Search

Introduction to OpenShift Applications

Tell us what you think!

Congratulations on completing at least 25% of the course. We would appreciate feedback on your learning experience. Click below to fill out a brief survey.

TAKE ME TO THE SURVEY (HTTPS://SURVEY.OLE.REDHAT.COM/SURVEY?COURSE=DO101&MODALITY=COURSE&OFFERING=101)

NOT RIGHT NOW

□ Don't ask again

TRANSLATIONS ▾

P (/rol/app/courses/do101-4.2/pages/pr01) (/rol/app/courses/do101-4.2/pages/pr01) (/rol/app/courses/do101-4.2/pages/pr01s02) 1 (/rol/app/courses/do101-4.2/pages/ch01) (/rol/app/courses/do101-4.2/pages/ch01)
(/rol/app/courses/do101-4.2/pages/ch01s02) (/rol/app/courses/do101-4.2/pages/ch01s03) (/rol/app/courses/do101-4.2/pages/ch01s04)
(/rol/app/courses/do101-4.2/pages/ch01s05) (/rol/app/courses/do101-4.2/pages/ch01s06) (/rol/app/courses/do101-4.2/pages/ch01s07) 2 (/rol/a
(/rol/app/courses/do101-4.2/pages/ch02) (/rol/app/courses/do101-4.2/pages/ch02s02) (/rol/app/courses/do101-4.2/pages/ch02s03) 3 (/rol/a 4.2/pa
(/rol/app/courses/do101-4.2/pages/ch03) (/rol/app/courses/do101-4.2/pages/ch03s02) (/rol/app/courses/do101-4.2/pages/ch03s03) 4.2/pa
(/rol/app/courses/do101-4.2/pages/ch03s04) (/rol/app/courses/do101-4.2/pages/ch03s05) (/rol/app/courses/do101-4.2/pages/ch03s06)
(/rol/app/courses/do101-4.2/pages/ch03s07) 4 (/rol/app/courses/do101-4.2/pages/ch04) (/rol/app/courses/do101-4.2/pages/ch04) (/rol/app/courses/do101-4.2/pages/ch04s02)
(/rol/app/courses/do101-4.2/pages/ch04s03) 5 (/rol/app/courses/do101-4.2/pages/ch05) (/rol/app/courses/do101-4.2/pages/ch05) (/rol/app/courses/do101-4.2/pages/ch05s02)
(/rol/app/courses/do101-4.2/pages/ch05s03)

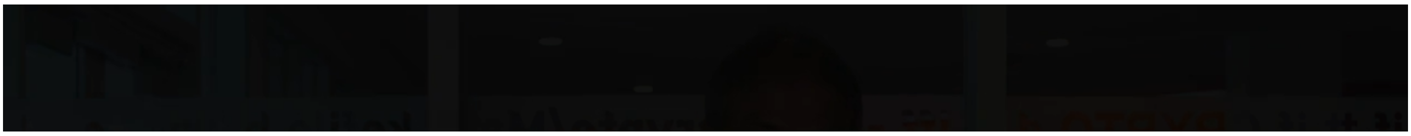
➜ NEXT (/ROL/APP/COURSES/DO101-4.2/PAGES/CH02S02)

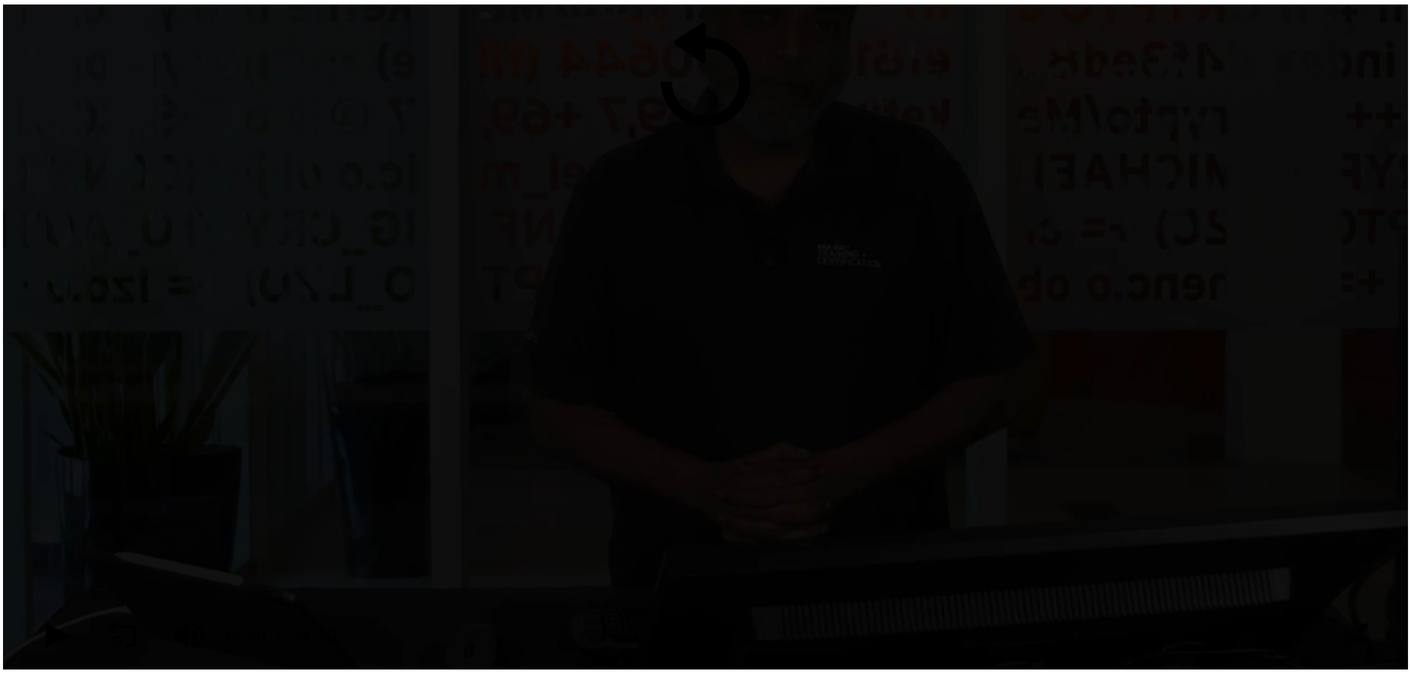
Chapter 2. Deploying Applications to Red Hat OpenShift Container Platform

Abstract

Goal	Deploy an application to OpenShift.
Objectives	Deploy an application to OpenShift.
Sections	Deploying an Application to Red Hat OpenShift Container Platform (and Guided Exercise)

Deploying an Application to Red Hat OpenShift Container Platform





Objectives

After completing this section, you should be able to deploy an application to OpenShift.

Introducing OpenShift Container Platform

Red Hat OpenShift Container Platform is a self-service platform where development teams can deploy their applications. The platform integrates the tools to build and run applications, and manages the complete application life cycle from initial development to production.

OpenShift offers several deployment scenarios. One typical workflow starts when a developer provides the Git repository URL for an application to OpenShift.

The platform automatically retrieves the source code from Git, and then builds and deploys the application. The developer can also configure OpenShift to detect new Git commits, and then automatically rebuild and redeploy the application.

By automating the build and deployment processes, OpenShift allows developers to focus on application design and development. By rebuilding your application with every change you commit, OpenShift gives you immediate feedback. You can detect and fix errors early in the development process, before they become an issue in production.

OpenShift provides the building mechanisms, libraries, and runtime environments for the most popular languages, such as Java, Ruby, Python, PHP, .NET, Node.js, and many more. It also comes with a collection of additional services that you can directly use for your application, such as databases.

As traffic and load to your web application increases, OpenShift can rapidly provision and deploy new instances of the application components. For the Operations team, it provides additional tools for logging and monitoring.

OpenShift Container Platform Architecture

Red Hat OpenShift Online, at https://www.openshift.com/ (https://www.openshift.com/), is a public OpenShift instance run by Red Hat. With that cloud platform, customers can directly deploy their applications online, without needing to install, manage, and update their own instance of the platform.

Red Hat also provides the Red Hat OpenShift Container Platform that companies can deploy on their own infrastructure. By deploying your own instance of OpenShift Container Platform, you can fine tune the cluster performance specific to your needs. In this classroom, you will be provided access to a private OpenShift cluster.

Application Architecture

Several development teams or customers usually share the same OpenShift platform. For security and isolation between projects and customers, OpenShift builds and runs applications in isolated `containers`.

A container is a way to package an application with all its dependencies, such as runtime environments and libraries.

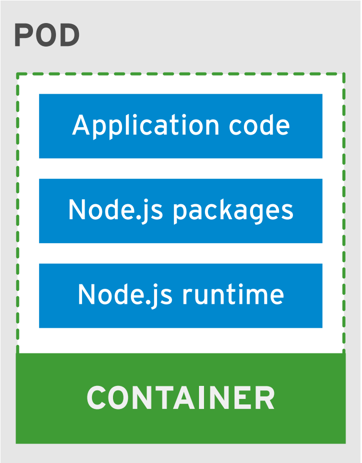


Figure 2.1: A container in a pod

The previous diagram shows a container for a Node.js application. The container groups the Node.js runtime, the Node.js packages required by the application, and the application code itself.

To manage the containerized applications, OpenShift adds a layer of abstraction known as the `pod`.

Pods are the basic unit of work for OpenShift. A pod encapsulates a container, and other parameters, such as a unique IP address or storage. A pod can also group several related containers that share resources.

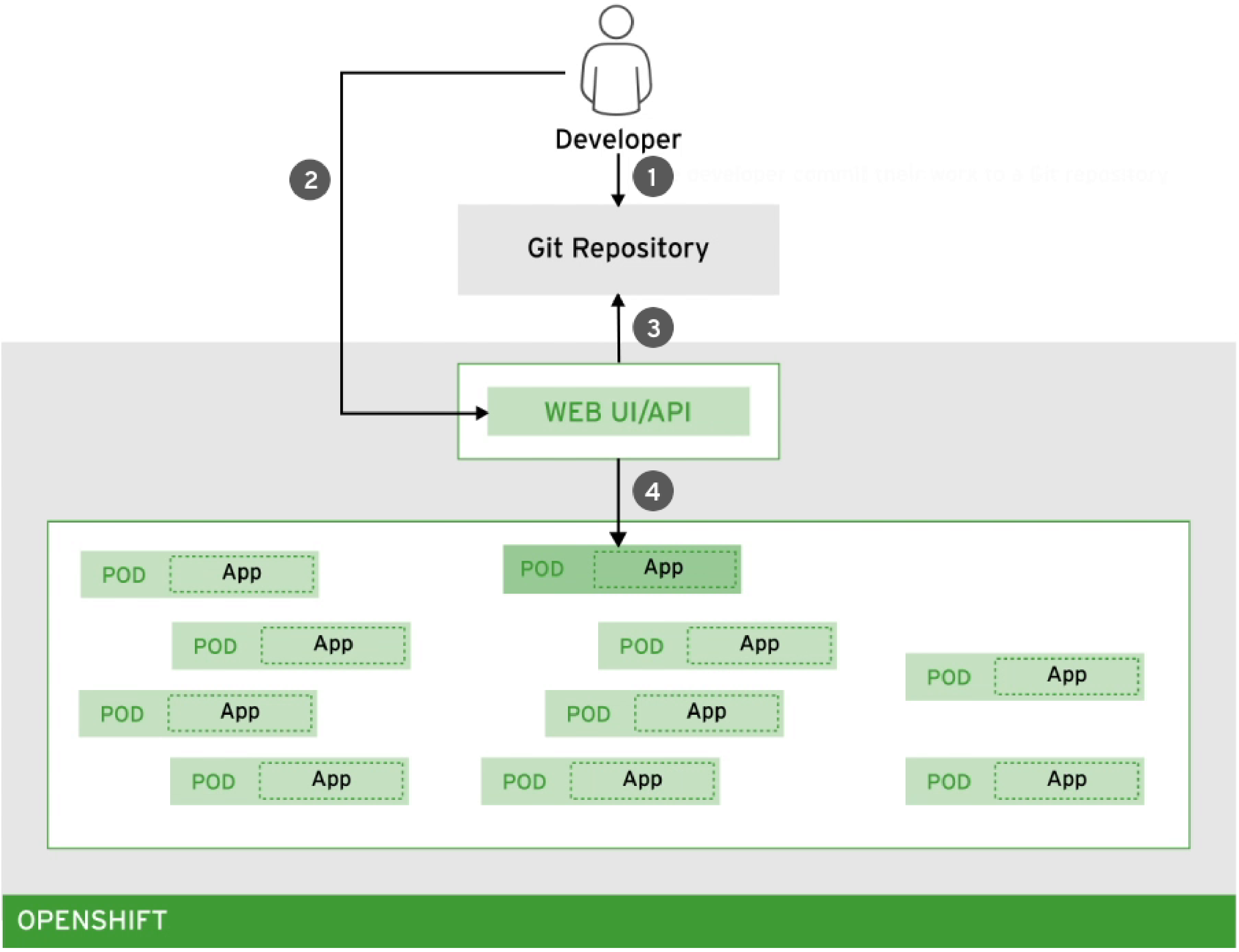


Figure 2.2: Pods running on OpenShift

The preceding diagram shows an OpenShift platform, hosting several applications, and running in pods. To deploy a new application, use the following workflow:

1. The developers commit work to a Git repository.

2. When ready to deploy their code, the developers use the OpenShift `web console` to create the application. The URL to the Git repository is one of the required parameters.
3. OpenShift retrieves the code from the Git repository and builds the application.
4. OpenShift deploys the application in a pod.

OpenShift Resource Types

OpenShift uses `resources` to describe the components of an application. When you deploy a new application, OpenShift creates those resources for you, and you can view and edit them through the web console.

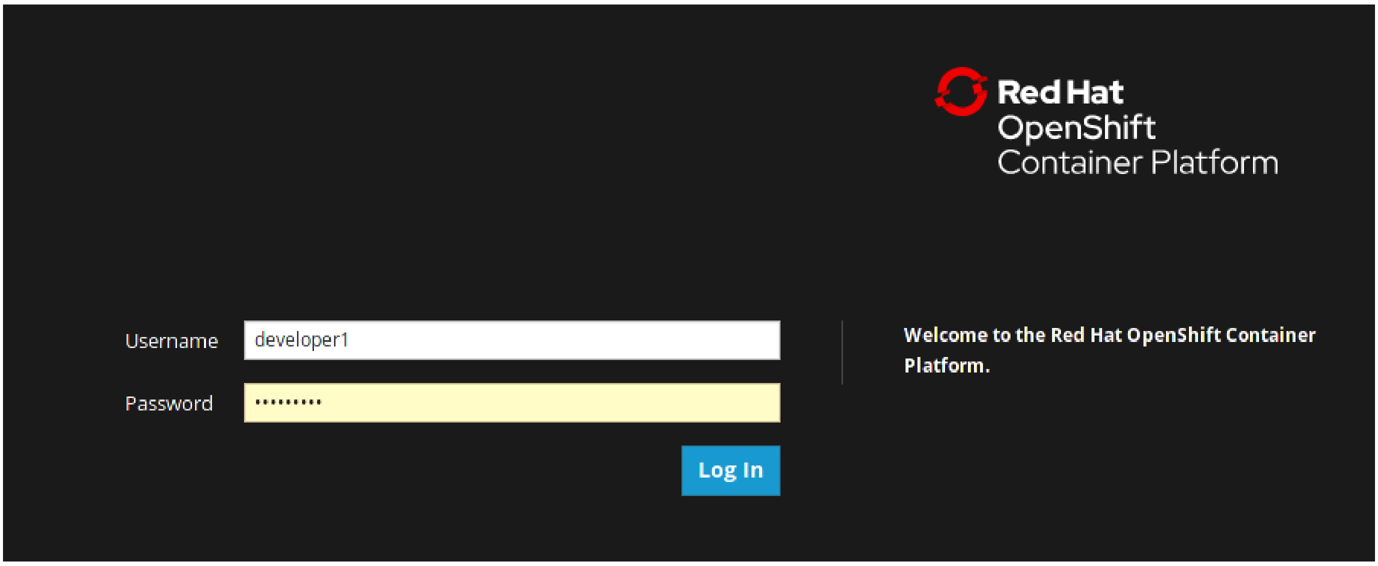
For example, the Pod resource type represents a container running on the platform. A Route resource associates a public URL to the application, so your customers can reach it from outside OpenShift.

Introducing the Developer Web Console for OpenShift

The OpenShift web console is a browser-based user interface that provides a graphical alternative to the command-line tools. With the web UI, developers can easily deploy and manage their applications.

Logging in and Accessing the Developer Perspective

To access the web console, use the URL of your OpenShift platform. To use OpenShift, each developer must have an account. The following screen capture shows the login page.



Red Hat OpenShift Container Platform

Username developer1

Password •••••••••

Welcome to the Red Hat OpenShift Container Platform.

Figure 2.3: Logging in to OpenShift

The web console provides two perspectives, one for developers, and the other for administrators and operators. As shown in the following screen capture, the perspective is selected from the menu on the left. As a developer, you usually select the `Developer` perspective.

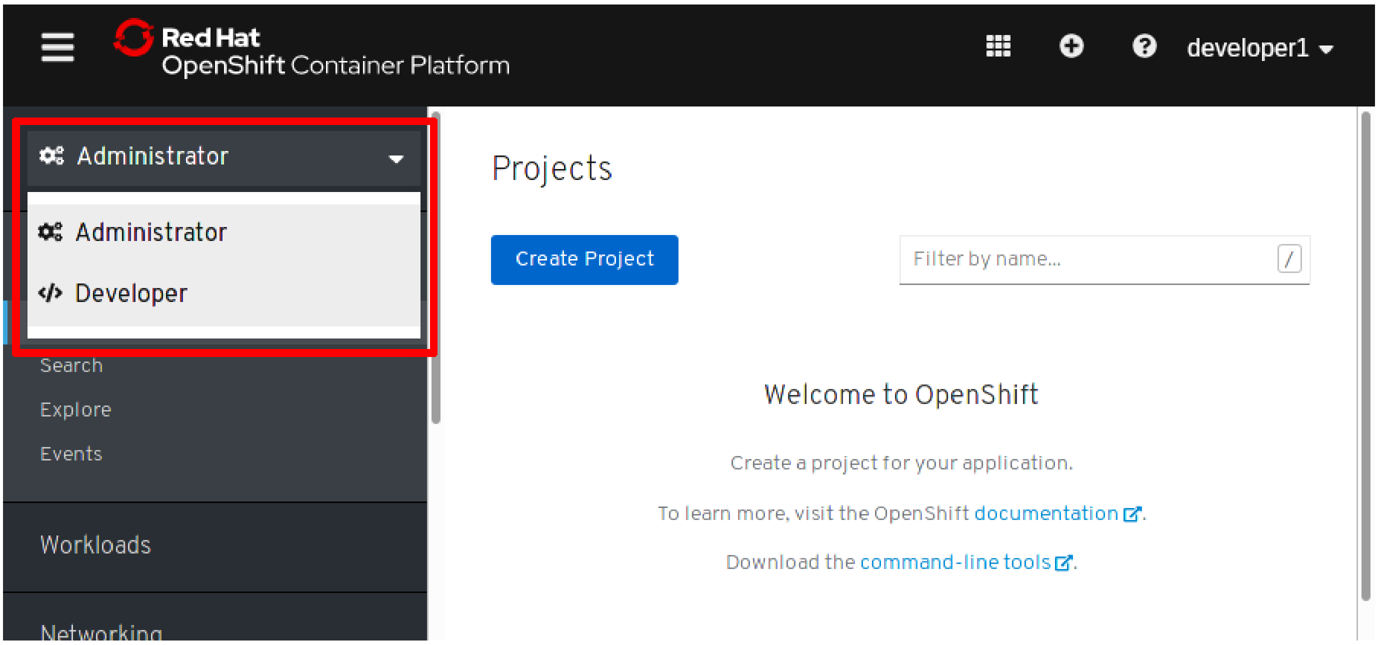


Figure 2.4: The Administrator and Developer perspectives

Creating a Project

OpenShift groups applications in `projects`. Using projects, developer teams can organize content in isolation from other teams. Projects enable both grouping the individual components of an application (front end, back end, and database), and creating life cycle environments (development, QA, production).

To create a project, use the **Advanced** → **Projects** menu.

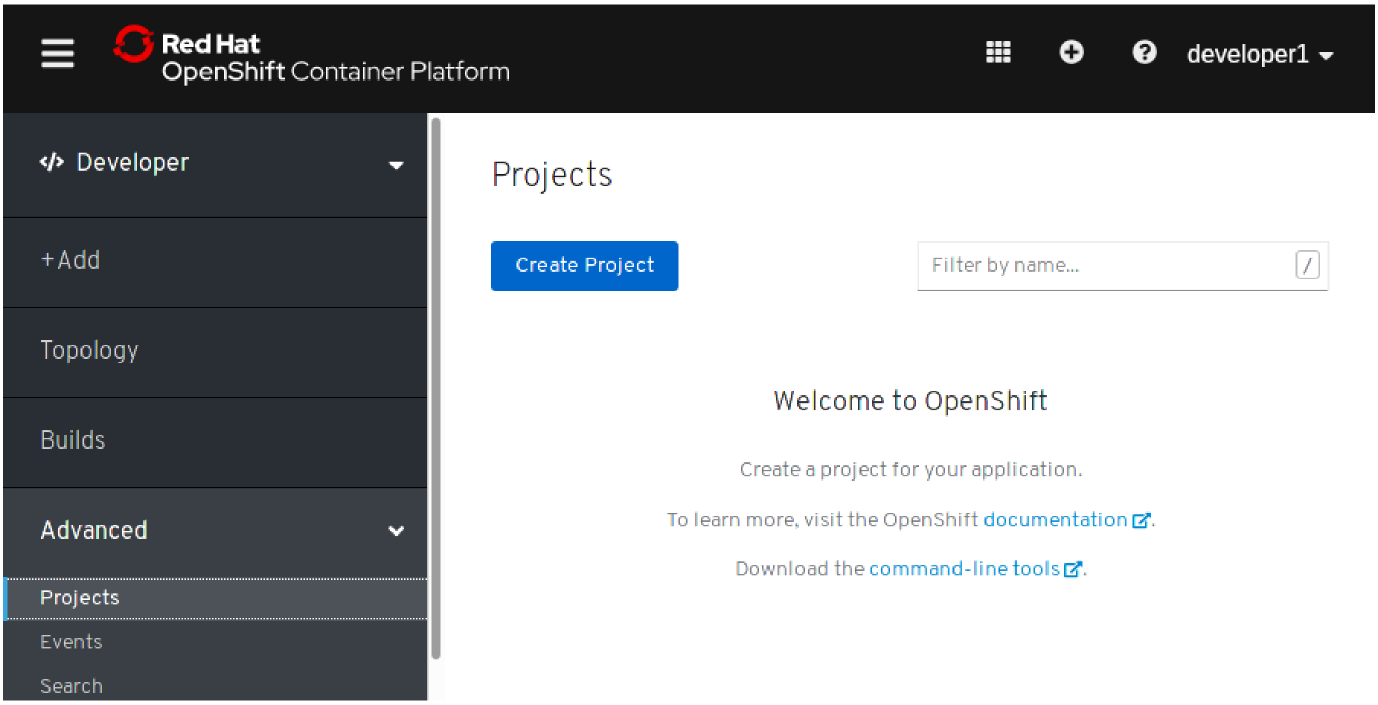
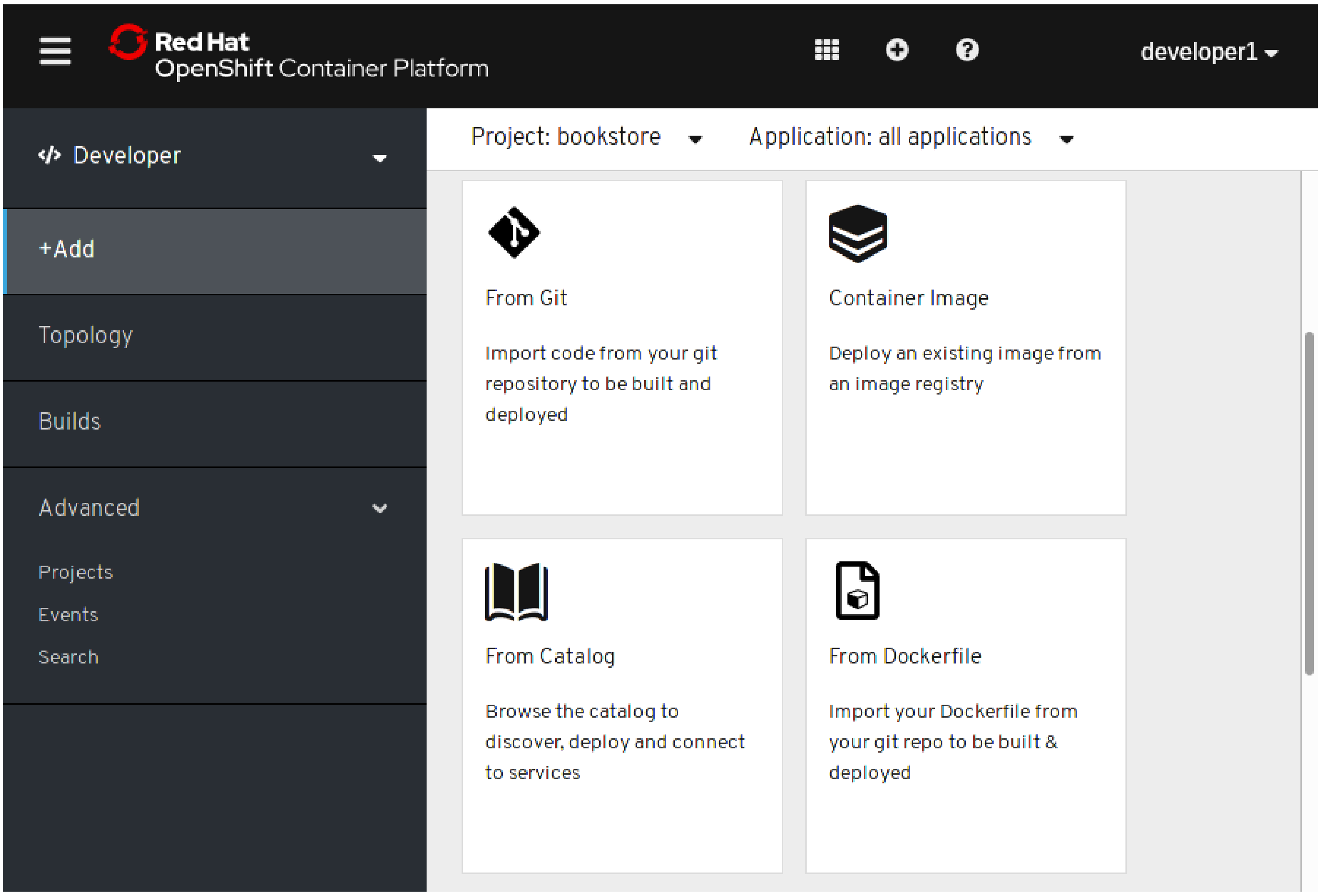

Red Hat
OpenShift Container Platform
developer1
Developer
+Add
Topology
Builds
Advanced
Projects
Events
Search
Projects
Create Project
Filter by name...
Welcome to OpenShift
Create a project for your application.
To learn more, visit the OpenShift documentation.
Download the command-line tools.

Figure 2.5: Creating a project

Deploying a New Application

OpenShift provides several methods to add a new application. The `Add` option is the entry point to an assistant that allows you to choose between the available methods to deploy an application to the OpenShift cluster as part of a specific project.

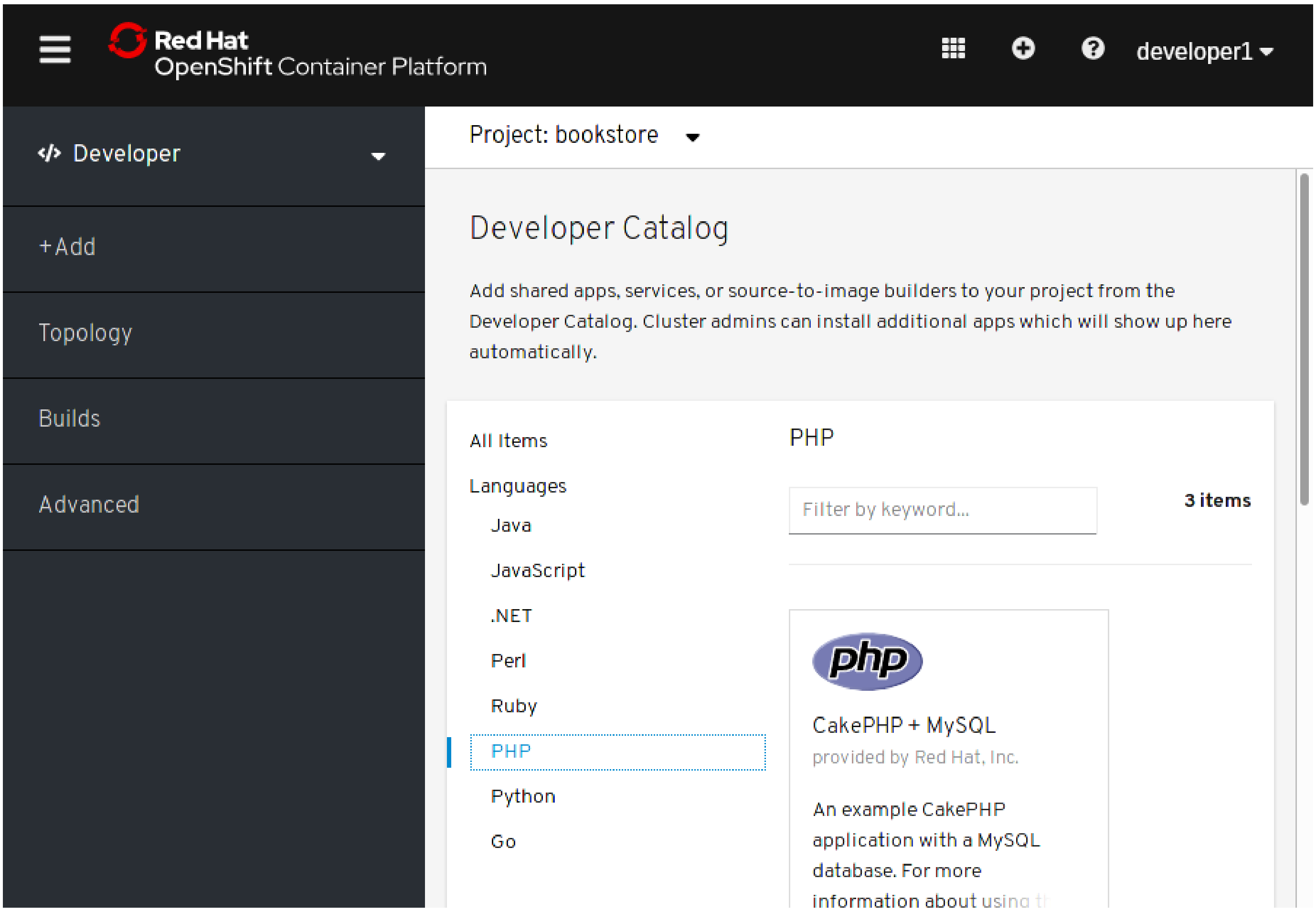


Red Hat
OpenShift Container Platform
developer1
Developer
+Add
Topology
Builds
Advanced
Projects
Events
Search
Project: bookstore
Application: all applications
From Git
Import code from your git repository to be built and deployed
Container Image
Deploy an existing image from an image registry
From Catalog
Browse the catalog to discover, deploy and connect to services
From Dockerfile
Import your Dockerfile from your git repo to be built & deployed

Figure 2.6: Selecting a method to deploy a new application in OpenShift

With the `From Catalog` method, you can list and deploy the available ready to use applications, for example, a MariaDB database. You can also select the language of your application, and provide its source code from a Git repository.

The following screen capture shows some of the available languages.



Project: bookstore

Developer Catalog

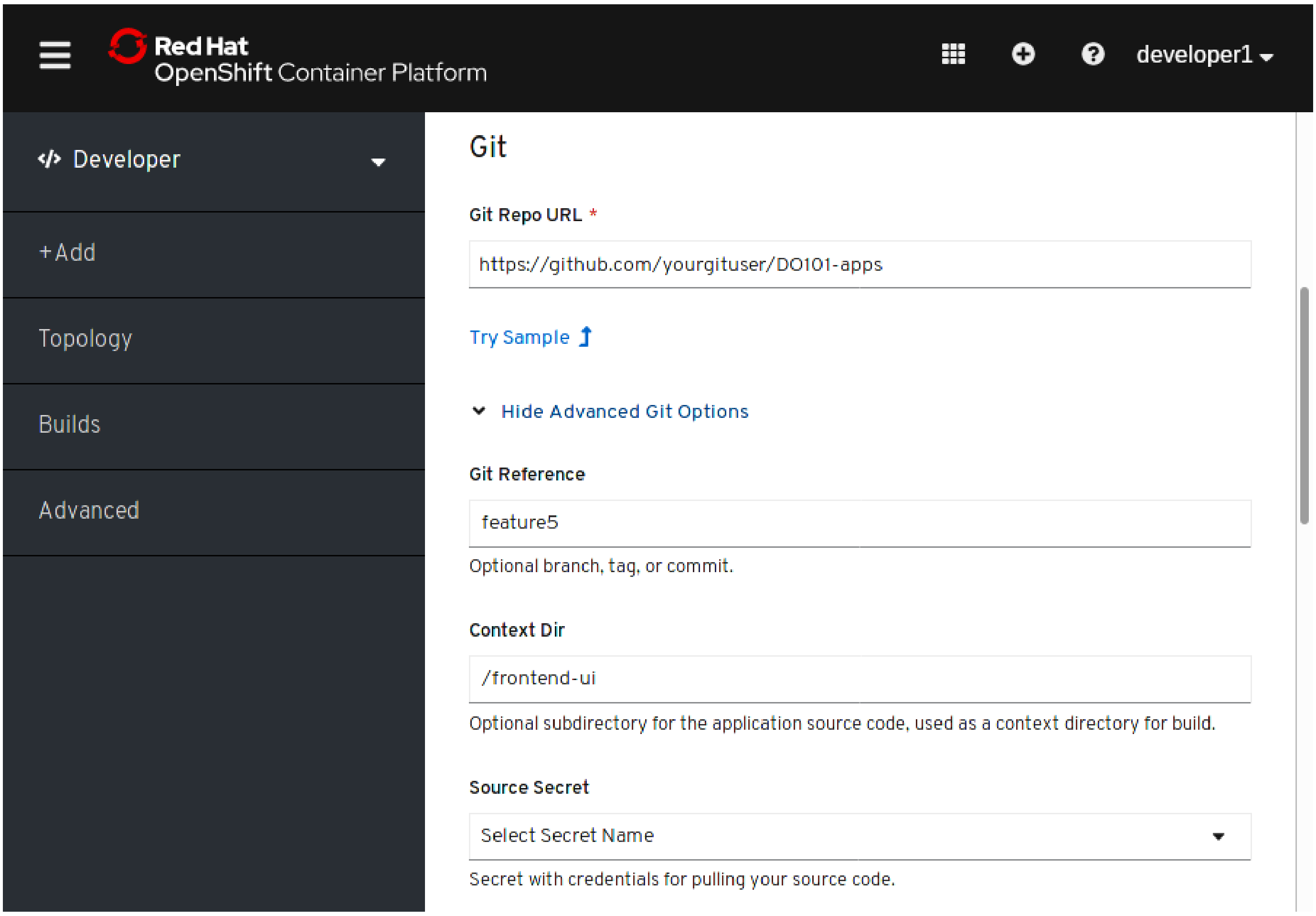
Add shared apps, services, or source-to-image builders to your project from the Developer Catalog. Cluster admins can install additional apps which will show up here automatically.

CakePHP + MySQL
provided by Red Hat, Inc.

An example CakePHP application with a MySQL database. For more information about using t

Figure 2.7: Available languages in the catalog

After selecting the application language from the catalog, OpenShift provides a form to collect the application details.



Git

Git Repo URL *

https://github.com/yourgituser/DO101-apps

Git Reference

feature5

Optional branch, tag, or commit.

Context Dir

/frontend-ui

Optional subdirectory for the application source code, used as a context directory for build.

Source Secret

Select Secret Name

Secret with credentials for pulling your source code.

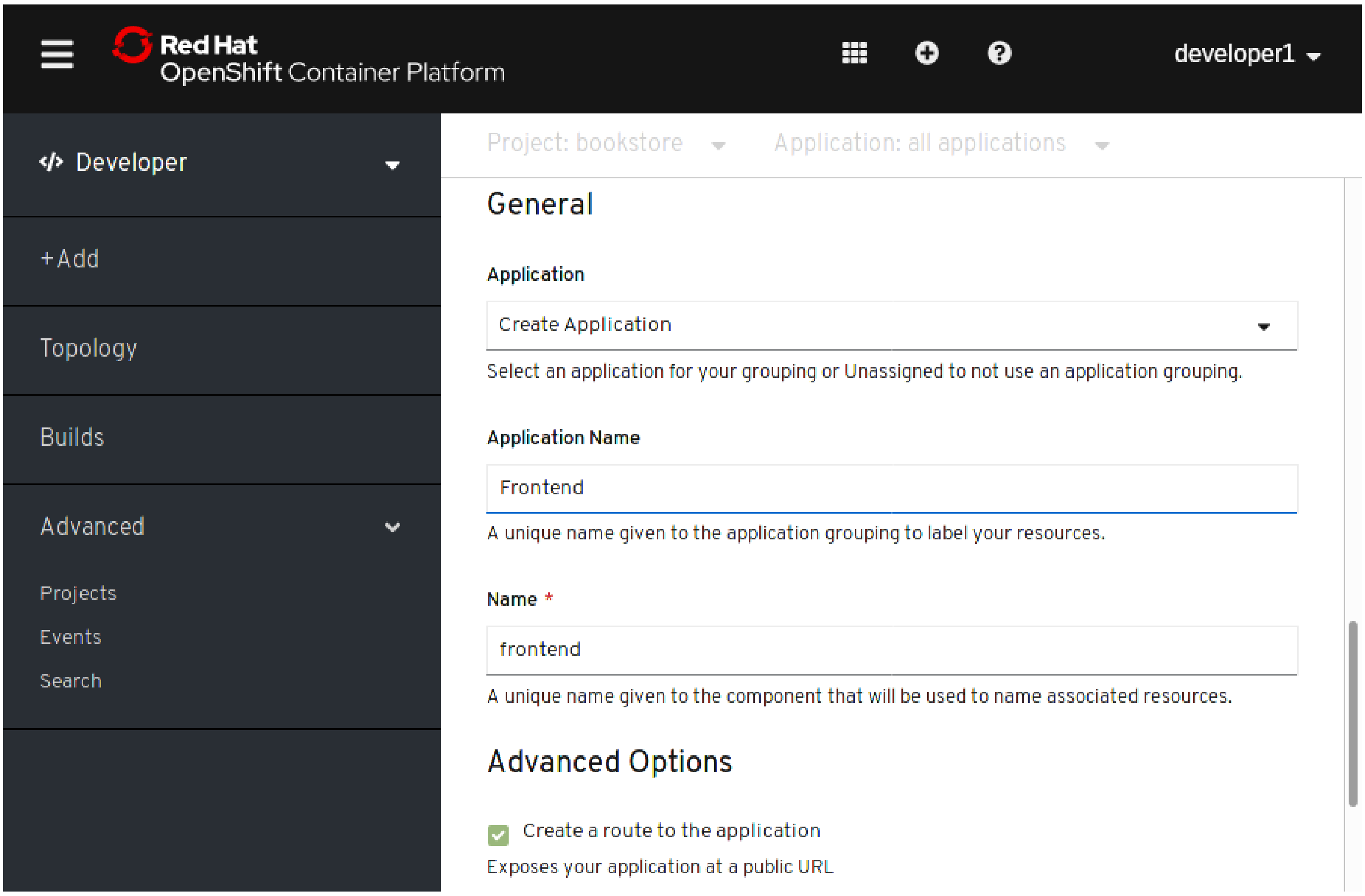
Figure 2.8: Deploying an application — Git details

Use the fields in the first section of the form to provide Git repository details for the application.

If the source code to deploy is not in the Git `master` branch, then provide the branch name in the `Git Reference` field. Use the `Context Dir` field when the application to deploy is in a subdirectory, and not at the root of the Git repository.

In the preceding screen capture, the application is stored in the `frontend-ui` subdirectory, under the https://github.com/yourgituser/DO101-apps Git repository. Also note that OpenShift should fetch the `feature5` branch, rather than the `master` branch.

Under the `General` section of the form, use the `Application Name` field to name the application. OpenShift uses this name to refer to your application in the web console pages. The `Name` field is an internal name that OpenShift uses to identify all the resources it creates during build and deployment. That name is used, for example, for the route resource that associates a public URL to the application.



General

Application

Create Application

Select an application for your grouping or Unassigned to not use an application grouping.

Application Name

Frontend

A unique name given to the application grouping to label your resources.

Name *

frontend

A unique name given to the component that will be used to name associated resources.

Advanced Options

☑ Create a route to the application

Exposes your application at a public URL

Figure 2.9: Naming the application

Reviewing an Application in the Web Console

The `Topology` option provides an overview of the applications in a project. The following screen capture shows two applications running in the `bookstore` project: the `Frontend` PHP application and a database.

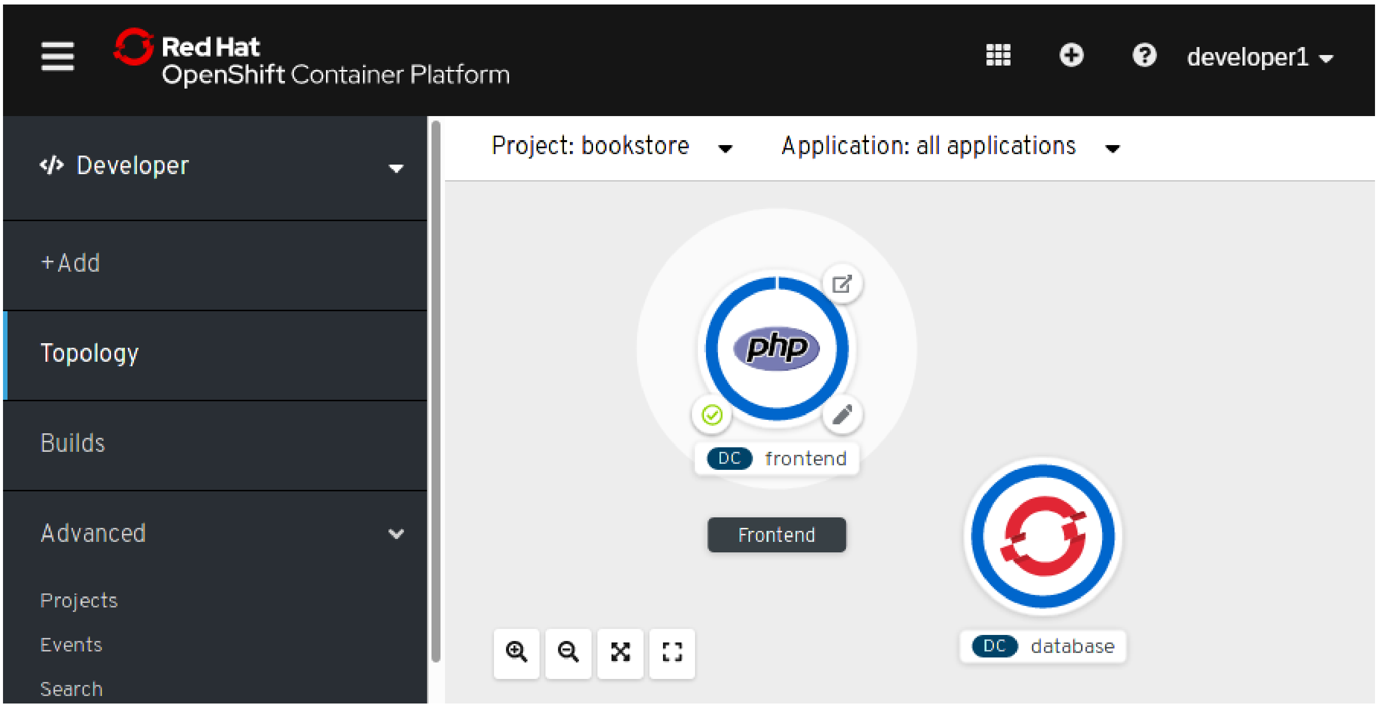
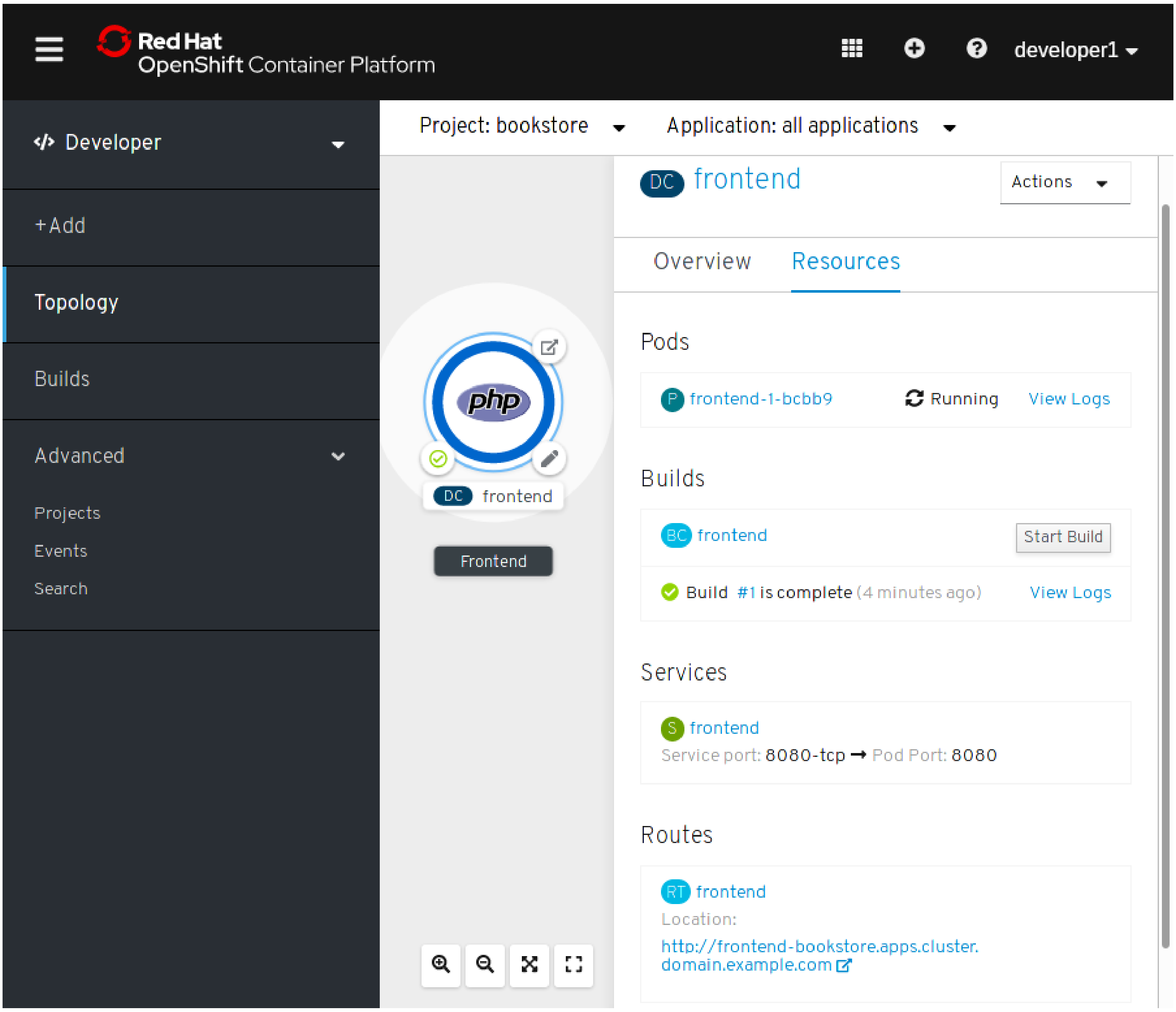


Figure 2.10: Overview of the applications in a project

Click the **application** icon to access application details. The following screen capture shows the resources that OpenShift creates for the application. Notice that one pod is running, the build is complete, and a route provides external access to the application.



DC frontend

Pods

P frontend-1-bcbb9 Running View Logs

Builds

BC frontend Start Build

Build #1 is complete (4 minutes ago) View Logs

Services

S frontend

Service port: 8080-tcp → Pod Port: 8080

Routes

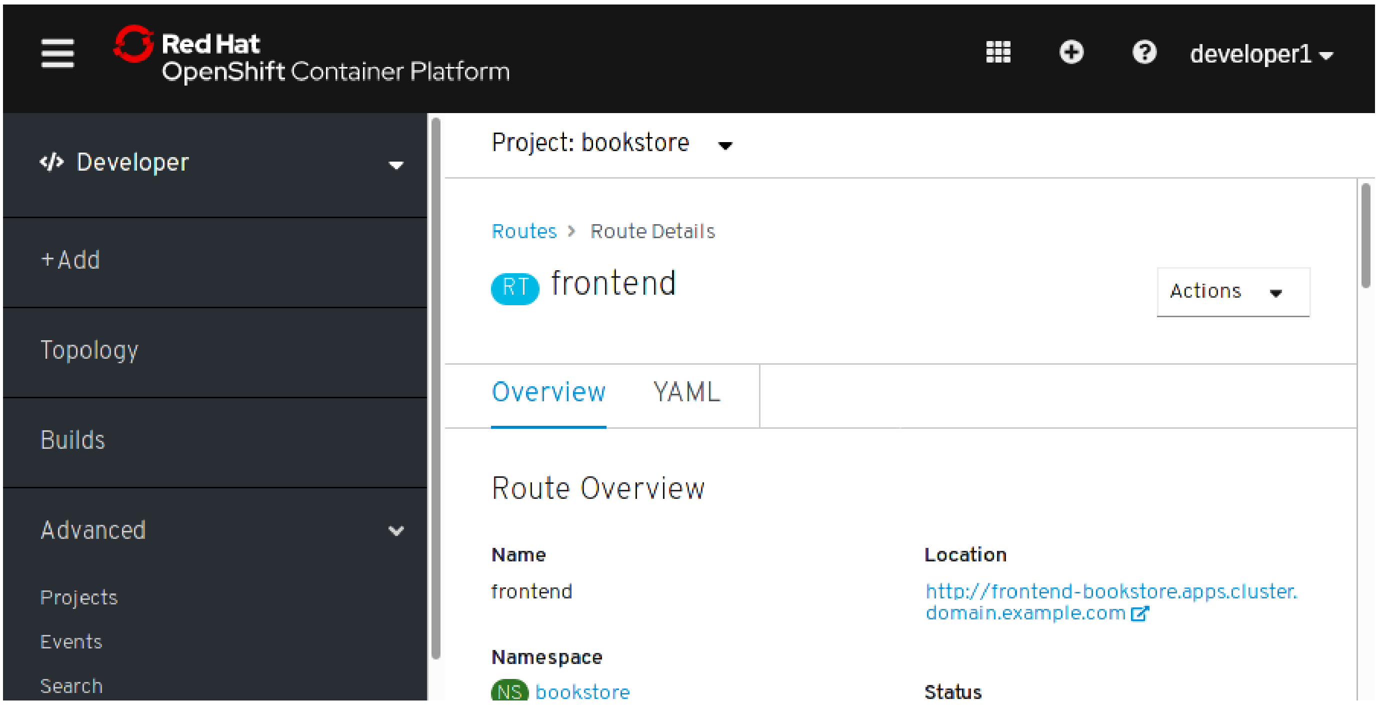
RT frontend

Location:

http://frontend-bookstore.apps.cluster.domain.example.com

Figure 2.11: Resources of an application

Usually, when the web console displays a resource name, it is a link to the details page for that resource. The following screen capture displays the details of the Route resource.



RT frontend

Route Overview

Name

frontend

Location

http://frontend-bookstore.apps.cluster.domain.example.com

Namespace

NS bookstore

Status

Figure 2.12: Application's route details

Editing OpenShift Resources

Most resource details pages from the OpenShift web console provide an **Actions** button that displays a menu.

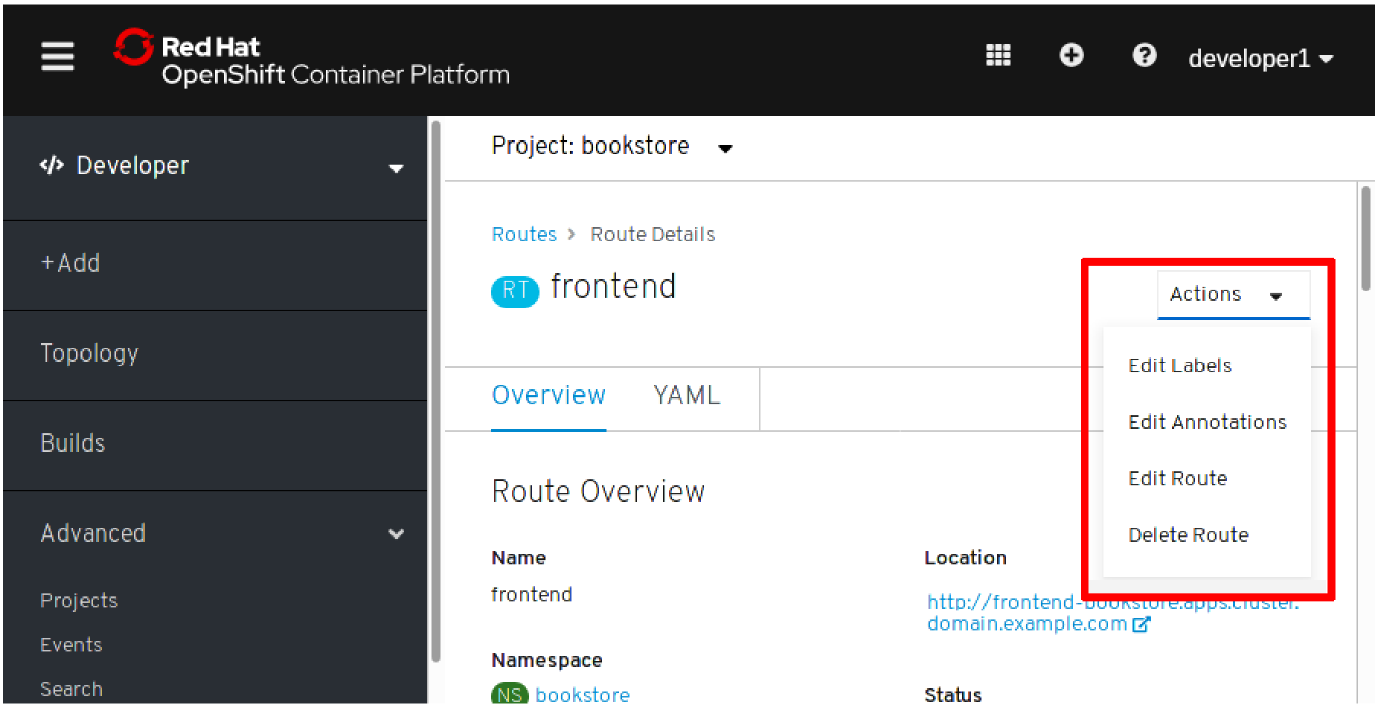


Figure 2.13: Available actions for a route resource

This menu usually provides options to edit and delete the resource.

REFERENCES

For more information, refer to the Product Documentation for Red Hat OpenShift Container Platform at https://access.redhat.com/documentation/en-us/openshift_container_platform (https://access.redhat.com/documentation/en-us/openshift_container_platform)

Introduction to OpenShift (https://www.openshift.com/about/)

➜ NEXT (/ROL/APP/COURSES/DO101-4.2/PAGES/CH02S02)