



Red Hat Process Automation Manager 7.2

Deploying a Red Hat Process Automation
Manager trial environment on Red Hat
OpenShift Container Platform

Red Hat Process Automation Manager 7.2 Deploying a Red Hat Process Automation Manager trial environment on Red Hat OpenShift Container Platform

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Copyright © 2020 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to deploy a Red Hat Process Automation Manager 7.2 trial environment on Red Hat OpenShift Container Platform.

Table of Contents

PREFACE	3
CHAPTER 1. OVERVIEW OF RED HAT PROCESS AUTOMATION MANAGER ON RED HAT OPENSIFT CONTAINER PLATFORM	4
CHAPTER 2. ENSURING THE AVAILABILITY OF IMAGE STREAMS AND THE IMAGE REGISTRY	6
CHAPTER 3. DEPLOYING A TRIAL ENVIRONMENT	8
CHAPTER 4. OPENSIFT TEMPLATE REFERENCE INFORMATION	9
4.1. RHPAM72-TRIAL-EPHEMERAL.YAML TEMPLATE	9
4.1.1. Parameters	9
4.1.2. Objects	21
4.1.2.1. Services	21
4.1.2.2. Routes	21
4.1.2.3. Deployment Configurations	21
4.1.2.3.1. Triggers	21
4.1.2.3.2. Replicas	22
4.1.2.3.3. Pod Template	22
4.1.2.3.3.1. Service Accounts	22
4.1.2.3.3.2. Image	22
4.1.2.3.3.3. Readiness Probe	22
4.1.2.3.3.4. Liveness Probe	23
4.1.2.3.3.5. Exposed Ports	23
4.1.2.3.3.6. Image Environment Variables	23
4.1.2.4. External Dependencies	39
4.1.2.4.1. Secrets	39
4.2. OPENSIFT USAGE QUICK REFERENCE	39
APPENDIX A. VERSIONING INFORMATION	41

PREFACE

As a system engineer, you can deploy a Red Hat Process Automation Manager trial environment on Red Hat OpenShift Container Platform to evaluate or demonstrate development and use of rules and other business assets.

Prerequisites

- At least three gigabytes of memory must be available in the OpenShift cluster/namespace.
- The OpenShift project for the deployment must be created.
- You must be logged in to the project using the **oc** command. For more information about the **oc** command-line tool, see the OpenShift [CLI Reference](#). If you want to use the OpenShift Web console to deploy templates, you must also be logged on using the Web console.

CHAPTER 1. OVERVIEW OF RED HAT PROCESS AUTOMATION MANAGER ON RED HAT OPENSIFT CONTAINER PLATFORM

You can deploy Red Hat Process Automation Manager into a Red Hat OpenShift Container Platform environment.

In this solution, components of Red Hat Process Automation Manager are deployed as separate OpenShift pods. You can scale each of the pods up and down individually, providing as few or as many containers as necessary for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.

The following key components of Red Hat Process Automation Manager are available on OpenShift:

- Process Server, also known as *Execution Server* or *KIE Server*, is the infrastructure element that runs decision services, process applications, and other deployable assets (collectively referred to as *services*). All logic of the services runs on execution servers.

A database server is normally required for Process Server. You can provide a database server in another OpenShift pod or configure an execution server on OpenShift to use any other database server. Alternatively, Process Server can use an H2 database; in this case, the pod cannot be scaled.

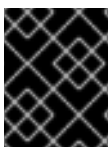
You can freely scale up a Process Server pod, providing as many copies as necessary, running on the same host or different hosts. As you scale a pod up or down, all its copies use the same database server and run the same services. OpenShift provides load balancing and a request can be handled by any of the pods.

You can deploy a separate Process Server pod to run a different group of services. That pod can also be scaled up or down. You can have as many separate replicated Process Server pods as necessary.

- Business Central is a web-based interactive environment for authoring services. It also provides a management and monitoring console. You can use Business Central to develop services and deploy them to Process Servers. You can also use Business Central to monitor the execution of processes.

Business Central is a centralized application. However, you can configure it for high availability, where multiple pods run and share the same data.

Business Central includes a Git repository that holds the source for the services that you develop on it. It also includes a built-in Maven repository. Depending on configuration, Business Central can place the compiled services (KJAR files) into the built-in Maven repository or (if configured) into an external Maven repository.



IMPORTANT

In the current version, high-availability Business Central functionality is a technology preview.

- Business Central Monitoring is a web-based management and monitoring console. It can manage deployment of services to Process Servers and provide monitoring information, but does not include authoring capabilities. You can use this component to manage staging and production environments.
- Smart Router is an optional layer between Process Servers and other components that interact with them. It is required if you want Business Central or Business Central Monitoring to interact with several different Process Servers. Also, when your environment includes many services

running on different Process Servers, Smart Router provides a single endpoint to all client applications. A client application can make a REST API call requiring any service. Smart Router automatically determines which Process Server must be called for any particular request.

You can arrange these and other components into various environment configurations within OpenShift.

The following environment types are typical:

- *Authoring*: An environment for creating and modifying services using Business Central. It consists of pods that provide Business Central for the authoring work and a Process Server for test execution of the services. For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform](#).
- *Managed deployment*: An environment for running existing services for staging and production purposes. This environment includes several groups of Process Server pods; you can deploy and undeploy services on every such group and also scale the group up or down as necessary. Use Business Central Monitoring to deploy, run, and stop the services and to monitor their execution. For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager managed server environment on Red Hat OpenShift Container Platform](#).
- *Deployment with immutable servers*: An alternate environment for running existing services for staging and production purposes. In this environment, when you deploy a Process Server pod, it builds an image that loads and starts a service or group of services. You cannot stop any service on the pod or add any new service to the pod. If you want to use another version of a service or modify the configuration in any other way, you deploy a new server image and displace the old one. In this system, the Process Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows and do not need to use any other tools to manage the pods. Optionally, you can use Business Central Monitoring to monitor the performance of the environment and to stop and restart some of the service instances, but not to deploy additional services to any Process Server or undeploy any existing ones (you can not add or remove containers). For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform](#).

You can also deploy a *trial* or evaluation environment. This environment includes Business Central and a Process Server. You can set it up quickly and use it to evaluate or demonstrate developing and running assets. However, the environment does not use any persistent storage, and any work you do in the environment is not saved. For instructions about deploying this environment, see [Deploying a Red Hat Process Automation Manager trial environment on Red Hat OpenShift Container Platform](#).

To deploy a Red Hat Process Automation Manager environment on OpenShift, you can use the templates that are provided with Red Hat Process Automation Manager. You can modify the templates to ensure that the configuration suits your environment.

CHAPTER 2. ENSURING THE AVAILABILITY OF IMAGE STREAMS AND THE IMAGE REGISTRY

To deploy Red Hat Process Automation Manager components of Red Hat OpenShift Container Platform, you must ensure that OpenShift can download the correct images from the Red Hat registry. To download the images, OpenShift requires the information about their location (known as *image streams*). OpenShift also must be configured to authenticate with the Red Hat registry using your service account user name and password.

Some versions of the OpenShift environment include the required image streams. You must check if they are available. If image streams are available in OpenShift by default, you can use them if the OpenShift infrastructure is configured for registry authentication server. The administrator must complete the registry authentication configuration when installing the OpenShift environment.

Otherwise, you can configure registry authentication in your own project and install the image streams in the same project.

Procedure

1. Determine whether Red Hat OpenShift Container Platform was configured with the user name and password for Red Hat registry access. For details about the required configuration, see [Configuring a Registry Location](#). If you are using an OpenShift Online subscription, it is configured for Red Hat registry access.
2. If Red Hat OpenShift Container Platform was configured with the user name and password for Red Hat registry access, run the following commands:

```
$ oc get imagestreamtag -n openshift | grep rhpam72-businesscentral
$ oc get imagestreamtag -n openshift | grep rhpam72-kieserver
```

If the outputs of both commands are not empty, the required image streams are available in the **openshift** namespace and no further action is required.

3. If the output of one or both of the commands is empty or if OpenShift was not configured with the user name and password for Red Hat registry access, complete the following steps:
 - a. Ensure you are logged in to OpenShift with the **oc** command and that your project is active.
 - b. Complete the steps documented in [Registry Service Accounts for Shared Environments](#). You must log on to Red Hat Customer Portal to access the document and to complete the steps to create a registry service account.
 - c. Select the **OpenShift Secret** tab and click the link under **Download secret** to download the YAML secret file.
 - d. View the downloaded file and note the name that is listed in the **name:** entry.
 - e. Run the following commands:

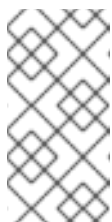
```
oc create -f <file_name>.yaml
oc secrets link default <secret_name> --for=pull
oc secrets link builder <secret_name> --for=pull
```

Where **<file_name>** is the name of the downloaded file and **<secret_name>** is the name that is listed in the **name:** entry of the file.

- f. Download the **rhpm-7.2.0-openshift-templates.zip** product deliverable file from the [Software Downloads](#) page and extract the **rhpm72-image-streams.yaml** file.
- g. Complete one of the following actions:
 - Run the following command:

```
$ oc create -f rhpm72-image-streams.yaml
```

- Using the OpenShift Web UI, select **Add to Project** → **Import YAML / JSON** and then choose the file or paste its contents.



NOTE

If you complete these steps, you install the image streams into the namespace of your project. If you install the image streams using these steps, you must set the **IMAGE_STREAM_NAMESPACE** parameter to the name of this project when deploying templates.

CHAPTER 3. DEPLOYING A TRIAL ENVIRONMENT

You can deploy a trial (evaluation) Red Hat Process Automation Manager environment. It consists of Business Central for authoring or managing services and Process Server for test execution of services.

This environment does not include permanent storage. Assets that you create or modify in a trial environment are not saved.

This environment is intended for test and demonstration access. It supports cross-origin resource sharing (CORS). This means that Process Server endpoints can be accessed using a browser when other resources on the page are provided by other servers. Process Server endpoints are normally intended for REST calls, but browser access can be needed in some demonstration configurations.

The procedure is minimal. There are no required settings and all passwords are set to a single value (the default password is **RedHat**).

To deploy a single authoring environment, use the **rhpam72-trial-ephemeral.yaml** template file. You can extract this file from the **rhpam-7.2.0-openshift-templates.zip** product deliverable file. You can download the file from the [Software Downloads](#) page.

Procedure

1. Use one of the following methods to deploy the template:

- In the OpenShift Web UI, select **Add to Project** → **Import YAML / JSON** and then select or paste the **rhpam72-trial-ephemeral.yaml** file. In the **Add Template** window, ensure **Process the template** is selected and click **Continue**.
- To use the OpenShift command line console, prepare the following command line:

```
oc new-app -f <template-path>/rhpam72-trial-ephemeral.yaml
```

In this command line, replace **<template-path>** with the path to the downloaded template file.

2. Optionally, set any parameters as described in the template. A typical trial deployment requires only the following parameter:

- **ImageStream Namespace (IMAGE_STREAM_NAMESPACE)**: The namespace where the image streams are available. If the image streams were already available in your OpenShift environment (see [Chapter 2, Ensuring the availability of image streams and the image registry](#)), the namespace is **openshift**. If you installed the image streams file, the namespace is the name of the OpenShift project.

3. Complete the creation of the environment, depending on the method that you are using:

- In the OpenShift Web UI, click **Create**.
 - A **This will create resources that may have security or project behavior implications** pop-up message might be displayed. If it is displayed, click **Create Anyway**.
- Complete and run the command line.

CHAPTER 4. OPENSIFT TEMPLATE REFERENCE INFORMATION

Red Hat Process Automation Manager provides the following OpenShift templates. To access the templates, download and extract the **rhcam-7.2.0-openshift-templates.zip** product deliverable file from the [Software Downloads](#) page of the Red Hat customer portal.

- **rhcam72-trial-ephemeral.yaml** provides a Business Central and a Process Server connected to the Business Central. This environment uses an ephemeral configuration without any persistent storage. For details about this template, see [Section 4.1, “rhcam72-trial-ephemeral.yaml template”](#).

4.1. RHPAM72-TRIAL-EPHEMERAL.YAML TEMPLATE

Application template for an ephemeral authoring and testing environment, for Red Hat Process Automation Manager 7.2

4.1.1. Parameters

Templates allow you to define parameters which take on a value. That value is then substituted wherever the parameter is referenced. References can be defined in any text field in the objects list field. Refer to the [Openshift documentation](#) for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
DEFAULT_PASSWORD	KIE_ADMIN_PASSWORD	Default password used for multiple components for user convenience in this trial environment	RedHat	True
KIE_ADMIN_USER	KIE_ADMIN_USER	KIE administrator username	adminUser	False
KIE_SERVER_USER	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	executionUser	False
KIE_SERVER_BYPASS_AUTH_USER	KIE_SERVER_BYPASS_AUTH_USER	KIE server bypass auth user (Sets the org.kie.server.bypass.auth.user system property)	false	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CONTROLLER_USER	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	controllerUser	False
KIE_MBEANS	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	true	False
KIE_SERVER_HOSTNAME_HTTP	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Origin response header value in the KIE Server (useful for CORS support)	*	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Methods response header value in the KIE Server (useful for CORS support)	GET, POST, OPTIONS, PUT	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Headers response header value in the KIE Server (useful for CORS support)	Accept, Authorization, Content-Type, X-Requested-With	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Credentials response header value in the KIE Server (useful for CORS support)	true	False
KIE_SERVER_ACCESS_CONTROL_MAX_AGE	AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Max-Age response header value in the KIE Server (useful for CORS support)	1	False
BUSINESS_CENTRAL_HOSTNAME_HTTP	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-rhpamcentr-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStreams for Red Hat Middleware images are installed. These ImageStreams are normally installed in the openshift namespace. You should only need to modify this if you installed the ImageStreams in a different namespace/project.	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	The name of the image stream to use for KIE server. Default is "rhpam72-kieserver-openshift".	rhpam72-kieserver-openshift	True
IMAGE_STREAM_TAG	–	A named pointer to an image in an image stream. Default is "1.1".	1.1	True
KIE_SERVER_CONTAINER_DEPLOYMENT	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration in format: containerId=groupId:artifactId:version c2=g2:a2:v2	–	False
MAVEN_REPO_ID	MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	my-repo-id	False

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_REPO_URL	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	–	False
BUSINESS_CENTRAL_MAVEN_USERNAME	KIE_MAVEN_USER	Username to access the Maven service hosted by Business Central inside EAP.	mavenUser	True
GIT_HOOKS_DIRECTORY	GIT_HOOKS_DIRECTORY	The directory to use for git hooks, if required.	/opt/eap/standalone/data/kie/git/hooks	False
BUSINESS_CENTRAL_MEMORY_LIMIT	–	Business Central Container memory limit	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit	1Gi	False
SSO_URL	SSO_URL	RH-SSO URL	https://rh-sso.example.com/auth	False
SSO_REALM	SSO_REALM	RH-SSO Realm name	–	False
BUSINESS_CENTRAL_SSO_CLIENT	SSO_CLIENT	Business Central RH-SSO Client name	–	False

Variable name	Image Environment Variable	Description	Example value	Required
BUSINESS_CENTRAL_SSO_SECRET	SSO_SECRET	Business Central RH-SSO Client Secret	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER_SSO_CLIENT	SSO_CLIENT	KIE Server RH-SSO Client name	–	False
KIE_SERVER_SSO_SECRET	SSO_SECRET	KIE Server RH-SSO Client Secret	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn't exist	–	False
SSO_PASSWORD	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	false	False
SSO_PRINCIPAL_ATTRIBUTE	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	preferred_username	False
AUTH_LDAP_URL	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication	ldap://myldap.example.com	False
AUTH_LDAP_BIND_DN	AUTH_LDAP_BIND_DN	Bind DN used for authentication	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication	Password	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_JAAS_SECURITY_DOMAIN	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	–	False
AUTH_LDAP_BASE_CTX_DN	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	SUBTREE_SCOPE	False
AUTH_LDAP_SEARCH_TIME_LIMIT	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	10000	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	distinguishedName	False
AUTH_LDAP_PARSE_USERNAME	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with <code>usernameBeginString</code> and <code>usernameEndString</code> .	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_USERNAME_END_STRING	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	AUTH_LDAP_ROLE_ATTRIBUTE_ID	Name of the attribute containing the user roles.	<code>memberOf</code>	False
AUTH_LDAP_ROLE_CTX_DN	AUTH_LDAP_ROLE_CTX_DN	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	<code>ou=groups,ou=example,ou=com</code>	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_FILTER	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	(memberOf={1})	False
AUTH_LDAP_ROLE_RECURSION	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	1	False
AUTH_LDAP_DEFAULT_ROLE	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users	guest	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributesDN property is set to true, this property is used to find the role object's name attribute.	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttribute ID. If set to true, the DN is checked for the roleNameAttribute ID. If set to false, the DN is not checked for the roleNameAttribute ID. This flag can improve the performance of LDAP queries.	false	False
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeId attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	false	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_REFERRAL_USE_R_ATTRIBUTE_ID_TO_CHECK	AUTH_LDAP_REFERRAL_USE_R_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	–	False
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	–	False

4.1.2. Objects

The CLI supports various object types. A list of these object types as well as their abbreviations can be found in the [Openshift documentation](#).

4.1.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. Refer to the [container-engine documentation](#) for more information.

Service	Port	Name	Description
\${APPLICATION_NAME}-rhpamcentr	8080	http	All the Business Central web server's ports.
	8001	git-ssh	
\${APPLICATION_NAME}-kieserver	8080	–	All the KIE server web server's ports.

4.1.2.2. Routes

A route is a way to expose a service by giving it an externally-reachable hostname such as **www.example.com**. A defined route and the endpoints identified by its service can be consumed by a router to provide named connectivity from external clients to your applications. Each route consists of a route name, service selector, and (optionally) security configuration. Refer to the [Openshift documentation](#) for more information.

Service	Security	Hostname
\${APPLICATION_NAME}-rhpamcentr-http	none	\${BUSINESS_CENTRAL_HOSTNAME_HTTP}
\${APPLICATION_NAME}-kieserver-http	none	\${KIE_SERVER_HOSTNAME_HTTP}

4.1.2.3. Deployment Configurations

A deployment in OpenShift is a replication controller based on a user defined template called a deployment configuration. Deployments are created manually or in response to triggered events. Refer to the [Openshift documentation](#) for more information.

4.1.2.3.1. Triggers

A trigger drives the creation of new deployments in response to events, both inside and outside OpenShift. Refer to the [Openshift documentation](#) for more information.

Deployment	Triggers
\${APPLICATION_NAME}-rhpamcentr	ImageChange

Deployment	Triggers
<code>\${APPLICATION_NAME}-kieserver</code>	ImageChange

4.1.2.3.2. Replicas

A replication controller ensures that a specified number of pod "replicas" are running at any one time. If there are too many, the replication controller kills some pods. If there are too few, it starts more. Refer to the [container-engine documentation](#) for more information.

Deployment	Replicas
<code>\${APPLICATION_NAME}-rhpmcentr</code>	1
<code>\${APPLICATION_NAME}-kieserver</code>	1

4.1.2.3.3. Pod Template

4.1.2.3.3.1. Service Accounts

Service accounts are API objects that exist within each project. They can be created or deleted like any other API object. Refer to the [Openshift documentation](#) for more information.

Deployment	Service Account
<code>\${APPLICATION_NAME}-rhpmcentr</code>	<code>\${APPLICATION_NAME}-rhpmsvc</code>
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${APPLICATION_NAME}-rhpmsvc</code>

4.1.2.3.3.2. Image

Deployment	Image
<code>\${APPLICATION_NAME}-rhpmcentr</code>	rhpm72-businesscentral-openshift
<code>\${APPLICATION_NAME}-kieserver</code>	<code>\${KIE_SERVER_IMAGE_STREAM_NAME}</code>

4.1.2.3.3.3. Readiness Probe

`${APPLICATION_NAME}-rhpmcentr`

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${DEFAULT_PASSWORD}'
http://localhost:8080/kie-wb.jsp
```

`${APPLICATION_NAME}-kieserver`

-

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

4.1.2.3.3.4. Liveness Probe

\${APPLICATION_NAME}-rhpamcentr

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${DEFAULT_PASSWORD}'
http://localhost:8080/kie-wb.jsp
```

\${APPLICATION_NAME}-kieserver

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

4.1.2.3.3.5. Exposed Ports

Deployments	Name	Port	Protocol
\${APPLICATION_NAME}-rhpamcentr	jolokia	8778	TCP
	http	8080	TCP
	git-ssh	8001	TCP
\${APPLICATION_NAME}-kieserver	jolokia	8778	TCP
	http	8080	TCP

4.1.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
\${APPLICATION_NAME}-rhpamcentr	WORKBENCH_ROUTE_NAME	–	\${APPLICATION_NAME}-rhpamcentr
	KIE_ADMIN_USER	KIE administrator username	\${KIE_ADMIN_USER}
	KIE_ADMIN_PWD	Default password used for multiple components for user convenience in this trial environment	\${DEFAULT_PASSWORD}

Deployment	Variable name	Description	Example value
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	\${KIE_MBEANS}
	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	\${KIE_SERVER_CONTROLLER_USER}
	KIE_SERVER_CONTROLLER_PWD	Default password used for multiple components for user convenience in this trial environment	\${DEFAULT_PASSWORD}
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	\${KIE_SERVER_USER}
	KIE_SERVER_PWD	Default password used for multiple components for user convenience in this trial environment	\${DEFAULT_PASSWORD}
	MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	\${MAVEN_REPO_ID}
	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	\${MAVEN_REPO_URL}
	MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	\${MAVEN_REPO_USERNAME}
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	\${MAVEN_REPO_PASSWORD}
	KIE_MAVEN_USER	Username to access the Maven service hosted by Business Central inside EAP.	\${BUSINESS_CENTRAL_MAVEN_USERNAME}

Deployment	Variable name	Description	Example value
	KIE_MAVEN_PWD	Default password used for multiple components for user convenience in this trial environment	`\${DEFAULT_PASSWORD}`
	GIT_HOOKS_DIR	The directory to use for git hooks, if required.	`\${GIT_HOOKS_DIR}`
	SSO_URL	RH-SSO URL	`\${SSO_URL}`
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name	`\${SSO_REALM}`
	SSO_SECRET	Business Central RH-SSO Client Secret	`\${BUSINESS_CENTRAL_SSO_SECRET}`
	SSO_CLIENT	Business Central RH-SSO Client name	`\${BUSINESS_CENTRAL_SSO_CLIENT}`
	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn't exist	`\${SSO_USERNAME}`
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	`\${SSO_PASSWORD}`
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	`\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}`
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	`\${SSO_PRINCIPAL_ATTRIBUTE}`
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-rhpamcentr-<project>.<default-domain-suffix>	`\${BUSINESS_CENTRAL_HOSTNAME_HTTP}`

Deployment	Variable name	Description	Example value
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication	\${AUTH_LDAP_URL}
	AUTH_LDAP_BIND_DN	Bind DN used for authentication	\${AUTH_LDAP_BIND_DN}
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication	\${AUTH_LDAP_BIND_CREDENTIAL}
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	\${AUTH_LDAP_JAAS_SECURITY_DOMAIN}
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	\${AUTH_LDAP_BASE_CTX_DN}
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	\${AUTH_LDAP_BASE_FILTER}
	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	\${AUTH_LDAP_SEARCH_SCOPE}
	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	\${AUTH_LDAP_SEARCH_TIME_LIMIT}

Deployment	Variable name	Description	Example value
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	`\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}`
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with <code>usernameBeginString</code> and <code>usernameEndString</code> .	`\${AUTH_LDAP_PARSE_USERNAME}`
	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.	`\${AUTH_LDAP_USERNAME_BEGIN_STRING}`
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.	`\${AUTH_LDAP_USERNAME_END_STRING}`

Deployment	Variable name	Description	Example value
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	Name of the attribute containing the user roles.	`\${AUTH_LDAP_ROLE_ATTRIBUTE_ID}`
	AUTH_LDAP_ROLE_S_CTX_DN	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	`\${AUTH_LDAP_ROLE_S_CTX_DN}`
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a <code>{0}</code> expression is used. The authenticated userDN is substituted into the filter anywhere a <code>{1}</code> is used. An example search filter that matches on the input username is <code>(member={0})</code> . An alternative that matches on the authenticated userDN is <code>(member={1})</code> .	`\${AUTH_LDAP_ROLE_FILTER}`
	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	`\${AUTH_LDAP_ROLE_RECURSION}`
	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users	`\${AUTH_LDAP_DEFAULT_ROLE}`

Deployment	Variable name	Description	Example value
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributesDN property is set to true, this property is used to find the role object's name attribute.	\${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	\${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeID attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	\${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}

Deployment	Variable name	Description	Example value
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	\${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	\${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	\${AUTH_ROLE_MAPPER_REPLACE_ROLE}
\${APPLICATION_NAME}-kieserver	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	\${DROOLS_SERVER_FILTER_CLASSES}
	KIE_ADMIN_USER	KIE administrator username	\${KIE_ADMIN_USER}

Deployment	Variable name	Description	Example value
	KIE_ADMIN_PWD	Default password used for multiple components for user convenience in this trial environment	`\${DEFAULT_PASSWORD}`
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	`\${KIE_MBEANS}`
	KIE_SERVER_BYPASS_AUTH_USER	KIE server bypass auth user (Sets the org.kie.server.bypass.auth.user system property)	`\${KIE_SERVER_BYPASS_AUTH_USER}`
	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	`\${KIE_SERVER_CONTROLLER_USER}`
	KIE_SERVER_CONTROLLER_PWD	Default password used for multiple components for user convenience in this trial environment	`\${DEFAULT_PASSWORD}`
	KIE_SERVER_CONTROLLER_SERVICE	–	`\${APPLICATION_NAME}-rhpamcentr`
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_ROUTE_NAME	–	`\${APPLICATION_NAME}-kieserver`
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	`\${KIE_SERVER_USER}`
	KIE_SERVER_PWD	Default password used for multiple components for user convenience in this trial environment	`\${DEFAULT_PASSWORD}`

Deployment	Variable name	Description	Example value
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration in format: containerId=groupId:artifactId:version	c2=g2:a2:v2
	\${KIE_SERVER_CONTAINER_DEPLOYMENT}	MAVEN_REPOS	–
	RHPAMCENTR,EXTERNAL	RHPAMCENTR_MAVEN_REPO_SERVICE	–
	\${APPLICATION_NAME}-rhpamcentr	RHPAMCENTR_MAVEN_REPO_PATH	–
	/maven2/	RHPAMCENTR_MAVEN_REPO_USERNAME	Username to access the Maven service hosted by Business Central inside EAP.
	\${BUSINESS_CENTRAL_MAVEN_USERNAME}	RHPAMCENTR_MAVEN_REPO_PASSWORD	Default password used for multiple components for user convenience in this trial environment
	\${DEFAULT_PASSWORD}	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.
	\${MAVEN_REPO_ID}	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.
	\${MAVEN_REPO_URL}	EXTERNAL_MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.
	\${MAVEN_REPO_USERNAME}	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.
	\${MAVEN_REPO_PASSWORD}	SSO_URL	RH-SSO URL
	\${SSO_URL}	SSO_OPENIDCONNECT_DEPLOYMENTS	–

Deployment	Variable name	Description	Example value
	ROOT.war	SSO_REALM	RH-SSO Realm name
	\${SSO_REALM}	SSO_SECRET	KIE Server RH-SSO Client Secret
	\${KIE_SERVER_SSO_SECRET}	SSO_CLIENT	KIE Server RH-SSO Client name
	\${KIE_SERVER_SSO_CLIENT}	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn't exist
	\${SSO_USERNAME}	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client
	\${SSO_PASSWORD}	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation
	\${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.
	\${SSO_PRINCIPAL_ATTRIBUTE}	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>
	\${KIE_SERVER_HOSTNAME_HTTP}	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication
	\${AUTH_LDAP_URL}	AUTH_LDAP_BIND_DN	Bind DN used for authentication
	\${AUTH_LDAP_BIND_DN}	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication

Deployment	Variable name	Description	Example value
	<code>\${AUTH_LDAP_BIND_CREDENTIAL}</code>	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.
	<code>\${AUTH_LDAP_JAAS_SECURITY_DOMAIN}</code>	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.
	<code>\${AUTH_LDAP_BASE_CTX_DN}</code>	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a <code>{0}</code> expression is used. A common example for the search filter is <code>(uid={0})</code> .
	<code>\${AUTH_LDAP_BASE_FILTER}</code>	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.
	<code>\${AUTH_LDAP_SEARCH_SCOPE}</code>	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.
	<code>\${AUTH_LDAP_SEARCH_TIME_LIMIT}</code>	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.

Deployment	Variable name	Description	Example value
	<code>\${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}</code>	<code>AUTH_LDAP_PARSE_USERNAME</code>	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with <code>usernameBeginString</code> and <code>usernameEndString</code> .
	<code>\${AUTH_LDAP_PARSE_USERNAME}</code>	<code>AUTH_LDAP_USERNAME_BEGIN_STRING</code>	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.
	<code>\${AUTH_LDAP_USERNAME_BEGIN_STRING}</code>	<code>AUTH_LDAP_USERNAME_END_STRING</code>	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with <code>usernameEndString</code> and only taken into account if <code>parseUsername</code> is set to true.
	<code>\${AUTH_LDAP_USERNAME_END_STRING}</code>	<code>AUTH_LDAP_ROLE_ATTRIBUTE_ID</code>	Name of the attribute containing the user roles.
	<code>\${AUTH_LDAP_ROLE_ATTRIBUTE_ID}</code>	<code>AUTH_LDAP_ROLE_S_CTX_DN</code>	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.

Deployment	Variable name	Description	Example value
	<code>\${AUTH_LDAP_ROLES_CTX_DN}</code>	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a <code>{0}</code> expression is used. The authenticated userDN is substituted into the filter anywhere a <code>{1}</code> is used. An example search filter that matches on the input username is <code>(member={0})</code> . An alternative that matches on the authenticated userDN is <code>(member={1})</code> .
	<code>\${AUTH_LDAP_ROLE_FILTER}</code>	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.
	<code>\${AUTH_LDAP_ROLE_RECURSION}</code>	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users
	<code>\${AUTH_LDAP_DEFAULT_ROLE}</code>	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributesDN property is set to true, this property is used to find the role object's name attribute.

Deployment	Variable name	Description	Example value
	`\${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}`	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.
	`\${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}`	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeID attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.
	`\${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}`	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.

Deployment	Variable name	Description	Example value
	<code>#{AUTH_LDAP_REFERENCE_USER_ATTRIBUTE_ID_TO_CHECK}</code>	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is <code>original_role=role1,role2,role3</code>
	<code>#{AUTH_ROLE_MAPPER_ROLES_PROPERTIES}</code>	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.
	<code>#{AUTH_ROLE_MAPPER_REPLACE_ROLE}</code>	FILTERS	–
	<code>AC_ALLOW_ORIGIN,AC_ALLOW_METHODS,AC_ALLOW_HEADERS,AC_ALLOW_CREDENTIALS,AC_MAX_AGE</code>	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_NAME	–
	Access-Control-Allow-Origin	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Origin response header value in the KIE Server (useful for CORS support)
	<code>#{KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN}</code>	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_NAME	–
	Access-Control-Allow-Methods	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Methods response header value in the KIE Server (useful for CORS support)

Deployment	Variable name	Description	Example value
	<code>#{KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS}</code>	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_NAME	–
	Access-Control-Allow-Headers	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Headers response header value in the KIE Server (useful for CORS support)
	<code>#{KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS}</code>	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_NAME	–
	Access-Control-Allow-Credentials	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Credentials response header value in the KIE Server (useful for CORS support)
	<code>#{KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS}</code>	AC_MAX_AGE_FILTER_RESPONSE_HEADER_NAME	–
	Access-Control-Max-Age	AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Max-Age response header value in the KIE Server (useful for CORS support)

4.1.2.4. External Dependencies

4.1.2.4.1. Secrets

This template requires the following secrets to be installed for the application to run.

4.2. OPENSIFT USAGE QUICK REFERENCE

To deploy, monitor, manage, and undeploy Red Hat Process Automation Manager templates on Red Hat OpenShift Container Platform, you can use the OpenShift Web console or the **oc** command.

For instructions about using the Web console, see [Create and build an image using the Web console](#) .

For detailed instructions about using the **oc** command, see [CLI Reference](#). The following commands are likely to be required:

- To create a project, use the following command:

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

For more information, see [Creating a project using the CLI](#).

- To deploy a template (create an application from a template), use the following command:

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

For more information, see [Creating an application using the CLI](#).

- To view a list of the active pods in the project, use the following command:

```
$ oc get pods
```

- To view the current status of a pod, including information whether or not the pod deployment has completed and it is now in a running state, use the following command:

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

You can also use the **oc describe** command to view the current status of other objects. For more information, see [Application modification operations](#).

- To view the logs for a pod, use the following command:

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

- To view deployment logs, look up a **DeploymentConfig** name in the template reference and run the following command:

```
$ oc logs -f dc/<deployment-config-name>
```

For more information, see [Viewing deployment logs](#).

- To view build logs, look up a **BuildConfig** name in the template reference and run the command:

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

For more information, see [Accessing build logs](#).

- To scale a pod in the application, look up a **DeploymentConfig** name in the template reference and run the command:

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

For more information, see [Manual scaling](#).

- To undeploy the application, you can delete the project by using the command:

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

Alternatively, you can use the **oc delete** command to remove any part of the application, such as a pod or replication controller. For details, see [Application modification operations](#).

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Tuesday, May 28, 2019.