



Red Hat JBoss Middleware for OpenShift 3

OpenShift Primer

Get started with OpenShift

Red Hat JBoss Middleware for OpenShift 3 OpenShift Primer

Get started with OpenShift

Legal Notice

Copyright © 2018 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, 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 Software Collections 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

Step by step instructions for setting up OpenShift for running Red Hat JBoss Middleware for OpenShift images

Table of Contents

CHAPTER 1. OVERVIEW	3
1.1. ABOUT THIS GUIDE	3
1.2. WHY SHOULD I USE OPENSIFT?	3
CHAPTER 2. GET STARTED	4
CHAPTER 3. INSTALL OPENSIFT ENTERPRISE 3	5
3.1. ATTACH OPENSIFT SUBSCRIPTION	5
3.2. SET UP REPOSITORIES	6
3.3. INSTALL THE OPENSIFT ENTERPRISE PACKAGE	6
3.4. RUN THE INSTALLER	6
3.5. START OPENSIFT	7
CHAPTER 4. INTERACT WITH OPENSIFT ENTERPRISE 3	8
CHAPTER 5. UNDERSTAND ROLES AND AUTHENTICATION	9
5.1. CHANGE LOG IN IDENTITY PROVIDER	9
5.2. CREATE USER ACCOUNTS	9
CHAPTER 6. DEPLOY THE OPENSIFT ROUTER	11
CHAPTER 7. DEPLOY A DOCKER REGISTRY	12
CHAPTER 8. CREATE PERSISTENT STORAGE FOR THE REGISTRY	13
8.1. PROVISION THE PERSISTENT VOLUME	13
8.2. CREATE THE PERSISTENT VOLUME CLAIM	13
8.3. ADD THE CLAIM TO THE REGISTRY	14
CHAPTER 9. DEPLOY YOUR FIRST IMAGE	15

CHAPTER 1. OVERVIEW

OpenShift Enterprise 3 is a platform as a service (PaaS) offering from Red Hat that brings together Docker, Kubernetes, and an API to manage these services. OpenShift Enterprise 3 (OSE) allows you to create and manage containers. Containers are standalone processes that run within their own environment.

1.1. ABOUT THIS GUIDE

This guide introduces you to the basic concepts of OSE3, and helps you install and configure a basic application from an xPaaS image. This guide is not suitable for deploying or installing a production environment of OpenShift. For that, please refer to the official [OpenShift documentation](#).

1.2. WHY SHOULD I USE OPENSIFT?

Containers are standalone processes that run within their own environment, independent of operating system and the underlying infrastructure. OpenShift helps you to develop, deploy, and manage container-based applications. It provides you with a self-service platform to create, modify, and deploy applications on demand. Thus enabling faster development and release life cycles.

The most relatable example is that of a web server. The developer creates, for example, a JBoss Web Server image with a specific configuration and you download this image from the registry. You then run this image as a container within your own OpenShift install, then use this web server for running your own website. This whole process of downloading and running your web server takes no more than a few minutes.

Think of images as cookie cutters and containers as the actual cookies.

Think of OpenShift as an operating system, images as applications that you run on them, and the containers as the actual running instances of those applications.

CHAPTER 2. GET STARTED

To get started, you will need to:

1. Install OpenShift on a physical or virtual RHEL 7+ system
2. Understand the various ways to interact with OpenShift
3. Understand authentication and set up a basic account
4. Set up the Router and the Registry (with storage)
5. Get a base image and run it as a container

CHAPTER 3. INSTALL OPENSIFT ENTERPRISE 3

To install OpenShift Enterprise, you will need:

- At least two physical or virtual RHEL 7+ machines, with fully qualified domain names (either real world or within a network) and password-less ssh access to each other. This guide, uses `master.openshift.example.com` and `node.openshift.example.com`. These machines must be able to ping each other using these domain names.
- A valid Red Hat subscription.
- Wildcard DNS resolution that resolves your domain to the IP of the node. So, an entry like the following in your DNS server:

```
master.openshift.example.com. 300 IN A <IP of the master>
node.openshift.example.com. 300 IN A <IP of the node>
*.apps.openshift.example.com. 300 IN A <IP of the node>
```



WHY THE APPS IN YOUR DOMAIN NAME FOR THE WILDCARD ENTRY?

When using OpenShift to deploy applications, an internal router needs to proxy incoming requests to the corresponding application pod. By using `apps` as part of the application domains, the application traffic is accurately marked to the right pod.

You can use anything other than `apps`. For example,

```
*.cloudapps.openshift.example.com. 300 IN A <IP of the node>
```

will work just as well.

Once you have acquired these items, use these steps to set up a two-machine OpenShift Enterprise install.

3.1. ATTACH OPENSIFT SUBSCRIPTION

1. On the target machines (both master and node), as root, use `subscription-manager` to register the systems with Red Hat:

```
$ subscription-manager register
```

2. List the available subscriptions:

```
$ subscription-manager list --available
```

3. Find the pool ID that provides OpenShift Subscription and attach it:

```
$ subscription-manager attach --pool=<pool_id>
```

4. Replace the string `<pool_id>` with the pool ID of the pool that provides OSE 3. The pool ID is a long alphanumeric string.

These RHEL systems are now authorized to install OpenShift. Now you need to tell the systems where to get OpenShift from.

3.2. SET UP REPOSITORIES

On both master and node, use `subscription-manager` to enable the repositories that are necessary in order to install OpenShift (you may have already enabled the first two repositories in this example):

```
$ subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-extras-rpms" --enable="rhel-7-server-ose-3.2-rpms"
```



NOTE

Instead of `rhel-7-server-ose-3.2-rpms`, use `rhel-7-server-ose-3.3-rpms` for OpenShift Container Platform 3.3 and `rhel-7-server-ose-3.4-rpms` for OpenShift Container Platform 3.4

This command tells your RHEL system that the tools required to install OpenShift will be available from these repositories. Now we need the OpenShift installer that is based on Ansible.

3.3. INSTALL THE OPENSIFT ENTERPRISE PACKAGE

The installer for OpenShift Enterprise is provided by the `atomic-openshift-utils` package. Install it using `yum` on both the master and the node, after running `yum update`:

```
$ yum update
$ yum install atomic-openshift-utils
```

3.4. RUN THE INSTALLER

Before running the installer on the master, set up password-less ssh access as this is required by the installer to gain access to the machines. On the master:

```
$ ssh-keygen
```

Follow the prompts and just hit enter when asked for passphrase. Once done:

```
for host in master.openshift.example.com node.openshift.example.com; do
ssh-copy-id -i ~/.ssh/id_rsa.pub $host; done
```

Then, run the installer on the master:

```
$ atomic-openshift-installer install
```

This is an interactive install process that guides you through the various steps. In most cases, you want the default options. When it starts, select the option for OpenShift Enterprise 3.2 (or OpenShift Container Platform 3.3/3.4). Also, you are installing one master and one node and the domain name is the FQDN as mentioned at the start of this section (`master.openshift.example.com` and `node.openshift.example.com`).



IMPORTANT

At the step where the installer asks you for the FQDN for the routes, you **must** use `apps.openshift.example.com` (or `cloudapps.openshift.example.com` as discussed earlier) and **NOT** `openshift.example.com`. If you make an error, you can edit the `/etc/origin/master/master-config.yaml` at the end of the install process and make this change yourself by looking for the `subdomain` entry.

This install process takes about 5-10 minutes.

3.5. START OPENSIFT

After a successful install, use the following command to start OpenShift:

```
systemctl start atomic-openshift-master
```

Run the following command to verify that OpenShift Enterprise was installed and started successfully. You will get a listing of the master and node, in the **Ready** status.

```
oc get nodes
```

Once installed and started, before you add a new project, you need to set up basic authentication, user access, and routes.

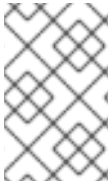
CHAPTER 4. INTERACT WITH OPENSIFT ENTERPRISE 3

OpenShift provides two command line utilities to interact with it.

- `oc`: for normal project and application management
- `oadm`: for administrative tasks

Use `oc --help` and `oadm --help` to view all available options.

In addition, you can use the web console to manage projects and applications. The web console is available at <https://<master-fqdn>:8443/console>. In the next section, you will see how to create user accounts for accessing the console.



NOTE

You can interact with your OpenShift instance from a remote system as well, using these command line utilities. Bundled as the OpenShift CLI, you can download these utilities for Windows, Mac, or Linux environments [here](#).

CHAPTER 5. UNDERSTAND ROLES AND AUTHENTICATION

By default, when installed for the first time, there are no roles or user accounts created in OSE, so you need to create them. You have the option to either create new roles or define a policy that allows anyone to log in (to start you off).

Before you do anything else, log in at least one time with the default `system:admin` user, on the master:

```
$ oc login -u system:admin
```



NOTE

All commands from now on should be executed on the master, unless otherwise indicated.

By logging in at least one time with this account, you will create the `system:admin` user's configuration file, which will allow you to log in subsequently.

There is no password for this system account.

5.1. CHANGE LOG IN IDENTITY PROVIDER

The default behavior of a freshly installed OSE instance is to deny any user from logging in. To change the authentication method to HTPasswd:

1. Open the `/etc/origin/master/master-config.yaml` file in editing mode.
2. Find the `identityProviders` section.
3. Change `DenyAllPasswordIdentityProvider` to `HTPasswdPasswordIdentityProvider` provider.
4. Change the value of the name label to `htpasswd_auth` and add a new line `file: /etc/origin/openshift-passwd` in the provider section.
An example `identityProviders` section with `HTPasswdPasswordIdentityProvider` would look like this:

```
identityProviders:
- challenge: true
  login: true
  name: htpasswd_auth
  provider:
    apiVersion: v1
    file: /etc/origin/openshift-passwd
    kind: HTPasswdPasswordIdentityProvider
```

5. Save the file.

5.2. CREATE USER ACCOUNTS

Now that you are using the `HTPasswdPasswordIdentityProvider` provider, you need to generate these user accounts.

1. You can use the **httpd-tools** package to obtain the **htpasswd** binary that can generate these accounts.

```
yum -y install httpd-tools
```

2. Create a user account:

```
touch /etc/origin/openshift-passwd  
htpasswd -b /etc/origin/openshift-passwd admin redhat
```

Note that you have created a user with the username of **admin** and password of **redhat**.

3. Restart OpenShift before going forward.

```
systemctl restart atomic-openshift-master
```

4. Give this user account **cluster-admin** privileges (which allows it to do everything):

```
oadm policy add-cluster-role-to-user cluster-admin admin
```

5. Now, you can use this username/password combination to log in via the web console or the command line. To test this:

```
oc login -u admin
```

Before going forward, change to the **default** project.

```
oc project default
```

If you need more details on [roles](#) and [authentication](#), see the corresponding sections in the OpenShift docs.

CHAPTER 6. DEPLOY THE OPENSIFT ROUTER

The OpenShift router is the entry point for external network traffic destined for OpenShift services. It supports HTTP, HTTPS, and any TLS-enabled traffic that uses SNI, which enables the router to send traffic to the correct service.

Without the router, OpenShift services and pods are unable to communicate with any resource outside of the OpenShift instance.

From OpenShift 3.2 onwards, the installer creates a default router.

1. Delete the default router using the following command:

```
oc delete all -l router=router
```

2. Create a new default router:

```
$ oadm router --replicas=1 --service-account=router
```

The OpenShift documentation contains detailed information on [router deployment](#).

CHAPTER 7. DEPLOY A DOCKER REGISTRY

OpenShift provides an internal, integrated Docker registry that can be deployed to locally manage images. OpenShift uses the **docker-registry** to store, retrieve, and build Docker images, as well as deploy and manage them throughout their lifecycle.

From OpenShift 3.2 and onwards, the installer creates a default registry.

1. Delete the default registry using the following command:

```
oc delete all -l docker-registry=default
```

2. Create the **docker-registry** service in the **default** project using the **registry** service account:

```
$ oadm registry
```

The OpenShift documentation contains detailed information on [docker registry](#).

CHAPTER 8. CREATE PERSISTENT STORAGE FOR THE REGISTRY

The registry that you created in the previous step stores images and metadata, and uses an ephemeral volume for any pod deployment if persistent storage is not configured. This ephemeral volume is destroyed when the pod exits, losing all data, including any images built or pushed into the registry.

To configure persistent storage for the registry:

- provision a volume that points to a storage server on your network (we will just create it on the master)
- create a volume claim.
- manually add the claim to the registry service.



NOTE

The following steps to configure persistent storage for the registry apply to storage for any image that requires persistent data and not just for the registry. The registry is just another image in the OpenShift environment.

8.1. PROVISION THE PERSISTENT VOLUME

1. Create a registry volume file on your master, as shown here, and call it *registry-volume.yaml*;

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: registry-volume
spec:
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteMany
  nfs:
    path: /root/storage
    server: master.openshift.example.com
```

The folder */root/storage* must exist. Make sure to change the server entry to point to your master.

2. Create the registry persistent volume in OpenShift:

```
$ oc create -f registry-volume.yaml
```

8.2. CREATE THE PERSISTENT VOLUME CLAIM

Create a claim to bind the persistent volume created earlier. This claim is what ties the registry service to the persistent volume.

1. Create another file called *registry-volume-claim.yaml*:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: registry-volume-claim
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 3Gi
```

2. Create the claim:

```
$ oc create -f registry-volume-claim.yaml
```

You have now created the Persistent Volume and the Persistent Volume Claim, and now need to add this claim to the registry.

8.3. ADD THE CLAIM TO THE REGISTRY

```
$ oc volume dc/docker-registry --add --overwrite -t persistentVolumeClaim
--claim-name=registry-volume-claim --name=registry-storage
```

The **docker-registry** will now use the 3 GB persistent volume created above for storing image and metadata.

CHAPTER 9. DEPLOY YOUR FIRST IMAGE

At this stage, you are ready to get a basic Hello World image deployed and running in your OpenShift instance.

1. Log in to the OpenShift web console at <https://master.openshift.example.com:8443/console/>. Use the username and password that you created in the roles and authentication section.
2. Click on **New Project** and enter the name of the project as **hello-openshift**. The next screen will show you the available images to add to your project. Since we only set up storage for the registry image, we will only be able to add those images that don't require persistence.
3. Click on a desired image. For testing purposes, select **jenkins-ephemeral**.
4. You don't need to change any details for this image. Click **Create** to create the image and start the deployment of this image as a pod.
5. Click **Continue to Overview** and wait for the Jenkins pod to be deployed. The pod is deployed once the pod's outline becomes blue and **1** appears in the middle of it. This may take a few minutes.
6. The pod, once deployed, will allow you to play with your Jenkins instance. The link to this instance is displayed in the Overview page, next to the pod and will be similar to <https://jenkins-hello-openshift.apps.openshift.example.com/>.

You have successfully installed OpenShift and deployed your first pod!