



# OpenShift Dedicated 3

## Getting Started

Getting Started with OpenShift Dedicated 3



# OpenShift Dedicated 3 Getting Started

---

Getting Started with OpenShift Dedicated 3

## 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

Whether you are a developer or a platform administrator, you can get started with OpenShift using the topics in this book. Administrators can use the installation utility and an interactive CLI tool, to quickly install and configure a new OpenShift instance across multiple hosts. Developers can use the OpenShift CLI or the web console to log in to an existing OpenShift instance and start creating applications.

---

## Table of Contents

<b>CHAPTER 1. OVERVIEW</b> .....	<b>3</b>
<b>CHAPTER 2. CREATE AND BUILD AN IMAGE USING THE WEB CONSOLE</b> .....	<b>4</b>
2.1. OVERVIEW	4
2.1.1. Browser Requirements	5
2.2. BEFORE YOU BEGIN	5
2.3. FORKING THE SAMPLE REPOSITORY	5
2.4. CREATING A PROJECT	6
2.5. CREATING AN APPLICATION	6
2.6. VERIFY THE APPLICATION IS RUNNING	6
2.7. CONFIGURING AUTOMATED BUILDS	7
2.8. WRITING A CODE CHANGE	7
2.8.1. Manually Rebuilding Images	8
<b>CHAPTER 3. CREATE AND BUILD AN IMAGE USING THE CLI</b> .....	<b>9</b>
3.1. OVERVIEW	9
3.2. BEFORE YOU BEGIN	10
3.3. FORKING THE SAMPLE REPOSITORY	10
3.4. CREATING A PROJECT	10
3.5. CREATING AN APPLICATION	11
3.6. CREATE A ROUTE	12
3.7. VERIFY THE APPLICATION IS RUNNING	12
3.8. CONFIGURING AUTOMATED BUILDS	12
3.9. WRITING A CODE CHANGE	13
3.9.1. Manually Rebuilding Images	13
3.10. TROUBLESHOOTING	13
<b>CHAPTER 4. ADMINISTERING AN OPENSIFT DEDICATED CLUSTER</b> .....	<b>15</b>
4.1. OVERVIEW	15
4.2. DOWNLOADING THE CLI	15
4.3. LOGGING IN AND VERIFYING PERMISSIONS	15
4.4. MANAGING DEDICATED ADMINISTRATORS	16
4.5. GRANTING PERMISSIONS TO USERS OR GROUPS	16
4.6. CREATING SERVICE ACCOUNTS	17
4.7. MANAGING QUOTAS AND LIMIT RANGES	17
4.8. WHAT'S NEXT?	17



## CHAPTER 1. OVERVIEW

If you already have OpenShift Dedicated installed, find the appropriate topic based on your role to get started:

I am a...	Links to relevant topics
Developer	Step through the basics of how to <a href="#">Create and Build an Image Using the Web Console</a> and create your first project and application.
Dedicated administrator	Understand your <a href="#">dedicated administrator</a> role.

## CHAPTER 2. CREATE AND BUILD AN IMAGE USING THE WEB CONSOLE

### 2.1. OVERVIEW

This getting started experience walks you through the simplest way to get a sample project up and running on OpenShift Dedicated. There are a few different ways to launch images within a project, but this topic focuses on the quickest and easiest method.

If this is the first part of the documentation you have read, and you are unfamiliar with the core concepts of OpenShift Dedicated version 3 (v3), you might want to start by reading about [what's new](#). This version of OpenShift Dedicated is significantly different from version 2 (v2).

OpenShift Dedicated 3 provides out of the box a set of [languages](#) and [databases](#) for developers with corresponding implementations and tutorials that allow you to kickstart your application development. Language support centers around the [Quickstart templates](#), which in turn leverage [builder images](#).

Language	Implementations and Tutorials
Ruby	<a href="#">Rails</a>
Python	<a href="#">Django</a>
Node.js	<a href="#">Node.js</a>
PHP	<a href="#">CakePHP</a>
Perl	<a href="#">Dancer</a>
Java	

Other images provided by OpenShift Dedicated include:

- [MySQL](#)
- [MongoDB](#)
- [PostgreSQL](#)
- [Jenkins](#)

In addition, JBoss Middleware has put together a broad range of [OpenShift Dedicated templates](#) as well as [images](#) as part of their xPaaS services.

The technologies available with the xPaaS services in particular include:

- Java EE 6 Application Server provided by JBoss EAP 6
- Integration and Messaging Services provided by JBoss Fuse and JBoss A-MQ
- Data Grid Service provided by JBoss Data Grid



- Real Time Decision Service provided by JBoss BRMS
- Java Web Server 3.0 provided by Tomcat 7 and Tomcat 8

With each of these offerings, a series of combinations are provided:

- HTTP only versus HTTP and HTTPS
- No database required, or the use of either MongoDB, PostgreSQL, or MySQL
- If desired, integration with A-MQ

To help illustrate constructing such applications, the following sections guide you through creating a project that contains a sample Node.js application that will serve a welcome page and the current hit count (stored in a database).



#### NOTE

This topic discusses both [Quickstart](#) and [Instant App](#) templates and applications. Quickstarts provide a starting point for application development, but they rely on your development to create a useful application. In contrast, Instant Apps like Jenkins are instantly usable.

### 2.1.1. Browser Requirements

Review the [browser versions and operating systems](#) that can be used to access the web console.

## 2.2. BEFORE YOU BEGIN

Before you can get started:

- You must be able to access a running instance of OpenShift Dedicated. If you do not have access, contact your cluster administrator.
- Your instance must be pre-configured by a cluster administrator with the [Instant App templates](#) and [builder images](#). If they are not available, contact your cluster administrator.
- You must have the OpenShift Dedicated CLI [downloaded and installed](#).

## 2.3. FORKING THE SAMPLE REPOSITORY

1. Visit the [Ruby example](#) page while you are logged in to GitHub.



#### NOTE

This topic follows the Ruby example, but you can follow along using any of the language examples [provided in the OpenShift Dedicated GitHub project](#).

2. [Fork the repository](#).  
You are redirected to your new fork.
3. Copy the clone URL for your fork.
4. Clone the repository to your local machine.

## 2.4. CREATING A PROJECT

To create an application, you must first create a new project, then select an InstantApp template. From there, OpenShift Dedicated begins the build process and creates a new deployment.

1. Visit the OpenShift Dedicated web console in your browser. The web console uses a self-signed certificate, so if prompted, continue past a browser warning.
2. Log in using the username and password recommended to you by your administrator.
3. To create a new project, click **New Project**.
4. Type a unique name, display name, and description for the new project.
5. Click **Create**.  
The web console's welcome screen loads.

## 2.5. CREATING AN APPLICATION

The Select Image or Template page gives you the option to create from a publicly accessible git repository, or from a template:

1. If creating a new project did not automatically redirect you to the Select Image or Template page, you might need to click **Add to Project**.
2. Click **Browse**, then select **ruby** from the drop-down list.
3. Click the **ruby:latest** builder image.
4. Type a **name** for your application, and specify the **Git Repository URL**, which is [https://github.com/<your\\_github\\_username>/ruby-ex.git](https://github.com/<your_github_username>/ruby-ex.git).
5. Optionally, click **Show advanced routing, build, and deployment options**, though by default this example application automatically creates a route, webhook trigger, and build change triggers.
6. Click **Create**.



### NOTE

After creation, some of these settings can be modified from the web console by clicking **Browse**, **Builds**, select your build, then click **Actions**, and either **Edit** or **Edit YAML**.

Creating your application might take some time. You can follow along on the Overview page of the web console to see the new resources being created, and watch the progress of the build and deployment.

While the Ruby pod is being created, its status is shown as pending. The Ruby pod then starts up and displays its newly-assigned IP address. When the Ruby pod is running, the build is complete.

## 2.6. VERIFY THE APPLICATION IS RUNNING

If your DNS is correctly configured, then your new application can be accessed using a web browser. If you cannot access your application, then speak with your system administrator.


To view your new application:

1. In the web console, view the overview page to determine the web address for the application. For example, under **SERVICE: RUBY-EX** you should see something similar to: **`ruby-ex-my-test.example.openshiftapps.com`**.
2. Visit the web address for your new application.

## 2.7. CONFIGURING AUTOMATED BUILDS

You forked the source code for this application from the [OpenShift Dedicated GitHub repository](#). Therefore, you can use a webhook to automatically trigger a rebuild of your application whenever you push code changes to your forked repository.

To set up a webhook for your application:

1. From the Web Console, navigate to the project containing your application.
2. Click the **Browse** tab, then click **Builds**.
3. Click your build name, then click the **Configuration** tab.
4. Click  next to **GitHub webhook URL** to copy your webhook payload URL.
5. Navigate to your forked repository on GitHub, then click **Settings**.
6. Click **Webhooks & Services**.
7. Click **Add webhook**.
8. Paste your webhook URL into the **Payload URL** field.
9. Click **Add webhook** to save.

GitHub now attempts to send a ping payload to your OpenShift Dedicated server to ensure that communication is successful. If you see a green check mark appear next to your webhook URL, then it is correctly configured. Hover your mouse over the check mark to see the status of the last delivery.

The next time you push a code change to your forked repository, your application will automatically rebuild.

## 2.8. WRITING A CODE CHANGE

To work locally and then push changes to your application:

1. On your local machine, use a text editor to change the sample application's source for the file **`ruby-ex/config.ru`**
2. Make a code change that will be visible from within your application. For example: on line 229, change the title from **Welcome to your Ruby application on OpenShift** to **This is my Awesome OpenShift Application**, then save your changes.
3. Commit the change in git, and push the change to your fork.

If your webhook is correctly configured, your application will immediately rebuild itself based on your changes. Once the rebuild is successful, view your updated application using the route that was created earlier.

Now going forward, all you need to do is push code updates and OpenShift Dedicated handles the rest.

### 2.8.1. Manually Rebuilding Images

You may find it useful to manually rebuild an image if your webhook is not working, or if a build fails and you do not want to change the code before restarting the build. To manually rebuild the image based on your latest committed change to your forked repository:

1. Click the **Browse** tab, then click **Builds**.
2. Find your build, then click **Start Build**.

## CHAPTER 3. CREATE AND BUILD AN IMAGE USING THE CLI

### 3.1. OVERVIEW

This getting started experience walks you through the simplest way to get a sample project up and running on OpenShift Dedicated. There are a few different ways to launch images within a project, but this topic focuses on the quickest and easiest method.

If this is the first part of the documentation you have read, and you are unfamiliar with the core concepts of OpenShift Dedicated version 3 (v3), you might want to start by reading about [what's new](#). This version of OpenShift Dedicated is significantly different from version 2 (v2).

OpenShift Dedicated 3 provides out of the box a set of [languages](#) and [databases](#) for developers with corresponding implementations and tutorials that allow you to kickstart your application development. Language support centers around the [Quickstart templates](#), which in turn leverage [builder images](#).

Language	Implementations and Tutorials
Ruby	<a href="#">Rails</a>
Python	<a href="#">Django</a>
Node.js	<a href="#">Node.js</a>
PHP	<a href="#">CakePHP</a>
Perl	<a href="#">Dancer</a>
Java	

Other images provided by OpenShift Dedicated include:

- [MySQL](#)
- [MongoDB](#)
- [PostgreSQL](#)
- [Jenkins](#)

In addition, JBoss Middleware has put together a broad range of [OpenShift Dedicated templates](#) as well as [images](#) as part of their xPaaS services.

The technologies available with the xPaaS services in particular include:

- Java EE 6 Application Server provided by JBoss EAP 6
- Integration and Messaging Services provided by JBoss Fuse and JBoss A-MQ
- Data Grid Service provided by JBoss Data Grid
- Real Time Decision Service provided by JBoss BRMS

- Java Web Server 3.0 provided by Tomcat 7 and Tomcat 8

With each of these offerings, a series of combinations are provided:

- HTTP only versus HTTP and HTTPS
- No database required, or the use of either MongoDB, PostgreSQL, or MySQL
- If desired, integration with A-MQ

To help illustrate constructing such applications, the following sections guide you through creating a project that contains a sample Node.js application that will serve a welcome page and the current hit count (stored in a database).



#### NOTE

This topic discusses both [Quickstart](#) and [Instant App](#) templates and applications. Quickstarts provide a starting point for application development, but they rely on your development to create a useful application. In contrast, Instant Apps like Jenkins are instantly usable.

## 3.2. BEFORE YOU BEGIN

Before you can get started:

- You must be able to access a running instance of OpenShift Dedicated. If you do not have access, contact your cluster administrator.
- Your instance must be pre-configured by a cluster administrator with the [Instant App templates](#) and [builder images](#). If they are not available, contact your cluster administrator.
- You must have the OpenShift Dedicated CLI [downloaded and installed](#).

## 3.3. FORKING THE SAMPLE REPOSITORY

1. Visit the [Ruby example](#) page while you are logged in to GitHub.



#### NOTE

This topic follows the Ruby example, but you can follow along using any of the language examples [provided in the OpenShift Dedicated GitHub project](#).

2. [Fork the repository](#).  
You are redirected to your new fork.
3. Copy the clone URL for your fork.
4. Clone the repository to your local machine.

## 3.4. CREATING A PROJECT

To create an application, you must create a new project and specify the location of the source. From there, OpenShift Dedicated begins the build process and creates a new deployment.

## 1. Log into OpenShift Dedicated from the CLI:

- With username and password:

```
$ oc login -u=<username> -p=<password> --server=<your-openshift-server> --insecure-skip-tls-verify
```

- With oauth token:

```
$ oc login <https://api.your-openshift-server.com> --token=<tokenID>
```

## 2. To create a new project:

```
$ oc new-project <projectname> --description="<description>" --display-name="<display_name>"
```

After creating the new project, you will be automatically switched to the new project namespace.

### 3.5. CREATING AN APPLICATION

To create a new application from the code in your forked repository:

## 1. Create the application by specifying the source of the code:

```
$ oc new-app openshift/ruby-20-centos7-https://github.com/<your_github_username>/ruby-ex
```

OpenShift Dedicated finds the matching builder image (which in this case is **ruby-20-centos7**) and then creates resources for the application (image stream, build configuration, deployment configuration, service). It also schedules the build.

## 2. Track the progress of the build:

```
$ oc logs -f bc/ruby-ex
```

## 3. Once the build is complete and the resulting image has successfully pushed to your registry, check the status of your application:

```
$ oc status
```

Or you can view the build from the web console.

Creating your application might take some time. You can follow along on the Overview page of the web console to see the new resources being created, and watch the progress of the build and deployment. You can also use the **oc get pods** command to check when the pod is up and running, or the **oc get builds** command to see build statistics.

While the Ruby pod is being created, its status is shown as pending. The Ruby pod then starts up and displays its newly-assigned IP address. When the Ruby pod is running, the build is complete.

The **oc status** command tells you what IP address the service is running; the default port it deploys to is 8080.

## 3.6. CREATE A ROUTE

An OpenShift Dedicated route exposes a service at a host name, so that external clients can reach it by name. To create a route to your new application:

1. Expose a service for **ruby-ex**:

```
$ oc expose service ruby-ex
```

2. View your new route:

```
$ oc get route
```

3. Copy the route location, which will be something like **ruby-ex-my-test.example.openshiftapps.com**.

## 3.7. VERIFY THE APPLICATION IS RUNNING

To view your new application, paste the route location that you copied (in the previous section) into the address bar of your web browser and hit enter.

The example **ruby-ex** application is a simple welcome screen, and contains details on how to deploy code changes, manage your application, and other development resources.

Next, configure automated builds via a GitHub webhook trigger, so that code changes in your forked repository cause your application to rebuild.

## 3.8. CONFIGURING AUTOMATED BUILDS

You forked the source code for this application from the [OpenShift Dedicated GitHub repository](#). Therefore, you can use a webhook to automatically trigger a rebuild of your application whenever you push code changes to your forked repository.

To set up a webhook for your application:

1. View the triggers section of the **BuildConfig** to verify that a GitHub webhook trigger exists:

```
$ oc edit bc/ruby-ex
```

You should see something similar to this:

```
triggers
- github:
  secret: Q1tGY0i9f1ZFihQbX07S
  type: GitHub
```

The secret ensures that only you and your repository can trigger the build.

2. Run the following command to display the webhook URLs associated with your **BuildConfig**:

```
$ oc describe bc ruby-ex
```

3. Copy the GitHub webhook payload URL output by the above command.



4. Navigate to your forked repository on GitHub, then click **Settings**.
5. Click **Webhooks & Services**.
6. Click **Add webhook**.
7. Paste your webhook URL into the **Payload URL** field.
8. Click **Add webhook** to save.

GitHub now attempts to send a ping payload to your OpenShift Dedicated server to ensure that communication is successful. If you see a green check mark appear next to your webhook URL, then it is correctly configured. Hover your mouse over the check mark to see the status of the last delivery.

The next time you push a code change to your forked repository, your application will automatically rebuild.

## 3.9. WRITING A CODE CHANGE

To work locally and then push changes to your application:

1. On your local machine, use a text editor to change the sample application's source for the file ***ruby-ex/config.ru***
2. Make a code change that will be visible from within your application. For example: on line 229, change the title from **Welcome to your Ruby application on OpenShift** to **This is my Awesome OpenShift Application**, then save your changes.
3. Commit the change in git, and push the change to your fork.  
If your webhook is correctly configured, your application will immediately rebuild itself based on your changes. Once the rebuild is successful, view your updated application using the route that was created earlier.

Now going forward, all you need to do is push code updates and OpenShift Dedicated handles the rest.

### 3.9.1. Manually Rebuilding Images

You may find it useful to manually rebuild an image if your webhook is not working, or if a build fails and you do not want to change the code before restarting the build. To manually rebuild the image based on your latest committed change to your forked repository:

```
$ oc start-build ruby-ex
```

## 3.10. TROUBLESHOOTING

### Changing Projects

Although the `oc new-project` command automatically sets your current project to the one you've just created, you can always change projects by running:

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

To view a list of projects:

```
$ oc get projects
```

### Manually Triggering Builds

If the build does not start automatically, start a build and stream the logs:

```
$ oc start-build ruby-ex --follow
```

Alternatively, do not include **--follow** in the above command, and instead issue the following command after triggering the build, where **n** is the number of the build to track:

```
$ oc logs -f build/ruby-ex-n
```

# CHAPTER 4. ADMINISTERING AN OPENSIFT DEDICATED CLUSTER

## 4.1. OVERVIEW

As an administrator of an OpenShift Dedicated cluster, your account has additional permissions and access to all user-created projects in your organization's cluster. While logged in to an account with this role, the basic [developer CLI](#) (the `oc` command) allows you increased visibility and management capabilities over objects across projects, while the [administrator CLI](#) (commands under the `oc adm` command, and formerly the `oadm` command) open up additional operations.



### NOTE

While your account does have these increased permissions, the actual cluster maintenance and host configuration is still performed by the OpenShift Operations Team. If you would like to request a change to your cluster that you cannot perform using the administrator CLI, open a support case on the [Red Hat Customer Portal](#).

## 4.2. DOWNLOADING THE CLI

The `oc` CLI used for both normal developer operations and administrator operations is available for download from the [Command Line Tools](#) page in the web console. See [Get Started with the CLI](#) for more detailed installation steps.

## 4.3. LOGGING IN AND VERIFYING PERMISSIONS

You can log in as an OpenShift Dedicated cluster administration via the web console or CLI, just as you would if you were an application developer.

When you [log in to the web console](#), all user-created projects across the cluster are visible from the main **Projects** page.

Use the standard `oc login` command to log in with the CLI:

```
$ oc login <your_instance_url>
```

All projects are visible using:

```
$ oc get projects
```

When your account has the **dedicated-cluster-admin** cluster role [bound](#) to it, you are automatically bound to the **dedicated-project-admin** for any new projects that are created by users in the cluster.

To verify if your account has administrator privileges, run the following command against a user-created project to view its default [role bindings](#). If you are a cluster administrator, you will see your account listed under subjects for the **dedicated-project-admin** role binding for the project:

```
$ oc describe rolebinding.rbac -n <project_name>
```

```
Name: admin
Labels: <none>
Annotations: <none>
```

```

Role:
  Kind: ClusterRole
  Name: admin
Subjects:
  Kind Name      Namespace
  ---- ----      -
  User fred@example.com 1

Name: dedicated-project-admin
Labels: <none>
Annotations: <none>
Role:
  Kind: ClusterRole
  Name: dedicated-project-admin
Subjects:
  Kind Name      Namespace
  ---- ----      -
  User alice@example.com 2
  User bob@example.com 3
...

```

**1** The **fred@example.com** user is a normal, project-scoped administrator for this project.

**2** **3** The **alice@example.com** and **bob@example.com** users are cluster administrators.

To view details on your increased permissions, and the sets of [verbs and resources](#) associated with the **dedicated-cluster-admin** and **dedicated-project-admin** roles, run the following:

```

$ oc describe clusterrole.rbac dedicated-cluster-admin
$ oc describe clusterrole.rbac dedicated-project-admin

```

## 4.4. MANAGING DEDICATED ADMINISTRATORS

Administrator roles are managed using a **dedicated-admins** group on the cluster. Existing members of this group can edit membership. To view a list of current dedicated administrators by user name, you can use the following command:

```

$ oc describe group dedicated-admins

```

To add a new member to the **dedicated-admins** group:

```

$ oc adm groups add-users dedicated-admins <user_name>

```

To remove an existing user from the **dedicated-admins** group:

```

$ oc adm groups remove-users dedicated-admins <user_name>

```

If this group is currently empty or if you need assistance editing group membership, open a support case on the [Red Hat Customer Portal](#).

## 4.5. GRANTING PERMISSIONS TO USERS OR GROUPS

To grant permissions to other [users or groups](#), you can add, or *bind*, a role to them using the following commands:

```
$ oc adm policy add-role-to-user <role> <user_name>
$ oc adm policy add-role-to-group <role> <group_name>
```

See [Managing RBAC](#) for more details on these and related authorization tasks.

## 4.6. CREATING SERVICE ACCOUNTS

You can create a [service account](#) to be able to run applications like Jenkins that make calls back to OpenShift Dedicated.

See the [Developer Guide](#) for basic service account management tasks, which as a cluster administrator you can perform in any user-created project, and see [Configuring Service Accounts](#) for more advanced, cluster-wide settings.

## 4.7. MANAGING QUOTAS AND LIMIT RANGES

As an administrator, you are able to view, create, and modify [quotas](#) and [limit ranges](#) on other projects. This allows you to better constrain how compute resources and objects are consumed by users across the cluster.

Defaults can be set for quotas and limit ranges for new projects at creation. To request such a change, open a support case on the [Red Hat Customer Portal](#).

## 4.8. WHAT'S NEXT?

Further explore the [Cluster Administration](#) guide for more reference information on what's possible with your role and what other cluster settings can be configured for you by the OpenShift Operations Team.