



# Red Hat Process Automation Manager 7.13

Red Hat Process Automation Manager のインス  
トールおよび設定





## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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

本書では、Red Hat JBoss EAP クラスター環境、JBoss Web サーバーおよびその他のサポート環境において、Red Hat JBoss EAP に Red Hat Process Automation Manager コンポーネントをインストールする方法を説明します。

## 目次

|  |    |
|--|----|
| 前書き .....  | 7  |
| 多様性を受け入れるオープンソースの強化 .....  | 8  |
| パート I. RED HAT PROCESS AUTOMATION インストールの計画 .....                                      | 9  |
| 第1章 RED HAT PROCESS AUTOMATION MANAGER のバージョン管理 .....                                  | 10 |
| 第2章 RED HAT PROCESS AUTOMATION MANAGER コンポーネント .....                                   | 12 |
| 第3章 RED HAT PROCESS AUTOMATION MANAGER インストールプラットフォーム .....                            | 13 |
| 3.1. RED HAT JBOSS EAP .....   | 13 |
| 3.2. RED HAT JBOSS WEB SERVER .....  | 13 |
| 3.3. RED HAT OPENSIFT CONTAINER PLATFORM .....   | 13 |
| 3.4. ORACLE WEBLOGIC SERVER .....  | 14 |
| 3.5. IBM WEBSHERE APPLICATION SERVER .....   | 14 |
| 第4章 RED HAT PROCESS AUTOMATION MANAGER でサポートされるリポジトリ .....                             | 15 |
| 4.1. GIT リポジトリ .....   | 15 |
| 4.2. APACHE MAVEN .....  | 15 |
| 第5章 RED HAT PROCESS AUTOMATION MANAGER のインストール環境オプション .....                            | 16 |
| 第6章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー .....                                | 17 |
| 第7章 RED HAT PROCESS AUTOMATION MANAGER 参照実装 .....                                      | 19 |
| 第8章 RED HAT PROCESS AUTOMATION MANAGER との統合 .....                                      | 20 |
| 8.1. RED HAT PROCESS AUTOMATION MANAGER SPRING BOOT ビジネスアプリケーション .....                 | 20 |
| 8.2. RED HAT FUSE の統合 .....  | 20 |
| 8.3. RED HAT SINGLE SIGN-ON の統合 .....  | 20 |
| 第9章 関連情報 .....   | 21 |
| パート II. RED HAT JBOSS EAP 7.4 への RED HAT PROCESS AUTOMATION MANAGER のインストールおよび設定 ..... | 22 |
| 第10章 RED HAT PROCESS AUTOMATION MANAGER とは .....                                       | 23 |
| 第11章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー .....                               | 24 |
| 第12章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード .....                        | 26 |
| 第13章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用 .....                               | 27 |
| 13.1. インタラクティブモードでのインストーラーの使用 .....  | 27 |
| 13.2. CLI モードでのインストーラーの使用 .....  | 30 |
| 13.3. インストーラーでの PROCESS AUTOMATION MANAGER コントローラーを使用する KIE SERVER の設定 .....           | 32 |
| 第14章 ZIP ファイルからの RED HAT PROCESS AUTOMATION MANAGER のインストール .....                      | 34 |
| 14.1. ZIP ファイルからの BUSINESS CENTRAL のインストール .....                                       | 34 |
| 14.2. ZIP ファイルからの KIE SERVER のインストール .....   | 35 |
| 14.3. KIE SERVER への JDBC データソースの設定 .....   | 36 |
| 14.4. ユーザーの作成 .....  | 38 |
| 14.5. BUSINESS CENTRAL に接続する KIE SERVER の設定 .....                                      | 40 |
| 14.6. スレッド効率 .....   | 43 |

|  |     |
|--|-----|
| 第15章 キーストアを使用したパスワードセキュリティーの確保 .....   | 44  |
| 第16章 SSH が RSA を使用するように設定する手順 .....  | 46  |
| 第17章 RED HAT PROCESS AUTOMATION MANAGER インストールの検証 .....                                    | 47  |
| 第18章 RED HAT PROCESS AUTOMATION MANAGER の実行 .....  | 48  |
| 第19章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行 .....                            | 50  |
| 19.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストール                                      | 50  |
| 19.2. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーの実行  | 53  |
| 第20章 スタンドアロン BUSINESS CENTRAL の設定および実行 .....   | 55  |
| 第21章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリ .....                           | 57  |
| 21.1. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加                                   | 57  |
| 21.2. BUSINESS CENTRAL および KIE SERVER への外部 MAVEN リポジトリの設定                                  | 61  |
| 21.3. オフラインで使用する MAVEN ミラーリポジトリの準備   | 62  |
| 第22章 GIT リポジトリからのプロジェクトのインポート .....  | 65  |
| 第23章 BUSINESS CENTRAL ブランディングのカスタマイズ .....   | 66  |
| 23.1. BUSINESS CENTRAL ログインページのカスタマイズ  | 66  |
| 23.2. BUSINESS CENTRAL アプリケーションヘッダーのカスタマイズ   | 66  |
| 第24章 LDAP と SSL の統合 .....  | 67  |
| 付録A BUSINESS CENTRAL システムプロパティー .....  | 68  |
| 第25章 関連情報 .....  | 74  |
| 第26章 次のステップ .....  | 75  |
| パート III. RED HAT JBOSS EAP クラスター環境への RED HAT PROCESS AUTOMATION MANAGER のインストールおよび設定 ..... | 76  |
| 第27章 RED HAT PROCESS AUTOMATION MANAGER クラスター .....  | 77  |
| 第28章 開発 (オーサリング) 環境における RED HAT PROCESS AUTOMATION MANAGER クラスター .....                     | 78  |
| 28.1. RED HAT DATA GRID のインストールおよび設定   | 78  |
| 28.2. AMQ BROKER のダウンロードおよび設定  | 80  |
| 28.3. NFS バージョン 4 サーバーの設定  | 80  |
| 28.4. RED HAT JBOSS EAP 7.4 および RED HAT PROCESS AUTOMATION MANAGER のダウンロードおよびデプロイメント       | 81  |
| 28.5. クラスターでの BUSINESS CENTRAL の設定および実行  | 82  |
| 28.6. 高可用性 (HA) オンプレミスインフラストラクチャーのテスト  | 84  |
| 28.7. RED HAT PROCESS AUTOMATION MANAGER クラスターの検証  | 86  |
| 第29章 ランタイム環境における KIE SERVER クラスター .....  | 88  |
| 29.1. RED HAT JBOSS EAP 7.4 および KIE SERVER のダウンロードおよびデプロイメント                               | 88  |
| 29.2. KIE SERVER 向けの RED HAT JBOSS EAP 7.4 クラスターの設定および実行                                   | 89  |
| 29.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーを使用した KIE SERVER のクラスタリング                    | 92  |
| 29.4. 失敗した KIE SERVER ノードの RUNNING 状態にあるジョブのフェイルオーバーの設定                                    | 94  |
| 第30章 SMART ROUTER のインストールおよび設定 .....   | 97  |
| 30.1. SMART ROUTER による KIE サーバーインスタンスのロードバランシング  | 97  |
| 30.2. TLS 対応の SMART ROUTER の設定   | 100 |
| 30.3. エンドポイント認証用の SMART ROUTER の設定   | 101 |

|  |            |
|--|------------|
| 30.4. SMART ROUTER 動作の設定   | 102        |
| 30.5. 設定ファイルを使用した SMART ROUTER の設定   | 103        |
| <b>第31章 QUARTZ タイマーサービスの設定</b>   | <b>106</b> |
| <b>第32章 関連情報</b>   | <b>108</b> |
| <b>パート IV. RED HAT JBOSS WEB SERVER での RED HAT PROCESS AUTOMATION MANAGER のインストールと設定</b> | <b>109</b> |
| <b>第33章 RED HAT PROCESS AUTOMATION MANAGER とは</b>  | <b>110</b> |
| 33.1. RED HAT PROCESS AUTOMATION MANAGER コンポーネント   | 111        |
| 33.2. RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー                                      | 111        |
| <b>第34章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード</b>                         | <b>113</b> |
| <b>第35章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用</b>                                | <b>114</b> |
| 35.1. インタラクティブモードでのインストーラーの使用  | 114        |
| 35.2. CLI モードでのインストーラーの使用  | 117        |
| <b>第36章 KIE SERVER ZIP ファイルのインストールと設定</b>  | <b>120</b> |
| 36.1. ZIP ファイルからの KIE SERVER のインストール   | 120        |
| 36.2. JDBC WEB SERVER データソースの設定  | 122        |
| <b>第37章 キースタアを使用したパスワードセキュリティの確保</b>   | <b>126</b> |
| <b>第38章 KIE SERVER インストールの確認</b>   | <b>129</b> |
| <b>第39章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのダウンロードおよびインストール</b>                     | <b>130</b> |
| <b>第40章 スタンドアロン BUSINESS CENTRAL の設定および実行</b>  | <b>133</b> |
| <b>第41章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリ</b>                        | <b>135</b> |
| 41.1. プロジェクト設定ファイル (POM.XML) を使用した MAVEN の設定   | 135        |
| 41.2. MAVEN 設定ファイルの修正  | 135        |
| 41.3. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加                                 | 136        |
| 41.4. オフラインで使用する MAVEN ミラーリポジトリの準備   | 139        |
| <b>第42章 GIT リポジトリからのプロジェクトのインポート</b>   | <b>142</b> |
| <b>第43章 LDAP と SSL の統合</b>   | <b>143</b> |
| <b>付録B BUSINESS CENTRAL システムプロパティー</b>   | <b>144</b> |
| <b>パート V. IBM WEBSHERE APPLICATION SERVER への KIE SERVER のインストールおよび設定</b>                 | <b>150</b> |
| <b>第44章 KIE SERVER</b>   | <b>151</b> |
| <b>第45章 IBM WEBSHERE APPLICATION SERVER</b>  | <b>152</b> |
| <b>第46章 IBM WEBSHERE APPLICATION SERVER のインストールおよび実行</b>                                 | <b>153</b> |
| <b>第47章 KIE SERVER への IBM WEBSHERE APPLICATION SERVER の設定</b>                            | <b>155</b> |
| 47.1. 管理セキュリティの有効化   | 155        |
| 47.2. IBM WEBSHERE APPLICATION SERVER への JDBC データソースの設定                                  | 156        |
| 47.3. JAVA MESSAGE SERVICE (JMS) の設定   | 158        |
| 47.4. IBM WEBSHERE APPLICATION SERVER へのシステムプロパティーの設定                                    | 161        |
| 47.5. IBM WEBSHERE APPLICATION SERVER の停止および再起動  | 164        |

|  |     |
|--|-----|
| <b>第48章 IBM WEBSPPHERE APPLICATION SERVER への KIE SERVER のインストールおよび実行</b> .....                             | 166 |
| 48.1. KIE SERVER のグループおよびロールの作成  | 167 |
| 48.2. KIE SERVER のグループおよびロールのマッピング   | 167 |
| 48.3. KIE SERVER のクラスローディングの設定   | 168 |
| 48.4. インストールの検証  | 168 |
| <b>第49章 IBM WEBSPPHERE APPLICATION SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行</b> ..... | 170 |
| 49.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーへのクラスローディングの設定   | 171 |
| 49.2. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティーの設定   | 171 |
| 49.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーグループおよびロールのマッピング   | 173 |
| 49.4. インストールの検証  | 173 |
| <b>第50章 IBM WEBSPPHERE APPLICATION SERVER での埋め込みのプロセスエンジンとデシジョンエンジンの設定</b>                                 | 175 |
| <b>第51章 キーストアを使用したパスワードセキュリティの確保</b> .....   | 180 |
| <b>第52章 次のステップ</b> .....   | 182 |
| <b>パート VI. ORACLE WEBLOGIC SERVER への KIE SERVER のインストールおよび設定</b> .....                                     | 183 |
| <b>第53章 KIE SERVER</b> .....   | 184 |
| <b>第54章 ORACLE WEBLOGIC SERVER</b> .....   | 185 |
| <b>第55章 ORACLE WEBLOGIC SERVER のインストールおよび実行</b> .....  | 186 |
| <b>第56章 KIE SERVER への ORACLE WEBLOGIC SERVER の設定</b> .....   | 188 |
| 56.1. KIE SERVER のグループおよびユーザーの設定   | 188 |
| 56.2. ORACLE WEBLOGIC SERVER への JDBC データソースの設定   | 188 |
| 56.3. JAVA MESSAGE SERVICE (JMS) の設定   | 191 |
| 56.4. ORACLE WEBLOGIC SERVER へのシステムプロパティーの設定   | 194 |
| 56.5. ORACLE WEBLOGIC SERVER の停止および再起動   | 195 |
| <b>第57章 ORACLE WEBLOGIC SERVER への KIE SERVER のインストール</b> .....   | 197 |
| 57.1. ORACLE WEBLOGIC SERVER への KIE SERVER インストールの確認   | 197 |
| <b>第58章 ORACLE WEBLOGIC SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行</b> .....            | 199 |
| 58.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティーの設定   | 200 |
| 58.2. インストールの検証  | 201 |
| <b>第59章 ORACLE WEBLOGIC SERVER での埋め込みプロセスエンジンまたはデシジョンエンジンの設定</b> ..  | 203 |
| <b>第60章 キーストアを使用したパスワードセキュリティの確保</b> .....   | 208 |
| <b>第61章 次のステップ</b> .....   | 210 |
| <b>パート VII. RED HAT PROCESS AUTOMATION MANAGER の修正およびアップグレード</b> .....                                     | 211 |
| <b>第62章 RED HAT PROCESS AUTOMATION MANAGER のバージョン管理</b> .....  | 212 |
| <b>第63章 RED HAT カスタマーポータルでの通知の有効化</b> .....  | 214 |
| <b>第64章 RED HAT PROCESS AUTOMATION MANAGER へのパッチ更新およびマイナーリリースアップグレードの適用</b> .....                          | 215 |
| <b>第65章 関連情報</b> .....   | 219 |



---

|                   |     |
|-------------------|-----|
| 付録C バージョン情報 ..... | 220 |
| 付録D お問い合わせ先 ..... | 221 |



## 前書き

開発者またはシステム管理者は、Red Hat JBoss EAP クラスター環境、JBoss Web サーバーおよびその他のサポート環境において、Red Hat JBoss EAP に Red Hat Process Automation Manager コンポーネントをインストールできます。

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みにより、これらの変更は今後の複数のリリースに対して段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

## パート I. RED HAT PROCESS AUTOMATION インストールの計画

システム管理者が選択できる Red Hat Process Automation のインストールオプションは複数あります。

# 第1章 RED HAT PROCESS AUTOMATION MANAGER のバージョン管理

Red Hat Process Automation Manager のバージョンには、メジャーリリース、マイナーリリース、およびパッチリリースのそれぞれの数字を組み合わせた **Major.Minor.Patch** という書式が使用されています (例: 7.13.5)。メジャーリリースが **7.x.x**、マイナーリリースが **7.13.x**、パッチリリースが 7.13.5 となります。メジャーリリースでは、大抵の場合はデータ移行が求められますが、マイナーリリースの更新やパッチの更新は、通常 Red Hat Process Automation Manager リリースアーティファクトで提供される更新ツールで管理されます。

以下は、Red Hat Process Automation Manager の一般的なリリースの種類になります。

## メジャーリリースの移行

Red Hat Process Automation Manager のメジャーリリースには、非常に多くの機能強化、セキュリティ更新、バグ修正が含まれており、再設計された機能も含まれている可能性があります。Red Hat JBoss BPM Suite 6.4.x から Red Hat Process Automation Manager 7.0 など、メジャーリリースから別のメジャーリリースにアプリケーションを移行すると、通常データ移行が必要です。移行を容易にするために、Red Hat Process Automation Manager の新しいメジャーバージョンで、自動化した移行ツールがしばしば提供されますが、特定のデータおよび設定の手動処理が必要となる可能性が高くなります。サポートされる移行パスは、製品のお知らせおよびドキュメントに指定されています。

たとえば、移行の方法については、[Red Hat JBoss BPM Suite 6.4 から Red Hat Process Automation Manager 7.0 への移行](#) を参照してください。

## マイナーリリースのアップグレード

Red Hat Process Automation Manager のマイナーリリースには、機能拡張、セキュリティ更新、バグ修正が含まれます。Red Hat Process Automation Manager 7.5.x から 7.6 など、アプリケーションを別のマイナーリリースに移行する場合には、データの移行が必要な場合があります。大抵の場合は、Business Central、KIE Server、ヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の特定コンポーネントの更新を容易にする自動更新ツールが Red Hat Process Automation Manager のパッチ更新と新規マイナーバージョンで提供されます。デシジョンエンジンやスタンドアロンの Business Central など、その他の Red Hat Process Automation Manager アーティファクトは、各マイナーリリースが含まれる新しいアーティファクトとしてリリースされるため、再インストールして更新を適用する必要があります。

新たなマイナーリリースにアップグレードする前に、お使いの Red Hat Process Automation Manager に最新のパッチ更新を適用し、マイナーリリースのアップグレードが成功するようにしてください。

## パッチの更新

Red Hat Process Automation Manager のパッチの更新には、最新のセキュリティ更新、およびバグ修正が含まれます。予定されているパッチ更新には、製品のマイナーバージョンに対してすでにリリースされているパッチ更新がすべて含まれるため、更新を最新の状態にするために、パッチの更新を1つ1つ適用していく必要はありません。たとえば、Red Hat Process Automation Manager 7.5.0 または 7.5.1 は Red Hat Process Automation Manager 7.5.2 に更新できます。Red Hat Process Automation Manager で最適なパフォーマンスを得るためには、製品の更新が利用可能になるとすぐに適用する必要があります。また、Red Hat では既存製品の通常の更新サイクルとは別に、予定されていないパッチ更新がリリースされる場合があります。これには、特定の問題を修正するために Red Hat グローバルサポートサービス (GSS) によって提供されるセキュリティまたはその他の更新が含まれる場合があります。累積的な更新ではない場合があります。

大抵の場合は、Business Central、KIE Server、ヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の特定コンポーネントの更新を容易にする自動更新ツールが Red Hat Process Automation Manager のパッチ更新と新規マイナーバージョンで提

供されます。デシジョンエンジンやスタンドアロンの Business Central など、その他の Red Hat Process Automation Manager アーティファクトは、各マイナーリリースが含まれる新しいアーティファクトとしてリリースされるため、再インストールして更新を適用する必要があります。

リリース間での移行を最適に保ち、Red Hat Process Automation Manager ディストリビューションに最新の機能強化および修正が含まれるようにするには、製品リリースや更新が Red Hat カスタマーポータルで利用可能になった時点で、Red Hat Process Automation Manager に適用します。Red Hat カスタマーポータルで製品の通知を有効にすることも検討してください。

## 第2章 RED HAT PROCESS AUTOMATION MANAGER コンポーネント

この製品は、Business Central と KIE Server で構成されています。

- Business Central は、ビジネスルールを作成して管理するグラフィカルユーザーインターフェイスです。Business Central は、Red Hat JBoss EAP インスタンスまたは Red Hat OpenShift Container Platform (OpenShift) にインストールできます。  
Business Central は、スタンドアロンの JAR ファイルとしても使用できます。Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。
- KIE Server では、ルール、およびその他のアーティファクトが実行されます。これは、ルールをインスタンス化して実行し、計画の問題を解決するために使用されます。KIE Server は、Red Hat JBoss EAP インスタンス、Red Hat JBoss EAP クラスター、OpenShift、Oracle WebLogic Server インスタンス、IBM WebSphere Application Server インスタンスに、または Spring Boot アプリケーションの一部としてインストールできます。  
KIE Server は、管理モードまたは非管理モードで動作するように設定できます。KIE Server が非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE コンテナは、プロジェクトの特定のバージョンです。KIE Server が管理されている場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Process Automation Manager コントローラーは Business Central と統合します。Red Hat JBoss EAP に Business Central をインストールする場合は、Execution Server ページを使用して KIE コンテナを作成および維持します。ただし、Business Central をインストールしない場合は、ヘッドレス Process Automation Manager コントローラーをインストールし、REST API または KIE Server Java Client API を使用してそのコントローラーと対話します。

- Red Hat ビルドの OptaPlanner は、Business Central および KIE Server に統合されています。また、組み込み可能な軽量プランニングエンジンで、プランニングの問題を最適化します。Red Hat ビルドの OptaPlanner は、最適化のためのヒューリスティック法およびメタヒューリスティック法を効率的なスコア計算と組み合わせ、Java プログラマーがプランニングの問題を効率的に解決できるようにします。



## 第3章 RED HAT PROCESS AUTOMATION MANAGER インストールプラットフォーム

Red Hat Process Automation Manager の全リリースは、オペレーティングシステム、JVM、Web ブラウザー、データベースのさまざまな組み合わせに対して認定されています。Red Hat は、サポートされる設定およびテスト済みの設定に対して、実稼働サポートと開発サポートをお客様のサブスクリプション契約に従って提供します。サポート対象の設定とバージョン番号の詳細は、以下のページを参照してください。

- [Red Hat JBoss Data Virtualization 7 でサポートされる設定](#)
- [Red Hat Process Automation Manager 7 コンポーネントの詳細](#)

Red Hat Process Automation Manager 7.13 は、以下のアプリケーションプラットフォームで利用できます。

- Red Hat JBoss EAP
- Red Hat JBoss Web Server
- Red Hat OpenShift Container Platform
- Oracle WebLogic Server
- IBM WebSphere Application Server

### 3.1. RED HAT JBOSS EAP

Red Hat JBoss Enterprise Application Platform (Red Hat JBoss EAP) は、Java Enterprise Edition 7 (Java EE 7) の Full Profile 仕様および Web Profile 仕様の認定実装です。Red Hat JBoss EAP には、高可用性クラスタリング、メッセージング、分散キャッシングなどの機能に対する事前設定オプションが用意されています。ユーザーは、Red Hat JBoss EAP が提供するさまざまな API およびサービスを使用して、アプリケーションを開発、デプロイ、および実行することもできます。

Business Central と KIE Server の両方を単一の Red Hat JBoss EAP インスタンスにインストールできます。ただし、実稼働環境では、別のインスタンスにインストールする必要があります。

### 3.2. RED HAT JBOSS WEB SERVER

Red Hat JBoss Web Server は Tomcat をベースとしたエンタープライズレベルの Web サーバーで、中規模および大規模のアプリケーション用に設計されています。Red Hat JBoss Web Server は、Java Server Pages (JSP) および Java Servlet テクノロジー、PHP、ならびに CGI をデプロイするための単一プラットフォームを提供します。

KIE Server とヘッドレス Process Automation Manager コントローラーは、Red Hat JBoss Web Server にインストールすることができます。

### 3.3. RED HAT OPENSIFT CONTAINER PLATFORM

Red Hat OpenShift Container Platform (OpenShift) は、Kubernetes を使用し、API を提供してこれらのサービスを管理します。OpenShift を使用すると、コンテナを作成および管理できます。

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。

OpenShift には、Business Central と KIE Server の両方をインストールできます。

### 3.4. ORACLE WEBLOGIC SERVER

Oracle WebLogic Server は、分散型 Java アプリケーションを作成するために API の標準セットを提供する Java EE アプリケーションサーバーで、データベース、メッセージングサービス、外部のエンタープライズシステムへの接続など、さまざまなサービスにアクセスできます。ユーザーは、Web ブラウザクライアントまたは Java クライアントを使用してこれらのアプリケーションにアクセスします。

Oracle WebLogic Server インスタンスに KIE Server をインストールし、ヘッドレス Process Automation Manager コントローラーと、REST API または KIE Server Java Client API を使用して KIE Server と対話します。また、スタンドアロンの Business Central を使用して KIE Server と対話することもできます。

### 3.5. IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server は、Java ベースの Web アプリケーションをホストし、Java EE 認定ランタイム環境を提供する、柔軟性がある安全な Web アプリケーションです。IBM WebSphere 9.0 は Java SE 8 に対応しており、バージョン 8.5.5.6 以降の Java EE 7 に完全に準拠しています。

IBM WebSphere Application Server インスタンスに KIE Server をインストールし、ヘッドレス Process Automation Manager コントローラーを使用して、KIE Server と対話します。また、スタンドアロンの Business Central を使用して KIE Server と対話することもできます。

## 第4章 RED HAT PROCESS AUTOMATION MANAGER でサポートされるリポジトリ

Git リポジトリと Apache Maven を Red Hat Process Automation Manager で使用できます。

### 4.1. GIT リポジトリ

Git リポジトリは Business Central 内で使用され、オーサリング環境で作成したプロセス、ルール、その他のアーティファクトをすべて保存します。Git は分散バージョン管理システムです。リビジョンをコミットオブジェクトとして実装します。変更をリポジトリにコミットすると、Git リポジトリに新規コミットオブジェクトが作成されます。Business Central でプロジェクトを作成すると、Business Central に接続される Git リポジトリに追加されます。

他の Git リポジトリにプロジェクトがある場合は、それらを Business Central のスペースにインポートできます。Git フックを使用して、内部 Git リポジトリを外部リポジトリに同期できます。

### 4.2. APACHE MAVEN

[Apache Maven](#) は分散型構築自動化ツールで、ソフトウェアプロジェクトのビルドおよび管理を行うために Java アプリケーション開発で使用されます。Maven を使用することで、ご自分の Red Hat Process Automation Manager プロジェクトをビルド、公開、およびデプロイできます。Maven には以下のメリットがあります。

- ビルドプロセスが容易で、すべてのプロジェクトに対して統一された構築システムが実装される。
- プロジェクトに必要なすべての JAR ファイルがコンパイル時に利用可能になる。
- 適切なプロジェクト構造が設定される。
- 依存関係およびバージョンが適切に管理される。
- Maven では事前定義されたさまざまな出力タイプ (JAR および WAR 等) にビルドされるので、追加のビルドプロセスが不要である。

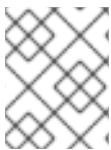
Maven はレポジトリを使用して Java ライブラリー、プラグイン、および他のビルドアーティファクトを格納します。これらのリポジトリは、ローカルまたはリモートいずれかの形態をとることができます。Red Hat Process Automation Manager によりローカルおよびリモート maven リポジトリが維持され、それをご自分のプロジェクトに追加してルール、プロセス、イベント、およびその他のプロジェクト依存関係にアクセスできます。プロジェクトおよびアーキタイプをビルドする際に、Maven はローカルまたはリモートリポジトリから Java ライブラリーおよび Maven プラグインを動的に取得します。これにより、プロジェクト全体を通じて依存関係の共有および再利用が促進されます。

## 第5章 RED HAT PROCESS AUTOMATION MANAGER のインストール環境オプション

Red Hat Process Automation Manager を使用して、ビジネスアプリケーションを開発する開発環境、デシジョンをサポートするアプリケーションを実行するランタイム環境、またはその両方を設定できます。

- **開発環境:** 通常、1つの Business Central インストールと1つ以上の KIE Server インストールで構成されます。開発者は Business Central を使用して、デシジョンやその他のアーティファクトを作成できるほか、KIE Server を使用して、作成したアーティファクトをテストできます。
- **ランタイム環境:** Business Central の有無にかかわらず、1つ以上の KIE Server インスタンスで構成されます。Business Central には Process Automation Manager コントローラーが組み込まれています。Business Central をインストールしている場合は、**Menu → Deploy → Execution Server** ページを使用してコンテナを作成および維持します。Business Central なしで KIE Server の管理を自動化する場合は、ヘッドレス Process Automation Manager コントローラーを使用できます。

また、開発環境およびランタイム環境の両方をクラスター化することもできます。クラスター化した開発環境またはランタイム環境は、統合グループまたは2台以上のサーバーのクラスターで構成されます。Red Hat Process Automation Manager の開発環境をクラスター化すると、高可用性やコラボレーションの強化などの利点があります。また、Red Hat Process Automation Manager のランタイム環境をクラスター化すると、高可用性および負荷分散の利点が得られます。高可用性を実現することで、1台のサーバーで障害が発生したときにデータが損失する可能性を減らすことができます。サーバーでの障害発生時に、別のサーバーにより、障害が発生したサーバーにあるデータのコピーを提供して、障害に対応します。障害が発生したサーバーは、再度オンラインになると、クラスターに復帰します。



### 注記

ランタイム環境のクラスター化は、現在 Red Hat JBoss EAP 7.4 および Red Hat OpenShift Container Platform でのみサポートされています。

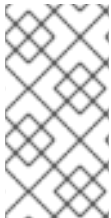
## 第6章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー

Business Central または KIE 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 または KIE Server のインストール時に、ユーザーとロールを作成できます。

Business Central と KIE Server の両方が単一のインスタンスで動作している場合、Business Central で認証されたユーザーは KIE Server にもアクセスできます。

ただし、Business Central と KIE Server が別のインスタンスで動作している場合、Business Central で認証されたユーザーが KIE Server にアクセスするには、別途認証が必要です。たとえば、Business Central で認証されているものの、KIE Server で認証されていないユーザーが Business Central でプロセス定義を表示または管理しようとする、401 エラーがログファイルに記録され、**Invalid credentials to load data from remote server.Contact your system administrator.** メッセージが Business Central に表示されます。

このセクションでは、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 KIE Server 用に予約されています。このため、Business Central または KIE Server のいずれか、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、ビジネスインジケター、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、KIE Server REST 機能にアクセスできます。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7章 RED HAT PROCESS AUTOMATION MANAGER 参照実装

Red Hat Process Automation Manager は、スターターアプリケーションとして使用できる参照実装を提供します。参照実装は、**Red Hat Process Automation Manager 7.13.5 参照実装**で提供されます。これは、Red Hat カスタマーポータルの Software Downloads ページにある Red Hat Process Automation Manager からダウンロードできます。

### 従業員勤務表の参照実装

従業員勤務表の参照実装により、組織内のさまざまなポジションのシフトに従業員を割り当てるアプリケーションの作成が可能となります。たとえば、アプリケーションを使用して、病院での看護師のシフト、さまざまな場所での警備勤務シフト、作業者の組み立てラインのシフトを割り当てます。

### 運搬経路プランニングの参照実装

運搬経路プランニングの参照実装により、実際の地図、道路、および目的地へ向かう物資輸送車(車両ごとに積載量を指定)に関する運搬経路プランニングの問題を解決するアプリケーションを作成できます。詳細は、参照実装のダウンロードに含まれる運搬経路 ZIP ファイルの README ファイルを参照してください。

### 授業の時間割参照実装

授業の時間割参照実装により、授業を教室や時間枠と関連付けて、学生および教師の制約を考慮して競合を回避する REST アプリケーションを Spring Boot に構築できます。

### 高可用性イベント駆動型デシジョン機能の参照実装

高可用性イベント駆動型デシジョン機能の参照実装により、複雑なイベント処理で開発したルールを含む、ステートフル処理を必要とする Drools エンジンコードを OpenShift 環境にデプロイできます。これにより、デシジョンエンジンは、高可用性の複雑なイベントシリーズを処理できるようになります。

## 第8章 RED HAT PROCESS AUTOMATION MANAGER との統合

Red Hat Process Automation Manager は、Spring Boot、Red Hat Fuse、Red Hat Single Sign-On などの他の製品やコンポーネントと統合できます。

### 8.1. RED HAT PROCESS AUTOMATION MANAGER SPRING BOOT ビジネスアプリケーション

Spring Framework は、Java アプリケーション開発用に包括的なインフラストラクチャーサポートを提供する Java プラットフォームです。Spring Boot は、Spring Boot スターターをベースにした、軽量フレームワークです。Spring Boot スターターは、**pom.xml** ファイルで、このファイルには Spring Boot プロジェクトに追加可能な依存関係の記述子セットが含まれます。Red Hat は、Red Hat Runtime の一部として、Red Hat Process Automation Manager Spring Boot プロジェクトのサポートおよびメンテナンスを提供します。

Red Hat Process Automation Manager Spring Boot ビジネスアプリケーションは、特定のビジネス機能を提供する個別サービスを柔軟に、UI に依存せず、論理的にグループ化します。ビジネスアプリケーションは、Spring Boot スターターをベースにしています。通常、個別にデプロイされ、個別にバージョン管理できます。完全なビジネスアプリケーションを使用すると、ドメインは特定のビジネスゴール (たとえば、注文管理や宿泊管理など) を達成できます。ビジネスアプリケーションを作成して設定した後には、OpenShift で、既存のサービスやクラウドにデプロイできます。

### 8.2. RED HAT FUSE の統合

Fuse は、アジャイル統合ソリューションの一部である、分散型のクラウドネイティブ統合プラットフォームです。チームはこの分散アプローチを使用することで、必要に応じて統合サービスをデプロイできます。

Apache Karaf コンテナプラットフォームまたは Red Hat JBoss Enterprise Application Platform に Red Hat Fuse をインストールしてから、このコンテナに Red Hat Process Automation Manager をインストールし、設定してください。

### 8.3. RED HAT SINGLE SIGN-ON の統合

Red Hat シングルサインオン (RH-SSO) は、ブラウザーアプリケーションと REST Web サービス、および Git へのアクセスのセキュリティーを確保するために使用できるシングルサインオンソリューションです。

Red Hat Process Automation Manager と RH-SSO を統合する際に、Red Hat Process Automation Manager 向けに SSO と IDM (アイデンティティ管理) を作成します。RH-SSO のセッション管理機能により、一度認証するだけで、Web 上でさまざまな Red Hat Process Automation Manager 環境を使用できます。

Red Hat Process Automation Manager では、Red Hat JBoss EAP 7.4 上で RH-SSO 7.5 がサポートされています。



## 第9章 関連情報

### Installation

- [Red Hat JBoss EAP 7.4 への Red Hat Process Automation Manager のインストールおよび設定](#)
- [Red Hat JBoss EAP クラスター環境への Red Hat Process Automation Manager のインストールおよび設定](#)
- [Red Hat JBoss Web Server での Red Hat Process Automation Manager のインストールと設定](#)
- [IBM WebSphere Application Server への KIE Server のインストールおよび設定](#)
- [Oracle WebLogic Server への KIE Server のインストールおよび設定](#)

### 統合

- [Spring Boot での Red Hat Process Automation Manager ビジネスアプリケーションの作成](#)
- [Red Hat Process Automation Manager と Red Hat Fuse の統合](#)
- [Red Hat Process Automation Manager と Red Hat シングルサインオンの統合](#)

### Red Hat ビルドの OptaPlanner

- [Red Hat Process Automation Manager でのソルバーの開発](#)

### OpenShift

- [Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#)
- [テンプレートを使用した Red Hat OpenShift Container Platform 3 への Red Hat Process Automation Manager 環境のデプロイメント](#)

## パート II. RED HAT JBOSS EAP 7.4 への RED HAT PROCESS AUTOMATION MANAGER のインストールおよび設定

本書は、Red Hat JBoss Enterprise Application Platform 7.4 インスタンスに Red Hat Process Automation Manager をインストールする方法を説明します。

### 前提条件

- [Red Hat Process Automation Manager インストールの計画](#)の内容を確認済みである。
- Red Hat JBoss Enterprise Application Platform 7.4 の最新の修正リリースがインストールされている。Red Hat JBoss EAP のインストールに関する情報は [Red Hat JBoss Enterprise Application Platform インストールガイド](#)を参照。

## 第10章 RED HAT PROCESS AUTOMATION MANAGER とは

Red Hat Process Automation Manager は、ビジネスの自動化アプリケーションとマイクロサービスを作成する Red Hat ミドルウェアプラットフォームです。これにより、企業のビジネスユーザーと IT ユーザーが、ビジネスプロセスおよびポリシーを文書化、シミュレート、管理、自動化、およびモニターできます。ビジネスユーザーおよび IT ユーザーがより効果的に協力できるように設計されているため、ビジネスアプリケーションは簡単にすばやく変更できます。

この製品は、Business Central と KIE Server で構成されています。

- KIE Server では、ルール、およびその他のアーティファクトが実行されます。これは、ルールをインスタンス化して実行し、計画の問題を解決するために使用されます。KIE Server はビジネスアセットのランタイム環境を提供し、アセットリポジトリ (ナレッジストア) に保存されたデータにアクセスします。
- Business Central は、KIE Server が実行するビジネスルールを作成して管理するグラフィカルユーザーインターフェイスです。これにより、以下のタスクを実行できます。
  - ルール、プロセス、および関連アセットを作成、管理、および編集します。
  - 接続された KIE Server インスタンスおよびその KIE コンテナを管理します (デプロイメントユニット)。
  - Business Central に接続した KIE Server インスタンスのプロセスおよびタスクに対してランタイム操作を実行します。  
Business Central は、スタンドアロンの JAR ファイルとしても使用できます。Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。

Red Hat JBoss Enterprise Application Platform (Red Hat JBoss EAP) 7.4 は、Java Enterprise Edition 8 (Java EE 8) の Full Profile 仕様および Web Profile 仕様の認定実装です。Red Hat JBoss EAP には、高可用性クラスタリング、メッセージング、分散キャッシングなどの機能に対する事前設定オプションが用意されています。ユーザーは、Red Hat JBoss EAP が提供するさまざまな API およびサービスを使用して、アプリケーションを開発、デプロイ、および実行することもできます。

本書では、Red Hat JBoss EAP 7.4 サーバーインスタンスに Red Hat Process Automation Manager をインストールする方法を説明します。

他の環境への Red Hat Process Automation Manager のインストール方法に関する説明は、以下のドキュメントを参照してください。

- [IBM WebSphere Application Server への KIE Server のインストールおよび設定](#)
- [Oracle WebLogic Server への KIE Server のインストールおよび設定](#)
- [Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#)
- [テンプレートを使用した Red Hat OpenShift Container Platform 3 への Red Hat Process Automation Manager 環境のデプロイメント](#)

サポートされるコンポーネントについては、以下のドキュメントを参照してください。

- [What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#)
- [Red Hat JBoss Data Virtualization 7 でサポートされる設定](#)

## 第11章 RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー

Business Central または KIE 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 または KIE Server のインストール時に、ユーザーとロールを作成できます。

Business Central と KIE Server の両方が単一のインスタンスで動作している場合、Business Central で認証されたユーザーは KIE Server にもアクセスできます。

ただし、Business Central と KIE Server が別のインスタンスで動作している場合、Business Central で認証されたユーザーが KIE Server にアクセスするには、別途認証が必要です。たとえば、Business Central で認証されているものの、KIE Server で認証されていないユーザーが Business Central でプロセスマニフェストを表示または管理しようとすると、401 エラーがログファイルに記録され、**Invalid credentials to load data from remote server.Contact your system administrator.** メッセージが Business Central に表示されます。

このセクションでは、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 KIE Server 用に予約されています。このため、Business Central または KIE Server のいずれか、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、ビジネスインジケター、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、KIE Server REST 機能にアクセスできます。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12章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード

インストーラー JAR ファイルまたはデプロイ可能な ZIP ファイルを使用して Red Hat Process Automation Manager をインストールすることができます。インストーラーは、インタラクティブモードまたはコマンドラインインターフェイス (CLI) モードで実行できます。もしくは、Business Central および KIE Server のデプロイ可能な ZIP ファイルをデプロイメントして設定できます。Business Central をアプリケーションサーバーにデプロイせずに実行する場合は、Business Central スタンドアロン JAR ファイルをダウンロードします。

お使いの環境およびインストール要件に応じた Red Hat Process Automation Manager ディストリビューションをダウンロードします。

### 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. お好みのインストール方法に従って、以下の製品ディストリビューションのいずれかをダウンロードします。



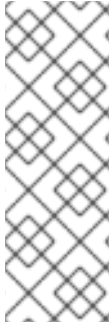
### 注記

ダウンロードするのは、どれか1つのディストリビューションのみです。

- インストーラーを使用して Red Hat JBoss EAP 7.4 に Red Hat Process Automation Manager をインストールする場合は、**Red Hat Process Automation Manager 7.13.5 Installer (rhpam-installer-7.13.5.jar)** をダウンロードします。インストーラーのグラフィックユーザーインターフェイスに従って、インストールプロセスを進めます。
- デプロイ可能な zip ファイルを使用して Red Hat JBoss EAP 7.4 に Red Hat Process Automation Manager をインストールするには、以下のファイルをダウンロードします。
  - **Red Hat Process Automation Manager 7.13.5 KIE Server for All Supported EE8 Containers (rhpam-7.13.5-kie-server-ee8.zip)**
  - **Red Hat Process Automation Manager 7.13.5 Business Central Deployable for EAP 7 (rhpam-7.13.5-business-central-eap7-deployable.zip)**
  - **Red Hat Process Automation Manager 7.13.5 Add Ons (rhpam-7.13.5-add-ons.zip)**
- アプリケーションサーバーにデプロイせずに Business Central を実行するには、**Red Hat Process Automation Manager 7.13.5 Business Central Standalone (rhpam-7.13.5-business-central-standalone.jar)** をダウンロードします。

## 第13章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用

本セクションでは、インストーラー JAR ファイルを使用して Business Central および KIE Server をインストールする方法を説明します。JAR ファイルは、既存の Red Hat JBoss EAP 7.4 サーバーインストールに Red Hat Process Automation Manager をインストールする実行ファイルです。インストーラーは、インタラクティブモードまたはコマンドラインインターフェイス (CLI) モードで実行できます。



### 注記

Red Hat Process Automation Manager JAR ファイルインストーラーは、yum または RPM Package Manager を使用してインストールした Red Hat JBoss EAP ディストリビューションを **サポートしません**。このようにインストールされた Red Hat JBoss EAP に Red Hat Process Automation Manager をインストールするには、**Red Hat Process Automation Manager 7.13 Deployable for Red Hat JBoss EAP 7.4** ファイルをダウンロードし、[14章ZIP ファイルからの Red Hat Process Automation Manager のインストール](#) の手順に従います。



### 注記

IBM JDK では、その他の JDK で生成したキーストアを使用できないため、別の JDK で生成したキーストアを使用して、IBM JDK で実行している既存の Red Hat JBoss EAP に Red Hat Process Automation Manager をインストールすることはできません。

### 次のステップ:

以下のセクションのいずれかに記載される手順を行います。

- [「インタラクティブモードでのインストーラーの使用」](#)
- [「CLI モードでのインストーラーの使用」](#)
- [「インストーラーでの Process Automation Manager コントローラーを使用する KIE Server の設定」](#)

### 13.1. インタラクティブモードでのインストーラーの使用

Red Hat Process Automation Manager のインストーラーは、実行可能な JAR ファイルです。このインストーラーを使用して、既存の Red Hat JBoss EAP 7.4 サーバーインストールに Red Hat Process Automation Manager をインストールできます。



### 注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

### 前提条件

- **Red Hat Process Automation Manager 7.13.5 Installer**をダウンロードしている。手順は、[12章Red Hat Process Automation Manager インストールファイルのダウンロード](#)を参照してください。

- サポート対象の JDK がインストールされている。サポート対象の JDK のリストについては、[Red Hat Process Automation Manager 7 Supported Configurations](#) を参照してください。
- バックアップを作成してある Red Hat JBoss EAP 7.4 のサーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。
- `$PATH` 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、`java-$JAVA_VERSION-openjdk-devel` パッケージに含まれています。



### 注記

Red Hat Process Automation Manager は、UTF-8 エンコーディングで機能するように設計されています。基礎となる JVM で別のエンコーディングシステムを使用すると、予期せぬエラーが発生する可能性があります。JVM で UTF-8 を使用するには、`-Dfile.encoding=UTF-8` のシステムプロパティを使用します。システムプロパティのリストは、[付録A Business Central システムプロパティ](#) を参照してください。

### 手順

1. 端末ウィンドウで、インストーラー JAR ファイルをダウンロードしたディレクトリーに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar
```



### 注記

Windows でインストーラーを実行すると、インストール時に管理者の認証情報が求められる場合があります。この要求を回避するには、インストールコマンドに `izpack.mode=privileged` オプションを追加します。

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.13.5.jar
```

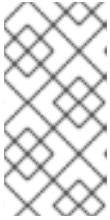
また、32 ビットの Java 仮想マシンでインストーラーを実行している場合には、メモリー不足になる可能性があります。この問題を防ぐには、以下のコマンドを実行します。

```
java -XX:MaxHeapSize=4g -jar  
rhpam-installer-7.13.5.jar
```

グラフィカルインストーラーにスプラッシュ画面と使用許諾契約書のページが表示されます。

2. **I accept the terms of this license agreement (本使用許諾契約書の内容に同意します)** をクリックし、**Next** をクリックします。
3. Red Hat Process Automation Manager をインストールする Red Hat JBoss EAP 7.4 サーバーのホームを指定して、**Next** をクリックします。
4. インストールするコンポーネントを選択し、**Next** をクリックします。

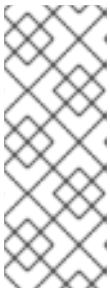




## 注記

Business Central と KIE Server は同じサーバーにインストールできます。ただし、実稼働環境では、Business Central と KIE Server は異なるサーバーにインストールすることが推奨されます。そのためには、インストーラーを2回実行します。

5. ユーザーを作成して **Next** をクリックします。デフォルトでは、同じコンテナに Business Central と KIE Server の両方をインストールする場合は、新しいユーザーに **admin** ロール、**kie-server** ロール、および **rest-all** ロールが割り当てられます。KIE Server のみをインストールした場合には、ユーザーには **kie-server** ロールが割り当てられます。別のロールを選択する場合は、**admin** ロールを外します。ロールの詳細は、[11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。



## 注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。たとえば、**admin** という名前のユーザーは作成しないでください。

パスワードは8文字以上で、数字と、英数字以外の文字をそれぞれ1文字以上使用する必要があります。ただし&の文字は使用できません。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

6. **Installation Overview** ページで、インストールするコンポーネントを確認し、**Next** をクリックしてインストールを開始します。
7. インストールが完了したら、**Next** をクリックします。
8. KIE Server がインストールされている場合には、**Component Installation** で **Configure Runtime** の手順が表示されます。**Configure Runtime Environment** ページで、デフォルトのインストールを実行するか、詳細設定を行うかを選択します。**Perform advanced configuration** を選択した場合は、データベース設定、または特定の KIE Server オプションのカスタマイズが選択できます。
9. **JDBC Drive Configuration** ページで **Customize database settings** を選択した場合は、データソースの JDBC ドライバーのベンダーを選択し、ドライバーの JAR ファイルを1つ以上選択し、**Next** をクリックします。  
データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。
10. **KIE Server Properties Configuration** で **Customize KIE Server properties** を選択した場合は、以下のいずれかのプロパティを変更します。
  - **KIE Server ID**の値を変更して、KIE Server プロパティの名前を変更します。
  - **Controller URL**の値を変更して、ヘッドレス Process Automation Manager コントローラーの URL を変更します。
  - 無効にする KIE Server 機能の選択を解除します。

11. **Next** をクリックして、ランタイム環境を設定します。
12. 画面上部に **Processing finished** が表示されたら、**Next** をクリックしてインストールを完了します。
13. 必要に応じて、**Generate Installation Script and Properties File** をクリックして、XML ファイルでインストールデータを保存し、**Done** をクリックします。  
 インストーラーは、2つのファイルを生成します。**auto.xml** ファイルは、今後のインストールを自動化し、**auto.xml.variables** ファイルは、ユーザーのパスワードと他の機密情報に関連する変数を保存します。**auto.xml** ファイルを使用して、元のインストールと同じタイプのサーバーおよび同じ設定の複数のシステムで Red Hat Process Automation Manager のインストールを繰り返します。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新します。XML ファイルを使用してインストールを実行するには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

これで、インストーラーを使用して Red Hat Process Automation Manager がインストールできました。Business Central だけをインストールした場合は、この手順を繰り返して、別のサーバーに KIE Server をインストールします。

### 注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、**ALLOW\_SNAPSHOT\_ISOLATION** と **READ\_COMMITTED\_SNAPSHOT** を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 13.2. CLI モードでのインストーラーの使用

コマンドラインインターフェイス (CLI) を使用して Red Hat Process Automation Manager インストーラーを実行できます。

### 注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

### 前提条件

- **Red Hat Process Automation Manager 7.13.5 Installer** をダウンロードしている。手順は、[12 章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) を参照してください。
- サポート対象の JDK がインストールされている。サポート対象の JDK のリストについては、[Red Hat Process Automation Manager 7 Supported Configurations](#) を参照してください。
- バックアップを作成してある Red Hat JBoss EAP 7.4 のサーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

- `$PATH` 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、`java-$JAVA_VERSION-openjdk-devel` パッケージに含まれています。



### 注記

Red Hat Process Automation Manager は、UTF-8 エンコーディングで機能するように設計されています。基礎となる JVM で別のエンコーディングシステムを使用すると、予期せぬエラーが発生する可能性があります。JVM で UTF-8 を使用するようするには、`-Dfile.encoding=UTF-8` のシステムプロパティを使用します。システムプロパティのリストは、[付録A Business Central システムプロパティ](#)を参照してください。

### 手順

1. 端末ウィンドウにおいて、インストーラーファイルをダウンロードしたディレクトリーに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar -console
```

コマンドラインの対話プロセスが開始し、使用許諾契約書が表示されます。

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

2. 使用許諾契約書を読んで **1** を入力し、**Enter** キーを押して続行します。

```
Specify the home directory of one of the following servers: Red Hat JBoss EAP 7 or Red Hat JBoss Web Server 5. For more information, see https://access.redhat.com/articles/3405381[Red Hat Process Automation Manager 7 Supported Configurations].
```

3. 既存の Red Hat JBoss EAP 7.4 インストールの親ディレクトリーを入力します。インストーラーが、指定したインストール場所を確認します。**1** を入力して確認し、続行します。



### 注記

Business Central と KIE Server は同じサーバーにインストールできます。ただし、実稼働環境では、Business Central と KIE Server は異なるサーバーにインストールすることが推奨されます。

4. インストーラーの指示に従って、インストールを完了します。



## 注記

ユーザー名およびパスワードを作成する場合は、指定したユーザー名が既存のロールまたはグループの名前と競合しないようにしてください。たとえば、**admin** というロールがある場合は、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし **&** の文字を使用することは **できません**。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

5. インストールが完了すると、以下のメッセージが表示されます。

```
Would you like to generate an automatic installation script and properties file?
```

6. **y** を入力してインストールデータが含まれる XML ファイルを作成します。あるいは、**n** を入力してインストールを完了します。**y** を入力すると、XML ファイルのパスを指定するように求められます。
7. パスを入力するか、Enter キーを押して提案されたパスを了承します。インストーラーは、2 つのファイルを生成します。**auto.xml** ファイルは、今後のインストールを自動化し、**auto.xml.variables** ファイルは、ユーザーのパスワードと他の機密情報に関連する変数を保存します。複数のシステムで **auto.xml** ファイルを使用して、元のインストールと同じ設定の同じタイプのサーバーに、Red Hat Process Automation Manager を簡単に繰り返しインストールできます。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新します。XML ファイルを使用してインストールを実行するには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

8. Business Central だけをインストールした場合は、この手順を繰り返して、別のサーバーに KIE Server をインストールします。



## 注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、ALLOW\_SNAPSHOT\_ISOLATION と READ\_COMMITTED\_SNAPSHOT を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 13.3. インストーラーでの PROCESS AUTOMATION MANAGER コントローラーを使用する KIE SERVER の設定

KIE Server は、Process Automation Manager コントローラーで管理することも、非管理モードにすることも可能です。KIE Server が非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE Server が管理されている場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Process Automation Manager コントローラーは Business Central と統合します。Business Central をインストールしている場合は、Business Central の **Execution Server** ページを使用して Process Automation Manager コントローラーと対話します。

インストーラーは対話モードまたは CLI モードで使用し、Business Central と KIE Server をインストールして、Process Automation Manager コントローラーで KIE Server を設定します。

### 前提条件

- バックアップを作成済みの Red Hat JBoss EAP 7.4 サーバーインストールが設定された 2 台のコンピューターが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

### 手順

1. 1 台目のコンピューターで、インタラクティブモードまたは CLI モードでインストーラーを実行します。詳細は以下のセクションのいずれかを参照してください。
  - [「インタラクティブモードでのインストーラーの使用」](#)
  - [「CLI モードでのインストーラーの使用」](#)
2. **Component Selection** ページで、**KIE Server** チェックボックスを外します。
3. Business Central インストールを完了します。
4. 2 台目のコンピューターで、インタラクティブモードまたは CLI モードでインストーラーを実行します。
5. **Component Selection** ページで **Business Central** チェックボックスを外します。
6. **Configure Runtime Environment** ページで **Perform Advanced Configuration** を選択します。
7. **Customize KIE Server properties** を選択し、**Next** をクリックします。
8. Business Central のコントローラー URL を入力し、KIE Server に追加のプロパティを設定します。コントローラー URL は、以下の形式を取ります。<**HOST:PORT**> は、2 台目のコンピューターの Business Central のアドレスに置き換えます。  

```
<HOST:PORT>/business-central/rest/controller
```
9. インストールを完了します。
10. Process Automation Manager コントローラーが Business Central と統合されていることを確認するには、Business Central の **Execution Servers** ページに移動して、設定した KIE Server が **REMOTE SERVERS** に表示されていることを確認します。

## 第14章 ZIP ファイルからの RED HAT PROCESS AUTOMATION MANAGER のインストール

Red Hat Process Automation Manager の ZIP ファイル (Business Central 用および KIE Server 用) を使用して、インストーラーを使用せずに Red Hat Process Automation Manager をインストールできます。



### 注記

実稼働環境では、Business Central と KIE Server は異なるサーバーにインストールすることが推奨されます。

ヘッドレス Process Automation Manager コントローラーのインストールは、[19章ヘッドレス Process Automation Manager コントローラーのインストールおよび実行](#)を参照してください。

### 14.1. ZIP ファイルからの BUSINESS CENTRAL のインストール

Business Central は、KIE Server が実行するビジネスルールを作成して管理するグラフィカルユーザーインターフェイスです。デプロイ可能な ZIP ファイルを使用して Business Central をインストールおよび設定できます。

#### 前提条件

- バックアップを作成済みの Red Hat JBoss EAP システム (バージョン 7.4) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP\_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。
- [12章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) に従って以下のファイルがダウンロードされている。  
**rhpam-7.13.5-business-central-eap7-deployable.zip**

#### 手順

1. **rhpam-7.13.5-business-central-eap7-deployable.zip** ファイルを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
2. **TEMP\_DIR/rhpam-7.13.5-business-central-eap7-deployable/jboss-eap-7.4** ディレクトリーの内容を **EAP\_HOME** にコピーします。プロンプトが表示されたら、ファイルをマージまたは置き換えます。



### 警告

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

## 14.2. ZIP ファイルからの KIE SERVER のインストール

KIE Server はビジネスアセットのランタイム環境を提供し、アセットリポジトリ (ナレッジストア) に保存されたデータにアクセスします。デプロイ可能な ZIP ファイルを使用して KIE Server をインストールおよび設定します。

### 前提条件

- バックアップを作成済みの Red Hat JBoss EAP システム (バージョン 7.4) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP\_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。
- [12章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) に従って以下のファイルがダウンロードされている。  
**rhpmam-7.13.5-kie-server-ee8.zip**

### 手順

1. **rhpmam-7.13.5-kie-server-ee8.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
2. **TEMP\_DIR/rhpmam-7.13.5-kie-server-ee8/kie-server.war** ディレクトリーを **EAP\_HOME/standalone/deployments/** にコピーします。



#### 警告

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

3. **TEMP\_DIR/rhpmam-7.13.5-kie-server-ee8/rhpmam-7.13.5-kie-server-ee8/SecurityPolicy/** の内容を **EAP\_HOME/bin** にコピーします。
4. ファイルを上書きするように求められたら、**Replace** をクリックします。
5. **EAP\_HOME/standalone/deployments/** ディレクトリーに、**kie-server.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動すると KIE Server が自動的にデプロイされます。

#### 注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、**ALLOW\_SNAPSHOT\_ISOLATION** と **READ\_COMMITTED\_SNAPSHOT** を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 14.3. KIE SERVER への JDBC データソースの設定

データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。

通常、Red Hat Process Automation Manager を使用するソリューションは、1つのトランザクション内で複数のリソースを管理します。非同期のジョブ、イベント、タイマーなどの JMS。Red Hat Process Automation Manager では、データの原子性と一貫した結果を保証するために、可能な限りデータソースに XA ドライバーを必要とします。異なるスキーマのトランザクションコードがリスナー内に存在する場合や、jBPM エンジンが提供するフックから取得する場合は、XA ドライバーも必要となります。

1つのトランザクションに複数のリソースが参加していないことが確認できない限り、XA 以外のデータソースを使用しないでください。



### 注記

実稼働環境の場合は、実際のデータソースを指定します。実稼働環境で、データソースの例は使用しないでください。

### 前提条件

- [Red Hat JBoss Enterprise Application Platform 設定ガイド](#) のデータソースの作成と JDBC ドライバーのセクションで説明されているように、データベース接続の作成に使用する JDBC プロバイダーが、KIE Server をデプロイするすべてのサーバーに設定されている。
- [Red Hat Process Automation Manager 7.13.5 Add Ons\(rhpam-7.13.5-add-ons.zip\)](#) ファイルを、Red Hat カスタマーポータル [の Software Downloads](#) ページからダウンロードしている。

### 手順

1. 以下の手順を実行して、データベースを準備します。
  - a. **TEMP\_DIR** などの一時ディレクトリーに **rhpam-7.13.5-add-ons.zip** を展開します。
  - b. **TEMP\_DIR/rhpam-7.13.5-migration-tool.zip** を展開します。
  - c. 現在のディレクトリーから、**TEMP\_DIR/rhpam-7.13.5-migration-tool/ddl-scripts** ディレクトリーに移動します。このディレクトリーには、複数のデータベースタイプの DDL スクリプトが含まれています。
  - d. 使用するデータベースに、お使いのデータベースタイプの DDL スクリプトをインポートします。

以下の例は、PostgreSQL で jBPM データベース構造を作成します。

```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



**注記**

PostgreSQL または Oracle を Spring Boot と併用する場合は、対応する Spring Boot の DDL スクリプト (`/ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql` または `/ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql`) をインポートする必要があります。

**注記**

PostgreSQL DDL スクリプトは、**@LOB** アノテーションが付けられたエンティティ属性の自動インクリメント整数値 (OID) 列で PostgreSQL スキーマを作成します。OID ではなく BYTEA などの他のバイナリ列タイプを使用するには、**postgresql-bytea-jbpm-schema.sql** スクリプトで PostgreSQL スキーマを作成し、Red Hat Process Automation Manager **org.kie.persistence.postgresql.useText=true** フラグおよび **org.kie.persistence.postgresql.useBytea=true** フラグを設定する必要があります。BYTEA ベースのスキーマの作成時に **postgresql-jbpm-lo-trigger-clob.sql** スクリプトを使用しないでください。Red Hat Process Automation Manager は、OID ベースから BYTEA ベースのスキーマに変更する移行ツールを提供しません。

2. テキストエディターで `EAP_HOME/standalone/configuration/standalone-full.xml` を開き、`<system-properties>` タグの場所を特定します。
3. 以下のプロパティを `<system-properties>` タグに追加します。`<DATASOURCE>` はデータソースの JNDI 名で、`<HIBERNATE_DIALECT>` はデータベースの Hibernate 方言です。

**注記**

**org.kie.server.persistence.ds** プロパティのデフォルト値は `java:jboss/datasources/ExampleDS` です。**org.kie.server.persistence.dialect** プロパティのデフォルト値は `org.hibernate.dialect.H2Dialect` です。

```
<property name="org.kie.server.persistence.ds" value="<DATASOURCE>"/>
<property name="org.kie.server.persistence.dialect" value="<HIBERNATE_DIALECT>"/>
```

以下の例では、PostgreSQL hibernate 方言のデータソースの設定方法を紹介しています。

```
<system-properties>
  <property name="org.kie.server.repo" value="${jboss.server.data.dir}"/>
  <property name="org.kie.example" value="true"/>
  <property name="org.jbpm.designer.perspective" value="full"/>
  <property name="designerdataobjects" value="false"/>
  <property name="org.kie.server.user" value="rhpamUser"/>
  <property name="org.kie.server.pwd" value="rhpam123!"/>
  <property name="org.kie.server.location" value="http://localhost:8080/kie-server/services/rest/server"/>
  <property name="org.kie.server.controller" value="http://localhost:8080/business-central/rest/controller"/>
  <property name="org.kie.server.controller.user" value="kieserver"/>
  <property name="org.kie.server.controller.pwd" value="kieserver1!"/>
  <property name="org.kie.server.id" value="local-server-123"/>
</system-properties>
```

```
<!-- Data source properties. -->
<property name="org.kie.server.persistence.ds"
value="java:jboss/datasources/KieServerDS"/>
<property name="org.kie.server.persistence.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/>
</system-properties>
```

以下の方言がサポートされます。

- DB2: **org.hibernate.dialect.DB2Dialect**
- MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
- MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
- MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
- Oracle: **org.hibernate.dialect.Oracle10gDialect**
- PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
- PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**
- Sybase: **org.hibernate.dialect.SybaseASE157Dialect**

## 14.4. ユーザーの作成

デプロイ可能な ZIP ファイルを使用して Red Hat Process Automation Manager をインストールした場合は、Business Central または KIE Server にログインする前にユーザーを作成する必要があります。本セクションでは、**admin**、**rest-all**、および **kie-server** のロールを持つ Business Central ユーザーと、**kie-server** ロールを持つ KIE Server ユーザーを作成する方法を説明します。ロールの詳細は、[11 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。



### 注記

Red Hat Process Automation Manager は、ユーザーデータをプロパティセットまたはファイルセットとして保存します。ファイルベースのストレージは、SSH ログインやユーザーメンテナンス UI など、複数の追加機能を提供します。

このドキュメントのユーザースクリプトの例では、プロパティベースのユーザースクリプト **add-users.sh** の代わりにファイルベースのユーザースクリプト **jboss-cli.sh** を使用しています。

### 前提条件

- Red Hat Process Automation Manager が Red Hat JBoss EAP インストールのベースディレクトリー (**EAP\_HOME**) にインストールされている。

### 手順

1. 必要に応じて、Red Hat Process Automation Manager でプロパティベースのユーザーストレージの使用をファイルベースのユーザーストレージに変更する場合は、以下の手順を実行します。
  - a. 以下のコマンドを実行して、**kie-fs-realm** パッチを適用します。

-

```
$ ./bin/elytron-tool.sh filesystem-realm --users-file application-users.properties --roles-file application-roles.properties --output-location kie-fs-realm-users
```

- b. 各 **kie-fs-realm-users**/\*/<USER>.xml ファイルを開きます。<USER> は Red Hat Process Automation Manager のユーザーです。
  - c. <attribute name="roles" value= を &lt;attribute name="role" value= に置き換えます。
2. 端末アプリケーションで **EAP\_HOME/bin** ディレクトリーに移動します。
  3. **admin**、**rest-all**、および **kie-server** ロールを持つユーザーを作成し、Business Central へのログインに使用します。



### 注記

**admin** ロールを持つユーザーは、Business Central の管理者です。**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** ロールを持つユーザーは、KIE Server REST 機能にアクセスできます。ユーザーが Business Central の **Manage** ビューおよび **Track** ビューにアクセスするには、**kie-server** ロールが必須となります。

以下のコマンドの <USERNAME> および <password> を、作成するユーザーとそのパスワードに置き換えます。

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password=<PASSWORD>}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[admin,rest-all,kie-server])"
```



### 注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。たとえば、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし & の文字は使用できません。

4. KIE Server へのログインに使用する **kie-server** ロールのユーザーを作成します。

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[kie-server])"
```

5. ユーザー名とパスワードを書き留めておきます。
6. オプション: Business Central と KIE Server を同じサーバーにインストールした場合は、両方のロールを持つユーザーを 1 つ作成できます。

-

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[admin,rest-all,kie-server])"
```



### 注記

実稼働環境では、Business Central と KIE Server は異なるサーバーにインストールすることが推奨されます。

- オプション: 一度に複数のユーザーを作成するには、ユーザーデータが含まれるファイルを作成し、以下のコマンドを実行します。<USER\_DATA>.cli はユーザーデータを含むファイルです。

```
$ ./bin/jboss-cli.sh --file=<USER_DATA>.cli
```

<USER\_DATA>.cli ファイルには、以下の例のようなデータが含まれている必要があります。

```
embed-server --std-out=echo

# first user
/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>)
/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'})
/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=[admin,role,group])

# second user
...

```

## 14.5. BUSINESS CENTRAL に接続する KIE SERVER の設定



### 警告

このセクションでは、テスト目的で使用可能なサンプルの設定を紹介します。一部の値は、実稼働環境には適しておらず、その旨を記載しています。

KIE Server を Red Hat Process Automation Manager 環境に設定していない場合、または Red Hat Process Automation Manager 環境に KIE Server を追加する必要がある場合は、KIE Server を設定して Business Central に接続する必要があります。



## 注記

Red Hat OpenShift Container Platform に KIE Server をデプロイする場合は、[Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#) で、Business Central に接続する設定手順を参照してください。

## 前提条件

- 以下のセクションのとおり、Business Central および KIE Server が Red Hat JBoss EAP インストールのベースディレクトリー (**EAP\_HOME**) にインストールされている。
  - 「ZIP ファイルからの Business Central のインストール」
  - 「ZIP ファイルからの KIE Server のインストール」
- 以下のロールを持つユーザーが存在している
  - Business Central: **rest-all** ロールを持つユーザー
  - KIE Server: **kie-server** ロールを持つユーザー  
詳細は、「[ユーザーの作成](#)」を参照。

## 手順

1. Red Hat Process Automation Manager インストールディレクトリーで、**standalone-full.xml** ファイルに移動します。たとえば、Red Hat Process Automation Manager に Red Hat JBoss EAP インストールを使用する場合は **\$EAP\_HOME/standalone/configuration/standalone-full.xml** に移動します。
2. **standalone-full.xml** ファイルを開き、**<system-properties>** タグの下に、以下の JVM プロパティを設定します。

表14.1 KIE Server インスタンスの JVM プロパティ

| プロパティ                                     | 値   | 注記  |
|---|---|---|
| <b>org.kie.server.id</b>                  | <b>default-kie-server</b>                                     | KIE Server ID。  |
| <b>org.kie.server.controller</b>          | <b>http://localhost:8080/business-central/rest/controller</b> | Business Central の場所<br>Business Central の API に接続する URL。 |
| <b>org.kie.server.controller.user</b>     | <b>controllerUser</b>   | Business Central にログイン可能な <b>rest-all</b> ロールを持つユーザー名。    |
| <b>org.kie.server.controller.password</b> | <b>controllerUser1234;</b>                                    | Business Central にログインできるユーザーのパスワード。                      |
| <b>org.kie.server.location</b>            | <b>http://localhost:8080/kie-server/services/rest/server</b>  | KIE Server の場所<br>KIE Server の API に接続する URL。             |

表14.2 Business Central インスタンスの JVM プロパティ

| プロパティ                            | 値                                | 注記                             |
|----------------------------------|----------------------------------|--------------------------------|
| <code>org.kie.server.user</code> | <code>controllerUser</code>      | <b>kie-server</b> ロールを持つユーザー名。 |
| <code>org.kie.server.pwd</code>  | <code>controllerUser1234;</code> | ユーザーのパスワード。                    |

以下の例は、KIE Server インスタンスを設定する方法を示しています。

```
<property name="org.kie.server.id" value="default-kie-server"/>
<property name="org.kie.server.controller" value="http://localhost:8080/business-central/rest/controller"/>
<property name="org.kie.server.controller.user" value="controllerUser"/>
<property name="org.kie.server.controller.pwd" value="controllerUser1234;"/>
<property name="org.kie.server.location" value="http://localhost:8080/kie-server/services/rest/server"/>
```

以下の例は、Business Central インスタンスに設定する方法を示しています。

```
<property name="org.kie.server.user" value="controllerUser"/>
<property name="org.kie.server.pwd" value="controllerUser1234;"/>
```

- KIE サーバーが正常に起動したことを確認するには、KIE サーバーが動作しているときに、`http://SERVER:PORT/kie-server/services/rest/server/` に GET リクエストを送信します。KIE サーバー上での Red Hat Process Automation Manager の実行に関する詳細は、[Red Hat Process Automation Manager の実行](#) を参照してください。認証に成功すると、以下の例のような XML 応答が返されます。

```
<response type="SUCCESS" msg="Kie Server info">
  <kie-server-info>
    <capabilities>KieServer</capabilities>
    <capabilities>BRM</capabilities>
    <capabilities>BPM</capabilities>
    <capabilities>CaseMgmt</capabilities>
    <capabilities>BPM-UI</capabilities>
    <capabilities>BRP</capabilities>
    <capabilities>DMN</capabilities>
    <capabilities>Swagger</capabilities>
    <location>http://localhost:8230/kie-server/services/rest/server</location>
    <messages>
      <content>Server KieServerInfo{serverId='first-kie-server', version='7.5.1.Final-redhat-1', location='http://localhost:8230/kie-server/services/rest/server', capabilities=[KieServer, BRM, BPM, CaseMgmt, BPM-UI, BRP, DMN, Swagger]}started successfully at Mon Feb 05 15:44:35 AEST 2018</content>
      <severity>INFO</severity>
      <timestamp>2018-02-05T15:44:35.355+10:00</timestamp>
    </messages>
    <name>first-kie-server</name>
    <id>first-kie-server</id>
    <version>7.5.1.Final-redhat-1</version>
  </kie-server-info>
</response>
```

4. 登録が正常に完了したことを確認します。
  - a. Business Central にログインします。
  - b. **Menu** → **Deploy** → **Execution Servers** の順にクリックします。  
正常に登録されている場合は、登録されたサーバーの ID が表示されます。

## 14.6. スレッド効率

最適なスレッド数を確保するには、スレッドシステムプロパティの値を CPU 数と CPU の合計に設定します。

1. Red Hat Process Automation Manager インストールディレクトリーで、**standalone-full.xml** ファイルに移動します。たとえば、Red Hat Process Automation Manager に Red Hat JBoss EAP インストールを使用する場合は **\$EAP\_HOME/standalone/configuration/standalone-full.xml** に移動します。
2. **standalone-full.xml** ファイルを開きます。
3. **<system-properties>** タグの下で、以下の JVM プロパティの値を CPU の数に設定します。

```
org.appformer.concurrent.managed.thread.limit  
org.appformer.concurrent.unmanaged.thread.limit  
org.appformer.concurrent.indexing.thread.limit  
org.appformer.concurrent.rest.api.thread.limit
```



### 注記

CPU 数と 1 つは、すべてのプロパティの有効なベースライン値です。追加のテストに基づいてさらに微調整する必要がある場合があります。

## 第15章 キーストアを使用したパスワードセキュリティの確保

キーストアを使用して、Business Central と KIE Server の間の通信に使用するパスワードを暗号化できます。コントローラーと KIE Server のパスワードを暗号化する必要があります。Business Central と KIE Server を別のアプリケーションサーバーにデプロイする場合は、いずれのアプリケーションサーバーもキーストアを使用する必要があります。

Java Cryptography Extension KeyStore (JCEKS) は、対称鍵をサポートするため、キーストアには JCEKS を使用してください。JDK インストールに含まれる KeyTool を使用して、新しい JCEKS を作成します。



### 注記

KIE Server が JCEKS で設定されていない場合、KIE Server のパスワードはシステムプロパティにプレーンテキスト形式で保存されます。

### 前提条件

- Red Hat JBoss EAP に KIE Server がインストールされている。
- Java 8 以降がインストールされている。

### 手順

1. Red Hat JBoss EAP のホームディレクトリーで、以下のコマンドを入力して、**kie-server** ロールを割り当てた KIE Server ユーザーを作成し、パスワードを指定します。以下の例で、**<USERNAME>** および **<PASSWORD>** を、任意のユーザー名とパスワードに置き換えます。

```
$<EAP_HOME>./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>)/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=['kie-server'])"
```

2. KeyTool を使用して JCEKS を作成するには、Java 8 のホームディレクトリーで以下のコマンドを実行します。

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass <ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass <KEYSTORE_PASSWORD> -storetype JCEKS
```

上記の例では、以下の変数を置き換えてください。

- **<KEYSTORE\_PATH>**: キーストアの保存先のパス
  - **<KEYSTORE\_PASSWORD>**: キーストアのパスワード
  - **<ALIAS\_KEY\_PASSWORD>**: エイリアスで保存した値にアクセスする時に使用するパスワード
  - **<PASSWORD\_ALIAS>**: プロセスへのエントリーに使用するエイリアス
3. プロンプトが表示されたら、作成した KIE Server ユーザーのパスワードを入力します。



4. **EAP\_HOME/standalone/configuration/standalone-full.xml** ファイルに以下のシステムプロパティを設定して、以下の表に記載のプレースホルダーを置き換えます。

```
<system-properties>
  <property name="kie.keystore.keyStoreURL" value="<KEYSTORE_URL>"/>
  <property name="kie.keystore.keyStorePwd" value="<KEYSTORE_PWD>"/>
  <property name="kie.keystore.key.server.alias" value="<KEY_SERVER_ALIASES>"/>
  <property name="kie.keystore.key.server.pwd" value="<KEY_SERVER_PWD>"/>
  <property name="kie.keystore.key.ctrl.alias" value="<KEY_CONTROL_ALIASES>"/>
  <property name="kie.keystore.key.ctrl.pwd" value="<KEY_CONTROL_PWD>"/>
</system-properties>
```

表15.1 KIE Server JCEKS を読み込む時に使用するシステムプロパティ

| システムプロパティ                            | プレースホルダー                           | 説明   |
|--------------------------------------|------------------------------------|--|
| <b>kie.keystore.keyStoreURL</b>      | <b>&lt;KEYSTORE_URL&gt;</b>        | 使用する JCEKS の URL (例:<br><b>file:///home/kie/keystores/keystore.jceks</b> ) |
| <b>kie.keystore.keyStorePwd</b>      | <b>&lt;KEYSTORE_PWD&gt;</b>        | JCEKS のパスワード   |
| <b>kie.keystore.key.server.alias</b> | <b>&lt;KEY_SERVER_ALIASES&gt;</b>  | パスワードの保存先となる REST サービスのキーのエイリアス  |
| <b>kie.keystore.key.server.pwd</b>   | <b>&lt;KEY_SERVER_PWD&gt;</b>      | 保存したパスワードを使用する REST サービスのエイリアスのパスワード                                       |
| <b>kie.keystore.key.ctrl.alias</b>   | <b>&lt;KEY_CONTROL_ALIASES&gt;</b> | パスワードの保存先のデフォルトの REST Process Automation Controller のキーのエイリアス              |
| <b>kie.keystore.key.ctrl.pwd</b>     | <b>&lt;KEY_CONTROL_PWD&gt;</b>     | 保存したパスワードを使用する、デフォルトの REST Process Automation Controller のエイリアスのパスワード      |

5. KIE Server を起動して、設定を検証します。

## 第16章 SSH が RSA を使用するように設定する手順

Git リポジトリのクローンには、SSH を使用します。デフォルトで、Business Central には DSA 暗号化アルゴリズムが含まれます。ただし、Fedora 23 環境の SSH クライアントなどは、DSA アルゴリズムではなく、RSA アルゴリズムを使用します。Business Central にはシステムプロパティーが含まれており、必要に応じて DSA から RSA に切り替えることができます。



### 注記

Red Hat Enterprise Linux 7 などの、サポート対象の設定で SSH クライアントを使用している場合は、この問題の影響はありません。サポート対象の設定に関するリストは、[Red Hat Process Automation Manager 7 でサポートされる設定](#) を参照してください。

### 手順

以下のタスクの1つを実行して、このシステムプロパティーを有効にします。

- 以下のようにクライアント側の `~/.ssh/config` ファイルを変更して、SSH クライアントが強制的に非推奨の DSA アルゴリズムを受け入れるようにします。

```
Host <SERVER_IP>  
  HostKeyAlgorithms +ssh-dss
```

- 以下のように、Business Central の起動時に、`-Dorg.uberfire.nio.git.ssh.algorithm=RSA` パラメーターを追加します。

```
$ ./standalone.sh -c standalone-full.xml  
-Dorg.uberfire.nio.git.ssh.algorithm=RSA
```

## 第17章 RED HAT PROCESS AUTOMATION MANAGER インストールの検証

Red Hat Process Automation Manager をインストールしたら、アセットを作成してシステムが機能していることを検証します。

### 手順

1. 端末ウィンドウで **EAP\_HOME/bin** ディレクトリーに移動し、以下のコマンドを実行して Red Hat Process Automation Manager を起動します。

```
./standalone.sh -c standalone-full.xml
```



### 注記

KIE Server を使用せず、Business Central だけをデプロイした場合は、**standalone.sh** スクリプトに **standalone-full.xml** ファイルを指定せず Business Central だけを起動できます。この場合は、Business Central を起動する前に **standalone.xml** ファイルの設定を変更しておく必要があります。

2. Web ブラウザーで **localhost:8080/business-central** を開きます。
  - ドメイン名から実行するように Red Hat Process Automation Manager を設定した場合は、以下のように **localhost** をドメイン名に置き換えます。  
**http://www.example.com:8080/business-central**
  - クラスターで実行するように Red Hat Process Automation Manager を設定した場合は、以下のように **localhost** を、特定ノードの IP アドレスに置き換えます。  
**http://<node\_IP\_address>:8080/business-central**
3. インストール時に作成した **admin** ユーザーの認証情報を入力します。Business Central ホームページが表示されます。
4. **Menu** → **Deploy** → **Execution Servers** の順に選択します。
5. **default-kieserver** が **Server Configurations** 下に記載されていることを確認します。
6. **Menu** → **Design** → **Projects** の順に選択します。
7. **MySpace** スペースを開きます。
8. **Try Samples** → **Mortgages** → **OK** の順にクリックします。**Assets** ウィンドウが表示されます。
9. **Add Asset** → **Data Object** をクリックします。
10. **Data Object** フィールドに **MyDataObject** と入力し、**OK** をクリックします。
11. **Spaces** → **mySpace** → **Mortgages** の順にクリックし、アセットリストに **MyDataObject** があることを確認します。
12. **Mortgages** プロジェクトを選択します。

## 第18章 RED HAT PROCESS AUTOMATION MANAGER の実行

Red Hat JBoss EAP 上で Red Hat Process Automation Manager をスタンドアロンモードで実行するには、以下の手順を実行します。

### 前提条件

- Red Hat Process Automation Manager がインストールされ、設定されている。



### 注記

デフォルトのホスト (**localhost**) またはデフォルトのポート (**9990**) を変更した場合には、Red Hat Process Automation Manager を実行する前に **business-central.war/WEB-INF/classes/datasource-management.properties** ファイルおよび **business-central.war/WEB-INF/classes/security-management.properties** ファイルを [Solution 3519551](#) に記載の説明に沿って編集する必要があります。

### 手順

- ターミナルアプリケーションで **EAP\_HOME/bin** に移動します。
- スタンドアロン設定を実行します。

- Linux または UNIX ベースのシステムの場合:

```
$. /standalone.sh -c standalone-full.xml
```

- Windows の場合:

```
standalone.bat -c standalone-full.xml
```



### 注記

KIE Server を使用せず、Business Central だけをデプロイした場合は、**standalone.sh** スクリプトに **standalone-full.xml** ファイルを指定せず Business Central だけを起動できます。この場合は、Business Central を起動する前に **standalone.xml** ファイルの設定を変更しておく必要があります。

Linux または UNIX ベースのシステムの場合:

```
$. /standalone.sh
```

Windows の場合:

```
standalone.bat
```

- Web ブラウザーで、URL **localhost:8080/business-central** を開きます。  
ドメイン名から実行するように Red Hat Process Automation Manager を設定した場合は、以下のように **localhost** をドメイン名に置き換えます。

**http://www.example.com:8080/business-central**

4. 「[ユーザーの作成](#)」の Business Central で作成したユーザーの認証情報を使用してログインします。

## 第19章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行

KIE Server は、管理モードまたは非管理モードで動作するように設定できます。KIE Server が非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE Server が管理されている場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Business Central には Process Automation Manager コントローラーが組み込まれています。Business Central をインストールしている場合は、**Execution Server** ページを使用して KIE コンテナを作成および維持します。Business Central なしで KIE Server の管理を自動化する場合は、ヘッドレス Process Automation Manager コントローラーを使用できます。

### 19.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストール

ヘッドレス Process Automation Manager コントローラーをインストールして、REST API または KIE Server Java Client API を使用して対話します。

#### 前提条件

- バックアップを作成済みの Red Hat JBoss EAP システム (バージョン 7.4) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP\_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

#### 手順

1. Red Hat カスタマーポータル [の Software Downloads ページ](#) に移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons(**rhpam-7.13.5-add-ons.zip** ファイル) をダウンロードします。
3. **rhpam-7.13.5-add-ons.zip** ファイルを展開します。 **rhpam-7.13.5-controller-ee7.zip** ファイルは展開したディレクトリーにあります。
4. **rhpam-7.13.5-controller-ee7.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
5. **TEMP\_DIR/rhpam-7.13.5-controller-ee7/controller.war** ディレクトリーを **EAP\_HOME/standalone/deployments/** にコピーします。



### 警告

コピーするヘッドレス Process Automation Manager コントローラーデプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

6. **TEMP\_DIR/rhpam-7.13.5-controller-ee7/SecurityPolicy/** ディレクトリーの内容を **EAP\_HOME/bin** にコピーします。
7. ファイルの上書きを求めるプロンプトが出されたら、**Yes** を選択します。
8. **EAP\_HOME/standalone/deployments/** ディレクトリーに、**controller.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動するとヘッドレス Process Automation Manager コントローラーが自動的にデプロイされます。

## 19.1.1. ヘッドレス Process Automation Manager コントローラーのユーザー作成

ヘッドレス Process Automation Manager コントローラーを使用する前に、**kie-server** ロールを持つユーザーを作成する必要があります。

### 前提条件

- ヘッドレス Process Automation Manager コントローラーが Red Hat JBoss EAP インストールのベースディレクトリー (**EAP\_HOME**) にインストールされている。

### 手順

1. 端末アプリケーションで **EAP\_HOME/bin** ディレクトリーに移動します。
2. 以下のコマンドを入力し、**<USERNAME>** および **<PASSWORD>** を、作成するユーザー名およびパスワードに置き換えます。

```
$ ./bin/jboss-cli.sh --commands="embed-server --std-out=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=['kie-server'])"
```



### 注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。たとえば、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし & の文字は使用できません。

3. ユーザー名とパスワードを書き留めておきます。

## 19.1.2. KIE Server およびヘッドレス Process Automation Manager コントローラーの設定

KIE Server をヘッドレス Process Automation Manager コントローラーから管理する場合は、KIE Server インストールの **standalone-full.xml** ファイルと、ヘッドレス Process Automation Manager コントローラーインストールの **standalone.xml** ファイルを編集する必要があります。

### 前提条件

- KIE Server が **EAP\_HOME** にインストールされている。
- ヘッドレス Process Automation Manager コントローラーが **EAP\_HOME** にインストールされている。



### 注記

実稼働環境では KIE Server およびヘッドレス Process Automation Manager コントローラーを異なるサーバーにインストールすることを推奨します。ただし、開発環境など、KIE Server およびヘッドレス Process Automation Manager コントローラーを同じサーバーにインストールする場合は、併せて共有の **standalone-full.xml** ファイルを変更します。

- KIE Server ノードに、**kie-server** ロールのあるユーザーが作成されている。
- サーバーノードに、**kie-server** ロールのあるユーザーが作成されている。詳細は、「[ユーザーの作成](#)」を参照。

### 手順

1. **EAP\_HOME/standalone/configuration/standalone-full.xml** ファイルの **<system-properties>** セクションに以下のプロパティを追加し、**<USERNAME>** および **<USER\_PWD>** を、**kie-server** ロールを持つユーザーの認証情報に置き換えます。

```
<property name="org.kie.server.user" value="<USERNAME>"/>
<property name="org.kie.server.pwd" value="<USER_PWD>"/>
```

2. KIE Server の **EAP\_HOME/standalone/configuration/standalone-full.xml** ファイルの **<system-properties>** セクションに以下のプロパティを追加します。

```
<property name="org.kie.server.controller.user" value="<CONTROLLER_USER>"/>
<property name="org.kie.server.controller.pwd" value="<CONTROLLER_PWD>"/>
<property name="org.kie.server.id" value="<KIE_SERVER_ID>"/>
<property name="org.kie.server.location" value="http://<HOST>:<PORT>/kie-server/services/rest/server"/>
<property name="org.kie.server.controller" value="<CONTROLLER_URL>"/>
```

3. このファイルで、以下の値を置き換えます。
  - **<CONTROLLER\_USER>** および **<CONTROLLER\_PWD>** を **kie-server** ロールを持つユーザーの認証情報に置き換えます。
  - **<KIE\_SERVER\_ID>** を KIE Server システムの ID または名前に置き換えます (例: **rhcam-7.13.5-kie-server-1**)。



- **<HOST>** を KIE Server ホストの ID または名前に置き換えます (例: **localhost** または **192.7.8.9**)。
- **<PORT>** を KIE Server ホストのポートに置き換えます (例: **8080**)。



#### 注記

**org.kie.server.location** プロパティで KIE Server の場所を指定します。

- **<CONTROLLER\_URL>** をヘッドレス Process Automation Manager コントローラーの URL で置き換えます。起動中に KIE Server がこの URL に接続します。

## 19.2. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーの実行

ヘッドレス Process Automation Manager コントローラーを Red Hat JBoss EAP にインストールしたら、以下の手順に従ってヘッドレス Process Automation Manager コントローラーを実行します。

### 前提条件

- ヘッドレス Process Automation Manager コントローラーが Red Hat JBoss EAP インストールのベースディレクトリー (**EAP\_HOME**) にインストールされ設定されている。

### 手順

1. ターミナルアプリケーションで **EAP\_HOME/bin** に移動します。
2. ヘッドレス Process Automation Manager コントローラーを、KIE Server をインストールした Red Hat JBoss EAP インスタンスと同じ Red Hat JBoss EAP インスタンスにインストールしている場合は、以下のいずれかのコマンドを実行します。

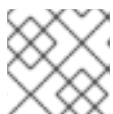
- Linux または UNIX ベースのシステムの場合:

```
┌ $ ./standalone.sh -c standalone-full.xml
```

- Windows の場合:

```
┌ standalone.bat -c standalone-full.xml
```

3. ヘッドレス Process Automation Manager コントローラーを、KIE Server をインストールした Red Hat JBoss EAP インスタンスとは別の Red Hat JBoss EAP インスタンスにインストールしている場合は、**standalone.sh** スクリプトで Process Automation Manager コントローラーを開始します。



#### 注記

この場合は、**standalone.xml** ファイルに必要な設定変更を加えます。

- Linux または UNIX ベースのシステムの場合:

```
┌ $ ./standalone.sh
```

- Windows の場合:

```
standalone.bat
```

4. ヘッドレス Process Automation Manager コントローラーが Red Hat JBoss EAP 上で動作していることを確認するには、以下のコマンドを入力します。ここで、**<CONTROLLER>** と **<CONTROLLER\_PWD>** は、「[ヘッドレス Process Automation Manager コントローラーのユーザー作成](#)」で作成したユーザー名とパスワードの組み合わせになります。このコマンドにより、KIE Server インスタンスに関する情報が出力されます。

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```



#### 注記

あるいは、KIE Server Java API Client を使用して、ヘッドレス Process Automation Manager コントローラーにアクセスすることもできます。

## 第20章 スタンドアロン BUSINESS CENTRAL の設定および実行

Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。設定ファイルのサンプルを使用して、追加設定なしで Business Central スタンドアロン JAR ファイルを実行するか、要件に合わせてサンプルファイルをカスタマイズできます。



### 注記

この JAR ファイルは、Red Hat Enterprise Linux で実行している場合にのみサポートされます。

### 前提条件

- 12章 *Red Hat Process Automation Manager インストールファイルのダウンロード* の記載通りに、Red Hat Process Automation Manager 7.13.5 Business Central Standalone (**rhpmam-7.13.5-business-central-standalone.jar**) ファイルおよび Red Hat Process Automation Manager 7.13.5 Add Ons (**rhpmam-7.13.5-add-ons.zip**) ファイルが、Red Hat Process Automation Manager 7.13 の [Software Downloads](#) ページからダウンロードされている。

### 手順

1. ダウンロードした **rhpmam-7.13.5-addons.zip** ファイルを一時ディレクトリーに展開します。このアーカイブには **rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルが含まれます。
2. **rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルを、**rhpmam-7.13.5-business-central-standalone.jar** ファイルが含まれるディレクトリーに展開します。**rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルには、以下の設定ファイルのサンプルが含まれています。
  - **application-script.cli**: ユーザーおよび kie サーバーシステムプロパティを追加するスクリプトの例
  - **kie-fs-realm-users**: ユーザーデータのサンプル  
設定ファイルで提供されるサンプルデータを使用して **rhpmam-7.13.5-business-central-standalone.jar** ファイルを実行するか、要件に合わせてデータをカスタマイズできます。
3. 設定データをカスタマイズするには、以下の手順を実行します。
  - a. **application-script.cli** ファイルを編集し、**admin**、**user**、**rest-all**、**rest-client**、**kie-server** のロールを持つ管理者ユーザーを追加します。以下の例で、**<USERNAME>** および **<PASSWORD>** を、作成するユーザーのユーザー名とパスワードに置き換えます。

```
/subsystem=elytron/filesystem-realm=KieRealm:add-identity(identity=<USERNAME>)
/subsystem=elytron/filesystem-realm=KieRealm:set-password(identity=<USERNAME>,
clear={password="<PASSWORD>"})
/subsystem=elytron/filesystem-realm=KieRealm:add-identity-attribute(identity=
<USERNAME>, name=role, value=["admin","user","rest-all","rest-client","kie-server"])
```

4. Business Central スタンドアロン JAR ファイルを実行するには、以下のコマンドを入力します。

```
java -jar rhpmam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli
```

5. JAR ファイルの実行時にアプリケーションプロパティを設定するには、コマンドに -**D<PROPERTY>=<VALUE>** パラメーターを追加します。ここで、<PROPERTY> はサポートされるアプリケーションプロパティの名前で、<VALUE> はプロパティの値になります。

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -  
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

たとえば、Business Central を実行し、ユーザー **controllerUser** として KIE Server に接続するには、次のコマンドを実行します。

```
java -jar rhpam-7.13.5-business-central-standalone.jar \  
--cli-script=application-script.cli \  
-Dorg.kie.server.user=controllerUser \  
-Dorg.kie.server.pwd=controllerUser1234
```

これにより、コンテナを KIE Server にデプロイできます。詳細は、[付録A Business Central システムプロパティ](#) を参照してください。



### 注記

Business Central でユーザーとグループの管理を有効にするには、**org.uberfire.ext.security.management.wildfly.cli.folderPath** プロパティの値を **kie-fs-realm-users** に設定します。

## 第21章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリ

Red Hat Process Automation Manager プロジェクトの作成時に、Business Central は Business Central 用に設定された Maven リポジトリを使用します。Maven project object model (POM) ファイル (**pom.xml**) を変更することで、Maven グローバルまたはユーザー設定を使用して、すべての Red Hat Process Automation Manager プロジェクトに対して Red Hat Process Automation Manager の公開リポジトリから依存関係を取得するように指示できます。また、Business Central と KIE Server が外部の Maven リポジトリを使用するか、Maven のミラーを準備してオフラインで使用できるように設定できます。

Red Hat Process Automation Manager のパッケージ化およびデプロイメントのオプションに関する詳細は、[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。

### 21.1. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加

Red Hat Process Automation Manager プロジェクトで適切な Maven 依存関係を使用するには、プロジェクトの **pom.xml** ファイルに Red Hat Business Automation の BOM (bill of materials) ファイルを追加します。Red Hat Business Automation BOM は、Red Hat Decision Manager と Red Hat Process Automation Manager の両方に適用されます。BOM ファイルを追加すると、提供される Maven リポジトリから、推移的依存関係の適切なバージョンがプロジェクトに含められます。

Red Hat Business Automation BOM (Bill of Materials) に関する詳細情報は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#)を参照してください。

#### 手順

1. Red Hat Business Automation BOM を **pom.xml** ファイルで宣言します。

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.redhat.ba</groupId>
      <artifactId>ba-platform-bom</artifactId>
      <version>7.13.5.redhat-00002</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
<dependencies>
  <!-- Your dependencies -->
</dependencies>
```

2. **<dependencies>** タグでお使いのプロジェクトに必要な依存関係を宣言します。製品の BOM をプロジェクトにインポートしたら、ユーザー向け製品依存関係のバージョンが定義されるため、**<dependency>** 要素のサブ要素 **<version>** を指定する必要はありません。ただし、**<dependency>** 要素を使用して、プロジェクトで使用する依存関係を宣言する必要があります。

- Business Central に作成されない標準のプロジェクトでは、お使いのプロジェクトに必要な依存関係をすべて指定します。Business Central に作成するプロジェクトでは、基本的なデザインエンジンとプロセスエンジンの依存関係が Business Central に自動的に提供されます。
- 標準的な Red Hat Process Automation Manager プロジェクトでは、使用する機能に応じて、以下の依存関係を宣言します。

### 埋め込みプロセスエンジンの依存関係

```
<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>
```

```

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>${logback.version}</version>
</dependency>

```

- CDI を使用する Red Hat Process Automation Manager プロジェクトでは、通常、以下の依存関係を宣言します。

### CDI が有効化されたプロセスエンジンの依存関係

```

<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>

```

- 標準的な Red Hat Process Automation Manager プロジェクトでは、以下の依存関係を宣言します。

### 埋め込みデシジョンエンジン依存関係

```

<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>

```

```

<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>

```

- KIE Server を使用するには、以下の依存関係を宣言します。

#### クライアントアプリケーション KIE Server の依存関係

```

<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-client</artifactId>
</dependency>

```

- Red Hat Process Automation Manager にリモートクライアントを作成するには、以下の依存関係を宣言します。

#### クライアントの依存関係

```

<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>

```

- ルール、プロセス定義など、アセットを含む JAR ファイルを作成する場合は、お使いの Maven プロジェクトのパッケージングの種類を **kjar** と指定し、**org.kie:kie-maven-plugin** を使用して、**<project>** 要素に置かれた **kjar** パッケージングタイプを処理します。以下の例の **\${kie.version}** は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) に記載されている Maven ライブラリーのバージョンです。

```

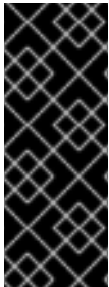
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>

```



## 21.2. BUSINESS CENTRAL および KIE SERVER への外部 MAVEN リポジトリの設定

Business Central および KIE Server が、内部のリポジトリではなく、Nexus や Artifactory などの外部の Maven リポジトリを使用するように設定できます。このように設定することで、Business Central と KIE Server は外部の Maven リポジトリで管理されているアーティファクトにアクセスしてダウンロードできます。



### 重要

Maven ではアーティファクトが不変である必要があるため、リポジトリ内のアーティファクトは自動セキュリティパッチを受け取りません。その結果、既知のセキュリティ問題のパッチがないアーティファクトはリポジトリに残り、これらに依存するビルドが破損しないようにします。パッチが適用されたアーティファクトのバージョン番号がインクリメントされます。詳細は、[JBoss Enterprise Maven リポジトリ](#) を参照してください。



### 注記

Red Hat OpenShift Container Platform のオーサリング環境向けに外部の Maven リポジトリを設定する方法については、以下のドキュメントを参照してください。

- [Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#)
- [テンプレートを使用した Red Hat OpenShift Container Platform 3 への Red Hat Process Automation Manager 環境のデプロイメント](#)

### 前提条件

- Business Central および KIE Server がインストールされている。インストールオプションは、[Red Hat Process Automation Manager インストールの計画](#) を参照してください。

### 手順

1. 外部リポジトリの接続およびアクセスの詳細が含まれる Maven **settings.xml** ファイルを作成します。**settings.xml** ファイルの詳細は Maven の [Settings Reference](#) を参照してください。
2. 既知の場所 (例: `/opt/custom-config/settings.xml`) にファイルを保存します。
3. Red Hat Process Automation Manager インストールディレクトリで、**standalone-full.xml** ファイルに移動します。たとえば、Red Hat Process Automation Manager に Red Hat JBoss EAP インストールを使用する場合は `$EAP_HOME/standalone/configuration/standalone-full.xml` に移動します。
4. **standalone-full.xml** の `<system-properties>` タグで、**kie.maven.settings.custom** プロパティに **settings.xml** ファイルのフルパス名を設定します。以下に例を示します。

```
<property name="kie.maven.settings.custom" value="/opt/custom-config/settings.xml"/>
```

5. Business Central と KIE Server を起動または再起動します。

### 次のステップ

KJAR アーティファクトとして外部の Maven リポジトリにエクスポートまたはプッシュする Business Central のプロジェクトごとに、プロジェクトの **pom.xml** ファイルにリポジトリの情報を追加する必要があります。方法については、[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。

## 21.3. オフラインで使用する MAVEN ミラーリポジトリの準備

パブリックインターネットへの送信アクセスが、Red Hat Process Automation Manager のデプロイメントに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、このリポジトリをお使いの環境で使用できるようにする必要があります。



### 注記

Red Hat Process Automation Manager デプロイメントがインターネットに接続されている場合には、この手順を実行する必要はありません。

### 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. 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。
2. Red Hat カスタマーポータル[の Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
    - a. **Red Hat Process Automation Manager 7.13.5 Offliner Content List(rhpam-7.13.5-offliner.zip)** の製品配信可能ファイルをダウンロードして展開します。
    - b. **rhpam-7.13.5-offliner.zip** ファイルの内容を任意のディレクトリーに展開します。
    - c. ディレクトリーに移動し、以下のコマンドを入力します。

```
./offline-repo-builder.sh offliner.txt
```

このコマンドは、**repository** サブディレクトリーを作成し、必要なアーティファクトをこのサブディレクトリーにダウンロードします。これはミラーリポジトリです。

一部のダウンロードが失敗したことを示すメッセージが表示された場合は、同じコマンドを再度実行してください。ダウンロードが再び失敗する場合は、Red Hat サポートに連絡してください。

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` を、プロジェクトの **pom.xml** ファイルの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`) の内容を、作成した **repository** サブディレクトリーにコピーします。
4. **repository** サブディレクトリーの内容を、Red Hat Process Automation Manager をデプロイしたコンピューターのディレクトリーにコピーします。このディレクトリーがオフラインの Maven ミラーリポジトリになります。
  5. 「[Business Central および KIE Server への外部 Maven リポジトリの設定](#)」の説明に従って、Red Hat Process Automation Manager デプロイメント向けに、**settings.xml** ファイルを作成して設定します。
  6. **settings.xml** ファイルで以下を変更します。

- `<profile>` タグの下に、`<repositories>` タグまたは `<pluginReposiores>` タグがない場合は、不足しているタグを追加します。
- `<repositories>` の下に、以下のコンテンツを追加します。

```
<repository>
  <id>offline-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

`/path/to/repo` は、ローカルの Maven ミラーリポジトリのディレクトリーに対する完全パスに置き換えます。

- `<pluginRepositories>` の下に、以下のコンテンツを追加します。

```
<repository>
  <id>offline-plugin-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

`/path/to/repo` は、ローカルの Maven ミラーリポジトリのディレクトリーに対する完全パスに置き換えます。

7. Business Central の `kie.maven.offline.force` プロパティを `true` に設定します。Business Central のプロパティの設定手順は、[Red Hat JBoss EAP 7.4 への Red Hat Process Automation Manager のインストールおよび設定](#) を参照してください。

## 第22章 GIT リポジトリからのプロジェクトのインポート

Git は分散バージョン管理システムです。リビジョンをコミットオブジェクトとして実装します。リポジトリに変更を保存すると、Git リポジトリに新しいコミットオブジェクトが作成されます。

Business Central は Git を使用してプロジェクトデータ (ルールやプロセスなどのアセットを含む) を格納します。Business Central でプロジェクトを作成すると、Business Central に埋め込まれている Git リポジトリに追加されます。他の Git リポジトリにプロジェクトがある場合は、Business Central スペースから、そのプロジェクトを Business Central の Git リポジトリにインポートできます。

### 前提条件

- Red Hat Process Automation Manager プロジェクトが外部の Git リポジトリに存在している。
- 外部の Git リポジトリへの読み取りアクセスに必要な認証情報がある。

### 手順

1. Business Central で **Menu** → **Design** → **Projects** の順にクリックします。
2. プロジェクトをインポートするスペースを選択または作成します。デフォルトのスペースは **MySpace** です。
3. プロジェクトをインポートするには、次のいずれかを実行します。
  - **Import Project** をクリックします。
  - ドロップダウンリストから **Import Project** を選択します。
4. **Import Project** ウィンドウに、インポートするプロジェクトが含まれる Git リポジトリの URL および認証情報を入力し、**Import** をクリックします。プロジェクトが Business Central の Git リポジトリに追加され、現在のスペースで利用できるようになります。

## 第23章 BUSINESS CENTRAL ブランディングのカスタマイズ

独自のイメージに置き換えることで、Business Central ログインページやアプリケーションヘッダーのブランディングをカスタマイズできます。

### 23.1. BUSINESS CENTRAL ログインページのカスタマイズ

Business Central ログインページで、企業のロゴおよびプロジェクトロゴをカスタマイズできます。

#### 手順

1. Red Hat JBoss EAP を起動し、Web ブラウザーで Business Central を開きます。
2. SVG 形式のイメージを、Red Hat Process Automation Manager インストールの **EAP\_HOME/standalone/deployments/business-central.war/img/** ディレクトリーにコピーします。
3. **EAP\_HOME//standalone/deployments/business-central.war/img/** ディレクトリーで、既存の **redhat\_logo.png** ファイルを移動するか名前を変更します。
4. PNG ファイル **redhat\_logo.png** の名前を変更します。
5. **User name** フィールドおよび **Password** フィールドの上に表示されるプロジェクトロゴを変更するには、デフォルトのイメージ **BC\_Logo.png** を新しい SVG ファイルに置き換えます。
6. ログインページを強制的に完全に再読み込みし、キャッシュを消去して変更を表示します。たとえば、Linux および Windows の多くの Web ブラウザーでは、Ctrl+F5 を押します。

### 23.2. BUSINESS CENTRAL アプリケーションヘッダーのカスタマイズ

Business Central アプリケーションヘッダーをカスタマイズできます。

#### 手順

1. Red Hat JBoss EAP を起動し、Web ブラウザーで Business Central を開き、お使いのユーザー認証情報でログインします。
2. SVG 形式の新しいアプリケーションヘッダーイメージを、Red Hat Process Automation Manager システムの **EAP\_HOME/standalone/deployments/business-central.war/banner/** ディレクトリーにコピーします。
3. テキストエディターで **EAP\_HOME/standalone/deployments/business-central.war/banner/banner.html** ファイルを開きます。
4. **<img>** タグの **logo.png** を、新しい **image:admin-and-config/** のファイル名に置き換えます。

```

```

5. ログインページを強制的に完全に再読み込みし、キャッシュを消去して変更を表示します。たとえば、Linux および Windows の多くの Web ブラウザーでは、Ctrl+F5 を押します。

## 第24章 LDAP と SSL の統合

Red Hat Process Automation Manager では、Red Hat Single Sign-On を通じて LDAP と SSL を統合することができます。詳細は [Red Hat Single Sign-On サーバー管理ガイド](#) を参照してください。

## 付録A BUSINESS CENTRAL システムプロパティ

このセクションに記載の Business Central のシステムプロパティは **standalone\*.xml** ファイルに渡されます。

### Git ディレクトリー

以下のプロパティを使用して、Business Central Git ディレクトリーの場所と名前を設定します。

- **org.uberfire.nio.git.dir**: Business Central の Git ディレクトリーの場所。
- **org.uberfire.nio.git.dirname**: Business Central の Git ディレクトリーの名前。デフォルト値は **.niogit** です。
- **org.uberfire.nio.git.ketch**: Git ketch を有効化または無効化。
- **org.uberfire.nio.git.hooks**: Business Central の Git ディレクトリーの場所。

### HTTP 経由の Git

次のプロパティを使用して、HTTP 経由で Git リポジトリーにアクセスできるように設定します。

- **org.uberfire.nio.git.proxy.ssh.over.http**: SSH が HTTP プロキシを使用するかどうかを指定します。デフォルト値は **false** です。
- **http.proxyHost**: HTTP プロキシのホスト名を定義します。デフォルト値は **null** です。
- **http.proxyPort**: HTTP プロキシのポート (整数値) を定義します。デフォルト値は **null** です。
- **http.proxyUser**: HTTP プロキシ名を定義します。
- **http.proxyPassword**: HTTP プロキシのユーザーパスワードを定義します。
- **org.uberfire.nio.git.http.enabled**: HTTP デモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.http.host**: このデーモンは、HTTP デモンが有効な場合にホストの識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.hostname**: HTTP デモンが有効な場合に、このデーモンはホスト名の識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.port**: このデーモンは、HTTP デモンが有効な場合にポート番号としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **8080** です。

### HTTPS 経由の Git

次のプロパティを使用して、HTTPS 経由で Git リポジトリーにアクセスできるように設定します。

- **org.uberfire.nio.git.proxy.ssh.over.https**: SSH が HTTPS プロキシを使用するかどうかを指定します。デフォルト値は **false** です。



- **https.proxyHost**: HTTPS プロキシのホスト名。デフォルト値は **null** です。
- **https.proxyPort**: HTTPS プロキシのポート (整数値)。デフォルト値は **null** です。
- **https.proxyUser**: HTTPS プロキシ名を定義します。
- **https.proxyPassword**: HTTPS プロキシのユーザーパスワードを定義します。
- **user.dir**: ユーザーディレクトリーの場所。
- **org.uberfire.nio.git.https.enabled**: HTTPS デーモンを有効または無効にします。デフォルト値は **false** です。
- **org.uberfire.nio.git.https.host**: このデーモンは、HTTPS デーモンが有効な場合にホストの識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.hostname**: このデーモンは、HTTPS デーモンが有効な場合にホスト名の識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.port**: このデーモンは、HTTPS デーモンが有効な場合にポート番号としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **8080** です。

## JGit

- **org.uberfire.nio.jgit.cache.instances**: JGit キャッシュサイズを定義します。
- **org.uberfire.nio.jgit.cache.overflow.cleanup.size**: JGit キャッシュオーバーフローのクリーンアップサイズを定義します。
- **org.uberfire.nio.jgit.remove.eldest.iterations**: 最も古い JGit の反復を削除するかどうかを定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.duration**: JGit 退避のしきい値の期間を定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.time.unit**: JGit 退避のしきい値の時間単位を定義します。

## Git デーモン

次のプロパティを使用して、Git デーモンを有効にして設定します。

- **org.uberfire.nio.git.daemon.enabled**: Git デーモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.daemon.host**: Git デーモンが有効な場合は、このプロパティをローカルホストの識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.daemon.hostname**: Git デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値は **localhost** です。

- **org.uberfire.nio.git.daemon.port**: Git デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値は **9418** です。
- **org.uberfire.nio.git.http.sslVerify**: Git リポジトリを確認する SSL 証明書を有効または無効にします。デフォルト値は **true** です。



### 注記

デフォルトポートまたは割り当てられたポートがすでに使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

## Git SSH

次のプロパティを使用して、Git SSH デーモンを有効にして設定します。

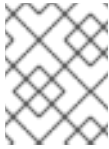
- **org.uberfire.nio.git.ssh.enabled**: SSH デーモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.ssh.host**: SSH デーモンが有効な場合は、このプロパティをローカルホスト識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.ssh.hostname**: SSH デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.SSH.port**: SSH デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値は **8001** です。



### 注記

デフォルトポートまたは割り当てられたポートがすでに使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

- **org.uberfire.nio.git.ssh.cert.dir**: ローカルの証明書が保存される **.security** ディレクトリーの場所。デフォルトは作業ディレクトリーです。
- **org.uberfire.nio.git.ssh.idle.timeout**: SSH のアイドルタイムアウトを設定します。
- **org.uberfire.nio.git.ssh.passphrase**: SCP スタイルの URL を持つ Git リポジトリのクローンを作成する場合に、オペレーティングシステムの公開キーストアにアクセスするためのパスフレーズ。たとえば、**git@github.com:user/repository.git** です。
- **org.uberfire.nio.git.ssh.algorithm**: SSH で使用されるアルゴリズム。デフォルト値は **RSA** です。
- **org.uberfire.nio.git.gc.limit**: GC の制限を設定します。
- **org.uberfire.nio.git.ssh.ciphers**: コンマ区切りの暗号化の文字列。利用可能な暗号化は **aes128-ctr**、**aes192-ctr**、**aes256-ctr**、**arcfour128**、**arcfour256**、**aes192-cbc**、**aes256-cbc** です。このプロパティを使用しない場合は、すべての暗号化が読み込まれます。
- **org.uberfire.nio.git.ssh.macs**: コンマ区切りのメッセージ認証コード (MAC) の文字列。利用可能な MAC は **hmac-md5**、**hmac-md5-96**、**hmac-sha1**、**hmac-sha1-96**、**hmac-sha2-256**、**hmac-sha2-512** です。このプロパティを使用しない場合は、すべての MAC が読み込まれます。



## 注記

RSA、または DSA 以外のアルゴリズムを使う場合は、Bouncy Castle JCE ライブラリーを使用するようにアプリケーションサーバーを設定します。

### KIE Server ノードおよび Process Automation Manager コントローラー

以下のプロパティを使用して Process Automation Manager コントローラーから KIE Server ノードへの接続を設定します。

- **org.kie.server.controller**: この URL は Process Automation Manager コントローラーへの接続に使用されます。たとえば、**ws://localhost:8080/business-central/websocket/controller** などです。
- **org.kie.server.user**: Process Automation Manager コントローラーから KIE Server ノードへの接続時に使用するユーザー名。このプロパティは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。
- **org.kie.server.pwd**: Process Automation Manager コントローラーから KIE Server ノードに接続する際に使用するパスワード。このプロパティは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。

### Maven など

以下のプロパティを使用して、Maven などの機能を設定します。

- **kie.maven.offline.force**: Maven のオフライン動作を強制します。true に設定すると、オンラインの依存関係解決が無効になります。デフォルト値は **false** です。



## 注記

このプロパティは、Business Central にのみ使用してください。他のコンポーネントとランタイム環境を共有する場合は、設定を分離して、Business Central にだけ適用してください。

- **org.uberfire.gzip.enable**: **GzipFilter** 圧縮フィルターで Gzip の圧縮を有効にするか、無効にします。デフォルト値は **true** です。
- **org.kie.workbench.profile**: Business Central プロファイルを選択します。許容値は、**FULL** または **PLANNER\_AND\_RULES** です。プリフィックス **FULL\_** で、プロファイルを設定し、管理者設定にこのプロファイルの設定が表示されないようにします。デフォルト値は **FULL** です。
- **org.appformer.m2repo.url**: Business Central は依存関係を検索する時に、Maven リポジトリのデフォルトの場所を使用します。デフォルト値は、**http://localhost:8080/business-central/maven2** など、Business Central 内の Maven リポジトリを参照します。このプロパティは、Business Central が起動する前に設定してください。デフォルト値は、内部の **m2** リポジトリへのファイルパスです。
- **appformer.ssh.keystore**: クラス名を指定して Business Central で使用する、カスタムの SSH キーストアを定義します。このプロパティが指定されていない場合はデフォルトの SSH キーストアを使用します。
- **appformer.ssh.keys.storage.folder**: このプロパティは、デフォルトの SSH キーストアを使用する場合にユーザーの SSH 公開鍵の保存フォルダーを定義します。このプロパティを指定しないと、この公開鍵は Business Central の **.security** フォルダーに保存され

ます。

- **appformer.experimental.features**: 実験的機能のフレームワークを有効にします。デフォルト値は **false** です。
- **org.kie.demo**: GitHub 外部へのデモアプリケーションのクローン作成を有効にします。
- **org.uberfire.metadata.index.dir**: Lucene の **.index** ディレクトリーが保存される場所。デフォルトは作業ディレクトリーです。
- **org.uberfire ldap.regex.role\_mapper**: LDAP プリンシパル名をアプリケーションのロール名にマッピングするのに使用する regex パターン。プリンシパルの値とロール名が一致する場合は、アプリケーションのロール名が変数ロールに置き換えられるため、変数ロールはパターンの一部でなければならない点に注意してください。
- **org.uberfire.sys.repo.monitor.disabled**: 設定モニターを無効にします。無効にした場合の影響を正しく理解していない場合は、無効にしないでください。デフォルト値は **false** です。
- **org.uberfire.secure.key**: パスワードの暗号化で使用するパスワード。デフォルト値は **org.uberfire.admin** です。
- **org.uberfire.secure.alg**: パスワードの暗号化で使用する暗号化アルゴリズム。デフォルト値は **PBEWithMD5AndDES** です。
- **org.uberfire.domain**: uberfire が使用するセキュリティドメイン名。デフォルト値は **ApplicationRealm** です。
- **org.guvnor.m2repo.dir**: Maven リポジトリーディレクトリーが保存される場所。デフォルト値は **<working-directory>/repositories/kie** です。
- **org.guvnor.project.gav.check.disabled**: グループ ID、アーティファクト ID、およびバージョン (GAV) のチェックを無効にします。デフォルト値は **false** です。
- **org.kie.build.disable-project-explorer**: Project Explorer で選択したプロジェクトの自動ビルドを無効にします。デフォルト値は **false** です。
- **org.kie.builder.cache.size**: プロジェクトビルダーのキャッシュサイズを定義します。デフォルト値は **20** です。
- **org.kie.library.assets\_per\_page**: プロジェクト画面のページごとのアセット数をカスタマイズできます。デフォルト値は **15** です。
- **org.kie.verification.disable-dtable-realtime-verification**: デシジョンテーブルのリアルタイム確認および検証を無効にします。デフォルト値は **false** です。

## Process Automation Manager コントローラー

以下のプロパティーを使用して、Process Automation Manager コントローラーへの接続方法を設定します。

- **org.kie.workbench.controller**: Process Automation Manager コントローラーとの接続に使用する URL。例: **ws://localhost:8080/kie-server-controller/websocket/controller**
- **org.kie.workbench.controller.user**: Process Automation Manager コントローラーのユーザー。デフォルト値は **kieserver** です。

- **org.kie.workbench.controller.pwd**: Process Automation Manager コントローラーのパスワード。デフォルト値は **kieserver1!** です。
- **org.kie.workbench.controller.token**: Process Automation Manager コントローラーとの接続に使用するトークン文字列

### Java Cryptography Extension KeyStore (JCEKS)

JCEKS を設定するには、以下のプロパティを使用します。

- **kie.keystore.keyStoreURL**: Java Cryptography Extension KeyStore (JCEKS) の読み込みに使用する URL。たとえば、**file:///home/kie/keystores/keystore.jceks** です。
- **kie.keystore.keyStorePwd**: JCEKS に使用するパスワード。
- **kie.keystore.key.ctrl.alias**: デフォルトの REST Process Automation Manager コントローラーに使用するキーのエイリアス。
- **kie.keystore.key.ctrl.pwd**: デフォルトの REST Process Automation Manager コントローラーのエイリアスのパスワード

### レンダリング

以下のプロパティを使用して、Business Central と KIE Server のレンダリングフォームを切り替えます。

- **org.jbpm.wb.forms.renderer.ext**: Business Central と KIE Server のフォームのレンダリングを切り替えます。デフォルトでは、フォームのレンダリングは Business Central が行います。デフォルト値は **false** です。
- **org.jbpm.wb.forms.renderer.name**: Business Central と KIE Server のレンダリングフォームを切り替えることができます。デフォルト値は **workbench** です。

## 第25章 関連情報

- [Red Hat JBoss EAP クラスター環境への Red Hat Process Automation Manager のインストールおよび設定](#)

---

## 第26章 次のステップ

- [Red Hat Process Automation Manager と Red Hat シングルサインオンの統合](#)
- [プロセスサービススタートガイド](#)

## パート III. RED HAT JBOSS EAP クラスター環境への RED HAT PROCESS AUTOMATION MANAGER のインストールおよび設定

システムエンジニアは、Red Hat Process Automation Manager クラスター環境を作成して、開発環境およびランタイム環境に高可用性および負荷分散を提供できます。

### 前提条件

- [Red Hat Process Automation Manager インストールの計画](#)の内容を確認済みである。



## 第27章 RED HAT PROCESS AUTOMATION MANAGER クラスター

2台以上のコンピューターをクラスタリングすると、高可用性、コラボレーションの強化、負荷分散の利点があります。高可用性により、1台のコンピューターで障害が発生したときにデータが損失する可能性を減らすことができます。その障害が発生したコンピューターにあるデータのコピーを提供することで、コンピューターに障害が発生したときに、別のコンピューターが不足を補います。障害が発生したコンピューターが再度オンラインになったら、クラスターに戻ります。

Red Hat Process Automation Manager コンポーネントをクラスタリングを行う方法は複数あります。本書は、以下のシナリオにおけるクラスタリングの方法を説明します。

- [28章 開発\(オーサリング\) 環境における Red Hat Process Automation Manager クラスター](#)
- [29章 ランタイム環境における KIE Server クラスター](#)

## 第28章 開発 (オーサリング) 環境における RED HAT PROCESS AUTOMATION MANAGER クラスター

開発者は、Red Hat Process Automation Manager を使用して、ユーザーの意思決定をサポートするルールとプロセスを作成できます。

Red Hat Process Automation Manager をクラスター開発環境として設定すると、高可用性の利点が得られます。クラスター環境では、開発者がノードで作業していて、そのノードで障害が発生した場合、この開発者が作業した内容はクラスターの別のノードに保存され、そちらで確認できます。

多くの開発環境には、ルールとプロセスを作成する Business Central と、このルールとプロセスをテストする 1 台以上の KIE Server があります。

Red Hat Process Automation Manager のクラスター開発環境を構築するには、以下のタスクを実行する必要があります。

1. クラスターのノードとなる各システムに以下のコンポーネントを設定します。
  - Red Hat JBoss EAP 7.4 with Red Hat Data Grid 8.1 を設定します。  
Red Hat Data Grid は Infinispan のオープンソースソフトウェアプロジェクトで構築されています。Red Hat Data Grid は、インデックス化の機能が含まれた、分散型インメモリーキー/値のデータストアで、大量のデータを素早くほぼリアルタイムに保存、検索、および解析できます。Red Hat Process Automation Manager のクラスター環境では、クラスターノード全体にわたる複雑な検索を効率的に実施できます。
  - Java メッセージングサーバー (JMS) ブローカーである、AMQ Broker を設定します。  
JMS ブローカーは、ローカルでメッセージを受信して保存し、そのメッセージを受信者に転送するソフトウェアコンポーネントです。AMQ Broker を使用すると、アプリケーションがメッセージングプロバイダーと通信できます。また、メッセージ駆動型 Bean、Enterprise JavaBean、servlet などのコンポーネントがどのようにメッセージを送受信するかを指定します。
  - NFS ファイルサーバーを設定します。
2. Red Hat JBoss EAP 7.4 および Red Hat Process Automation Manager 7.13 をダウンロードして、各システムにインストールします。各システムがクラスターノードになります。
3. クラスターの各ノードで Business Central を設定して起動します。

### 28.1. RED HAT DATA GRID のインストールおよび設定

クラスターノード全体でより効率的な検索を行うために、Red Hat Process Automation Manager のクラスター化環境に Red Hat Data Grid をインストールして設定します。

Red Hat Data Grid の高度なインストールおよび設定オプション、ならびに Red Hat JBoss EAP の Red Hat Data Grid モジュールに関する情報は、[Red Hat Data Grid Server Guide](#) を参照してください。



#### 注記

Business Central と同じノードに Red Hat Data Grid はインストールしないでください。

#### 前提条件

- Java 8.0 以降と互換性のある Java Virtual Machine (JVM) 環境がインストールされている。

## 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **Product:** Data Grid
  - **Version:** 8.1
2. **Red Hat Data Grid 8.1.0 Server(redhat-datagrid-8.1.0-server.zip)** のインストールファイルをダウンロードして、システム上の任意の場所に展開します。以下の例では、デプロイメントしたディレクトリーは **JDG\_HOME** と呼ばれます。
3. Red Hat Data Grid を最新のバージョンに更新します。詳細は、[Red Hat Data Grid User Guide](#) を参照してください。
4. Red Hat Data Grid のユーザーを追加するには、**JDG\_HOME/bin** に移動し、以下のコマンドを入力します。

```
$ ./cli.sh user create <DATAGRID_USER_NAME> -p <DATA_GRID_PASSWORD> -r default
```

5. Red Hat Data Grid サーバーのメモリーパラメーターを変更するには、**JDG\_HOME/bin/server.conf** ファイルを開き、以下の行を見つけます。

```
-Xms64m -Xmx512m -XX:MetaspaceSize=64M
```

6. この行を以下の内容に置き換えます。

```
-Xms256m -Xmx2048m -XX:MetaspaceSize=256M
```

7. **JDG\_HOME/server/conf/infinispan.xml** ファイルを開き、以下の行を見つけます。

```
<hotrod-connector name="hotrod"/>
```

8. この行を以下の内容に置き換えます。

```
<hotrod-connector name="hotrod">
  <authentication>
    <sasl mechanisms="SCRAM-SHA-512 SCRAM-SHA-384 SCRAM-SHA-256
      SCRAM-SHA-1 DIGEST-SHA-512 DIGEST-SHA-384
      DIGEST-SHA-256 DIGEST-SHA DIGEST-MD5 PLAIN"
      server-name="infinispan"
      qop="auth"/>
  </authentication>
</hotrod-connector>
```

9. Red Hat Data Grid を実行するには、**JDG\_HOME** に移動して以下のコマンドを入力します。

```
$ ./server.sh -b <HOST>
```

**<HOST>** は、Red Hat Data Grid をインストールしたサーバーの IP アドレスまたはホスト名に置き換えます。

## 28.2. AMQ BROKER のダウンロードおよび設定

Red hat AMQ Broker により、アプリケーションがメッセージングプロバイダーに接続できるようになります。また、メッセージ駆動型 Bean、Enterprise JavaBean、servlet などのコンポーネントがどのようにメッセージを送受信するかを指定します。

高度なインストールおよび設定オプションの詳細は、[Getting started with AMQ Broker](#) を参照してください。

### 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - Product: AMQ Broker
  - Version: 7.8.2
2. Red Hat AMQ Broker 7.8.2([amq-broker-7.8.2-bin.zip](#)) の横の **Download** をクリックします。
3. [amq-broker-7.8.2-bin.zip](#) ファイルをデプロイメントします。
4. [amq-broker-7.8.2-bin/amq-broker-7.8.2/bin](#) のディレクトリーに移動します。
5. 以下のコマンドを入力し、ブローカーおよびブローカーユーザーを作成します。

```
./artemis create --host <HOST> --user <AMQ_USER> --password <AMQ_PASSWORD> --require-login <BROKER_NAME>
```

上記の例で、以下のプレースホルダーを置き換えてください。

- **<HOST>** は、AMQ Broker をインストールしたサーバーの IP アドレスまたはホスト名に置き換えます。
  - **<AMQ\_USER>** および **<AMQ\_PASSWORD>** は、任意のユーザー名とパスワードの組み合わせに置き換えます。
  - **<BROKER\_NAME>** は作成するブローカーの名前に置き換えます。
6. AMQ Broker を実行するには、[amq-broker-7.8.2-bin/amq-broker-7.8.2/bin](#) ディレクトリーで以下のコマンドを入力します。

```
<BROKER_NAME>/bin/artemis run
```

## 28.3. NFS バージョン 4 サーバーの設定

Business Central クラスター環境には、NFS バージョン 4 サーバーの共有ファイルシステムが必要で、その共有ファイルシステムに、各クラスターノードからアクセスできる必要がある。

### 手順

1. NFS バージョン 4 共有をエクスポートするようにサーバーを設定します。Red Hat Enterprise Linux での NFS 共有のエクスポートの方法については、[ファイルシステムの管理](#) の **NFS 共有のエクスポート** を参照してください。NFS サーバーの作成に関する情報は、[RHEL 7 で NFS を設定する](#) を参照してください。

2. テキストエディターでこのサーバーの `/etc/exports` ファイルを開きます。
3. `/etc/exports` ファイルに以下の行を追加します。<HOST\_LIST> は、サーバーへの接続が許可されているホストのスペース区切りリストに置き換えます。

```
/opt/kie/data <HOST_LIST>
```

以下に例を示します。

```
/opt/kie/data 192.168.1.0/24(rw,sync) 192.168.1.1/24(no_root_squash)
```

これにより、NFS に必要となる最小オプション `rw,sync,no_root_squash` を使用して `/opt/kie/data` 共有が作成されます。



#### 注記

`/opt/kie/data` の代わりに別の共有名を使用できます。別の共有名を使用する場合には、Business Central を実行する全ノードの設定時に、この名前を使用する必要があります。

## 28.4. RED HAT JBOSS EAP 7.4 および RED HAT PROCESS AUTOMATION MANAGER のダウンロードおよびデプロイメント

クラスタの各ノードで Red Hat JBoss EAP 7.4 および Red Hat Process Automation Manager 7.13 をダウンロードして、インストールします。

### 手順

1. クラスタの各ノードに Red Hat JBoss EAP 7.4 をインストールします。
  - a. Red Hat カスタマーポータル [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
    - **Product: Enterprise Application Platform**
    - **バージョン: 7.4**
  - b. Red Hat JBoss Enterprise Application Platform 7.4.14の横にある **Download** をクリックします。 (**JBEAP-7.4.14/jboss-eap-7.4.14.zip**)。
2. **jboss-eap-7.4.14.zip** ファイルをデプロイメントします。以下の手順で **EAP\_HOME** は、**jboss-eap-7.4/jboss-eap-7.4** のディレクトリーです。
3. 最新の Red Hat JBoss EAP パッチが利用できる場合には、ダウンロードして適用します。
4. クラスタの各ノードで Red Hat Process Automation Manager をダウンロードします。
  - a. Red Hat カスタマーポータル [Software Downloads](#) ページに移動し、ドロップダウンオプションから製品およびバージョンを選択します。
    - **PRODUCT: Process Automation Manager**
    - **バージョン: 7.13.5**

- b. **Red Hat Process Automation Manager 7.13.5 Business Central Deployable for Red Hat JBoss EAP 7 (rhpmam-7.13.5-business-central-eap7-deployable.zip)** をダウンロードします。
5. **rhpmam-7.13.5-business-central-eap7-deployable.zip** ファイルを一時ディレクトリーに展開します。以下のコマンドでは、このディレクトリーを **TEMP\_DIR** とします。
6. **TEMP\_DIR/rhpmam-7.13.5-business-central-eap7-deployable/jboss-eap-7.4** の内容を **EAP\_HOME** にコピーします。
7. 最新の Red Hat Process Automation Manager のパッチが利用できる場合には、ダウンロードして適用します。
8. 高可用性環境向けに Red Hat Single Sign-On を設定します。詳細は、[Red Hat Process Automation Manager と Red Hat Single Sign-On の統合](#) および [Red Hat Single Sign-On サーバー管理ガイド](#) を参照してください。

## 28.5. クラスターでの BUSINESS CENTRAL の設定および実行

Red Hat JBoss EAP と Business Central のインストール後に、Red Hat Data Grid と AMQ Broker を使用してクラスターを設定できます。クラスターの各ノードで以下の手順を実行します。



### 注記

この手順では、基本的なクラスター設定を説明します。より詳細な設定は [Red Hat JBoss EAP 7.4 設定ガイド](#) を参照してください。



### 注記

高可用性 (HA) のオンプレミス環境で KIE Server を Business Central に接続しないでください。

Business Central インスタンスは、各 KIE Server のステータスと同期し続けることができません。たとえば、KIE Server が稼働中だが同期していない場合、Business Central はそのインスタンスからデプロイできなくなります。

### 前提条件

- 「[Red Hat Data Grid のインストールおよび設定](#)」の説明通りに、Red Hat Data Grid 8.1 がインストールされている。
- 「[AMQ Broker のダウンロードおよび設定](#)」の説明通りに AMQ Broker がインストールされ、設定されている。
- 「[Red Hat JBoss EAP 7.4 および Red Hat Process Automation Manager のダウンロードおよびデプロイメント](#)」の説明通りに、クラスターの各ノードに Red Hat JBoss EAP および Red Hat Process Automation Manager がインストールされている。
- 「[NFS バージョン 4 サーバーの設定](#)」で記載されているように、共有ディレクトリーを備えた NFS サーバーを利用できる。

### 手順

1. NFS で共有されているディレクトリーを **/data** としてマウントします。Root ユーザーで以下のコマンドを入力します。

```
mkdir /data
mount <NFS_SERVER_IP>:<DATA_SHARE> /data
```

<NFS\_SERVER\_IP> は、NFS サーバシステムの IP アドレスまたはホスト名に置き換えてください。<DATA\_SHARE> は、設定した共有名 (例: /opt/kie/data) に置き換えます。

2. /data NFS ディレクトリーに **kie-wb-playground** ディレクトリーを作成します。

```
mkdir /kie-wb-playground
```

3. **EAP\_HOME/bin** ディレクトリーに **kie-wb-playground** ディレクトリーを作成し、ディレクトリーをマウントします。

```
mount -o rw,sync,actimeo=1 <NFS_SERVER_IP>:<DATA_SHARE>/kie-wb-playground kie-wb-playground
```

4. テキストエディターで **EAP\_HOME/standalone/configuration/standalone-full.xml** ファイルを開きます。

5. <system-properties> 要素でプロパティを編集または追加し、以下のプレースホルダーを置き換えます。

- <AMQ\_USER> および <AMQ\_PASSWORD> は、AMQ Broker の作成時に定義した認証情報に置き換えます。
- <AMQ\_BROKER\_IP\_ADDRESS> は AMQ Broker の IP アドレスに置き換えます。
- <DATA\_GRID\_NODE\_IP> は、Red Hat Data Grid のインストール先の IP アドレスに置き換えます。
- <SERVER\_NAME> は、Red Hat Data Grid のサーバ設定で指定したサーバ名に置き換えます。
- <SASL\_QOP> は、Red Hat Data Grid サーバ設定の **auth**、**auth-int**、**auth-conf** の値の組み合わせです。
- <DATAGRID\_USER\_NAME> および <DATA\_GRID\_PASSWORD> は、Red Hat Data Grid の作成時に定義した認証情報です。

```
<system-properties>
  <property name="appformer-jms-connection-mode" value="REMOTE"/>
  <property name="appformer-jms-username" value="<AMQ_USER>"/>
  <property name="appformer-jms-password" value="<AMQ_USER_PASSWORD>"/>
  <property name="appformer-jms-url"
    value="tcp://<AMQ_BROKER_IP_ADDRESS>:61616?
    ha=true&retryInterval=1000&retryIntervalMultiplier=1.0&reconnectAttempts=
    -1"/>
  <property name="org.appformer.ext.metadata.infinispan.port"
    value="11222"/>
  <property name="org.appformer.ext.metadata.infinispan.host"
    value="<DATA_GRID_NODE_IP>"/>
  <property name="org.appformer.ext.metadata.infinispan.realm"
    value="ApplicationRealm"/>
  <property name="org.appformer.ext.metadata.infinispan.cluster"
    value="kie-cluster"/>
```

```

<property name="org.appformer.ext.metadata.index"
  value="infinispan"/>
<property name="org.uberfire.nio.git.dir"
  value="/data"/>
<property name="es.set.netty.runtime.available.processors"
  value="false"/>
<property name="org.appformer.ext.metadata.infinispan.username" value="
<DATAGRID_USER_NAME>"/>
<property name="org.appformer.ext.metadata.infinispan.password" value="
<DATA_GRID_PASSWORD>"/>
<property name="org.appformer.ext.metadata.index" value="infinispan"/>
<property name="org.appformer.ext.metadata.infinispan.sasl.qop" value="auth"/>
<property name="org.appformer.ext.metadata.infinispan.server.name"
value="infinispan"/>
<property name="org.appformer.ext.metadata.infinispan.realm" value="default"/>
<property name="org.appformer.concurrent.managed.thread.limit" value="1000"/>
<property name="org.appformer.concurrent.unmanaged.thread.limit" value="1000"/>
<property name="org.appformer.concurrent.indexing.thread.limit" value="0"/>
<property name="org.appformer.ext.metadata.infinispan.server.name"
  value="<SERVER_NAME>"/>
<property name="org.appformer.ext.metadata.infinispan.sasl.qop"
  value="<SASL_QOP>"/>
</system-properties>

```

6. **standalone-full.xml** ファイルを保存します。

7. クラスタを起動するには **EAP\_HOME/bin** に移動して、以下のコマンドを入力します。

```
$ ./standalone.sh -c standalone-full.xml -b <HOST>
```

**<HOST>** は、Red Hat Process Automation Manager をインストールしたサーバーの IP アドレスまたはホスト名に置き換えます。

## 28.6. 高可用性 (HA) オンプレミスインフラストラクチャーのテスト

Business Central の実稼働環境可能な高可用性 (HA) オンプレミスインフラストラクチャーを作成する場合は、実行可能な HA 環境のハードウェアおよびパフォーマンスの最低要件を満たしていることを確認する必要があります。HA オンプレミスインフラストラクチャーは、Business Central、メッセージシステム (AMQ)、インデックスサーバー (Red Hat Data Grid)、および共有ファイルシステム (NFS/GlusterFS/Ceph) の 4 つの主要コンポーネントで設定されます。

### 前提条件

- 3 つ以上のノードのネットワーク環境は、以下のレイアウトで設定される。  
ノード 1: Business Central  
  
ノード 2: Business Central  
  
ノード 3: AMQ、Red Hat Data Grid、および NFS

### 手順

- ネットワーク速度をテストします。
  - 各サーバーノードのコマンド端末で、**iPerf3** をインストールします。



```
$ dnf install iperf3
```

- b. NFS サーバーノード (サーバーノード 3) のコマンドターミナルで、**iPerf3** をサーバーモードで起動します。

```
$ iperf3 -s
```

- c. 各 Business Central サーバーノードのコマンド端末で以下のコマンドを入力して、ホストとして NFS サーバーノードを設定して、クライアントモードで **iPerf3** を起動します。

```
$ iperf3 -c <NFS_SERVER_IP>
```

```
+
```

In this example, replace ``<NFS_SERVER_IP>`` with the IP address of the NFS server.

- d. 各サーバーノードの結果を以下の最小値の例と比較します。

```
iperf3 -c 172.31.47.103
Connecting to host 172.31.47.103, port 5201
[ 5] local 172.31.39.4 port 44820 connected to 172.31.47.103 port 5201
[ ID] Interval      Transfer  Bitrate    Retr  Cwnd
[ 5] 0.00-1.00 sec  143 MBytes  1.20 Gbits/sec  0  419 KBytes
[ 5] 1.00-2.00 sec  111 MBytes  928 Mbits/sec  6  848 KBytes
[ 5] 2.00-3.00 sec  53.8 MBytes  451 Mbits/sec  0  1.08 MBytes
[ 5] 3.00-4.00 sec  52.5 MBytes  440 Mbits/sec  1  1022 KBytes
[ 5] 4.00-5.00 sec  53.8 MBytes  451 Mbits/sec  1  935 KBytes
[ 5] 5.00-6.00 sec  53.8 MBytes  451 Mbits/sec  1  848 KBytes
[ 5] 6.00-7.00 sec  52.5 MBytes  440 Mbits/sec  0  1.08 MBytes
[ 5] 7.00-8.00 sec  53.8 MBytes  451 Mbits/sec  1  1.01 MBytes
[ 5] 8.00-9.00 sec  53.8 MBytes  451 Mbits/sec  1  953 KBytes
[ 5] 9.00-10.00 sec  52.5 MBytes  440 Mbits/sec  1  856 KBytes
-----
[ ID] Interval      Transfer  Bitrate    Retr
[ 5] 0.00-10.00 sec  680 MBytes  570 Mbits/sec  12      sender
[ 5] 0.00-10.04 sec  677 MBytes  566 Mbits/sec           receiver

iperf Done.
```

2. NFS 情報を確認します。

- a. 各 Business Central サーバーノードのコマンド端末で、NFS ノードをマウントします。

```
$ mount -o actimeo=1 <NFS_SERVER_IP>:/opt/nfs/kie /opt/kie/niogit
```

- b. マウントされた各ノードのコマンド端末で、**nfsiostat** と入力します。

```
$ nfsiostat
```

- c. 各サーバーノードの結果を以下の最小値の例と比較します。

```
nfsiostat
ops/s rpc bklog
6.415 0.000
```

```

read:
ops/s kB/s kB/op retrans avg RTT (ms) avg exe (ms) avg queue (ms) errors
0.031 0.045 1.452 0 (0.0%) 0.129 0.166 0.019 0 (0.0%)

write:
ops/s kB/s kB/op retrans avg RTT (ms) avg exe (ms) avg queue (ms) errors
0.517 0.467 0.903 0 (0.0%) 1.235 1.269 0.018 0 (0.0%)

```

3. ディスクが SSD であることを確認します。

- a. NFS サーバーのコマンド端末で、**df -h** を入力して、以下の例で示すようにディスクを特定します。

```

$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G  0 3.8G  0% /dev
tmpfs           3.9G  0 3.9G  0% /dev/shm
tmpfs           3.9G 33M 3.8G  1% /run
tmpfs           3.9G  0 3.9G  0% /sys/fs/cgroup
/dev/xvda2      25G  3.2G 22G 13% /
tmpfs           781M  0 781M  0% /run/user/1000
172.31.47.103:/root/nfs 25G  2.1G 23G  9% /root/nfs

```

- b. **lsblk -d** を入力して、ディスクが SSD であることを確認します。

```
$ lsblk -d
```

- c. **hdparm -Tt** と入力して、ディスクをテストします。

```
$ hdparm -Tt /dev/xvda2
```

- d. 各サーバーノードの結果を以下の最小値の例と比較します。

```

$ hdparm -Tt /dev/xvda2

/dev/xvda2:
Timing cached reads: 18670 MB in 1.99 seconds = 9389.01 MB/sec
Timing buffered disk reads: 216 MB in 3.03 seconds = 71.40 MB/sec

```

## 28.7. RED HAT PROCESS AUTOMATION MANAGER クラスターの検証

Red Hat Process Automation Manager を設定したら、アセットを作成してシステムが機能していることを検証します。

### 手順

1. Web ブラウザーで、**<node-IP-address>:8080/business-central** を入力します。**<node-IP-address>** は特定のノードの IP アドレスに置き換えます。
2. インストール時に作成した **admin** ユーザーの認証情報を入力します。Business Central ホームページが表示されます。
3. **Menu** → **Design** → **Projects** の順に選択します。

4. **MySpace** スペースを開きます。
5. **Try Samples** → **Mortgages** → **OK** の順にクリックします。 **Assets** ウィンドウが表示されます。
6. **Add Asset** → **Data Object** をクリックします。
7. **Data Object** フィールドに **MyDataObject** と入力し、 **OK** をクリックします。
8. **Spaces** → **mySpace** → **Mortgages** の順にクリックし、アセットリストに **MyDataObject** があることを確認します。
9. Web ブラウザーに以下の URL を入力します。 **<node\_IP\_address>** には、クラスターの別のノードのアドレスに置き換えます。  
**http://<node\_IP\_address>:8080/business-central**
10. **MyDataObject** アセットを作成した最初のノードの Business Central にログインするときに使用した認証情報と同じものを入力します。
11. **Menu** → **Design** → **Projects** の順に選択します。
12. **MySpace** スペースを開きます。
13. **Mortgages** プロジェクトを選択します。
14. **MyDataObject** がアセットリストにあることを確認します。
15. **Mortgages** プロジェクトを選択します。

## 第29章 ランタイム環境における KIE SERVER クラスター

ランタイム環境では、KIE Server は、ビジネス上の決定をサポートするルールおよびプロセスが含まれるサービスを実行します。KIE Server ランタイム環境をクラスタリングする主な利点は負荷分散です。クラスターのノードの1つでアクティビティが増えると、そのアクティビティはクラスターの残りのノードと共有されるため、パフォーマンスが改善します。

KIE Server のクラスター化ランタイム環境を作成するには、Red Hat JBoss EAP 7.4 および KIE Server をダウンロードしてデプロイメントします。次に、クラスターに Red Hat JBoss EAP 7.4 を設定し、クラスターを起動し、各クラスターノードに KIE Server をインストールします。

任意で、ヘッドレス Process Automation Manager コントローラーおよび Smart Router をクラスター化できます。



### 注記

高可用性 (HA) のオンプレミス環境で KIE Server を Business Central に接続しないでください。

Business Central インスタンスは、各 KIE Server のステータスと同期し続けることができません。たとえば、KIE Server が稼働中だが同期していない場合、Business Central はそのインスタンスからデプロイできなくなります。

### 29.1. RED HAT JBOSS EAP 7.4 および KIE SERVER のダウンロードおよびデプロイメント

本セクションの手順を行い、Red Hat JBoss EAP 7.4 をダウンロードしてインストールし、クラスター環境にインストールするために KIE Server をダウンロードして再パッケージ化します。

#### 手順

1. クラスターの各ノードに Red Hat JBoss EAP 7.4 をインストールします。
  - a. Red Hat カスタマーポータル [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
    - **Product: Enterprise Application Platform**
    - **バージョン: 7.4**
  - b. Red Hat JBoss Enterprise Application Platform 7.4.14の横にある **Download** をクリックします。 (**jboss-eap-7.4.14.zip**)。
2. **jboss-eap-7.4.14.zip** ファイルをデプロイメントします。 **jboss-eap-7.4/jboss-eap-7.4** ディレクトリーは、以下のように参照されます。 **EAP\_HOME**
3. 最新の Red Hat JBoss EAP パッチが利用できる場合には、ダウンロードして適用します。
4. KIE Server のダウンロード:
  - a. Red Hat カスタマーポータル [Software Downloads](#) ページに移動し、ドロップダウンオプションから製品およびバージョンを選択します。
    - **PRODUCT: Process Automation Manager**
    - **バージョン: 7.13.5**

- b. Red Hat Process Automation Manager 7.13.5 KIE Server for All Supported EE8 Containers (**rhpm-7.13.5-kie-server-ee8.zip**) をダウンロードします。
5. **rhpm-7.13.5-kie-server-ee8.zip** アrchiveを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
6. **TEMP\_DIR/rhpm-7.13.5-kie-server-ee8/rhpm-7.13.5-kie-server-ee8/kie-server.war** ディレクトリーを **EAP\_HOME/standalone/deployments/** にコピーします。



#### 警告

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

7. **TEMP\_DIR/rhpm-7.13.5-kie-server-ee8/rhpm-7.13.5-kie-server-ee8/SecurityPolicy/** の内容を **EAP\_HOME/bin** にコピーします。
8. ファイルを上書きするように求められたら、**Replace** をクリックします。
9. **EAP\_HOME/standalone/deployments/** ディレクトリーに、**kie-server.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動すると KIE Server が自動的にデプロイされます。
10. 最新の Red Hat Process Automation Manager のパッチが利用できる場合には、ダウンロードして適用します。
11. **EAP\_HOME/bin** ディレクトリーに移動します。
12. KIE Server へのログインに使用する **kie-server** ロールのユーザーを作成します。

```
$ ./bin/jboss-cli.sh --commands="embed-server --stdout=echo,/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity(identity=<USERNAME>),/subsystem=elytron/filesystem-realm=ApplicationRealm:set-password(identity=<USERNAME>, clear={password='<PASSWORD>'}),/subsystem=elytron/filesystem-realm=ApplicationRealm:add-identity-attribute(identity=<USERNAME>, name=role, value=['kie-server'])"
```

13. ユーザー名とパスワードを書き留めておきます。

## 29.2. KIE SERVER 向けの RED HAT JBOSS EAP 7.4 クラスターの設定および実行

KIE Server 向けに Red Hat JBoss EAP クラスターを設定して、起動します。

### 手順

1. JDBC ドライバーを、このクラスターの一部となる Red Hat JBoss EAP の全インスタンスにインストールします。詳細は、Red Hat JBoss EAP 7.4 設定ガイドの [JDBC ドライバー セクション](#)を参照してください。

2. テキストエディターで **EAP\_HOME/standalone/configuration/standalone-full.xml** ファイルを開きます。
3. **data-stores** プロパティと、その上の **timer-service thread-pool-name** を編集します。
  - **datasource-jndi-name** は、この手順の最初で指定したデータベースの JNDI 名です。
  - **partition** プロパティの値にはあらゆる名前を入力できますが、ノードには、同じパーティション名を持つその他のノードのタイマーのみが表示されます。パーティション名を割り当てて、パーティションでノードをグループ分けすると、大規模なクラスターを複数の小規模クラスターに分割できます。これによりパフォーマンスが改善します。これを行うと、パフォーマンスが向上します。たとえば、ノードが100個含まれるクラスターで、ノード100個すべてが同じタイマーを実行して更新しようとする代わりに、ノード5個ずつのグループに異なるパーティション名を指定し、ノード5個ずつのクラスターを20個作成できます。
  - **default-data-store** 属性値を **ejb\_timer\_ds** に置き換えます。
  - **refresh-interval** の値をミリ秒で設定して、EJB タイマーがデータベースに接続して同期し、処理するタスクをロードする頻度を指定します。

```
<timer-service thread-pool-name="default" default-data-store="ejb_timer_ds">
<data-stores>
  <database-data-store name="ejb_timer_ds" datasource-jndi-
name="java:jboss/datasources/ejb_timer" database="postgresql"
partition="ejb_timer_part" refresh-interval="30000"/>
</data-stores>
</timer-service>
```

以下の表は、サポートされるデータベースと、対応する **database** 属性値を示しています。

表29.1 サポートされているデータベース

| データベース                            | 属性値               |
|-----------------------------------|-------------------|
| Hyper SQL (デモを目的としており、サポートはされません) | <b>hsqldb</b>     |
| PostgreSQL                        | <b>postgresql</b> |
| Oracle                            | <b>oracle</b>     |
| IBM DB2                           | <b>db2</b>        |
| Microsoft SQL Server              | <b>mssql</b>      |
| MySQL および MariaDB                 | <b>mysql</b>      |

4. KIE Server および EJB タイマーデータソースを **standalone-full.xml** ファイルに追加します。この例では、**<DATABASE>** はデータベース名、**<SERVER\_NAME>** は JNDI データベースのホスト名、**<USER\_NAME>** および **<USER\_PWD>** はそのデータベースの認証情報になります。

- 以下のように、データソースを追加して、KIE Server がデータベースに接続できるようにします。

```
<xa-datasource jndi-name="java:/jboss/datasources/rhpam" pool-name="rhpam-RHPAM"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>
```

- 以下のように、データソースを追加して、EJB タイマーを有効化します。

```
<xa-datasource jndi-name="java:/jboss/datasources/ejb_timer" pool-name="ejb_timer"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>
```



### 警告

KIE Server ランタイムデータおよび EJB タイマーデータに、別々のデータベースを使用する必要があります。

5. **<system-properties>** 要素に以下のプロパティを追加し、以下のプレースホルダーを置き換えます。
  - **<JNDI\_NAME>** は、データソースの JNDI 名です。Red Hat Process Automation Manager の場合は **java:/jboss/datasources/rhpam** です。
  - **<DIALECT>** は、データベースの hibernate ダイアレクトです。以下の方言がサポートされます。
    - DB2: **org.hibernate.dialect.DB2Dialect**

- MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
- MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
- MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
- Oracle: **org.hibernate.dialect.Oracle10gDialect**
- PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
- PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**

```
<system-properties>
  <property name="org.kie.server.persistence.ds" value="<JNDI_NAME>"/>
  <property name="org.kie.server.persistence.dialect" value="<DIALECT>"/>
  <property name="org.jbpm.ejb.timer.tx" value="true"/>
</system-properties>
```

6. **standalone-full.xml** ファイルを保存します。
7. クラスタを起動するには **EAP\_HOME/bin** に移動して、以下のコマンドの1つを入力します。
  - Linux または UNIX ベースのシステムの場合:

```
$ ./standalone.sh -c standalone-full.xml
```

- Windows の場合:

```
standalone.bat -c standalone-full.xml
```

### 29.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーを使用した KIE SERVER のクラスタリング

Process Automation Manager コントローラーは Business Central と統合します。ただし、Business Central をインストールしない場合は、ヘッドレス Process Automation Manager コントローラーをインストールし、REST API または KIE Server Java Client API を使用してそのコントローラーと対話します。

#### 前提条件

- バックアップを作成してある Red Hat JBoss EAP システム (バージョン 7.4 またはそれ以降) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP\_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。
- 「[NFS バージョン 4 サーバーの設定](#)」 で記載されているように、共有ディレクトリーを備えた NFS サーバーを利用できる。

#### 手順

1. Red Hat カスタマーポータル [の Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。



- **PRODUCT:** Process Automation Manager
  - バージョン: 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons(**rhpmam-7.13.5-add-ons.zip** ファイル)をダウンロードします。
  3. **rhpmam-7.13.5-add-ons.zip** ファイルを展開します。 **rhpmam-7.13.5-controller-ee7.zip** ファイルは展開したディレクトリーにあります。
  4. **rhpmam-7.13.5-controller-ee7.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
  5. **TEMP\_DIR/rhpmam-7.13.5-controller-ee7/controller.war** ディレクトリーを **EAP\_HOME/standalone/deployments/** にコピーします。



### 警告

コピーするヘッドレス Process Automation Manager コントローラーデプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

6. **TEMP\_DIR/rhpmam-7.13.5-controller-ee7/SecurityPolicy/** ディレクトリーの内容を **EAP\_HOME/bin** にコピーします。
7. ファイルの上書きを求めるプロンプトが出されたら、**Yes** をクリックします。
8. **EAP\_HOME/standalone/deployments/** ディレクトリーに、**controller.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動するとヘッドレス Process Automation Manager コントローラーが自動的にデプロイされます。
9. テキストエディターで **EAP\_HOME/standalone/configuration/standalone.xml** ファイルを開きます。
10. 以下のプロパティーを **<system-properties>** 要素に追加し、**<NFS\_STORAGE>** を、テンプレート設定が保存されている NFS ストレージへの絶対パスに置き換えます。

```
<system-properties>
  <property name="org.kie.server.controller.templatefile.watcher.enabled" value="true"/>
  <property name="org.kie.server.controller.templatefile" value="<NFS_STORAGE>"/>
</system-properties>
```

テンプレートファイルには、特定のデプロイメントシナリオのデフォルト設定が含まれます。

**org.kie.server.controller.templatefile.watcher.enabled** プロパティーの値を **true** に設定すると、別のスレッドが開始してテンプレートファイルの修正を監視します。この確認の間隔はデフォルトで 30000 ミリ秒になり、**org.kie.server.controller.templatefile.watcher.interval** システムプロパティーで制御できます。このプロパティーの値を **false** に設定すると、テンプレートファイルへの変更の検出が、サーバーの再起動時に制限されます。

11. ヘッドレス Process Automation Manager コントローラーを開始するには、**EAP\_HOME/bin** に移動して、以下のコマンドを実行します。

- Linux または UNIX ベースのシステムの場合:

```
┌ $ ./standalone.sh
```

- Windows の場合:

```
┌ standalone.bat
```

## 29.4. 失敗した KIE SERVER ノードの RUNNING 状態にあるジョブのフェイルオーバーの設定

デフォルトでは、クラスターで複数の KIE Server ノードを実行している場合は、1つの KIE Server ノードに障害が発生しても、その他の KIE Server ノードは、失敗した KIE Server ノードから **QUEUED** 状態または **RETRYING** 状態のジョブを選択します。ただし、これらのノードは **RUNNING** 状態のジョブを選択しません。

別の KIE Server ノードに再度キューに置かれるように、**RUNNING** 状態のジョブを組み込むようにクラスターを設定できます。障害が発生した KIE Server ノードのプロセスにタイマー開始ノードがある場合、プロセスはいずれかのノードでのみ開始します。

### 前提条件

- KIE Server に Red Hat JBoss EAP クラスターを設定している。

### 手順

1. 環境を準備します。
  - a. 以下のエクステンションが **standalone.xml** ファイルに存在することを確認し、必要な場合は追加します。

```
┌ <extension module="org.jboss.as.clustering.infinispan"/>
┌ <extension module="org.jboss.as.clustering.jgroups"/>
```

- b. **standalone.xml** ファイルで、**<subsystem xmlns="urn:jboss:domain:infinispan:9.0">** サブシステムを見つけ、以下の例のように、**nodes** という名前のキャッシュと **jobs** という名前のキャッシュを含む **jbpm** という名前のキャッシュコンテナエントリーを作成します。

```
┌ <cache-container name="jbpm">
┌ <transport lock-timeout="60000"/>
┌ <replicated-cache name="nodes">
┌ <transaction mode="BATCH"/>
┌ </replicated-cache>
┌ <replicated-cache name="jobs">
┌ <transaction mode="BATCH"/>
┌ </replicated-cache>
┌ </cache-container>
```

- c. レプリケートされたキャッシュと分散キャッシュの **jgroups** サブシステムを追加します。

```

<subsystem xmlns="urn:jboss:domain:jgroups:7.0">
  <channels default="ee">
    <channel name="ee" stack="udp" cluster="ejb"/>
  </channels>
  <stacks>
    <stack name="udp">
      <transport type="UDP" socket-binding="jgroups-udp"/>
      <protocol type="PING"/>
      <protocol type="MERGE3"/>
      <socket-protocol type="FD_SOCKET" socket-binding="jgroups-udp-fd"/>
      <protocol type="FD_ALL"/>
      <protocol type="VERIFY_SUSPECT"/>
      <protocol type="pbcast.NAKACK2"/>
      <protocol type="UNICAST3"/>
      <protocol type="pbcast.STABLE"/>
      <protocol type="pbcast.GMS"/>
      <protocol type="UFC"/>
      <protocol type="MFC"/>
      <protocol type="FRAG3"/>
    </stack>
    <stack name="tcp">
      <transport type="TCP" socket-binding="jgroups-tcp"/>
      <socket-protocol type="MPING" socket-binding="jgroups-mping"/>
      <protocol type="MERGE3"/>
      <socket-protocol type="FD_SOCKET" socket-binding="jgroups-tcp-fd"/>
      <protocol type="FD_ALL"/>
      <protocol type="VERIFY_SUSPECT"/>
      <protocol type="pbcast.NAKACK2"/>
      <protocol type="UNICAST3"/>
      <protocol type="pbcast.STABLE"/>
      <protocol type="pbcast.GMS"/>
      <protocol type="MFC"/>
      <protocol type="FRAG3"/>
    </stack>
  </stacks>
</subsystem>

```

- d. 以下のインターフェイスを設定します。

```

<interface name="private">
  <inet-address value="{jboss.bind.address.private:127.0.0.1}"/>
</interface>

```

- e. 以下のソケットを設定します。

```

<socket-binding name="jgroups-mping" interface="private" multicast-
address="{jboss.default.multicast.address:230.0.0.4}" multicast-port="45700"/>

<socket-binding name="jgroups-tcp" interface="private" port="7600"/>
<socket-binding name="jgroups-tcp-fd" interface="private" port="57600"/>
<socket-binding name="jgroups-udp" interface="private" port="55200" multicast-
address="{jboss.default.multicast.address:230.0.0.4}" multicast-port="45688"/>
<socket-binding name="jgroups-udp-fd" interface="private" port="54200"/>

```

2. Infinispan 拡張機能を KIE Server に追加します。

- a. テキストエディターで **kie-server.war/WEB-INF/jboss-deployment-structure.xml** ファイルを開き、以下の依存関係をファイルに追加します。

```
<module name="org.infinispan" services="export"/>
<module name="org.jgroups" />
```

- b. JAR ファイルをダウンロードします。
- i. Red Hat カスタマーポータルの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
    - **製品:** Process Automation Manager
    - **バージョン:** 7.13.5
  - ii. **Red Hat Process Automation Manager 7.13.5 Maven Repository(rhpam-7.13.5-maven-repository.zip)** をダウンロードし、ZIP ファイルを展開します。
  - iii. 展開したファイルが含まれるディレクトリーに移動し、**maven-repository/org/kie/server/kie-server-services-jbpm-cluster/7.67.0.Final-redhat-00024/kie-server-services-jbpm-cluster-7.67.0.Final-redhat-00024.jar** ファイルを **kie-server.war/WEB-INF/lib** ディレクトリーにコピーします。

クラスターは、**RUNNING** 状態のジョブにフェイルオーバーを使用するように設定されます。

## 第30章 SMART ROUTER のインストールおよび設定

Smart Router (KIE Server ルーター) は、複数の KIE Server、クライアントアプリケーション、他のコンポーネント間の統合レイヤーとして使用可能な軽量の Java コンポーネントです。デプロイメントや実行環境に合わせて、Smart Router は複数の独立した KIE Server インスタンスを単一サーバーのように集約できます。Smart Router には以下の機能があります。

### データ集約

クライアントアプリケーションの要求があると、全 KIE Server インスタンス (各グループからインスタンス1つ) からデータを収集し、結果を1つの応答にまとめます。

### ルーティング

クライアントアプリケーションからサービスへの呼び出しを受信して、各呼び出しを自動的に個別サービスを実行する KIE Server にルーティングする単一のエンドポイントとして機能します。つまり、KIE Server に同じサービスをデプロイする必要はありません。

### 負荷分散

負荷分散を効率化します。Smart Router クラスターの負荷分散要求は、標準負荷分散ツールを使用して外部で管理する必要があります。

### 認証

システムプロパティフラグを使用して KIE Server インスタンスを認証し、HTTPS トラフィックを有効にできます。

### 環境管理

環境の変更を管理します (例: サーバーインスタンスの追加または削除)

## 30.1. SMART ROUTER による KIE サーバーインスタンスのロードバランシング

Smart Router は、複数の独立した KIE Server インスタンスを集約して1台のサーバーのように使用できます。Smart Router は、個々の KIE Server インスタンスに要求を送信し、複数の KIE Server インスタンスからデータを集約できるため、インテリジェントなロードバランサーのロールを果たします。エイリアスを介して、Smart Router はプロキシとなります。

### 前提条件

- 複数の KIE サーバーインスタンスがインストールされている。



### 注記

Smart Router を使用するには、KIE Server をマネージド外として設定する必要はない。

管理されていない KIE サーバーインスタンスがコントローラーに接続されない。たとえば、マネージド外の KIE Server インスタンスを Smart Router に接続してコントローラーで Smart Router を登録する場合には、Business Central は Smart Router を使用してマネージド外の KIE Server インスタンスに問い合わせる。

### 手順

1. Red Hat カスタマーポータル [の Software Downloads ページ](#) に移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。

- **PRODUCT:** Process Automation Manager

- バージョン: 7.13.5

2. Red Hat Process Automation Manager 7.13.5 Add Onsをダウンロードします。
3. ダウンロードした **rhpam-7.13.5-add-ons.zip** ファイルを一時ディレクトリーに展開します。 **rhpam-7.13.5-smart-router.jar** ファイルは展開した **rhpam-7.13.5-add-ons** ディレクトリーにあります。
4. ファイルを実行する場所に **rhpam-7.13.5-smart-router.jar** ファイルをコピーします。
5. 以下のコマンドを入力して、Smart Router を起動します。

```
java
-Dorg.kie.server.router.host=<ROUTER_HOST>
-Dorg.kie.server.router.port=<ROUTER_PORT>
-Dorg.kie.server.controller=<CONTROLLER_URL>
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.router.config.watcher.enabled=true
-Dorg.kie.server.router.repo=<NFS_STORAGE>
-jar rhpam-7.13.5-smart-router.jar
```

上のコマンドのプロパティーのデフォルト値は、以下のようになります。

```
org.kie.server.router.host=localhost
org.kie.server.router.port=9000
org.kie.server.controller= N/A
org.kie.server.controller.user=kieserver
org.kie.server.controller.pwd=kieserver1!
org.kie.server.router.repo= <CURRENT_WORKING_DIR>
org.kie.server.router.config.watcher.enabled=false
```

**org.kie.server.controller** は、以下のように、サーバーコントローラーの URL です。

```
org.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

**org.kie.server.router.config.watcher.enabled** は、ウォッチャーサービスシステムプロパティーを有効にする任意の設定です。



### 注記

コマンドラインで設定プロパティーを指定する代わりに、設定ファイルを使用することができます。ファイルを使用した Smart Router の設定は、「[設定ファイルを使用した Smart Router の設定](#)」を参照してください。

6. Smart Router に接続する必要があるすべての KIE サーバーインスタンスでは、**org.kie.server.router** システムプロパティーに Smart Router の URL を設定します。
7. クライアント側から Smart Router にアクセスするには、以下のように、KIE Server の URL の代わりに Smart Router の URL を使用します。

```
KieServicesConfiguration config =
KieServicesFactory.newRestConfiguration("http://smartrouter.example.com:9000",
"USERNAME", "PASSWORD");
```

この例では、**smartrouter.example.com** は Smart Router URL で、**USERNAME** および **PASSWORD** は、Smart Router 設定のログイン認証情報です。

- データサンプルを投入できるように、マネージド外の KIE Server に新しいコンテナを作成するには、以下の HTTP 要求を送信します。

```
$ curl -v -X POST -H 'Content-type: application/xml' -H 'X-KIE-Content-Type: xstream' -d
@create-container.xml -u ${KIE_CRED} http://${KIE_SERVER_HOST}:${KIE_SERVER-
PORT}/kie-server/services/rest/server/config/
```

- create-container.xml** ファイルの内容を確認します。

```
<script>
<create-container>
  <container container-id="example:timer-test:1.1">
    <release-id>
      <group-id>example</group-id>
      <artifact-id>timer-test</artifact-id>
      <version>1.1</version>
    </release-id>
    <config-items>
      <itemName>RuntimeStrategy</itemName>
      <itemValue>PER_PROCESS_INSTANCE</itemValue>
      <itemType></itemType>
    </config-items>
  </container>
</create-container>
</script>
```

Smart Router コンソールにデプロイされたメッセージが表示されます。以下に例を示します。

```
INFO: Added http://localhost:8180/kie-server/services/rest/server as server location for
container example:timer-test:1.1
```

- コンテナのリストを表示するには、次のコマンドを入力します。

```
$ curl http://localhost:9000/mgmt/list
```

コンテナのリストが表示されます。

```
{
  "containerInfo": [{
    "alias": "timer-test",
    "containerId": "example:timer-test:1.1",
    "releaseId": "example:timer-test:1.1"
  }],
  "containers": [
    {"example:timer-test:1.1": ["http://localhost:8180/kie-server/services/rest/server"]},
    {"timer-test": ["http://localhost:8180/kie-server/services/rest/server"]}
  ],
  "servers": [
    {"kieserver2": []},
  ]
}
```

```

    {"kieserver1": ["http://localhost:8180/kie-server/services/rest/server"]}
  ]
}

```

- Smart Router の URL を使用してプロセスを初期化するには以下のコマンドを入力します。

```

$ curl -s -X POST -H 'Content-type: application/json' -H 'X-KIE-Content-Type: json' -d
 '{"timerDuration":"9s"}' -u kieserver:kieserver1!
 http://localhost:9000/containers/example:timer-test:1.1/processes/timer-
 test.TimerProcess/instances

```

## 30.2. TLS 対応の SMART ROUTER の設定

TLS (Transport Layer Security) 対応の Smart Router (KIE Server Router) を設定して、HTTPS トラフィックを許可することができます。さらに、Smart Router に対する、セキュリティーで保護されていない HTTP 接続を無効にできます。

### 前提条件

- Red Hat JBoss EAP 7.4 クラスターの各ノードに KIE Server がインストールされている。
- Smart Router がインストールされ、設定されている。詳細は、[「Smart Router による KIE サーバーインスタンスのロードバランシング」](#) を参照。

### 手順

Smart Router を起動するには、以下のいずれかの方法を使用します。

- TLS サポートと HTTPS を有効にして Smart Router を起動し、HTTP 接続を許可するには、以下のコマンドを入力します。

```

java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
      -Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PASSWORD>
      -Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>
      -Dorg.kie.server.router.tls.port = <HTTPS_PORT>
      -jar rhpam-7.13.5-smart-router.jar

```

上記の例では、以下の変数を置き換えてください。

- <KEYSTORE\_PATH>: キーストアの保存先のパス
- <KEYSTORE\_PASSWORD>: キーストアのパスワード
- <KEYSTORE\_ALIAS>: 証明書の保存に使用されるエイリアス名。
- <HTTPS\_PORT>: HTTPS ポートデフォルトの HTTPS ポートは **9443** です。
- TLS サポートおよび HTTPS を有効にし、HTTP 接続を無効にして Smart Router を起動するには、以下のコマンドを入力します。

```

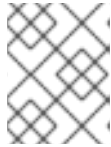
java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
      -Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PASSWORD>
      -Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>

```



```
-Dorg.kie.server.router.tls.port = <HTTPS_PORT>
-Dorg.kie.server.router.port=0
-jar rhpam-7.13.5-smart-router.jar
```

**org.kie.server.router.port** システムプロパティを **0** に設定すると、HTTP リスナーは登録されません。TLS が設定されており、HTTP リスナーが登録されていない場合は、Smart Router は HTTPS ポートでしかリッスンしません。



#### 注記

TLS を設定せずに、**org.kie.server.router.port** を **0** に設定して HTTP を無効にすると、エラーが発生して Smart Router が停止します。

### 30.3. エンドポイント認証用の SMART ROUTER の設定

エンドポイント認証用に Smart Router (KIE Server ルーター) を設定できます。

#### 前提条件

- Red Hat JBoss EAP 7.4 クラスターの各ノードに KIE Server がインストールされている。
- Smart Router がインストールされ、設定されている。詳細は、「[Smart Router による KIE サーバーインスタンスのロードバランシング](#)」を参照。

#### 手順

- エンドポイント認証が有効な Smart Router を起動するには、管理認証情報を設定します。
  - a. 以下のプロパティを KIE Server 設定に追加します。

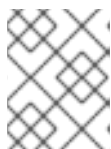
```
`org.kie.server.router.management.username`
`org.kie.server.router.management.password`
```

デフォルトの **username** は KIE Server ID です。

- b. 次のプロパティを Smart Router 設定に追加します。

```
`org.kie.server.router.management.password`
```

**password** プロパティの値は **true** または **false** (デフォルト) です。



#### 注記

エンドポイント認証を有効にすると、コンテナの表示または追加、削除操作を認証する必要があります。

1. 必要に応じて、Smart Router にユーザーを追加します。以下に例を示します。

```
java -jar rhpam-7.13.5-smart-router.jar -addUser <USERNAME> <PASSWORD>
```

2. 必要に応じて、Smart Router からユーザーを削除します。以下に例を示します。

```
java -jar rhpam-7.13.5-smart-router.jar -removeUser <USERNAME>
```

## 30.4. SMART ROUTER 動作の設定

KIE Server が複数あるクラスター環境では、デフォルトの動作は各 KIE Server に並行して要求を送信し、各 KIE Server のホストには "round-robin" 方式を使用して要求が送信されます。以下の例の環境では、各 KIE Server は同じ KJAR でデプロイされますが、KJAR のバージョンはそれぞれ異なります。

表30.1 環境の例

| サーバー名       | KJAR バージョン  | ホスト                             |
|-------------|---|---------------------------------|
| kie-server1 | kjar:1.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=1.0) | 129.0.1.1, 129.0.1.2, 129.0.1.3 |
| kie-server2 | kjar:2.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=2.0) | 129.0.2.1, 129.0.2.2, 129.0.2.3 |
| kie-server3 | kjar:3.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=3.0) | 129.0.3.1, 129.0.3.2, 129.0.3.3 |

要求を送信する場合には、要求が **kie-server1 (129.0.1.2)**、**kie-server2 (129.0.2.3)** および **kie-server3 (129.0.3.1)** に送信されます。

2 番目の要求を送信する場合には、要求は、各 KIE Server の次のホストに送信されます。例: **kie-server1 (129.0.1.3)**、**kie-server2 (129.0.2.1)** および **kie-server3 (129.0.3.2)**。

Smart Router には変更可能なコンポーネントが 3 つあり、この動作を変更できます。

### ContainerResolver

サーバーを操作する時に使用するコンテナ ID を検索するコンポーネント

### RestrictionPolicy

Smart Router が特定のエンドポイントを使用していないようにするコンポーネント

### ConfigRepository

Smart Router 設定を維持するコンポーネント。これは主にルーティングテーブルに関係します。

### IdentityService

独自の ID プロバイダーを使用できるようにするコンポーネント。これは KIE Server インスタンス用です。

Smart Router は **ServiceLoader** ユーティリティを使用してこれらのコンポーネントを実装します。

### ContainerResolver

**META-INF/services/org.kie.server.router.spi.ContainerResolver**

### RestrictionPolicy

**META-INF/services/org.kie.server.router.spi.RestrictionPolicy**

### ConfigRepository

**META-INF/services/org.kie.server.router.spi.ConfigRepository**

### IdentityService

**META-INF/services/org.kie.server.router.identity.IdentityService**

たとえば、上記のシナリオでは、Smart Router が利用可能な KIE Server から最新バージョンの KJAR

プロセスを検索して、そのプロセスで常に開始するように、**ContainerResolver** をカスタマイズできます。このシナリオでは、各 KIE Server は KJAR を1つホストして、バージョンはすべて同じエイリアスを共有します。

Smart Router は実行可能な jar であるため、拡張子を追加するには、コマンドを変更する必要があります。以下に例を示します。

```
java -cp LOCATION/router-ext-7.13.5.redhat-00002.jar:rhpm-7.13.5-smart-router.jar
org.kie.server.router.KieServerRouter
```

サービスが開始されると、コンポーネントに使用されている実装を示すログ出力が表示されます。

```
Mar 01, 2017 1:47:10 PM org.kie.server.router.KieServerRouter <init>
INFO: KIE Server router repository implementation is InMemoryConfigRepository
Mar 01, 2017 1:47:10 PM org.kie.server.router.proxy.KieServerProxyClient <init>
INFO: Using 'LatestVersionContainerResolver' container resolver and restriction policy
'ByPassUserNotAllowedRestrictionPolicy'
Mar 01, 2017 1:47:10 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.3.6.Final
Mar 01, 2017 1:47:10 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.3.6.Final
Mar 01, 2017 1:47:11 PM org.kie.server.router.KieServerRouter start
INFO: KieServerRouter started on localhost:9000 at Wed Mar 01 13:47:11 CET 2017
```

## 30.5. 設定ファイルを使用した SMART ROUTER の設定

コマンドラインで Smart Router 設定を行う代わりに、設定ファイルを使用することができます。この場合、パスワードを含む設定内容は、コマンドラインターミナルやサーバーのログには表示されません。

### 手順

1. 設定ファイルの作成このファイルには、**property=value** 形式で任意の数の行を含めることができます。ファイルには以下のいずれかのプロパティを含めることができます。すべてのプロパティは任意です。

表30.2 Smart Router の設定ファイルでサポートされているプロパティ

| プロパティ名                            | 説明                                   | デフォルト値           |
|-----------------------------------|--------------------------------------|------------------|
| <b>org.kie.server.router.id</b>   | 他のコンポーネントを識別するための Smart Router の識別子。 | 該当なし             |
| <b>org.kie.server.router.name</b> | 他のコンポーネントを識別するための Smart Router の名前。  | 該当なし             |
| <b>org.kie.server.router.host</b> | Smart Router を動作させるマシンのホスト名          | <b>localhost</b> |

| プロパティ名  | 説明  | デフォルト値             |
|---|---|--------------------|
| <b>org.kie.server.router.port</b>                   | 受信する HTTP 接続のポート。<br>TLS 接続を設定している場合は、このプロパティを <b>0</b> に設定して、HTTP 接続を無効にすることができます。   | <b>9000</b>        |
| <b>org.kie.server.router.url.external</b>           | Smart Router にアクセスするための外部 URL   | 該当なし               |
| <b>org.kie.server.router.tls.port</b>               | TLS 接続を受信するためのポート   | 該当なし               |
| <b>org.kie.server.router.tls.key store</b>          | TLS 接続用のキーストアファイル   | 該当なし               |
| <b>org.kie.server.router.tls.key store.password</b> | TLS 接続用のキーストアのパスワード   | 該当なし               |
| <b>org.kie.server.router.tls.key store.keyalias</b> | キーストア内の TLS 証明書を参照するエイリアス名  | 該当なし               |
| <b>org.kie.server.router.repo</b>                   | 現在のリポジトリを格納するディレクトリー  | 現在の作業ディレクトリー       |
| <b>org.kie.router.identity.provider</b>             | KIE サーバーインスタンスを Smart Router で認証するための任意のカスタムプロバイダークラスです。このクラスは、 <b>org.kie.server.router.identity.IdentityManager</b> インターフェイスを実装する必要があります。このインターフェイスのソースコードについては、 <a href="#">Git Hub リポジトリ</a> を参照してください。 | 該当なし               |
| <b>org.kie.server.controller</b>                    | コントローラーに接続するための URL   | 該当なし               |
| <b>org.kie.server.controller.user</b>               | コントローラーに接続するためのユーザー名  | <b>kieserver</b>   |
| <b>org.kie.server.controller.pwd</b>                | コントローラーに接続するためのパスワード  | <b>kieserver1!</b> |
| <b>org.kie.server.controller.token</b>              | コントローラーに接続するための認証トークン   | 該当なし               |

| プロパティ名   | 説明  | デフォルト値       |
|--|---|--------------|
| <b>org.kie.server.controller.retry.interval</b>      | コントローラーへの接続に失敗した場合に、接続を再試行する間隔 (秒)                                  | 10           |
| <b>org.kie.server.controller.retry.limit</b>         | コントローラーへの接続に失敗した場合の最大再試行回数  | infinite     |
| <b>org.kie.server.router.config.watcher.enabled</b>  | <b>true</b> に設定すると、Smart Router は設定ファイルを定期的にスキャンし、変更を適用します。         | <b>false</b> |
| <b>org.kie.server.router.config.watcher.interval</b> | 設定ファイルを再スキャンする時間 (秒)  | 5            |
| <b>org.kie.server.router.management.password</b>     | <b>true</b> に設定すると、Smart Router は KIE サーバーからの接続を認証するためにパスワードを要求します。 | <b>false</b> |

2. 以下のコマンドラインで Smart Router を起動します。

```
java -Dorg.kie.server.router.config.file=<CONFIG_FILE> -jar rhpam-7.13.5-smart-router.jar
```

<CONFIG\_FILE> を、設定ファイルの名前に置き換えます。

## 第31章 QUARTZ タイマーサービスの設定

クラスターで KIE Server を実行する場合には、Quartz タイマーサービスを設定する必要があります。

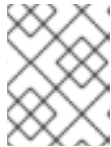
アプリケーションサーバーでデータベースを設定する前に、Quartz テーブルを作成するために Quartz データベースを準備する必要があります。このデータベースで、タイマーデータと Quartz 定義ファイルを保持します。

### 前提条件

- サポートのある JTA 以外のデータソース (例: PostgreSQL データソースなど) がアプリケーションサーバーに接続されている。

### 手順

1. データベースに Quartz テーブルを作成し、お使いのデータベース用の DDL スクリプトを使用してタイマーイベントが同期できるようにします。  
DDL スクリプトは、**QUARTZ\_HOME/docs/dbTables** にデプロイメントした、補足用の Zip アーカイブにあります。



### 注記

**quartz\_tables\_drop\_db2.sql** など、**drop** の用語が含まれるスクリプトは Quartz テーブルを作成する前に、**drop** が含まれるテーブルを除外します。

2. Quartz 設定ファイル **quartz-definition.properties** を、**JBoss\_HOME/Mode/configuration/** ディレクトリーに作成して、以下のサンプルコンテンツを追加します。

```
#=====
# Configure Main Scheduler Properties
#=====

org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO
#=====

# Configure ThreadPool
#=====

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 5
#=====

# Configure JobStore
#=====

org.quartz.jobStore.misfireThreshold = 60000
org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCMT
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
```

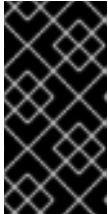
```

org.quartz.jobStore.tablePrefix=QRTZ_
org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000
#=====

# Configure Datasources
#=====

org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsDS
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quartzNotManagedDS
# Note the configured data sources that accommodate the two Quartz schemes at the very
end of the file.

```



## 重要

推奨されるクラスター検出の間隔は 20 秒で、**quartz-definition.properties** ファイルの **org.quartz.jobStore.clusterCheckinInterval** 属性に設定されています。システムへのパフォーマンスの影響を考慮し、必要に応じて設定を変更してください。

3. **org.quartz.properties** プロパティの **quartz-definition.properties** ファイルに絶対パスを指定します。
4. オプション: Quartz トリガーの再試行回数と遅延を設定するには、次のシステムプロパティを更新します。
  - **org.jbpm.timer.quartz.retries** (デフォルト値は 5)
  - **org.jbpm.timer.quartz.delay** (ミリ秒単位、デフォルト値は 1000)



## 注記

デフォルトでは、Quartz には 2 つのデータソースが必要です。

- プロセスエンジンのトランザクションに参加するマネージドデータソース。
- トランザクション処理を行わずにトリガーするタイマーを検索するためのマネージド外のデータソース。

Red Hat Process Automation Manager ビジネスアプリケーションでは、Quartz データベース (スキーマ) が Red Hat Process Automation Manager テーブルと共存することを想定しているため、Quartz のトランザクション操作に使用するデータソースを生成します。

他の (トランザクション以外) データソースを設定する必要がありますが、主なデータソースと同じデータベースを参照する必要があります。

## 第32章 関連情報

- [Red Hat JBoss EAP 7.4 への Red Hat Process Automation Manager のインストールおよび設定](#)
- [Red Hat Process Automation Manager インストールの計画](#)
- [Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#)
- [テンプレートを使用した Red Hat OpenShift Container Platform 3 への Red Hat Process Automation Manager 環境のデプロイメント](#)



## パート IV. RED HAT JBOSS WEB SERVER での RED HAT PROCESS AUTOMATION MANAGER のインストールと設定

本書では、JBoss Web Server に Red Hat Process Automation Manager 7.13 をインストールする方法を説明しています。



### 注記

Red Hat JBoss Web Server および Apache Tomcat での Red Hat Process Automation Manager へのサポートがメンテナンスフェーズに入りました。Red Hat では、上記のプラットフォームでの Red Hat Process Automation Manager のサポートを継続しますが、以下の制限があります。

- Red Hat では、新しい認定またはソフトウェアの機能をリリースしません。
- Red Hat では、重大な影響を与えるセキュリティパッチおよびミッションクリティカルなバグ修正パッチが含まれる認定セキュリティパッチのみをリリースします。

今後、Red Hat ではハイブリッドクラウド戦略と互換性のある新しいプラットフォームや製品コンポーネントに移行するようにお願いする場合があります。

### 前提条件

- [Red Hat Process Automation Manager インストールの計画](#)の内容を確認済みである。
- Red Hat JBoss Web Server 5.5.1 がインストールされている。Red Hat JBoss Web Server のインストールに関する情報は、[Red Hat JBoss Web Server インストールガイド](#)を参照してください。

## 第33章 RED HAT PROCESS AUTOMATION MANAGER とは

Red Hat Process Automation Manager は、ビジネスの自動化アプリケーションとマイクロサービスを作成する Red Hat ミドルウェアプラットフォームです。これにより、企業のビジネスユーザーと IT ユーザーが、ビジネスプロセスおよびポリシーを文書化、シミュレート、管理、自動化、およびモニターできます。ビジネスユーザーおよび IT ユーザーがより効果的に協力できるように設計されているため、ビジネスアプリケーションは簡単にすばやく変更できます。

この製品は、Business Central と KIE Server で構成されています。

- KIE Server では、ルール、およびその他のアーティファクトが実行されます。これは、ルールをインスタンス化して実行し、計画の問題を解決するために使用されます。KIE Server はビジネスアセットのランタイム環境を提供し、アセットリポジトリ (ナレッジストア) に保存されたデータにアクセスします。
- Business Central は、KIE Server が実行するビジネスルールを作成して管理するグラフィカルユーザーインターフェイスです。これにより、以下のタスクを実行できます。
  - ルール、プロセス、および関連アセットを作成、管理、および編集します。
  - 接続された KIE Server インスタンスおよびその KIE コンテナを管理します (デプロイメントユニット)。
  - Business Central に接続した KIE Server インスタンスのプロセスおよびタスクに対してランタイム操作を実行します。  
Business Central は、スタンドアロンの JAR ファイルとしても使用できます。Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。

Red Hat JBoss Web Server は Tomcat をベースとしたエンタープライズレベルの Web サーバーで、中規模および大規模のアプリケーション用に設計されています。Red Hat JBoss Web Server は、Java Server Pages (JSP) および Java Servlet テクノロジー、PHP、ならびに CGI をデプロイするための単一プラットフォームを提供します。

Red Hat JBoss Web Server インストールに KIE Server およびヘッドレス Process Automation Manager コントローラーをインストールできます。または、スタンドアロンの Business Central JAR ファイルを実行できます。

本書では、Red Hat JBoss Web Server インスタンスに Red Hat Process Automation Manager をインストールする方法を説明します。

他の環境への Red Hat Process Automation Manager のインストール方法に関する説明は、以下のドキュメントを参照してください。

- [Red Hat JBoss EAP 7.4 への Red Hat Process Automation Manager のインストールおよび設定](#)
- [IBM WebSphere Application Server への KIE Server のインストールおよび設定](#)
- [Oracle WebLogic Server への KIE Server のインストールおよび設定](#)
- [Operator を使用した Red Hat OpenShift Container Platform 4 への Red Hat Process Automation Manager 環境のデプロイメント](#)
- [テンプレートを使用した Red Hat OpenShift Container Platform 3 への Red Hat Process Automation Manager 環境のデプロイメント](#)

サポートされるコンポーネントについては、以下のドキュメントを参照してください。

- [What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#)
- [Red Hat JBoss Data Virtualization 7 でサポートされる設定](#)

## 33.1. RED HAT PROCESS AUTOMATION MANAGER コンポーネント

この製品は、Business Central と KIE Server で構成されています。

- Business Central は、ビジネスルールを作成して管理するグラフィカルユーザーインターフェイスです。Business Central は、Red Hat JBoss EAP インスタンスまたは Red Hat OpenShift Container Platform (OpenShift) にインストールできます。  
Business Central は、スタンドアロンの JAR ファイルとしても使用できます。Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。
- KIE Server では、ルール、およびその他のアーティファクトが実行されます。これは、ルールをインスタンス化して実行し、計画の問題を解決するために使用されます。KIE Server は、Red Hat JBoss EAP インスタンス、Red Hat JBoss EAP クラスター、OpenShift、Oracle WebLogic Server インスタンス、IBM WebSphere Application Server インスタンスに、または Spring Boot アプリケーションの一部としてインストールできます。  
KIE Server は、管理モードまたは非管理モードで動作するように設定できます。KIE Server が非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE コンテナは、プロジェクトの特定のバージョンです。KIE Server が管理されている場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Red Hat JBoss Web Server インストールに KIE Server およびヘッドレス Process Automation Manager コントローラーをインストールできます。または、スタンドアロンの Business Central JAR ファイルを実行できます。

## 33.2. RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー

Business Central または KIE 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 または KIE Server のインストール時に、ユーザーとロールを作成できます。

Business Central と KIE Server の両方が単一のインスタンスで動作している場合、Business Central で認証されたユーザーは KIE Server にもアクセスできます。

ただし、Business Central と KIE Server が別のインスタンスで動作している場合、Business Central で認証されたユーザーが KIE Server にアクセスするには、別途認証が必要です。たとえば、Business Central で認証されているものの、KIE Server で認証されていないユーザーが Business Central でプロセスマニフェストを表示または管理しようとすると、401 エラーがログファイルに記録され、**Invalid credentials to load data from remote server.Contact your system administrator.** メッセージが Business Central に表示されます。

このセクションでは、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 KIE Server 用に予約されています。このため、Business Central または KIE Server のいずれか、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、ビジネスインジケター、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、KIE Server REST 機能にアクセスできます。このロールは、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。

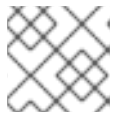
## 第34章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード

インストーラー JAR ファイルまたはデプロイ可能な ZIP ファイルを使用して Red Hat Process Automation Manager をインストールすることができます。インストーラーは、インタラクティブモードまたはコマンドラインインターフェイス (CLI) モードで実行できます。もしくは、Business Central および KIE Server のデプロイ可能な ZIP ファイルをデプロイメントして設定できます。Business Central をアプリケーションサーバーにデプロイせずに実行する場合は、Business Central スタンドアロン JAR ファイルをダウンロードします。

お使いの環境およびインストール要件に応じた Red Hat Process Automation Manager ディストリビューションをダウンロードします。

### 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. お好みのインストール方法に従って、以下の製品ディストリビューションのいずれかをダウンロードします。



### 注記

ダウンロードするのは、どれか1つのディストリビューションのみです。

- インストーラーを使用して Red Hat JBoss Web Server に Red Hat Process Automation Manager をインストールする場合は、**Red Hat Process Automation Manager 7.13.5 インストーラー (rhpm-installer-7.13.5.jar)** をダウンロードします。インストーラーのグラフィックユーザーインターフェイスに従って、インストールプロセスを進めます。
- デプロイ可能な zip ファイルを使用して Red Hat JBoss Web Server に KIE Server をインストールするには、以下のファイルをダウンロードします。
  - Red Hat Process Automation Manager 7.13.5 Add Ons(**rhpm-7.13.5-add-ons.zip**)
  - Red Hat Process Automation Manager 7.13.5 Maven リポジトリ(**rhpm-7.13.5-maven-repository.zip**)
- アプリケーションサーバーにデプロイせずに Business Central を実行するには、**Red Hat Process Automation Manager 7.13.5 Business Central Standalone (rhpm-7.13.5-business-central-standalone.jar)** をダウンロードします。

## 第35章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用

本セクションでは、インストーラー JAR ファイルを使用して Business Central および KIE Server をインストールする方法を説明します。JAR ファイルは、既存の Red Hat JBoss Web Server 5.5.1 サーバーインストールに Red Hat Process Automation Manager をインストールする実行ファイルです。インストーラーは、インタラクティブモードまたはコマンドラインインターフェイス (CLI) モードで実行できます。

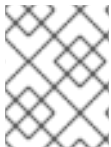
### 次のステップ:

以下のセクションのいずれかに記載される手順を行います。

- [「インタラクティブモードでのインストーラーの使用」](#)
- [「CLI モードでのインストーラーの使用」](#)

### 35.1. インタラクティブモードでのインストーラーの使用

Red Hat Process Automation Manager のインストーラーは、実行可能な JAR ファイルです。このインストーラーを使用して、既存の Red Hat JBoss Web Server 5.5.1 サーバーインストールに Red Hat Process Automation Manager をインストールできます。

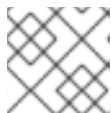


#### 注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

#### 前提条件

- Red Hat Process Automation Manager 7.13.5 Installer をダウンロードしている。手順は、[34章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) を参照してください。
- サポート対象の JDK がインストールされている。サポート対象の JDK のリストについては、[Red Hat Process Automation Manager 7 Supported Configurations](#) を参照してください。
- バックアップを作成してある Red Hat JBoss Web Server 5.5.1 サーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。



#### 注記

Tomcat への書き込み権限のあるユーザーとしてログインします。

- `$PATH` 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、`java-$JAVA_VERSION-openjdk-devel` パッケージに含まれています。



## 注記

Red Hat Process Automation Manager は、UTF-8 エンコーディングで機能するように設計されています。基礎となる JVM で別のエンコーディングシステムを使用すると、予期せぬエラーが発生する可能性があります。JVM で UTF-8 を使用するには、**-Dfile.encoding=UTF-8** のシステムプロパティを使用します。システムプロパティのリストは、[付録B Business Central システムプロパティ](#)を参照してください。

## 手順

1. 端末ウィンドウで、インストーラー JAR ファイルをダウンロードしたディレクトリーに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar
```



## 注記

Windows でインストーラーを実行すると、インストール時に管理者の認証情報が求められる場合があります。この要求を回避するには、インストールコマンドに **izpack.mode=privileged** オプションを追加します。

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.13.5.jar
```

また、32 ビットの Java 仮想マシンでインストーラーを実行している場合には、メモリー不足になる可能性があります。この問題を防ぐには、以下のコマンドを実行します。

```
java -XX:MaxHeapSize=4g -jar  
rhpam-installer-7.13.5.jar
```

グラフィカルインストーラーにスプラッシュ画面と使用許諾契約書のページが表示されます。

2. **I accept the terms of this license agreement (本使用許諾契約書の内容に同意します)** をクリックし、**Next** をクリックします。
3. Red Hat Process Automation Manager をインストールする Red Hat JBoss Web Server 5.5.1 のホームを指定して、**Next** をクリックします。
4. インストールするコンポーネントを選択し、**Next** をクリックします。  
Business Central は Red Hat JBoss Web Server にインストールすることができません。Business Central をインストールできるのは Red Hat JBoss EAP だけです。ただし、KIE Server とヘッドレス Process Automation Manager コントローラーは、Red Hat JBoss Web Server にインストールできます。ヘッドレス Process Automation Manager コントローラーは、KIE Server の管理に使用されます。複数の KIE Server インスタンスを管理する予定がある場合は、ヘッドレス Process Automation Manager コントローラーをインストールします。
5. ユーザーを作成して **Next** をクリックします。デフォルトでは、同じコンテナに Business Central と KIE Server の両方をインストールする場合は、新しいユーザーに **admin** ロール、**kie-server** ロール、および **rest-all** ロールが割り当てられます。KIE Server のみをインストールした場合には、ユーザーには **kie-server** ロールが割り当てられます。KIE Server REST 機能にアクセスするには **kie-server** ロールが必要です。



## 注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。たとえば、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし & の文字は使用できません。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

6. **Installation Overview** ページで、インストールするコンポーネントを確認し、**Next** をクリックしてインストールを開始します。
7. インストールが完了したら、**Next** をクリックします。
8. KIE Server がインストールされている場合には、**Component Installation** で **Configure Runtime** の手順が表示されます。**Configure Runtime Environment** ページで、デフォルトのインストールを実行するか、詳細設定を行うかを選択します。  
**Perform advanced configuration** を選択した場合は、データベース設定、または特定の KIE Server オプションのカスタマイズが選択できます。
9. **JDBC Drive Configuration** ページで **Customize database settings** を選択した場合は、データソースの JDBC ドライバーのベンダーを選択し、ドライバの JAR ファイルを 1 つ以上選択し、**Next** をクリックします。  
データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。
10. **KIE Server Properties Configuration** で **Customize KIE Server properties** を選択した場合は、以下のいずれかのプロパティを変更します。
  - KIE Server ID の値を変更して、KIE Server プロパティの名前を変更します。
  - 無効にする KIE Server 機能の選択を解除します。
11. **Next** をクリックして、ランタイム環境を設定します。
12. 画面上部に **Processing finished** が表示されたら、**Next** をクリックしてインストールを完了します。
13. 必要に応じて、**Generate Installation Script and Properties File** をクリックして、XML ファイルでインストールデータを保存し、**Done** をクリックします。  
インストーラーは、2 つのファイルを生成します。**auto.xml** ファイルは、今後のインストールを自動化し、**auto.xml.variables** ファイルは、ユーザーのパスワードと他の機密情報に関連する変数を保存します。**auto.xml** ファイルを使用して、元のインストールと同じタイプのサーバーおよび同じ設定の複数のシステムで Red Hat Process Automation Manager のインストールを繰り返します。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新します。XML ファイルを使用してインストールを実行するには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```



これで、インストーラーを使用して Red Hat Process Automation Manager がインストールできました。Business Central だけをインストールした場合は、この手順を繰り返して、別のサーバーに KIE Server をインストールします。



### 注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、**ALLOW\_SNAPSHOT\_ISOLATION** と **READ\_COMMITTED\_SNAPSHOT** を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 35.2. CLI モードでのインストーラーの使用

コマンドラインインターフェイス (CLI) を使用して Red Hat Process Automation Manager インストーラーを実行できます。



### 注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

### 前提条件

- Red Hat Process Automation Manager 7.13.5 Installer をダウンロードしている。手順は、[34章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) を参照してください。
- サポート対象の JDK がインストールされている。サポート対象の JDK のリストについては、[Red Hat Process Automation Manager 7 Supported Configurations](#) を参照してください。
- バックアップを作成してある Red Hat JBoss Web Server 5.5.1 サーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。



### 注記

Tomcat への書き込み権限のあるユーザーとしてログインします。

- **\$PATH** 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、**java-\$JAVA\_VERSION-openjdk-devel** パッケージに含まれています。



## 注記

Red Hat Process Automation Manager は、UTF-8 エンコーディングで機能するように設計されています。基礎となる JVM で別のエンコーディングシステムを使用すると、予期せぬエラーが発生する可能性があります。JVM で UTF-8 を使用するには、**-Dfile.encoding=UTF-8** のシステムプロパティを使用します。システムプロパティのリストは、[付録B Business Central システムプロパティ](#)を参照してください。

## 手順

1. 端末ウィンドウにおいて、インストーラーファイルをダウンロードしたディレクトリーに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar -console
```

コマンドラインの対話プロセスが開始し、使用許諾契約書が表示されます。

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

2. 使用許諾契約書を読んで **1** を入力し、**Enter** キーを押して続行します。

Specify the home directory of one of the following servers: Red Hat JBoss EAP 7 or Red Hat JBoss Web Server 5. For more information, see <https://access.redhat.com/articles/3405381>[Red Hat Process Automation Manager 7 Supported Configurations].

3. 既存の Red Hat JBoss Web Server 5.5.1 インストールの親ディレクトリーを入力します。インストーラーが、指定したインストール場所を確認します。**1** を入力して確認し、続行します。
4. インストーラーの指示に従って、インストールを完了します。



## 注記

ユーザー名およびパスワードを作成する場合は、指定したユーザー名が既存のロールまたはグループの名前と競合しないようにしてください。たとえば、**admin** というロールがある場合は、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし **&** の文字を使用することは **できません**。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

5. インストールが完了すると、以下のメッセージが表示されます。

```
Would you like to generate an automatic installation script and properties file?
```

6. **y** を入力してインストールデータが含まれる XML ファイルを作成します。あるいは、**n** を入力してインストールを完了します。**y** を入力すると、XML ファイルのパスを指定するように求められます。

7. パスを入力するか、Enter キーを押して提案されたパスを了承します。  
インストーラーは、2つのファイルを生成します。**auto.xml** ファイルは、今後のインストールを自動化し、**auto.xml.variables** ファイルは、ユーザーのパスワードと他の機密情報に関連する変数を保存します。複数のシステムで **auto.xml** ファイルを使用して、元のインストールと同じ設定の同じタイプのサーバーに、Red Hat Process Automation Manager を簡単に繰り返しインストールできます。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新します。XML ファイルを使用してインストールを実行するには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.13.5.jar <path-to-auto.xml-file>
```

8. KIE Server だけをインストールした場合は、この手順を繰り返して、別のサーバーにヘッドレス Process Automation Manager コントローラーをインストールします。



### 注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、ALLOW\_SNAPSHOT\_ISOLATION と READ\_COMMITTED\_SNAPSHOT を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON  
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 第36章 KIE SERVER ZIP ファイルのインストールと設定

カスタマーポータルでの Red Hat Process Automation Manager 7.13.5 Add Ons(**rhpam-7.13.5-add-ons.zip**) から入手できる **rhpam-7.13.5-kie-server-jws.zip** ファイルを使用して KIE Server をインストールし、Java Database Connectivity (JDBC) の Web サーバーデータソースを Red Hat JBoss Web Server に設定します。

### 36.1. ZIP ファイルからの KIE SERVER のインストール

KIE Server はビジネスアセットのランタイム環境を提供し、アセットリポジトリ (ナレッジストア) に保存されたデータにアクセスします。ZIP ファイルを使用して既存の Red Hat JBoss Web Server 5.5.1 サーバーインスタンスに KIE Server をインストールすることができます。



#### 注記

インストーラー JAR ファイルを使用した KIE Server のインストール方法は、[35章 Red Hat Process Automation Manager インストーラーの使用](#)を参照してください。

- [34章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) の説明に従い、以下のファイルがダウンロードされている。
  - Red Hat Process Automation Manager 7.13.5 Add Ons(**rhpam-7.13.5-add-ons.zip**)
  - Red Hat Process Automation Manager 7.13.5 Maven リポジトリ(**rhpam-7.13.5-maven-repository.zip**)
- バックアップを作成してある Red Hat JBoss Web Server 5.5.1 サーバーインストールが利用できる。Red Hat JBoss Web Server インストールのベースディレクトリーは **JWS\_HOME** とします。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

#### 手順

1. **rhpam-7.13.5-add-ons.zip** ファイルを展開します。
2. 展開した **rhpam-7.13.5-add-ons.zip** ファイルから、以下のファイルを展開します。
  - **rhpam-7.13.5-kie-server-jws.zip**
  - **rhpam-7.13.5-process-engine.zip**

以下の説明では、展開した **rhpam-7.13.5-kie-server-jws.zip** ファイルが含まれるディレクトリーは **JWS\_TEMP\_DIR** と呼ばれ、展開した **rhpam-7.13.5-process-engine.zip** ファイルが含まれるディレクトリーは **ENGINE\_TEMP\_DIR** と呼ばれます。

3. **JWS\_TEMP\_DIR/rhpam-7.13.5-kie-server-jws/kie-server.war** ディレクトリーを **JWS\_HOME/tomcat/webapps** ディレクトリーにコピーします。



#### 注記

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss Web Server インスタンスの既存デプロイメントと競合しないことを確認します。

4. **kie-server.war** ディレクトリーから **.war** 拡張子を削除します。
5. **kie-tomcat-integration-7.67.0.Final-redhat-00024.jar** ファイルを **ENGINE\_TEMP\_DIR** ディレクトリーから **JWS\_HOME/tomcat/lib** ディレクトリーに移動します。
6. **jboss-jacc-api-<VERSION>.jar** ファイル、**slf4j-api-<VERSION>.jar** ファイル、および **slf4j-jdk14-<VERSION>.jar** ファイルを **ENGINE\_TEMP\_DIR/lib** ディレクトリーから **JWS\_HOME/tomcat/lib** ディレクトリーに移動します。<VERSION> は **lib** ディレクトリーのバージョンのアーティファクトファイル名に置き換えます。
7. 最後の Valve 定義の後にある **JWS\_HOME/tomcat/conf/server.xml** ファイルの **<host>** 要素に、以下の行を追加します。

```
<Valve className="org.kie.integration.tomcat.JACCValve" />
```

8. テキストエディターで **JWS\_HOME/tomcat/conf/tomcat-users.xml** ファイルを開きます。
9. **JWS\_HOME/tomcat/conf/tomcat-users.xml** ファイルにユーザーおよびロールを追加します。以下の例で、**<ROLE\_NAME>** は Red Hat Process Automation Manager でサポートされているロールです。**<USER\_NAME>** および **<USER\_PWD>** は、選択したユーザー名とパスワードです。

```
<role rolename="<ROLE_NAME>"/>
<user username="<USER_NAME>" password="<USER_PWD>" roles="<ROLE_NAME>"/>
```

ユーザーに複数のロールを割り当てる場合は、以下の例に示すようにロールをコンマで区切ります。

```
<role rolename="admin"/>
<role rolename="kie-server"/>
<user username="rhpamUser" password="user1234" roles="admin,kie-server"/>
```

10. **JWS\_HOME/tomcat/bin** ディレクトリーで以下の手順の1つを実行します。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
CATALINA_OPTS="-Xmx1024m -Dorg.jboss.logging.provider=jdk"
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
```

## 注記

Microsoft SQL Server を使用する場合は、データベースに適したトランザクションの分離を使用していることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、**ALLOW\_SNAPSHOT\_ISOLATION** と **READ\_COMMITTED\_SNAPSHOT** を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

## 36.2. JDBC WEB SERVER データソースの設定

Java Database Connectivity (JDBC) は、Java で記述されたプログラムとデータベースを接続するための API 仕様です。データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。

通常、Red Hat Process Automation Manager を使用するソリューションは、1つのトランザクション内で複数のリソースを管理します。非同期のジョブ、イベント、タイマーなどの JMS。Red Hat Process Automation Manager では、データの原子性と一貫した結果を保証するために、可能な限りデータソースに XA ドライバーを必要とします。異なるスキーマのトランザクションコードがリスナー内に存在する場合や、jBPM エンジンが提供するフックから取得する場合は、XA ドライバーも必要となります。

1つのトランザクションに複数のリソースが参加していないことが確認できない限り、XA 以外のデータソースを使用しないでください。



### 注記

実稼働環境の場合は、実際のデータソースを指定します。実稼働環境で、データソースの例は使用しないでください。

### 前提条件

- Red Hat JBoss Web Server に Red Hat Process Automation Manager がインストールされている。
- [34章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) の記載どおりに、Red Hat Process Automation Manager 7.13.5 Maven リポジトリ (**rhpmam-7.13.5-maven-repository.zip**) および Red Hat Process Automation Manager 7.13.x Add-Ons (**rhpmam-7.13.5-add-ons.zip**) ファイルをダウンロードしている。
- 以下のサポート対象のデータベースおよび Hibernate 方言の1つを設定する。
  - DB2: **org.hibernate.dialect.DB2Dialect**
  - MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
  - MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
  - MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
  - Oracle: **org.hibernate.dialect.Oracle10gDialect**
  - PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
  - PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**
  - Sybase: **org.hibernate.dialect.SybaseASE157Dialect**

### 手順

1. 以下の手順を実行して、データベースを準備します。

- a. **TEMP\_DIR** などの一時ディレクトリーに **rhpm-7.13.5-add-ons.zip** を展開します。
- b. **TEMP\_DIR/rhpm-7.13.5-migration-tool.zip** を展開します。
- c. 現在のディレクトリーから、**TEMP\_DIR/rhpm-7.13.5-migration-tool/ddl-scripts** ディレクトリーに移動します。このディレクトリーには、複数のデータベースタイプの DDL スクリプトが含まれています。
- d. 使用するデータベースに、お使いのデータベースタイプの DDL スクリプトをインポートします。以下に例を示します。

```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



### 注記

PostgreSQL または Oracle を Spring Boot と併用する場合は、対応する Spring Boot の DDL スクリプト (**/ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql** または **/ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql**) をインポートする必要があります。

2. **rhpm-7.13.5-maven-repository.zip** オフライン Maven リポジトリファイルを展開します。
3. デプロイメントしたオフラインの Maven リポジトリから **JWS\_HOME/tomcat/lib** ディレクトリーにコピーします。**VERSION** は対象のライブラリーのバージョンに置き換えます。

```
org/jboss/spec/javax/transaction/jboss-transaction-api_1.2_spec/{VERSION}/jboss-transaction-api_1.2_spec-{VERSION}.jar
org/jboss/integration/narayana-tomcat/{VERSION}/narayana-tomcat-{VERSION}.jar
org/jboss/narayana/jta/narayana-jta/{VERSION}/narayana-jta-{VERSION}.jar
org/jboss/jboss-transaction-spi/{VERSION}/jboss-transaction-spi-{VERSION}.jar
```

4. データベースの JDBC ドライバーを **JWS\_HOME/tomcat/lib** ディレクトリーにコピーします。
5. **JWS\_HOME/tomcat/conf/context.xml** ファイルで、プーリングしている XA データソースを設定します。



### 注記

以下の例のプロパティーで、お使いのデータベースサーバーに該当しない場合があります。JDBC ドライバーのドキュメントを参照して、どのプロパティーを設定するかを判断してください。

- a. プーリング機能なしで XA データソースを設定します。この XA データソースは、対象のデータベースに新規接続を作成するときに使用します。以下の例では、XA データソースは **xads** で、変数は [表36.1「XA データソースの変数」](#) に定義されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
auth="Container"
databaseName="${datasource.dbName}"
description="XA Data Source"
factory="org.apache.tomcat.jdbc.naming.GenericNamingResourcesFactory"
loginTimeout="0"
```

```

name="xads"
uniqueName="xads"
portNumber="{datasource.port}"
serverName="{datasource.hostname}"
testOnBorrow="false"
type="{datasource.class}"
url="{datasource.url}"
URL="{datasource.url}"
user="{datasource.username}"
password="{datasource.password}"
driverType="4"
schema="{datasource.schema}"
/>
</Context>

```

表36.1 XA データソースの変数

| 変数                    | 説明   |
|-----------------------|--|
| <datasource.dbName>   | データベース名。   |
| <datasource.port>     | データベースのポート番号。  |
| <datasource.hostname> | データベースホスト名。  |
| <datasource.class>    | JDBC ドライバーの <b>XADatasource</b> クラス。   |
| <datasource.url>      | JDBC データベース接続の URL。データベースによって、URL プロパティが <b>url</b> 、または <b>URL</b> (H2 データベースなど) になります。 |
| <datasource.username> | データベース接続のユーザー名。  |
| <datasource.password> | データベース接続のパスワード。  |
| <datasource.schema>   | データベーススキーマ。  |

- b. 新規接続の作成時に XA データソースに依存するプーリングデータソースを設定します。以下の例では、データソースは **poolingXaDs** で、<datasource.username> はデータベース接続のユーザー名、<datasource.password> はデータベース接続のパスワードとなっています。

```

<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
name="poolingXaDs"
uniqueName="poolingXaDs"
auth="Container"
description="Pooling XA Data Source"
factory="org.jboss.narayana.tomcat.jta.TransactionDataSourceFactory"
testOnBorrow="true"
transactionManager="TransactionManager"

```



```

transactionSynchronizationRegistry="TransactionSynchronizationRegistry"
type="javax.sql.XADataSource"
username="${datasource.username}"
password="${datasource.password}"
xaDataSource="xads"
/>
</Context>

```

データソースが **java:comp/env/poolingXaDs** の JNDI 名で取得でき、次の手順に記載されているように、**org.kie.server.persistence.ds** システムプロパティを使用して KIE Server に渡すことができます。



### 注記

プーリングのデータソース設定は、特に **TransactionManager** や **TransactionSynchronizationRegistry** など、**kie-server** アプリケーションの **context.xml** ファイルで以前に設定された追加リソースに依存します。

6. KIE Server をがデータソースを使用するように設定します。

- a. テキストエディターで以下のスクリプトのいずれかを開きます。



### 注記

**setenv.sh** スクリプトまたは **setenv.bat** スクリプトはすでに存在しているはずですが、存在していない場合は作成してください。

- Linux または Unix の場合:

```
JWS_HOME/tomcat/bin/setenv.sh
```

- Windows の場合:

```
JWS_HOME/tomcat/bin/setenv.bat
```

- b. 以下のプロパティを **CATALINA\_OPTS** に追加します。 **<hibernate.dialect>** はお使いのデータベースの Hibernate 方言に置き換えます。

```

CATALINA_OPTS="-Xmx1024m
-Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.persistence.ds=java:comp/env/poolingXaDs
-Dorg.kie.server.persistence.tm=JBossTS
-Dorg.kie.server.persistence.dialect=${<hibernate.dialect>}"

```

## 第37章 キーストアを使用したパスワードセキュリティの確保

キーストアを使用して、Business Central と KIE Server の間の通信に使用するパスワードを暗号化できます。コントローラーと KIE Server のパスワードを暗号化する必要があります。Business Central と KIE Server を別のアプリケーションサーバーにデプロイする場合は、いずれのアプリケーションサーバーもキーストアを使用する必要があります。

Java Cryptography Extension KeyStore (JCEKS) は、対称鍵をサポートするため、キーストアには JCEKS を使用してください。JDK インストールに含まれる KeyTool を使用して、新しい JCEKS を作成します。



### 注記

KIE Server が JCEKS で設定されていない場合、KIE Server のパスワードはシステムプロパティにプレーンテキスト形式で保存されます。

### 前提条件

- Red Hat JBoss Web Server に KIE Server がインストールされている。
- Java 8 以降がインストールされている。

### 手順

1. テキストエディターで **JWS\_HOME/tomcat/conf/tomcat-users.xml** ファイルを開きます。
2. **kie-server** ロールを割り当てた KIE Server ユーザーを **JWS\_HOME/tomcat/conf/tomcat-users.xml** ファイルに追加します。以下の例では、**<USER\_NAME>** と **<PASSWORD>** は任意のユーザー名とパスワードに置き換えます。

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

3. KeyTool を使用して JCEKS を作成するには、Java 8 のホームディレクトリーで以下のコマンドを実行します。

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass
<ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

上記の例では、以下の変数を置き換えてください。

- **<KEYSTORE\_PATH>**: キーストアの保存先のパス
  - **<KEYSTORE\_PASSWORD>**: キーストアのパスワード
  - **<ALIAS\_KEY\_PASSWORD>**: エイリアスで保存した値にアクセスする時に使用するパスワード
  - **<PASSWORD\_ALIAS>**: プロセスへのエントリーに使用するエイリアス
4. プロンプトが表示されたら、作成した KIE Server ユーザーのパスワードを入力します。
  5. システムプロパティを設定するには、**JWS\_HOME/tomcat/bin** ディレクトリーで以下の手順の1つを実行し、変数は以下の表で記載されているように置き換えます。



## 注記

Business Central またはスタンドアロンのコントローラーが Red Hat JBoss Web Server とは別のインスタンスにインストールされている場合は、**kie.keystore.key.server.alias** と **kie.keystore.key.server.pwd** のプロパティを **CATALINA\_OPTS** に追加しないでください。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

表37.1 KIE Server JCEKS を読み込む時に使用するシステムプロパティ

| システムプロパティ                            | プレースホルダー            | 説明  |
|--------------------------------------|---------------------|---|
| <b>kie.keystore.keyStoreURL</b>      | <KEYSTORE_URL>      | 使用する JCEKS の URL (例: <b>file:///home/kie/keystores/keystore.jceks</b> ) |
| <b>kie.keystore.keyStorePwd</b>      | <KEYSTORE_PWD>      | JCEKS のパスワード  |
| <b>kie.keystore.key.server.alias</b> | <KEY_SERVER_ALIAS>  | パスワードの保存先となる REST サービスのキーのエイリアス   |
| <b>kie.keystore.key.server.pwd</b>   | <KEY_SERVER_PWD>    | 保存したパスワードを使用する REST サービスのエイリアスのパスワード                                    |
| <b>kie.keystore.key.ctrl.alias</b>   | <KEY_CONTROL_ALIAS> | パスワードの保存先のデフォルトの REST Process Automation Controller のキーのエイリアス           |
| <b>kie.keystore.key.ctrl.pwd</b>     | <KEY_CONTROL_PWD>   | 保存したパスワードを使用する、デフォルトの REST Process Automation Controller のエイリアスのパスワード   |

6. KIE Server を起動して、設定を検証します。

## 第38章 KIE SERVER インストールの確認

KIE Server が正しくインストールされていることを確認します。

### 前提条件

- KIE Server がインストールされ、設定されている。

### 手順

1. KIE Server を起動するには、**JWS\_HOME/tomcat/bin** ディレクトリーから端末を開いて以下のコマンドのいずれかを入力します。
  - Linux または UNIX ベースのシステムの場合:  

```
┆ $ ./startup.sh
```
  - Windows の場合:  

```
┆ startup.bat
```
2. 数分後に **JWS\_HOME/tomcat/logs** ディレクトリーのファイルを確認し、エラーが発生している場合は修正します。
3. Web ブラウザーに **http://localhost:8080/kie-server/services/rest/server** を入力し、KIE Server が正常に動作していることを確認します。
4. **tomcat-users.xml** ファイルに保存したユーザー名とパスワードを入力します。

## 第39章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのダウンロードおよびインストール

KIE Server は、管理モードまたは非管理モードで動作するように設定できます。KIE Server が非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE Server が管理されている場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Process Automation Manager コントローラーは Business Central と統合します。Business Central をインストールしている場合は、**Execution Server** ページを使用して KIE コンテナを作成および維持します。ただし、Business Central をインストールしない場合は、ヘッドレス Process Automation Manager コントローラーをインストールし、REST API または KIE Server Java Client API を使用してそのコントローラーと対話します。

### 前提条件

- [34章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) に従って、**Red Hat Process Automation Manager 7.13.5 Add Ons (rhpam-7.13.5-add-ons.zip)** ファイルがダウンロードされている。
- Red Hat JBoss Web Server 5.5.1 サーバーインストールが利用できる。Red Hat JBoss Web Server インストールのベースディレクトリーは **JWS\_HOME** とします。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

### 手順

1. **rhpam-7.13.5-add-ons.zip** ファイルを展開します。**rhpam-7.13.5-controller-jws.zip** ファイルは、展開したディレクトリーにあります。
2. **rhpam-7.13.5-controller-jws.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
3. **TEMP\_DIR/rhpam-7.13.5-controller-jws.zip/controller.war** ディレクトリーを **JWS\_HOME/tomcat/webapps** ディレクトリーにコピーします。



### 注記

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss Web Server インスタンスの既存デプロイメントと競合しないことを確認します。

4. **controller.war** ディレクトリーから **.war** 拡張子を削除します。
5. **TEMP\_DIR/rhpam-7.13.5-controller-jws/SecurityPolicy/** ディレクトリーの内容を **JWS\_HOME/bin** にコピーします。
6. ファイルの上書きを求めるプロンプトが出されたら、**Yes** を選択します。
7. **JWS\_HOME/tomcat/conf/tomcat-users.xml** ファイルに **kie-server** ロールおよびユーザーを追加します。以下の例で、**<USER\_NAME>** と **<PASSWORD>** は、任意のユーザー名とパスワードです。

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

8. KIE Server を実行するインスタンスの **JWS\_HOME/tomcat/bin** ディレクトリーで、以下のタスクの1つを実行します。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
CATALINA_OPTS="-Xmx1024m
-Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller"
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

上記の例では、以下の変数に置き換えてください。

- **<CONTROLLER\_USER>** および **<CONTROLLER\_PWD>** を、この手順の最初の方で定義した **kie-server** ロールを持つユーザー名とパスワードに置き換えます。
  - **<KIE\_SERVER\_ID>** を一意の識別子に置き換えます。
  - **<CONTROLLER\_HOST>:<CONTROLLER\_PORT>** をコントローラーの IP アドレス (ホストとポート) に置き換えます。KIE Server とコントローラーで同じサーバーを使用する場合は、**<CONTROLLER\_HOST>:<CONTROLLER\_PORT>** は **localhost:8080** に置き換えます。
9. ヘッドレス Process Automation Manager コントローラーを実行しているインスタンスの **JWS\_HOME/tomcat/bin** ディレクトリーに、以下の内容で読み取り可能な **setenv.sh** ファイルを作成します。ここでの **<USERNAME>** は KIE Server ユーザーで、**<USER\_PWD>** はそのユーザーのパスワードになります。
- ```
CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd=<USER_PWD>"
```
10. ヘッドレス Process Automation Manager コントローラーを起動するには、**JWS\_HOME/tomcat/bin** ディレクトリーに以下のいずれかのコマンドを実行します。

- Linux または UNIX ベースのシステムの場合:

```
$ ./startup.sh
```

- Windows の場合:

```
startup.bat
```

11. 数分後に **JWS\_HOME/tomcat/logs** ディレクトリーを確認し、エラーが発生している場合には修正します。
12. ヘッドレス Process Automation Manager コントローラーが正常に動作していることを確認するには、Web ブラウザーに **http://<CONTROLLER\_HOST>:<CONTROLLER\_PORT>/controller/rest/controller/management/servers** を入力します。KIE Server とコントローラーで同じサーバーを使用する場合は、<CONTROLLER\_HOST>:<CONTROLLER\_PORT> は **localhost:8080** に置き換えます。
13. **tomcat-users.xml** ファイルに保存したユーザー名とパスワードを入力します。



## 第40章 スタンドアロン BUSINESS CENTRAL の設定および実行

Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。設定ファイルのサンプルを使用して、追加設定なしで Business Central スタンドアロン JAR ファイルを実行するか、要件に合わせてサンプルファイルをカスタマイズできます。



### 注記

この JAR ファイルは、Red Hat Enterprise Linux で実行している場合にのみサポートされます。

### 前提条件

- 34章 [Red Hat Process Automation Manager インストールファイルのダウンロード](#) の記載通りに、Red Hat Process Automation Manager 7.13.5 Business Central Standalone(**rhpmam-7.13.5-business-central-standalone.jar**) ファイルおよび Red Hat Process Automation Manager 7.13.5 Add Ons (**rhpmam-7.13.5-add-ons.zip**) ファイルが、Red Hat Process Automation Manager 7.13 の [Software Downloads](#) ページからダウンロードされている。

### 手順

1. ダウンロードした **rhpmam-7.13.5-addons.zip** ファイルを一時ディレクトリーに展開します。このアーカイブには **rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルが含まれます。
2. **rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルを、**rhpmam-7.13.5-business-central-standalone.jar** ファイルが含まれるディレクトリーに展開します。**rhpmam-7.13.5-standalone-sample-configuration.zip** ファイルには、以下の設定ファイルのサンプルが含まれています。
  - **application-script.cli**: ユーザーおよび kie サーバーシステムプロパティを追加するスクリプトの例
  - **kie-fs-realm-users**: ユーザーデータのサンプル  
設定ファイルで提供されるサンプルデータを使用して **rhpmam-7.13.5-business-central-standalone.jar** ファイルを実行するか、要件に合わせてデータをカスタマイズできます。
3. 設定データをカスタマイズするには、以下の手順を実行します。
  - a. **application-script.cli** ファイルを編集し、**admin**、**user**、**rest-all**、**rest-client**、**kie-server** のロールを持つ管理者ユーザーを追加します。以下の例で、**<USERNAME>** および **<PASSWORD>** を、作成するユーザーのユーザー名とパスワードに置き換えます。

```
/subsystem=elytron/filesystem-realm=KieRealm:add-identity(identity=<USERNAME>
/subsystem=elytron/filesystem-realm=KieRealm:set-password(identity=<USERNAME>,
clear={password="<PASSWORD>"})
/subsystem=elytron/filesystem-realm=KieRealm:add-identity-attribute(identity=
<USERNAME>, name=role, value=["admin","user","rest-all","rest-client","kie-server"])
```

4. Business Central スタンドアロン JAR ファイルを実行するには、以下のコマンドを入力します。

```
java -jar rhpmam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli
```

5. JAR ファイルの実行時にアプリケーションプロパティを設定するには、コマンドに -**D<PROPERTY>=<VALUE>** パラメーターを追加します。ここで、<PROPERTY> はサポートされるアプリケーションプロパティの名前で、<VALUE> はプロパティの値になります。

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -  
D<PROPERTY>=<VALUE> -D<PROPERTY>=<VALUE>
```

たとえば、Business Central を実行し、ユーザー **controllerUser** として KIE Server に接続するには、次のコマンドを実行します。

```
java -jar rhpam-7.13.5-business-central-standalone.jar \  
--cli-script=application-script.cli \  
-Dorg.kie.server.user=controllerUser \  
-Dorg.kie.server.pwd=controllerUser1234
```

これにより、コンテナを KIE Server にデプロイできます。詳細は、[付録A Business Central システムプロパティ](#) を参照してください。



### 注記

Business Central でユーザーとグループの管理を有効にするには、**org.uberfire.ext.security.management.wildfly.cli.folderPath** プロパティの値を **kie-fs-realm-users** に設定します。

## 第41章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリー

Red Hat Process Automation Manager プロジェクトの作成時に、Business Central は Business Central 用に設定された Maven リポジトリーを使用します。Maven project object model (POM) ファイル (**pom.xml**) を変更することで、Maven グローバルまたはユーザー設定を使用して、すべての Red Hat Process Automation Manager プロジェクトに対して Red Hat Process Automation Manager の公開リポジトリーから依存関係を取得するように指示できます。また、Business Central と KIE Server が外部の Maven リポジトリーを使用するか、Maven のミラーを準備してオフラインで使用できるように設定できます。

Red Hat Process Automation Manager のパッケージ化およびデプロイメントのオプションに関する詳細は、[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。

### 41.1. プロジェクト設定ファイル (POM.XML) を使用した MAVEN の設定

Maven を使用して Red Hat Process Automation Manager プロジェクトをビルドおよび管理するには、POM ファイル (**pom.xml**) を作成および設定する必要があります。このファイルにはプロジェクトの設定情報が含まれます。詳細は [Apache Maven Project](#) を参照してください。

#### 手順

1. Maven プロジェクトを生成します。 **pom.xml** ファイルは、Maven プロジェクトの作成時に自動的に生成されます。
2. **pom.xml** ファイルを編集して、追加の依存関係および新たなリポジトリーを追加します。プロジェクトのコンパイルおよびパッケージング時に、Maven がすべての JAR ファイルおよび依存関係にある JAR ファイルを Maven リポジトリーからダウンロードします。

**pom.xml** ファイルのスキーマは、[http://maven.apache.org/maven-v4\\_0\\_0.xsd](http://maven.apache.org/maven-v4_0_0.xsd) から取得できます。POM ファイルの詳細は [Apache Maven Project POM](#) を参照してください。

### 41.2. MAVEN 設定ファイルの修正

Red Hat Process Automation Manager では、Maven の **settings.xml** ファイルを使用して Maven の実行を設定します。**settings.xml** ファイルにプロファイルを作成およびアクティベートし、Red Hat Process Automation Manager プロジェクトが使用する Maven リポジトリーを宣言します。

Maven の **settings.xml** ファイルの詳細は Apache Maven Project の [Setting Reference](#) を参照してください。

#### 手順

1. **settings.xml** ファイルでは、Red Hat Process Automation Manager プロジェクトが使用するリポジトリーを宣言します。通常、これは online Red Hat Process Automation Manager Maven リポジトリー、または Red Hat カスタマーポータルからダウンロードする Red Hat Process Automation Manager Maven リポジトリーおよび使用するカスタムアーティファクト用のリポジトリーのいずれかです。
2. Business Central または KIE Server が **settings.xml** ファイルを使用するように設定されていることを確認します。たとえば、**kie.maven.settings.custom=<SETTINGS\_FILE\_PATH>** プロパティを指定し、**<SETTINGS\_FILE\_PATH>** は **settings.xml** ファイルのパスに置き換えます。

Red Hat JBoss Web Server で、KIE Server の場合は **-Dkie.maven.settings.custom=<SETTINGS\_FILE\_PATH>** を **setenv.sh** (Linux) または **setenv.bat** (Windows) ファイルの **CATALINA\_OPTS** セクションに追加します。スタンドアロンの Business Central の場合は、以下のコマンドを入力します。

```
java -jar rhpam-7.13.5-business-central-standalone.jar --cli-script=application-script.cli -
Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>
```

## 41.3. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加

Red Hat Process Automation Manager プロジェクトで適切な Maven 依存関係を使用するには、プロジェクトの **pom.xml** ファイルに Red Hat Business Automation の BOM (bill of materials) ファイルを追加します。Red Hat Business Automation BOM は、Red Hat Decision Manager と Red Hat Process Automation Manager の両方に適用されます。BOM ファイルを追加すると、提供される Maven リポジトリから、推移的依存関係の適切なバージョンがプロジェクトに含まれます。

Red Hat Business Automation BOM (Bill of Materials) に関する詳細情報は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) を参照してください。

### 手順

1. Red Hat Business Automation BOM を **pom.xml** ファイルで宣言します。

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. **<dependencies>** タグでお使いのプロジェクトに必要な依存関係を宣言します。製品の BOM をプロジェクトにインポートしたら、ユーザー向け製品依存関係のバージョンが定義されるため、**<dependency>** 要素のサブ要素 **<version>** を指定する必要はありません。ただし、**<dependency>** 要素を使用して、プロジェクトで使用する依存関係を宣言する必要があります。
3. Business Central に作成されない標準のプロジェクトでは、お使いのプロジェクトに必要な依存関係をすべて指定します。Business Central に作成するプロジェクトでは、基本的なデシジョンエンジンとプロセスエンジンの依存関係が Business Central に自動的に提供されます。
  - 標準的な Red Hat Process Automation Manager プロジェクトでは、使用する機能に応じて、以下の依存関係を宣言します。

### 埋め込みプロセスエンジンの依存関係

```
<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
<dependency>
  <groupId>ch.qos.logback</groupId>
```

```
<artifactId>logback-classic</artifactId>
<version>${logback.version}</version>
</dependency>
```

- CDI を使用する Red Hat Process Automation Manager プロジェクトでは、通常、以下の依存関係を宣言します。

### CDI が有効化されたプロセスエンジンの依存関係

```
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>
```

- 標準的な Red Hat Process Automation Manager プロジェクトでは、以下の依存関係を宣言します。

### 埋め込みデシジョンエンジン依存関係

```
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>
```

```
<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>
```

- KIE Server を使用するには、以下の依存関係を宣言します。

#### クライアントアプリケーション KIE Server の依存関係

```
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-client</artifactId>
</dependency>
```

- Red Hat Process Automation Manager にリモートクライアントを作成するには、以下の依存関係を宣言します。

#### クライアントの依存関係

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- ルール、プロセス定義など、アセットを含む JAR ファイルを作成する場合は、お使いの Maven プロジェクトのパッケージングの種類を **kjar** と指定し、**org.kie:kie-maven-plugin** を使用して、**<project>** 要素に置かれた **kjar** パッケージングタイプを処理します。以下の例の **\${kie.version}** は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) に記載されている Maven ライブラリーのバージョンです。

```
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>
```

## 41.4. オフラインで使用する MAVEN ミラーリポジリーの準備

パブリックインターネットへの送信アクセスが、Red Hat Process Automation Manager のデプロイメントに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 リポジリーを用意して、このリポジリーをお使いの環境で使用できるようにする必要があります。



## 注記

Red Hat Process Automation Manager デプロイメントがインターネットに接続されている場合には、この手順を実行する必要はありません。

## 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. 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。
2. Red Hat カスタマーポータル [の Software Downloads ページ](#) に移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。

- **製品:** Process Automation Manager
- **バージョン:** 7.13.5
  - a. **Red Hat Process Automation Manager 7.13.5 Offliner Content List(rhpam-7.13.5-offliner.zip)** の製品配信可能ファイルをダウンロードして展開します。
  - b. **rhpam-7.13.5-offliner.zip** ファイルの内容を任意のディレクトリーに展開します。
  - c. ディレクトリーに移動し、以下のコマンドを入力します。

```
./offline-repo-builder.sh offliner.txt
```

このコマンドは、**repository** サブディレクトリーを作成し、必要なアーティファクトをこのサブディレクトリーにダウンロードします。これはミラーリポジトリーです。

一部のダウンロードが失敗したことを示すメッセージが表示された場合は、同じコマンドを再度実行してください。ダウンロードが再び失敗する場合は、Red Hat サポートに連絡してください。

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` を、プロジェクトの **pom.xml** ファイルの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`) の内容を、作成した **repository** サブディレクトリーにコピーします。
4. **repository** サブディレクトリーの内容を、Red Hat Process Automation Manager をデプロイしたコンピューターのディレクトリーにコピーします。このディレクトリーがオフラインの Maven ミラーリポジトリになります。
5. 「[Maven 設定ファイルの修正](#)」の説明に従って、Red Hat Process Automation Manager デプロイメント向けに、**settings.xml** ファイルを作成して設定します。
6. **settings.xml** ファイルで以下を変更します。
  - `<profile>` タグの下に、`<repositories>` タグまたは `<pluginRepositores>` タグがない場合は、不足しているタグを追加します。
  - `<repositories>` の下に、以下のコンテンツを追加します。

```
<repository>
  <id>offline-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

`/path/to/repo` は、ローカルの Maven ミラーリポジトリのディレクトリーに対する完全パスに置き換えます。

- `<pluginRepositories>` の下に、以下のコンテンツを追加します。

```
<repository>
  <id>offline-plugin-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

`/path/to/repo` は、ローカルの Maven ミラーリポジトリのディレクトリーに対する完全パスに置き換えます。

## 第42章 GIT リポジトリからのプロジェクトのインポート

Git は分散バージョン管理システムです。リビジョンをコミットオブジェクトとして実装します。リポジトリに変更を保存すると、Git リポジトリに新しいコミットオブジェクトが作成されます。

Business Central は Git を使用してプロジェクトデータ (ルールやプロセスなどのアセットを含む) を格納します。Business Central でプロジェクトを作成すると、Business Central に埋め込まれている Git リポジトリに追加されます。他の Git リポジトリにプロジェクトがある場合は、Business Central スペースから、そのプロジェクトを Business Central の Git リポジトリにインポートできます。

### 前提条件

- Red Hat Process Automation Manager プロジェクトが外部の Git リポジトリに存在している。
- 外部の Git リポジトリへの読み取りアクセスに必要な認証情報がある。

### 手順

1. Business Central で **Menu** → **Design** → **Projects** の順にクリックします。
2. プロジェクトをインポートするスペースを選択または作成します。デフォルトのスペースは **MySpace** です。
3. プロジェクトをインポートするには、次のいずれかを実行します。
  - **Import Project** をクリックします。
  - ドロップダウンリストから **Import Project** を選択します。
4. **Import Project** ウィンドウに、インポートするプロジェクトが含まれる Git リポジトリの URL および認証情報を入力し、**Import** をクリックします。プロジェクトが Business Central の Git リポジトリに追加され、現在のスペースで利用できるようになります。

## 第43章 LDAP と SSL の統合

Red Hat Process Automation Manager では、Red Hat Single Sign-On を通じて LDAP と SSL を統合することができます。詳細は [Red Hat Single Sign-On サーバー管理ガイド](#) を参照してください。

## 付録B BUSINESS CENTRAL システムプロパティ

このセクションに記載の Business Central のシステムプロパティは **standalone\*.xml** ファイルに渡されます。

### Git ディレクトリー

以下のプロパティを使用して、Business Central Git ディレクトリーの場所と名前を設定します。

- **org.uberfire.nio.git.dir**: Business Central の Git ディレクトリーの場所。
- **org.uberfire.nio.git.dirname**: Business Central の Git ディレクトリーの名前。デフォルト値は **.niogit** です。
- **org.uberfire.nio.git.ketch**: Git ketch を有効化または無効化。
- **org.uberfire.nio.git.hooks**: Business Central の Git ディレクトリーの場所。

### HTTP 経由の Git

次のプロパティを使用して、HTTP 経由で Git リポジトリーにアクセスできるように設定します。

- **org.uberfire.nio.git.proxy.ssh.over.http**: SSH が HTTP プロキシを使用するかどうかを指定します。デフォルト値は **false** です。
- **http.proxyHost**: HTTP プロキシのホスト名を定義します。デフォルト値は **null** です。
- **http.proxyPort**: HTTP プロキシのポート (整数値) を定義します。デフォルト値は **null** です。
- **http.proxyUser**: HTTP プロキシ名を定義します。
- **http.proxyPassword**: HTTP プロキシのユーザーパスワードを定義します。
- **org.uberfire.nio.git.http.enabled**: HTTP デモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.http.host**: このデーモンは、HTTP デモンが有効な場合にホストの識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.hostname**: HTTP デモンが有効な場合に、このデーモンはホスト名の識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.port**: このデーモンは、HTTP デモンが有効な場合にポート番号としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリーにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値は **8080** です。

### HTTPS 経由の Git

次のプロパティを使用して、HTTPS 経由で Git リポジトリーにアクセスできるように設定します。

- **org.uberfire.nio.git.proxy.ssh.over.https**: SSH が HTTPS プロキシを使用するかどうかを指定します。デフォルト値は **false** です。

- **https.proxyHost**: HTTPS プロキシのホスト名。デフォルト値は **null** です。
- **https.proxyPort**: HTTPS プロキシのポート (整数値)。デフォルト値は **null** です。
- **https.proxyUser**: HTTPS プロキシ名を定義します。
- **https.proxyPassword**: HTTPS プロキシのユーザーパスワードを定義します。
- **user.dir**: ユーザーディレクトリーの場所。
- **org.uberfire.nio.git.https.enabled**: HTTPS デーモンを有効または無効にします。デフォルト値は **false** です。
- **org.uberfire.nio.git.https.host**: このデーモンは、HTTPS デーモンが有効な場合にホストの識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.hostname**: このデーモンは、HTTPS デーモンが有効な場合にホスト名の識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.port**: このデーモンは、HTTPS デーモンが有効な場合にポート番号としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値は **8080** です。

## JGit

- **org.uberfire.nio.jgit.cache.instances**: JGit キャッシュサイズを定義します。
- **org.uberfire.nio.jgit.cache.overflow.cleanup.size**: JGit キャッシュオーバーフローのクリーンアップサイズを定義します。
- **org.uberfire.nio.jgit.remove.eldest.iterations**: 最も古い JGit の反復を削除するかどうかを定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.duration**: JGit 退避のしきい値の期間を定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.time.unit**: JGit 退避のしきい値の時間単位を定義します。

## Git デーモン

次のプロパティを使用して、Git デーモンを有効にして設定します。

- **org.uberfire.nio.git.daemon.enabled**: Git デーモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.daemon.host**: Git デーモンが有効な場合は、このプロパティをローカルホストの識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.daemon.hostname**: Git デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値は **localhost** です。

- **org.uberfire.nio.git.daemon.port**: Git デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値は **9418** です。
- **org.uberfire.nio.git.http.sslVerify**: Git リポジトリを確認する SSL 証明書を有効または無効にします。デフォルト値は **true** です。



### 注記

デフォルトポートまたは割り当てられたポートがすでに使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

## Git SSH

次のプロパティを使用して、Git SSH デーモンを有効にして設定します。

- **org.uberfire.nio.git.ssh.enabled**: SSH デーモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.ssh.host**: SSH デーモンが有効な場合は、このプロパティをローカルホスト識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.ssh.hostname**: SSH デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.SSH.port**: SSH デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値は **8001** です。



### 注記

デフォルトポートまたは割り当てられたポートがすでに使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

- **org.uberfire.nio.git.ssh.cert.dir**: ローカルの証明書が保存される **.security** ディレクトリーの場所。デフォルトは作業ディレクトリーです。
- **org.uberfire.nio.git.ssh.idle.timeout**: SSH のアイドルタイムアウトを設定します。
- **org.uberfire.nio.git.ssh.passphrase**: SCP スタイルの URL を持つ Git リポジトリのクローンを作成する場合に、オペレーティングシステムの公開キーストアにアクセスするためのパスフレーズ。たとえば、**git@github.com:user/repository.git** です。
- **org.uberfire.nio.git.ssh.algorithm**: SSH で使用されるアルゴリズム。デフォルト値は **RSA** です。
- **org.uberfire.nio.git.gc.limit**: GC の制限を設定します。
- **org.uberfire.nio.git.ssh.ciphers**: コンマ区切りの暗号化の文字列。利用可能な暗号化は **aes128-ctr**、**aes192-ctr**、**aes256-ctr**、**arcfour128**、**arcfour256**、**aes192-cbc**、**aes256-cbc** です。このプロパティを使用しない場合は、すべての暗号化が読み込まれます。
- **org.uberfire.nio.git.ssh.macs**: コンマ区切りのメッセージ認証コード (MAC) の文字列。利用可能な MAC は **hmac-md5**、**hmac-md5-96**、**hmac-sha1**、**hmac-sha1-96**、**hmac-sha2-256**、**hmac-sha2-512** です。このプロパティを使用しない場合は、すべての MAC が読み込まれます。



## 注記

RSA、または DSA 以外のアルゴリズムを使う場合は、Bouncy Castle JCE ライブラリーを使用するようにアプリケーションサーバーを設定します。

### KIE Server ノードおよび Process Automation Manager コントローラー

以下のプロパティを使用して Process Automation Manager コントローラーから KIE Server ノードへの接続を設定します。

- **org.kie.server.controller**: この URL は Process Automation Manager コントローラーへの接続に使用されます。たとえば、**ws://localhost:8080/business-central/websocket/controller** などです。
- **org.kie.server.user**: Process Automation Manager コントローラーから KIE Server ノードへの接続時に使用するユーザー名。このプロパティは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。
- **org.kie.server.pwd**: Process Automation Manager コントローラーから KIE Server ノードに接続する際に使用するパスワード。このプロパティは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。

### Maven など

以下のプロパティを使用して、Maven などの機能を設定します。

- **kie.maven.offline.force**: Maven のオフライン動作を強制します。true に設定すると、オンラインの依存関係解決が無効になります。デフォルト値は **false** です。



## 注記

このプロパティは、Business Central にのみ使用してください。他のコンポーネントとランタイム環境を共有する場合は、設定を分離して、Business Central にだけ適用してください。

- **org.uberfire.gzip.enable**: **GzipFilter** 圧縮フィルターで Gzip の圧縮を有効にするか、無効にします。デフォルト値は **true** です。
- **org.kie.workbench.profile**: Business Central プロファイルを選択します。許容値は、**FULL** または **PLANNER\_AND\_RULES** です。プリフィックス **FULL\_** で、プロファイルを設定し、管理者設定にこのプロファイルの設定が表示されないようにします。デフォルト値は **FULL** です。
- **org.appformer.m2repo.url**: Business Central は依存関係を検索する時に、Maven リポジトリのデフォルトの場所を使用します。デフォルト値は、**http://localhost:8080/business-central/maven2** など、Business Central 内の Maven リポジトリを参照します。このプロパティは、Business Central が起動する前に設定してください。デフォルト値は、内部の **m2** リポジトリへのファイルパスです。
- **appformer.ssh.keystore**: クラス名を指定して Business Central で使用する、カスタムの SSH キーストアを定義します。このプロパティが指定されていない場合はデフォルトの SSH キーストアを使用します。
- **appformer.ssh.keys.storage.folder**: このプロパティは、デフォルトの SSH キーストアを使用する場合にユーザーの SSH 公開鍵の保存フォルダーを定義します。このプロパティを指定しないと、この公開鍵は Business Central の **.security** フォルダーに保存され

ます。

- **appformer.experimental.features**: 実験的機能のフレームワークを有効にします。デフォルト値は **false** です。
- **org.kie.demo**: GitHub 外部へのデモアプリケーションのクローン作成を有効にします。
- **org.uberfire.metadata.index.dir**: Lucene の **.index** ディレクトリーが保存される場所。デフォルトは作業ディレクトリーです。
- **org.uberfire ldap.regex.role\_mapper**: LDAP プリンシパル名をアプリケーションのロール名にマッピングするのに使用する regex パターン。プリンシパルの値とロール名が一致する場合は、アプリケーションのロール名が変数ロールに置き換えられるため、変数ロールはパターンの一部でなければならない点に注意してください。
- **org.uberfire.sys.repo.monitor.disabled**: 設定モニターを無効にします。無効にした場合の影響を正しく理解していない場合は、無効にしないでください。デフォルト値は **false** です。
- **org.uberfire.secure.key**: パスワードの暗号化で使用するパスワード。デフォルト値は **org.uberfire.admin** です。
- **org.uberfire.secure.alg**: パスワードの暗号化で使用する暗号化アルゴリズム。デフォルト値は **PBEWithMD5AndDES** です。
- **org.uberfire.domain**: uberfire が使用するセキュリティドメイン名。デフォルト値は **ApplicationRealm** です。
- **org.guvnor.m2repo.dir**: Maven リポジトリーディレクトリーが保存される場所。デフォルト値は **<working-directory>/repositories/kie** です。
- **org.guvnor.project.gav.check.disabled**: グループ ID、アーティファクト ID、およびバージョン (GAV) のチェックを無効にします。デフォルト値は **false** です。
- **org.kie.build.disable-project-explorer**: Project Explorer で選択したプロジェクトの自動ビルドを無効にします。デフォルト値は **false** です。
- **org.kie.builder.cache.size**: プロジェクトビルダーのキャッシュサイズを定義します。デフォルト値は **20** です。
- **org.kie.library.assets\_per\_page**: プロジェクト画面のページごとのアセット数をカスタマイズできます。デフォルト値は **15** です。
- **org.kie.verification.disable-dtable-realtime-verification**: デシジョンテーブルのリアルタイム確認および検証を無効にします。デフォルト値は **false** です。

## Process Automation Manager コントローラー

以下のプロパティを使用して、Process Automation Manager コントローラーへの接続方法を設定します。

- **org.kie.workbench.controller**: Process Automation Manager コントローラーとの接続に使用する URL。例: **ws://localhost:8080/kie-server-controller/websocket/controller**
- **org.kie.workbench.controller.user**: Process Automation Manager コントローラーのユーザー。デフォルト値は **kieserver** です。



- **org.kie.workbench.controller.pwd**: Process Automation Manager コントローラーのパスワード。デフォルト値は **kieserver1!** です。
- **org.kie.workbench.controller.token**: Process Automation Manager コントローラーとの接続に使用するトークン文字列

### Java Cryptography Extension KeyStore (JCEKS)

JCEKS を設定するには、以下のプロパティを使用します。

- **kie.keystore.keyStoreURL**: Java Cryptography Extension KeyStore (JCEKS) の読み込みに使用する URL。たとえば、**file:///home/kie/keystores/keystore.jceks** です。
- **kie.keystore.keyStorePwd**: JCEKS に使用するパスワード。
- **kie.keystore.key.ctrl.alias**: デフォルトの REST Process Automation Manager コントローラーに使用するキーのエイリアス。
- **kie.keystore.key.ctrl.pwd**: デフォルトの REST Process Automation Manager コントローラーのエイリアスのパスワード

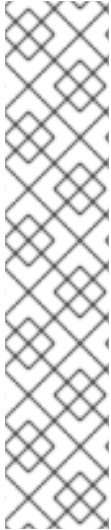
### レンダリング

以下のプロパティを使用して、Business Central と KIE Server のレンダリングフォームを切り替えます。

- **org.jbpm.wb.forms.renderer.ext**: Business Central と KIE Server のフォームのレンダリングを切り替えます。デフォルトでは、フォームのレンダリングは Business Central が行います。デフォルト値は **false** です。
- **org.jbpm.wb.forms.renderer.name**: Business Central と KIE Server のレンダリングフォームを切り替えることができます。デフォルト値は **workbench** です。

## パート V. IBM WEBSPHERE APPLICATION SERVER への KIE SERVER のインストールおよび設定

システム管理者は、Red Hat KIE Server に IBM WebSphere Application Server を設定し、IBM サーバーインスタンスに KIE Server をインストールできます。



### 注記

IBM WebSphere Application Server への Red Hat Process Automation Manager のサポートが、メンテナンスフェーズに入りました。Red Hat では、IBM WebSphere Application Server での Red Hat Process Automation Manager のサポートを継続しますが、以下の制限があります。

- Red Hat では、新しい認定またはソフトウェアの機能をリリースしません。
- Red Hat では、重大な影響を与えるセキュリティパッチおよびミッションクリティカルなバグ修正パッチが含まれる認定セキュリティパッチのみをリリースします。

今後、Red Hat ではハイブリッドクラウドストラテジーと互換性のある新しいプラットフォームや製品コンポーネントに移行するようにお願いする場合があります。

### 前提条件

- IBM WebSphere Application Server インスタンスのバージョン 9.0 以降がインストールされている。詳細なインストール手順は [IBM WebSphere Application Server 製品ページ](#) を参照してください。
- WebSphere 統合ソリューションコンソール (通常は <http://<HOST>:9060/ibm/console>) へのアクセスがある。

## 第44章 KIE SERVER

KIE Server は、Red Hat Process Automation Manager のルールおよびその他のアーティファクトの保存先および実行先のサーバーです。KIE Server はスタンドアロンで組み込みのコンポーネントで、REST、Java Message Service (JMS)、または Java クライアントサイドアプリケーションで利用可能なインターフェイスを介してルールをインスタンス化して実行し、ソルバーを介してプロセス、ジョブ、および Red Hat ビルドの OptaPlanner 機能を管理できます。

Web でデプロイ可能な WAR ファイルとして作成することで、KIE Server は Web コンテナであればどこにでもデプロイできます。KIE Server の現在のバージョンには、Red Hat Decision Manager および Red Hat Process Automation Manager の両方に対するデフォルトの拡張機能が含まれます。

KIE Server はメモリー消費が最小限でフットプリントが小さいため、クラウドインスタンスに簡単にデプロイできます。このサーバーの各インスタンスでは、複数のコンテナを開いてインスタンスを作成できるため、並行して複数のルールサービスを実行できます。

KIE Server は、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーと統合して、Red Hat Process Automation Manager のアプリケーション管理を合理化できます。

## 第45章 IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server は、Java ベースの Web アプリケーションをホストし、Java EE 認定ランタイム環境を提供する、柔軟性がある安全な Web アプリケーションです。IBM WebSphere 9.0 は Java SE 8 に対応しており、Java EE 7 に完全に準拠しています。

## 第46章 IBM WEBSHERE APPLICATION SERVER のインストールおよび実行

KIE Server に対応する多くの設定を適用するために、IBM WebSphere Application Server をインストールして実行する必要があります。本セクションは、IBM WebSphere をインストールして起動する方法を説明します。

インストールに関する最新の詳細説明は、[IBM Knowledge Center](#) を参照してください。

### 手順

1. IBM Installation Manager バージョン 1.8.5 以降を [IBM Installation Manager and Packaging Utility download links](#) ページからダウンロードします。IBM WebSphere のインストールには IBM Installation Manager が必要です。
2. ダウンロードしたアーカイブをデプロイメントし、作成されたディレクトリーで、root 権限で以下のコマンドを実行します。

```
sudo ./install
```

IBM Installation Manager が開きます。

3. **File** → **Preferences** の順に移動して、**Add Repository** をクリックします。
4. **Add Repository** ウィンドウに、IBM WebSphere 9.0 のリポジトリー URL を入力します。すべてのリポジトリー URL は、IBM Knowledge Center の [Online product repositories for IBM WebSphere Application Server offerings](#) ページで確認できます。
5. 端末で、インストール時に指定した IBM WebSphere Application Server ディレクトリーの場所に移動します。
6. **/bin** ディレクトリーに変更し、以下の例のようなコマンドを実行して、IBM WebSphere のプロファイル、ユーザー名、およびパスワードを作成します。プロファイルは、ランタイム環境を定義します。プロファイルには、ランタイム環境でサーバーが処理し、変更できるファイルがすべて含まれます。このユーザーはログインに必要になります。

```
sudo ./manageprofiles.sh -create -profileName testprofile -profilePath /profiles/testprofile -adminUserName websphere -adminPassword password123
```

7. 端末で、作成したプロファイルの **bin** ディレクトリー (例: **/profiles/testprofile/bin**) に移動し、以下のコマンドを実行して IBM WebSphere Application Server インスタンスを起動します。

```
sudo ./startServer.sh <SERVER_NAME>
```

**<SERVER\_NAME>** は、WebSphere 統合ソリューションコンソールの **Servers** → **Server Types** → **IBM WebSphere Application Servers** で定義した IBM WebSphere Application Server の名前に置き換えます。

8. Web ブラウザーで、以下の URL を開きます。

```
http://<HOST>:9060/ibm/console
```

**<HOST>** は、ターゲットサーバーの名前または IP アドレスです。

たとえば、システムで起動している IBM WebSphere のローカルインスタンスに対して WebSphere 統合ソリューションコンソールを起動する場合は、以下の URL を Web ブラウザーに入力します。

■ `http://localhost:9060/ibm/console`

9. WebSphere 統合ソリューションコンソールのログインページが表示されたら、管理者の認証情報を入力します。

## 第47章 KIE SERVER への IBM WEBSPPHERE APPLICATION SERVER の設定

IBM WebSphere Application Server に KIE Server をデプロイする前に、システムプロパティ、セキュリティ設定、JMS 要件、その他の IBM WebSphere プロパティを設定する必要があります。この設定により、KIE Server との最適な統合が促進されます。

### 前提条件

- IBM WebSphere Application Server をインストールして実行している。
- WebSphere 統合ソリューションコンソールにログインしている。

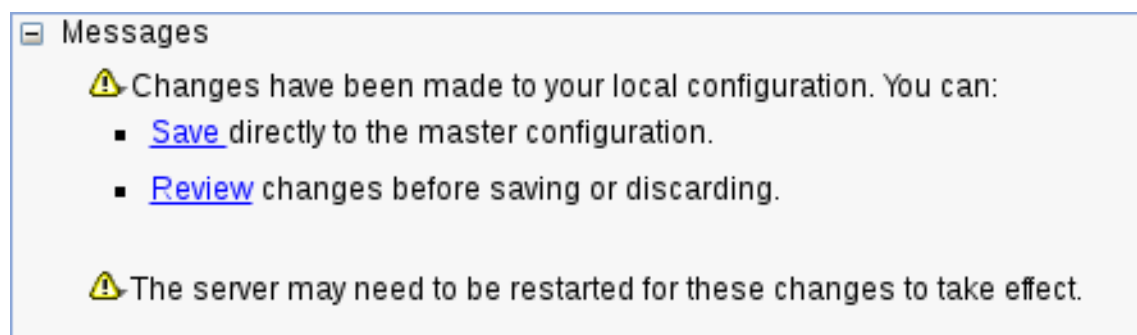
### 47.1. 管理セキュリティの有効化

WebSphere 統合ソリューションコンソールで管理セキュリティを有効にして、ユーザーおよびグループを作成するのに必要な権限を取得します。

#### 手順

1. WebSphere 統合ソリューションコンソールで、**Security** → **Global Security** の順にクリックし、**Enable Application Security** オプションが選択されていることを確認します。選択していても、サーバーレベルで上書きされている可能性があります。
2. **Security Configuration Wizard** をクリックし、**Next** をクリックします。
3. ユーザー情報を含むリポジトリを選択します。たとえば、ローカル設定で **Federated repositories** を選択します。
4. **Next** をクリックします。
5. **Primary administrative user name** および **Password** を入力します。
6. **Next** をクリックし、**Finish** をクリックします。
7. **Messages** ウィンドウで **Save** をクリックして、プライマリー設定への変更を保存します。

図47.1セキュリティ変更の保存



8. 端末で、インストール時に指定した IBM WebSphere Application Server の **/bin** ディレクトリーの場所に移動し、以下のコマンドを実行して IBM WebSphere を停止して再起動し、セキュリティの変更を適用します。

```
sudo ./stopServer.sh <SERVER_NAME>
```

```
sudo ./startServer.sh <SERVER_NAME>
```

<SERVER\_NAME> は、WebSphere 統合ソリューションコンソールの **Servers → Server Types → IBM WebSphere Application Servers** で定義した IBM WebSphere Application Server の名前に置き換えます。

## 47.2. IBM WEBSPHERE APPLICATION SERVER への JDBC データソースの設定

データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。IBM WebSphere Application Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。

通常、Red Hat Process Automation Manager を使用するソリューションは、1つのトランザクション内で複数のリソースを管理します。非同期のジョブ、イベント、タイマーなどの JMS。Red Hat Process Automation Manager では、データの原子性と一貫した結果を保証するために、可能な限りデータソースに XA ドライバーを必要とします。異なるスキーマのトランザクションコードがリスナー内に存在する場合や、jBPM エンジンが提供するフックから取得する場合は、XA ドライバーも必要となります。

1つのトランザクションに複数のリソースが参加していないことが確認できない限り、XA 以外のデータソースを使用しないでください。

### 前提条件

- データベース接続を作成するのに使用する JDBC プロバイダーが、データソースをデプロイするすべてのサーバーに設定されている。JDBC プロバイダーの詳細は、IBM Knowledge Center の [Configuring a JDBC provider](#) を参照してください。

### 手順

1. Red Hat カスタマーポータル の [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons をダウンロードします。
3. 以下の手順を実行して、データベースを準備します。
  - a. **TEMP\_DIR** などの一時ディレクトリーに **rhpm-7.13.5-add-ons.zip** を展開します。
  - b. **TEMP\_DIR/rhpm-7.13.5-migration-tool.zip** を展開します。
  - c. 現在のディレクトリーから、**TEMP\_DIR/rhpm-7.13.5-migration-tool/ddl-scripts** ディレクトリーに移動します。このディレクトリーには、複数のデータベースタイプの DDL スクリプトが含まれています。
  - d. 使用するデータベースに、お使いのデータベースタイプの DDL スクリプトをインポートします。以下に例を示します。



```
psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
```



### 注記

PostgreSQL または Oracle を Spring Boot と併用する場合は、対応する Spring Boot の DDL スクリプト (`/ddl-scripts/oracle/oracle-springboot-jbpm-schema.sql` または `/ddl-scripts/postgresql/postgresql-springboot-jbpm-schema.sql`) をインポートする必要があります。

4. WebSphere 統合ソリューションコンソールで、**Resources** → **JDBC** → **Data sources** の順に移動します。
5. データソースを使用するアプリケーションでスコープを選択します。セル、ノード、クラスター、またはサーバーを選択できます。
6. **New** をクリックし、**Create a data source** ウィザードを開きます。
7. **Data source name** フィールドに一意の名前を入力し、**JNDI name** フィールドに JNDI (Java Naming and Directory Interface) 名を入力します。アプリケーションサーバーは JNDI 名を使用して、アプリケーションのリソース参照をこのデータソースにバインドします。複数のリソースタイプ (データソース、J2C 接続ファクトリー/JMS 接続ファクトリー) に割り当てる JNDI 名を重複させないでください。

同じスコープに、同じタイプの複数のリソースに割り当てる JNDI 名を重複させないでください。

8. プロバイダーを作成している場合は、**Next** をクリックして **Select an existing JDBC provider** を選択します。作成していない場合は **Create new JDBC provider** をクリックして、新しいプロバイダーの詳細を定義します。(JDBC プロバイダーは、データソースを追加するための前提条件であるため、事前に作成しておくことが推奨されます。)
9. CMP (container managed persistence) のエンタープライズ Bean がこのデータソースにアクセスする必要がある場合は、**Enter database specific properties for the data source** パネルで **Use this data source in container managed persistence (CMP)** をクリックします。これにより、その他のデータベース関連のプロパティーが書き込まれます。
10. 必要に応じて、データソースのセキュリティーエイリアスを設定します。すべての認証メソッドで **None** を選択するか、以下のいずれを選択できます。
  - **Component-managed authentication alias:** コンポーネントリソース参照の `res-auth` 値が **Application** の場合に使用する認証エイリアスを指定します。新しいエイリアスを定義するには、**Related Items** → **J2EE Connector Architecture (J2C) authentication data entries** に移動します。コンポーネント管理エイリアスは、データソース認証に対してアプリケーションに指定した ID およびパスワードの組み合わせとなります。したがって、データソースに設定したエイリアスは、アプリケーションコードのエイリアスと同一にする必要があります。
  - **Mapping-configuration alias:** コンポーネントリソース参照にログイン設定がない場合にのみ利用します。`res-auth` 値を **Container** に設定した場合は、ログイン設定およびコンポーネントリソース参照の関連プロパティーを指定することが、認証ストラテジーを定義するのに推奨される方法です。**DefaultPrincipalMapping** ログイン設定を指定する場合の関連プロパティーは JAAS - J2C 認証データエントリエイリアスです。
  - **Container-managed authentication alias:** コンポーネントリソース参照にログイン設定がない場合にのみ利用します。`res-auth` 値を **Container** に設定する場合は、ログイン設定

と、コンポーネントリソース参照の関連プロパティを指定することで、コンテナ管理認証ストラテジーが判断されます。

11. **Next** をクリックしてデータソースの情報を確認し、**Finish** をクリックして設定を保存してウィザードを終了します。

**Data sources** パネルには、新しい設定と、同じスコープに設定したその他のデータソースがテーブルに表示されます。

IBM WebSphere Application Server データソースの詳細は、IBM Knowledge Center の [Configuring a JDBC provider and data source](#) を参照してください。

## 47.3. JAVA MESSAGE SERVICE (JMS) の設定

Java Message Service (JMS) は、KIE Server が、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーとメッセージを交換するために使用する Java API です。KIE Server を経由して JMS メッセージを送受信するようにアプリケーションサーバーを設定し、2 台のサーバー間で連携されるようにします。

### 47.3.1. サービスバスを作成し、IBM WebSphere Application Server を追加します。

JMS を使用するためには、サービスバスを作成し、IBM WebSphere Application Server をメンバーとして追加します。

#### 手順

1. WebSphere 統合ソリューションコンソールで、**Service Integration** → **Buses** → **New** の順に移動します。
2. 新しいバス名を入力し、**Bus Security** オプションの選択を解除します。
3. **Next** をクリックしてから **Finish** をクリックし、サービスバスを作成します。
4. 作成しておいたサービスバスを選択します。
5. **Topology** で **Bus Members** → **Add** を選択します。
6. **Add a New Bus Member** ウィザードを使用し、IBM WebSphere Application Server と、永続性に関するメッセージストアのタイプを選択します。メッセージストアのプロパティを指定することもできます。
7. **Finish** をクリックし、バスメンバーを追加します。

### 47.3.2. JMS 接続ファクトリーの作成

KIE Server でメッセージングを有効にするには、メッセージの送受信に JMS 接続ファクトリーをいくつか作成する必要があります。

#### 前提条件

- IBM WebSphere Application Server にサービスバスを作成している。

#### 手順

1. WebSphere 統合ソリューションコンソールで、**Resources** → **JMS** → **Connection Factories** の順に移動します。

- 正しいスコープを選択し、**New** をクリックします。
- Default Messaging Provider** オプションを選択し、**OK** をクリックします。
- 以下に示す必須の接続ファクトリーに、接続ファクトリー名 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/cf/KIE.SERVER.REQUEST**) を入力して、**Bus Name** ドロップダウンリストからサービスバス名を選択します。残りのオプションはデフォルト値のままにします。
- Apply**、**Save** の順にクリックして プライマリー設定への変更を保存し、必要な各ファクトリーに上述の手順を繰り返します。

### 47.3.2.1. KIE Server の JMS 接続ファクトリー

以下の表は、KIE Server で JMS メッセージングを有効にするために必要な Java Message Service (JMS) 接続ファクトリーになります。

表47.1 KIE Server に必要な JMS 接続ファクトリー

| 名前                         | デフォルト値                            | 用途                       |
|----------------------------|-----------------------------------|--------------------------|
| <b>KIE.SERVER.REQUEST</b>  | <b>jms/cf/KIE.SERVER.REQUEST</b>  | 全要求を KIE Server へ送信      |
| <b>KIE.SERVER.RESPONSE</b> | <b>jms/cf/KIE.SERVER.RESPONSE</b> | KIE Server が生成する応答をすべて受信 |
| <b>KIE.SERVER.EXECUTOR</b> | <b>jms/cf/KIE.SERVER.EXECUTOR</b> | KIE Server エグゼキューターサービス  |

### 47.3.3. JMS キューの作成

JMS キューは、ポイントツーポイントメッセージング (point-to-point messaging) の宛先エンドポイントになります。KIE Server で JMS メッセージングを有効にするには JMS キューをいくつか作成する必要があります。

#### 前提条件

- IBM WebSphere Application Server にサービスバスを作成している。

#### 手順

- WebSphere 統合ソリューションコンソールで、**Resources** → **JMS** → **Queues** の順に移動します。
- 正しいスコープを選択し、**New** をクリックします。
- Default Messaging Provider** オプションを選択し、**OK** をクリックします。
- 以下に示す必須のキューに、キューの名前 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/KIE.SERVER.REQUEST**) を入力して、**Bus Name** ドロップダウンリストからサービスバスを選択します。
- Queue Name** ドロップダウンリストで **Create Service Integration Bus Destination** を選択して一意の識別子を入力し、事前に作成したバスメンバーを選択します。

6. **Apply**、**Save** の順にクリックして プライマリー設定への変更を保存し、必要な各キューに上述の手順を繰り返します。

#### 47.3.3.1. KIE Server 向けの JMS キュー

以下の表は、KIE Server で JMS メッセージングを有効にするために必要な Java Message Service (JMS) キューです。

表47.2 KIE Server に必要な JMS キュー

| 名前                         | デフォルト値                               | 用途                       |
|----------------------------|--------------------------------------|--------------------------|
| <b>KIE.SERVER.REQUEST</b>  | <code>jms/KIE.SERVER.REQUEST</code>  | 全要求を KIE Server へ送信      |
| <b>KIE.SERVER.RESPONSE</b> | <code>jms/KIE.SERVER.RESPONSE</code> | KIE Server が生成する応答をすべて受信 |
| <b>KIE.SERVER.EXECUTOR</b> | <code>jms/KIE.SERVER.EXECUTOR</code> | KIE Server エグゼキューターサービス  |

#### 47.3.4. JMS アクティベーション指定の作成

キューと、JMS を有効にするメッセージ駆動型 Bean との間にブリッジを行うには、JMS アクティベーション指定が必要です。

##### 前提条件

- IBM WebSphere Application Server にサービスバスを作成している。
- JMS キューを作成している。

##### 手順

1. WebSphere 統合ソリューションコンソールで、**Resources** → **JMS** → **Activation Specifications** の順に移動します。
2. 正しいスコープを選択し、**New** をクリックします。
3. **Default Messaging Provider** オプションを選択し、**OK** をクリックします。
4. 以下に示す必須のアクティベーション指定に、アクティベーション指定名 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/activation/KIE.SERVER.REQUEST**) を入力して、**Bus Name** ドロップダウンリストからサービスバス名を選択します。
5. **Destination Type** ドロップダウンリストで **Queue** を選択し、**Destination lookup** に対応するキューの名前 (例: **jms/KIE.SERVER.REQUEST**) を入力します。
6. **Apply**、**Save** の順にクリックして プライマリー設定への変更を保存し、必要な各アクティベーション指定に上述の手順を繰り返します。

##### 47.3.4.1. KIE Server の JMS アクティベーション仕様

以下の表は、KIE Server で JMS メッセージングを有効にするために必要な Java Message Service (JMS) アクティベーション仕様になります。

表47.3 KIE Server に必要な JMS アクティベーション仕様

| 名前                         | デフォルト値                                    | 用途                       |
|----------------------------|-------------------------------------------|--------------------------|
| <b>KIE.SERVER.REQUEST</b>  | <b>jms/activation/KIE.SERVER.REQUEST</b>  | 全要求を KIE Server へ送信      |
| <b>KIE.SERVER.RESPONSE</b> | <b>jms/activation/KIE.SERVER.RESPONSE</b> | KIE Server が生成する応答をすべて受信 |
| <b>KIE.SERVER.EXECUTOR</b> | <b>jms/activation/KIE.SERVER.EXECUTOR</b> | KIE Server エグゼキューターサービス  |

## 47.4. IBM WEBSPPHERE APPLICATION SERVER へのシステムプロパティの設定

KIE Server をデプロイする前に、IBM WebSphere Application Server に、本セクションに記載するシステムプロパティを設定します。

### 手順

1. WebSphere 統合ソリューションコンソールで、**Servers** → **Server Types** → **IBM WebSphere Application Servers** の順にクリックします。
2. アプリケーションサーバーのリストで、KIE Server をデプロイするサーバーを選択します。
3. **Server Infrastructure** で **Java and Process Management** → **Process Definition** をクリックします。

図47.2 WebSphere 設定ページ

The screenshot shows the 'Application servers' configuration page for 'server1'. The page is divided into several sections:

- General Properties:** Includes fields for Name (server1) and Node name (dhcp-4-116Node01). There are checkboxes for 'Run in development mode', 'Parallel start' (checked), and 'Start components as needed'. A dropdown menu for 'Access to internal server classes' is set to 'Allow'.
- Server-specific Application Settings:** Includes a dropdown for 'Classloader policy' (Multiple) and a dropdown for 'Class loading mode' (Classes loaded with parent class loader first).
- Container Settings:** A list of expandable settings including Session management, SIP Container Settings, Web Container Settings, Portlet Container Settings, EJB Container Settings, Container Services, and Business Process Services.
- Applications:** A list of expandable settings including Installed applications.
- Server messaging:** A list of expandable settings including Messaging engines, Messaging engine inbound transports, WebSphere MQ link inbound transports, and SIB service.
- Server Infrastructure:** A list of expandable settings including Java and Process Management (with sub-items Class loader, Process definition, and Process execution) and Administration (with sub-item Java SDKs).

At the bottom of the configuration area, there are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

4. **Additional Properties** で **Java Virtual Machine** をクリックします。

図47.3 プロセスの定義設定ページ

これにより、IBM WebSphere を起動するのに使用する JVM の設定プロパティが開きます。

5. **Initial heap size** および **Maximum heap size** の両方を **2048** に設定し、**Apply** をクリックして Java Virtual Machine (JVM) メモリーサイズを増やします。KIE Server はこの値でテストされています。JVM メモリーサイズを増やさないと、KIE Server のデプロイ時に IBM WebSphere Application Server がフリーズするかエラーが発生します。
6. **Additional Properties** で **Custom Properties** をクリックします。
7. **New** → **Custom JVM Properties** の順にクリックし、以下のプロパティを IBM WebSphere に追加します。

表47.4 KIE Server のシステムプロパティ

| 名前                                    | 値                              | 説明                                                    |
|---------------------------------------|--------------------------------|-------------------------------------------------------|
| <b>kie.server.jms.queues.response</b> | <b>jms/KIE.SERVER.RESPONSE</b> | KIE Server が使用する応答用の JMS キューの JNDI 名。                 |
| <b>org.kie.server.domain</b>          | <b>WSLogin</b>                 | JMS の使用時にユーザーの認証に使用される JAAS <b>LoginContext</b> ドメイン。 |

| 名前                                              | 値                                                                                        | 説明                                                                                                                                                                                                                                                    |
|-------------------------------------------------|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>org.kie.server.persistence.ds</code>      | <code>jdbc/jbpm</code>                                                                   | KIE Server のデータソースの JNDI 名。                                                                                                                                                                                                                           |
| <code>org.kie.server.persistence.tm</code>      | <code>org.hibernate.engine.transaction.jta.platform.internal.WebSphereJtaPlatform</code> | Hibernate プロパティを設定するためのトランザクションマネージャープラットフォーム。                                                                                                                                                                                                        |
| <code>org.kie.server.persistence.dialect</code> | 例:<br><code>org.hibernate.dialect.H2Dialect</code>                                       | 使用する Hibernate 方言を指定します。データソースに従って設定します。                                                                                                                                                                                                              |
| <code>org.kie.executor.jms.queue</code>         | <code>jms/KIE.SERVER.EXECUTOR</code>                                                     | KIE Server へのジョブエグゼキューターの JMS キュー。                                                                                                                                                                                                                    |
| <code>org.kie.executor.jms.cf</code>            | <code>jms/cf/KIE.SERVER.EXECUTOR</code>                                                  | KIE Server へのジョブエグゼキューターの JMS 接続ファクトリー。                                                                                                                                                                                                               |
| <code>org.kie.server.router</code>              | 例: <code>http://localhost:9000</code>                                                    | (オプション) クラスター化された KIE Server 環境で、アプリケーションサーバーがその一部である 1 つ以上の KIE Server ルーター (Smart Router) の URL を 1 つ以上指定します。                                                                                                                                       |
| <code>org.jboss.logging.provider</code>         | <code>jdk</code>                                                                         | このプロパティは、 <b>CA SiteMinder TAI (SMTAI)</b> が環境にインストールされている場合にのみ必要になります。このプロパティを使用すると、Dashbuilder でログを記録するために、Hibernate が <b>log4j</b> ではなく、 <b>JDK</b> を強制的に使用しようとします。 <b>CA SiteMinder TAI (SMTAI)</b> には古いバージョンの <b>log4j</b> が含まれており、これにより競合が生じます。 |

8. **保存** をクリックして、プライマリー設定への変更を保存します。

## 47.5. IBM WEBSHERE APPLICATION SERVER の停止および再起動

IBM WebSphere Application Server に必要なシステムプロパティをすべて設定したあと、IBM サーバーを停止および再起動して、設定が適用されていることを確認します。



## 手順

端末で、インストール時に指定した IBM WebSphere Application Server の **/bin** ディレクトリーの場所に移動し、以下のコマンドを実行して IBM WebSphere を停止して再起動し、設定変更を適用します。

```
sudo ./stopServer.sh <SERVER_NAME>
```

```
sudo ./startServer.sh <SERVER_NAME>
```

**<SERVER\_NAME>** は、WebSphere 統合ソリューションコンソールの **Servers** → **Server Types** → **IBM WebSphere Application Servers** で定義した IBM WebSphere Application Server の名前に置き換えます。

## 第48章 IBM WEBSPHERE APPLICATION SERVER への KIE SERVER のインストールおよび実行

IBM WebSphere Application Server に必要なシステムプロパティをすべて設定したら、IBM WebSphere に KIE Server をインストールして、Red Hat Process Automation Manager アプリケーション管理を合理化します。

### 前提条件

- [47章 KIE Server への IBM WebSphere Application Server の設定](#) の説明どおりに IBM WebSphere Application Server インスタンスを設定している。

### 手順

1. Red Hat カスタマーポータル [の Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. **Red Hat Process Automation Manager 7.13.5 KIE Server for All Supported EE7 Containers** をダウンロードします。
3. **rhnam-7.13.5-kie-server-ee7.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
4. **kie-server.war** ディレクトリーを再パッケージ化します。
  - a. **TEMP\_DIR/rhnam-7.13.5-kie-server-ee7/kie-server.war** ディレクトリーに移動します。
  - b. **TEMP\_DIR/rhnam-7.13.5-kie-server-ee7/kie-server.war** ディレクトリーの内容を選択し、**kie-server.zip** ファイルを作成します。
  - c. **kie-server.zip** の名前を **kie-server.war** に変更します。このファイルを使用して、KIE Server をデプロイします。
  - d. 必要に応じて、新しい **kie-server.war** ファイルを、デプロイしやすい便利な場所にコピーします。
5. WebSphere 統合ソリューションコンソールで、**Applications → Application Types → WebSphere Enterprise Applications** の順に移動します。
6. **InstCII** をクリックします。
7. 再パッケージ化した **kie-server.war** ファイルに移動して選択し、アップロードします。
8. **Fast Path** を選択し、**Next** をクリックします。**Install New Application** ウィザードが開きます。
9. **Application Name** を **kie-server** に変更し、**Next** をクリックします。
10. 要件に合わせて KIE Server モジュールをサーバーにマッピングし、**Next** をクリックします。
11. **Bind Listeners for Message-Driven Beans** の場合は、両 Bean に **Activation Specification** を選択し、**Target Resource JNDI Name** フィールドに **jms/activation/KIE.SERVER.REQUEST**

を入力し、**KIE.SERVER.REQUEST** 接続ファクトリーの **jms/cf/KIE.SERVER.REQUEST** JNDI 名を入力します。

12. **Map Virtual Hosts for Web Modules** セクションでは、デフォルト値をそのままにして、**Next** をクリックします。
13. コンテキストのルートをも **kie-server** に設定します。
14. **Metadata for Modules** セクションで、デフォルト値をそのままにして **Next** をクリックします。
15. **Finish** をクリックして KIE Server をインストールし、**Save** をクリックしてプライマリー設定に変更を保存します。

## 48.1. KIE SERVER のグループおよびロールの作成

KIE Server をインストールしたら、**kie-server** グループと、ユーザーを作成します。

### 前提条件

- KIE Server が IBM WebSphere Application Server インスタンスにインストールされている。

### 手順

1. WebSphere 統合ソリューションコンソールで、**Users and Groups** → **Manage Groups** の順にクリックします。
2. **Manage Groups** 画面で **Create** をクリックします。
3. **Create a Group** 画面の **Group name** ボックスに **kie-server** と入力し、**Create** をクリックします。
4. **kie-server** グループに追加するユーザーを作成するには、**Users and Groups** → **Manage Users** の順にクリックします。
5. **Create a User** セクションで、必要な情報を入力します。
6. **Group Membership** をクリックします。
7. **Group Membership** 画面で **kie-server** をクリックして **Mapped To** に移動し、**Close** をクリックします。
8. **Create a User** 画面で **Create** をクリックします。

## 48.2. KIE SERVER のグループおよびロールのマッピング

KIE Server をインストールしたら、WebSphere 統合ソリューションコンソールで **kie-server** ロールを **kie-server** グループにマッピングして、KIE Server を実行します。

### 前提条件

- KIE Server が IBM WebSphere Application Server インスタンスにインストールされている。
- IBM WebSphere Application Server に、1人以上のユーザーが追加されている **kie-server** グループがある。

## 手順

1. WebSphere 統合ソリューションコンソールで、**Applications** → **Application Types** → **WebSphere Enterprise Applications** に移動し、新たにインストールした **kie-server** アプリケーションを選択します。
2. **Detail Properties** で **Security Role to User/Group Mapping** をクリックします。
3. **kie-server** ロールを選択し、**Map Groups** をクリックして、**kie-server** グループを検索します。
4. **kie-server** グループを、**Available** リストから **Selected** リストに移動し、**OK** をクリックします。  
このマッピングにより、IBM WebSphere Application Server の **kie-server** グループのユーザーに、KIE Server へのアクセスが付与されます。
5. **Save** をクリックしてマッピングを完了します。

### 48.3. KIE SERVER のクラスローディングの設定

KIE Server をインストールしたら、親クラスを最後に読み込むように、クラスローディングを設定する必要があります。

## 手順

1. **Applications** → **Application Types** → **WebSphere Enterprise Applications** に移動し、**kie-server** をクリックします。
2. 左側の **Detail Properties** の見出しにある **Class Loading and Update Detection** をクリックします。
3. プロパティで、**Class Loader Order** を **Classes loaded with local class loader first (parent last)** に、**WAR Class Loader Policy** を **Single class loader for application** に変更します。
4. 変更をプライマリー設定に保存します。

### 48.4. インストールの検証

KIE Server をインストールして KIE Server グループマッピングを定義したら、サーバーが実行していることを確認します。

## 前提条件

- KIE Server が IBM WebSphere Application Server インスタンスにインストールされている。
- ヘッドレス Process Automation Manager コントローラーに必要なシステムプロパティをすべて設定している。
- IBM WebSphere Application Server に KIE Server グループマッピングを定義している。

## 手順

1. サーバーが実行していることを確認するには、以下のタスクのいずれかを実行します。
  - KIE Server の URL (**http://<HOST>:<PORT>/kie-server**) に移動します。

- **http://<HOST>:<PORT>/kie-server/services/rest/server** に **GET** 要求を送信し、KIE Server REST API が応答するかどうかを確認します。  
上記の例では、以下のプレースホルダーを置き換えてください。
    - **<HOST>** は、ヘッドレスの Process Automation Manager コントローラーの ID または名前で、たとえば、**localhost** または **192.7.8.9** です。
    - **<PORT>** は、KIE Server ホストのポート番号に置き換えます (例: **9060**)。
2. KIE Server が実行していない場合は、IBM WebSphere Application Server インスタンスを停止して再起動し、KIE Server の URL または API に再度アクセスしてみてください。

## 第49章 IBM WEBSPHERE APPLICATION SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行

KIE Server REST API または Java Client API を使用して KIE Server に接続するには、IBM WebSphere Application Server にヘッドレス Process Automation Manager コントローラーをインストールします。ヘッドレス Process Automation Manager コントローラーは、一元化された方法で KIE Server 設定を管理するため、このコントローラーを使用してコンテナの作成および維持を行い、サーバーレベルのタスクを実行できます。



### 注記

実稼働環境で最適な結果を得るには、KIE Server とヘッドレス Process Automation Manager コントローラーを別のサーバーにインストールします。開発環境の場合は、KIE Server とヘッドレス Process Automation Manager コントローラーを同じサーバーにインストールします。

### 前提条件

- [47章 KIE Server への IBM WebSphere Application Server の設定](#) の説明どおりに IBM WebSphere Application Server インスタンスを設定している。
- KIE Server が IBM WebSphere Application Server インスタンスにインストールされている。
- インストールを完了するのに必要なユーザーパーミッションがある。

### 手順

1. Red Hat カスタマーポータル [の Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons をダウンロードします。
3. ダウンロードした **rhpmam-7.13.5-add-ons.zip** ファイルを一時ディレクトリーに展開します。
4. **rhpmam-7.13.5-add-ons.zip** ファイルを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP\_DIR** とします。
5. **rhpmam-7.13.5-add-ons/rhpmam-7.13.5-controller-ee7.zip** ファイルを展開します。
6. **controller.war** ディレクトリーを再パッケージ化します。
  - a. **TEMP\_DIR/rhpmam-7.13.5-add-ons/rhpmam-7.13.5-controller-ee7/controller.war** ディレクトリーに移動します。
  - b. **TEMP\_DIR/rhpmam-7.13.5-add-ons/rhpmam-7.13.5-controller-ee7/controller.war** ディレクトリーの内容を選択して、**controller.zip** ファイルを作成します。
  - c. **controller.zip** の名前を **controller.war** に変更します。これは、ヘッドレス Process Automation Manager コントローラーをデプロイするのに使用するファイルです。

- d. 必要に応じて、新しい **controller.war** ファイルを、デプロイしやすい場所にコピーします。
7. WebSphere 統合ソリューションコンソールで、**Applications** → **Application Types** → **WebSphere Enterprise Applications** の順に移動します。
8. **Install** をクリックします。
9. 再パッケージ化した **controller.war** ファイルに移動し、選択してアップロードします。
10. **Fast Path** を選択し、**Next** をクリックします。  
**Install New Application** ウィザードが開きます。
11. **Application Name** を **controller** に変更し、**Next** をクリックします。
12. 要件に合わせてヘッドレス Process Automation Manager コントローラーモジュールをサーバーにマッピングし、**Next** をクリックします。
13. **Bind Listeners for Message-Driven Beans** の場合は、両 Bean に **Activation Specification** を選択し、**Target Resource JNDI Name** フィールドに **jms/activation/KIE.SERVER.REQUEST** を入力し、**KIE.SERVER.REQUEST** 接続ファクトリーの **jms/cf/KIE.SERVER.REQUEST** JNDI 名を入力します。
14. **Map Virtual Hosts for Web Modules** セクションでは、デフォルト値をそのままにして、**Next** をクリックします。
15. コンテキストルートを **controller** に設定します。
16. **Metadata for Modules** セクションで、デフォルト値をそのままにして **Next** をクリックします。
17. **Finish** をクリックしてヘッドレス Process Automation Manager コントローラーをインストールし、**Save** をクリックしてプライマリー設定への変更を保存します。

## 49.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーへのクラスローディングの設定

ヘッドレス Process Automation Manager コントローラーをインストールしたら、親クラスを最後にロードするように、KIE Server クラスローディングを設定する必要があります。

### 手順

1. **Applications** → **Application Types** → **WebSphere Enterprise Applications** に移動し、**kie-server** をクリックします。
2. 左側の **Detail Properties** の見出しにある **Class Loading and Update Detection** をクリックします。
3. プロパティーで、**Class Loader Order** を **Classes loaded with local class loader first (parent last)** に、**WAR Class Loader Policy** を **Single class loader for application** に変更します。
4. 変更をプライマリー設定に保存します。

## 49.2. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティーの設定

ヘッドレス Process Automation Manager コントローラーをインストールしたら、本セクションで紹介するシステムプロパティをアプリケーションサーバーに設定して、ヘッドレス Process Automation Manager コントローラーとの適切な接続を有効にします。



## 注記

実稼働環境で最適な結果を得るには、KIE Server とヘッドレス Process Automation Manager コントローラーを別のサーバーにインストールします。開発環境の場合は、KIE Server とヘッドレス Process Automation Manager コントローラーを同じサーバーにインストールします。いずれの場合も、ヘッドレス Process Automation Manager コントローラーがインストールされているすべてのアプリケーションサーバーでこのプロパティを変更します。

## 前提条件

- KIE Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。

## 手順

1. ヘッドレス Process Automation Manager コントローラーがインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティ値を指定します。

表49.1 ヘッドレス Process Automation Manager コントローラーに必要なプロパティ

| 名前                         | 要件                                              |
|----------------------------|-------------------------------------------------|
| <b>org.kie.server.user</b> | <b>kie-server</b> ロールが割り当てられているユーザー             |
| <b>org.kie.server.pwd</b>  | <b>org.kie.server.user</b> プロパティに指定したユーザーのパスワード |

2. KIE Server がインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティ値を指定します。

表49.2 ヘッドレス Process Automation Manager コントローラーがインストールされている場合に KIE Server に必要なプロパティ

| 名前                                    | 要件                                                                                           |
|---------------------------------------|----------------------------------------------------------------------------------------------|
| <b>org.kie.server.controller.user</b> | <b>kie-server</b> ロールが割り当てられているユーザー                                                          |
| <b>org.kie.server.controller.pwd</b>  | <b>org.kie.server.controller.user</b> プロパティに指定したユーザーのパスワード                                   |
| <b>org.kie.server.id</b>              | KIE Server インストールの ID または名前 ( <b>rhdm700-decision-server-1</b> など)                           |
| <b>org.kie.server.location</b>        | KIE Server の URL ( <b>http://&lt;HOST&gt;:&lt;PORT&gt;/kie-server/services/rest/server</b> ) |



| 名前                               | 要件                                                                                                                   |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>org.kie.server.controller</b> | ヘッドレス Process Automation Manager コントローラーの URL ( <b>http://&lt;HOST&gt;:&lt;PORT&gt;/controller/rest/controller</b> ) |

<HOST> は、KIE Server ホストの ID または名前に置き換えます (例: **localhost** または **192.7.8.9**)。

<PORT> は、KIE Server ホストのポート番号に置き換えます (例: **7001**)。

### 49.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーグループおよびロールのマッピング

ヘッドレス Process Automation Manager コントローラーをインストールしたら、WebSphere 統合ソリューションコンソールに **kie-server** グループを作成し、**kie-server** ロールを **kie-server** グループにマッピングします。

#### 前提条件

- ヘッドレス Process Automation Manager コントローラーが IBM WebSphere Application Server インスタンスにインストールされている。

#### 手順

1. WebSphere 統合ソリューションコンソールで、[KIE Server グループおよびロールの作成](#)の記載通りに、**kie-server** グループと、このグループのユーザーを作成します。
2. **Applications** → **Application Types** → **WebSphere Enterprise Applications** に移動し、新たにインストールしたヘッドレス Process Automation Manager コントローラーを選択します。
3. **Detail Properties** で **Security Role to User/Group Mapping** をクリックします。
4. **kie-server** ロールを選択し、**Map Groups** をクリックして、**kie-server** グループを検索します。
5. **kie-server** グループを、**Available** リストから **Selected** リストに移動し、**OK** をクリックします。  
このマッピングにより、事前に作成した管理者ユーザーに Process Automation Manager へのアクセスが付与されます。
6. **Save** をクリックしてマッピングを完了します。

### 49.4. インストールの検証

ヘッドレス Process Automation Manager コントローラーをインストールし、必要なシステムプロパティとロール要件をアプリケーションサーバーに定義したら、ヘッドレス Process Automation Manager コントローラーが正しく動作していることを確認します。

#### 前提条件

- KIE Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。
- アプリケーションサーバーで、ヘッドレス Process Automation Manager コントローラーに必要なシステムプロパティおよびロール要件をすべて設定している。

## 手順

端末で以下のコマンドを実行して、ヘッドレス Process Automation Manager コントローラーが動作していることを確認します。

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```

<HOST> は、ヘッドレスの Process Automation Manager コントローラーホストの ID または名前です。たとえば、**localhost** または **192.7.8.9** です。

<PORT> は、ヘッドレスの Process Automation Manager コントローラーホストのポート番号で、たとえば、**7001** です。

<CONTROLLER> および <CONTROLLER\_PWD> は、本セクションで作成したユーザーの認証情報です。

このコマンドにより、KIE Server インスタンスに関する情報が返されます。



### 注記

あるいは、KIE Server Java API Client を使用して、ヘッドレス Process Automation Manager コントローラーにアクセスすることもできます。

ヘッドレス Process Automation Manager コントローラーが実行していない場合は、アプリケーションサーバーインスタンスを停止して再起動し、ヘッドレス Process Automation Manager コントローラーの URL または API へ再度アクセスしてみてください。

## 第50章 IBM WEBSPPHERE APPLICATION SERVER での埋め込みのプロセスエンジンとデシジョンエンジンの設定

埋め込みエンジンとは、デシジョンとビジネスプロセスを実行できるようにする軽量のワークフローおよびルールエンジンです。埋め込みエンジンは、Red Hat Process Automation Manager アプリケーションに含めるか、OpenShift、Kubernetes および Docker 経由でサービスとしてデプロイできます。また、API を使用して Red Hat Process Automation Manager にエンジンを埋め込むか、contexts and dependency injection (CDI) サービスセットの一部として、埋め込むことができます。

Red Hat Process Automation Manager アプリケーションで埋め込みされたエンジンを使用する予定の場合には、Red Hat Business Automation 部品表 (BOM) ファイルをプロジェクトの **pom.xml** ファイルに追加して、プロジェクトに、Maven の依存関係を追加する必要があります。Red Hat Business Automation BOM は、Red Hat Decision Manager と Red Hat Process Automation Manager の両方に適用されます。Red Hat Business Automation BOM (Bill of Materials) に関する詳細情報は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) を参照してください。

### 手順

1. Red Hat Business Automation BOM を **pom.xml** ファイルで宣言します。

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.redhat.ba</groupId>
      <artifactId>ba-platform-bom</artifactId>
      <version>7.13.5.redhat-00002</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
<dependencies>
  <!-- Your dependencies -->
</dependencies>
```

2. **<dependencies>** タグでお使いのプロジェクトに必要な依存関係を宣言します。製品の BOM をプロジェクトにインポートしたら、ユーザー向け製品依存関係のバージョンが定義されるため、**<dependency>** 要素のサブ要素 **<version>** を指定する必要はありません。ただし、**<dependency>** 要素を使用して、プロジェクトで使用する依存関係を宣言する必要があります。

- 標準的な Red Hat Process Automation Manager プロジェクトでは、使用する機能に応じて、以下の依存関係を宣言します。

### 埋め込みプロセスエンジンの依存関係

```
<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
```

```

    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-flow</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-flow-builder</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-bpmn2</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-runtime-manager</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-persistence-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-query-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-audit</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-kie-services</artifactId>
  </dependency>

  <!-- Dependency needed for default WorkItemHandler implementations. -->
  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-workitems-core</artifactId>
  </dependency>

  <!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
  <dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>${logback.version}</version>
  </dependency>

```

- CDI を使用する Red Hat Process Automation Manager プロジェクトでは、通常、以下の依存関係を宣言します。

### CDI が有効化されたプロセスエンジンの依存関係

```

<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>

```

### 埋め込みデシジョンエンジン依存関係

```

<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>

```

- KIE Server を使用するには、以下の依存関係を宣言します。

### クライアントアプリケーション KIE Server の依存関係

```

<dependency>
  <groupId>org.kie.server</groupId>

```

```
<artifactId>kie-server-client</artifactId>
</dependency>
```

- Red Hat Process Automation Manager にリモートクライアントを作成するには、以下の依存関係を宣言します。

### クライアントの依存関係

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- ルール、プロセス定義など、アセットを含む JAR ファイルを作成する場合は、お使いの Maven プロジェクトのパッケージングの種類を **kjar** と指定し、**org.kie:kie-maven-plugin** を使用して、**<project>** 要素に置かれた **kjar** パッケージングタイプを処理します。以下の例の **\${kie.version}** は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) に記載されている Maven ライブラリーのバージョンです。

```
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>
```

- プロジェクトで永続サポートのあるプロセスエンジンまたはデシジョンエンジンを使用する場合には、Red Hat Business Automation BOM ファイルから **version.org.hibernate-4ee7** プロパティをコピーして、**pom.xml** ファイルの **dependencyManagement** セクションに以下の Hibernate の依存関係を宣言する必要があります。

### Hibernate の依存関係

```
<!-- hibernate dependencies -->
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

■

## 第51章 キーストアを使用したパスワードセキュリティの確保

キーストアを使用して、Business Central と KIE Server の間の通信に使用するパスワードを暗号化できます。コントローラーと KIE Server のパスワードを暗号化する必要があります。Business Central と KIE Server を別のアプリケーションサーバーにデプロイする場合は、いずれのアプリケーションサーバーもキーストアを使用する必要があります。

Java Cryptography Extension KeyStore (JCEKS) は、対称鍵をサポートするため、キーストアには JCEKS を使用してください。



### 注記

KIE Server が JCEKS で設定されていない場合、KIE Server のパスワードはシステムプロパティにプレーンテキスト形式で保存されます。

### 前提条件

- KIE Server が IBM WebSphere Application Server にインストールされている。
- 「[KIE Server のグループおよびロールの作成](#)」の記載通りに、**kie-server** ロールを割り当てた KIE Server ユーザーが作成されている。
- Java 8 以降がインストールされている。

### 手順

1. JCEKS キーストアを作成します。
2. プロンプトが表示されたら、作成した KIE Server ユーザーのパスワードを入力します。
3. 以下の表に記載のシステムプロパティを設定します。

表51.1 KIE Server JCEKS を読み込む時に使用するシステムプロパティ

| システムプロパティ                            | プレースホルダー            | 説明                                                                      |
|--------------------------------------|---------------------|-------------------------------------------------------------------------|
| <b>kie.keystore.keyStoreURL</b>      | <KEYSTORE_URL>      | 使用する JCEKS の URL (例: <b>file:///home/kie/keystores/keystore.jceks</b> ) |
| <b>kie.keystore.keyStorePwd</b>      | <KEYSTORE_PWD>      | JCEKS のパスワード                                                            |
| <b>kie.keystore.key.server.alias</b> | <KEY_SERVER_ALIAS>  | パスワードの保存先となる REST サービスのキーのエイリアス                                         |
| <b>kie.keystore.key.server.pwd</b>   | <KEY_SERVER_PWD>    | 保存したパスワードを使用する REST サービスのエイリアスのパスワード                                    |
| <b>kie.keystore.key.ctrl.alias</b>   | <KEY_CONTROL_ALIAS> | パスワードの保存先のデフォルトの REST Process Automation Controller のキーのエイリアス           |



| システムプロパティ                        | プレースホルダー                       | 説明                                                                    |
|----------------------------------|--------------------------------|-----------------------------------------------------------------------|
| <b>kie.keystore.key.ctrl.pwd</b> | <b>&lt;KEY_CONTROL_PWD&gt;</b> | 保存したパスワードを使用する、デフォルトの REST Process Automation Controller のエイリアスのパスワード |

4. KIE Server を起動して、設定を検証します。

## 第52章 次のステップ

- [デシジョンサービスのスタートガイド](#)
- [ガイド付きデシジョンテーブルを使用したデシジョンサービスの作成](#)

## パート VI. ORACLE WEBLOGIC SERVER への KIE SERVER のインストールおよび設定

システム管理者は、Red Hat KIE Server に Oracle WebLogic Server を設定し、Oracle サーバーインスタンスに KIE Server をインストールできます。



### 注記

Oracle WebLogic Server への Red Hat Process Automation Manager のサポートが、メンテナンスフェーズに入りました。Red Hat では、Oracle WebLogic Server での Red Hat Process Automation Manager のサポートを継続しますが、以下の制限があります。

- Red Hat では、新しい認定またはソフトウェアの機能をリリースしません。
- Red Hat では、重大な影響を与えるセキュリティーパッチおよびミッションクリティカルなバグ修正パッチが含まれる認定セキュリティーパッチのみをリリースします。

今後、Red Hat ではハイブリッドクラウドストラテジーと互換性のある新しいプラットフォームや製品コンポーネントに移行するようにお願いする場合があります。

### 前提条件

- Oracle WebLogic Server インスタンスのバージョン 12.2.1.3.0 以降がインストールされている。詳細なインストール手順は [Oracle WebLogic Server 製品ページ](#) を参照してください。
- Oracle WebLogic Server 管理コンソール (通常は <http://<HOST>:7001/console>) へのアクセスがある。

## 第53章 KIE SERVER

KIE Server は、Red Hat Process Automation Manager のルールおよびその他のアーティファクトの保存先および実行先のサーバーです。KIE Server はスタンドアロンで組み込みのコンポーネントで、REST、Java Message Service (JMS)、または Java クライアントサイドアプリケーションで利用可能なインターフェイスを介してルールをインスタンス化して実行し、ソルバーを介してプロセス、ジョブ、および Red Hat ビルドの OptaPlanner 機能を管理できます。

Web でデプロイ可能な WAR ファイルとして作成することで、KIE Server は Web コンテナであればどこにでもデプロイできます。KIE Server の現在のバージョンには、Red Hat Decision Manager および Red Hat Process Automation Manager の両方に対するデフォルトの拡張機能が含まれます。

KIE Server はメモリー消費が最小限でフットプリントが小さいため、クラウドインスタンスに簡単にデプロイできます。このサーバーの各インスタンスでは、複数のコンテナを開いてインスタンスを作成できるため、並行して複数のルールサービスを実行できます。

KIE Server は、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーと統合して、Red Hat Process Automation Manager のアプリケーション管理を合理化できます。

## 第54章 ORACLE WEBLOGIC SERVER

Oracle WebLogic Server は、分散型 Java アプリケーションを作成するために API の標準セットを提供する Java EE アプリケーションサーバーで、データベース、メッセージングサービス、外部のエンタープライズシステムへの接続など、さまざまなサービスにアクセスできます。ユーザークライアントは、Web ブラウザークライアントまたは Java クライアントを使用してこれらのアプリケーションにアクセスします。

## 第55章 ORACLE WEBLOGIC SERVER のインストールおよび実行

KIE Server に対応する多くの設定を適用するために、Oracle WebLogic Server をインストールして実行する必要があります。本セクションは、スタンドアロンの Oracle WebLogic Server ドメインに Oracle WebLogic Server をインストールして起動する方法を説明します。

最新の詳細情報は [Oracle WebLogic Server 製品ページ](#) を参照してください。



### 注記

起動するサーバーと同じリスナーポートを使用する Oracle WebLogic Server インスタンスを実行している場合は、新たにサーバーを起動する前に、稼働しているサーバーを停止する必要があります。

### 手順

1. Oracle WebLogic Server の [ダウンロードページ](#) から、Oracle WebLogic Server 12.2.1.3.0 以降をダウンロードします。
2. ターゲットシステムにサインインし、認定済み JDK がシステムにインストールされていることを確認します。インストーラーには認定済み JDK が必要です。システム要件は、[Oracle Fusion Middleware Systems Requirements and Specifications](#) を参照してください。JDK のダウンロードは、[Planning an Installation of Oracle Fusion Middleware](#) の About JDK Requirements for an Oracle Fusion Middleware Installation のセクションを参照してください。
3. インストールプログラムをダウンロードしたディレクトリーに移動します。
4. 以下の例のように、システムの JDK ディレクトリーから **java -jar** を実行して、インストールプログラムを起動します。  
UNIX ベースのオペレーティングシステムの場合は、以下のコマンドを入力します。

```
/home/Oracle/jdk/jdk1.8.0_131/bin/java -jar fmw_12.2.1.3.0_wls_generic.jar
```

Windows オペレーティングシステムの場合は、以下のコマンドを入力します。

```
C:\Program Files\Java\jdk1.8.0_131\bin\java -jar fmw_12.2.1.3.0_wls_generic.jar
```

この例の JDK の場所は、実際にシステムに保存されている JDK の場所に置き換えます。

5. インストールウィザードプロンプトに従い、インストールを完了します。
6. インストールが完了したら、**WLS\_HOME/user\_projects/<DOMAIN\_NAME>** に移動します。**<DOMAIN\_NAME>** はドメインディレクトリーに置き換えます。以下の例では、**mydomain** はドメインディレクトリーです。

```
WLS\user_projects\mydomain
```

7. 以下のいずれかのコマンドを実行して、Oracle WebLogic Server を起動します。  
UNIX ベースのオペレーティングシステムの場合は、以下のコマンドを入力します。

```
startWebLogic.sh
```

Windows オペレーティングシステムの場合は、以下のコマンドを入力します。

■

```
startWebLogic.cmd
```

起動スクリプトが多数のメッセージを表示しますが、最後に以下のようなメッセージを表示します。

```
<Dec 8, 2017 3:50:42 PM PDT> <Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

8. Web ブラウザーで、以下の URL を開きます。

```
http://<HOST>:<PORT>/console
```

このコマンドでは、以下のプレースホルダーを置き換えてください。

- **<HOST>** は、ホストサーバーのシステム名または IP アドレスに置き換えます。
- **<PORT>** は、ホストサーバーが要求をリッスンしているポートの番号 (デフォルトでは 7001) に置き換えます。

たとえば、システムで起動している Oracle WebLogic Server のローカルインスタンスに対して管理コンソールを起動するには、以下の URL をブラウザーに入力します。

```
http://localhost:7001/console/
```

SSL (secure socket layer) を使用して管理コンソールを起動したら、**http** の後に **s** を追加して、**https://<HOST>:<PORT>/console** にする必要があります。

9. WebLogic 管理コンソールのログインページが表示されたら、管理者の認証情報を入力します。

## 第56章 KIE SERVER への ORACLE WEBLOGIC SERVER の設定

Oracle WebLogic Server に KIE Server をデプロイする前に、システムプロパティー、セキュリティー設定、JMS 要件、その他の Oracle WebLogic Server プロパティーを設定する必要があります。この設定により、KIE Server との最適な統合が促進されます。

### 前提条件

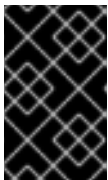
- Oracle WebLogic Server をインストールし、実行している。
- WebLogic 管理コンソールにログインしている。

### 56.1. KIE SERVER のグループおよびユーザーの設定

WebLogic 管理コンソールでユーザーを **kie-server** グループに割り当て、Oracle WebLogic Server で、コンテナ管理の認証メカニズムを有効にします。

#### 手順

1. WebLogic 管理コンソールで **Security Realms** に移動します。
2. 必要なセキュリティーレルムを選択するか、**New** をクリックして新しいセキュリティーレルムを作成します。
3. **Users and Groups** → **Groups** → **New** の順に移動して、**kie-server** グループを作成します。
4. **Users** → **New** の順に移動し、新しいユーザーを作成します。
5. **server-user** などのユーザーとそのパスワードを入力し、**OK** をクリックします。



#### 重要

選択したユーザー名が、ロールまたはグループの名前と競合しないようにします。たとえば、**kie-server** という名前のロールが存在する場合は、**kie-server** という名前のユーザーを作成しないでください。

6. 新たに作成したユーザーをクリックして、**Groups** タブに戻ります。
7. 選択ツールを使用して、**kie-server** グループを **Available** フィールドから **Chosen** フィールドに移動し、**Save** をクリックします。

### 56.2. ORACLE WEBLOGIC SERVER への JDBC データソースの設定

データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。Oracle WebLogic Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。

通常、Red Hat Process Automation Manager を使用するソリューションは、1つのトランザクション内で複数のリソースを管理します。非同期のジョブ、イベント、タイマーなどの JMS。Red Hat Process Automation Manager では、データの原子性と一貫した結果を保証するために、可能な限りデータソースに XA ドライバーを必要とします。異なるスキーマのトランザクションコードがリスナー内に存在する場合や、jBPM エンジンが提供するフックから取得する場合は、XA ドライバーも必要となります。



1つのトランザクションに複数のリソースが参加していないことが確認できない限り、XA 以外のデータソースを使用しないでください。

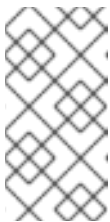
## 前提条件

- データベース接続を作成するのに使用する JDBC ドライバーが、データソースをデプロイするすべてのサーバーにインストールされている。一部の JDBC ドライバーが Oracle WebLogic Server にインストールされている (DB2、Informix、MS SQL Server、および Sybase 用の WebLogic ブランドの DataDirect JDBC ドライバーなど)。JDBC ドライバーの詳細は、Oracle Help Center の [Using JDBC Drivers with WebLogic Server](#) を参照してください。

## 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons をダウンロードします。
3. 以下の手順を実行して、データベースを準備します。
  - a. **TEMP\_DIR** などの一時ディレクトリーに **rhpm-7.13.5-add-ons.zip** を展開します。
  - b. **TEMP\_DIR/rhpm-7.13.5-migration-tool.zip** を展開します。
  - c. 現在のディレクトリーから、**TEMP\_DIR/rhpm-7.13.5-migration-tool/ddl-scripts** ディレクトリーに移動します。このディレクトリーには、複数のデータベースタイプの DDL スクリプトが含まれています。
  - d. 使用するデータベースに、お使いのデータベースタイプの DDL スクリプトをインポートします。以下に例を示します。

```
psql jbpmp < /ddl-scripts/postgresql/postgresql-jbpmp-schema.sql
```



### 注記

PostgreSQL または Oracle を Spring Boot と併用する場合は、対応する Spring Boot の DDL スクリプト (**/ddl-scripts/oracle/oracle-springboot-jbpmp-schema.sql** または **/ddl-scripts/postgresql/postgresql-springboot-jbpmp-schema.sql**) をインポートする必要があります。

4. WebLogic 管理コンソールで、**Change Center** → **Lock & Edit** の順に移動します。
5. **Domain Structure** で、**Services** → **Data Sources** の順に移動します。
6. **Summary of Data Sources** ページで、**New** → **Generic Data Source** の順にクリックします。
7. **JDBC Data Sources Properties** ページで、以下の情報を入力するか、選択します。
  - **Name:** この JDBC データソースの名前を入力します。この名前は設定ファイル (**config.xml**) と管理コンソールでこのデータソースを参照するのに使用されます。

- **JNDI Name:** JDBC データソースをバインドする JNDI バスを入力します。アプリケーションは、接続を予約する際に、この名前を使用して、JNDI ツリーでデータソースを検索します。
  - **Database Type:** 接続するデータベースの DBMS を選択します。DBMS が記載されていない場合は **Other** を選択します。
8. **Next** をクリックして、続行します。
  9. データベースに接続するのに使用する **データベースドライバ** を選択します。このリストには、選択した DBMS 用の一般的な JDBC ドライバと、すでにインストールされているその他の JDBC ドライバが含まれます。
  10. **Transaction Options** ページで、**Supports Global Transactions** オプションを選択したままにし、利用可能なトランザクションオプションから選択します。このチェックボックスの選択をはずし、このデータソースでグローバルトランザクションを無効(無視)することもできます。多くの場合、最適なデータ効率を得るために、このオプションは選択したままにする必要があります。
    - **Two-Phase Commit:** このオプションを選択すると、標準の XA 処理が有効になります。このオプションは、XA JDBC ドライバを選択して、データベース接続を作成する場合に限り使用できます。
    - **Logging Last Resource:** このオプションを選択すると、LLR (Logging Last Resource) トランザクションの最適化を使用するグローバルトランザクションに非 XA JDBC 接続を使用できるようになります。このオプションは、Emulate Two-Phase Commit よりも推奨されません。このオプションは、非 XA JDBC ドライバを選択してデータベース接続を作成する場合に限り使用できます。
    - **Emulate Two-Phase Commit:** このオプションを選択すると XA JDBC 接続が有効になり、JTA を使用する分散トランザクションでの使用がエミュレートされます。このオプションは、アプリケーションがヒューリスティック条件を容認する場合に限り選択します。このオプションは、非 XA JDBC ドライバを選択してデータベース接続を作成する場合に限り使用できます。
    - **One-Phase Commit:** このオプションを選択すると、グローバルトランザクションで使用する非 XA 接続のみをトランザクションに使用します。このオプションは、非 XA JDBC ドライバを選択してデータベース接続を作成する場合に限り使用できます。
  11. **Next** をクリックして、続行します。
  12. **Connection Properties** ページで、以下のプロパティに値を入力します。
    - **Service Name:** 接続するデータベースのサービス名を指定します。データソースが1つ以上提供されている場合は、各データソースに同じサービス名を指定する必要があります。このフィールドは、Oracle Real Application Clusters (RAC) に利用可能なサービスインスタンス接続ドライバが1つ選択されている場合に限り使用できます。
    - **Database Name:** 接続するデータベースの名前を入力します。データベースの名前要件は、JDBC ドライバおよび DBMS によって異なります。
    - **Host Name:** データベースをホストするサーバーの DNS 名または IP アドレスを入力します。Oracle GridLink サービスインスタンス接続を作成し、データソースを1つ以上提供している場合は、各データソースに対する接続は同じにする必要があります。
    - **Port:** 接続要求に対してデータベースサーバーがリッスンするポートを入力します。

- **Database User Name:** データソースの各接続で使用するデータベースのユーザーアカウント名を入力します。
- **Password/Confirm Password:** データベースユーザーアカウントのパスワードを入力します。
- **oracle.jdbc.DRCPConnectionClass:** お使いの環境で必要な場合は、オプションで、DCRP (Database Resident Connection Pooling) 接続クラスを入力します。

13. **Next** をクリックして、続行します。

14. **Test Database Connection** ページで接続パラメーターを確認して、**Test Configuration** をクリックします。

Oracle WebLogic Server は、管理サーバーからデータベースへの接続を作成しようとします。接続テストの結果がページ上部に表示されます。テストに失敗した場合は、設定エラーを修正して、再度テストしてみてください。

15. **Next** をクリックして続行します。選択した JDBC ドライバーが管理サーバーにインストールされていない場合は、この手順を省略できます。

16. **Select Targets** ページで、データソースをデプロイするサーバーまたはクラスターを選択し、**Finish** をクリックします。

17. WebLogic 管理コンソールのメインメニューに戻り、**Change Center** → **Activate Changes** の順に選択します。

Oracle WebLogic Server データソースの詳細は、Oracle Help Center の [JDBC Data Sources for Oracle WebLogic Server](#) を参照してください。

## 56.3. JAVA MESSAGE SERVICE (JMS) の設定

Java Message Service (JMS) は、KIE Server が、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーとメッセージを交換するために使用する Java API です。KIE Server を経由して JMS メッセージを送受信するようにアプリケーションサーバーを設定し、2 台のサーバー間で連携されるようにします。

### 56.3.1. JMS サーバーの作成

KIE Server および Oracle WebLogic Server で JMS を使用する JMS サーバーを作成します。

#### 手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Servers** の順に移動します。
2. **New** をクリックして、JMS サーバーを新たに作成します。
3. JMS サーバーの名前を入力して、**Next** をクリックします。
4. KIE Server デプロイメントに使用するターゲットサーバーを選択します。
5. **Finish** をクリックします。

### 56.3.2. JMS モジュールの作成

接続ファクトリー、キューなどの JMS リソースを格納する JMS モジュールを作成する必要があります。

## 前提条件

- JMS サーバーを作成している。

## 手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動します。
2. **New** をクリックしてモジュールを作成します。
3. モジュール名を入力し、**Next** をクリックします。
4. KIE Server デプロイメントに使用するターゲットサーバーを選択し、**Finish** をクリックします。
5. 新たに作成したモジュール名をクリックし、**Subdeployments** をクリックします。
6. **New** をクリックして、モジュールのサブデプロイメントを作成します。
7. サブデプロイメントの名前を入力し、**Next** をクリックします。
8. チェックボックスを選択して、事前に作成した JMS サーバーを選択します。
9. **Finish** をクリックして、サブデプロイメントの設定を完了します。

### 56.3.3. JMS 接続ファクトリーの作成

KIE Server でメッセージングを有効にするには、メッセージの送受信に JMS 接続ファクトリーをいくつか作成する必要があります。

## 前提条件

- JMS サーバーを作成している。
- JMS モジュールを作成している。

## 手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動して、JMS モジュールのリストを確認します。
2. 事前に作成したモジュールを選択し、**New** をクリックして新しい JMS リソースを作成します。
3. **Connection Factory** を選択し、**Next** をクリックします。
4. 以下の表に記載されている必須の各接続ファクトリーに、接続ファクトリー名 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/cf/KIE.SERVER.REQUEST**) を入力し、**Next** をクリックします。接続ファクトリーが、JMS モジュールにデフォルトで割り当てるサーバーを自動的に選択します。

表56.1 KIE Server に必要な JMS 接続ファクトリー

| 名前                         | デフォルト値                            | 用途                       |
|----------------------------|-----------------------------------|--------------------------|
| <b>KIE.SERVER.REQUEST</b>  | <b>jms/cf/KIE.SERVER.REQUEST</b>  | 全要求を KIE Server へ送信      |
| <b>KIE.SERVER.RESPONSE</b> | <b>jms/cf/KIE.SERVER.RESPONSE</b> | KIE Server が生成する応答をすべて受信 |
| <b>KIE.SERVER.EXECUTOR</b> | <b>jms/cf/KIE.SERVER.EXECUTOR</b> | KIE Server エグゼキューターサービス  |

5. **Finish** をクリックして接続ファクトリーを追加し、必要な各ファクトリーに対して上記手順を繰り返します。

### 56.3.4. JMS キューの作成

JMS キューは、ポイントツーポイントメッセージング (point-to-point messaging) の宛先エンドポイントになります。KIE Server で JMS メッセージングを有効にするには JMS キューをいくつか作成する必要があります。

#### 前提条件

- JMS サーバーを作成している。
- JMS モジュールを作成している。

#### 手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動して、JMS モジュールのリストを確認します。
2. 事前に作成したモジュールを選択し、**New** をクリックして新規 JMS リソースを作成します。
3. **Queue** を選択し、**Next** をクリックします。
4. 以下の表に示す必須の各キューに、キューの名前 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/KIE.SERVER.REQUEST**) を入力して、**Next** をクリックします。

表56.2 KIE Server に必要な JMS キュー

| 名前                         | デフォルト値                         | 用途                       |
|----------------------------|--------------------------------|--------------------------|
| <b>KIE.SERVER.REQUEST</b>  | <b>jms/KIE.SERVER.REQUEST</b>  | 全要求を KIE Server へ送信      |
| <b>KIE.SERVER.RESPONSE</b> | <b>jms/KIE.SERVER.RESPONSE</b> | KIE Server が生成する応答をすべて受信 |
| <b>KIE.SERVER.EXECUTOR</b> | <b>jms/KIE.SERVER.EXECUTOR</b> | KIE Server エグゼキューターサービス  |

5. JMS サーバーに接続する JMS モジュールのサブデプロイメントを選択します。
6. **Finish** をクリックしてキューを追加し、必要な各キューに上記手順を繰り返します。

## 56.4. ORACLE WEBLOGIC SERVER へのシステムプロパティーの設定

KIE Server をデプロイする前に、Oracle WebLogic Server に、本セクションに記載するシステムプロパティーを設定します。

### 手順

1. Java Virtual Machine (JVM) のメモリーサイズを増やすには、以下のシステムプロパティーを設定します。

```
USER_MEM_ARGS=-Xms512m -Xmx1024m
```

JVM メモリーサイズを増やさないと、KIE Server のデプロイ時に Oracle WebLogic Server がフリーズするか、エラーが発生します。

2. Oracle WebLogic Server インスタンスに、以下の KIE Server システムプロパティーを指定します。

表56.3 KIE Server のシステムプロパティー

| 名前                                        | 値                                                                      | 説明                                                    |
|-------------------------------------------|------------------------------------------------------------------------|-------------------------------------------------------|
| <b>kie.server.jms.queues.response</b>     | <b>jms/KIE.SERVER.RESPONSE</b>                                         | KIE Server が使用する応答用の JMS キューの JNDI 名。                 |
| <b>org.kie.server.domain</b>              | <b>OracleDefaultLoginConfiguration</b>                                 | JMS の使用時にユーザーの認証に使用される JAAS <b>LoginContext</b> ドメイン。 |
| <b>org.kie.server.persistence.ds</b>      | <b>jdbc/jbpm</b>                                                       | KIE Server のデータソースの JNDI 名。                           |
| <b>org.kie.server.persistence.tm</b>      | <b>org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform</b> | Hibernate プロパティーを設定するためのトランザクションマネージャープラットフォーム。       |
| <b>org.kie.server.persistence.dialect</b> | 例:<br><b>org.hibernate.dialect.H2Dialect</b>                           | 使用する Hibernate 方言を指定します。データソースに従って設定します。              |
| <b>org.kie.executor.jms.queue</b>         | <b>jms/KIE.SERVER.EXECUTOR</b>                                         | KIE Server へのジョブエグゼキューターの JMS キュー。                    |

| 名前                                   | 値                                       | 説明                                                                                                              |
|--------------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <code>org.kie.executor.jms.cf</code> | <code>jms/cf/KIE.SERVER.EXECUTOR</code> | KIE Server へのジョブエグゼキューターの JMS 接続ファクトリー。                                                                         |
| <code>org.kie.server.router</code>   | 例: <code>http://localhost:9000</code>   | (オプション) クラスター化された KIE Server 環境で、アプリケーションサーバーがその一部である 1 つ以上の KIE Server ルーター (Smart Router) の URL を 1 つ以上指定します。 |

3. **JAVA\_OPTIONS** 環境編集に同じプロパティ値を設定します。

```

JAVA_OPTIONS="-Dkie.server.jms.queues.response=jms/KIE.SERVER.RESPONSE
-Dorg.kie.server.domain=OracleDefaultLoginConfiguration
-Dorg.kie.executor.jms.cf=jms/cf/KIE.SERVER.EXECUTOR
-Dorg.kie.executor.jms.queue=jms/KIE.SERVER.EXECUTOR
-Dorg.kie.server.persistence.ds=jdbc/jbpm
-
Dorg.kie.server.persistence.tm=org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform

-Dorg.kie.server.persistence.dialect=org.hibernate.dialect.H2Dialect
// Optional server router, for clustered server environment
-Dorg.kie.server.router=http://localhost:9000

```

## 56.5. ORACLE WEBLOGIC SERVER の停止および再起動

Oracle WebLogic Server で必要なシステムプロパティをすべて設定したら、Oracle サーバーを停止して再起動し、設定が適用されていることを確認します。

### 手順

1. WebLogic 管理コンソールで、**Change Center** → **Lock & Edit** の順に移動します。
2. **Domain Structure** で **Environment** → **Servers** → **Control** の順にクリックします。
3. 停止するサーバーを選択して、**Shutdown** をクリックします。
4. **When Work Completes** を選択してサーバーを正常に停止するか、**Force Shutdown Now** を選択して継続中のタスクの完了を待たずに直ちにサーバーを停止します。
5. **Server Life Cycle Assistant** ペインで **Yes** をクリックして、シャットダウンを完了します。
6. シャットダウンが完了したら、端末でドメインディレクトリーに移動し、**WLS\_HOME/user\_projects/<DOMAIN\_NAME>** に移動します。以下に例を示します。

```
WLS\user_projects\mydomain
```

7. 以下のいずれかのコマンドを実行して Oracle WebLogic Server を再起動し、新しい設定を適用します。

UNIX ベースのシステムの場合:

```
startWebLogic.sh
```

Windows オペレーティングシステムの場合:

```
startWebLogic.cmd
```

8. Web ブラウザーで管理コンソール (例: <http://localhost:7001/console/>) を開き、認証情報を使用してログインします。



## 第57章 ORACLE WEBLOGIC SERVER への KIE SERVER のインストール

Oracle WebLogic Server に必要なシステムプロパティをすべて設定したら、Oracle WebLogic Server とともに KIE Server をインストールして、Red Hat Process Automation Manager アプリケーション管理を合理化します。

### 前提条件

- [56章 KIE Server への Oracle WebLogic Server の設定](#) の説明どおりに、Oracle WebLogic Server インスタンスが設定されている。

### 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. **Red Hat Process Automation Manager 7.13.5 KIE Server for All Supported EE7 Containers** をダウンロードします。
3. ダウンロードした **rhpmam-7.13.5-kie-server-ee7.zip** ファイルを一時ディレクトリーに展開します。
4. WebLogic 管理コンソールで **Deployments** に移動し、既存のアプリケーションをすべて表示します。
5. **Install** をクリックします。
6. **rhpmam-7.13.5-kie-server-ee7.zip** ファイルをダウンロードして展開した一時ディレクトリーに移動し、**rhpmam-7.13.5-kie-server-ee7/kie-server.war** に移動します。
7. **kie-server.war** ファイルを選択し、**Next** をクリックして続行します。
8. ターゲットとするスタイルとして **Install this deployment as an application** を選択し、**Next** をクリックします。
9. アプリケーション名を **kie-server** に設定し、セキュリティモデルを **DD Only** に設定します。残りのオプションはデフォルトのままにし、**Next** をクリックして続行します。
10. **Additional Configuration** セクションで **No, I will review the configuration later** を選択し、**Finish** をクリックします。

### 57.1. ORACLE WEBLOGIC SERVER への KIE SERVER インストールの確認

Oracle WebLogic Server に KIE Server をインストールしたら、インストールが成功していることを確認します。

### 前提条件

- [56章 KIE Server への Oracle WebLogic Server の設定](#) の説明どおりに、Oracle WebLogic Server インスタンスが設定されている。

- [57章 Oracle WebLogic Server への KIE Server のインストール](#) の記載通りに KIE Server がインストールされている。

## 手順

1. Web ブラウザーに KIE Server の URL (**http://<HOST>:<PORT>/kie-server/services/rest/server**) を入力します。
2. KIE Server が実行していることを確認します。  
KIE Server が実行していない場合は、Oracle WebLogic Server インスタンスを停止して再起動し、KIE Server の URL に再度アクセスしてみてください。

## 第58章 ORACLE WEBLOGIC SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行

KIE Server REST API または Java Client API を使用して KIE Server に接続するには、Oracle WebLogic Server にヘッドレス Process Automation Manager コントローラーをインストールします。ヘッドレス Process Automation Manager コントローラーは、一元化された方法で KIE Server 設定を管理するため、このコントローラーを使用してコンテナの作成および維持を行い、サーバーレベルのタスクを実行できます。

### 前提条件

- [56章 KIE Server への Oracle WebLogic Server の設定](#) の説明どおりに、Oracle WebLogic Server インスタンスが設定されている。
- KIE Server が Oracle WebLogic Server インスタンスにインストールされている。
- インストールを完了するのに必要なユーザーパーミッションがある。

### 手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
  - **製品:** Process Automation Manager
  - **バージョン:** 7.13.5
2. Red Hat Process Automation Manager 7.13.5 Add Ons をダウンロードします。
3. ダウンロードした **rhpmam-7.13.5-add-ons.zip** ファイルを一時ディレクトリーに展開します。
4. WebLogic 管理コンソールで、**Security Realms** → **Users and Groups** の順に移動します。
5. 事前に作成した **kie-server** グループに移動して、ヘッドレス Process Automation Manager コントローラーのユーザー (**controller** など) およびこの新しいユーザーのパスワードを作成します。OK をクリックします。グループとユーザーの作成方法は [「KIE Server のグループおよびユーザーの設定」](#) を参照してください。
6. **Deployments** に移動して、既存のアプリケーションをすべて表示します。
7. **Install** をクリックします。
8. **rhpmam-7.13.5-add-ons.zip** ファイルをダウンロードして展開した一時ディレクトリーに移動し、**rhpmam-7.13.5-add-ons/rhpmam-7.13.5-controller-ee7.zip/controller.war** に移動します。
9. **controller.war** ファイルを選択し、**Next** をクリックして続行します。
10. ターゲットとするスタイルとして **Install this deployment as an application** を選択し、**Next** をクリックします。
11. アプリケーション名を **controller** のままにし、セキュリティーモデルを **DD Only** に設定します。残りのオプションはデフォルトのままにし、**Next** をクリックして続行します。

12. **Additional Configuration** セクションで **No, I will review the configuration later** を選択し、**Finish** をクリックします。

## 58.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティーの設定

ヘッドレス Process Automation Manager コントローラーをインストールしたら、本セクションで紹介するシステムプロパティーをアプリケーションサーバーに設定して、ヘッドレス Process Automation Manager コントローラーとの適切な接続を有効にします。



### 注記

実稼働環境で最適な結果を得るには、KIE Server とヘッドレス Process Automation Manager コントローラーを別のサーバーにインストールします。開発環境の場合は、KIE Server とヘッドレス Process Automation Manager コントローラーを同じサーバーにインストールします。いずれの場合も、ヘッドレス Process Automation Manager コントローラーがインストールされているすべてのアプリケーションサーバーでこのプロパティーを変更します。

### 前提条件

- KIE Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。

### 手順

1. ヘッドレス Process Automation Manager コントローラーがインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティー値を指定します。

表58.1 ヘッドレス Process Automation Manager コントローラーに必要なプロパティー

| 名前                         | 要件                                               |
|----------------------------|--------------------------------------------------|
| <b>org.kie.server.user</b> | <b>kie-server</b> ロールが割り当てられているユーザー              |
| <b>org.kie.server.pwd</b>  | <b>org.kie.server.user</b> プロパティーに指定したユーザーのパスワード |

2. KIE Server がインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティー値を指定します。

表58.2 ヘッドレス Process Automation Manager コントローラーがインストールされている場合に KIE Server に必要なプロパティー

| 名前                                    | 要件                                                          |
|---------------------------------------|-------------------------------------------------------------|
| <b>org.kie.server.controller.user</b> | <b>kie-server</b> ロールが割り当てられているユーザー                         |
| <b>org.kie.server.controller.pwd</b>  | <b>org.kie.server.controller.user</b> プロパティーに指定したユーザーのパスワード |

| 名前                               | 要件                                                                                                                   |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>org.kie.server.id</b>         | KIE Server インストールの ID または名前 ( <b>rhdm700-decision-server-1</b> など)                                                   |
| <b>org.kie.server.location</b>   | KIE Server の URL ( <b>http://&lt;HOST&gt;:&lt;PORT&gt;/kie-server/services/rest/server</b> )                         |
| <b>org.kie.server.controller</b> | ヘッドレス Process Automation Manager コントローラーの URL ( <b>http://&lt;HOST&gt;:&lt;PORT&gt;/controller/rest/controller</b> ) |

**<HOST>** は、KIE Server ホストの ID または名前に置き換えます (例: **localhost** または **192.7.8.9**)。

**<PORT>** は、KIE Server ホストのポート番号に置き換えます (例: **7001**)。

## 58.2. インストールの検証

ヘッドレス Process Automation Manager コントローラーをインストールし、必要なシステムプロパティとロール要件をアプリケーションサーバーに定義したら、ヘッドレス Process Automation Manager コントローラーが正しく動作していることを確認します。

### 前提条件

- KIE Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。
- アプリケーションサーバーで、ヘッドレス Process Automation Manager コントローラーに必要なシステムプロパティおよびロール要件をすべて設定している。

### 手順

端末で以下のコマンドを実行して、ヘッドレス Process Automation Manager コントローラーが動作していることを確認します。

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```

**<HOST>** は、ヘッドレスの Process Automation Manager コントローラーホストの ID または名前で、たとえば、**localhost** または **192.7.8.9** です。

**<PORT>** は、ヘッドレスの Process Automation Manager コントローラーホストのポート番号で、たとえば、**7001** です。

**<CONTROLLER>** および **<CONTROLLER\_PWD>** は、本セクションで作成したユーザーの認証情報です。

このコマンドにより、KIE Server インスタンスに関する情報が返されます。



## 注記

あるいは、KIE Server Java API Client を使用して、ヘッドレス Process Automation Manager コントローラーにアクセスすることもできます。

ヘッドレス Process Automation Manager コントローラーが実行していない場合は、アプリケーションサーバーインスタンスを停止して再起動し、ヘッドレス Process Automation Manager コントローラーの URL または API へ再度アクセスしてみてください。

## 第59章 ORACLE WEBLOGIC SERVER での埋め込みプロセスエンジンまたはデシジョンエンジンの設定

埋め込みエンジンとは、デシジョンとビジネスプロセスを実行できるようにする軽量のワークフローおよびルールエンジンです。埋め込みエンジンは、Red Hat Process Automation Manager アプリケーションに含めるか、OpenShift、Kubernetes および Docker 経由でサービスとしてデプロイできます。また、API を使用して Red Hat Process Automation Manager にエンジンを埋め込むか、contexts and dependency injection (CDI) サービスセットの一部として、埋め込むことができます。

Red Hat Process Automation Manager アプリケーションで埋め込みされたエンジンを使用する予定の場合には、Red Hat Business Automation 部品表 (BOM) ファイルをプロジェクトの **pom.xml** ファイルに追加して、プロジェクトに、Maven の依存関係を追加する必要があります。Red Hat Business Automation BOM は、Red Hat Decision Manager と Red Hat Process Automation Manager の両方に適用されます。Red Hat Business Automation BOM (Bill of Materials) に関する詳細情報は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) を参照してください。

### 手順

1. Red Hat Business Automation BOM を **pom.xml** ファイルで宣言します。

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.13.5.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. **<dependencies>** タグでお使いのプロジェクトに必要な依存関係を宣言します。製品の BOM をプロジェクトにインポートしたら、ユーザー向け製品依存関係のバージョンが定義されるため、**<dependency>** 要素のサブ要素 **<version>** を指定する必要はありません。ただし、**<dependency>** 要素を使用して、プロジェクトで使用する依存関係を宣言する必要があります。
  - 標準的な Red Hat Process Automation Manager プロジェクトでは、使用する機能に応じて、以下の依存関係を宣言します。

### 埋め込みプロセスエンジンの依存関係

```
<!-- Public KIE API -->
<dependency>
<groupId>org.kie</groupId>
<artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
```

```

    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-flow</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-flow-builder</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-bpmn2</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-runtime-manager</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-persistence-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-query-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-audit</artifactId>
  </dependency>

  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-kie-services</artifactId>
  </dependency>

  <!-- Dependency needed for default WorkItemHandler implementations. -->
  <dependency>
    <groupId>org.jbpm</groupId>
    <artifactId>jbpm-workitems-core</artifactId>
  </dependency>

  <!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
  <dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>${logback.version}</version>
  </dependency>

```

- CDI を使用する Red Hat Process Automation Manager プロジェクトでは、通常、以下の依存関係を宣言します。

### CDI が有効化されたプロセスエンジンの依存関係



```

<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>

```

### 埋め込みデジジョンエンジン依存関係

```

<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>

```

- KIE Server を使用するには、以下の依存関係を宣言します。

### クライアントアプリケーション KIE Server の依存関係

```

<dependency>
  <groupId>org.kie.server</groupId>

```

```
<artifactId>kie-server-client</artifactId>
</dependency>
```

- Red Hat Process Automation Manager にリモートクライアントを作成するには、以下の依存関係を宣言します。

### クライアントの依存関係

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

- ルール、プロセス定義など、アセットを含む JAR ファイルを作成する場合は、お使いの Maven プロジェクトのパッケージングの種類を **kjar** と指定し、**org.kie:kie-maven-plugin** を使用して、**<project>** 要素に置かれた **kjar** パッケージングタイプを処理します。以下の例の **\${kie.version}** は、[What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#) に記載されている Maven ライブラリーのバージョンです。

```
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>
```

- プロジェクトで永続サポートのあるプロセスエンジンまたはデシジョンエンジンを使用する場合には、Red Hat Business Automation BOM ファイルから **version.org.hibernate-4ee7** プロパティをコピーして、**pom.xml** ファイルの **dependencyManagement** セクションに以下の Hibernate の依存関係を宣言する必要があります。

### Hibernate の依存関係

```
<!-- hibernate dependencies -->
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

■

## 第60章 キーストアを使用したパスワードセキュリティの確保

キーストアを使用して、Business Central と KIE Server の間の通信に使用するパスワードを暗号化できます。コントローラーと KIE Server のパスワードを暗号化する必要があります。Business Central と KIE Server を別のアプリケーションサーバーにデプロイする場合は、いずれのアプリケーションサーバーもキーストアを使用する必要があります。

Java Cryptography Extension KeyStore (JCEKS) は、対称鍵をサポートするため、キーストアには JCEKS を使用してください。JDK インストールに含まれる KeyTool を使用して、新しい JCEKS を作成します。



### 注記

KIE Server が JCEKS で設定されていない場合、KIE Server のパスワードはシステムプロパティにプレーンテキスト形式で保存されます。

### 前提条件

- KIE Server が Oracle WebLogic Server にインストールされている。
- 「[KIE Server のグループおよびユーザーの設定](#)」の記載通りに、**kie-server** ロールを割り当てた KIE Server ユーザーが作成されている。
- Java 8 以降がインストールされている。

### 手順

1. KeyTool を使用して JCEKS を作成するには、Java 8 のホームディレクトリーで以下のコマンドを実行します。

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass
<ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

上記の例では、以下の変数を置き換えてください。

- **<KEYSTORE\_PATH>**: キーストアの保存先のパス
  - **<KEYSTORE\_PASSWORD>**: キーストアのパスワード
  - **<ALIAS\_KEY\_PASSWORD>**: エイリアスで保存した値にアクセスする時に使用するパスワード
  - **<PASSWORD\_ALIAS>**: プロセスへのエントリーに使用するエイリアス
2. プロンプトが表示されたら、作成した KIE Server ユーザーのパスワードを入力します。
  3. 以下の表に記載のシステムプロパティを設定します。

表60.1 KIE Server JCEKS を読み込む時に使用するシステムプロパティ

| システムプロパティ | プレースホルダー | 説明 |
|-----------|----------|----|
|-----------|----------|----|

| システムプロパティ                            | プレースホルダー                              | 説明                                                                               |
|--------------------------------------|---------------------------------------|----------------------------------------------------------------------------------|
| <b>kie.keystore.keyStoreURL</b>      | <b>&lt;KEYSTORE_URL&gt;</b>           | 使用する JCEKS の URL (例:<br><b>file:///home/kie/keystores/key<br/>store.jceks</b> )  |
| <b>kie.keystore.keyStorePwd</b>      | <b>&lt;KEYSTORE_PWD&gt;</b>           | JCEKS のパスワード                                                                     |
| <b>kie.keystore.key.server.alias</b> | <b>&lt;KEY_SERVER_ALI<br/>AS&gt;</b>  | パスワードの保存先となる REST<br>サービスのキーのエイリアス                                               |
| <b>kie.keystore.key.server.pwd</b>   | <b>&lt;KEY_SERVER_PWD<br/>&gt;</b>    | 保存したパスワードを使用する<br>REST サービスのエイリアスのパ<br>スワード                                      |
| <b>kie.keystore.key.ctrl.alias</b>   | <b>&lt;KEY_CONTROL_AL<br/>IAS&gt;</b> | パスワードの保存先のデフォルト<br>の REST Process Automation<br>Controller のキーのエイリアス             |
| <b>kie.keystore.key.ctrl.pwd</b>     | <b>&lt;KEY_CONTROL_P<br/>WD&gt;</b>   | 保存したパスワードを使用する、<br>デフォルトの REST Process<br>Automation Controller のエイリア<br>スのパスワード |

4. KIE Server を起動して、設定を検証します。

## 第61章 次のステップ

- [デシジョンサービスのスタートガイド](#)
- [ガイド付きデシジョンテーブルを使用したデシジョンサービスの作成](#)

## パート VII. RED HAT PROCESS AUTOMATION MANAGER の修正およびアップグレード

Red Hat Process Automation Manager を使用する場合は、Red Hat カスタマーポータルで公開されている Red Hat Process Automation Manager への更新を、お使いの Red Hat Process Automation Manager ディストリビューションに適用して、機能強化および修正を最新に保つことができます。お客様がより簡単に Red Hat Process Automation Manager および KIE Server に更新を適用できるように、Red Hat では、新しいパッチリリースに対応する更新ツールおよび製品通知をご用意しています。

### 前提条件

- Red Hat カスタマーポータルのアカウントがある。
- Red Hat Process Automation Manager をインストールしている。インストールオプションは、[Red Hat Process Automation Manager インストールの計画](#)を参照してください。



### 注記

Red Hat Process Automation Manager 7.10 を使用していて、Red Hat Process Automation Manager 7.13 にアップグレードする場合は、[Red Hat Process Automation Manager 7.10の修正およびアップグレード](#)を参照してください。

## 第62章 RED HAT PROCESS AUTOMATION MANAGER のバージョン管理

Red Hat Process Automation Manager のバージョンには、メジャーリリース、マイナーリリース、およびパッチリリースのそれぞれの数字を組み合わせた **Major.Minor.Patch** という書式が使用されています (例: **7.5.1**)。この例では、メジャーリリースが **7.x.x**、マイナーリリースが **7.5.x**、パッチリリースが **7.5.1** となります。メジャーリリースでは、大抵の場合はデータ移行が求められますが、マイナーリリースの更新やパッチの更新は、通常 Red Hat Process Automation Manager リリースアーティファクトで提供される更新ツールで管理されます。

以下は、Red Hat Process Automation Manager の一般的なリリースの種類になります。

### メジャーリリースの移行

Red Hat Process Automation Manager のメジャーリリースには、非常に多くの機能強化、セキュリティ更新、バグ修正が含まれており、再設計された機能も含まれている可能性があります。Red Hat JBoss BPM Suite 6.4.x から Red Hat Process Automation Manager 7.0 など、メジャーリリースから別のメジャーリリースにアプリケーションを移行すると、通常データ移行が必要です。移行を容易にするために、Red Hat Process Automation Manager の新しいメジャーバージョンで、自動化した移行ツールがしばしば提供されますが、特定のデータおよび設定の手動処理が必要となる可能性が高くなります。サポートされる移行パスは、製品のお知らせおよびドキュメントに指定されています。

たとえば、移行の方法については、[Red Hat JBoss BPM Suite 6.4 から Red Hat Process Automation Manager 7.0 への移行](#) を参照してください。

### マイナーリリースのアップグレード

Red Hat Process Automation Manager のマイナーリリースには、機能拡張、セキュリティ更新、バグ修正が含まれます。Red Hat Process Automation Manager 7.5.x から 7.6 など、アプリケーションを別のマイナーリリースに移行する場合には、データの移行が必要な場合があります。大抵の場合は、Business Central、KIE Server、ヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の特定コンポーネントの更新を容易にする自動更新ツールが Red Hat Process Automation Manager のパッチ更新と新規マイナーバージョンで提供されます。デシジョンエンジンやスタンドアロンの Business Central など、その他の Red Hat Process Automation Manager アーティファクトは、各マイナーリリースが含まれる新しいアーティファクトとしてリリースされるため、再インストールして更新を適用する必要があります。

新たなマイナーリリースにアップグレードする前に、お使いの Red Hat Process Automation Manager に最新のパッチ更新を適用し、マイナーリリースのアップグレードが成功するようにしてください。

### パッチの更新

Red Hat Process Automation Manager のパッチの更新には、最新のセキュリティ更新、およびバグ修正が含まれます。予定されているパッチ更新には、製品のマイナーバージョンに対してすでにリリースされているパッチ更新がすべて含まれるため、更新を最新の状態にするために、パッチの更新を1つ1つ適用していく必要はありません。たとえば、Red Hat Process Automation Manager 7.5.0 または 7.5.1 は Red Hat Process Automation Manager 7.5.2 に更新できます。Red Hat Process Automation Manager で最適なパフォーマンスを得るためには、製品の更新が利用可能になるとすぐに適用する必要があります。また、Red Hat では既存製品の通常の更新サイクルとは別に、予定されていないパッチ更新がリリースされる場合があります。これには、特定の問題を修正するために Red Hat グローバルサポートサービス (GSS) によって提供されるセキュリティまたはその他の更新が含まれる場合があります、累積的な更新ではない場合があります。

大抵の場合は、Business Central、KIE Server、ヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の特定コンポーネントの更新を容易にする自動更新ツールが Red Hat Process Automation Manager のパッチ更新と新規マイナーバージョンで提



供されます。デシジョンエンジンやスタンドアロンの Business Central など、その他の Red Hat Process Automation Manager アーティファクトは、各マイナーリリースが含まれる新しいアーティファクトとしてリリースされるため、再インストールして更新を適用する必要があります。

リリース間での移行を最適に保ち、Red Hat Process Automation Manager ディストリビューションに最新の機能強化および修正が含まれるようにするには、製品リリースや更新が Red Hat カスタマーポータルで利用可能になった時点で、Red Hat Process Automation Manager に適用します。Red Hat カスタマーポータルで製品の通知を有効にすることも検討してください。

## 第63章 RED HAT カスタマーポータルでの通知の有効化

Red Hat カスタマーポータルで通知を有効にして、製品の更新とお知らせを受け取ることができます。これらの通知は、お使いのインストールに関連するドキュメントの更新や追加、製品のリリース、パッチの更新についてお知らせします。通知を有効にすると、Red Hat カスタマーポータルで利用可能になった製品の更新をより迅速に適用し、お使いのディストリビューションを最新の拡張機能と修正で最新の状態に保つことができます。

### 前提条件

- Red Hat カスタマーポータルのアカウントがあり、ログインしている。

### 手順

1. Red Hat カスタマーポータルウィンドウの右上隅で、プロフィール名をクリックして **通知** をクリックします。
2. **通知** タブを選択し、**通知の管理** をクリックします。
3. **フォロー** の横にあるドロップダウンメニューから **製品** を選択し、表示されたドロップダウンメニューから **Red Hat Process Automation Manager** を選択します。
4. **通知設定を保存** をクリックします。

必要に応じて、その他の製品の通知も同じように追加できます。

## 第64章 RED HAT PROCESS AUTOMATION MANAGER へのパッチ更新およびマイナーリリースアップグレードの適用

大抵の場合は、Business Central、KIE Server、ヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の特定コンポーネントの更新を容易にする自動更新ツールが Red Hat Process Automation Manager のパッチ更新と新規マイナーバージョンで提供されます。デシジョンエンジンやスタンドアロンの Business Central など、その他の Red Hat Process Automation Manager アーティファクトは、各マイナーリリースが含まれる新しいアーティファクトとしてリリースされるため、再インストールして更新を適用する必要があります。

同じ自動更新ツールを使用して、Red Hat Process Automation Manager 7.13 にパッチ更新とマイナーリリースアップグレードの両方を適用することができます。バージョン 7.13 から 7.13.5 への更新といった Red Hat Process Automation Manager のパッチ更新には、最新のセキュリティー更新とバグ修正が含まれています。バージョン 7.12.x から 7.13 へのアップグレードといった Red Hat Process Automation Manager のマイナーリリースアップグレードには、機能強化、セキュリティー更新、バグ修正が含まれます。



### 注記

Red Hat Process Automation Manager への更新だけが、Red Hat Process Automation Manager パッチ更新に含まれます。Red Hat JBoss EAP への更新は、Red Hat JBoss EAP パッチ配信を使用して適用する必要があります。詳細は、[Red Hat JBoss EAP パッチおよびアップグレードガイド](#) を参照してください。

### 前提条件

- Red Hat Process Automation Manager インスタンスおよび KIE Server インスタンスを実行していない。Red Hat Process Automation Manager または KIE Server のインスタンスを実行している間は更新を適用しないでください。

### 手順

- Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。  
バージョン 7.12.x から 7.13 などのように、Red Hat Process Automation Manager の新たなマイナーリリースにアップグレードする場合は、お使いの Red Hat Process Automation Manager に最新のパッチ更新を適用してから、以下の手順にしたがって新たなマイナーリリースにアップグレードしてください。
- Patches** をクリックし、**Red Hat Process Automation Manager [VERSION] Patch Update** をダウンロードし、ダウンロードした **rhpm-\$VERSION-update.zip** ファイルを一時ディレクトリーにデプロイメントします。  
この更新ツールは、Business Central、KIE Server、およびヘッドレス Process Automation Manager コントローラーなど、Red Hat Process Automation Manager の一定のコンポーネントの更新を自動化します。この更新ツールを使用して最初に更新を適用し、Red Hat Process Automation Manager ディストリビューションに関連するその他の更新、または新しいリリースアーティファクトをインストールします。
- 更新ツールにファイルが更新されないようにするには、デプロイメントした **rhpm-\$VERSION-update** フォルダーに移動し、**blacklist.txt** ファイルを開き、更新しないファイルの相対パスを追加します。  
ファイルが **blacklist.txt** ファイルのリストに追加されていると、更新スクリプトは、そのファイルを新しいバージョンに置き換えずにそのまま残し、新しいバージョンのファイルに **.new** 接尾辞を付けて追加します。ブロックファイルが配布されなくなると、更新ツール

は、**.removed** 接尾辞の付いた、空のマーカーファイルを作成します。次に、これらの新しいファイルを手動で保持、マージ、または削除することを選択できます。

**blacklist.txt** ファイルで除外されるファイルの例:

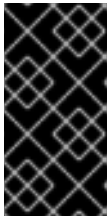
```
WEB-INF/web.xml // Custom file
styles/base.css // Obsolete custom file kept for record
```

更新後の、ブロックされたファイルディレクトリー内のコンテンツ:

```
$ ls WEB-INF
web.xml web.xml.new
```

```
$ ls styles
base.css base.css.removed
```

4. コマンドの端末で、**rhpam-\$VERSION-update.zip** ファイルからデプロイメントした一時ディレクトリーに移動し、以下の形式で **apply-updates** スクリプトを実行します。



### 重要

更新を行う前に、Red Hat Process Automation Manager インスタンスおよび KIE Server インスタンスが実行していないことを確認します。Red Hat Process Automation Manager または KIE Server のインスタンスを実行している間は更新を適用しないでください。

Linux システムまたは Unix ベースのシステムの場合:

```
$ ./apply-updates.sh $DISTRO_PATH $DISTRO_TYPE
```

Windows の場合:

```
$ .\apply-updates.bat $DISTRO_PATH $DISTRO_TYPE
```

**\$DISTRO\_PATH** の部分は、関連するディストリビューションディレクトリーへのパスで、**\$DISTRO\_TYPE** の部分は、更新しているディストリビューションの種類となります。

Red Hat Process Automation Manager 更新ツールでは、以下のディストリビューションの種類がサポートされます。

- **rhpam-business-central-eap7-deployable**: Business Central (**business-central.war**) を更新します。
- **rhpam-kie-server-ee8**: KIE Server (**kie-server.war**) を更新します。
- **rhpam-kie-server-jws**: Red Hat JBoss Web Server で KIE Server を更新します (**kie-server.war**)。
- **rhpam-controller-ee7**: ヘッドレス Process Automation Manager controller (**controller.war**) を更新します。
- **rhpam-controller-jws**: Red Hat JBoss Web Server でヘッドレスの Process Automation Manager コントローラーを更新します (**controller.war**)。

Red Hat JBoss EAP で、Red Hat Process Automation Manager の完全ディストリビューションに対する Business Central および KIE Server への更新の例:

```
$ ./apply-updates.sh ~EAP_HOME/standalone/deployments/business-central.war
rhpam-business-central-eap7-deployable
```

```
$ ./apply-updates.sh ~EAP_HOME/standalone/deployments/kie-server.war rhpam-kie-
server-ee8
```

ヘッドレス Process Automation Manager コントローラーへの更新例 (使用している場合):

```
$ ./apply-updates.sh ~EAP_HOME/standalone/deployments/controller.war rhpam-
controller-ee7
```

この更新スクリプトは、デプロイメントした **rhpam-\$VERSION-update** ディレクトリーに、指定したディストリビューションのコピーを含む **backup** ディレクトリーを作成してから、更新を行います。

5. 更新ツールが完了したら、更新ツールをダウンロードした、Red Hat カスタマーポータル **Software Downloads** ページに戻り、Red Hat Process Automation Manager ディストリビューションに関するその他の更新または新しいリリースアーティファクトをインストールします。デシジョンエンジンまたはその他のアドオンに関する **.jar** ファイルなど、Red Hat Process Automation Manager ディストリビューションにすでに存在しているファイルについては、ファイルの既存のバージョンを Red Hat カスタマーポータルから取得した新しいバージョンに置き換えます。
6. エアギャップ環境など、スタンドアロンの Red Hat Process Automation Manager 7.13.5 **Maven リポジトリー** アーティファクト (**rhpam-7.13.5-maven-repository.zip**) を使用する場合は、Red Hat Process Automation Manager 7.13.5 **Maven リポジトリー** をダウンロードして、ダウンロードした **rhpam-7.13.5-maven-repository.zip** ファイルを既存の **~/maven-repository** ディレクトリーに展開して、関連するコンテンツを更新します。

Maven リポジトリーの更新例:

```
$ unzip -o rhpam-7.13.5-maven-repository.zip 'rhba-7.13.5.GA-maven-repository/maven-
repository/*' -d /tmp/rhbaMavenRepoUpdate
```

```
$ mv /tmp/rhbaMavenRepoUpdate/rhba-7.13.5.GA-maven-repository/maven-repository/
$REPO_PATH/
```



### 注記

更新が完了したら **/tmp/rhbaMavenRepoUpdate** ディレクトリーを削除してください。

7. オプション: Red Hat Process Automation Manager をプロパティーベースのユーザーストレージの使用からファイルベースのユーザーストレージに変更する場合、以下の手順を実行します。
  - a. **\$JBOSS\_HOME** ディレクトリーに移動し、以下のコマンドのいずれかを実行します。

- Linux システムまたは Unix ベースのシステムの場合:

```
$ ./bin/standalone.sh --admin-only -c standalone-full.xml
```

```
$ ./bin/jboss-cli.sh --connect --file=rhpam-$VERSION-update/elytron/add-kie-fs-realm.cli
```

- Windows の場合:

```
$ ./bin/standalone.bat --admin-only -c standalone-full.xml
```

```
$ ./bin/jboss-cli.bat --connect --file=rhpam-$VERSION-update/elytron/add-kie-fs-realm.cli
```

b. 以下のコマンドを実行します。

- Linux システムまたは Unix ベースのシステムの場合:

```
$ ./bin/elytron-tool.sh filesystem-realm --users-file
standalone/configuration/application-users.properties --roles-file
standalone/configuration/application-roles.properties --output-location
standalone/configuration/kie-fs-realm-users --filesystem-realm-name kie-fs-realm-
users
```

- Windows の場合:

```
$ ./bin/elytron-tool.bat filesystem-realm --users-file
standalone/configuration/application-users.properties --roles-file
standalone/configuration/application-roles.properties --output-location
standalone/configuration/kie-fs-realm-users --filesystem-realm-name kie-fs-realm-
users
```

c. **rhpam-\$VERSION-update.zip** ファイルを展開したディレクトリーに移動し、以下のいずれかのコマンドを実行して **kie-fs-realm** パッチを適用します。

- Linux システムまたは Unix ベースのシステムの場合:

```
$ ./elytron/kie-fs-realm-patch.sh ~/$JBOSS_HOME/standalone/configuration/kie-fs-
realm-users/
```

- Windows の場合:

```
$ ./elytron/kie-fs-realm-patch.bat ~/$JBOSS_HOME/standalone/configuration/kie-fs-
realm-users/
```

8. 関連する更新をすべて適用したら、Red Hat Process Automation Manager および KIE Server を起動して、Business Central にログインします。

9. Business Central 内のすべてのプロジェクトデータが存在して正確であることを確認し、Business Central ウィンドウの右上隅でプロファイル名をクリックし、**About** をクリックして、更新した製品バージョン番号を確認します。

Business Central でエラーが発生したり、データが不足していることが通知されたら、**rhpam-\$VERSION-update** ディレクトリーの **backup** ディレクトリーにコンテンツを復元し、更新ツールへの変更を戻します。Red Hat カスタマーポータルで Red Hat Process Automation Manager の以前のバージョンから、関連するリリースアーティファクトを再インストールできます。以前のディストリビューションを復元したら、更新を再実行してください。

---

## 第65章 関連情報

- [Red Hat Process Automation Manager インストールの計画](#)
- [Red Hat Process Automation Manager 7.10 の修正およびアップグレード](#)
- [Red Hat Process Automation Manager 7.13 リリースノート](#)

## 付録C バージョン情報

本書の最終更新日: 2024 年 3 月 14 日 (木)



## 付録D お問い合わせ先

Red Hat Process Automation Manager のドキュメントチーム: [brms-docs@redhat.com](mailto:brms-docs@redhat.com)