



Red Hat build of Quarkus 1.7

Deploying your Quarkus applications as
OpenShift Serverless services

Red Hat build of Quarkus 1.7 Deploying your Quarkus applications as OpenShift Serverless services

Legal Notice

Copyright © 2021 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 guide describes how to deploy a Quarkus application on Red Hat OpenShift Container Platform using Serverless Serving.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. DEPLOYING A SERVERLESS QUARKUS JAVA APPLICATION TO OPENSIFT USING THE OPENSIFT EXTENSION	5
CHAPTER 2. DEPLOYING A CONTAINER IMAGE FOR A QUARKUS NATIVE APPLICATION IN A CONTINUOUS INTEGRATION AS A SERVERLESS APPLICATION	7
CHAPTER 3. ADDITIONAL RESOURCES	9

PREFACE

As an application developer, you can deploy a Quarkus application on Red Hat OpenShift Container Platform using OpenShift Serverless Knative Serving.

By using OpenShift Serverless Knative Serving, you can scale services up and down depending on the load size. Scaling down services that are currently not requested improves memory capabilities.



NOTE

Quarkus is supported to run as a native executable or as a Java application using an OpenJDK. For native executables, use the Red Hat UBI 8 minimal image. For OpenJDK, use the Red Hat 8 UBI Java image.

Prerequisites

- OpenJDK 11 is installed and the **JAVA_HOME** environment variable specifies the location of the Java SDK. Red Hat build of Open JDK is available from the [Software Downloads](#) page in the Red Hat Customer Portal (login required).
- Apache Maven 3.6.2 or higher is installed. Maven is available from the [Apache Maven Project](#) website.
- You have a Quarkus Maven project. For instructions on building a simple Quarkus application with Maven, see [Getting started with Quarkus](#).



NOTE

For a completed example of a Quarkus Maven project, download the [Quarkus quickstart archive](#) or clone the **Quarkus Quickstarts** Git repository. The example is in the **getting-started** directory.

- You have access to a Red Hat OpenShift Container Platform cluster and the latest version of the OpenShift CLI (oc) is installed. For information about installing oc, see the "Installing the CLI" section of the [Installing and configuring OpenShift Container Platform clusters](#) guide.
- OpenShift Serverless operator is installed and verified. See the [Installing OpenShift Serverless](#) guide for more details.
- OpenShift Knative Serving is installed and verified. See the [Installing Knative Serving](#) section of the [Installing OpenShift Serverless](#) guide for more details.
- For native compilation, a container environment like Podman or Docker is required.
- Optionally install the **kn** CLI tool.



NOTE

Currently, only Knative Serving is supported in the Red Hat build of Quarkus. Knative Eventing is currently tech preview.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. DEPLOYING A SERVERLESS QUARKUS JAVA APPLICATION TO OPENSIFT USING THE OPENSIFT EXTENSION

The traditional source-to-image (S2I) source workflow generates the deployable artifacts of your application inside OpenShift. The Quarkus OpenShift extension uses the S2I binary workflow to provide a more streamlined deployment process. Instead of building from the source, the extension uploads the JAR files from the local file system. As a result, the build process is up to ten times faster than the traditional S2I method. You can use the Quarkus OpenShift extension when developing locally as well as from a build server or continuous integration (CI) system to perform repeatable builds from source.

The following procedure demonstrates how to deploy a Serverless Quarkus Java application to OpenShift using the Quarkus OpenShift extension.

Prerequisites

- You have a Quarkus Maven project.
- OpenShift Serverless operator is installed.
- OpenShift Knative Serving is installed.
- You have access to a Red Hat OpenShift Container Platform cluster and the latest version of the OpenShift CLI (oc) is installed.
- OpenJDK 11 is installed and the **JAVA_HOME** environment variable specifies the location of the Java SDK.
- Apache Maven 3.6.2 or higher is installed. Maven is available from the Apache Maven Project website.
- For native compilation, a container environment like Podman or Docker is required.
- The **kn** CLI tool is installed.

Procedure

1. Change to the directory that contains your Quarkus project.
2. Add the OpenShift extension to an existing project:

```
./mvnw quarkus:add-extension -Dextensions="openshift"
```

When you add the OpenShift extension, the script adds the following dependency to the **pom.xml** file:

```
<dependency>  
<groupId>io.quarkus</groupId>  
<artifactId>quarkus-openshift</artifactId>  
</dependency>
```

3. If you are using an untrusted certificate while in a development or testing environment, add the following line to the **src/main/resources/application.properties** file:

```
quarkus.kubernetes-client.trust-certs=true
```

- To direct OpenShift to use the Open JDK 11 Red Hat Enterprise Linux 8 image, add the following line to the **application.properties** file:

```
quarkus.s2i.base-jvm-image=registry.access.redhat.com/ubi8/openjdk-11
```

- To direct OpenShift Serverless to pull your container image from the OpenShift internal registry, add the following line to the **application.properties** file:

```
quarkus.container-image.registry=image-registry.openshift-image-registry.svc:5000
```

- To enable the creation of deployment descriptors for Knative Serving, add the following line to the **application.properties** file:

```
quarkus.kubernetes.deployment-target=knative
```

- Save the changes to the **application.properties** file.

- Log in to the OpenShift CLI (oc):

```
oc login
```

- To create a new OpenShift project, enter the following command where **<project_name>** is the name of your new project:

```
oc new-project <project_name>
```



NOTE

The **<project_name>** should be the same as the **username** of the host system otherwise Quarkus cannot pull the image from the image registry. If you set a different **<project_name>**, make sure to add the following line to the **application.properties** file:

```
quarkus.container-image.group=<project_name>
```

- To deploy your project to OpenShift, enter the following command:

```
./mvnw clean package -Dquarkus.kubernetes.deploy=true
```

- To verify that the application was deployed correctly, run the following command:

```
kn service list
```

- To view the full URL to the application, enter the following command where **<application_name>** is the name of an application deployed in your OpenShift project:

```
kn route describe <application_name> -o jsonpath='{.status.url}'
```

CHAPTER 2. DEPLOYING A CONTAINER IMAGE FOR A QUARKUS NATIVE APPLICATION IN A CONTINUOUS INTEGRATION AS A SERVERLESS APPLICATION

You can separate the native build, container build, and deployment steps when deploying a native serverless application. The following procedure demonstrates how to deploy a container image for a Quarkus native application in a continuous integration (CI) as a serverless application.

Prerequisites

- OpenShift Serverless operator is installed.
- OpenShift Knative Serving is installed.
- For native compilation, a container environment like Podman or Docker is required.
- The **kn** CLI tool is installed.

Procedure

1. Change to the directory that contains your Quarkus project.
2. Build a Linux executable using one of the following methods:

- a. For Docker use:

```
./mvnw package -Pnative -Dquarkus.native.container-build=true -
Dquarkus.native.builder-image=registry.access.redhat.com/quarkus/mandrel-20-
rhel8:20.3
```

- b. For Podman use:

```
./mvnw package -Pnative -Dquarkus.native.container-build=true -
Dquarkus.native.container-runtime=podman -Dquarkus.native.builder-
image=registry.access.redhat.com/quarkus/mandrel-20-rhel8:20.3
```

3. Open the **src/main/docker/Dockerfile.native** file and set the **<image_name>** and **<version>** parameters:

- a. For Docker use:

```
docker build -f src/main/docker/Dockerfile.native -t <image_name>:<version> .
```

- b. For Podman use:

```
podman build -f src/main/docker/Dockerfile.native -t <image_name>:<version>.
```

4. Push the container to a repository that your CI environment and your OpenShift environment can access, where **<registry>** is your registry URL:

- a. For Docker use:

```
docker tag <image_name>:<version> <registry>/<image_name>:<version>  
docker push <registry>/<image_name>:<version>
```

- b. For Podman use:

```
podman tag <image_name>:<version> <registry>/<image_name>:<version>  
podman push <registry>/<image_name>:<version>
```

5. Log in to the OpenShift CLI (oc):

```
oc login
```

6. To create a new OpenShift project, enter the following command where **<project_name>** is the name of your new project:

```
oc new-project <project_name>
```

7. To deploy your container as a serverless application using the OpenShift Serverless CLI (kn), enter the following command where **<service_name>** is the name for your service:

```
kn service create <service_name> --image REPOSITORY/<image_name>:<version>
```

8. To verify that the service is ready, enter the following command.

```
kn service list <service_name>
```

The output in the column called "READY" reads **true** if the service is ready.



NOTE

The **kn service** command returns **true** when the necessary components are created, not when the image is pulled down and ready.

CHAPTER 3. ADDITIONAL RESOURCES

- For information about creating Quarkus applications with Maven, see [Developing and compiling your Quarkus applications with Apache Maven](#).
- For more information on how to compile the Quarkus applications to native executables and native executable testing, see [Compiling your Quarkus applications to native executables](#)
- For information about deploying Quarkus Maven applications on Red Hat OpenShift Container Platform, see [Deploying your Quarkus applications on Red Hat OpenShift Container Platform](#) .

Revised on 2021-04-19 12:03:15 UTC