Guided Exercise

Creating an Application with the Web Console

In this exercise, you will create, build, and deploy an application to an OpenShift cluster using the OpenShift web console.

Outcomes

You should be able to create, build, and deploy an application to an OpenShift cluster using the web console.

Before You Begin

Get the lab files by executing the lab script:

```
[student@workstation ~]$ lab openshift-webconsole start
```

The lab script verifies that the OpenShift cluster is running.

1. Inspect the PHP source code for the sample application and create and push a new branch named **console** to use during this exercise.

   1.1. Enter your local clone of the **DO180-apps** Git repository and checkout the **master** branch of the course's repository to ensure you start this exercise from a known good state:

```
[student@workstation ~]$ cd ~/DO180-apps
    [student@workstation DO180-apps]$ git checkout master
    ...output omitted...
```

   1.2. Create a new branch to save any changes you make during this exercise:

```
[student@workstation DO180-apps]$ git checkout -b console
    Switched to a new branch 'console'
    [student@workstation DO180-apps]$ git push -u origin console
    ...output omitted...
     * [new branch]      console -> console
    Branch console set up to track remote branch console from origin.
```

   1.3. Review the PHP source code of the application, inside the the **php-helloworld** folder.

   Open the **index.php** file in the **/home/student/DO180-apps/php-helloworld** folder:

```
<?php
    print "Hello, World! php version is " . PHP_VERSION . "\n";
    ?>
```

The application implements a simple response which returns the PHP version it is running.

2. Open a web browser and navigate to `https://console-openshift-console.${RHT_OCP4_WILDCARD_DOMAIN}` to access the OpenShift web console. Log in and create a new project named ***youruser*-console**.

   2.1. Load your classroom environment configuration.

   Run the following command to load the environment variables created in the first guided exercise:

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
```

   2.2. Retrieve the value of the wildcard domain specific to your cluster, using the **$RHT_OCP4_WILDCARD_DOMAIN**

```
[student@workstation ~]$ echo $RHT_OCP4_WILDCARD_DOMAIN
apps.cluster.lab.example.com
```

   2.3. Open the Firefox browser and navigate to `https://console-openshift-console.${RHT_OCP4_WILDCARD_DOMAIN}` to access the OpenShift web console. Log in to the OpenShift console using your credentials.

   2.4. Create a new project named ***youruser*-console**. You can type any values you prefer in the other fields.

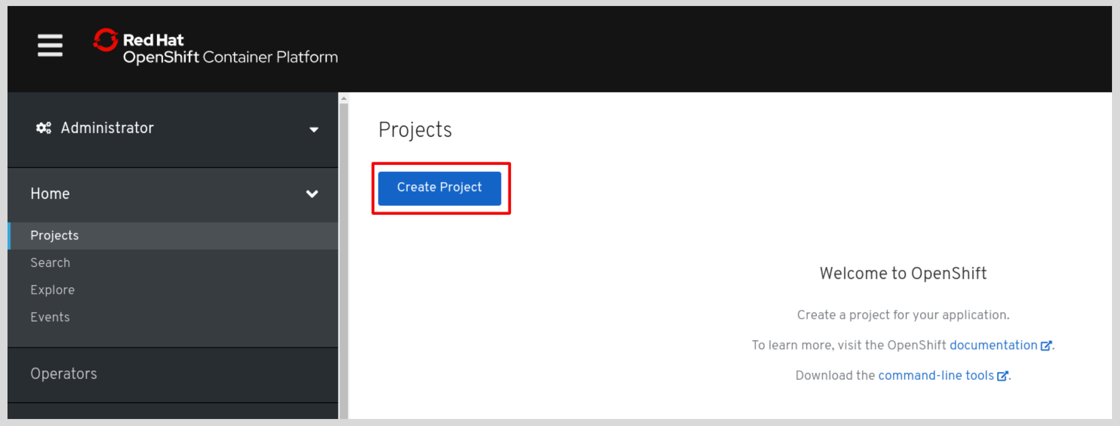
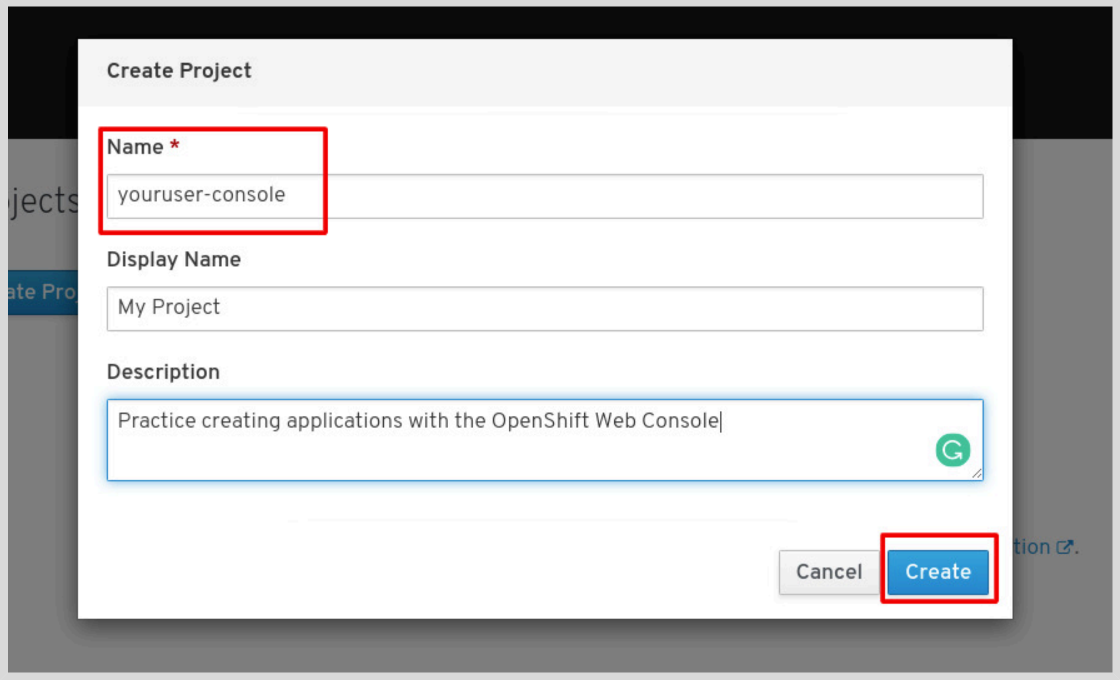

Red Hat
OpenShift Container Platform
Administrator
Home
Projects
Search
Explore
Events
Operators
Projects
Create Project
Welcome to OpenShift
Create a project for your application.
To learn more, visit the OpenShift documentation.
Download the command-line tools.

Figure 6.13: Create a new project - step 1



Create Project

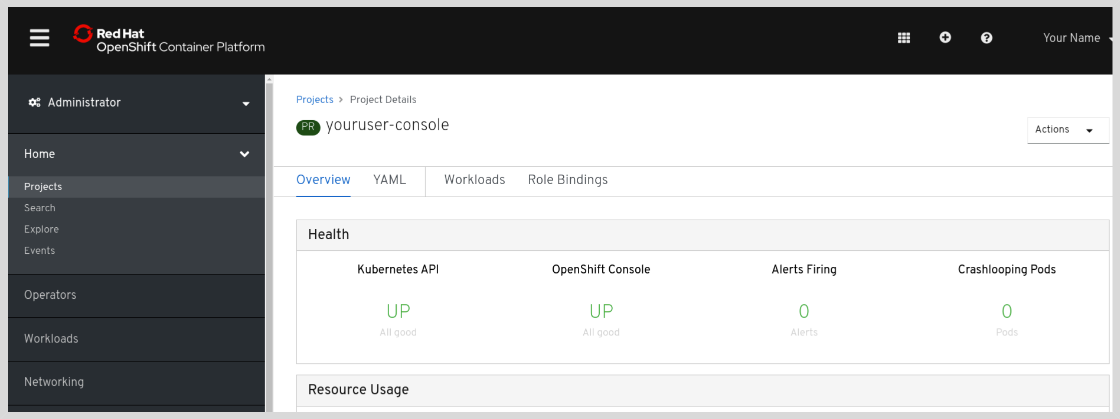
Name *: youruser-console

Display Name: My Project

Description: Practice creating applications with the OpenShift Web Console

Figure 6.14: Create a new project - step 2

2.5. After you have completed the required fields, click **Create** in the **`Create Project`** dialog box to go to the **`Project Status`** page for the ***`youruser`*****`-console`** project:



Projects > Project Details

PR youruser-console

Health

Kubernetes API	OpenShift Console	Alerts Firing	Crashlooping Pods
UP	UP	0	0
All good	All good	Alerts	Pods

Resource Usage

Figure 6.15: Project Status page

3. Create the new php-helloworld application with a PHP template.

3.1. Switch to the **Developer** perspective using the drop-down at the top of the left-hand menu:

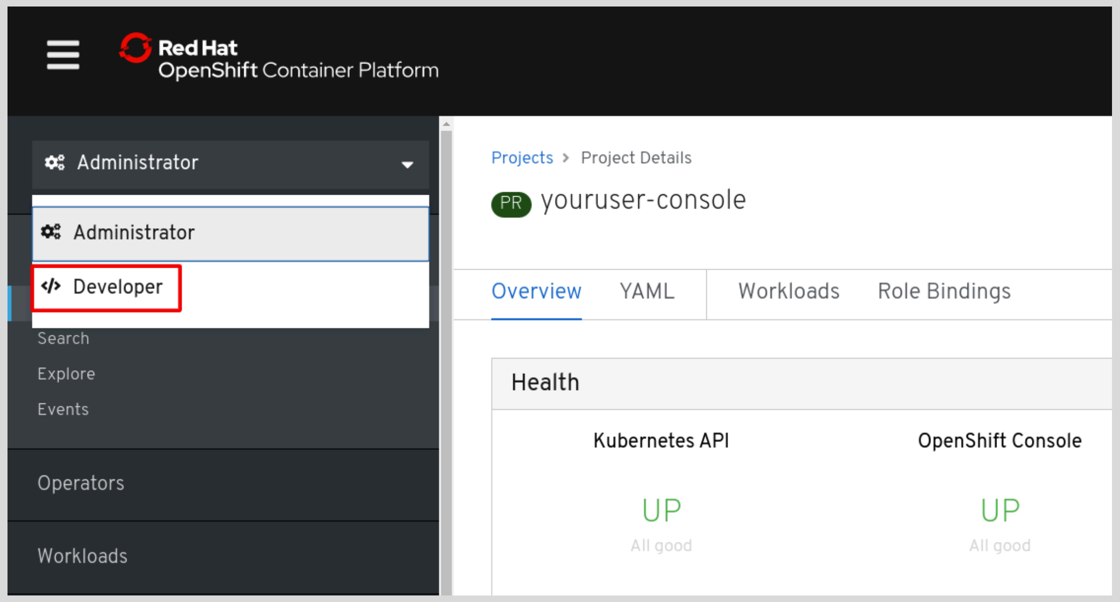
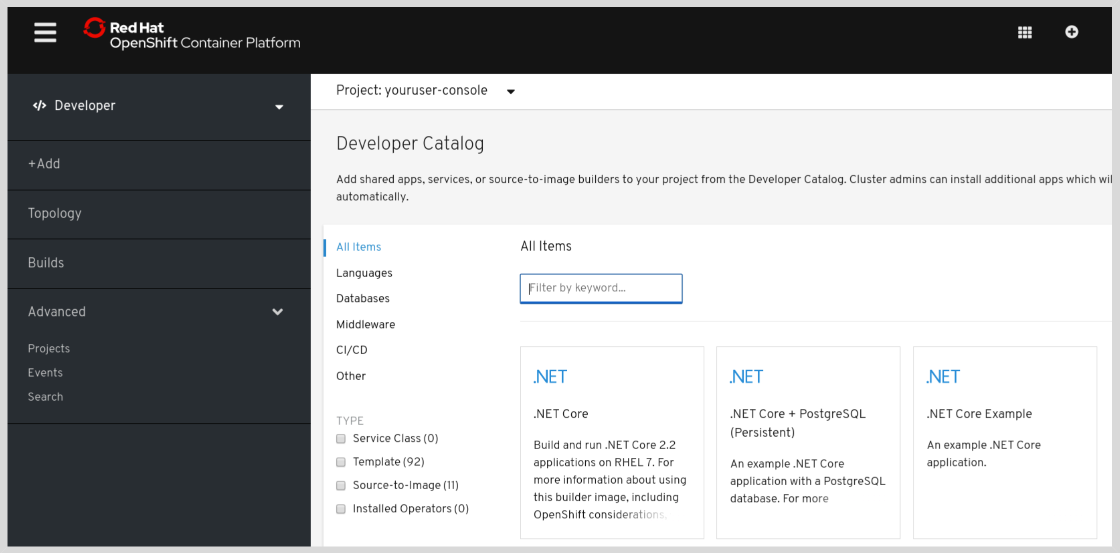


Figure 6.16: Developer perspective drop-down

3.2. Click **From Catalog** to display a list of technology templates.



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

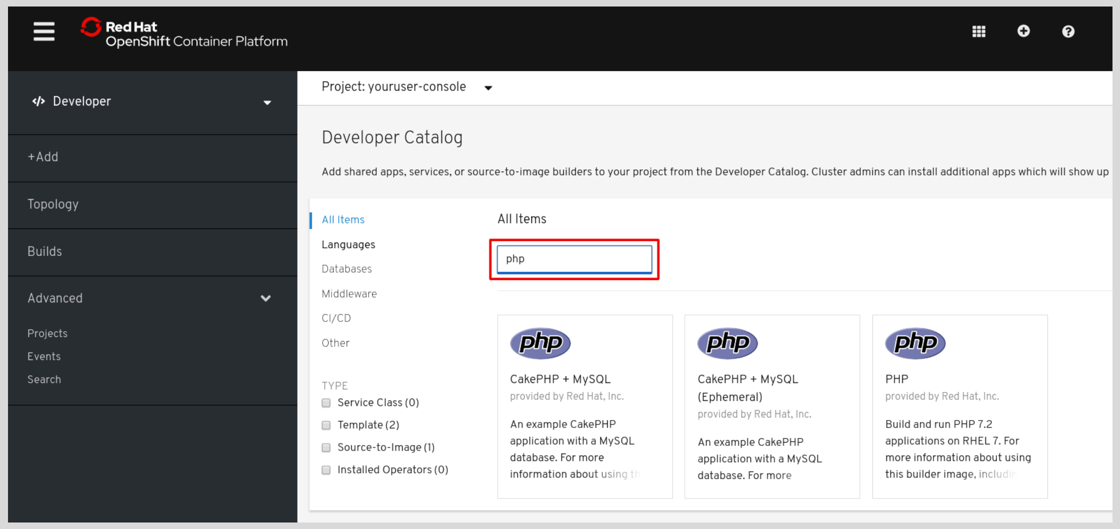
Build and run .NET Core 2.2 applications on RHEL 7. For more information about using this builder image, including OpenShift considerations.

An example .NET Core application with a PostgreSQL database. For more

An example .NET Core application.

Figure 6.17: Developer Catalog page

3.3. Enter **php** in the **Filter by keyword** field.



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

php

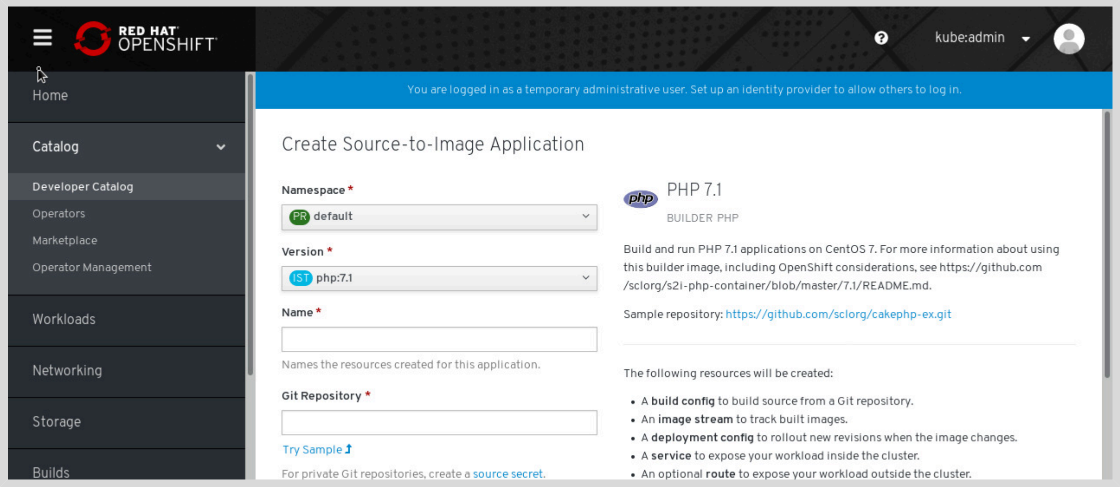
CakePHP + MySQL
provided by Red Hat, Inc.
An example CakePHP application with a MySQL database. For more information about using th

CakePHP + MySQL (Ephemeral)
provided by Red Hat, Inc.
An example CakePHP application with a MySQL database. For more

PHP
provided by Red Hat, Inc.
Build and run PHP 7.2 applications on RHEL 7. For more information about using this builder image, includi

Figure 6.18: Finding PHP-related templates

3.4. After filtering, click the PHP template to display the PHP dialog box. Click **Create Application** to display the **Create Source-to-Image Application** page.



You are logged in as a temporary administrative user. Set up an identity provider to allow others to log in.

Create Source-to-Image Application

Namespace * PR default

Version * IST php:7.1

Name *

Names the resources created for this application.

Git Repository *

For private Git repositories, create a source secret.

PHP 7.1
BUILDER PHP

Build and run PHP 7.1 applications on CentOS 7. For more information about using this builder image, including OpenShift considerations, see https://github.com/sclorg/s2i-php-container/blob/master/7.1/README.md.

Sample repository: https://github.com/sclorg/cakephp-ex.git

The following resources will be created:

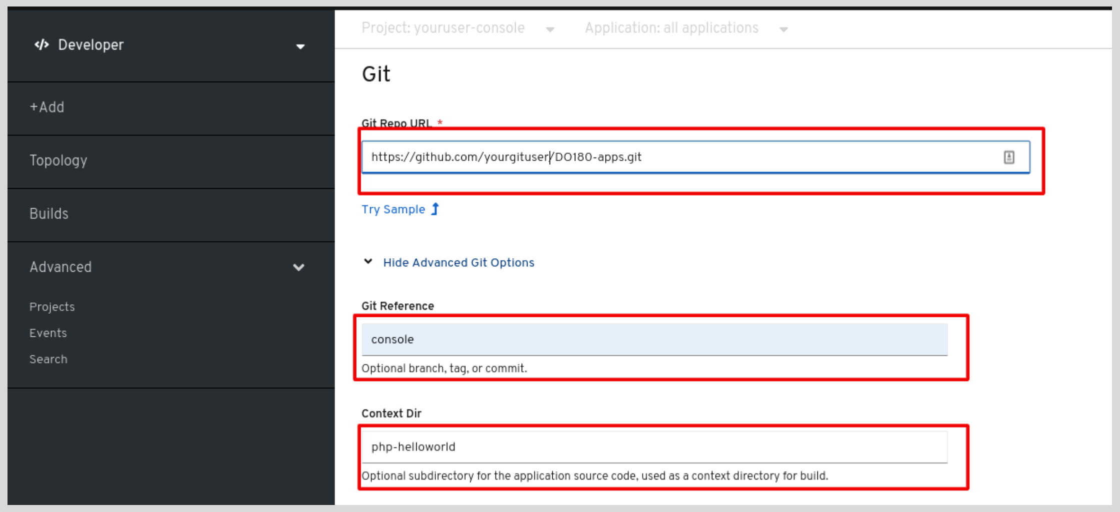
- A **build config** to build source from a Git repository.
- An **image stream** to track built images.
- A **deployment config** to rollout new revisions when the image changes.
- A **service** to expose your workload inside the cluster.
- An optional **route** to expose your workload outside the cluster.

Figure 6.19: Configuring Source-to-Image for a PHP application

3.5. Change the **Version** to PHP version 7.1.

Specify the location of the source code git repository: `https://github.com/`*`yourgituser`*`/DO180-apps`.

Use the **Advanced Git Options** to set the context directory to **`php-helloworld`** and branch **`console`** for this exercise


Developer
+Add
Topology
Builds
Advanced
Projects
Events
Search
Project: youruser-console
Application: all applications
Try Sample
Hide Advanced Git Options

Git

Git Repo URL *

https://github.com/yourgituser/DO180-apps.git

Git Reference

console

Optional branch, tag, or commit.

Context Dir

php-helloworld

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

Figure 6.20: Setting Advanced Git Options for the application

Enter **php-helloworld** for both the application name and the name used for associated resources.

Scroll to the bottom of the page, and select **Create route**. Click **Create** to create the required OpenShift and Kubernetes resources for the application.

You are redirected to the **Topology** page:

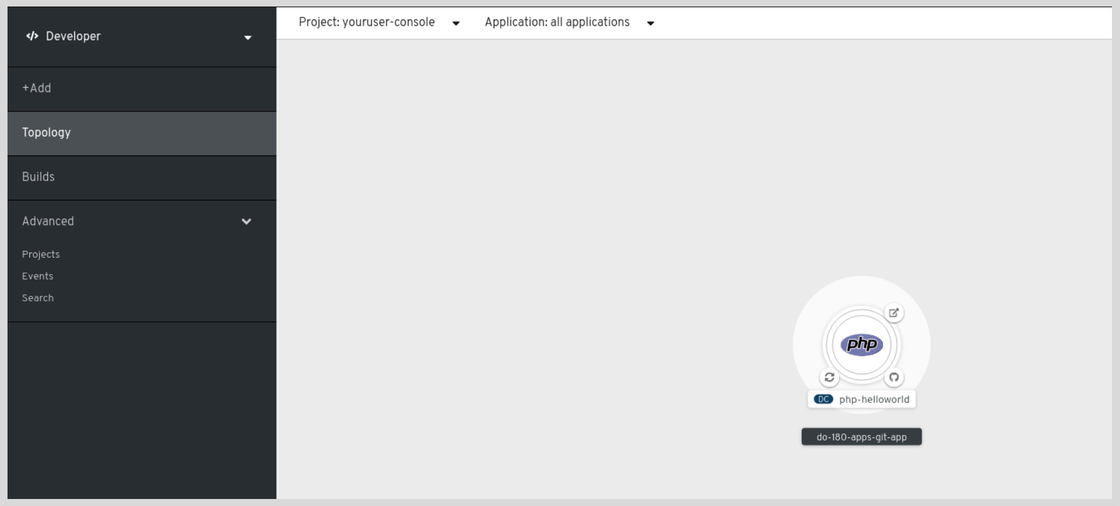


Figure 6.21: Topology page

This page indicates that the php-helloworld application is created. The **DC** annotation to the left of the **php-helloworld** link is an acronym for **Deployment Config**. This link redirects to a page containing information about the application's deployment configuration.

3.6. Switch back to the **Administrator** perspective for the remainder of the exercise:

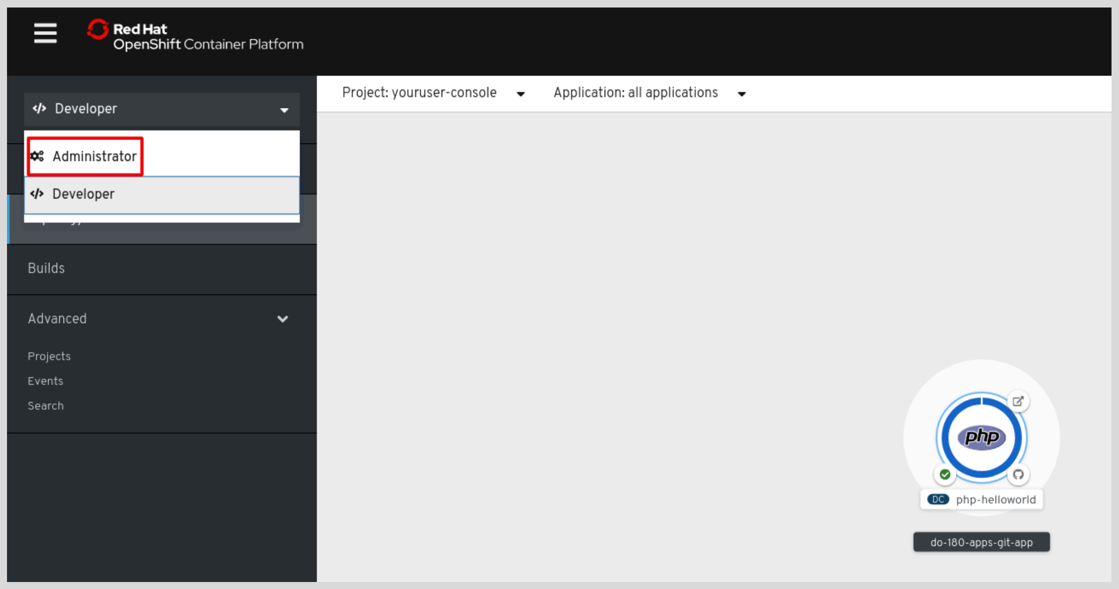
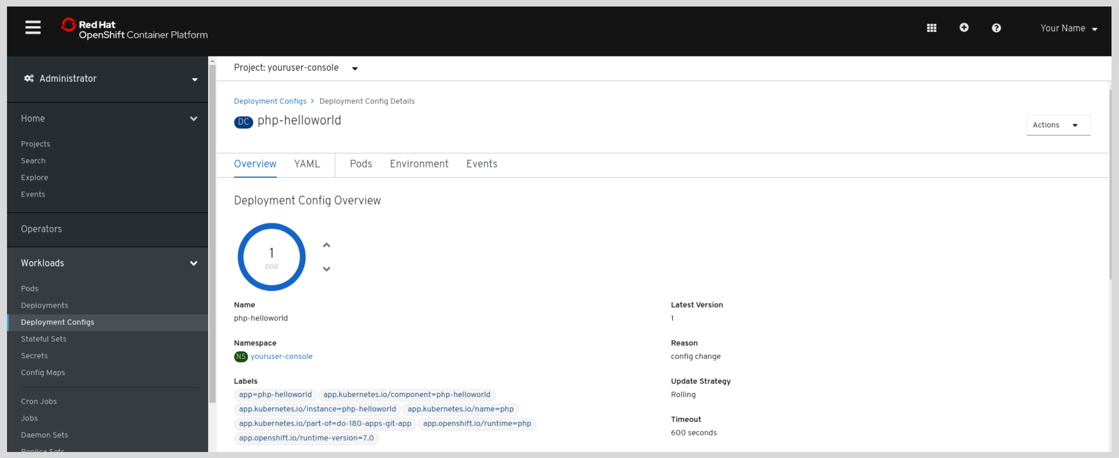


Figure 6.22: Administrator perspective drop-down

4. Use the navigation bar on the left side of the OpenShift web console to locate information for the application's OpenShift and Kubernetes resources:

   - DeploymentConfig
   - BuildConfig
   - Build Logs
   - Service
   - Route

   4.1. Examine the deployment configuration. In the navigation bar, click **Workloads** to reveal more menu choices. Click **Deployment Configs** to display a list of deployment configurations for the ***youruser*-console** project. Click the **php-helloworld** link to display deployment configuration information.



Deployment Config Overview

Name
php-helloworld

Namespace
NS youruser-console

Labels
app=php-helloworld
app.kubernetes.io/component=php-helloworld
app.kubernetes.io/instance=php-helloworld
app.kubernetes.io/name=php
app.kubernetes.io/part-of=do-180-apps-git-app
app.openshift.io/runtime=php
app.openshift.io/runtime-version=7.0

Latest Version
1

Reason
config change

Update Strategy
Rolling

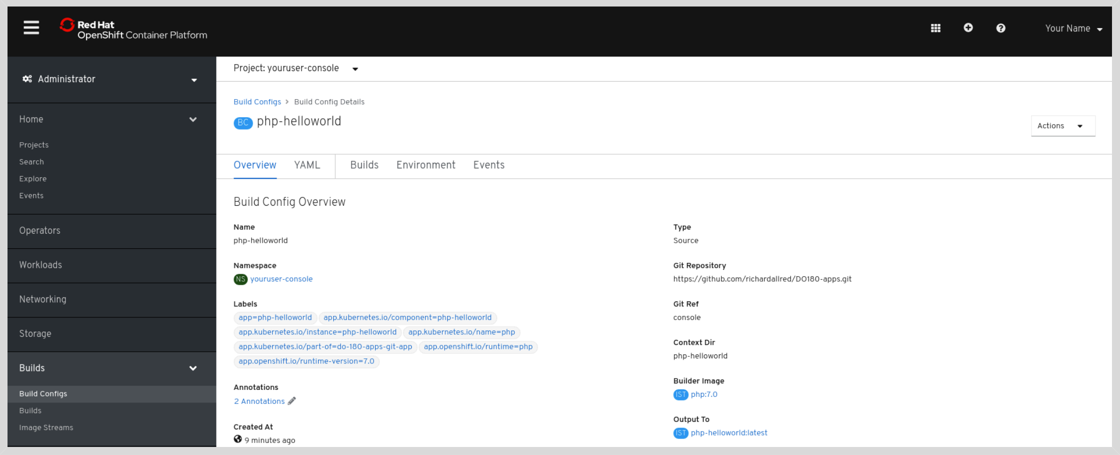
Timeout
600 seconds

Figure 6.23: Application deployment configuration overview page

Explore the available information from the **Overview** tab. The build may still be running when you reach this page, so the DC might not have a value of **`1 pod`**, yet.

If you click the up and down arrow icons next to the doughnut chart that indicates the number of pods, you can scale the application up and down horizontally.

4.2. Examine the build configuration. In the navigation bar, click **Builds** to reveal more menu choices. Click **Build Configs** to display a list of build configurations for the ***`youruser`*****`-console`** project. Click the **`php-helloworld`** link to display the build configuration for the application.



Project: youruser-console

Build Configs > Build Config Details

BC php-helloworld

Build Config Overview

Name
php-helloworld

Namespace
NS youruser-console

Labels
app=php-helloworld
app.kubernetes.io/component=php-helloworld
app.kubernetes.io/instance=php-helloworld
app.kubernetes.io/name=php
app.kubernetes.io/part-of=do-180-apps-git-app
app.openshift.io/runtime=php
app.openshift.io/runtime-version=7.0

Annotations
2 Annotations

Created At
9 minutes ago

Type
Source

Git Repository
https://github.com/richardallred/DO180-apps.git

Git Ref
console

Context Dir
php-helloworld

Builder Image
IST php:7.0

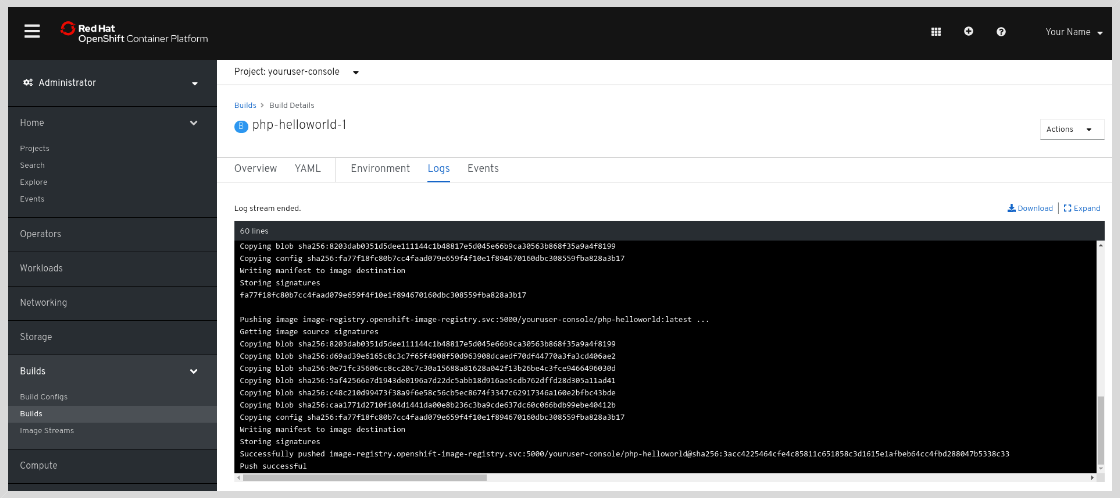
Output To
IST php-helloworld:latest

Figure 6.24: Application build configuration overview page

Explore the available information from the **Overview** tab. The **YAML** tab allows you to view and edit the build configuration as a YAML file. The **Builds** tab provides an historical list of builds, along with a link to more information for each build. The **Environment** tab allows you to view and edit environment variables for the application's build environment. The **Events** tab displays a list of build related events and metadata.

4.3. Examine the logs for the Source-to-Image build of the application. In the **Builds** menu, click **Builds** to display a list of recent builds for the ***`youruser`*****`-console`** project.

Click the **`php-helloworld-1`** link to access information for the first build of the **`php-helloworld`** application:



Log stream ended.

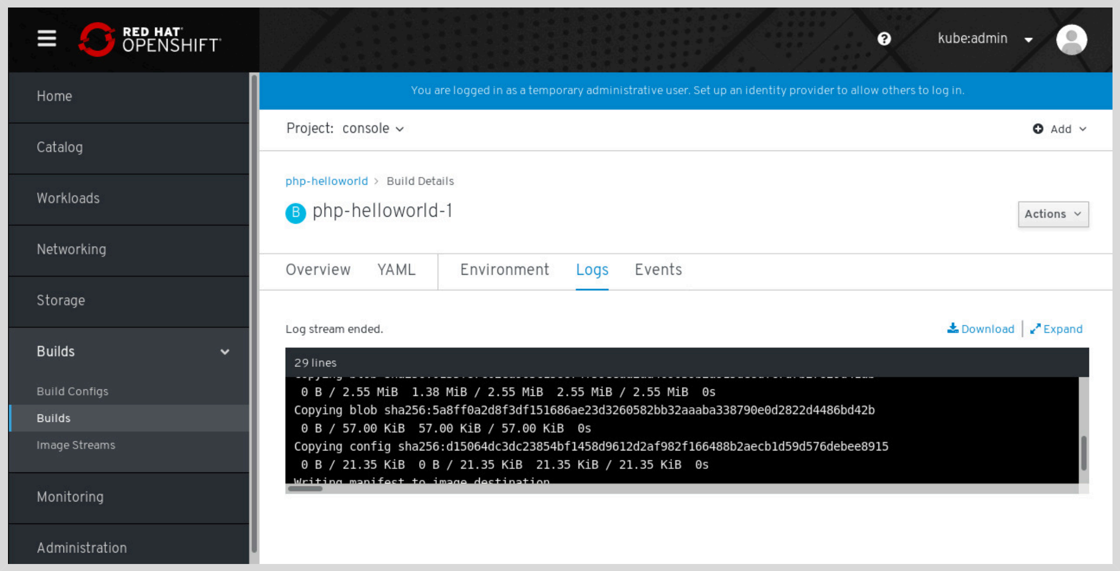
60 lines

```
Copying blob sha256:8203dab0351d5dee111144c1b48817e5d045e66b9ca30563b868f35a9a4f8199
Copying config sha256:fa77f18fc80b7cc4faad079e659f4f10e1f894670160dbc308559fba828a3b17
Writing manifest to image destination
Storing signatures
fa77f18fc80b7cc4faad079e659f4f10e1f894670160dbc308559fba828a3b17

Pushing image image-registry.openshift-image-registry.svc:5000/youruser-console/php-helloworld:latest ...
Getting image source signatures
Copying blob sha256:8203dab0351d5dee111144c1b48817e5d045e66b9ca30563b868f35a9a4f8199
Copying blob sha256:d69ad39e6165c8c3c7f65f4908f50d963908dcaedf70df44770a3fa3cd406ae2
Copying blob sha256:0e71fc35606cc8cc20c7c30a15688a81628a042f13b26be4c3fce9466496030d
Copying blob sha256:5af42566e7d1943de0196a7d22dc5abb18d916ae5cdb762dffd28d305a11ad41
Copying blob sha256:c48c210d99473f38a9f6e58c56cb5ec8674f3347c62917346a160e2bfbc43bde
Copying blob sha256:caa1771d2710f104d1441da00e8b236c3ba9cde637dc60c066bdb99ebe40412b
Copying config sha256:fa77f18fc80b7cc4faad079e659f4f10e1f894670160dbc308559fba828a3b17
Writing manifest to image destination
Storing signatures
Successfully pushed image-registry.openshift-image-registry.svc:5000/youruser-console/php-helloworld@sha256:3acc4225464cfe4c85811c651858c3d1615e1afbeb64cc4fbd288047b5338c33
Push successful
```

Figure 6.25: An application build overview page

Explore the available information from the **Overview** tab. Next, click the **Logs** tab. A scrollable text box contains output from the build process:



You are logged in as a temporary administrative user. Set up an identity provider to allow others to log in.

Log stream ended.

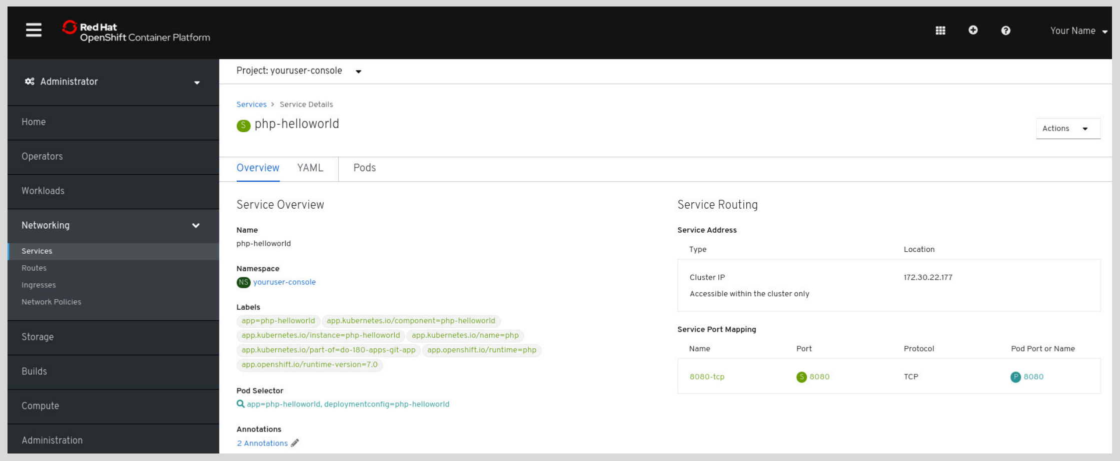
29 lines

```
 0 B / 2.55 MiB  1.38 MiB / 2.55 MiB  2.55 MiB / 2.55 MiB  0s
Copying blob sha256:5a8ff0a2d8f3df151686ae23d3260582bb32aaaba338790e0d2822d4486bd42b
 0 B / 57.00 KiB  57.00 KiB / 57.00 KiB  0s
Copying config sha256:d15064dc3dc23854bf1458d9612d2af982f166488b2aecb1d59d576debee8915
 0 B / 21.35 KiB  0 B / 21.35 KiB  21.35 KiB / 21.35 KiB  0s
Writing manifest to image destination
```

Figure 6.26: Logs for an application build

When Podman builds a container image, similar output is observed compared with the output shown in the browser.

4.4. Locate information for the **php-helloworld** application's service. In the navigation bar, click **Networking** to reveal more menu choices. Click **Services** to display a list of services for the ***youruser*-console** project. Click the **php-helloworld** link to display the information associated with the application's service:



Project: youruser-console

Services > Service Details

S php-helloworld

Service Overview

Name
php-helloworld

Namespace
NS youruser-console

Labels
app=php-helloworld app.kubernetes.io/component=php-helloworld app.kubernetes.io/instance=php-helloworld app.kubernetes.io/name=php app.kubernetes.io/part-of=do-180-apps-git-app app.openshift.io/runtime=php app.openshift.io/runtime-version=7.0

Pod Selector
app=php-helloworld, deploymentconfig=php-helloworld

Annotations
2 Annotations

Service Routing

Service Address

Type	Location
Cluster IP Accessible within the cluster only	172.30.22.177

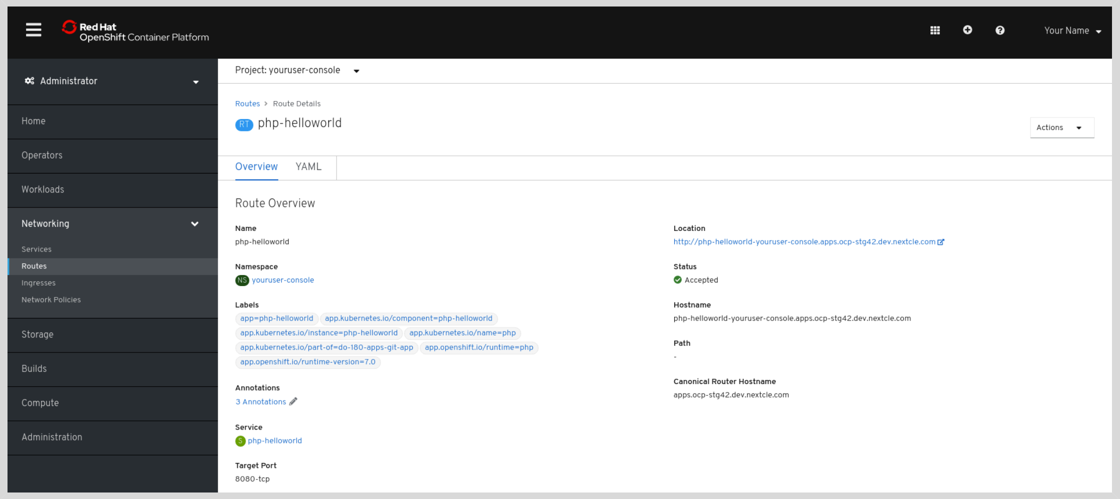
Service Port Mapping

Name	Port	Protocol	Pod Port or Name
8080-tcp	S 8080	TCP	P 8080

Figure 6.27: Service overview page

Explore the available information from the **Overview** tab. The **YAML** tab allows you to view and edit the service configuration, as a YAML file. The **Pods** tab displays the current list of pods that provide the application service.

4.5. Locate external route information for the application. On the navigation bar, click **Networking** → **Routes** to display a list of configured routes for the ***youruser*-console** project. Click the **php-helloworld** link to display information associated with the application's route:



Project: youruser-console

Routes > Route Details

RT php-helloworld

Route Overview

Name
php-helloworld

Namespace
NS youruser-console

Labels
app=php-helloworld app.kubernetes.io/component=php-helloworld app.kubernetes.io/instance=php-helloworld app.kubernetes.io/name=php app.kubernetes.io/part-of=do-180-apps-git-app app.openshift.io/runtime=php app.openshift.io/runtime-version=7.0

Annotations
3 Annotations

Service
S php-helloworld

Target Port
8080-tcp

Location
http://php-helloworld-youruser-console.apps.ocp-stg42.dev.nextcle.com

Status
Accepted

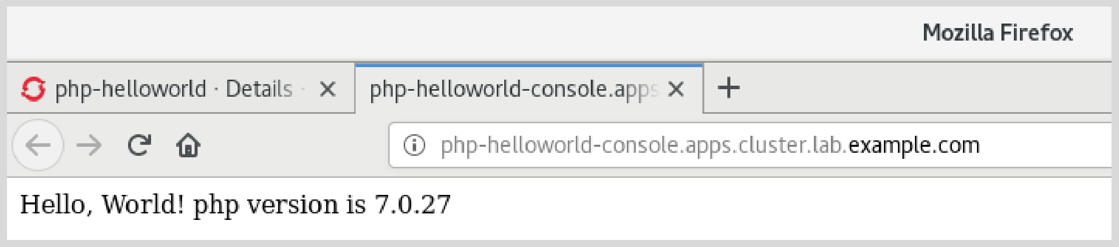
Hostname
php-helloworld-youruser-console.apps.ocp-stg42.dev.nextcle.com

Path
-

Canonical Router Hostname
apps.ocp-stg42.dev.nextcle.com

Figure 6.28: Route overview page

Explore the available information from the **Overview** tab. The **LOCATION** field provides a link to the external route for the application; `http://php-helloworld-${RHT_OCP4_DEV_USER}-console.${RHT_OCP4_WILDCARD_DOMAIN}`. Click the link to access the application in a new tab:


Mozilla Firefox
php-helloworld · Details
php-helloworld-console.apps
php-helloworld-console.apps.cluster.lab.example.com

Hello, World! php version is 7.0.27

Figure 6.29: Initial PHP application results

5. Modify the application code, commit the change, push the code to the remote Git repository, and trigger a new application build.

    5.1. Enter the source code directory:

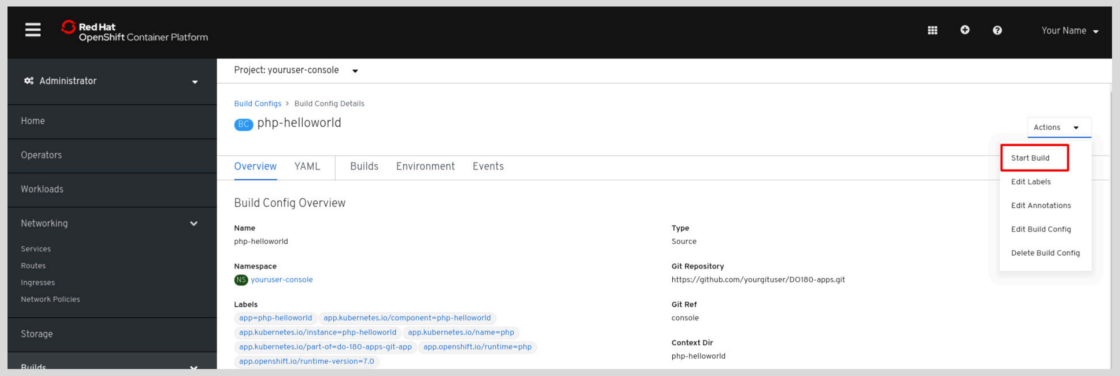
```
[student@workstation DO180-apps]$ cd ~/DO180-apps/php-helloworld
```

    5.2. Add the second print line statement in the **index.php** page to read "A change is in the air!" and save the file. Add the change to the Git index, commit the change, and push the changes to the remote Git repository.

```
[student@workstation php-helloworld]$ vim index.php
[student@workstation php-helloworld]$ cat index.php
<?php
print "Hello, World! php version is " . PHP_VERSION . "\n";
print "A change is in the air!\n";
?>
[student@workstation php-helloworld]$ git add index.php
[student@workstation php-helloworld]$ git commit -m 'updated app'
[console d198fb5] updated app
...output omitted...
 1 file changed, 1 insertion(+), 1 deletion(-)
[student@workstation php-helloworld]$ git push origin console
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 409 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
...output omitted...
```

    5.3. Trigger an application build manually from the web console.

    On the navigation bar, click **Builds** → **Build Configs** and then click the **php-helloworld** link to access the Build Config Overview page. From the **Actions** menu in the upper right of the screen, click **Start Build**:



Project: youruser-console

Build Configs > Build Config Details

BC php-helloworld

Build Config Overview

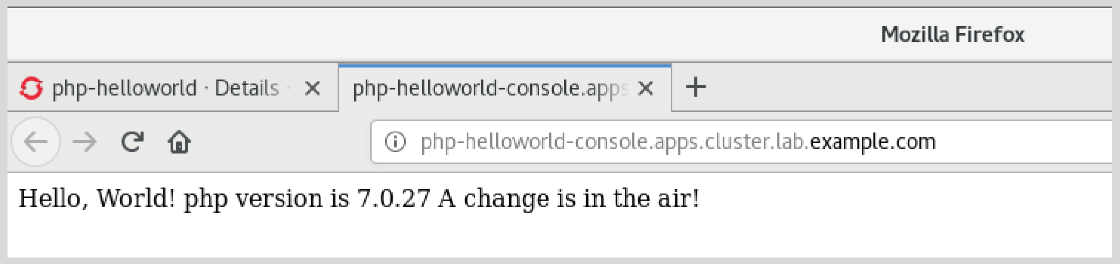
Name php-helloworld	**Type** Source
Namespace NS youruser-console	**Git Repository** https://github.com/yourgituser/DO180-apps.git
Labels app=php-helloworld app.kubernetes.io/component=php-helloworld app.kubernetes.io/instance=php-helloworld app.kubernetes.io/name=php app.kubernetes.io/part-of=do-180-apps-git-app app.openshift.io/runtime=php app.openshift.io/runtime-version=7.0	**Git Ref** console **Context Dir** php-helloworld

Figure 6.30: Start an application build

You are redirected to a Build Overview page for the new build. Click the **Logs** tab to monitor progress of the build. The last line of a successful build contains **`Push successful`**.

When the build completes, the deploy starts. Go to the **Workloads** → **Pods** section, and wait for the new pod is deployed and running.

5.4. Reload the `http://php-helloworld-${RHT_OCP4_DEV_USER}-console.${RHT_OCP4_WILDCARD_DOMAIN}` URL in the browser. The application response corresponds to the updated source code:

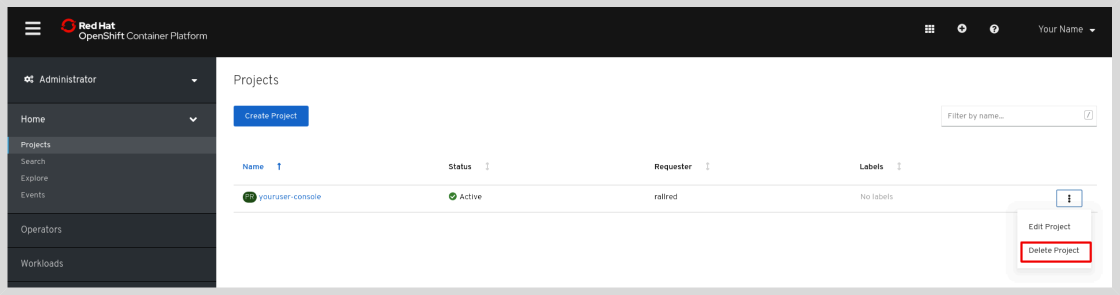


php-helloworld-console.apps.cluster.lab.example.com

```
Hello, World! php version is 7.0.27 A change is in the air!
```

Figure 6.31: Updated web application output

6. Delete the project. On the navigation bar, click **Home** → **Projects**. Click the icon at the right side of the row containing an entry for the ***`youruser`*****`-console`** project. Click **Delete Project** from the menu that appears.



Name	Status	Requester	Labels
PR youruser-console	Active	rallred	No labels

Figure 6.32: Delete a project

Enter ***`youruser`*****`-console`** in the confirmation dialog box, and click **Delete**.

Finish

On **`workstation`**, run the **`lab openshift-webconsole finish`** script to complete this lab.

```
[student@workstation php-helloworld]$ lab openshift-webconsole finish
```

This concludes the guided exercise.