Red Hat OpenShift Application Runtimes 1

Install and Configure the Fabric8 Launcher Tool

For Use with Red Hat OpenShift Application Runtimes

Last Updated: 2018-10-24

Red Hat OpenShift Application Runtimes 1 Install and Configure the Fabric8 Launcher Tool

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 instructions for installing the Fabric8 Launcher tool on a Single-node OpenShift Cluster.

Table of Contents

PREFACE 3

CHAPTER 1. HOW THE FABRIC8 LAUNCHER TOOL WORKS 4

CHAPTER 2. INSTALLING A SINGLE-NODE OPENSHIFT CLUSTER 5

CHAPTER 3. STARTING AND CONFIGURING THE SINGLE-NODE OPENSHIFT CLUSTER FOR THE FABRIC8 LAUNCHER TOOL 7

CHAPTER 4. CREATING A GITHUB PERSONAL ACCESS TOKEN 9

CHAPTER 5. INSTALLING FABRIC8 LAUNCHER TOOL 10

5.1. INSTALLING FABRIC8 LAUNCHER TOOL AS A MINISHIFT ADD-ON 10

5.2. INSTALLING FABRIC8 LAUNCHER TOOL MANUALLY 11

APPENDIX A. USING A NEXUS REPOSITORY SERVER ON A SINGLE-NODE OPENSHIFT CLUSTER . . . 15

APPENDIX B. GLOSSARY 18

B.1. PRODUCT AND PROJECT NAMES 18

B.2. TERMS SPECIFIC TO FABRIC8 LAUNCHER 18

PREFACE

This guide walks you through the process of installing the Fabric8 Launcher tool to run on a local cloud as provisioned by a Single-node OpenShift Cluster. This includes Minishift, an all-in-one VM that includes a community version of OpenShift Origin, or Red Hat Container Development Kit, a VM that includes OpenShift Container Platform.

CHAPTER 1. HOW THE FABRIC8 LAUNCHER TOOL WORKS

The Fabric8 Launcher tool runs on OpenShift and lets you create functional example applications called missions. The Fabric8 Launcher tool walks you through:

- choosing the mission you want to generate,
- choosing the runtime you want to use, and
- choosing how you want to build and execute the mission.

The Fabric8 Launcher tool uses your choices to generate a custom project, called a booster, and either launches it directly to the same OpenShift instance, or provides a downloadable ZIP version of the project.

CHAPTER 2. INSTALLING A SINGLE-NODE OPENSHIFT CLUSTER

To use the Fabric8 Launcher tool on a local cloud, you must have a Single-node OpenShift Cluster installed and configured. You can use either Minishift or Red Hat Container Development Kit.

Prerequisites

- Have a Red Hat Developers account

Procedure

1. Follow the instructions for installing the Single-node OpenShift Cluster:
   - The installation steps for Minishift are available in the OpenShift documentation.
   - The installation steps for Red Hat Container Development Kit are available in the Red Hat Container Development Kit Installation Guide.

   NOTE

   The steps for installing Single-node OpenShift Cluster vary by platform.

2. Verify you have the Single-node OpenShift Cluster installed and configured:

```
$ minishift version
```

3. Start and stop the Single-node OpenShift Cluster:

```
$ minishift start
...
$ minishift stop
Stopping local OpenShift cluster...
Cluster stopped.
```

4. Determine the command to add the correct version of the **oc** binary to your path and run the command:

   Example Result of `oc-env` on Red Hat Container Development Kit

```
$ minishift oc-env
export PATH="/Users/john/.minishift/cache/oc/v3.5.5.8:$PATH"
# Run this command to configure your shell:
# eval $(minishift oc-env)

$ eval $(minishift oc-env)
```

WARNING

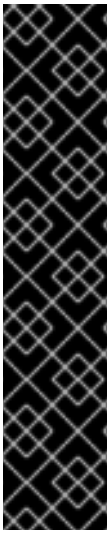
You must have the **oc** binary installed and it *must* match the version of the Single-node OpenShift Cluster you are using.

CHAPTER 3. STARTING AND CONFIGURING THE SINGLE-NODE OPENSHIFT CLUSTER FOR THE FABRIC8 LAUNCHER TOOL

This chapter contains instructions for starting the Single-node OpenShift Cluster and configuring it to execute the Fabric8 Launcher tool.

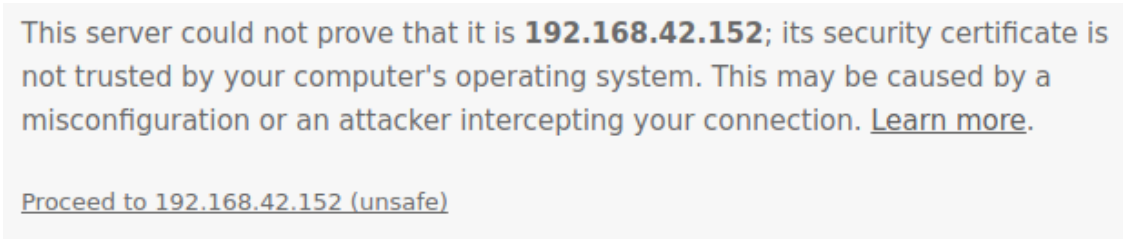
NOTE

Starting your Single-node OpenShift Cluster can trigger a download of large virtual machines or Linux container images. This can take a long time. Subsequent startups are expected to be shorter as long as the virtual machines and Linux container images remain cached.



IMPORTANT

Because a Single-node OpenShift Cluster is intended for development purposes, it uses HTTPS for the web console and only provides a self-signed certificate. If your browser prevents you from accessing the page due to an SSL error, you must allow your browser to bypass SSL security policies for the Single-node OpenShift Cluster URL to use it. The screenshot below shows the warning message in the Google Chrome browser.



Prerequisites

- Single-node OpenShift Cluster installed.

Procedure

1. Start the Single-node OpenShift Cluster with the default virtual machine driver:

NOTE

Depending on your operating system, virtual machine driver, and the number of boosters you run, the memory allocated Single-node OpenShift Cluster can be insufficient. In this case, increase the memory allocation.

```
$ minishift start --memory=4096
...
OpenShift server started.

The server is accessible via web console at:
    https://192.168.42.152:8443
```

Alternatively, specify a virtual machine driver other than the default using the **`--vm-driver`** flag:

```
$ minishift start --memory=4096 --vm-driver=virtualbox
```

Depending on your system configuration, it is possible that you must manually specify an alternative virtual machine driver. You must have virtual machine software, such as VirtualBox, installed before you specify it.

NOTE

On macOS, the default virtual machine driver, **`xhyve`**, can be unreliable. If you experience issues, specifying **`VirtualBox`** is a reliable alternative.

2. Open the Single-node OpenShift Cluster Web console.

```
$ minishift console
```

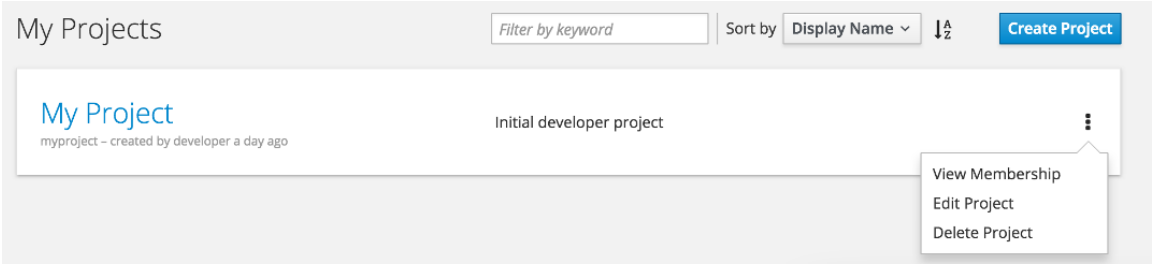
Alternatively, use the URL provided in the log information.

3. Log in using the **`developer`** username and an arbitrary password.

4. Optionally, delete the preconfigured project:

   a. Next to the project name, click the three-dot menu icon.

   b. Select *Delete Project*.



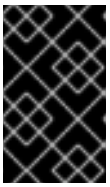
CHAPTER 4. CREATING A GITHUB PERSONAL ACCESS TOKEN

To install the Fabric8 Launcher tool on a Single-node OpenShift Cluster, you must provide the Fabric8 Launcher tool with a GitHub personal access token. This enables the Fabric8 Launcher tool to create booster applications and save them as Git repositories in your GitHub namespace.

- A GitHub account.

Procedure

1. Using a web browser, navigate to **`https://github.com/settings/tokens`**.
2. Select *Generate new token.*
3. Add a token description, for example **`Fabric8 Launcher tool on a Single-node OpenShift Cluster`**.
4. Select the check boxes of the following parent scopes and all their children:
   - **`public_repo`**
   - **`read:org`**
   - **`admin:repo_hook`**
5. Click *Generate token*.
6. Save the hex code of the personal access token. You need this to complete the installation of the Fabric8 Launcher tool on your Single-node OpenShift Cluster.



IMPORTANT

This hex code is displayed only once and cannot be retrieved after you leave the page. If you lose the code, you will need to create a new personal access token to install the Fabric8 Launcher tool on a Single-node OpenShift Cluster again.

CHAPTER 5. INSTALLING FABRIC8 LAUNCHER TOOL

Install a local customized instance of the Fabric8 Launcher tool, which allows you to test the functionality or make modifications to the service using a web interface.

5.1. INSTALLING FABRIC8 LAUNCHER TOOL AS A MINISHIFT ADD-ON

Add-ons enable you to extend the behavior of Single-node OpenShift Cluster, and the Fabric8 Launcher tool provides an add-on as an installation option.

Prerequisites

- Single-node OpenShift Cluster running.
- A GitHub personal access token.

Procedure

1. Ensure you are running a minishift instance which has at least 4 GB memory to run launcher

```
$ minishift config set memory 4096
$ minishift start
```

2. Download and install the Fabric8 Launcher add-on.

```
$ git clone git@github.com:minishift/minishift-addons.git
$ minishift addon install minishift-addons/add-ons/fabric8-launcher/
$ minishift addons apply fabric8-launcher \
       --addon-env GITHUB_USERNAME=GH_USERNAME \
       --addon-env GITHUB_TOKEN=TOKEN (1)
```

(1) Use your GitHub username and personal access token

Installing the add-on creates a new project called **rhoarpad** where the Fabric8 Launcher tool runs.

3. Monitor the status of the Fabric8 Launcher tool until it completes start up.

```
$ oc project rhoarpad
$ oc get pods -w
NAME                              READY     STATUS
RESTARTS   AGE
configmapcontroller-1-deploy   1/1       Running             0
46s
configmapcontroller-1-aaaaa    0/1       ContainerCreating   0
44s
launcher-backend-1-deploy      1/1       Running             0
46s
...
launcher-backend-2-aaaaa       0/1       Running             0
5s
launcher-frontend-2-aaaaa      0/1       Running             0
6s
```

4. Obtain the route of your Fabric8 Launcher tool.

```
$ oc get routes
NAME       HOST/PORT                                   PATH
SERVICES            PORT      TERMINATION   WILDCARD
launcher   launcher-rhoarpad.LOCAL_OPENSHIFT_HOSTNAME
launcher-frontend   <all>                   None
```

5. Navigate to your Fabric8 Launcher tool and start using your Fabric8 Launcher tool to launch booster applications.
This is the same service as **`https://developers.redhat.com/launch`** but running in a Single-node OpenShift Cluster.

Additional Resources

- See the Getting Started with Red Hat OpenShift Application Runtimes for a walk-through of running a booster application.
- Read the runtime guides for an overview of the runtimes and their boosters:
  - Spring Boot Runtime Guide
  - Eclipse Vert.x Runtime Guide
  - Thorntail Runtime Guide
  - Node.js Runtime Guide

5.2. INSTALLING FABRIC8 LAUNCHER TOOL MANUALLY

Install a local customized instance of the Fabric8 Launcher tool, which allows you to test the functionality or make modifications to the service using a web interface.

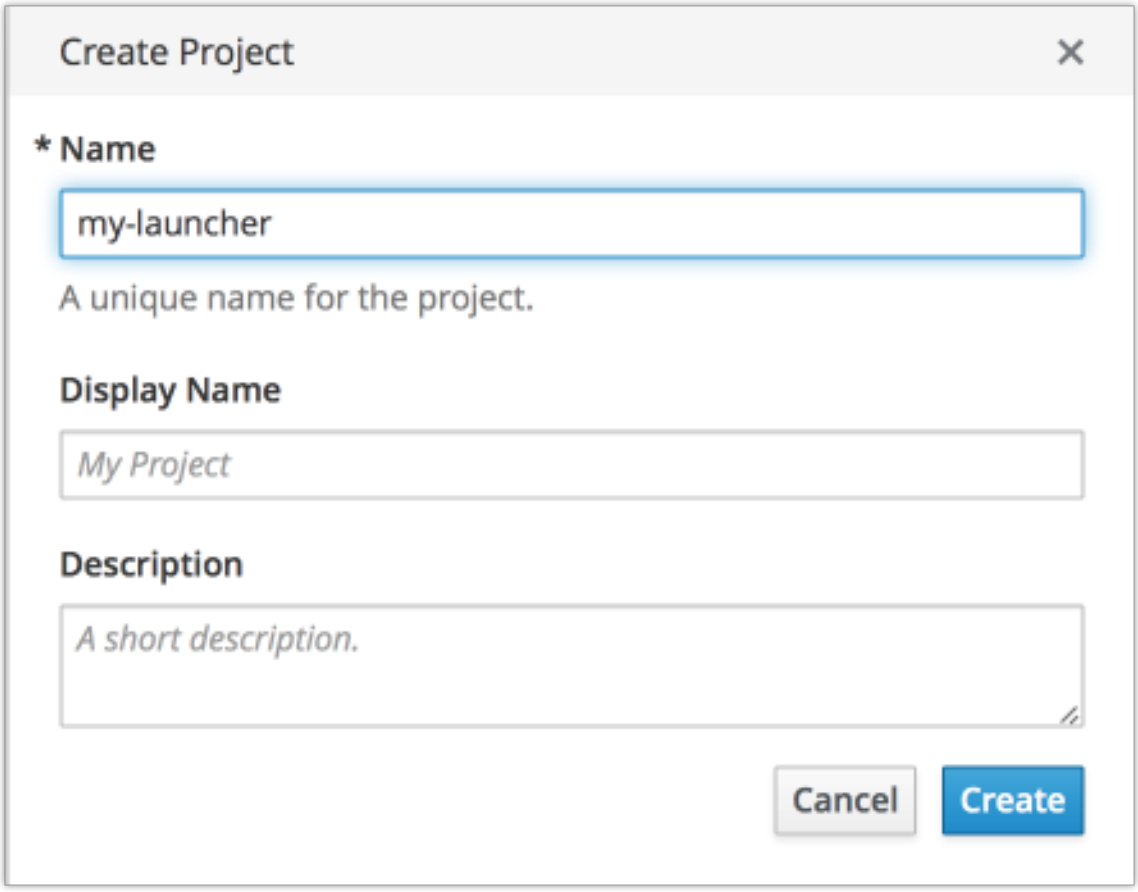
Prerequisites

- Single-node OpenShift Cluster running.
- A GitHub personal access token.

Procedure

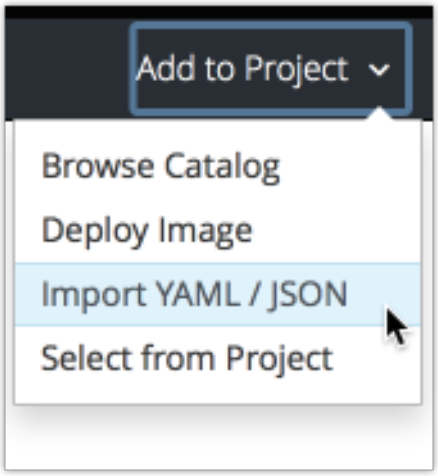
1. Open the Single-node OpenShift Cluster Web console and log in.
2. Click *New Project* to create a new OpenShift project to house the Fabric8 Launcher tool.

3. Name the project and optionally provide a description. This example uses **`my-launcher`** for the project's name.



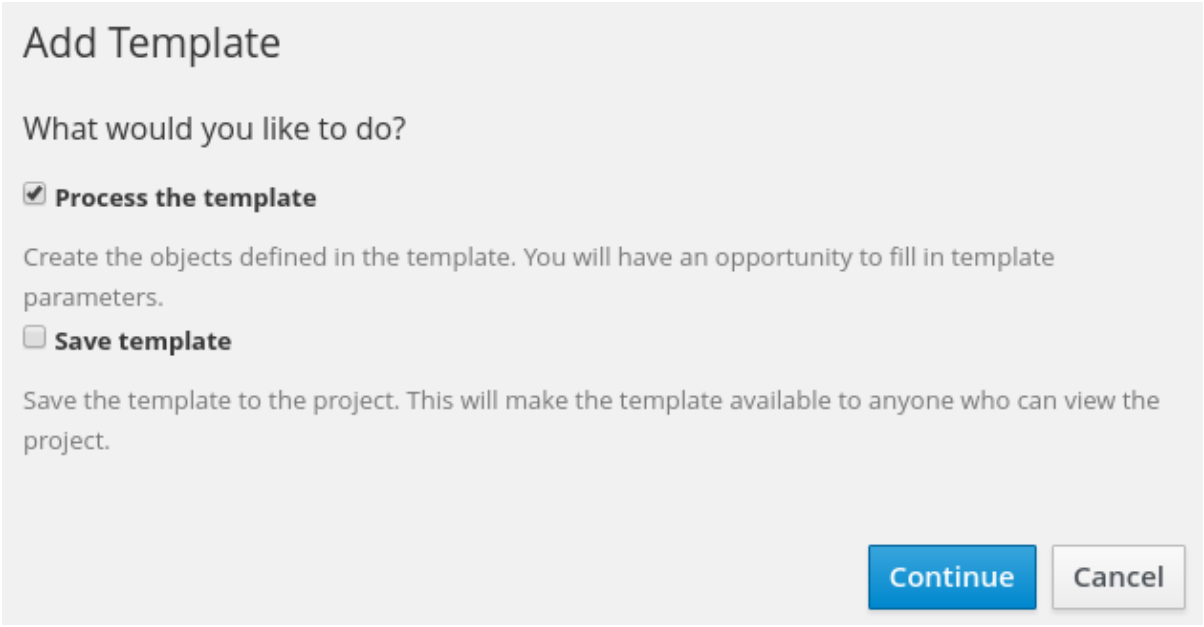
4. Click *Create* to complete the project creation.

5. Click *Import YAML/JSON* to add services to your new project from a template.



6. Copy the contents of the current Fabric8 Launcher template from the GitHub repository and paste it into the text box provided.

7. Click *Create*, ensure that only the *Process the template* option is selected, and click *Continue*.



8. Fill out the following fields.

   - *Your GitHub username*.
   - *Your GitHub Mission Control access token* is your personal access token for GitHub.
   - The *Target OpenShift Console URL* is the OpenShift Console URL from your Single-node OpenShift Cluster. This should be the same base URL you are currently using to complete the form, for example **`https://192.168.42.152:8443`**.
   - OpenShift username and password from your Single-node OpenShift Cluster, for example **`developer`** for the username and password.
   - *KeyCloak URL* and *KeyCloak Realm* **MUST** be cleared out.

WARNING

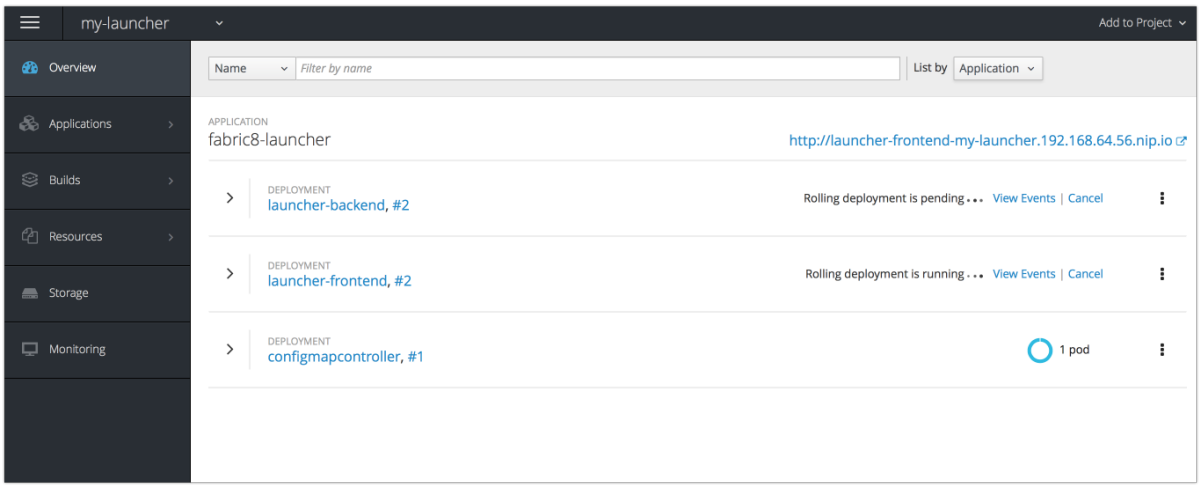
You must clear these fields out for the Fabric8 Launcher tool on your Single-node OpenShift Cluster to be configured correctly.

- Do not modify *Catalog Git Repository* and *Catalog Git Reference* unless you are developing against a specific catalog repository.

9. Before proceeding to the next steps, confirm all the fields are correct. Also confirm that *KeyCloak URL* and *KeyCloak Realm* **have been cleared out**.

10. Click **`Create`** to complete the setup. You will see a screen confirming that the service has been created. Click *Continue to overview*.

11. On the overview page, wait and confirm that the four pods for the Fabric8 Launcher tool have completed starting up.



12. When all pods are running, click the link at the top of all pods, which typically ends in **`nip.io`**. A new browser tab opens with the Fabric8 Launcher tool. This is the same service as **`https://developers.redhat.com/launch`** but running in a Single-node OpenShift Cluster.

13. Start using your Fabric8 Launcher tool to launch booster applications.

Additional resources

- See the Getting Started with Red Hat OpenShift Application Runtimes for a walk-through of running a booster application.
- Read the runtime guides for an overview of the runtimes and their boosters:
  - Spring Boot Runtime Guide
  - Eclipse Vert.x Runtime Guide
  - Thorntail Runtime Guide
  - Node.js Runtime Guide

APPENDIX A. USING A NEXUS REPOSITORY SERVER ON A SINGLE-NODE OPENSHIFT CLUSTER

While developing your cloud-native applications with Java and Maven, you may be required to build them repeatedly. You can deploy a Nexus Repository server alongside the Fabric8 Launcher tool on your Single-node OpenShift Cluster and use it to fetch artifacts from the Maven Central repository and cache them locally. This reduces the network load when building your application, and helps accelerate the build and rolling updates.

Prerequisites

- Your Single-node OpenShift Cluster set up. Follow the instructions in Install and Configure the Fabric8 Launcher Tool.
- Your Single-node OpenShift Cluster configured to use at least 4096 MiB of RAM and use the required **oc** CLI tool version.

```
minishift delete # Delete the previous instance of Single-node
OpenShift Cluster
minishift config set memory 4096
minishift start
```

WARNING

The procedure described below works with Minishift. It has not been tested for use with CDK.

- The Fabric8 Launcher tool deployed to your Single-node OpenShift Cluster.
- A booster application deployed to your Single-node OpenShift Cluster.

Procedure

1. Log in to your Single-node OpenShift Cluster instance.

```
oc login https://LOCAL_OPENSHIFT_URL:PORT -u developer -p developer
```

You can reuse the Docker daemon instance used by Single-node OpenShift Cluster to download the latest versions of the Nexus Docker container image.

```
eval $(minishift docker-env)
docker pull openshift/jenkins-2-centos7
docker pull openshiftio/launchpad-jenkins-slave
docker pull sonatype/nexus
```

2. Create a new project to contain the Nexus server. You can also use the *New Project* button on the Web console to do this.

```
oc new-project NEXUS_PROJECT_NAME
```

3. Deploy the Nexus container image.

```
oc new-app sonatype/nexus
```

4. Expose the service route URL of the Nexus server.

```
oc expose svc/nexus
```

5. Attach a persistent volume claim with a minimum size of 10 GiB to the pod running your Nexus application.

```
oc volumes dc/nexus --add --name 'nexus-volume-1' --type 'pvc' --
mount-path '/sonatype-work/' --claim-name 'nexus-pv' --claim-size
'10Gi' --overwrite
```

6. Navigate to the project containing your booster application.

```
oc project MY_PROJECT_NAME
```

7. Define and start a new build.
Pass in a parameter to set the value of the **`MAVEN_MIRROR_URL`** to match the service URL of your Nexus application:

NOTE

Ensure that the YAML template of the builder image you are using for your application has the **`MAVEN_MIRROR_URL`** environment variable defined. If it does not, see the Nexus documentation for instructions on configuring your build manually before proceeding.

```
$ oc new-build MY_APP_NAME:latest~SCM_REPOSITORY_URL \
-e
MAVEN_MIRROR_URL='http://nexus.NEXUS_PROJECT_NAME:8081/nexus/content
/groups/public'
```

Nexus comes pre-configured for the Maven Central repository, but you may need other repositories for your application. To access images provided by Red Hat, add the Red Hat JBoss Middleware Early Access Maven Repository to your Nexus instance.

8. Ensure that your new build is using Nexus to retrieve artifacts. Do one of the following:

- Navigate to **`http://nexus-NEXUS_PROJECT_NAME.LOCAL_OPENSHIFT_HOSTNAME/nexus/content/groups/public`** to view the list of artifacts stored in your local repository.
- Check the log output of the build:

```
$ oc logs build/MY_APP_NAME-1 --follow
```

If your build uses Nexus to retrieve artifacts, the build log output should reference the path **`http://nexus.NEXUS_PROJECT_NAME:8081/`**.

APPENDIX B. GLOSSARY

B.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.

B.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.