



Red Hat Decision Manager 7.2

Red Hat OpenShift Container Platform への Red Hat Decision Manager イミュータブルサー バー環境のデプロイメント

ガイド

Red Hat Decision Manager 7.2 Red Hat OpenShift Container Platform への Red Hat Decision Manager イミュータブルサーバー環境のデプロイメント

ガイド

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Deploying_a_Red_Hat_Decision_Manager_immutable_server_environment_on_Red_Hat_OpenS 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 Decision Manager 7.2 イミュータブルサーバー環境をデプロイする方法を説明します。

目次

はじめに	3
第1章 RED HAT OPENSIFT CONTAINER PLATFORM における RED HAT DECISION MANAGER の概要	4
第2章 OPENSIFT 環境に RED HAT DECISION MANAGER をデプロイする準備	6
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	6
2.2. DECISION SERVER にシークレットの作成	7
2.3. S2I ビルドに使用する DECISION CENTRAL からのソースコードの展開	8
2.4. オフラインで使用する MAVEN リポジトリの準備	8
第3章 イミュータブルサーバーを使用した環境	9
3.1. サービスソースコードからのイミュータブル DECISION SERVER のデプロイ	9
3.2. KJAR サービスからのイミュータブル DECISION SERVER のデプロイ	13
3.3. LDAP ロールマッピングファイルの指定	16
第4章 OPENSIFT テンプレートの参考資料	18
4.1. RHDM72-PROD-IMMUTABLE-KIESERVER.YAML TEMPLATE	18
4.1.1. パラメーター	18
4.1.2. オブジェクト	32
4.1.2.1. サービス	32
4.1.2.2. ルート	33
4.1.2.3. ビルド設定	33
4.1.2.4. デプロイメント設定	33
4.1.2.4.1. トリガー	33
4.1.2.4.2. レプリカ	33
4.1.2.4.3. Pod テンプレート	34
4.1.2.5. 外部の依存関係	44
4.1.2.5.1. シークレット	44
4.2. OPENSIFT の使用に関するクイックリファレンス	44
付録A バージョン情報	47

はじめに

システムエンジニアは、Red Hat OpenShift Container Platform に Red Hat Decision Manager イミュータブルサーバー環境をデプロイして、サービ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します。イミュータブル Decision Server イメージを管理する標準統合ツールを使用できます。新しいサーバーイメージを作成して、ビジネスアセットを追加および更新できます。

前提条件

- 最低でも OpenShift クラスター/名前空間で 2 ギガバイト以上のメモリーが利用できる。
- デプロイメント用の OpenShift プロジェクトが作成されている。
- プロジェクトにログインするには、**oc** コマンドを使用します。**oc** コマンドランツールに関する詳細は、OpenShift の [CLI リファレンス](#) を参照してください。OpenShift Web コンソールを使用してテンプレートをデプロイするには、Web コンソールを使用してログインしている必要もあります。

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

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

この場合、Red Hat Decision Manager のコンポーネントは、別の OpenShift Pod としてデプロイされます。各 Pod のスケールアップと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 Decision Manager の主要コンポーネントが OpenShift で利用できます。

- Decision Server (**実行サーバー (Execution Server)** または **KIE Server** と呼ばれる) は、インフラストラクチャーの要素でデシジョンサービスやその他のデプロイ可能なアセットを実行します (これらすべてで総称で **サービス** と呼ぶ)。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。

Decision Server Pod は自由にスケールアップして、同一または異なるホストで実行するコピーに必要な数だけ提供できます。Pod をスケールアップまたはスケールダウンすると、そのコピーはすべて同じプロセスで実行します。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。

個別の Decision Server Pod をデプロイして、異なるサービス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。個別の複製 Decision Server Pod を必要な数だけ設定することができます。

- Decision Central は、オーサリングサービス用の Web ベースの対話環境で、Decision Central では管理コンソールが提供されています。また、管理コンソールも提供します。Decision Central を使用してサービスを開発し、Decision Server にデプロイできます。Decision Central は集中型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。

Decision 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、Decision 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、Decision Central の高可用性機能はテクノロジープレビューです。

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。

- **オーサリングまたは管理対象の環境:** Decision Central 経由でサービスの作成や変更、Decision Server でのサービスの実行に使用可能な環境のアーキテクチャー。この環境は、オーサリング作業用の Decision Central を提供する Pod と、サービス実行用の Decision Server 1 つまたは複数を提供する Pod で設定されます。Decision Server それぞれが 1 つの Pod となっており、必要に応じてスケールアップやスケールダウンすることで複製が可能です。Decision Central を使用して、各 Decision Server にサービスをデプロイおよびアンデプロイできます。この環境をデプロイする方法については、[Red Hat OpenShift Container Platform への Red Hat Decision Manager オーサリングまたは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 の Pod のデプロイ時に、サービスまたはサービスグループをロードおよび起動するイメージをビルドします。この Pod でサービスを停止したり、新しいサービスを追加したりすることはできません。サービスの別のバージョンを使用したり、別の方法で設定を変更する必要がある場合は、新規のサーバーイメージをデプロイして、古いサーバーと入れ替えます。このシステムでは、Decision Server は OpenShift 環境の他の Pod のように実行されるので、コンテナベースの統合ワークフローはどれでも使用でき、別のツールを使用して Pod を管理する必要はありません。このような環境のデプロイメント手順は、[Red Hat OpenShift Container Platform への Red Hat Decision Manager イミュータブルサーバー環境のデプロイメント](#) 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、Decision Central と Decision 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方法については、[Red Hat OpenShift Container Platform への Red Hat Decision Manager 試用環境のデプロイ](#) を参照してください。

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

第2章 OPENSIFT 環境に RED HAT DECISION MANAGER をデプロイする準備

OpenShift 環境に Red Hat Decision Manager をデプロイする前に、準備タスクをいくつか完了する必要があります。追加イメージ (たとえば、デシジョンサービスの新しいバージョン、または別のデシジョンサービス) をデプロイする場合は、この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 OpenShift Container Platform で Red Hat Decision Manager コンポーネントをデプロイするには、OpenShift が Red Hat レジストリーから正しいイメージをダウンロードできるようにする必要があります。これらのイメージをダウンロードするには、OpenShift ではイメージの場所情報 (イメージストリームと呼ばれる) が必要です。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。

それ以外の場合には、独自のプロジェクトでレジストリーの認証を設定して、その同じプロジェクト内にイメージストリームを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 は、Red Hat のレジストリーアクセス用にユーザー名とパスワードで設定する場合には、以下のコマンドを使用します。

```
$ oc get imagestreamtag -n openshift | grep rhdm72-decisioncentral-openshift
$ oc get imagestreamtag -n openshift | grep rhdm72-kieserver-openshift
```

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

3. コマンドの1つまたは複数の出力が空白の場合や、Red Hat レジストリーにアクセスするために、OpenShift をユーザー名およびパスワードで設定していない場合は、以下の手順を実行してください。
 - a. **oc** コマンドで OpenShift にログインして、プロジェクトがアクティブであることを確認します。
 - b. [Registry Service Accounts for Shared Environments](#) で説明されている手順を実行します。Red Hat カスタマーポータルにログインして、このドキュメントにアクセスし、レジストリーサービスアカウントを作成する手順を実行します。
 - 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 [rhdm-7.2.0-openshift-templates.zip](#) 製品の配信可能ファイルをダウンロードし、**rhdm72-image-streams.yaml** ファイルを展開してください。

g. 以下のアクションの1つを完了します。

- 次のコマンドを実行します。

```
$ oc create -f rhdm72-image-streams.yaml
```

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



注記

上記の手順を完了したら、イメージストリームを独自のプロジェクトの名前空間にインストールします。この手順を使用してイメージストリームをインストールした場合には、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. DECISION SERVER にシークレットの作成

OpenShift は、シークレット と呼ばれるオブジェクトを使用してパスワードやキーストアなどの機密情報を保持します。詳細は、OpenShift ドキュメントの [Secrets](#) の章を参照してください。

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



注記

HTTPS のサポートなしの Decision Server のデプロイを予定している場合は、シークレットオブジェクトを作成しなくても構いません。

手順

1. Decision Server の SSL 暗号化の秘密鍵および公開鍵を使用して SSL キーストアを生成します。プロダクション環境で、期待する Decision Server の URL に一致する有効な署名付き証明書を生成します。キーストアを **keystore.jks** ファイルに保存します。証明書の名前と、キーストアファイルのパスワードを記録します。
自己署名型または購入する SSL 証明書を使用してキーストアを作成する方法は [SSL 暗号化キーおよび証明書の生成](#) を参照してください。
2. **oc** コマンドを使用して、新しいキーストアファイルからシークレット **kieserver-app-secret** を生成します。

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

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

オーサリングサービスに Decision Central を使用する場合は、サービスのソースコードを展開して、S2I ビルドを使用する別の Git リポジトリ (GitHub や GitLab のオンプレミスインストールなど) に配置できます。

手順

1. 以下のコマンドを使用してソースコードを展開します。

```
git clone ssh://adminUser@decision-central-host:8001/MySpace/MyProject
```

以下を置き換えます。

- **adminUser**: Decision Central の管理ユーザー
- **decision-central-host**: Decision Central を実行しているホスト
- **MySpace**: プロジェクトが配置された Decision Central 領域の名前
- **MyProject**: プロジェクトの名前

2. S2I ビルドの別の Git リポジトリへのソースコードのアップロード

2.4. オフラインで使用する MAVEN リポジトリの準備

Red Hat OpenShift Container Platform 環境に公開インターネットへの送信アクセスが設定されていない場合には、必要なアーティファクトすべてのミラーが含まれる Maven リポジトリを用意して、このリポジトリをソースからイメージへのビルド (S2I) で使用できるようにする必要があります。

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にはこの手順は省略してください。

手順

1. **mvn clean install** コマンドを使用して、任意のマシンでサービスのソースをビルドします。
2. ダウンロードした Maven アーティファクトをマシンから内部 Maven リポジトリ (Nexus など) にコピーします。
3. このリポジトリを Red Hat OpenShift Container Platform 環境で使用できるようにします。

第3章 イミュータブルサーバーを使用した環境

事前に読み込んだサービスで Decision Server を実行する Pod を1つまたは複数含めた環境をデプロイできます。必要に応じて、Decision Server Pod ごとに個別にスケーリング可能です。

この場合には、イメージ作成時に Decision Server にサービス (KJAR ファイル) を読み込む必要があります。実行中の Decision Server に対して、サービスの読み込みや読み込み解除はできません。このアプローチの利点は、サービスが含まれる Decision Server は他のコンテナ化されたサービスのように実行され、特別な管理が必要ない点です。Decision Server は、OpenShift 環境で他の Pod のように実行され、必要に応じて、コンテナベースの統合ワークフローを使用できます。

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

オーサリングサービスに Decision Central を使用する場合は、プロセスのソースを展開して、S2I ビルドで使用する別の Git リポジトリ (GitHub や、GitLab のオンプレミスインストールなど) に配置できます。

または、KJAR ファイルとしてすでにビルドされているサービスを使用して同様の Decision Server デプロイメントを作成できます。このような場合には、Maven リポジトリにサービスを提供する必要があります。KJAR ファイルは、Pod の起動時、Maven リポジトリから取得され、その後、更新および変更されません。KJAR ファイルは Pod の起動時に Maven リポジトリから取得し、その後更新も変更もされません。このファイルは再起動時や Pod のスケーリング時に必ず取得されるので、デプロイメントをイミュータブルに保つには、Maven リポジトリで変更がないようにする必要があります。

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

3.1. サービスソースコードからのイミュータブル DECISION SERVER のデプロイ

サービスのソースコードからイミュータブル Decision Server f をデプロイするには、**rhdm72-prod-immutable-kieserver.yaml** テンプレートファイルを使用します。**rhdm-7.2.0-openshift-templates.zip** の製品配信可能ファイルからこのファイルを展開できます。[Software Downloads](#) ページからこのファイルをダウンロードできます。

イミュータブル Decision Server をデプロイする場合は、デプロイメントの手順で、このサーバー上にあるサービスのソースコードを取得し、これらのサービスをビルドして、サーバーイメージに組み込みます。

手順

1. 以下の方法を使用してテンプレートをデプロイします。
 - OpenShift Web UI では、**Add to Project** → **Import YAML / JSON**を選択し、**rhdm72-prod-immutable-kieserver.yaml** ファイルを選択またはその内容を貼り付けます。**Add Template** ウィンドウで、**Process the template** が選択されていることを確認し、**Continue** をクリックします。
 - OpenShift コマンドラインコンソールを使用するには、以下のコマンドラインを準備します。

```
oc new-app -f <template-path>/rhdm72-prod-immutable-kieserver.yaml -p
KIE_SERVER_HTTPS_SECRET=kieserver-app-secret
```

このコマンドラインで以下を行います。

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

2. 必要に応じて以下のパラメーターを設定します。

- **KIE Server キーストアのシークレット名 (KIE_SERVER_HTTPS_SECRET):** 「[Decision Server にシークレットの作成](#)」で作成した Decision Server のシークレットの名前。
- **アプリケーション名 (APPLICATION_NAME):** OpenShift アプリケーションの名前。Decision Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。同じテンプレートと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で、Decision Server が参加する Decision Central でサーバー設定 (サーバーテンプレート) の名前が決まります。複数の Decision Server をデプロイする場合には、サーバーごとに異なるアプリケーション名を指定する必要があります。
- **KIE Server Certificate Name (KIE_SERVER_HTTPS_NAME):** 「[Decision 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):** 「[Decision Server にシークレットの作成](#)」で作成したキーストアのパスワード。
- **KIE Server Container Deployment (KIE_SERVER_CONTAINER_DEPLOYMENT):** ソースからビルドしたデシジョンサービス (KJAR ファイル) の識別情報。フォーマットは **<containerId>=<groupId>:<artifactId>:<version>** になります。区切り記号 | を使用して 2 つ以上の KJAR ファイルを指定できます (例: **containerId=groupId:artifactId:version|c2=g2:a2:v2**)。Maven ビルドプロセスは、Git リポジトリのソースからこのようなファイルをすべて生成する必要があります。
- **Git Repository URL (SOURCE_REPOSITORY_URL):** デシジョンサービスのソースを含む Git リポジトリの URL。
- **Git Reference (SOURCE_REPOSITORY_REF):** Git リポジトリのブランチ。
- **Context Directory (CONTEXT_DIR):** Git リポジトリからダウンロードしたプロジェクトのソースへのパス。
- **アーティファクトディレクトリー (ARTIFACT_DIR):** Maven のビルドに成功したあとに必要なバイナリーファイル (KJAR ファイル、およびその他の必要なファイル) を含むプロジェクトのパス。通常、このディレクトリーはビルドのターゲットディレクトリーです。ただし、Git リポジトリのこのディレクトリーにビルド済みのバイナリーを提供できません。
- **ImageStream 名前空間 (IMAGE_STREAM_NAMESPACE):** イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「[イメージストリームとイメージレジストリーの可用性確認](#)」を参照) は、namespace が **openshift** にな

ります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。

3. お使いのビルドに公開されている Maven ツリーで利用できない依存関係があり、別のリポジトリが必要な場合は、パラメーターを設定してこのリポジトリを提供します。
 - **Maven リポジトリの URL (MAVEN_REPO_URL):** Maven リポジトリの URL。
 - **Maven repository username (MAVEN_REPO_USERNAME):** Maven リポジトリのユーザー名。
 - **Maven リポジトリのパスワード (MAVEN_REPO_PASSWORD):** Maven リポジトリのパスワード。
4. RH-SSO または LDAP 認証を使用する場合は、以下の追加設定を実行します。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。
 - a. RH-SSO または LDAP サービスでは、デプロイメントパラメーターで全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**
 - b. Red Hat Single Sign On (RH-SSO) 認証を設定する場合には、Red Hat Decision Manager を適用する RH-SSO レルムが必要です。また、RH-SSO 内にクライアントが必要です。RH-SSO で設定可能なユーザールールについては、[ロールおよびユーザー](#) を参照してください。

以下の手順の1つを使用します。

- i. RH-SSO 内にすでに Red Hat Decision Manager のクライアントが存在する場合には、テンプレートで以下のパラメーターを設定してください。
 - **RH-SSO URL (SSO_URL):** RH-SSO の URL。
 - **RH-SSO レルム名 (SSO_REALM):** Red Hat Decision Manager の RH-SSO レルム。
 - **KIE Server の RH-SSO クライアント名 (KIE_SERVER_SSO_CLIENT):** Decision Server の RH-SSO クライアント名。
 - **KIE Server の RH-SSO クライアントのシークレット (KIE_SERVER_SSO_SECRET):** Decision Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
 - **RH-SSO 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。
- ii. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。
 - **RH-SSO URL (SSO_URL):** RH-SSO の URL。

- **RH-SSO レalm名 (SSO_REALM)**: Red Hat Decision Manager の RH-SSO レalm。
 - **KIE Server の RH-SSO クライアント名 (KIE_SERVER_SSO_CLIENT)**: Decision Server 向けに RH-SSO に作成するクライアント名。
 - **KIE Server の RH-SSO クライアントのシークレット (KIE_SERVER_SSO_SECRET)**: Decision Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
 - **KIE Server カスタムの http ルートホスト名 (KIE_SERVER_HOSTNAME_HTTP)**: Decision Server の HTTP エンドポイントに使用する完全修飾ホスト名。RH-SSO にクライアントを作成する必要がある場合には、このパラメーターを空白のままにしてください。
 - **KIE Server カスタムの https ルートホスト名 (KIE_SERVER_HOSTNAME_HTTPS)**: Decision Server の HTTPS エンドポイントに使用する完全修飾ホスト名。RH-SSO にクライアントを作成する必要がある場合には、このパラメーターを空白のままにしてください。
 - **RH-SSO レalmの管理者のユーザー名 (SSO_USERNAME) および RH-SSO レalmの管理者のパスワード (SSO_PASSWORD)**: Red Hat Decision Manager の RH-SSO レalmの管理者ユーザーに指定するユーザー名とパスワード
 - **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。
- c. LDAP を設定するには、テンプレートの **AUTH_LDAP*** パラメーターを設定します。これらのパラメーターは、Red Hat JBoss EAP の LdapExtended ログインモジュールの設定に対応します。これらの設定に関する説明は、[LdapExtended ログインモジュール](#) を参照してください。
- LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。
- **RoleMapping rolesProperties ファイルパス (AUTH_ROLE_MAPPER_ROLES_PROPERTIES)**:
`/opt/eap/standalone/configuration/rolemapping/rolemapping.properties` など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「[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。
5. 使用している方法に応じて、環境の作成を終了します。
- OpenShift Web UI の場合は **Create** をクリックします。
 - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。

- 完了し、コマンドラインを実行します。

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

KJAR サービスからイミュータブル Decision Server をデプロイするには、`rhdm72-kieserver.yaml` テンプレートファイルを使用します。このファイルは、

`rhdm-7.2.0-openshift-templates.zip` の製品配信可能ファイルから展開できます。 [Software Downloads](#) ページからこのファイルをダウンロードできます。

このデプロイメントの方法は、Decision Server を使用して Pod の起動時に必要な KJAR ファイルをすべて取得します。

手順

1. 以下の方法を使用してテンプレートをデプロイします。

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

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

このコマンドラインで以下を行います。

- `<template-path>` は、テンプレートファイルのパスに置き換えます。
- `<template-file-name>` は、テンプレート名に置き換えます。
- 必要なパラメーターに設定するために必要な数だけ `-p PARAMETER=value` ペアを使用します。テンプレートファイルを表示して、すべてのパラメーターの説明を確認します。

2. 必要に応じて以下のパラメーターを設定します。

- **KIE Server キーストアのシークレット名 (KIE_SERVER_HTTPS_SECRET):** [「Decision Server にシークレットの作成」](#) で作成した Decision Server のシークレットの名前。
- **アプリケーション名 (APPLICATION_NAME):** OpenShift アプリケーションの名前。Decision Server のデフォルト URL で使用されます。OpenShift はアプリケーション名を使用して、デプロイメント設定、サービス、ルート、ラベル、およびアーティファクトの個別のセットを作成します。同じテンプレートを同じプロジェクトで使用して複数のアプリケーションをデプロイすることもできますが、その場合はアプリケーション名を同じにすることはできません。また、アプリケーション名で、Decision Server が参加する Decision Central でサーバー設定 (サーバーテンプレート) の名前が決まります。複数の Decision Server をデプロイする場合には、サーバーごとに異なるアプリケーション名を指定する必要があります。
- **KIE Server Certificate Name (KIE_SERVER_HTTPS_NAME):** [「Decision 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):** 「[Decision Server にシークレットの作成](#)」 で作成したキーストアのパスワード。
 - **KIE Server コンテナのデプロイメント (KIE_SERVER_CONTAINER_DEPLOYMENT):** デプロイメントが Maven リポジトリからプルする必要のあるデシジョンサービス (KJAR ファイル) の識別情報。フォーマットは `<containerId>=<groupId>:<artifactId>:<version>` になります。区切り記号 `|` を使用して 2 つ以上の KJAR ファイルを指定できます (例: `containerId=groupId:artifactId:version|c2=g2:a2:v2`)。
 - **Maven リポジトリの URL (MAVEN_REPO_URL):** Maven リポジトリの URL。
 - **Maven repository username (MAVEN_REPO_USERNAME):** Maven リポジトリのユーザー名。
 - **Maven リポジトリのパスワード (MAVEN_REPO_PASSWORD):** Maven リポジトリのパスワード。
 - **KIE サーバー管理の無効化 (KIE_SERVER_MGMT_DISABLED):** イミュータブルデプロイメントの場合は、このパラメーターを **true** に設定する必要があります。
 - **KIE サーバーの起動ストラテジー (KIE_SERVER_STARTUP_STRATEGY):** イミュータブルデプロイメントの場合はこのパラメーターを **LocalContainersStartupStrategy** に設定する必要があります。
 - **ImageStream 名前空間 (IMAGE_STREAM_NAMESPACE):** イメージストリームが利用可能な名前空間。OpenShift 環境でイメージストリームが利用可能な場合 (「[イメージストリームとイメージレジストリーの可用性確認](#)」を参照) は、namespace が **openshift** になります。イメージストリームファイルをインストールしている場合は、名前空間が OpenShift プロジェクトの名前になります。
3. RH-SSO または LDAP 認証を使用する場合は、以下の追加設定を実行します。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。
- a. RH-SSO または LDAP サービスでは、デプロイメントパラメーターで全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**
 - b. Red Hat Single Sign On (RH-SSO) 認証を設定する場合には、Red Hat Decision Manager を適用する RH-SSO レalmが必要です。また、RH-SSO 内にクライアントが必要です。RH-SSO で設定可能なユーザールールについては、[ロールおよびユーザー](#) を参照してください。

以下の手順の 1 つを使用します。

- i. RH-SSO 内にすでに Red Hat Decision Manager のクライアントが存在する場合には、テンプレートで以下のパラメーターを設定してください。
 - **RH-SSO URL (SSO_URL):** RH-SSO の URL。
 - **RH-SSO レalm名 (SSO_REALM):** Red Hat Decision Manager の RH-SSO レalm。

- KIE Server の RH-SSO クライアント名 (**KIE_SERVER_SSO_CLIENT**): Decision Server の RH-SSO クライアント名。
 - KIE Server の RH-SSO クライアントのシークレット (**KIE_SERVER_SSO_SECRET**): Decision Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
 - RH-SSO 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。
- ii. RH-SSO に Red Hat Decision Manager のクライアントを作成する場合は、テンプレートで以下のパラメーターを設定します。
- RH-SSO URL (**SSO_URL**): RH-SSO の URL。
 - RH-SSO レalm名 (**SSO_REALM**): Red Hat Decision Manager の RH-SSO レalm。
 - KIE Server の RH-SSO クライアント名 (**KIE_SERVER_SSO_CLIENT**): Decision Server 向けに RH-SSO に作成するクライアント名。
 - KIE Server の RH-SSO クライアントのシークレット (**KIE_SERVER_SSO_SECRET**): Decision Server のクライアントに対して RH-SSO に設定するシークレットの文字列。
 - KIE Server カスタムの http ルートホスト名 (**KIE_SERVER_HOSTNAME_HTTP**): Decision Server の HTTP エンドポイントに使用する完全修飾ホスト名。RH-SSO にクライアントを作成する必要がある場合には、このパラメーターを空白のままにしてください。
 - KIE Server カスタムの https ルートホスト名 (**KIE_SERVER_HOSTNAME_HTTPS**): Decision Server の HTTPS エンドポイントに使用する完全修飾ホスト名。RH-SSO にクライアントを作成する必要がある場合には、このパラメーターを空白のままにしてください。
 - RH-SSO レalmの管理者のユーザー名 (**SSO_USERNAME**) および RH-SSO レalmの管理者のパスワード (**SSO_PASSWORD**): Red Hat Decision Manager の RH-SSO レalmの管理者ユーザーに指定するユーザー名とパスワード
 - 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。
- c. LDAP を設定するには、テンプレートの **AUTH_LDAP*** パラメーターを設定します。これらのパラメーターは、Red Hat JBoss EAP の `LdapExtended` ログインモジュールの設定に対応します。これらの設定に関する説明は、[LdapExtended ログインモジュール](#) を参照してください。
- LDAP サーバーがデプロイメントに必要な全ロールを定義していない場合は、LDAP グループを Red Hat Decision Manager ロールにマッピングしてください。LDAP のロールマッピングを有効にするには、以下のパラメーターを設定します。
- RoleMapping rolesProperties ファイルパス (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**):
`/opt/eap/standalone/configuration/rolemapping/rolemapping.properties` など、ロールのマッピングを定義するファイルの完全修飾パス名。このファイルを指定して、該当

するすべてのデプロイメント設定でこのパスにマウントする必要があります。これを実行する方法については、「[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。
4. 使用している方法に応じて、環境の作成を終了します。
 - OpenShift Web UI の場合は **Create** をクリックします。
 - **This will create resources that may have security or project behavior implications** メッセージが表示された場合は、**Create Anyway** をクリックします。
 - 完了し、コマンドラインを実行します。

3.3. 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. ファイルから OpenShift 設定マップを作成します。次のコマンドを実行します。

```
oc create configmap ldap_role_mapping --from-file=<new_name>=<existing_name>
```

<new_name> は、Pod に指定するファイルの名前

(**AUTH_ROLE_MAPPER_ROLES_PROPERTIES** ファイルで指定した名前と同じである必要があります) に置き換えます。また、**existing_name** は、作成したファイル名に置き換えます。

以下に例を示します。

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

3. ロールマッピング用に指定した全デプロイメント設定に設定マップをマウントします。この環境では、次のデプロイメント設定が影響を受ける可能性があります。
 - **myapp-rhpamcentrmon**: Decision Central Monitoring
 - **myapp-kieserver**: Decision Server

myapp はアプリケーション名です。複数の Decision Server の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可能性があります。

すべてのデプロイメント設定について、以下のコマンドを実行します。

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

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

第4章 OPENSIFT テンプレートの参考資料

Red Hat Decision Manager には、以下の OpenShift テンプレートが含まれています。このテンプレートにアクセスするには、Red Hat カスタマーポータルでの [Software Downloads](#) ページから、製品の配信可能ファイル `rhdm-7.2.0-openshift-templates.zip` をダウンロードして展開します。

- `rhdm72-prod-immutable-kieserver.yaml` で、イミュータブル Decision Server が設定されます。このテンプレートのデプロイメントには、Decision Server 上で実行予定の1つまたは複数サービスの source-to-image (S2I) ビルドが含まれます。このテンプレートの詳細は、[「rhdm72-prod-immutable-kieserver.yaml template」](#) を参照してください。

4.1. RHDM72-PROD-IMMUTABLE-KIESERVER.YAML TEMPLATE

Red Hat Decision Manager 7.2 での実稼働環境におけるイミュータブル KIE サーバー向けのアプリケーションテンプレート

4.1.1. パラメーター

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

変数名	イメージの環境変数	説明	値の例	必須
APPLICATION_NAME	–	アプリケーションの名前。	myapp	True
KIE_ADMIN_USER	KIE_ADMIN_USER	KIE 管理者のユーザー名	adminUser	False
KIE_ADMIN_PASSWORD	KIE_ADMIN_PASSWORD	KIE 管理者のパスワード	–	False
KIE_SERVER_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します)	executionUser	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_PWD	KIE_SERVER_PWD	KEI サーバーへの接続に使用する KIE サーバーパスワード。生成される値は s2i に使用する提案として使用できます (org.kie.server.pwd システムプロパティを設定します)	–	False
IMAGE_STREAM_NAMESPACE	–	Red Hat ミドルウェアイメージの ImageStreams がインストールされている namespace これらの ImageStreams は通常 OpenShift の名前空間にインストールされています。ImageStreams を別の namespace/プロジェクトにインストールしている場合には、これを変更するだけで結構です。	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	KIE Server に使用するイメージストリームの名前。デフォルトは "rhdm72-kieserver-openshift" です。	rhdm72-kieserver-openshift	True
IMAGE_STREAM_TAG	–	イメージストリーム内のイメージへの名前付きポインター。デフォルトは 1.1 です。	1.1	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 サーバーのコントローラーのユーザー名 (org.kie.server.controller.user システムプロパティーを設定します)	controllerUser	False
KIE_SERVER_CONTROLLER_PWD	KIE_SERVER_CONTROLLER_PWD	KIE サーバーのコントローラーのパスワード (org.kie.server.controller.pwd システムプロパティーを設定します)	–	False
KIE_SERVER_CONTROLLER_TOKEN	KIE_SERVER_CONTROLLER_TOKEN	Bearer 認証用の KIE サーバーコントローラーのトークン (org.kie.server.controller.token システムプロパティーを設定します)	–	False
KIE_SERVER_CONTROLLER_SERVICE	KIE_SERVER_CONTROLLER_SERVICE	オプションのスタンドアロンコントローラーのサービス名。アプリケーションは、このサービス名を使用してコントローラーに登録します。(設定されている場合は、ホストとポートの検出に使用されます)	–	False
KIE_SERVER_CONTROLLER_HOST	KIE_SERVER_CONTROLLER_HOST	KIE サーバーコントローラーのホスト (org.kie.server.controller システムプロパティーの設定に使用)	my-app-controller-ocpuser.os.example.com	False

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_CONTROLLER_PORT	KIE_SERVER_CONTROLLER_PORT	KIE サーバーコントローラーのポート (org.kie.server.controller システムプロパティーの設定に使用)	8080	False
DROOLS_SERVER_FILTER_CLASSES	DROOLS_SERVER_FILTER_CLASSES	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティーを設定)	true	False
KIE_MBEANS	KIE_MBEANS	KIE Server の mbeans の有効化/無効化 (システムプロパティー kie.mbeans および kie.scanner.mbeans を設定)。	enabled	False
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
KIE_SERVER_HOSTNAME_HTTPS	HOSTNAME_HTTPS	https サービスルートのカスタムのホスト名。デフォルト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_SERVER_USE_SECURE_ROUTE_NAME	KIE_SERVER_USE_SECURE_ROUTE_NAME	True の場合、secure-APPLICATION_NAME-kieserver vs APPLICATION_NAME-kieserver をルート名として使用します。	false	False
KIE_SERVER_HTTPS_SECRET	–	キーストアファイルを含むシークレット名	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。	keystore.jks	False
KIE_SERVER_HTTPS_NAME	HTTPS_NAME	サーバー証明書に関連付けられている名前	jboss	False
KIE_SERVER_HTTPS_PASSWORD	HTTPS_PASSWORD	キーストアおよび証明書のパスワード	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	KIE_SERVER_BYPASS_AUTH_USER	KIE サーバーのバイパス認証ユーザー (org.kie.server.bypass.auth.user システムプロパティを設定します)	false	False
KIE_SERVER_CONTAINER_DEPLOYMENT	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE サーバーコンテナのデプロイメント設定。形式: containerId=groupId:artifactId:version c2=g2:a2:v2	rhdm-kieserver-hellorules=org.openshift.quickstarts:rhdm-kieserver-hellorules:1.4.0-SNAPSHOT	True
SOURCE_REPOSITORY_URL	–	アプリケーションの Git ソース URL。	https://github.com/jboss-container-images/rhdm-7-openshift-image.git	True
SOURCE_REPOSITORY_REF	–	Git ブランチ/タグ参照。	master	False

変数名	イメージの環境変数	説明	値の例	必須
CONTEXT_DIR	–	ビルドする Git プロジェクト内のパス。ルートプロジェクトディレクトリーの場合は空になります。	quickstarts/hello-rules/hellorules	False
GITHUB_WEBHOOK_SECRET	–	GitHub トリガーシークレット。	–	True
GENERIC_WEBHOOK_SECRET	–	汎用ビルドのトリガーシークレット。	–	True
MAVEN_MIRROR_URL	–	S2I ビルドに使用する Maven ミラー	–	False
MAVEN_REPO_ID	EXTERNAL_MAVEN_REPO_ID	maven リポジトリーに使用する id (設定されている場合)。デフォルトは無作為に作成されます。	my-repo-id	False
MAVEN_REPO_URL	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリーへの完全修飾 URL。	–	False
MAVEN_REPO_USERNAME	EXTERNAL_MAVEN_REPO_USERNAME	Maven リポジトリーにアクセスするユーザー名 (必要な場合)	–	False
MAVEN_REPO_PASSWORD	EXTERNAL_MAVEN_REPO_PASSWORD	Maven リポジトリーにアクセスするパスワード (必要な場合)。	–	False
DECISION_CENTRAL_MAVEN_SERVICE	RHDMCENTRAL_MAVEN_REPO_SERVICE	必要な場合には、サービスルックアップができるようにオプションの Decision Central (maven リポジトリーの使用向け) に到達可能な場合に使用するサービス名	myapp-rhdmcentr	False

変数名	イメージの環境変数	説明	値の例	必須
DECISION_CENTRAL_MAVEN_USERNAME	RHDMCENTRAL_MAVEN_REPO_USERNAME	EAP 内の Decision 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
DECISION_CENTRAL_MAVEN_PASSWORD	RHDMCENTRAL_MAVEN_REPO_PASSWORD	EAP 内の Decision 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
ARTIFACT_DIR	—	deploymentto フォルダーにコピーするアーカイブ取得元のディレクトリを一覧。指定されていない場合は、全アーカイブまたはターゲットがコピーされます。	—	False
KIE_SERVER_MEMORY_LIMIT	—	KIE Server のコンテナのメモリー制限	1Gi	False
KIE_SERVER_MGMT_DISABLED	KIE_SERVER_MGMT_DISABLED	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。 org.kie.server.management.api.disabled プロパティを true に、 org.kie.server.startup.strategy プロパティを LocalContainersStartupStrategy に設定します。	true	True

変数名	イメージの環境変数	説明	値の例	必須
KIE_SERVER_STARTUP_STRATEGY	KIE_SERVER_STARTUP_STRATEGY	LocalContainersStartupStrategy に設定した場合には、コントローラーが設定されていて利用できないときでも KIE サーバーが起動し、ローカルの設定で機能できます。	LocalContainersStartupStrategy	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
KIE_SERVER_SSO_CLIENT	SSO_CLIENT	KIE Server の RH-SSO クライアント名。	–	False
KIE_SERVER_SSO_SECRET	SSO_SECRET	KIE Server の RH-SSO クライアントシークレット	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	SSO_USERNAME	クライアント作成に使用する RH-SSO レalm 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alm の管理者のパスワード。	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。	false	False
SSO_PRINCIPAL_ATTRIBUTE	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性	preferred_username	False

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

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

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

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

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

変数名	イメージの環境変数	説明	値の例	必須
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	roleAttributeID にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の 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。このプロパティーは、ロールを置換ロールに対してマップするプロパティーファイルまたはリソースの完全修飾ファイルパスまたはファイル名を定義します。形式は original_role=role1,role2,role3 になります。	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	AUTH_ROLE_MAPPER_REPLACE_ROLE	現在のロールを追加するか、マップされたロールに現在のロールを置き換えるか。true に設定した場合は、置き換えられます。	–	False

4.1.2. 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。

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

4.1.2.2. 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、サービスセクター、セキュリティ設定 (任意) で設定されます。詳細は、[Openshift ドキュメント](#) を参照してください。

サービス	セキュリティ	ホスト名
<code>\${APPLICATION_NAME}-kieserver-http</code>	なし	<code>\${KIE_SERVER_HOSTNAME_HTTP}</code>
<code>\${APPLICATION_NAME}-kieserver-https</code>	TLS パススルー	<code>\${KIE_SERVER_HOSTNAME_HTTPS}</code>

4.1.2.3. ビルド設定

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

S2I イメージ	リンク	ビルドの出力	BuildTriggers および設定
rhdm72-kieserver-openshift:1.1	rhdm-7/rhdm72-kieserver-openshift	<code>\${APPLICATION_NAME}-kieserver:latest</code>	GitHub、Generic、ImageChange、ConfigChange

4.1.2.4. デプロイメント設定

OpenShift のデプロイメントは、デプロイメント設定と呼ばれるユーザー定義のテンプレートを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。

4.1.2.4.1. トリガー

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

デプロイメント	トリガー
<code>\${APPLICATION_NAME}-kieserver</code>	ImageChange

4.1.2.4.2. レプリカ

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

デプロイメント	レプリカ
<code>\${APPLICATION_NAME}-kieserver</code>	2

4.1.2.4.3. Pod テンプレート

4.1.2.4.3.1. サービスアカウント

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

デプロイメント	サービスアカウント
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${APPLICATION_NAME}-kieserver</code>

4.1.2.4.3.2. イメージ

デプロイメント	イメージ
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${APPLICATION_NAME}-kieserver</code>

4.1.2.4.3.3. Readiness Probe

`${APPLICATION_NAME}-kieserver`

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

4.1.2.4.3.4. Liveness Probe

`${APPLICATION_NAME}-kieserver`

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

4.1.2.4.3.5. 公開されたポート

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

デプロイメント	名前	ポート	プロトコル
	ping	8888	TCP

4.1.2.4.3.6. イメージの環境変数

デプロイメント	変数名	説明	値の例
\${APPLICATION_NAME}-kieserver	DROOLS_SERVER_FILTER_CLASSES	KIE Server のクラスフィルター (org.drools.server.filter.classes システムプロパティを設定)	\${DROOLS_SERVER_FILTER_CLASSES}
	KIE_ADMIN_USER	KIE 管理者のユーザー名	\${KIE_ADMIN_USER}
	KIE_ADMIN_PWD	KIE 管理者のパスワード	\${KIE_ADMIN_PWD}
	KIE_MBEANS	KIE Server の mbeans の有効化/無効化 (システムプロパティ kie.mbeans および kie.scanner.mbeans を設定)。	\${KIE_MBEANS}
	KIE_SERVER_BYPASS_AUTH_USER	KIE サーバーのバイパス認証ユーザー (org.kie.server.bypass.auth.user システムプロパティを設定します)	\${KIE_SERVER_BYPASS_AUTH_USER}
	KIE_SERVER_CONTROLLER_USER	KIE サーバーの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します)	\${KIE_SERVER_CONTROLLER_USER}
	KIE_SERVER_CONTROLLER_PWD	KIE サーバーの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します)	\${KIE_SERVER_CONTROLLER_PWD}
	KIE_SERVER_CONTROLLER_TOKEN	Bearer 認証用の KIE サーバーコントローラーのトークン (org.kie.server.controller.token システムプロパティを設定します)	\${KIE_SERVER_CONTROLLER_TOKEN}

デプロイメント	変数名	説明	値の例
	KIE_SERVER_CONTROLLER_SERVICE	オプションのスタンドアロンコントローラーのサービス名。アプリケーションは、このサービス名を使用してコントローラーに登録します。(設定されている場合は、ホストとポートの検出に使用されます)	`\${KIE_SERVER_CONTROLLER_SERVICE}`
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_CONTROLLER_HOST	KIE サーバーコントローラーのホスト (org.kie.server.controller システムプロパティの設定に使用)	`\${KIE_SERVER_CONTROLLER_HOST}`
	KIE_SERVER_CONTROLLER_PORT	KIE サーバーコントローラーのポート (org.kie.server.controller システムプロパティの設定に使用)	`\${KIE_SERVER_CONTROLLER_PORT}`
	KIE_SERVER_ID	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_ROUTE_NAME	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_USE_SECURE_ROUTE_NAME	True の場合、secure-APPLICATION_NAME-kieserver vs APPLICATION_NAME-kieserver をルート名として使用します。	`\${KIE_SERVER_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します)	`\${KIE_SERVER_USER}`

デプロイメント	変数名	説明	値の例
	KIE_SERVER_PWD	KIE サーバーへの接続に使用する KIE サーバーパスワード。生成される値は s2i に使用する提案として使用できます (org.kie.server.pwd システムプロパティを設定します)	`\${KIE_SERVER_PWD}`
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE サーバーコンテナのデプロイメント設定。形式: containerId=groupId:artifactId:version	c2=g2:a2:v2
	`\${KIE_SERVER_CONTAINER_DEPLOYMENT}`	MAVEN_REPOS	—
RHDMCENTR,EXTERNAL	RHDMCENTR_MAVEN_REPO_SERVICE		必要な場合には、サービスルックアップができるようにオプションの Decision Central (maven リポジトリの使用向け) に到達可能な場合に使用するサービス名
	`\${DECISION_CENTRAL_MAVEN_SERVICE}`	RHDMCENTR_MAVEN_REPO_PATH	—
/maven2/	RHDMCENTR_MAVEN_REPO_USERNAME		EAP 内の Decision 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
	`\${DECISION_CENTRAL_MAVEN_USERNAME}`	RHDMCENTR_MAVEN_REPO_PASSWORD	EAP 内の Decision 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
	`\${DECISION_CENTRAL_MAVEN_PASSWORD}`	EXTERNAL_MAVEN_REPO_ID	maven リポジトリに使用する id (設定されている場合)。デフォルトは無作為に作成されません。

デプロイメント	変数名	説明	値の例
	<code>\${MAVEN_REPO_ID}</code>	EXTERNAL_MAVEN_REPO_URL	Maven リポジトリへの完全修飾 URL。
	<code>\${MAVEN_REPO_URL}</code>	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)
	<code>\${MAVEN_REPO_USERNAME}</code>	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)。
	<code>\${MAVEN_REPO_PASSWORD}</code>	HTTPS_KEYSTORE_DIR	–
	<code>/etc/kieserver-secret-volume</code>	HTTPS_KEYSTORE	シークレット内のキーストアファイルの名前。
	<code>\${KIE_SERVER_HTTPS_KEYSTORE}</code>	HTTPS_NAME	サーバー証明書に関連付けられている名前
	<code>\${KIE_SERVER_HTTPS_NAME}</code>	HTTPS_PASSWORD	キーストアおよび証明書のパスワード
	<code>\${KIE_SERVER_HTTPS_PASSWORD}</code>	KIE_SERVER_MGMT_DISABLED	管理 api を無効にして、KIE コントローラーがデプロイ/デプロイ解除または起動/停止できないようにします。 org.kie.server.mgmt.api.disabled プロパティを true に、 org.kie.server.startup.strategy プロパティを LocalContainersStartupStrategy に設定します。
	<code>\${KIE_SERVER_MGMT_DISABLED}</code>	KIE_SERVER_STARTUP_STRATEGY	LocalContainersStartupStrategy に設定した場合には、コントローラーが設定されていて利用できないときでも KIE サーバーが起動し、ローカルの設定で機能できます。
	<code>\${KIE_SERVER_STARTUP_STRATEGY}</code>	JGROUPS_PING_PROTOCOL	–

デプロイメント	変数名	説明	値の例
	openshift.DNS_PING	OPENSIFT_DNS_PING_SERVICE_NAME	–
	\${APPLICATION_NAME}-kieserver-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 の RH-SSO クライアントシークレット
	\${KIE_SERVER_SSO_SECRET}	SSO_CLIENT	KIE Server の RH-SSO クライアント名。
	\${KIE_SERVER_SSO_CLIENT}	SSO_USERNAME	クライアント作成に使用する RH-SSO レルムの管理者のユーザー名 (存在しない場合)
	\${SSO_USERNAME}	SSO_PASSWORD	クライアント作成に使用する RH-SSO レルムの管理者のパスワード。
	\${SSO_PASSWORD}	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO が無効な SSL 証明書の検証。
	\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}	SSO_PRINCIPAL_ATTRIBUTE	ユーザー名として使用する RH-SSO プリンシパル属性
	\${SSO_PRINCIPAL_ATTRIBUTE}	HOSTNAME_HTTP	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>)。

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

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

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

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

デプロイメント	変数名	説明	値の例
	<code>\${AUTH_LDAP_REFERENCE_USER_ATTRIBUTE_ID_TO_CHECK}</code>	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	このパラメーターがある場合には、RoleMappingのログインモジュールで、指定したファイルを使用するように設定します。このプロパティは、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 <code>original_role=role1,role2,role3</code> になります。
	<code>\${AUTH_ROLE_MAPPER_ROLES_REPLACE}</code>	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。

4.1.2.4.3.7. ボリューム

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

4.1.2.5. 外部の依存関係

4.1.2.5.1. シークレット

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

kieserver-app-secret

4.2. 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 Decision Manager テンプレートのデプロイ、モニタリング、管理、デプロイ解除するには、OpenShift Web コンソールまたは **oc** コマンドを使用できます。

Web コンソールの使用に関する説明は、[Web コンソールを使用したイメージの作成およびビルド](#) を参照してください。

oc コマンドの使用方法に関する詳細は、[CLI リファレンス](#) を参照してください。次のコマンドが必要になる可能性があります。

- プロジェクトを作成するには、以下のコマンドを使用します。

```
$ oc new-project <project-name>
```

詳細は、[CLI を使用したプロジェクトの作成](#) を参照してください。

- テンプレートをデプロイするには (またはテンプレートからアプリケーションを作成するには)、以下のコマンドを実行します。

```
$ oc new-app -f <template-name> -p <parameter>=<value> -p <parameter>=<value> ...
```

詳細は、[CLI を使用したアプリケーションの作成](#) を参照してください。

- プロジェクト内のアクティブな Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods
```

- Pod のデプロイメントが完了し、実行中の状態になっているかどうかなど、Pod の現在のステータスを表示するには、以下のコマンドを使用します。

```
$ oc describe pod <pod-name>
```

oc describe コマンドを使用して、他のオブジェクトの現在のステータスを表示できます。詳細は、[アプリケーションの変更操作](#) を参照してください。

- Pod のログを表示するには、以下のコマンドを使用します。

```
$ oc logs <pod-name>
```

- デプロイメントログを表示するには、テンプレートリファレンスで **DeploymentConfig** の名前を検索して、以下のコマンドを実行します。

```
$ 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、テンプレートリファレンスで **BuildConfig** の名前を検索して、以下のコマンドを実行します。

```
$ oc logs -f bc/<build-config-name>
```

詳細は、[ビルドログのアクセス](#) を参照してください。

- アプリケーションの Pod をスケーリングするには、テンプレートリファレンスで **DeploymentConfig** の名前を検索して、以下のコマンドを実行します。

```
$ oc scale dc/<deployment-config-name> --replicas=<number>
```

詳細は、[手動スケーリング](#) を参照してください。

- アプリケーションのデプロイメントを解除するには、以下のコマンドを使用してプロジェクトを削除します。

```
$ oc delete project <project-name>
```

または、**oc delete** コマンドを使用して、Pod またはレプリケーションコントローラーなど、アプリケーションの一部を削除できます。詳細は、[アプリケーションの修正操作](#) を参照してください。

付録A バージョン情報

本書の最終更新日: 2021年11月15日(月)