスターリング向けの 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 クラスターリングの JGroups ping ポート。
\${APPLICATION_NAME}-mysql	3306	–	MySQL サーバーのポート。

5.2.2.2. ルート

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手段です。ルーターは、定義したルートや、サービスで特定したエンドポイントを使用して、外部のクライアントからアプリケーションに名前付きの接続を提供します。各ルートは、ルート名、サービスセレクター、セキュリティー設定 (任意) で設定されます。詳細は、[Openshift ドキュメント](#) を参照してください。

サービス	セキュリティー	ホスト名
insecure- \${APPLICATION_NAME}-rhpamcentr-http	なし	\${BUSINESS_CENTRAL_HOSTNAME_HTTP}
\${APPLICATION_NAME}-rhpamcentr-https	TLS パススルー	\${BUSINESS_CENTRAL_HOSTNAME_HTTPS}

サービス	セキュリティー	ホスト名
insecure- \${APPLICATION_NAME}- kieserver-http	なし	\${KIE_SERVER_HOSTNAME}_HTTP}
\${APPLICATION_NAME}-kieserver-https	TLS パススルー	\${KIE_SERVER_HOSTNAME}_HTTPS}

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もとにするレプリケーションコントローラーです。デプロイメントは手動で作成されるか、トリガーされたイベントに対応するために作成されます。詳細は、[Openshift ドキュメント](#) を参照してください。

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。詳細は、[Openshift ドキュメント](#) を参照してください。

Deployment	トリガー
\${APPLICATION_NAME}-rhpamcentr	ImageChange
\${APPLICATION_NAME}-kieserver	ImageChange
\${APPLICATION_NAME}-amq-scaledown-controller	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、[コンテナエンジンのドキュメント](#) を参照してください。

Deployment	レプリカ
\${APPLICATION_NAME}-rhpamcentr	2
\${APPLICATION_NAME}-kieserver	2
\${APPLICATION_NAME}-amq-scaledown-controller	1
\${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。詳細は、[Openshift ドキュメント](#) を参照してください。

Deployment	サービスアカウント
<code>\${APPLICATION_NAME}-rhpamcentr</code>	<code>\${APPLICATION_NAME}-rhpamsvc</code>
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${APPLICATION_NAME}-rhpamsvc</code>
<code>\${APPLICATION_NAME}-amq-scaledown-controller</code>	<code>\${APPLICATION_NAME}-amq-scaledown-controller-sa</code>

5.2.2.3.3.2. Image

デプロイメント	イメージ
<code>\${APPLICATION_NAME}-rhpamcentr</code>	<code>\${BUSINESS_CENTRAL_IMAGE_STREAM_NAME}</code>
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${KIE_SERVER_IMAGE_STREAM_NAME}</code>
<code>\${APPLICATION_NAME}-amq-scaledown-controller</code>	<code>\${AMQ_SCALEDOWN_CONTROLLER_IMAGE_STREAM_NAME}</code>
<code>\${APPLICATION_NAME}-mysql</code>	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
<code>\${APPLICATION_NAME}-rhpamcentr</code>	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
	git-ssh	8001	TCP
	ping	8888	TCP
<code>\${APPLICATION_NAME}-kieserver</code>	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
<code>\${APPLICATION_NAME}-mysql</code>	–	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
<code>\${APPLICATION_NAME}-rhpamcentr</code>	<code>KIE_ADMIN_USER</code>	KIE 管理者のユーザー名	<code>\${KIE_ADMIN_USER}</code>
	<code>KIE_ADMIN_PWD</code>	KIE 管理者のパスワード	<code>\${KIE_ADMIN_PWD}</code>
	<code>KIE_MBEANS</code>	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 <code>kie.mbeans</code> および <code>kie.scanner.mbeans</code> を設定)	<code>\${KIE_MBEANS}</code>

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_OPENSIFT_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を設定)。	`\${KIE_SERVER_CONTROLLER_OPENSIFT_GLOBAL_DISCOVERY_ENABLED}`
	KIE_SERVER_CONTROLLER_OPENSIFT_PREFER_KIESERVER_SERVICE	OpenShift 内部サービスエンドポイント経由での KIE サービス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を設定します)。	`\${KIE_SERVER_CONTROLLER_OPENSIFT_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を設定)	`\${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}`
	KIE_WORKBENCH_CONTROLLER_OPENSIFT_ENABLED	–	true
	KIE_SERVER_CONTROLLER_USER	KIE サーバーコントローラーのユーザー名。 (org.kie.server.controller.user システムプロパティを設定)	`\${KIE_SERVER_CONTROLLER_USER}`
	KIE_SERVER_CONTROLLER_PWD	KIE サーバーコントローラーのパスワード。 (org.kie.server.controller.pwd システムプロパティを設定)	`\${KIE_SERVER_CONTROLLER_PWD}`
	KIE_SERVER_CONTROLLER_TOKEN	ベアラ認証用の KIE Server コントローラートークン。 (org.kie.server.controller.token システムプロパティを設定)	`\${KIE_SERVER_CONTROLLER_TOKEN}`

デプロイメント	変数名	説明	値の例
	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティーを設定)	`\${KIE_SERVER_USER}`
	KIE_SERVER_PWD	KIE サーバーのパスワード。 (org.kie.server.pwd システムプロパティーを設定)	`\${KIE_SERVER_PWD}`
	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、このミラーにはサービスのビルドおよびデプロイに必要なすべてのアーティファクトを含める必要があります。	`\${MAVEN_MIRROR_URL}`
	MAVEN_REPO_ID	Maven リポジトリに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。 MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	`\${MAVEN_REPO_ID}`
	MAVEN_REPO_URL	Maven リポジトリ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
	MAVEN_REPO_USE RNAME	Maven リポジトリに アクセスするユーザー名 (必要な場合)	\${MAVEN_REPO_US ERNAME}
	MAVEN_REPO_PAS SWORD	Maven リポジトリに アクセスするパスワード (必要な場合)。	\${MAVEN_REPO_PA SSWORD}
	KIE_MAVEN_USER	EAP 内の Business Central がホストする Maven サービスにアク セスするためのユーザー 名	\${BUSINESS_CENTR AL_MAVEN_USERN AME}
	KIE_MAVEN_PWD	EAP 内の Business Central がホストする Maven サービスにアク セスするためのパスワー ド	\${BUSINESS_CENTR AL_MAVEN_PASSW ORD}
	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_CENTR AL_HTTPS_KEYSTO RE}
	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_CENTR AL_HTTPS_NAME}
	HTTPS_PASSWORD	Business Central のキー ストアおよび証明書のパ スワード。	\${BUSINESS_CENTR AL_HTTPS_PASSW ORD}
	JGROUPS_PING_PR OTOCOL	–	openshift.DNS_PING
	OPENSIFT_DNS_PI NG_SERVICE_NAME	–	\${APPLICATION_NA ME}-rhpamcentr- ping

デプロイメント	変数名	説明	値の例
	OPENSIFT_DNS_PI NG_SERVICE_PORT	–	8888
	APPFORMER_INFINI SPAN_SERVICE_NA ME	–	`\${APPLICATION_NA ME}-datagrid`
	APPFORMER_INFINI SPAN_PORT	–	11222
	APPFORMER_JMS_ BROKER_ADDRESS	–	`\${APPLICATION_NA ME}-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_PASSWOR D	JMS ブローカーに接続 するためのパスワード。	`\${APPFORMER_JMS _BROKER_PASSWO RD}`
	SSO_URL	RH-SSO URL。	`\${SSO_URL}`
	SSO_OPENIDCONN ECT_DEPLOYMENT S	–	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_CENTR AL_SSO_SECRET}`
	SSO_CLIENT	Business Central RH- SSO クライアント名。	`\${BUSINESS_CENTR AL_SSO_CLIENT}`
	SSO_USERNAME	クライアント作成に使用 する RH-SSO レルムの 管理者のユーザー名 (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	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}`

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	`\${AUTH_LDAP_BASE_FILTER}`
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	`\${AUTH_LDAP_SEARCH_SCOPE}`
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	`\${AUTH_LDAP_SEARCH_TIME_LIMIT}`
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定した場合には、DN はユーザー名に対して解析されます。false に設定した場合には、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_USER NAME_BEGIN_STR ING	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_USE RNAME_BEGIN_STR ING}`
	AUTH_LDAP_USER NAME_END_STRIN G	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_USE RNAME_END_STRIN G}`
	AUTH_LDAP_ROLE_ ATTRIBUTE_ID	ユーザーロールを含む属性の名前。	`\${AUTH_LDAP_ROL E_ATTRIBUTE_ID}`
	AUTH_LDAP_ROLE S_CTX_DN	ユーザーロールを検索するコンテキストの固定 DN。これは、実際のロールが存在する DN ではなく、ユーザーロールを含むオブジェクトが存在する DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	`\${AUTH_LDAP_ROL ES_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。roleAttributesDN プロパティを true に設定すると、このプロパティ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。リファールを使用し、ロールオブジェクトがリファール内部にあると、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
	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 ティを設定)	\${KIE_SERVER_PERSISTENCE_DS}
	RHPAM_JTA	–	true
	KIE_ADMIN_USER	KIE 管理者のユーザー名	\${KIE_ADMIN_USER}
	KIE_ADMIN_PWD	KIE 管理者のパスワード	\${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 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を設定)	`\${PROMETHEUS_SERVER_EXT_DISABLED}`
	KIE_SERVER_BYPASS_AUTH_USER	KIE サーバーはタスク関連の操作は認証済みユーザーをスキップできる。 (org.kie.server.bypass.auth.user システムプロパティを設定)	`\${KIE_SERVER_BYPASS_AUTH_USER}`
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	KIE_SERVER_PWD	KIE サーバーのパスワード。 (org.kie.server.pwd システムプロパティを設定)	`\${KIE_SERVER_PWD}`
	KIE_SERVER_USER	KIE サーバーのユーザー名 (org.kie.server.user システムプロパティを設定)	`\${KIE_SERVER_USER}`
	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、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	EAP 内の Business Central がホストする Maven サービスにアクセスするためのユーザー名	`\${BUSINESS_CENTRAL_MAVEN_USERNAME}`
	RHPAMCENTR_MAVEN_REPO_PASSWORD	EAP 内の Business Central がホストする Maven サービスにアクセスするためのパスワード	`\${BUSINESS_CENTRAL_MAVEN_PASSWORD}`
	EXTERNAL_MAVEN_REPO_ID	Maven リポジトリに使用する ID。これが設定されている場合は、MAVEN_MIRROR_OF に追加して、必要に応じて設定したミラーから除外できます。たとえば、external:*,!repo-rhpamcentr,!repo-custom などがあります。 MAVEN_MIRROR_URL に設定されていても MAVEN_MIRROR_ID が設定されていない場合は、ID が無作為に生成され、MAVEN_MIRROR_OF では使用できません。	`\${MAVEN_REPO_ID}`

デプロイメント	変数名	説明	値の例
	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリまたはサービスへの完全修飾 URL。	`\${MAVEN_REPO_URL}`
	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリにアクセスするユーザー名 (必要な場合)	`\${MAVEN_REPO_USERNAME}`
	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリ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 レルムの管理者のユーザー名 (存在しない場合)	`\${SSO_USERNAME}`

デプロイメント	変数名	説明	値の例
	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。	\${SSO_PASSWORD}
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性	\${SSO_PRINCIPAL_ATTRIBUTE}
	HOSTNAME_HTTP	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}

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	<code>\${AUTH_LDAP_BASE_CTX_DN}</code>
	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。	<code>\${AUTH_LDAP_BASE_FILTER}</code>
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	<code>\${AUTH_LDAP_SEARCH_SCOPE}</code>
	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)。	<code>\${AUTH_LDAP_SEARCH_TIME_LIMIT}</code>
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自身の DN に正しいユーザーマッピングを妨げる特殊文字 (バックスラッシュなど) が含まれる場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	<code>\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}</code>

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

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

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

DNS_PING を機能させるには、以下の手順を実行する必要があります。

1. **OPENSIFT_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. **OPENSIFT_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) 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. **OPENSIFT_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. **OPENSIFT_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) が参加しようとしています。
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. OPENSIFT 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修正操作](#) 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11月15日（月）