



Red Hat Process Automation Manager 7.5

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

ガイド

Red Hat Process Automation Manager 7.5 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

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_fixed_managed_server_environment_on_file |.

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 7.5 固定管理サーバー環境をデプロイする方法を説明します。

目次

前書き	3
第1章 RED HAT OPENSIFT CONTAINER PLATFORM における RED HAT PROCESS AUTOMATION MANAGER の概要	4
第2章 OPENSIFT 環境に RED HAT PROCESS AUTOMATION MANAGER をデプロイする準備	7
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	7
2.2. PROCESS SERVER にシークレットの作成	8
2.3. BUSINESS CENTRAL へのシークレットの作成	9
2.4. SMART ROUTER のシークレットの作成	9
2.5. オフラインで使用する MAVEN ミラーリポジトリの用意	10
第3章 固定管理サーバー環境	12
3.1. 固定管理サーバー環境のデプロイ	12
3.1.1. 固定管理サーバー環境用のテンプレートの設定開始	12
3.1.2. 固定管理サーバー環境に必要なパラメーターの設定	13
3.1.3. 固定管理サーバー環境用のイメージストリームの名前空間の設定	15
3.1.4. 固定管理対象サーバー環境の Pod レプリカ数の設定	15
3.1.5. 固定管理サーバー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手順	16
3.1.6. 固定管理サーバー環境用の RH-SSO 認証パラメーターの設定	16
3.1.7. 固定管理サーバー環境用の LDAP 認証パラメーターの設定	18
3.1.8. 固定管理サーバー環境用に外部データベースサーバーを使用するパラメーターの設定	19
3.1.9. 固定管理サーバー環境用の Prometheus メトリクス収集の有効化	21
3.1.10. 固定管理サーバー環境のテンプレートのデプロイ	21
3.2. (任意) LDAP ロールマッピングファイルの指定	22
3.3. 管理環境のテンプレートの修正	23
3.4. 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	25
第4章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー	28
第5章 OPENSIFT テンプレートの参考資料	30
5.1. RHPAM75-PROD.YAML TEMPLATE	30
5.1.1. パラメーター	30
5.1.2. オブジェクト	49
5.1.2.1. サービス	49
5.1.2.2. ルート	50
5.1.2.3. デプロイメント設定	51
5.1.2.3.1. トリガー	51
5.1.2.3.2. レプリカ	51
5.1.2.3.3. Pod テンプレート	52
5.1.2.4. 外部の依存関係	88
5.1.2.4.1. ボリューム要求	88
5.1.2.4.2. シークレット	88
5.2. OPENSIFT の使用に関するクイックリファレンス	88
付録A バージョン情報	91

前書き

システムエンジニアは、Red Hat OpenShift Container Platform に Red Hat Process Automation Manager 固定管理サーバー環境をデプロイして、プロセスおよびその他のビジネスアセットを実行するインフラストラクチャーを提供できます。この環境には、単一のデプロイメントには、一定数の Process Server が含まれており、後でサーバーを追加したり削除したりできません。Business Central Monitoring を使用して、この環境の Process Server で実行するプロセスの管理および更新が可能になります。

前提条件

- Red Hat OpenShift Container Platform バージョン 3.11 がデプロイされている。
- OpenShift クラスター/namespace で 4 ギガバイト以上のメモリーが利用可能である。
- 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する必要があります。
- 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 の各セットに、デフォルトでデータベースに1つの 1Gi PV が設定されている。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 Monitoring には、1つの 64Mi PV が必要です。
 - Smart Router には、1つの 64Mi PV が必要です。
- 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している。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。



注記

Red Hat Process Automation Manager バージョン 7.5 以降、すべてのテンプレートを使用したインストールや Automation Broker(Ansible Playbook)を使用したインストールを含め、Red Hat OpenShift Container Platform 3.x のサポートが非推奨になりました。新機能が追加されない可能性があり、この機能は今後のリリースで削除予定です。

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 台で構成されます。この環境のデプロイメント手順については、[『Red 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 -F rhpam-businesscentral | grep -F 7.5
$ oc get imagestreamtag -n openshift | grep -F rhpam-kieserver | grep -F 7.5
```

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-7.5.1-openshift-templates.zip** の製品配信可能ファイルをダウンロードし、**rhpm75-image-streams.yaml** ファイルを展開します。
- g. 以下のコマンドを実行します。

```
$ oc apply -f rhpm75-image-streams.yaml
```



注記

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

2.2. 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
```

2.3. BUSINESS CENTRAL へのシークレットの作成

Business Central への HTTP アクセス用に SSL 証明書を作成し、これをシークレットとして OpenShift 環境に指定する必要があります。

Business Central と 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
```

2.4. SMART ROUTER のシークレットの作成

Smart Router 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。

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

手順

1. Smart Router 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、Smart Router の予想される URL と一致する有効な署名済み証明書を生成します。

2. 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 **smartrouter-app-secret** を生成します。

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

2.5. オフラインで使用する MAVEN ミラーリポジトリーを用意

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



注記

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

前提条件

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

手順

1. 書き込み可能な Maven リリースリポジトリーを準備します。このリポジトリーは、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、このリポジトリーへのアクセスが必要です。OpenShift 環境に、Nexus リポジトリーマネージャーをデプロイできます。OpenShift への Nexus の設定方法は、「[Nexus の設定](#)」を参照してください。このリポジトリーを別個のミラーリポジトリーとして使用します。
または、サービスにカスタムの外部リポジトリー (Nexus など) を使用する場合、同じリポジトリーをミラーリポジトリーとして使用できます。
2. 公開インターネットに送信アクセスができるコンピューターで、以下のアクションを実行します。
 - a. 最新バージョンの [Offliner ツール](#) をダウンロードします。
 - b. 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-7.5.1-offliner.txt** の製品配信可能ファイルをダウンロードします。
 - c. 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-7.5.1-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。

- d. 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。

第3章 固定管理サーバー環境

単一のデプロイメントで固定管理サーバー環境をデプロイし、Process Server を実行する異なる Pod を複数含めることができます。プロセスは、デフォルトではサーバーに読み込まれていません。データベースサーバーは、デフォルトで Pod で実行します。Process Server の各 Pod は、必要に応じて個別にスケールできます。

Business Central Monitoring を使用する Pod と、Smart Router を使用する Pod もデプロイできます。Process Server にプロセスをデプロイ、ロード、アンロードするには、Business Central Monitoring が必要になります。また、監視情報を表示することもできます。

Smart Router は、クライアントアプリケーションからプロセスへのコールを受け取れる単一エンドポイントで、プロセスを実行するサーバーへの各コールのルートを自動的に決定します。

後で簡単に Process Server を追加したり、削除したりできません。

サーバーにデプロイするプロセス (KJAR ファイル) がある Maven リポジトリを提供する必要があります。統合プロセスは、必要なバージョンのプロセスが Maven リポジトリにアップロードされるようにする必要があります。開発環境で Business Central を使用してプロセスを作成し、Maven リポジトリにアップロードできます。

3.1. 固定管理サーバー環境のデプロイ

テンプレート1つで固定管理サーバー環境をデプロイできます。テンプレートファイル名は **rhpm75-prod.yaml** です。

このテンプレートには、2つの Process Server Pod (および PostgreSQL データベース Pod)、高可用性設定の Smart Router、高可用性設定の Business Central Monitoring が含まれます。

デプロイメントの設定時に、全コンポーネントのレプリカ数を変更できます。独立した Process Server Pod の数を変更するか、別のデータベースサーバーを使用する場合には、テンプレートを変更する必要があります。テンプレートの変更に関する説明は、「[管理環境のテンプレートの修正](#)」を参照してください。



注記

固定管理環境テンプレートは、Red Hat Process Automation Manager 7.5 で非推奨になりました。これは今後のリリースで削除されます。

3.1.1. 固定管理サーバー環境用のテンプレートの設定開始

固定管理サーバー環境をデプロイするには、**rhpm75-prod.yaml** テンプレートファイルを使用します。

手順

1. Red Hat カスタマーポータル [の Software Downloads](#) ページから製品配信可能ファイル **rhpm75.1-openshift-templates.zip** をダウンロードします。
2. **rhpm75-prod.yaml** テンプレートファイルを展開します。
3. デフォルトで、テンプレートには2つの Process Server が含まれます。それぞれのサーバーは Pod で PostgreSQL データベースサーバーを使用します。Process Server の数を変更するか、Pod で MySQL データベースサーバーを使用するか、または外部データベースサーバーを使用するには、「[管理環境のテンプレートの修正](#)」の説明に従ってテンプレートを変更します。

4. 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、**rhcam75-prod.yaml** ファイルを選択するか、またはこれを貼り付けます。Add Template ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
- OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/rhcam75-prod.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。
- 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。「[固定管理サーバー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. 固定管理サーバー環境に必要なパラメーターの設定

テンプレートを使用して固定管理サーバー環境をデプロイするように設定するには、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

- 「[固定管理サーバー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 Monitoring 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。
- **Smart Router Keystore Secret Name (KIE_SERVER_ROUTER_HTTPS_SECRET):** 「[Smart Router のシークレットの作成](#)」で作成した Smart Router のシークレットの名前。
- **Business Central Monitoring 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 Monitoring 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。
 - **Smart Router Certificate Name (KIE_SERVER_ROUTER_HTTPS_NAME):** 「Smart Router のシークレットの作成」 で作成したキーストアの証明書名。
 - **Smart Router Keystore Password (KIE_SERVER_ROUTER_HTTPS_PASSWORD):** 「Smart Router のシークレットの作成」 で作成したキーストアのパスワード。
 - **アプリケーション名 (APPLICATION_NAME):** OpenShift アプリケーションの名前。これは、Business Central Monitoring および 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別個のセットを作成します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名を同じにすることはできません。また、アプリケーション名は、Process Server が Business Central Monitoring で参加するサーバーの設定 (サーバーテンプレート) の名前を決定するものとなります。複数の Process Server をデプロイしている場合、それぞれのサーバーに異なるアプリケーション名があることを確認する必要があります。
 - **Maven repository URL (MAVEN_REPO_URL):** Maven リポジトリーの URL。Process Server にデプロイするすべてのプロセス (KJAR ファイル) をこのリポジトリーにアップロードする必要があります。
 - **Maven リポジトリーの ID (MAVEN_REPO_ID):** Maven リポジトリーの ID。デフォルト値は **repo-custom** です。
 - **Maven repository username (MAVEN_REPO_USERNAME):** Maven リポジトリーのユーザー名。
 - **Maven リポジトリーのパスワード (MAVEN_REPO_PASSWORD):** Maven リポジトリーのパスワード。
 - **KIE Server モード (KIE_SERVER_MODE):** `rhcam75-kieserver-*.yaml` テンプレートで、デフォルト値は **PRODUCTION** です。**PRODUCTION** モードでは、**SNAPSHOT** バージョンの KJAR アーティファクトは Process 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できません。**PRODUCTION** モードで新規バージョンをデプロイするには、同じ Process Server で新規コンテナを作成します。**SNAPSHOT** バージョンをデプロイするか、または既存コンテナのアーティファクトのバージョンを変更するには、このパラメーターを **DEVELOPMENT** に設定します。
 - **ImageStream 名前空間 (IMAGE_STREAM_NAMESPACE):** イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが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。
- 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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」の手順に従います。

3.1.3. 固定管理サーバー環境用のイメージストリームの名前空間の設定

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

- 「[固定管理サーバー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

「[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。

3.1.4. 固定管理対象サーバー環境の Pod レプリカ数の設定

固定管理サーバー環境をデプロイするようにテンプレートを設定する時に、Process Server、Business Central、Smart Router のレプリカの初期数を設定できます。

前提条件

- 「[固定管理サーバー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

レプリカの数を設定するには、次のパラメーターを設定します。

- **Business Central Monitoring Container Replicas** (**BUSINESS_CENTRAL_MONITORING_CONTAINER_REPLICAS**): デプロイメント時に最初に Business Central Monitoring 用に最初に作成されるレプリカ数。Business Central Monitoring に高可用性設定を使用しない場合は、この数値を 1 に設定します。
- **KIE Server Container Replicas**(**KIE_SERVER_CONTAINER_REPLICAS**): デプロイメントで Process Server 用に最初に作成されるレプリカの数。
- **Smart Router Container Replicas** (**SMART_ROUTER_CONTAINER_REPLICAS**): デプロイメント時に最初に Smart Router 用に作成されるレプリカ数。

次のステップ

必要な場合は、追加のパラメーターを設定します。

デプロイを完了するには、「[固定管理サーバー環境のテンプレートのデプロイ](#)」の手順に従います。

3.1.5. 固定管理サーバー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手順

固定管理サーバー環境をデプロイするようにテンプレートを設定する時に、OpenShift 環境に公開インターネットへの接続がない場合は、「[オフラインで使用する Maven ミラーリポジトリーの用意](#)」に従って設定した Maven ミラーへのアクセスを設定する必要があります。

前提条件

- 「[固定管理サーバー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:*** です。この値の場合、Maven はミラーから必要なアーティファクトをすべて取得し、他のリポジトリーにクエリーを送信しません。
 - 外部の Maven リポジトリー (**MAVEN_REPO_URL**) を設定する場合は、ミラーからこのリポジトリー内のアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-custom**)。 **repo-custom** は、**MAVEN_REPO_ID** で設定した ID に置き換えます。
 - ビルトイン Business Central Maven リポジトリー (**BUSINESS_CENTRAL_MAVEN_SERVICE**) を設定する場合には、ミラーからこのリポジトリーのアーティファクトを除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhpamcentr**)。
 - 両方のリポジトリーを設定している場合は、ミラーからこのリポジトリーを両方除外するように **MAVEN_MIRROR_OF** を変更します (例: **external:*,!repo-rhpamcentr,!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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」の手順に従います。

3.1.6. 固定管理サーバー環境用の 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は、固定管理サーバー環境をデプロイするようにテンプレートを設定する時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 Monitoring 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 Monitoring 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。
- 「[固定管理サーバー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 Monitoring RH-SSO Client name (BUSINESS_CENTRAL_SSO_CLIENT):** Business Central Monitoring の RH-SSO クライアント名。
- テンプレートで定義した各 Process Server の場合:
 - **KIE Server_n RH-SSO Client name (KIE_SERVER_n_SSO_CLIENT):** この Process Server の RH-SSO クライアント名
 - **KIE Server_n RH-SSO Client Secret (KIE_SERVER_n_SSO_SECRET):** この Process Server のクライアントに対して RH-SSO で設定した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。
 - テンプレートで定義した各 Process Server の場合:
 - **KIE Server_n RH-SSO Client name (KIE_SERVER_n_SSO_CLIENT):** この Process Server に対して RH-SSO で作成したクライアント名
 - **KIE Server_n RH-SSO Client Secret (KIE_SERVER_n_SSO_SECRET):** この Process Server のクライアントに対して RH-SSO で設定した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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」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。

3.1.7. 固定管理サーバー環境用の 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、固定管理サーバー環境をデプロイするようにテンプレートを設定する時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。
- 「[固定管理サーバー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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」の手順に従います。

3.1.8. 固定管理サーバー環境用に外部データベースサーバーを使用するパラメーターの設定

「[管理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、固定管理サーバー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

- 「[固定管理サーバー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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」の手順に従います。

3.1.9. 固定管理サーバー環境用の 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

- 「[固定管理サーバー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、「[固定管理サーバー環境のテンプレートのデプロイ](#)」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。

3.1.10. 固定管理サーバー環境のテンプレートのデプロイ

OpenShift Web UI またはコマンドラインに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-rhpamcentrmon**: Business Central Monitoring
- **myapp-kieserver-n**: Process Server の数 **n**。デフォルトでは、この数字は、1と2です。

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. 管理環境のテンプレートの修正

ニーズに応じて管理環境を調整するには、**rh pam75-prod.yaml** テンプレートを変更してから、環境をデプロイする必要があります。

デフォルトでは、テンプレートはレプリケートした 2 つの Process Server Pod を作成します。各 Pod に異なるプロセスをデプロイできます。レプリケートした Process Server Pod をさらに追加するには、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。

デフォルトでは、テンプレートは、PostgreSQL Pod を作成して、レプリケートした各 Process Server にデータベースサーバーを提供します。代わりに PostgreSQL または (OpenShift プロジェクト外の) 外部サーバーを使用する場合は、環境をデプロイする前にテンプレートを修正する必要があります。

rh pam75-prod.yaml テンプレートでは、Business Central Monitoring におけるレプリカの初期数も調整できます。

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。

名前付きブロックはネストできます。

手順

- レプリケートした Process Server Pod をさらに追加する場合は、追加するすべての Pod で以下の手順を繰り返します。
 1. 新しい Pod の番号を選択します。デフォルトの Pod の番号は **1** および **2** ですが、新しい最初の Pod に **3**、次の Pod に **4** というように指定することもできます。
 2. ファイルの、以下のブロックの **BEGIN** コメントから **END** コメントまでを、末尾にコピーします。
 - **KIE server services 1**
 - **PostgreSQL service 1**
 - **KIE server routes 1**
 - **KIE server deployment config 1**
 - **PostgreSQL deployment config 1**
 - **PostgreSQL persistent volume claim 1**

3. 作成したコピーで、**-1** のすべてのインスタンスを新しい Pod 番号 (例: **-3**) に置き換えます。
- PostgreSQL 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、**rhpm75-kieserver-postgresql.yaml** ファイルのブロックに置き換え、新たに追加したブロックの一部を修正します。
 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 **rhpm75-kieserver-postgresql.yaml** ファイルから取得します)
レプリケートしたすべての Process Server の Pod 番号で以下の手順を繰り返します。**N** は Pod 番号 (例: **1**) です。
 - **PosgreSQL service N** ブロックを **MySQL service** ブロックに置き換えます。
 - **PosgreSQL driver settings N** ブロックを **MySQL driver settings** ブロックに置き換えます。
 - **PosgreSQL deployment config N** ブロックを **MySQL deployment config** ブロックに置き換えます。
 - **PosgreSQL persistent volume claim N** ブロックを **MySQL persistent volume claim** ブロックに置き換えます。
 - 新たに追加したブロックで、以下の置換を手動で行います。**N** は Pod の番号になります。
 - **-mysql** を **-mysql-N** に置き換えます (ただし、**-mysql-pvol** と **-mysql-claim** の **-mysql** は 除く)。
 - **-mysql-claim** を **-mysql-claim-N** 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、**rhpm75-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、新たに追加したブロックの一部を修正します。
 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 **rhpm75-kieserver-externaldb.yaml** ファイルから取得します)
レプリケートしたすべての Process Server の Pod 番号で以下の手順を繰り返します。**N** は Pod 番号 (例: **1**) です。
 - **PosgreSQL service N** ブロックを削除します。
 - **PosgreSQL deployment config N** ブロックを削除します。
 - **PosgreSQL persistent volume claim N** ブロックを削除します。
 - **PosgreSQL driver settings N** ブロックを **External database driver settings** ブロックに置き替えます。
 - 新しい **External database driver settings** ブロックで、以下のいずれかの値が、別の Process Server Pod と異なる場合は、対象の Pod の値を設定します。
 - **RHPAM_USERNAME**: データベースサーバーにログインするユーザー名
 - **RHPAM_PASSWORD**: データベースサーバーにログインするためのパスワード

- **RHPAM_XA_CONNECTION_PROPERTY_URL**: データベースサーバーにログインするための完全 URL
- **RHPAM_SERVICE_HOST**: データベースサーバーのホスト名
- **RHPAM_DATABASE**: データベースの名前



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

3.4. 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 でサポートされる構成](#)」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、**EXTENSION_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](#) ページから製品配信可能ファイル **rhpan-7.5.1-openshift-templates.zip** 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せん。
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 CodeReady Studio を使用してプロセスを表示できます。**developer** ロールが割り当てられているユーザーには、新規リポジトリ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とって必須となります。

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、**rhpam-7.5.1-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。

- **rhpam75-prod.yaml** では、高可用性の Business Central Monitoring インスタンス、Smart Router 1つ、Business Central と Smart Router に接続した異なる Process Server 2 台、PostgreSQL インスタンス 2 つが提供されます。各 Process Server は独自の PostgreSQL インスタンスを使用します。この環境を使用して、実稼働またはステージング環境でビジネスアセットを実行できます。コンポーネントごとに、レプリカ数を設定できます。このテンプレートの詳細は、「[rhpam75-prod.yaml template](#)」を参照してください。

5.1. RHPAM75-PROD.YAML TEMPLATE

Red Hat Process Automation Manager 7.5 向けの、管理 HA 実稼働ランタイム環境のアプリケーションテンプレート（非推奨）

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
MAVEN_MIRROR_URL	MAVEN_MIRROR_URL	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	–	False
MAVEN_MIRROR_OF	MAVEN_MIRROR_OF	KIE Server の Maven ミラー設定。	external:*	False

変数名	イメージの環境変数	説明	値の例	必須
MAVEN_REPO_ID	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/	True
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

変数名	イメージの環境変数	説明	値の例	必須
BUSINESS_CENTRAL_MAVEN_SERVICE	RHPAMCENTRAL_MAVEN_REPO_SERVICE	maven リポジトリーの使用状況など、サービスのルックアップを可能にする Business Central に到達可能な場合に使用するサービス名 (必要な場合)	myapp-rhpamcentr	False
BUSINESS_CENTRAL_MAVEN_USERNAME	RHPAMCENTRAL_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	mavenUser	False
BUSINESS_CENTRAL_MAVEN_PASSWORD	RHPAMCENTRAL_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	maven!!	False
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_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

変数名	イメージの環境変数	説明	値の例	必須
IMAGE_STREAM_NAMESPACE	–	Red Hat Process Automation Manager イメージの ImageStream がインストールされている名前空間。これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。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「rhpm-kieserver-rhel8」です。	rhpm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは「7.5.0」です。	7.5.0	True
SMART_ROUTER_HOSTNAME_HTTP	–	http サービスルーートのカスタムホスト名。デフォルトのホスト名を使用する場合には空白にします (例: <application-name>-smartrouter-<project>.<default-domain-suffix>')。	–	False

変数名	イメージの環境変数	説明	値の例	必須
SMART_ROUTER_HOSTNAME_HTTPS	–	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure- <application-name>- smartrouter- <project>.<default-domain-suffix>')。	–	False
KIE_SERVER_ROUTER_ID	KIE_SERVER_ROUTER_ID	コントローラーに接続する場合に使用されるルーター ID (ルータープロパティ org.kie.server.router.id)。	kie-server-router	True
KIE_SERVER_ROUTER_PROTOCOL	KIE_SERVER_ROUTER_PROTOCOL	KIE Server ルーターのプロトコル。 (org.kie.server.router.url.external プロパティのビルドに使用)	http	False
KIE_SERVER_ROUTER_URL_EXTERNAL	KIE_SERVER_ROUTER_URL_EXTERNAL	ルーターを見つけることのできるパブリック URL。形式: <a href="http://<host>:<port>">http://<host>:<port> ; (ルータープロパティ org.kie.server.router.url.external)	–	False
KIE_SERVER_ROUTER_NAME	KIE_SERVER_ROUTER_NAME	コントローラーに接続するときに使用されるルーター名 (ルータープロパティ org.kie.server.router.name)。	KIE Server ルーター	True

変数名	イメージの環境変数	説明	値の例	必須
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
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_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

変数名	イメージの環境変数	説明	値の例	必須
POSTGRESQL_IMAGE_STREAM_NAMESPACE	–	PostgreSQL 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
POSTGRESQL_IMAGE_STREAM_TAG	–	PostgreSQL イメージのバージョン。これは PostgreSQL バージョンに対応するように意図されています。デフォルトは「10」です。	10	False
KIE_SERVER_POSTGRESQL_USER	RHPAM_USERNAME	KIE サーバー PostgreSQL データベースのユーザー名。	rhpm	False
KIE_SERVER_POSTGRESQL_PASSWORD	RHPAM_PASSWORD	KIE Server PostgreSQL データベースのパスワード。	–	False
KIE_SERVER_POSTGRESQL_DATABASE	RHPAM_DATABASE	KIE Server PostgreSQL データベース名。	rhpm7	False
POSTGRESQL_MAX_PREPARED_TRANSACTIONS	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	PostgreSQL による XA トランザクションの処理を許可します。	100	True

変数名	イメージの環境変数	説明	値の例	必須
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
KIE_SERVER_POSTGRESQLELECT	KIE_SERVER_PERSISTENCE_DIALECT	KIE Server PostgreSQL Hibernate 方言。	org.hibernate.dialect.PostgreSQLDialect	True
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)	PRODUCTION	False
KIE_MBEANS	KIE_MBEANS	KIE Server の mbeans が有効/無効になっています。(システムプロパティ kie.mbeans および kie.scanner.mbeans を設定)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	DROOLS_SERVER_FILTER_CLASSES	KIE Server クラスのフィルタリング。 (org.drools.server.filter.classes システムプロパティを設定)	true	False

変数名	イメージの環境変数	説明	値の例	必須
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
BUSINESS_CENTRAL_HOSTNAME_HTTP	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合は空白にします (例: <application-name>-rhpamcentrmon-<project>.<default-domain-suffix>)。	—	False
BUSINESS_CENTRAL_HOSTNAME_HTTPS	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure-<application-name>-rhpamcentrmon-<project>.<default-domain-suffix>)。	—	False
KIE_SERVER_HOSTNAME_HTTP	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>)。	—	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER1_HOSTNAME_HTTPS	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure- <application-name>-kieserver- <project>.<default-domain-suffix>)。	–	False
KIE_SERVER1_USE_SECURE_ROUTE_NAME	KIE_SERVER_USE_SECURE_ROUTE_NAME	true の場合は KIE Server は、レポートする Business Central の KIE Server ルートエンドポイントとして secure- <application-name>-kieserver (vs. <application-name>-kieserver) を使用します。そのため、Business Central はユーザーにセキュアなリンクを表示します。	false	False
KIE_SERVER2_HOSTNAME_HTTP	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver- <project>.<default-domain-suffix>)。	–	False
KIE_SERVER2_HOSTNAME_HTTPS	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure- <application-name>-kieserver- <project>.<default-domain-suffix>)。	–	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER2_US E_SECURE_ROUTE_NAME	KIE_SERVER_U SE_SECURE_R OUTE_NAME	True の場合、 secure- APPLICATION_NA ME-kieserver-2 vs を使用します。 APPLICATION_NA ME-kieserver-2 を ルート名として使 用します。	false	False
BUSINESS_CEN TRAL_HTTPS_S ECRET	–	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_CEN TRAL_HTTPS_ KEYSTORE	HTTPS_KEYST ORE	シークレット内の キーストアファイ ル名	keystore.jks	False
BUSINESS_CEN TRAL_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_CEN TRAL_HTTPS_P ASSWORD	HTTPS_PASSW ORD	キーストアおよび 証明書のパスワード	mykeystorepass	False
KIE_SERVER_R OUTER_HTTPS _SECRET	–	Smart Router の キーストアファイ ルが含まれるシー クレットの名前	smartrouter-app- secret	True
KIE_SERVER_R OUTER_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_R OUTER_HTTPS _NAME	KIE_SERVER_R OUTER_TLS_K EYSTORE_KEY ALIAS	サーバー証明書に 関連付けられてい る名前。	jboss	False
KIE_SERVER_R OUTER_HTTPS _PASSWORD	KIE_SERVER_R OUTER_TLS_K EYSTORE_PAS SWORD	キーストアおよび 証明書のパスワード。	mykeystorepass	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_HTTPS_SECRET	–	KIE Server のキーストアファイルが含まれるシークレットの名前。	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	HTTPS_KEYSTORE	シークレット内のキーストアファイル名	keystore.jks	False
KIE_SERVER_HTTPS_NAME	HTTPS_NAME	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	KIE_SERVER_BYPASS_AUTH_USER	KIE Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。(org.kie.server.bypass.auth.user システムプロパティを設定)	false	False
TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	EJB タイマーサービスのデータベースデータストアの更新間隔を設定します。	30000	False
BUSINESS_CENTRAL_MEMORY_LIMIT	–	Business Central Monitoring コンテナのメモリー制限	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE Server のコンテナのメモリー制限。	1Gi	False
SMART_ROUTER_MEMORY_LIMIT	–	Smart Router コンテナのメモリー制限	512Mi	False

変数名	イメージの環境変数	説明	値の例	必須
BUSINESS_CENTRAL_MONITORING_CONTAINER_REPLICAS	–	Business Central Monitoring Container Replicas は、起動する Business Central Monitoring コンテナ数を定義します。	3	True
SMART_ROUTER_CONTAINER_REPLICAS	–	Smart Router Container Replicas は、起動する Smart Router コンテナ数を定義します。	2	True
KIE_SERVER_CONTAINER_REPLICAS	–	KIE Server Container Replicas は、起動する KIE Server のコンテナ数を定義します。	3	True
SSO_URL	SSO_URL	RH-SSO URL。	https://rh-sso.example.com/auth	False
SSO_REALM	SSO_REALM	RH-SSO レalm 名。	–	False
BUSINESS_CENTRAL_SSO_CLIENT	SSO_CLIENT	Business Central Monitoring RH-SSO クライアント名。	–	False
BUSINESS_CENTRAL_SSO_SECRET	SSO_SECRET	Business Central Monitoring RH-SSO クライアントシークレット。	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER1_SSO_CLIENT	SSO_CLIENT	KIE Server 1 の RH-SSO クライアント名。	–	False
KIE_SERVER1_SSO_SECRET	SSO_SECRET	KIE Server 1 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
KIE_SERVER2_SSO_CLIENT	SSO_CLIENT	KIE Server 2 の RH-SSO クライアント名。	–	False
KIE_SERVER2_SSO_SECRET	SSO_SECRET	KIE Server 2 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
AUTH_LDAP_URL	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。	ldap://myldap.example.com	False
AUTH_LDAP_BIND_DN	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	パスワード	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	–	False

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

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

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

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_RECURSION	AUTH_LDAP_ROLE_RECURSION	ロール検索が一致するコンテキストで行われる再帰のレベル数。再帰を無効にするには、これを 0 に設定します。	1	False
AUTH_LDAP_DEFAULT_ROLE	AUTH_LDAP_DEFAULT_ROLE	認証された全ユーザーに対して含まれるロール。	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	ロール名を含む roleCtxDN コンテキスト内の属性の名前。 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。	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_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

5.1.2. 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}-rhpamcentrmon	8080	http	すべての Business Central Monitoring Web サーバーのポート。
	8443	https	

サービス	ポート	名前	説明
\${APPLICATION_NAME}-rhpmcentrmon-ping	8888	ping	クラスタリング向けの JGroups ping ポート。
\${APPLICATION_NAME}-smartrouter	9000	http	smart router サーバーの http ポートおよび https ポート
	9443	https	
\${APPLICATION_NAME}-kieserver-1	8080	http	すべての KIE Server Web サーバーのポート。(1つ目の KIE server)
	8443	https	
\${APPLICATION_NAME}-kieserver-1-ping	8888	ping	クラスタリング向けの JGroups ping ポート。
\${APPLICATION_NAME}-kieserver-2	8080	http	すべての KIE Server Web サーバーのポート。(2つ目の KIE Server)
	8443	https	
\${APPLICATION_NAME}-kieserver-2-ping	8888	ping	クラスタリング向けの JGroups ping ポート。
\${APPLICATION_NAME}-postgresql-1	5432	–	最初のデータベースサーバーのポート。
\${APPLICATION_NAME}-postgresql-2	5432	–	2番目のデータベースサーバーのポート。

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) で構成されます。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
\${APPLICATION_NAME}-rhpmcentrmon-http	なし	\${BUSINESS_CENTRAL_HOSTNAME_HTTP}
\${APPLICATION_NAME}-rhpmcentrmon-https	TLS パススルー	\${BUSINESS_CENTRAL_HOSTNAME_HTTPS}
\${APPLICATION_NAME}-kieserver-1-http	なし	\${KIE_SERVER1_HOSTNAME_HTTP}

サービス	セキュリティー	ホスト名
<code>\${APPLICATION_NAME}-kieserver-1-https</code>	TLS パススルー	<code>\${KIE_SERVER1_HOSTNAME_HTTPS}</code>
<code>\${APPLICATION_NAME}-kieserver-2-http</code>	なし	<code>\${KIE_SERVER2_HOSTNAME_HTTP}</code>
<code>\${APPLICATION_NAME}-kieserver-2-https</code>	TLS パススルー	<code>\${KIE_SERVER2_HOSTNAME_HTTPS}</code>
<code>\${APPLICATION_NAME}-smartrouter-http</code>	なし	<code>\${SMART_ROUTER_HOSTNAME_HTTP}</code>
<code>\${APPLICATION_NAME}-smartrouter-https</code>	TLS パススルー	<code>\${SMART_ROUTER_HOSTNAME_HTTPS}</code>

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
<code>\${APPLICATION_NAME}-rhpamcentrmon</code>	ImageChange
<code>\${APPLICATION_NAME}-smartrouter</code>	ImageChange
<code>\${APPLICATION_NAME}-kieserver-1</code>	ImageChange
<code>\${APPLICATION_NAME}-postgresql-1</code>	ImageChange
<code>\${APPLICATION_NAME}-kieserver-2</code>	ImageChange
<code>\${APPLICATION_NAME}-postgresql-2</code>	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}-rhpamcentrmon</code>	3
<code>\${APPLICATION_NAME}-smartrouter</code>	2
<code>\${APPLICATION_NAME}-kieserver-1</code>	3
<code>\${APPLICATION_NAME}-postgresql-1</code>	1
<code>\${APPLICATION_NAME}-kieserver-2</code>	3
<code>\${APPLICATION_NAME}-postgresql-2</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}-smartrouter</code>	<code>\${APPLICATION_NAME}-smartrouter</code>
<code>\${APPLICATION_NAME}-kieserver-1</code>	<code>\${APPLICATION_NAME}-kieserver</code>
<code>\${APPLICATION_NAME}-kieserver-2</code>	<code>\${APPLICATION_NAME}-kieserver</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}-rhpamcentrmon</code>	<code>rhpam-businesscentral-monitoring-rhel8</code>
<code>\${APPLICATION_NAME}-smartrouter</code>	<code>rhpam-smartrouter-rhel8</code>
<code>\${APPLICATION_NAME}-kieserver-1</code>	<code>\${KIE_SERVER_IMAGE_STREAM_NAME}</code>
<code>\${APPLICATION_NAME}-postgresql-1</code>	<code>postgresql</code>
<code>\${APPLICATION_NAME}-kieserver-2</code>	<code>\${KIE_SERVER_IMAGE_STREAM_NAME}</code>
<code>\${APPLICATION_NAME}-postgresql-2</code>	<code>postgresql</code>

5.1.2.3.3.3. Readiness Probe

`${APPLICATION_NAME}-rhpamcentrmon`

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

`${APPLICATION_NAME}-kieserver-1`

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

`${APPLICATION_NAME}-postgresql-1`

`/usr/libexec/check-container`

`${APPLICATION_NAME}-kieserver-2`

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

`${APPLICATION_NAME}-postgresql-2`

`/usr/libexec/check-container`

5.1.2.3.3.4. Liveness Probe

`${APPLICATION_NAME}-rhpamcentrmon`

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

`${APPLICATION_NAME}-kieserver-1`

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

`${APPLICATION_NAME}-postgresql-1`

`/usr/libexec/check-container --live`

`${APPLICATION_NAME}-kieserver-2`

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

`${APPLICATION_NAME}-postgresql-2`

`/usr/libexec/check-container --live`

5.1.2.3.3.5. 公開されたポート

デプロイメント	名前	ポート	プロトコル
\${APPLICATION_NAME}-rhpamcentrmon	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
	ping	8888	TCP
\${APPLICATION_NAME}-smartrouter	http	9000	TCP
\${APPLICATION_NAME}-kieserver-1	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
	ping	8888	TCP
\${APPLICATION_NAME}-postgresql-1	–	5432	TCP
\${APPLICATION_NAME}-kieserver-2	jolokia	8778	TCP
	http	8080	TCP
	https	8443	TCP
	ping	8888	TCP
\${APPLICATION_NAME}-postgresql-2	–	5432	TCP

5.1.2.3.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
\${APPLICATION_NAME}-rhpamcentrmon	APPLICATION_USE_RS_PROPERTIES	–	/opt/kie/data/configuration/application-users.properties
	APPLICATION_ROLES_PROPERTIES	–	/opt/kie/data/configuration/application-roles.properties

デプロイメント	変数名	説明	値の例
	KIE_ADMIN_PWD	KIE 管理者のパスワード	\${KIE_ADMIN_PWD}
	KIE_ADMIN_USER	KIE 管理者のユーザー名	\${KIE_ADMIN_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	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	\${MAVEN_MIRROR_URL}
	MAVEN_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_SERVER_CONT ROLLER_USER	KIE サーバーコントロー ラーのユーザー名。 (org.kie.server.controll er.user システムプロパ ティを設定)	\${KIE_SERVER_CON TROLLER_USER}
	KIE_SERVER_CONT ROLLER_PWD	KIE サーバーコントロー ラーのパスワード。 (org.kie.server.controll er.pwd システムプロパ ティを設定)	\${KIE_SERVER_CON TROLLER_PWD}
	KIE_SERVER_CONT ROLLER_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)	\${KIE_SERVER_CON TROLLER_TOKEN}
	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_CENTR AL_HTTPS_KEYSTO RE}
	HTTPS_NAME	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	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}- rhpamcentrmon-ping

デプロイメント	変数名	説明	値の例
	OPENSIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL。	\${SSO_URL}
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レalm名。	\${SSO_REALM}
	SSO_SECRET	Business Central Monitoring RH-SSO クライアントシークレット。	\${BUSINESS_CENTRAL_SSO_SECRET}
	SSO_CLIENT	Business Central Monitoring RH-SSO クライアント名。	\${BUSINESS_CENTRAL_SSO_CLIENT}
	SSO_USERNAME	クライアント作成に使用する RH-SSO レalm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almの管理者のパスワード。	\${SSO_PASSWORD}
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性	\${SSO_PRINCIPAL_ATTRIBUTE}
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトのホスト名を使用する場合は空白にします (例: <application-name>-rhpamcentrmon-<project>.<default-domain-suffix>)。	\${BUSINESS_CENTRAL_HOSTNAME_HTTP}

デプロイメント	変数名	説明	値の例
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure- <application-name>- rhpamcentrmon- <project>.<default- domain-suffix>)。	`\${BUSINESS_CENTRAL_HOSTNAME_HTTPS}`
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。	`\${AUTH_LDAP_URL}`
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	`\${AUTH_LDAP_BIND_DN}`
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	`\${AUTH_LDAP_BIND_CREDENTIAL}`
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	`\${AUTH_LDAP_JAAS_SECURITY_DOMAIN}`
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	`\${AUTH_LDAP_BASE_CTX_DN}`
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	`\${AUTH_LDAP_BASE_FILTER}`
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	`\${AUTH_LDAP_SEARCH_SCOPE}`
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	`\${AUTH_LDAP_SEARCH_TIME_LIMIT}`

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれている場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定した場合には、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	DN の最初から削除される文字列を定義して、ユーザー名を表示します。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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	`\${AUTH_LDAP_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}`
`\${APPLICATION_NAME}`-smartrouter	KIE_SERVER_ROUTER_HOST	–	–
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PORT_TLS	–	9443
	KIE_SERVER_ROUTER_URL_EXTERNAL	ルーターを見つけることのできるパブリック URL。形式： <a href="http://<host>:<port>">http://<host>:<port> ; (ルータープロパティ org.kie.server.router.url.external)	`\${KIE_SERVER_ROUTER_URL_EXTERNAL}`
	KIE_SERVER_ROUTER_ID	コントローラーに接続する場合に使用されるルーター ID (ルータープロパティ org.kie.server.router.id)。 。	`\${KIE_SERVER_ROUTER_ID}`

デプロイメント	変数名	説明	値の例
	KIE_SERVER_ROUTER_NAME	コントローラーに接続するときには使用されるルーター名 (ルータープロパティ <code>org.kie.server.router.name</code>)。	<code>\${KIE_SERVER_ROUTER_NAME}</code>
	KIE_SERVER_ROUTER_ROUTE_NAME	–	<code>\${APPLICATION_NAME}-smartrouter</code>
	KIE_SERVER_ROUTER_SERVICE	–	<code>\${APPLICATION_NAME}-smartrouter</code>
	KIE_SERVER_ROUTER_PROTOCOL	KIE Server ルーターのプロトコル。 (<code>org.kie.server.router.url.external</code> プロパティのビルドに使用)	<code>\${KIE_SERVER_ROUTER_PROTOCOL}</code>
	KIE_SERVER_ROUTER_TLS_KEYSTORE_KEYALIAS	サーバー証明書に関連付けられている名前。	<code>\${KIE_SERVER_ROUTER_HTTPS_NAME}</code>
	KIE_SERVER_ROUTER_TLS_KEYSTORE_PASSWORD	キーストアおよび証明書のパスワード。	<code>\${KIE_SERVER_ROUTER_HTTPS_PASSWORD}</code>
	KIE_SERVER_ROUTER_TLS_KEYSTORE	–	<code>/etc/smartrouter-secret-volume/\${KIE_SERVER_ROUTER_HTTPS_KEYSTORE}</code>
	KIE_SERVER_CONTROLLER_USER	KIE サーバーコントローラーのユーザー名。 (<code>org.kie.server.controller.user</code> システムプロパティを設定)	<code>\${KIE_SERVER_CONTROLLER_USER}</code>
	KIE_SERVER_CONTROLLER_PWD	KIE サーバーコントローラーのパスワード。 (<code>org.kie.server.controller.pwd</code> システムプロパティを設定)	<code>\${KIE_SERVER_CONTROLLER_PWD}</code>

デプロイメント	変数名	説明	値の例
	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)	\${KIE_SERVER_CONTROLLER_TOKEN}
	KIE_SERVER_CONTROLLER_SERVICE	–	\${APPLICATION_NAME}-rhpamcentrmon
	KIE_SERVER_CONTROLLER_PROTOCOL	–	http
	KIE_SERVER_ROUTER_REPO	–	/opt/rhpam-smartrouter/data
	KIE_SERVER_ROUTER_CONFIG_WATCHER_ENABLED	–	true
\${APPLICATION_NAME}-kieserver-1	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}
	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}

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 Server は、タスク関連の操作 (たとえばクエリ) については認証ユーザーをスキップできます。 (org.kie.server.bypass.auth.user システムプロパティを設定)	`\${KIE_SERVER_BYPASS_AUTH_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)	`\${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)	`\${KIE_SERVER_CONTROLLER_TOKEN}`
	KIE_SERVER_CONTROLLER_SERVICE	–	`\${APPLICATION_NAME}-rhpamcentrmon`
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws

デプロイメント	変数名	説明	値の例
	KIE_SERVER_ID	–	\${APPLICATION_NAME}-kieserver-1
	KIE_SERVER_ROUTE_NAME	–	\${APPLICATION_NAME}-kieserver-1
	KIE_SERVER_USE_SECURE_ROUTE_NAME	true の場合は KIE Server は、レポートする Business Central の KIE Server ルートエンドポイントとして secure- <application-name>-kieserver (vs. <application-name>-kieserver) を使用しません。そのため、Business Central はユーザーにセキュアなリンクを表示します。	\${KIE_SERVER1_USE_SECURE_ROUTE_NAME}
	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}
	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_CONTAINER_DEPLOYMENT	–	
	MAVEN_MIRROR_URL	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	\${MAVEN_MIRROR_URL}
	MAVEN_MIRROR_OFF	KIE Server の Maven ミラー設定。	\${MAVEN_MIRROR_OFF}
	MAVEN_REPOS	–	RHPAMCENTR,EXTERNAL

デプロイメント	変数名	説明	値の例
	RHPAMCENTR_MAVEN_REPO_ID	–	repo-rhpamcentr
	RHPAMCENTR_MAVEN_REPO_SERVICE	maven リポジトリの使用状況など、サービスのルックアップを可能にする Business Central に到達可能な場合に使用するサービス名 (必要な場合)	\${BUSINESS_CENTRAL_MAVEN_SERVICE}
	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}`
	KIE_SERVER_ROUTER_SERVICE	–	`\${APPLICATION_NAME}-smartrouter`
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PROTOCOL	KIE Server ルーターのプロトコル。 (org.kie.server.router.url.external プロパティのビルドに使用)	`\${KIE_SERVER_ROUTER_PROTOCOL}`
	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}`
	DATASOURCES	–	RHPAM
	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_DATABASE	KIE Server PostgreSQL データベース名。	`\${KIE_SERVER_POSTGRES_DB}`
	RHPAM_DRIVER	–	postgresql
	KIE_SERVER_PERSISTENCE_DIALECT	KIE Server PostgreSQL Hibernate 方言。	`\${KIE_SERVER_POSTGRES_DIALECT}`

デプロイメント	変数名	説明	値の例
	RHPAM_USERNAME	KIE サーバー PostgreSQL データベースのユーザー名。	`\${KIE_SERVER_POSTGRESQL_USER}`
	RHPAM_PASSWORD	KIE Server PostgreSQL データベースのパスワード。	`\${KIE_SERVER_POSTGRESQL_PWD}`
	RHPAM_SERVICE_HOST	–	`\${APPLICATION_NAME}-postgresql-1`
	RHPAM_SERVICE_PORT	–	5432
	TIMER_SERVICE_DATA_STORE	–	`\${APPLICATION_NAME}-postgresql-1`
	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	EJB タイマーサービスのデータベースデータストアの更新間隔を設定します。	`\${TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL}`
	HTTPS_KEYSTORE_DIR	–	/etc/kieserver-secret-volume
	HTTPS_KEYSTORE	シークレット内のキーストアファイル名	`\${KIE_SERVER_HTTPS_KEYSTORE}`
	HTTPS_NAME	サーバー証明書に関連付けられている名前	`\${KIE_SERVER_HTTPS_NAME}`
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	`\${KIE_SERVER_HTTPS_PASSWORD}`
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSIFT_DNS_PING_SERVICE_NAME	–	`\${APPLICATION_NAME}-kieserver-1-ping`
	OPENSIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL。	`\${SSO_URL}`

デプロイメント	変数名	説明	値の例
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レalm名。	\${SSO_REALM}
	SSO_SECRET	KIE Server 1 の RH-SSO クライアントシークレット。	\${KIE_SERVER1_SSO_SECRET}
	SSO_CLIENT	KIE Server 1 の RH-SSO クライアント名。	\${KIE_SERVER1_SSO_CLIENT}
	SSO_USERNAME	クライアント作成に使用する RH-SSO レalm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almの管理者のパスワード。	\${SSO_PASSWORD}
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}
	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性	\${SSO_PRINCIPAL_ATTRIBUTE}
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>)。	\${KIE_SERVER1_HOSTNAME_HTTP}
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	\${KIE_SERVER1_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}`
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自体の DN に特殊文字(たとえば、正しいユーザーマッピングを防ぐバックスラッシュ)が含まれている場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`

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	DN の最初から削除される文字列を定義して、ユーザー名を表示します。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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	`\${AUTH_LDAP_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}-postgresql-1	POSTGRESQL_USER	KIE サーバー PostgreSQL データベースのユーザー名。	\${KIE_SERVER_POSTGRESQL_USER}
	POSTGRESQL_PASSWORD	KIE Server PostgreSQL データベースのパスワード。	\${KIE_SERVER_POSTGRESQL_PWD}
	POSTGRESQL_DATABASE	KIE Server PostgreSQL データベース名。	\${KIE_SERVER_POSTGRESQL_DB}
	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	PostgreSQL による XA トランザクションの処理を許可します。	\${POSTGRESQL_MAX_PREPARED_TRANSACTIONS}
\${APPLICATION_NAME}-kieserver-2	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}

デプロイメント	変数名	説明	値の例
	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}`
	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}`
	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 Server は、タスク関連の操作 (たとえばクエリー) については認証ユーザーをスキップできます。 (org.kie.server.bypass.auth.user システムプロパティを設定)	`\${KIE_SERVER_BYPASS_AUTH_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)	`\${KIE_SERVER_CONTROLLER_USER}`

デプロイメント	変数名	説明	値の例
	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)	\${KIE_SERVER_CONTROLLER_TOKEN}
	KIE_SERVER_CONTROLLER_SERVICE	–	\${APPLICATION_NAME}-rhpamcentrmon
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	\${APPLICATION_NAME}-kieserver-2
	KIE_SERVER_ROUTE_NAME	–	\${APPLICATION_NAME}-kieserver-2
	KIE_SERVER_USE_SECURE_ROUTE_NAME	True の場合、secure-APPLICATION_NAME-kieserver-2 vs を使用します。 APPLICATION_NAME-kieserver-2 をルート名として使用します。	\${KIE_SERVER2_USE_SECURE_ROUTE_NAME}
	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}
	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}

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTAINER_DEPLOYMENT	–	
	MAVEN_MIRROR_URL	KIE Server が使用する必要のある Maven ミラー。ミラーを設定する場合には、このミラーにはサービスのデプロイに必要なすべてのアーティファクトを含める必要があります。	`\${MAVEN_MIRROR_URL}`
	MAVEN_MIRROR_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	maven リポジトリの使用状況など、サービスのルックアップを可能にする Business Central に到達可能な場合に使用するサービス名 (必要な場合)	`\${BUSINESS_CENTRAL_MAVEN_SERVICE}`
	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}
	KIE_SERVER_ROUTER_SERVICE	–	\${APPLICATION_NAME}-smartrouter
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PROTOCOL	KIE Server ルーターの protocol。 (org.kie.server.router.url.external プロパティのビルドに使用)	\${KIE_SERVER_ROUTER_PROTOCOL}
	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}

デプロイメント	変数名	説明	値の例
	DATASOURCES	–	RHPAM
	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_DATABASE	KIE Server PostgreSQL データベース名。	`\${KIE_SERVER_POSTGRES_DB}`
	RHPAM_DRIVER	–	postgresql
	KIE_SERVER_PERSISTENCE_DIALECT	KIE Server PostgreSQL Hibernate 方言。	`\${KIE_SERVER_POSTGRES_DIALECT}`
	RHPAM_USERNAME	KIE サーバー PostgreSQL データベースのユーザー名。	`\${KIE_SERVER_POSTGRES_USER}`
	RHPAM_PASSWORD	KIE Server PostgreSQL データベースのパスワード。	`\${KIE_SERVER_POSTGRES_PWD}`
	RHPAM_SERVICE_HOST	–	`\${APPLICATION_NAME}-postgresql-2`
	RHPAM_SERVICE_PORT	–	5432
	TIMER_SERVICE_DATA_STORE	–	`\${APPLICATION_NAME}-postgresql-2`
	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	EJB タイマーサービスのデータベースデータストアの更新間隔を設定します。	`\${TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL}`
	HTTPS_KEYSTORE_DIR	–	/etc/kieserver-secret-volume
	HTTPS_KEYSTORE	シークレット内のキーストアファイル名	`\${KIE_SERVER_HTTPS_KEYSTORE}`

デプロイメント	変数名	説明	値の例
	HTTPS_NAME	サーバー証明書に関連付けられている名前	\${KIE_SERVER_HTTPS_NAME}
	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	\${KIE_SERVER_HTTPS_PASSWORD}
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSIFT_DNS_PING_SERVICE_NAME	–	\${APPLICATION_NAME}-kieserver-2-ping
	OPENSIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL。	\${SSO_URL}
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO レルム名。	\${SSO_REALM}
	SSO_SECRET	KIE Server 2 の RH-SSO クライアントシークレット。	\${KIE_SERVER2_SSO_SECRET}
	SSO_CLIENT	KIE Server 2 の RH-SSO クライアント名。	\${KIE_SERVER2_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}

デプロイメント	変数名	説明	値の例
	HOSTNAME_HTTP	http サービスルートのカスタムホスト名。デフォルトホスト名の場合は空白にします (例: <application-name>-kieserver-<project>.<default-domain-suffix>)。	`\${KIE_SERVER2_HOSTNAME_HTTP}`
	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルトのホスト名を使用する場合には空白にします (例: secure-<application-name>-kieserver-<project>.<default-domain-suffix>)。	`\${KIE_SERVER2_HOSTNAME_HTTPS}`
	AUTH_LDAP_URL	認証用に接続する LDAP エンドポイント。	`\${AUTH_LDAP_URL}`
	AUTH_LDAP_BIND_DN	認証に使用するバインド DN	`\${AUTH_LDAP_BIND_DN}`
	AUTH_LDAP_BIND_CREDENTIAL	認証に使用する LDAP の認証情報	`\${AUTH_LDAP_BIND_CREDENTIAL}`
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。	`\${AUTH_LDAP_JAAS_SECURITY_DOMAIN}`
	AUTH_LDAP_BASE_CTX_DN	ユーザー検索を開始する最上位コンテキストの LDAP ベース DN	`\${AUTH_LDAP_BASE_CTX_DN}`

デプロイメント	変数名	説明	値の例
	AUTH_LDAP_BASE_FILTER	認証するユーザーのコンテキストの検索に使用する LDAP 検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。検索フィルターの一般的な例は (uid={0}) です。	`\${AUTH_LDAP_BASE_FILTER}`
	AUTH_LDAP_SEARCH_SCOPE	使用する検索範囲。	`\${AUTH_LDAP_SEARCH_SCOPE}`
	AUTH_LDAP_SEARCH_TIME_LIMIT	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。	`\${AUTH_LDAP_SEARCH_TIME_LIMIT}`
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれている場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`
	AUTH_LDAP_PARSE_USERNAME	DN がユーザー名に対して解析されるかどうかを示すフラグ。true に設定した場合には、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_STRING	DN の最初から削除される文字列を定義して、ユーザー名を表示します。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_USER_NAME_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、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、これは、ユーザーアカウントが存在する DN です。	`\${AUTH_LDAP_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}`
`\${APPLICATION_NAME}`-postgresql-2	POSTGRESQL_USER	KIE サーバー PostgreSQL データベースのユーザー名。	`\${KIE_SERVER_POSTGRESQL_USER}`
	POSTGRESQL_PASSWORD	KIE Server PostgreSQL データベースのパスワード。	`\${KIE_SERVER_POSTGRESQL_PWD}`
	POSTGRESQL_DATABASE	KIE Server PostgreSQL データベース名。	`\${KIE_SERVER_POSTGRESQL_DB}`
	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	PostgreSQL による XA トランザクションの処理を許可します。	`\${POSTGRESQL_MAX_PREPARED_TRANSACTIONS}`

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}`-rhpamcentrmon	businesscentral-keystore-volume	/etc/businesscentral-secret-volume	ssl certs	True
`\${APPLICATION_NAME}`-smartrouter	`\${APPLICATION_NAME}`-smartrouter	/opt/rhpam-smartrouter/data	—	false

デプロイメント	名前	mountPath	目的	readOnly
<code>\${APPLICATION_NAME}-kieserver-1</code>	kieserver-keystore-volume	<code>/etc/kieserver-secret-volume</code>	ssl certs	True
<code>\${APPLICATION_NAME}-postgresql-1</code>	<code>\${APPLICATION_NAME}-postgresql-pvol</code>	<code>/var/lib/pgsql/data</code>	postgresql	false
<code>\${APPLICATION_NAME}-kieserver-2</code>	kieserver-keystore-volume	<code>/etc/kieserver-secret-volume</code>	ssl certs	True
<code>\${APPLICATION_NAME}-postgresql-2</code>	<code>\${APPLICATION_NAME}-postgresql-pvol</code>	<code>/var/lib/pgsql/data</code>	postgresql	false

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
<code>\${APPLICATION_NAME}-postgresql-claim-1</code>	ReadWriteOnce
<code>\${APPLICATION_NAME}-postgresql-claim-2</code>	ReadWriteOnce
<code>\${APPLICATION_NAME}-smartrouter-claim</code>	ReadWriteMany
<code>\${APPLICATION_NAME}-rhpmcentr-claim</code>	ReadWriteMany

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 smartrouter-app-secret kieserver-app-secret

5.2. 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 をスケールするには、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6月25日（金）