Red Hat Process Automation Manager 7.6

Operator を使用した Red Hat OpenShift Container Platform への Red Hat Process Automation Manager 環境のデプロイメント

ガイド

Last Updated: 2021-07-22

Red Hat Process Automation Manager 7.6 Operator を使用した Red Hat OpenShift Container Platform への Red Hat Process Automation Manager 環境のデプロイメント

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

概要

本書では、Red Hat OpenShift Container Platform に Operator を使用して Red Hat Process Automation Manager 7.6 環境をデプロイする方法を説明します。

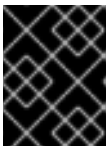
目次

はじめに

システムエンジニアは、Red Hat OpenShift Container Platform に Red Hat Process Automation Manager 環境をデプロイしてプロセスや他のビジネスアセットを開発または実行するインフラストラクチャーを提供できます。OpenShift Operators を使用して、構造化された YAML ファイルに定義された環境をデプロイして、必要に応じてこの環境を維持して変更できます。

前提条件

- Red Hat OpenShift Container Platform 環境が利用可能であること。Operator は Red Hat OpenShift Container Platform バージョン 4.1 および 4.2 でサポートされます。
- OpenShift 環境で 4 ギガバイト以上のメモリーが利用できる。
- デプロイメントする OpenShift プロジェクトが作成されている。
- OpenShift Web コンソールを使用してプロジェクトにログインしている。
- 動的永続ボリューム (PV) のプロビジョニングが有効化されている。または、動的 PV プロビジョニングが有効でない場合は、十分な永続ボリュームが利用できる状態でなければなりません。デフォルトでは、デプロイされるコンポーネントには以下の PV サイズが必要です。
  - それぞれのデプロイされた Process Server Pod のセットには、デフォルトでデータベースに 1 つの 1Gi PV が必要になります。データベース PV のサイズは変更することができます。複数のイミュータブルサーバーをデプロイでき、それぞれには別個のデータベース PV が必要になります。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。
  - デフォルトでは、Business Central は 1 Gi 分の PV が必要です。テンプレートパラメーターで、Business Central 永続ストレージの PV サイズを変更できます。
  - 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。
- Business Central または Business Central Monitoring Pod のいずれかをスケーリングする予定がある場合には、OpenShift 環境では、**ReadWriteMany** モードで永続ボリュームがサポートされます。ご使用の環境がこのモードに対応していない場合、NFS を使用してボリュームをプロビジョニングできます。



重要

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

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

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

この場合に、Red Hat Process Automation Manager のコンポーネントは、別の OpenShift Pod としてデプロイされます。各 Pod のスケールアップおよびスケールダウンを個別に行い、特定のコンポーネントに必要な数だけコンテナーを提供できます。標準の OpenShift の手法を使用して Pod を管理し、負荷を分散できます。

以下の Red Hat Process Automation Manager の主要コンポーネントが OpenShift で利用できます。

- Process Server (**実行サーバー (Execution Server)** または **KIE Server** とも呼ばれる) 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 をスケーリングできません。

  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サービスを使用し、同じサービスを実行します。OpenShift は負荷分散を提供しているため、要求はどの Pod でも処理できます。

  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、管理および監視コンソールを提供します。Business Central を使用してサービスを開発して Process Server にそれらのサービスをデプロイできます。Business Central を使用してサービスを開発し、それらを 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が、高可用性用に設定できます。複数の Pod を実行し、同じデータを共有する高可用性用に設定できます。

  Business Central には開発するサービスのソースを保管する Git リポジトリーが含まれます。また、ビルトインの Maven リポジトリーも含まれます。設定に応じて、Business Central はコンパイルしたサービス (KJAR ファイル) をビルドイン Maven リポジトリーに配置できます (設定した場合は外部 Maven リポジトリーにも可能)。

- Business Central Monitoring は Web ベースの管理および監視コンソールです。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。

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

OpenShift 環境に Red Hat Process Automation Manager をデプロイする前に、タスクをいくつか完了する必要があります。追加イメージ (たとえば、プロセスの新しいバージョン、または別のプロセス) をデプロイする場合は、このタスクを繰り返す必要はありません。

2.1. RED HAT レジストリーに対してお使いの環境が認証されていることを確認する方法

Red Hat OpenShift Container Platform の Red Hat Process Automation Manager コンポーネントをデプロイするには、OpenShift が Red Hat レジストリーから適切なイメージをダウンロードできることを確認します。

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。この設定は namespace ごとに固有であり、Operator が機能している場合には、**openshift** namespace に対する設定がすでに完了しています。

ただし、Red Hat Process Automation Manager のイメージストリームが **openshift** namespace にない場合や、Red Hat Process Automation Manager を新規バージョンに自動更新するように設定されている場合には、Operator はこのプロジェクトの namespace にイメージをダウンロードする必要があります。対象の namespace の認証設定を完了する必要があります。

手順

1. **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。
2. 「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。
3. **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。
4. 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。
5. 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。

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。

HTTPS アクセスを提供するために、Process Server では SSL 証明書を使用します。Business Central に SSL 証明書を作成し、OpenShift 環境にシークレットとして提供します。ただし、実稼働環境では、

Process Server に SSL 証明書を作成し、これをシークレットとして OpenShift 環境に提供する必要があります。

Procedure

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

HTTPS アクセスを提供するために、Business Central では SSL 証明書を使用します。Business Central に SSL 証明書を作成し、OpenShift 環境にシークレットとして提供します。ただし、実稼働環境では、Business Central の SSL 証明書を作成し、これをシークレットとして OpenShift 環境に提供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。

Procedure

1. Business Central の SSL 暗号化の秘密鍵および公開鍵を使用して、SSL キーストアを生成します。自己署名または購入した SSL 証明書でキーストアを作成する方法は、「SSL 暗号化キーおよび証明書」を参照してください。

注記

実稼働環境で、Business Central の予想される URL と一致する有効な署名済み証明書を生成します。

2. キーストアを **keystore.jks** ファイルに保存します。
3. 証明書の名前をメモします。Red Hat Process Automation Manager 設定におけるこのデフォルト名は **jboss** です。

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

5. **oc** コマンドを使用して、新しいキーストアファイルからシークレット **businesscentral-app-secret** を生成します。

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

2.4. AMQ ブローカー接続のシークレットの作成

Process Server を AMQ ブローカーに接続し、AMQ ブローカー接続に SSL を使用する場合は、接続の SSL 証明書を作成し、これを OpenShift 環境にシークレットとして指定する必要があります。

Procedure

1. AMQ ブローカー接続の 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、AMQ ブローカー接続の予想される URL に一致する有効な署名済みの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 **broker-app-secret** という名前のシークレットを生成します。

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

2.5. SMART ROUTER のシークレットの作成

HTTPS アクセスを提供するために、Smart Router では SSL 証明書を使用します。Business Central に SSL 証明書を作成し、OpenShift 環境にシークレットとして提供します。ただし、実稼働環境では、Smart Router の SSL 証明書を作成し、これをシークレットとして OpenShift 環境に提供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。

Procedure

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
```

2.6. NFS を使用した READWRITEMANY アクセスモードの永続ボリュームのプロビジョニング

非高可用性オーサリング環境のデフォルト設定である H2 データベースを使用する Business Central Monitoring、高可用性 Business Central または Process Server をデプロイする場合、環境は **ReadWriteMany** アクセスモードで永続ボリュームをプロビジョニングする必要があります。

お使いの設定で **ReadWriteMany** アクセスモードの永続ボリュームのプロビジョニングが必要であるものの、環境がそのようなプロビジョニングに対応しない場合、NFS を使用してボリュームをプロビジョニングします。それ以外の場合、この手順は省略します。

手順

NFS サーバーをデプロイし、NFS を使用して永続ボリュームをプロビジョニングします。NFS を使用して永続ボリュームをプロビジョニングする方法については、『OpenShift Container Platform 4.2 ストレージ』の「NFS を使用した永続ストレージ」のセクションを参照してください。

2.7. オフラインで使用する MAVEN ミラーリポジトリーの用意

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

注記

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

前提条件

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

Procedure

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. 公共のインターネットに送信アクセスができるコンピューターで、以下の手順を実行します。

   a. 最新バージョンの Offliner tool 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-7.6.0-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-7.6.0-offliner.txt
   ```

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

   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。アーティファクトをアップロードするには、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。アーティファクトをアップロードするには、Maven Repository Provisioner ユーティリティーを使用できます。

第3章 OPENSHIFT OPERATOR を使用した RED HAT PROCESS AUTOMATION MANAGER 環境のデプロイと管理

Red Hat Process Automation Manager 環境をデプロイするために、OpenShift Operator は環境を記述する YAML ソースを使用します。Red Hat Process Automation Manager は、YAML ソースを作成し、環境をデプロイするために使用できるインストーラーを提供します。

Business Automation Operator で環境をデプロイする場合には、環境の YAML 記述を作成し、環境が常にこの記述と一致していることを確認します。記述を編集して環境を変更することができます。

3.1. BUSINESS AUTOMATION OPERATOR のサブスクライブ

Operator を使用して Red Hat Process Automation Manager をデプロイできるようにするには、OpenShift で Business Automation Operator にサブスクライブする必要があります。

手順

1. OpenShift Web クラスターコンソールでプロジェクトに移動します。
2. OpenShift Web コンソールのナビゲーションパネルで、**Catalog → OperatorHub** または **Operators → OperatorHub** を選択します。
3. **Business Automation** を検索し、これを選択してから **Install** をクリックします。
4. **Create Operator Subscription** ページで、ターゲットの名前空間および承認ストラテジーを選択します。
   必要に応じて、**承認ストラテジー** を **Automatic** に設定して、Operator の自動更新を有効にします。Operator の更新は直ちに製品を更新しませんが、製品を更新する前に必要になります。特定のすべての製品デプロイメントの設定を使用して、自動または手動の製品更新を設定します。
5. **Subscribe** をクリックしてサブスクリプションを作成します。

3.2. OPERATOR を使用した RED HAT PROCESS AUTOMATION MANAGER 環境のデプロイ

Business Automation Operator にサブスクライブした後に、インストーラーウィザードを使用して Red Hat Process Automation Manager 環境を設定し、デプロイできます。

重要

Red Hat Process Automation Manager 7.6 では、Operator インストーラーウィザード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についての詳細は、「テクノロジプレビュー機能のサポート範囲」を参照してください。

3.2.1. Business Automation Operator の使用による Red Hat Process Automation Manager 環境のデプロイメントの開始

Business Automation Operator を使用して Red Hat Process Automation Manager 環境のデプロイメントを開始するには、インストーラーウィザードにアクセスします。インストーラーウィザードは Operator にサブスクライブする際にデプロイされます。

前提条件

- Business Automation Operator にサブスクライブしている。Operator にサブスクライブする方法については、「Business Automation Operator のサブスクライブ」 を参照してください。

Procedure

1. Red Hat OpenShift Container Platform Web クラスターコンソールメニューで、**Catalog → Installed operators** または **Operators → Installed operators** を選択します。
2. **businessautomation** が含まれる Operator の名前をクリックします。この Operator の情報が表示されます。
3. Red Hat OpenShift Container Platform バージョン 4.1 の場合は、ウィンドウの左側の **Installer** リンクを、Red Hat OpenShift Container Platform バージョン 4.2 以降の場合は、ウィンドウの右側のリンクをクリックします。
4. プロンプトが出されたら、OpenShift 認証情報でログインします。

結果

ウィザードの **Installation** タブが表示されます。

3.2.2. 環境の基本設定の構成

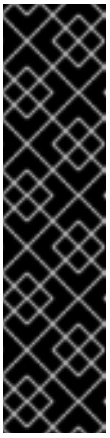
Business Automation Operator を使用して Red Hat Process Automation Manager 環境のデプロイを開始した後に、環境のタイプを選択し、他の基本的な設定を行う必要があります。

前提条件

- 「Business Automation Operator の使用による Red Hat Process Automation Manager 環境のデプロイメントの開始」 の説明に従って、Business Automation Operator を使用して Red Hat Process Automation Manager 環境のデプロイを開始し、インストーラーウィザードにアクセスしている。

Procedure

1. **Application Name** フィールドに、OpenShift アプリケーションの名前を入力します。この名前は、すべてのコンポーネントのデフォルト URL で使用されます。
2. **Environment** 一覧で、環境のタイプを選択します。このタイプは、デフォルトの設定を定めるものです。この設定を必要に応じて変更することができます。以下のタイプは Red Hat Process Automation Manager で利用できます。
   - **rhpam-trial**: すばやく設定でき、アセットの開発や実行の評価やデモに使用できる試用版の環境。Business Central と Process Server 1 台が含まれています。この環境では永続ストレージを使用しないので、この環境で実行した作業内容は保存されません。
   - **rhpam-authoring**: Business Central を使用してサービスを作成および修正する環境。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。
   - **rhpam-authoring-ha**: Business Central を使用してサービスを作成し、変更する 環境。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。このバージョンのオーサリング環境は、高可用性が確保されるように Business Central Pod のスケーリングをサポートします。



重要

Red Hat Process Automation Manager 7.6 では、Operator を使用した高可用性 Business Central 機能のデプロイメントはテクノロジープレビューとしてのみご利用いただけます。Red Hat のテクノロジープレビュー機能についての詳細は、「テクノロジプレビュー機能のサポート範囲」を参照してください。完全にサポートされた高可用性可用性デプロイメントの場合、Red Hat OpenShift Container Platform バージョン 3.11 で高可用性オーサリングテンプレートを使用します。このテンプレートをデプロイする手順については、『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。

- **rhpam-production**: ステージングおよび実稼働用として既存のサービスを実行するために使用する環境。この環境には、Business Central Monitoring、Smart Router、および Process Server Pod の 2 つのグループが含まれます。このようなすべてのグループに対してサービスのデプロイおよびデプロイ解除を実行できます。このようなすべてのグループに対してサービスのデプロイおよびデプロイ解除を実行します。必要に応じてこれらのグループのスケールアップおよびスケールダウンを実行できます。Business Central Monitoring を使用してサービスをデプロイし、実行し、停止し、またそれらの実行を監視します。

- **rhpam-production-immutable**: 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別の環境。ソースからサービスをビルドしたり、Maven リポジトリーからサービスをプルする 1 つ以上の Process Server Pod を設定できます。その後、必要に応じて各 Pod を複製できます。
  Pod からサービスを削除したり、新しいサービスを Pod に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するか、他の方法で設定を変更する場合は、新規のサーバーイメージをデプロイして、以前のイメージを置き換えます。コンテナーベースの統合ワークフローを使用して、Pod を管理できます。

  この環境を設定する場合は、KIE Servers タブで Process Server **をカスタマイズし、Set immutable server configuration ボタンをクリックするか、KIE_SERVER_CONTAINER_DEPLOYMENT** 環境変数を設定します。Process Server の設定手順は、「環境のカスタム Process Server 設定の構成」 を参照してください。

  必要に応じて、**Console** タブを使用して、この環境に Business Central Monitoring を追加して、プロセスサービスの実行を監視、停止、および再起動することもできます。Business Central Monitoring の設定手順は、「環境の Business Central 設定の構成」 を参照してください。

3. 新しいバージョンへの自動アップグレードを有効にするには、**Enable Upgrades** ボックスを選択します。このボックスを選択すると、Red Hat Process Automation Manager 7.6 の新しいパッチバージョンが利用可能になると、Operator は自動的にこのバージョンにデプロイメントをアップグレードします。サービスはすべて確保され、アップグレードプロセス全体で通常通り利用できます。
   同じ自動アップグレードプロセスを、Red Hat Process Automation Manager 7.x の新規マイナーバージョンが利用できる場合にも有効にする場合は、**Include minor version upgrade** のチェックボックスを選択します。

4. **Custom registry** のカスタムイメージレジストリーを使用する場合、**Image registry** フィールドにレジストリーの URL を入力します。このレジストリーに適切に署名され、認識された SSL 証明書がない場合には、**Insecure** ボックスを選択します。

5. **Admin user** の **Username** および **Password** フィールドに、Red Hat Process Automation Manager の管理者ユーザーのユーザー名およびパスワードを入力します。RH-SSO または

LDAP 認証を使用する場合、同じユーザーを、Red Hat Process Automation Manager の **kie-server,rest-all,admin** ロールで認証システムに設定する必要があります。

6. イメージのカスタムバージョンタグを使用する必要がある場合には、以下の手順を実行します。
   a. **Next** をクリックして **Security** タブにアクセスします。
   b. ウィンドウの下部までスクロールします。
   c. イメージタグを **Image tag** フィールドに入力します。

次のステップ

デフォルト設定で環境をデプロイする必要がある場合は、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合は、引き続き他の設定パラメーターの設定を行います。

3.2.3. 環境のセキュリティー設定の構成

Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本的な設定を行った後に、環境の認証 (セキュリティー) 設定をオプションで実行することができます。

前提条件

- 「環境の基本設定の構成」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本設定を行っていること。
- 認証に RH-SSO または LDAP を使用する必要がある場合には、認証システムに適切なロールを持つユーザーを作成していること。少なくとも以下のユーザーを作成する必要があります。
  - **kie-server,rest-all,admin** ロールを持つ管理者ユーザー (例: **adminUser**)
  - **kie-server,rest-all,guest** ロールを持つ **controllerUser** という名前のユーザー
  - **kie-server,rest-all,guest** ロールを持つ **executionUser** という名前のユーザー
- RH-SSO 認証を使用する必要がある場合は、環境のすべてのコンポーネントの RH-SSO システムでクライアントを作成しており、正しい URL を指定している。この動作により、最大限の制御が確保されます。他の方法として、デプロイメントでクライアントを作成できます。

手順

1. **Installation** タブが開いている場合は、**Next** をクリックして **Security** タブを表示します。
2. **Authentication mode** 一覧で、以下のモードのいずれかを選択します。
   - **Internal**: 環境のデプロイ時に初期ユーザーを設定します。このユーザーは Business Central を使用して他のユーザーを随時セットアップできます。
   - **RH-SSO**: Red Hat Process Automation Manager は認証に Red Hat Single Sign-On を使用します。
   - **LDAP**: Red Hat Process Automation Manager は認証に LDAP を使用します。
3. 選択した **Authentication mode** に基づいてセキュリティー設定を完了します。

Internal を選択している場合、**KIE Server password** フィールドをオプションで設定できます。アプリケーションはこのパスワードを持つ **executionUser** ユーザー名を使用して、REST API 要求をこの環境の Process Server に送信します。

RH-SSO を選択してい場合は、RH-SSO 認証を設定します。

a. **RH-SSO URL** フィールドに、RH-SSO URL を入力します。

b. **Realm** フィールドに、RH-SSO レルム名を入力します。

c. 環境のコンポーネントに RH-SSO クライアントを作成していない場合は、**SSO admin user** フィールドおよび **SSO admin password** フィールドに、RH-SSO システムの管理者ユーザーの認証情報を入力します。

d. RH-SSO システムに適切な署名済みの SSL 証明書がない場合は、**Disable SSL cert validation** ボックスを選択します。

e. **Principal attribute** フィールドで、ユーザー名に使用される RH-SSO プリンシパル属性を変更する必要がある場合は、新規属性の名前を入力します。

f. **Controller password** フィールドで、 **controllerUser** ユーザーの RH-SSO に設定したパスワードを入力します。

g. **KIE Server password** フィールドで、 **executionUser** ユーザーの RH-SSO に設定したパスワードを入力します。

LDAP を選択した場合は、LDAP 認証を設定します。

a. **LDAP URL** フィールドに、LDAP URL を入力します。

b. Red Hat JBoss EAP の LdapExtended ログインモジュールの設定に対応する LDAP パラメーターを設定します。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。

c. **Controller password** フィールドで、 **controllerUser** ユーザーの LDAP に設定したパスワードを入力します。

d. **KIE Server password** フィールドで、 **executionUser** ユーザーの LDAP に設定したパスワードを入力します。

4. 他のパスワードを設定します (必要な場合)。

- **AMQ password** および **AMQ cluster password** は、JMS API を使用した ActiveMQ との対話に使用するパスワードです。
- **Maven password** は **mavenUser** のパスワードです。環境に Business Central が含まれている場合には、このユーザーを使用してビルトイン Maven リポジトリーにアクセスできます。
- **Keystore password** は、HTTPS 通信のシークレットで使用されるキーストアファイルのパスワードです。「Process Server にシークレットの作成」 または 「Business Central へのシークレットの作成」 の説明にしたがってシークレットを作成した場合には、このパスワードを設定します。
- **Database password** は、環境の一部であるデータベースサーバー Pod のパスワードです。

次のステップ

すべてのコンポーネントのデフォルト設定で環境をデプロイする必要がある場合には、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合には、引き続き Business Central、Process Server、および Smart Router の設定パラメーターを設定します。

3.2.4. 環境の Business Central 設定の構成

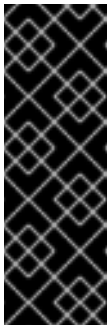
Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本的なセキュリティー設定を行った後に、環境の Business Central または Business Central Monitoring コンポーネントの設定をオプションで実行することができます。

前提条件

- 「環境の基本設定の構成」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本設定を行っていること。
- 認証に RH-SSO または LDAP を使用する必要がある場合は、「環境のセキュリティー設定の構成」 の説明に従ってセキュリティー設定を完了していること。

Procedure

1. **Installation** または **Security** タブが開いている場合は、**Console** タブが表示されるまで **Next** をクリックします。
2. 「Business Central へのシークレットの作成」 の説明に従って Business Central **のシークレットを作成している場合、Secret** フィールドにシークレットの名前を入力します。
3. オプションで、Business Central または Business Central monitoring のレプリカ数を **Replicas** フィールドに入力します。この数は **rhpam-authoring** 環境では変更しません。
4. オプションで、**Resource quotas** 下のフィールドに必要な CPU およびメモリーの上限値を入力します。
5. RH-SSO 認証を選択している場合は、Business Central の RH-SSO を設定します。
   a. **Client name** フィールドにクライアント名を入力し、**Client secret** フィールドにクライアントシークレットを入力します。この名前を持つクライアントが存在しない場合は、デプロイメントでこの名前およびシークレットを持つ新規クライアントの作成を試行します。
   b. デプロイメントで新規クライアントを作成する場合、Process Server へのアクセスに使用する HTTP および HTTPS URL を **SSO HTTP URL** および **SSO HTTPS URL** フィールドに入力します。この情報は、クライアントに記録されます。
6. オプションで、環境変数を随時設定します。環境変数を設定するには、**Add new Environment variable** をクリックしてから、変数の名前および値を **Name** フィールドおよび **Value** フィールドに入力します。
   - 外部 Maven リポジトリーを使用する必要がある場合は、以下の変数を設定します。
     - **MAVEN_REPO_URL**: Maven リポジトリーの URL
     - **MAVEN_REPO_ID**: Maven リポジトリーの ID (例: **repo-custom**)
     - **MAVEN_REPO_USERNAME**: Maven リポジトリーのユーザー名
     - **MAVEN_REPO_PASSWORD** Maven リポジトリーのパスワード



重要

オーサリング環境で、Business Central を使用して外部の Maven リポジトリーにプロジェクトをプッシュする場合には、デプロイメント時にこのリポジトリーを設定して、全プロジェクトのリポジトリーへのエクスポートを設定する必要があります。外部の Maven リポジトリーへの Business Central プロジェクトのエクスポートに関する情報は、『Red**Hat Process Automation Manager プロジェクトのパッケージ化およびデプロイ**』を参照してください。

- OpenShift 環境が公開インターネットに接続されていない場合は、「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 ミラーにアクセスできるように設定します。以下の変数を設定してください。
  - **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」でセットアップした 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**: 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。

    オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合、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**)。

次のステップ

Process Server および Smart Router のデフォルト設定で環境をデプロイする必要がある場合には、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合には、引き続き Process Server および Smart Router の設定パラメーターを設定します。

3.2.5. 環境のカスタム Process Server 設定の構成

Business Automation Operator のすべての環境タイプには、デフォルトで 1 つまたは複数の Process Server が含まれます。

オプションで、Process Server のカスタム設定を構成できます。この場合、デフォルトの Process Server は作成されず、設定する Process Server のみがデプロイされます。

前提条件

- 「環境の基本設定の構成」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本設定を行っていること。
- 認証に RH-SSO または LDAP を使用する必要がある場合は、「環境のセキュリティー設定の構成」 の説明に従ってセキュリティー設定を完了していること。

Procedure

1. **Installation**、**Security**、または **Console** タブが開いている場合は、**KIE Servers** タブが表示されるまで **Next** をクリックします。
2. **Add new KIE Server** をクリックして、新規の Process Server 設定を追加します。
3. **Id** フィールドに、Process Server の ID を入力します。Process Server が Business Central または Business Central Monitoring インスタンスに接続される場合、この ID はサーバーが加わるサーバーグループを決めるものとなります。
4. **Name** フィールドで、Process Server の名前を入力します。
5. **Deployments** フィールドで、デプロイする同様の Process Server の数を入力します。インストーラーは、同じ設定で複数の Process Server をデプロイできます。Process Server のID および名前は自動的に変更され、一意な状態に保たれます。
6. 「Process Server にシークレットの作成」 の説明に従って Process Server **のシークレットを作成している場合、Keystore secret** フィールドにシークレットの名前を入力します。
7. オプションで、Process Server のレプリカ数を **Replicas** フィールドに入力します。
8. Process Server のカスタムイメージを使用する必要がある場合には、以下の追加の手順を実行します。
    a. **Set KIE Server image** をクリックします。
    b. OpenShift イメージストリームタグではなく Docker イメージ名を使用する必要がある場合は、**Kind** の値を **DockerImage** に変更します。
    c. イメージストリームの名前を **Name** フィールドに入力します。
    d. イメージストリームが **openshift** 名前空間にない場合は、名前空間を **Namespace** フィールドに入力します。
9. Source to Image(S2I)ビルドを使用してイミュータブル Process Server を設定する必要がある場合は、以下の追加の手順を実行します。

重要

Maven リポジトリーからサービスをプルするイミュータブル Process Server **を設定する必要がある場合は、Set Immutable server configuration をクリックせず**、この手順は実行しないでください。代わりに、**KIE_SERVER_CONTAINER_REPLOYMENT** 環境変数を設定します。

    a. **Set Immutable server configuration** をクリックします。
    b. **KIE Server コンテナーデプロイメント** フィールドに、デプロイメントが Source to Image (S2I) ビルドの結果から展開する必要があるサービスの識別情報 (KJAR ファイル) を入力します。形式は **<containerId>=<groupId>:<artifactId>:<version>** になります。また、コンテナーのエイリアス名で指定する場合には、形式は **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>** になります。以下の例に示されるように、区切り文字 | 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(alias2)=g2:a2:v2**)。
    c. 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、Maven mirror URL フィールドに設定した **Maven ミラーの URL** を入力します。

d. **Artifact directory** フィールドで、Maven が正常にビルドされた後に、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) が含まれるプロジェクト内のパスを入力します。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 リポジトリー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す。

e. S2I ビルドにカスタムベース Process Server イメージを使用する必要がある場合は、**Set Base build image** をクリックします。イメージストリームが **openshift** 名前空間にない場合は、名前空間を **Namespace** フィールドに入力します。OpenShift イメージストリームタグではなく Docker イメージ名を使用する必要がある場合は、**Kind** の値を **DockerImage** に変更します。

f. **Set Git source** をクリックし、以下のフィールドに情報を入力します。

- **S2I Git URI**: サービスのソースが含まれる Git リポジトリーの URI。
- **Reference**: Git リポジトリーのブランチ。
- **コンテキストディレクトリー**: (オプション) Git リポジトリーからダウンロードされたプロジェクト内のソースへのパス。デフォルトで、ダウンロードされたプロジェクトのルートディレクトリーはソースディレクトリーです。

g. Git Webhook を設定して Git リポジトリーの変更が Process Server **の自動リビルドをトリガーするように設定する必要がある場合には、Add new Webhook** をクリックします。**Type** 一覧から Webhook **のタイプを選択し、Secret** フィールドで Webhook のシークレット文字列を入力します。

10. オプションで、**Resource quotas** 下のフィールドに必要な CPU およびメモリーの上限値を入力します。複数の Process Server を設定している場合は、制限値はそれぞれのサーバーに別々に適用されます。

11. RH-SSO 認証を選択している場合は、Process Server の RH-SSO を設定します。

a. **Client name** フィールドにクライアント名を入力し、**Client secret** フィールドにクライアントシークレットを入力します。この名前を持つクライアントが存在しない場合は、デプロイメントでこの名前およびシークレットを持つ新規クライアントの作成を試行します。

b. デプロイメントで新規クライアントを作成する場合、Process Server へのアクセスに使用する HTTP および HTTPS URL を **SSO HTTP URL** および **SSO HTTPS URL** フィールドに入力します。この情報は、クライアントに記録されます。

12. 外部 AMQ メッセージブローカーを使用して JMS API から Process Server と対話する場合は、**Enable JMS Integration** 設定を有効にします。JMS 統合を設定するための追加のフィールドが表示され、必要に応じて値を入力する必要があります。

- **User name**、**Password**: ブローカーのユーザー認証が環境で必要な場合の、標準ブローカーユーザーのユーザー名およびパスワード。
- **Executor**：この設定を選択して JMS Executor を無効にします。Executor はデフォルトで有効になります。
- **Executor transacted**: この設定を選択して、Executor キューで JMS トランザクションを有効にします。
- **Enable signal**: この設定を選択して JMS 経由でシグナルの設定を有効にします。
- **Enable audit** この設定を選択して JMS 経由で監査ロギングを有効にします。

- **Audit transacted**: この設定を選択して、監査キューで JMS トランザクションを有効にします。
- **Queue executor**、**Queue request**、**Queue response**、**Queue signal**、**Queue audit** 使用するキューのカスタム JNDI 名。これらの値のいずれかを設定する場合は、**AMQ キュー** パラメーターも設定する必要があります。
- **AMQ Queues**: AMQ キュー名はコンマで区切られます。これらのキューはブローカーの起動時に自動的に作成され、JBoss EAP サーバーの JNDI リソースとしてアクセスできます。カスタムキュー名を使用する場合は、このフィールドでサーバーが使用するすべてのキューの名前を入力する必要があります。
- **Enable SSL integration**: AMQ ブローカーへの SSL 接続を使用する場合は、この設定を選択します。この場合、「AMQ ブローカー接続のシークレットの作成」 で作成したシークレットの名前や、シークレットに使用したキーストアおよび信頼ストアの名前およびパスワードも指定する必要があります。

13. Process Server が使用する必要のあるデータベースを選択します。以下の値を使用できます。
    - **mysql**: 個別の Pod に作成される MySQL サーバー。
    - **postgresql**: 個別の Pod に作成される PostgreSQL サーバー。他の設定を使用する特別な理由のない限り、この設定を使用します。
    - **h2**: 個別の Pod を必要としないビルトインされた **h2** データベースエンジン。この設定を使用する場合には、Process Server Pod をスケーリングしないでください。
    - **external**: 外部データベースサーバー。

注記

Red Hat Process Automation Manager 7.6 では、Business Automation Operator を使用して環境をデプロイする場合、MySQL および PostgreSQL 外部データベースサーバーのみがサポートされます。

14. オプションで、**Size** フィールドに、データベースサーバー用に作成する永続ボリュームのサイズを入力します。
15. 外部データベースサーバーを選択している場合は、追加のフィールドに以下の情報を入力します。
    a. **Driver**: サーバーの種類に応じてデータベースサーバードライバーを入力します。
        - **mysql**
        - **postgresql**
        - **mariadb**
        - **mssql**
        - **db2**
        - **oracle**
        - **sybase**

b. **Dialect**: サーバーの種類に応じて、サーバーの Hibernate ダイアレクトを入力します。

- **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**

c. **Host**: 外部データベースサーバーのホスト名を入力します。

d. **Port**: 外部データベースサーバーのポート番号を入力します。

e. **Jdbc URL**: 外部データベースサーバーの JDBC URL を入力します。

f. **NonXA**: データソースを XA 以外のモードで設定する必要がある場合にこのボックスを選択します。

g. **JNDI name**: アプリケーションがデータソースに使用する JNDI 名を入力します。

h. **User name** および **Password**: 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を入力します。

i. **Background validation**: オプションとして、このボックスを選択してバックグラウンド SQL 検証を有効にし、バックグラウンド検証の間隔を入力します。

j. オプションとして、最小および最大の接続プールサイズ、およびデータベースサーバーの例外ソータークラス (exception sorter class) を設定します。

16. オプションで、環境変数を随時設定します。環境変数を設定するには、**Add new Environment variable** をクリックしてから、変数の名前および値を **Name** および **Value** フィールドに入力します。

- 設定した Maven リポジトリーからサービスをプルするイミュータブル KIE Server を設定する必要がある場合は、以下の設定を入力します。
  i. **KIE_SERVER_CONTAINER_DEPLOYMENT** 環境変数を設定します。変数には、デプロイメントが Maven リポジトリーからプルする必要のあるサービス (KJAR ファイル) の ID 情報が含まれている必要があります。形式は **<containerId>=<groupId>:<artifactId>:<version>** になります。また、コンテナーのエイリアス名で指定する場合には、形式は **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>** になります。以下の例に示されるように、区切り文字 **|** 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(alias2)=g2:a2:v2**)。
  ii. 外部 Maven リポジトリーの設定
- 外部 Maven リポジトリーを設定する必要がある場合には、以下の変数を設定します。
  - **MAVEN_REPO_URL**: Maven リポジトリーの URL
  - **MAVEN_REPO_ID**: Maven リポジトリーの ID (例: **repo-custom**)
  - **MAVEN_REPO_USERNAME**: Maven リポジトリーのユーザー名

- **MAVEN_REPO_PASSWORD**: Maven リポジトリーのパスワード

- OpenShift 環境が公開インターネットに接続されていない場合は、「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 ミラーにアクセスできるように設定します。以下の変数を設定してください。

  - **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」でセットアップした 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。この Process Server を S2I として設定している場合は、この URL をすでに入力されています。

  - **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。この Process Server を S2I として設定している場合は、この値を設定しません。**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。

    オーサリング環境でビルトイン Business Central Maven リポジトリーを使用する場合、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**)。

- 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は、**PROMETHEUS_SERVER_EXT_DISABLED** 環境変数を **false** に設定します。Prometheus メトリクス収集の設定方法については、『Managing and monitoring Process Server』の「{URL_MANAGING_SETTINGS}#prometheus-monitoring-ocp-proc_execution-server[**Managing and monitoring Process Server**」を参照してください。

次のステップ

追加の Process Server を設定するには、**Add new KIE Server**を再びクリックし、新規サーバー設定の手順を繰り返します。

デフォルト設定の Smart Router を使って環境をデプロイする必要がある場合には、**Finish** をクリックしてから **Deploy** をクリックして環境をデプロイします。それ以外の場合には、引き続き Smart Router の設定パラメーターの設定を行います。

3.2.6. 環境の Smart Router 構成の設定

デフォルトでは、**rhpam-production** タイプの環境には Smart Router が含まれます。他の環境タイプにはデフォルトで Smart Router は含まれていません。

Smart Router が存在しない場合は、環境に追加できます。Smart Router の構成オプションを設定することもできます。

前提条件

- 「環境の基本設定の構成」 の説明に従って、インストーラーウィザードで Business Automation Operator を使用して Red Hat Process Automation Manager 環境の基本設定を行っていること。

Procedure

1. **Installation**、**Security**、**Console**、または **KIE Servers** タブが開いている場合、**Smart Router** タブが表示されるまで **Next** をクリックします。
2. Smart Router が存在しない場合、**Set Smart Router** をクリックして Smart Router を環境に追加し、Smart Router 構成を設定します。
3. 「Smart Router のシークレットの作成」 の説明に従って Smart Router **のシークレットを作成している場合、Secret** フィールドにシークレットの名前を入力します。
4. オプションで、Smart Router のレプリカ数を **Replicas** フィールドに入力します。
5. オプションで、**Resource quotas** 下のフィールドに必要な CPU およびメモリーの上限値を入力します。

次のステップ

Finish をクリックしてから **Deploy** をクリックし、環境をデプロイします。

3.3. OPERATOR を使用してデプロイした環境の変更

環境を Operator を使用してデプロイした場合は、通常の OpenShift の手法を使用して環境を変更することはできません。たとえば、デプロイメント設定またはサービスを削除しても、これは同じパラメーターで自動的に再作成されます。

環境を変更するには、環境の YAML の記述を変更する必要があります。パスワードなどの一般的な設定を変更し、新規 Process Server を追加し、Process Server をスケーリングできます。

Procedure

1. OpenShift Web クラスターコンソールでプロジェクトに移動します。
2. OpenShift Web コンソールナビゲーションパネルで **Catalog → Installed operators** または **Operators → Installed operators** を選択します。
3. 表で **Business Automation** Operator 行を見つけ、その行で **KieApp** をクリックします。この Operator を使用してデプロイした環境の情報が表示されます。
4. デプロイした環境の名前をクリックします。
5. **YAML** タブを選択します。YAML ソースが表示されます。
6. パスワードなどの共通の設定を変更するには、**commonConfig:** の値を編集します。
7. 新しい Process Server を追加する場合は、以下の例に示されているように、**servers:** のブロックの最後にそれらの記述を追加します。
   - 名前が **server-a** と **server-a-2** のサーバー 2 台を追加するには、以下の行を追加します。

     ```
     - deployments: 2
       name: server-a
     ```

   - S2I プロセスのソースからビルドされるサービスを含む、イミュータブルな Process Server を追加するには、以下の行を追加します。

     ```
     - build:
         kieServerContainerDeployment: <deployment>
     ```

```
gitSource:
  uri: <url>
  reference: <branch>
  contextDir: <directory>
```

以下の値を置き換えます。

- **<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。形式は **<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 リポジトリー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。
- **<url>**: デシジョンサービスのソースを含む Git リポジトリーの URL。
- **<branch>**: Git リポジトリーのブランチ。
- **<directory>**: Git リポジトリーからダウンロードしたプロジェクトのソースへのパス。

8. Process Server をスケーリングする場合は、**servers:** のブロックに含まれるサーバーの記述を検索して、その記述の下に **replicas:** 設定を追加します。たとえば、**replicas: 3** はサーバーを Pod 3 つにスケーリングします。
9. **Save** をクリックしてから **has been updated** ポップアップメッセージを待機します。
10. **Reload** をクリックして、環境の新しい YAML の記述を表示します。

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）