



Red Hat OpenShift Application Runtimes 1

Getting Started with Red Hat OpenShift Application Runtimes

For Use with Red Hat OpenShift Application Runtimes

Red Hat OpenShift Application Runtimes 1 Getting Started with Red Hat OpenShift Application Runtimes

For Use with Red Hat OpenShift Application Runtimes

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

This guide provides a getting started experience for Red Hat OpenShift Application Runtimes.

Table of Contents

PREFACE	3
CHAPTER 1. INTRODUCTION TO RED HAT OPENSIFT APPLICATION RUNTIMES	4
CHAPTER 2. DEPLOYING A BOOSTER TO OPENSIFT ONLINE	6
CHAPTER 3. CREATING AND DEPLOYING A BOOSTER USING YOUR SINGLE-NODE OPENSIFT CLUSTER	8
CHAPTER 4. DEPLOYING A BOOSTER TO OPENSIFT CONTAINER PLATFORM	9
CHAPTER 5. UPDATING YOUR BOOSTER AND DEPLOYING THE CHANGES	11
CHAPTER 6. USING RED HAT JBOSS DEVELOPER STUDIO WITH A BOOSTER PROJECT	12
6.1. IMPORTING YOUR BOOSTER CODE TO RED HAT JBOSS DEVELOPER STUDIO	12
6.2. COMMITTING AND PUSHING CHANGES TO THE GITHUB REPOSITORY OF YOUR BOOSTER	12
APPENDIX A. GLOSSARY	14
A.1. PRODUCT AND PROJECT NAMES	14
A.2. TERMS SPECIFIC TO FABRIC8 LAUNCHER	14

PREFACE

This guide covers basic information to get started using Red Hat OpenShift Application Runtimes.

Linux containers allow you to use different ecosystems, languages, and runtimes at the same time for solving business problems. Container platforms such as OpenShift manage, secure, and automate the deployment and monitoring of your applications. This enables you to break your business problems into smaller microservices while letting OpenShift provide a simpler way to deploy, monitor, and maintain microservices.

Red Hat OpenShift Application Runtimes provides you with cloud-native runtimes and prescribes approaches to developing, building, and deploying applications that use those cloud-native runtimes on OpenShift. These cloud-native development approaches take advantage of the capabilities provided by OpenShift and enable you to use patterns such as Circuit Breaker, Health Check, and Service Discovery.

This guide takes you through cloud-native development on OpenShift. It shows you how to create and deploy example cloud-native applications called boosters to OpenShift using developers.redhat.com/launch or the Fabric8 Launcher tool. These example applications can serve as the foundation for your actual cloud-native microservices since they can be updated and redeployed using the same deployment process detailed in this guide.

CHAPTER 1. INTRODUCTION TO RED HAT OPENSIFT APPLICATION RUNTIMES

This section provides a brief overview of the foundational concepts behind Red Hat OpenShift Application Runtimes.

Red Hat OpenShift Application Runtimes is:

- A collection of runtimes, such as Thorntail and Spring Boot, designed to run on OpenShift.
- A prescriptive approach to cloud-native development on OpenShift.

The goal of Red Hat OpenShift Application Runtimes is to provide prescriptive architectures, design patterns, tools, and best practices in ready-made example applications that you can execute on OpenShift to enable cloud-native development.

Cloud-native Development

Cloud-native development is an approach to developing, deploying, and running applications that take full advantage of cloud computing.

OpenShift

Red Hat OpenShift is a container application platform that brings [Docker](#) and [Kubernetes](#) to the enterprise. [OpenShift Online](#) is a public cloud hosted by Red Hat. A Single-node OpenShift Cluster is a local cloud that you can install and execute on your local machine; this functionality is provided for example by [Red Hat Container Development Kit](#) or [Minishift](#).

developers.redhat.com/launch

[developers.redhat.com/launch](#) is a project generator offered by Red Hat to accelerate your experience with cloud-native development on OpenShift. It provides a hassle-free way of creating example applications, called boosters, as well as an easy way to build and deploy those boosters to OpenShift. To use [developers.redhat.com/launch](#):

- Navigate to [developers.redhat.com/launch](#).
- Choose the details for your example application.
- Deploy application to OpenShift automatically or manually.

You can find more details on using [developers.redhat.com/launch](#) in the [Chapter 2, Deploying a booster to OpenShift Online](#) section.

The Fabric8 Launcher Tool

The Fabric8 Launcher is the upstream project from which [developers.redhat.com/launch](#) is based.

You can also install and execute the Fabric8 Launcher tool on your Single-node OpenShift Cluster to use the same capabilities within your Single-node OpenShift Cluster. For more information, see the [Installing the Fabric8 Launcher Tool](#) chapter of the [Install and Configure the Fabric8 Launcher Tool](#) guide.

Missions and Boosters

A mission implements a [Microservice pattern](#) such as:

- Health Check

- REST or HTTP endpoint
- [Externalized Configuration](#)
- Circuit Breaker

Missions use these patterns to show how to create the fundamental building blocks of cloud-native applications and services, such as:

- Creating HTTP APIs.
- Integrating your application with a relational database to provide persistent data storage.
- Implementing the health check and circuit-breaker patterns to ensure that your services can withstand a traffic overload and network issues.
- Externalizing the configuration of your applications to make them more secure and easier to scale.

Each mission is implemented in one or more runtimes. Both the specific implementation and the actual project that contains your code are called a booster. Any booster can be updated or extended for your own use case.

Process for Building and Deploying to OpenShift

When using developers.redhat.com/launch, you can create and deploy a booster to OpenShift using the *Build and Deploy to OpenShift* build process, which is based on the [Source-to-Image \(S2I\) build process](#). *Build and Deploy to OpenShift* configures OpenShift to pull the code of your booster from your GitHub repository, build the code, and deploy it to OpenShift.

The benefit of this process is that it handles all the configuration, building, and deployment steps needed to get your booster running in OpenShift. It also allows you to quickly deploy code updates and see your changes in OpenShift.

OpenShift supports different [build strategies](#) including Source-to-Image and Jenkins Pipeline. The Red Hat OpenShift Application Runtimes boosters use the [Source-to-Image strategy](#) for simplicity of use and implementation. Other build strategies, such as the Jenkins Pipeline, are more complex, for example employing multiple servers for building or using webhooks to trigger builds.

Boosters created using the *Build and Deploy to OpenShift* build process can also be executed locally, as well as built locally and manually deployed to OpenShift. For more information on these options, see the **README.adoc** file of your booster.

CHAPTER 2. DEPLOYING A BOOSTER TO OPENSIFT ONLINE

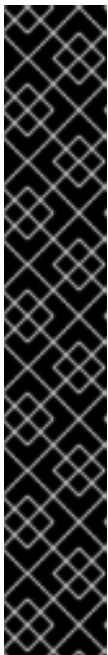
You can quickly create, build, and deploy a booster to OpenShift Online using developers.redhat.com/launch. developers.redhat.com/launch creates a new repository in your GitHub account containing the booster's code and deploys it to your OpenShift Online account. It can also create a ZIP of the booster for you to download and run on your localhost.

Prerequisites

- [Have an OpenShift Online account](#)
- Check which OpenShift Online cluster you belong to using <https://manage.openshift.com>.
- [Have a Red Hat Developers account](#)
- [Have a GitHub account](#)
- [Login to your Red Hat Developers account](#)

Procedure

1. Navigate to developers.redhat.com/launch using your browser.
2. Select *Launch your Project*.
 - You may be prompted to log in to your Red Hat Developers account. If you are, click *Log in* or *register* and complete the authentication steps.
3. You may be prompted to grant developers.redhat.com/launch access to your GitHub and OpenShift Online accounts. This is a one-time action.



IMPORTANT

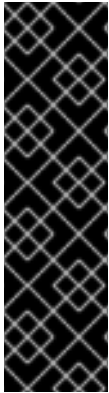
developers.redhat.com/launch uses the **rh-developers-launch** SSO realm to connect your [Red Hat Developers account](#), your GitHub account, and your OpenShift Online account together. When you add these connections, developers.redhat.com/launch is granted access to those accounts and is allowed to use those accounts when creating and deploying boosters.

When connecting these three accounts together in **rh-developers-launch**, they may only be used with each other within the **rh-developers-launch** SSO realm. For example, if your GitHub account is associated with a different Red Hat Developers account within the **rh-developers-launch** SSO realm, you must deauthorize it from the other Red Hat Developers account or add a different GitHub account.

At this time, you cannot use the **us-east-2** OpenShift Starter cluster with developers.redhat.com/launch. This cluster is used with OpenShift.io and is configured differently.

- a. Click *Grant Access*.
- b. Click the *Add* button next to the *GitHub* field and authorize developers.redhat.com/launch to access your GitHub account.

- c. Click the *Add* button next to the OpenShift Online cluster you use.
4. Follow the instructions to create a booster based on a mission, runtime, and target environment.
5. Open the **README.adoc** file in your booster's project and follow the instructions for building, deploying, and interacting with your booster.



IMPORTANT

When you are done interacting with and [updating](#) your booster, it is recommended to delete your project because the resources allocated to your OpenShift Online account are limited. For instructions, see the [Deleting a Project](#) chapter of the OpenShift Online documentation. You can redeploy the booster later using the instructions in the relevant [runtime guide](#).

You have [quotas](#) for your OpenShift Online account. If you exceed your account quota, you will not be able to launch new boosters using developers.redhat.com/launch. The quota for your account varies depending on your subscription.

Additional resources

See the runtime guides to learn more about runtimes and boosters that are available for them:

- [Spring Boot Runtime Guide](#)
- [Eclipse Vert.x Runtime Guide](#)
- [Thorntail Runtime Guide](#)
- [Node.js Runtime Guide](#)

CHAPTER 3. CREATING AND DEPLOYING A BOOSTER USING YOUR SINGLE-NODE OPENSIFT CLUSTER

You can quickly create, build, and deploy a booster to a Single-node OpenShift Cluster using the Fabric8 Launcher tool. The Fabric8 Launcher tool creates a new repository in your GitHub account containing the booster's code and deploys it to your Single-node OpenShift Cluster. It can also create a ZIP of the booster for you to download and run on your localhost.

Prerequisites

- [Have a Red Hat Developers account](#)
- [Have a GitHub account](#)
- [Have a Single-node OpenShift Cluster installed](#)
- [Have the Fabric8 Launcher tool installed to your Single-node OpenShift Cluster.](#)

Procedure

1. Navigate to the Fabric8 Launcher tool on your Single-node OpenShift Cluster using your browser.
 - Optionally, use the following command to open the Web Console of your Single-node OpenShift Cluster in your browser in order to navigate to your Fabric8 Launcher tool:

```
$ minishift console
```
2. Select *Launch Your Project*.
3. Follow the instructions to create a booster based on a mission, runtime, and target environment.
4. Open the **README.adoc** file in your booster project for more details about interacting with your booster.

Additional resources

Take a look the different runtime guides to learn more about runtimes as well as their boosters:

- [Spring Boot Runtime Guide](#)
- [Eclipse Vert.x Runtime Guide](#)
- [Thorntail Runtime Guide](#)
- [Node.js Runtime Guide](#)

CHAPTER 4. DEPLOYING A BOOSTER TO OPENSIFT CONTAINER PLATFORM

Prerequisites

- Have a booster created using developers.redhat.com/launch or the Fabric8 Launcher tool in [a Single-node OpenShift Cluster](#).
- Have access to an OpenShift Container Platform Web console.
- Have the **oc** CLI client installed.

Procedure

1. Clone your booster Github repository, or download and extract the booster ZIP archive.
2. Get the command containing your authentication token for using the **oc** CLI client with your OpenShift Container Platform Web Console:
 - a. Login to your OpenShift Container Platform Web console.
 - b. Click on the question mark icon in the top right-hand corner of the Web console, next to your user name.
 - c. Select *Command Line Tools* in the drop-down menu.
 - d. Find the text box that contains the **oc login** ... command with the hidden token, and click the button next to it to copy its content to your clipboard.
3. Paste the command you copied in the previous step into a terminal to authenticate your **oc** CLI client with your OpenShift Container Platform account by using your authentication token.

```
$ oc login OCP_URL --token=MYTOKEN
```

4. Create a new project.

```
$ oc new-project MY_PROJECT_NAME
```

5. Navigate to the root directory of your booster.
6. Deploy your booster to OpenShift Container Platform.

- a. If your booster uses Maven:

```
$ mvn clean fabric8:deploy -Popenshift
```

- b. If your booster uses **npm**.

```
$ npm install && npm run openshift
```

7. Open the **README.adoc** file in your booster's project and follow the instructions for interacting with your booster.

Additional resources

Take a look at the different runtime guides to learn more about runtimes as well as their boosters:

- [Spring Boot Runtime Guide](#)
- [Eclipse Vert.x Runtime Guide](#)
- [Thorntail Runtime Guide](#)
- [Node.js Runtime Guide](#)

CHAPTER 5. UPDATING YOUR BOOSTER AND DEPLOYING THE CHANGES

To push changes to this booster running in OpenShift Online or your Single-node OpenShift Cluster, you can push a new commit to the GitHub repository and start a new build.

Prerequisites

- Have a booster deployed to [OpenShift Online](#) or [Single-node OpenShift Cluster](#) using the *Deploy in OpenShift Online* option.

Procedure

1. Clone your project from GitHub:

```
$ git clone git@github.com:USERNAME/PROJECT-NAME.git
```

2. Open your project in your desired IDE or editor, such as [JBDS](#).

3. Make an update to the project.

4. Save your changes.

5. Commit and push your changes back to GitHub:

```
$ git add .  
$ git commit -m "Made an update"  
$ git push
```

6. Navigate back to the OpenShift Online or Single-node OpenShift Cluster Web console and ensure the project of your mission is selected.

7. Click on the three dots on the right side of your application's deployment and choose *Start Build*. Your deployment is complete when you see *Build #2 is complete* under *Builds* and a **#2** next to the application name under *Deployment*. This number will increase for each additional build and deployment.

8. Confirm your changes when your deployment is complete.

CHAPTER 6. USING RED HAT JBOSS DEVELOPER STUDIO WITH A BOOSTER PROJECT

Red Hat JBoss Developer Studio is an integrated development environment, or IDE, that provides tooling for writing software. You can use it to make changes to your booster's code.

Prerequisites

- Have a booster deployed to [OpenShift Online](#) or [Single-node OpenShift Cluster](#) using the *Deploy in OpenShift Online* option.
- Have Red Hat JBoss Developer Studio [downloaded from the Red Hat Developer Portal](#) and installed.

6.1. IMPORTING YOUR BOOSTER CODE TO RED HAT JBOSS DEVELOPER STUDIO

Importing your booster's code creates a project in Red Hat JBoss Developer Studio and enables you to start making changes.

Prerequisites

- Your booster created and downloaded from developers.redhat.com/launch.
- Red Hat JBoss Developer Studio running.

Procedure

1. In Developer Studio, Click *File* → *Import*.
2. Choose *Git* → *Projects from Git* and click *Next*.
3. Choose *Existing local repository* and click *Next*.
4. Click *Add*.
5. Click *Browse*.
6. Navigate to the project folder of your booster and click *Open*.
7. Click the check box next to your booster project and click *Finish*.
8. Make sure your booster project is selected and click *Next*.
9. Select *Import as general project* and click *Next*.
10. Click *Finish*.
11. Start editing your project after your import is complete.

6.2. COMMITTING AND PUSHING CHANGES TO THE GITHUB REPOSITORY OF YOUR BOOSTER

Red Hat JBoss Developer Studio provides tooling for committing and pushing code through Git directly in the IDE. Alternatively, you can use Red Hat JBoss Developer Studio to make changes and push those changes with the Git CLI.

Prerequisites

1. All files with changes that you intend to commit are saved.

Procedure

1. In Developer Studio, right click on the root of the project *Project Explorer* or *Package Explorer* view.
2. Choose *Team* → *Commit...*
3. Drag all the changes you want to commit from the *Unstaged Changes* section to the *Staged Changes* section.
4. Enter your commit message and click *Commit and Push*.
5. Click *Close* in the *Push Results* window when prompted.
6. Observe your changes are now available in your GitHub repository.

APPENDIX A. GLOSSARY

A.1. PRODUCT AND PROJECT NAMES

developers.redhat.com/launch

developers.redhat.com/launch is a standalone getting started experience offered by Red Hat for jumpstarting cloud-native application development on OpenShift. It provides a hassle-free way of creating functional example applications, called missions, as well as an easy way to build and deploy those missions to OpenShift.

Fabric8 Launcher

The Fabric8 Launcher is the upstream project on which developers.redhat.com/launch is based.

Single-node OpenShift Cluster

An OpenShift cluster running on your machine using Minishift.

A.2. TERMS SPECIFIC TO FABRIC8 LAUNCHER

Booster

A language-specific implementation of a particular [mission](#) on a particular [runtime](#). Boosters are listed in a [booster catalog](#).

For example, a booster is a web service with a REST API implemented using the Thorntail runtime.

Booster Catalog

A Git repository that contains information about boosters.

Mission

An application specification, for example *a web service with a REST API*.

Missions generally do not specify which language or platform they should run on; the description only contains the intended functionality.

Runtime

A platform that executes [boosters](#). For example, Thorntail or Eclipse Vert.x.