Implementing an OpenShift (OCP) Source to Image (S2I) Pipeline

April 10, 2020

Contents

[Git Repository 1](#_Toc37423280)

[Repository URL 1](#_Toc37423281)

[PHP Source Code 2](#_Toc37423282)

[Console Output 2](#_Toc37423283)

[Create a Project 2](#_Toc37423284)

[Create a New OCP Application 3](#_Toc37423285)

[Check OCP Status 4](#_Toc37423286)

[Check OCP Status with Suggest Switch 5](#_Toc37423287)

[Check OCP Pods 5](#_Toc37423288)

[Check OCP Pods Again 6](#_Toc37423289)

[Check OCP Pods Once More 6](#_Toc37423290)

[Get Service Name 6](#_Toc37423291)

[Expose Service 7](#_Toc37423292)

[Get the Route to the Application Pod 7](#_Toc37423293)

[Use Curl to Access the Application 7](#_Toc37423294)

[User Browser to Access Application 8](#_Toc37423295)

[Start Another Build 8](#_Toc37423296)

[Check Pods 9](#_Toc37423297)

[Check Status 9](#_Toc37423298)

[Check Status Again 10](#_Toc37423299)

[Check Status Once More 10](#_Toc37423300)

[Check Pods 11](#_Toc37423301)

[Get the Service Name 11](#_Toc37423302)

[Get Routes 12](#_Toc37423303)

[Use Curl to Access Application 12](#_Toc37423304)

[Start Another Build 13](#_Toc37423305)

[Check Pods 13](#_Toc37423306)

[Check Pods Again 13](#_Toc37423307)

[Check Pods Once More 14](#_Toc37423308)

[Check Pods Again 14](#_Toc37423309)

[Check Pods Again 14](#_Toc37423310)

[Use Curl to Access Application 15](#_Toc37423311)

[Use Browser to Access Application 15](#_Toc37423312)

[OCP Artifacts that are Created 16](#_Toc37423313)

[Deployment Config 16](#_Toc37423314)

[Config Maps 17](#_Toc37423315)

[Replication Controllers 18](#_Toc37423316)

[Services 19](#_Toc37423317)

[Routes 19](#_Toc37423318)

[Build Configs 20](#_Toc37423319)

[Builds 20](#_Toc37423320)

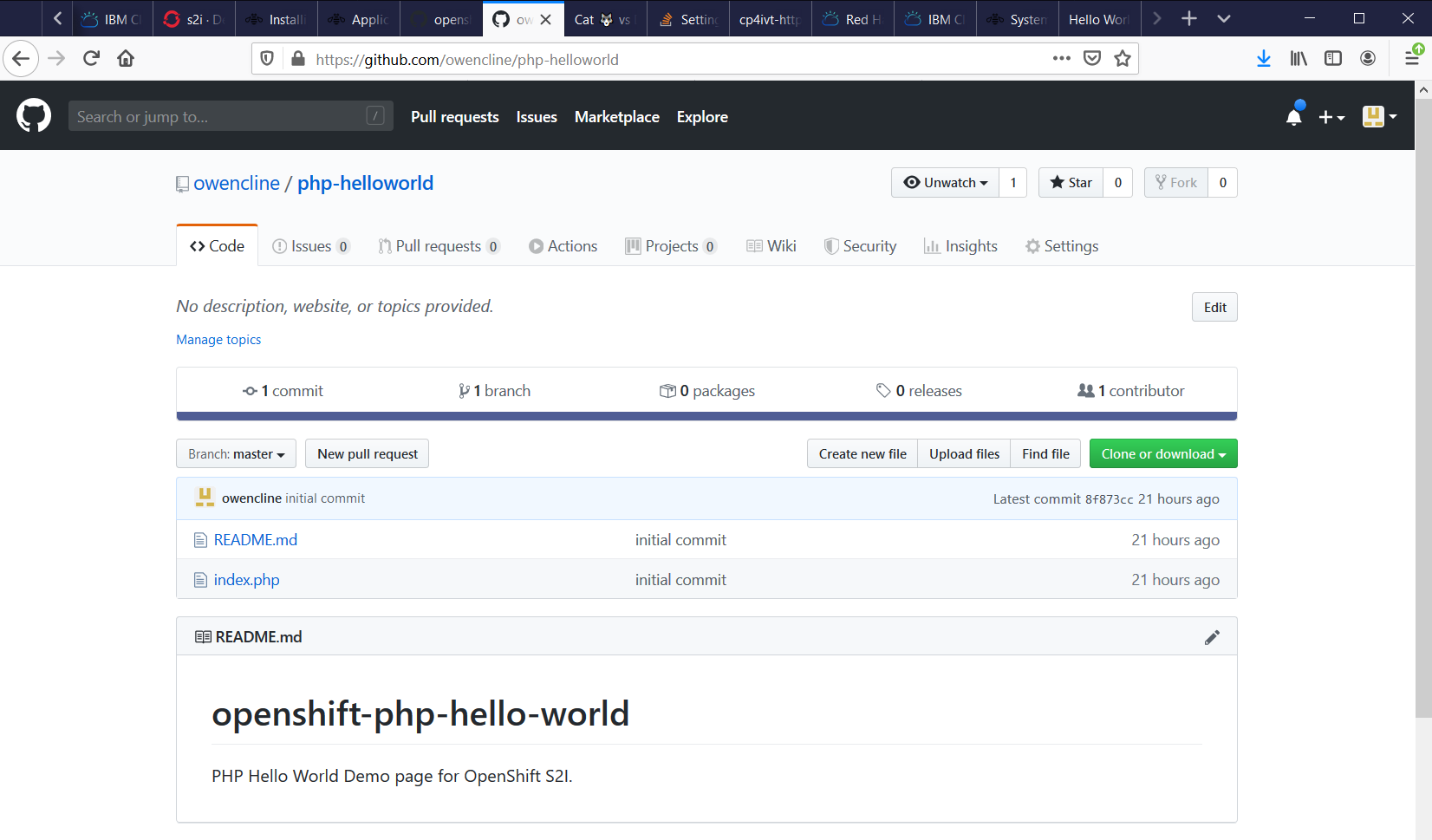
[Image Streams 21](#_Toc37423321)

# Git Repository

You should create a public Git repository and push your source code there.

## Repository URL

https://github.com/owencline/php-helloworld



## PHP Source Code



# Console Output

You should go to a console and log into your OCP cluster.

## Create a Project

root@jumpserver01:~/dockerfile-create# oc project s2i

Now using project "s2i" on server "https://api.orc7.owencline.com:6443".

## Create a New OCP Application

root@jumpserver01:~/dockerfile-create# oc new-app --name php-helloworld https://github.com/owencline/php-helloworld.git

--> Found image 7216d5f (6 weeks old) in image stream "openshift/php" under tag "7.2" for "php"

Apache 2.4 with PHP 7.2

-----------------------

PHP 7.2 available as container is a base platform for building and running various PHP 7.2 applications and frameworks. PHP is an HTML-embedded scripting language. PHP attempts to make it easy for developers to write dynamically generated web pages. PHP also offers built-in database integration for several commercial and non-commercial database management systems, so writing a database-enabled webpage with PHP is fairly simple. The most common use of PHP coding is probably as a replacement for CGI scripts.

Tags: builder, php, php72, rh-php72

\* The source repository appears to match: php

\* A source build using source code from https://github.com/owencline/php-helloworld.git will be created

\* The resulting image will be pushed to image stream tag "php-helloworld:latest"

\* Use 'oc start-build' to trigger a new build

\* This image will be deployed in deployment config "php-helloworld"

\* Ports 8080/tcp, 8443/tcp will be load balanced by service "php-helloworld"

\* Other containers can access this service through the hostname "php-helloworld"

--> Creating resources ...

imagestream.image.openshift.io "php-helloworld" created

buildconfig.build.openshift.io "php-helloworld" created

deploymentconfig.apps.openshift.io "php-helloworld" created

service "php-helloworld" created

--> Success

Build scheduled, use 'oc logs -f bc/php-helloworld' to track its progress.

Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:

'oc expose svc/php-helloworld'

Run 'oc status' to view your app.

## Check OCP Status

root@jumpserver01:~/dockerfile-create# oc status

In project s2i on server https://api.orc7.owencline.com:6443

svc/php-helloworld - 172.30.74.79 ports 8080, 8443

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

build #1 pending for 6 seconds

deployment #1 waiting on image or update

2 infos identified, use 'oc status --suggest' to see details.

root@jumpserver01:~/dockerfile-create# oc status

In project s2i on server https://api.orc7.owencline.com:6443

svc/php-helloworld - 172.30.74.79 ports 8080, 8443

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

build #1 running for 14 seconds

deployment #1 waiting on image or update

2 infos identified, use 'oc status --suggest' to see details.

## Check OCP Status with Suggest Switch

root@jumpserver01:~/dockerfile-create# oc status --suggest

In project s2i on server https://api.orc7.owencline.com:6443

svc/php-helloworld - 172.30.74.79 ports 8080, 8443

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

build #1 running for 27 seconds - 8f873cc: initial commit (owencline <orcline@yahoo.com>)

deployment #1 waiting on image or update

Info:

\* dc/php-helloworld has no readiness probe to verify pods are ready to accept traffic or ensure deployment is successful.

try: oc set probe dc/php-helloworld --readiness ...

\* dc/php-helloworld has no liveness probe to verify pods are still running.

try: oc set probe dc/php-helloworld --liveness ...

View details with 'oc describe <resource>/<name>' or list everything with 'oc get all'.

## Check OCP Pods

The reason that we are checking the pods repeatedly is to verify that the application is being built successfully. The build is triggered automatically when the project was created above (see oc new-app).

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 1/1 Running 0 50s

## Check OCP Pods Again

root@jumpserver01:~/dockerfile-create# oc get pods -w

NAME READY STATUS RESTARTS AGE

php-helloworld-1-247jq 0/1 ContainerCreating 0 7s

php-helloworld-1-build 0/1 Completed 0 87s

php-helloworld-1-deploy 1/1 Running 0 15s

NAME READY STATUS RESTARTS AGE

php-helloworld-1-247jq 0/1 ContainerCreating 0 8s

php-helloworld-1-247jq 1/1 Running 0 19s

php-helloworld-1-deploy 0/1 Completed 0 28s

php-helloworld-1-deploy 0/1 Completed 0 28s

## Check OCP Pods Once More

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-247jq 1/1 Running 0 58s

php-helloworld-1-build 0/1 Completed 0 2m18s

php-helloworld-1-deploy 0/1 Completed 0 66s

## Get Service Name

From the list of pods above, we can see that the application is built and running (see php-helloworld-1-247jq). OCP should have created a service to act as a façade and load balancer to the pod running the application.

root@jumpserver01:~/dockerfile-create# oc get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

php-helloworld ClusterIP 172.30.74.79 <none> 8080/TCP,8443/TCP 2m44s

## Expose Service

If you want to access the application from the internet, you have to expose its service. When you expose the service, a route is created.

root@jumpserver01:~/dockerfile-create# oc expose svc/php-helloworld

route.route.openshift.io/php-helloworld exposed

## Get the Route to the Application Pod

The host/port is the URL to access your application.

root@jumpserver01:~/dockerfile-create# oc get routes

NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD

php-helloworld php-helloworld-s2i.apps.orc7.owencline.com php-helloworld 8080-tcp None

## Use Curl to Access the Application

root@jumpserver01:~/dockerfile-create# curl php-helloworld-s2i.apps.orc7.owencline.com

<html>

<head>

<title>Hello World!</title>

</head>

<body>

Hello World!<h3>My hostname is php-helloworld-1-247jq<br /><br />

PHP\_HELLOWORLD\_SERVICE listening on port 8443/TCP available at 8443<br />PHP\_HELLOWORLD\_SERVICE listening on port 8080/TCP available at 8080<br />PHP\_HELLOWORLD listening on port 8443/TCP available at tcp://172.30.74.79:8443<br />PHP\_HELLOWORLD listening on port 8080/TCP available at tcp://172.30.74.79:8080<br />KUBERNETES listening on port 443/TCP available at tcp://172.30.0.1:443<br />

</body>

</html>

## User Browser to Access Application



## Start Another Build

You can initiate new builds. For example, you can edit your source code and then, using Git, add, commit and push the changes from your local Git repository to the public Git repository. If you do make changes, you need to initiate another build.

root@jumpserver01:~/dockerfile-create# oc start-build php-helloworld

build.build.openshift.io/php-helloworld-2 started

## Check Pods

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-247jq 1/1 Running 0 7m41s

php-helloworld-1-build 0/1 Completed 0 9m1s

php-helloworld-1-deploy 0/1 Completed 0 7m49s

php-helloworld-2-build 0/1 Init:0/2 0 7s

## Check Status

root@jumpserver01:~/dockerfile-create# oc status --suggest

In project s2i on server https://api.orc7.owencline.com:6443

http://php-helloworld-s2i.apps.orc7.owencline.com to pod port 8080-tcp (svc/php-helloworld)

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

deployment #2 deployed 19 hours ago - 1 pod

deployment #1 deployed 19 hours ago

Info:

\* pod/php-helloworld-1-deploy has no liveness probe to verify pods are still running.

try: oc set probe pod/php-helloworld-1-deploy --liveness ...

\* pod/php-helloworld-2-deploy has no liveness probe to verify pods are still running.

try: oc set probe pod/php-helloworld-2-deploy --liveness ...

\* dc/php-helloworld has no readiness probe to verify pods are ready to accept traffic or ensure deployment is successful.

try: oc set probe dc/php-helloworld --readiness ...

\* dc/php-helloworld has no liveness probe to verify pods are still running.

try: oc set probe dc/php-helloworld --liveness ...

View details with 'oc describe <resource>/<name>' or list everything with 'oc get all'.

## Check Status Again

root@jumpserver01:~/dockerfile-create# oc status

In project s2i on server https://api.orc7.owencline.com:6443

http://php-helloworld-s2i.apps.orc7.owencline.com to pod port 8080-tcp (svc/php-helloworld)

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

deployment #2 deployed 19 hours ago - 1 pod

deployment #1 deployed 19 hours ago

4 infos identified, use 'oc status --suggest' to see details.

## Check Status Once More

root@jumpserver01:~/dockerfile-create# oc status --suggest

In project s2i on server https://api.orc7.owencline.com:6443

http://php-helloworld-s2i.apps.orc7.owencline.com to pod port 8080-tcp (svc/php-helloworld)

dc/php-helloworld deploys istag/php-helloworld:latest <-

bc/php-helloworld source builds https://github.com/owencline/php-helloworld.git on openshift/php:7.2

deployment #2 deployed 19 hours ago - 1 pod

deployment #1 deployed 19 hours ago

Info:

\* pod/php-helloworld-1-deploy has no liveness probe to verify pods are still running.

try: oc set probe pod/php-helloworld-1-deploy --liveness ...

\* pod/php-helloworld-2-deploy has no liveness probe to verify pods are still running.

try: oc set probe pod/php-helloworld-2-deploy --liveness ...

\* dc/php-helloworld has no readiness probe to verify pods are ready to accept traffic or ensure deployment is successful.

try: oc set probe dc/php-helloworld --readiness ...

\* dc/php-helloworld has no liveness probe to verify pods are still running.

try: oc set probe dc/php-helloworld --liveness ...

View details with 'oc describe <resource>/<name>' or list everything with 'oc get all'.

## Check Pods

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 18h

php-helloworld-1-deploy 0/1 Completed 0 18h

php-helloworld-2-7nktj 1/1 Running 0 18h

php-helloworld-2-build 0/1 Completed 0 18h

php-helloworld-2-deploy 0/1 Completed 0 18h

## Get the Service Name

root@jumpserver01:~/dockerfile-create# oc get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

php-helloworld ClusterIP 172.30.74.79 <none> 8080/TCP,8443/TCP 19h

root@jumpserver01:~/dockerfile-create# oc expose svc/php-helloworld

Error from server (AlreadyExists): routes.route.openshift.io "php-helloworld" already exists

## Get Routes

root@jumpserver01:~/dockerfile-create# oc get routes

NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD

php-helloworld php-helloworld-s2i.apps.orc7.owencline.com php-helloworld 8080-tcp None

## Use Curl to Access Application

root@jumpserver01:~/dockerfile-create# curl php-helloworld-s2i.apps.orc7.owencline.com

<html>

<head>

<title>Hello World! This is Owen's PHP app.</title>

</head>

<body>

Hello World!<h3>My hostname is php-helloworld-2-7nktj<br /><br />

PHP\_HELLOWORLD\_SERVICE listening on port 8443/TCP available at 8443<br />PHP\_HELLOWORLD\_SERVICE listening on port 8080/TCP available at 8080<br />PHP\_HELLOWORLD listening on port 8443/TCP available at tcp://172.30.74.79:8443<br />PHP\_HELLOWORLD listening on port 8080/TCP available at tcp://172.30.74.79:8080<br />KUBERNETES listening on port 443/TCP available at tcp://172.30.0.1:443<br />

</body>

</html>

## Start Another Build

root@jumpserver01:~/dockerfile-create# oc start-build php-helloworld

build.build.openshift.io/php-helloworld-3 started

## Check Pods

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 19h

php-helloworld-1-deploy 0/1 Completed 0 19h

php-helloworld-2-7nktj 1/1 Running 0 18h

php-helloworld-2-build 0/1 Completed 0 18h

php-helloworld-2-deploy 0/1 Completed 0 18h

php-helloworld-3-build 1/1 Running 0 17s

## Check Pods Again

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 19h

php-helloworld-1-deploy 0/1 Completed 0 19h

php-helloworld-2-7nktj 1/1 Running 0 18h

php-helloworld-2-build 0/1 Completed 0 18h

php-helloworld-2-deploy 0/1 Completed 0 18h

php-helloworld-3-build 1/1 Running 0 34s

## Check Pods Once More

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 19h

php-helloworld-1-deploy 0/1 Completed 0 19h

php-helloworld-2-7nktj 1/1 Running 0 18h

php-helloworld-2-build 0/1 Completed 0 18h

php-helloworld-2-deploy 0/1 Completed 0 18h

php-helloworld-3-build 1/1 Running 0 43s

## Check Pods Again

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 19h

php-helloworld-1-deploy 0/1 Completed 0 19h

php-helloworld-2-7nktj 1/1 Running 0 18h

php-helloworld-2-build 0/1 Completed 0 18h

php-helloworld-2-deploy 0/1 Completed 0 18h

php-helloworld-3-build 1/1 Running 0 46s

## Check Pods Again

root@jumpserver01:~/dockerfile-create# oc get pods

NAME READY STATUS RESTARTS AGE

php-helloworld-1-build 0/1 Completed 0 19h

php-helloworld-1-deploy 0/1 Completed 0 19h

php-helloworld-2-build 0/1 Completed 0 19h

php-helloworld-2-deploy 0/1 Completed 0 19h

php-helloworld-3-build 0/1 Completed 0 5m43s

php-helloworld-3-deploy 0/1 Completed 0 4m54s

php-helloworld-3-dns56 1/1 Running 0 4m46s

## Use Curl to Access Application

root@jumpserver01:~/dockerfile-create# curl php-helloworld-s2i.apps.orc7.owencline.com

<html>

<head>

<title>Hello World! This is Owen's PHP app.</title>

</head>

<body>

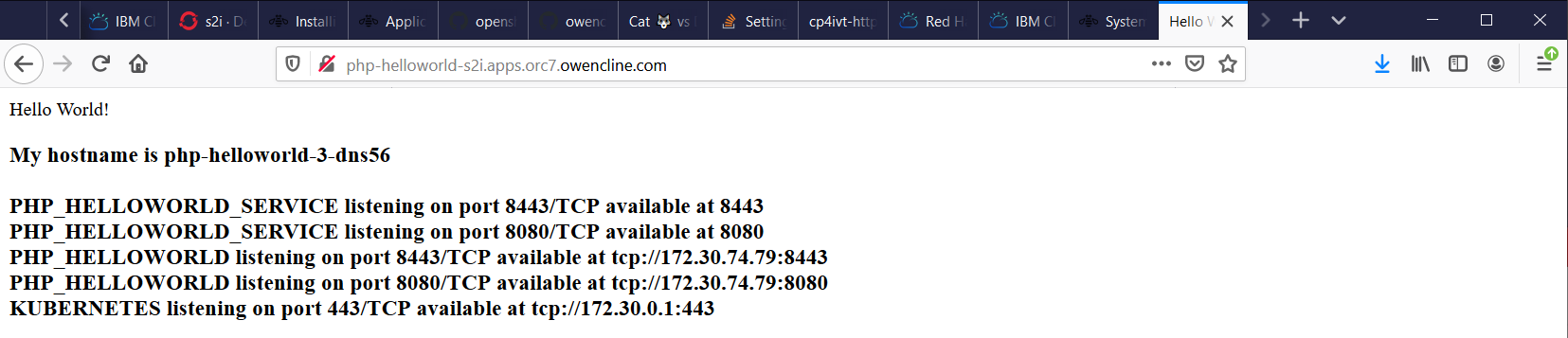
Hello World!<h3>My hostname is php-helloworld-3-dns56<br /><br />

PHP\_HELLOWORLD\_SERVICE listening on port 8443/TCP available at 8443<br />PHP\_HELLOWORLD\_SERVICE listening on port 8080/TCP available at 8080<br />PHP\_HELLOWORLD listening on port 8443/TCP available at tcp://172.30.74.79:8443<br />PHP\_HELLOWORLD listening on port 8080/TCP available at tcp://172.30.74.79:8080<br />KUBERNETES listening on port 443/TCP available at tcp://172.30.0.1:443<br />

</body>

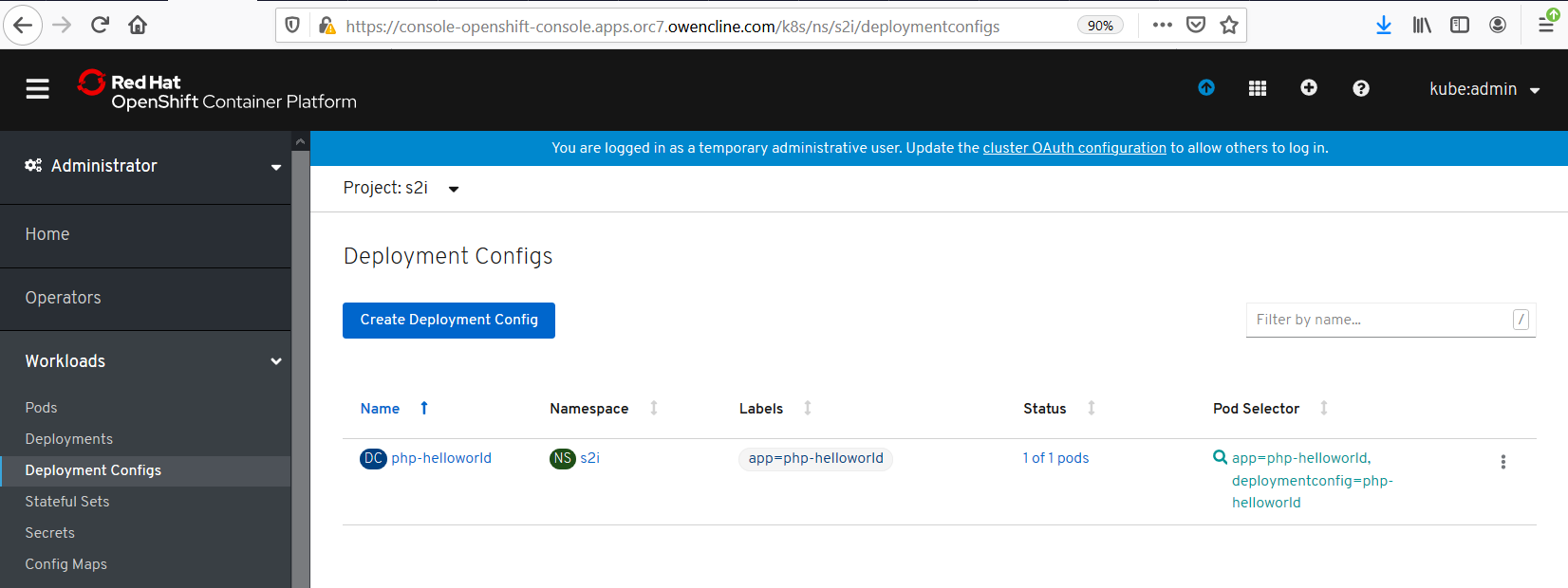
</html>

## Use Browser to Access Application

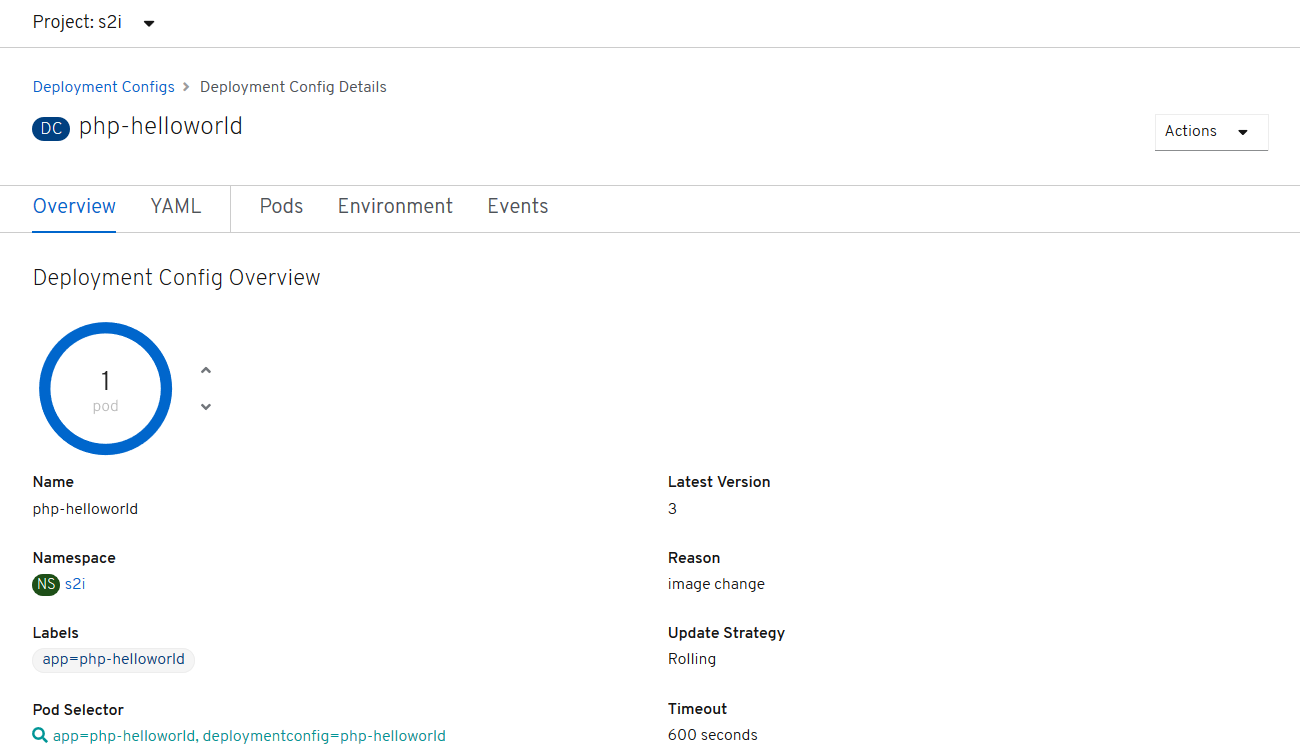


# OCP Artifacts that are Created

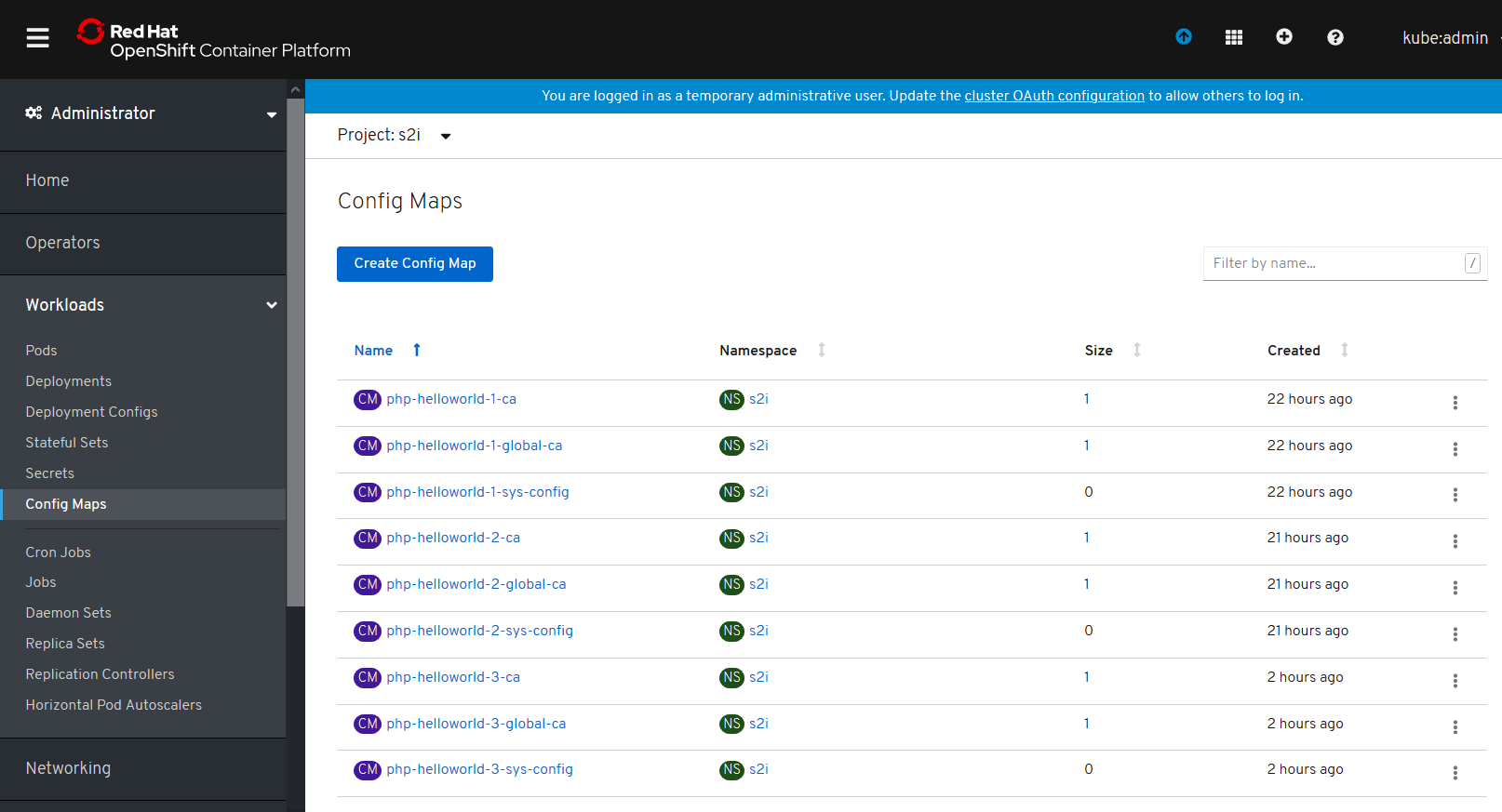
## Deployment Config







## Config Maps



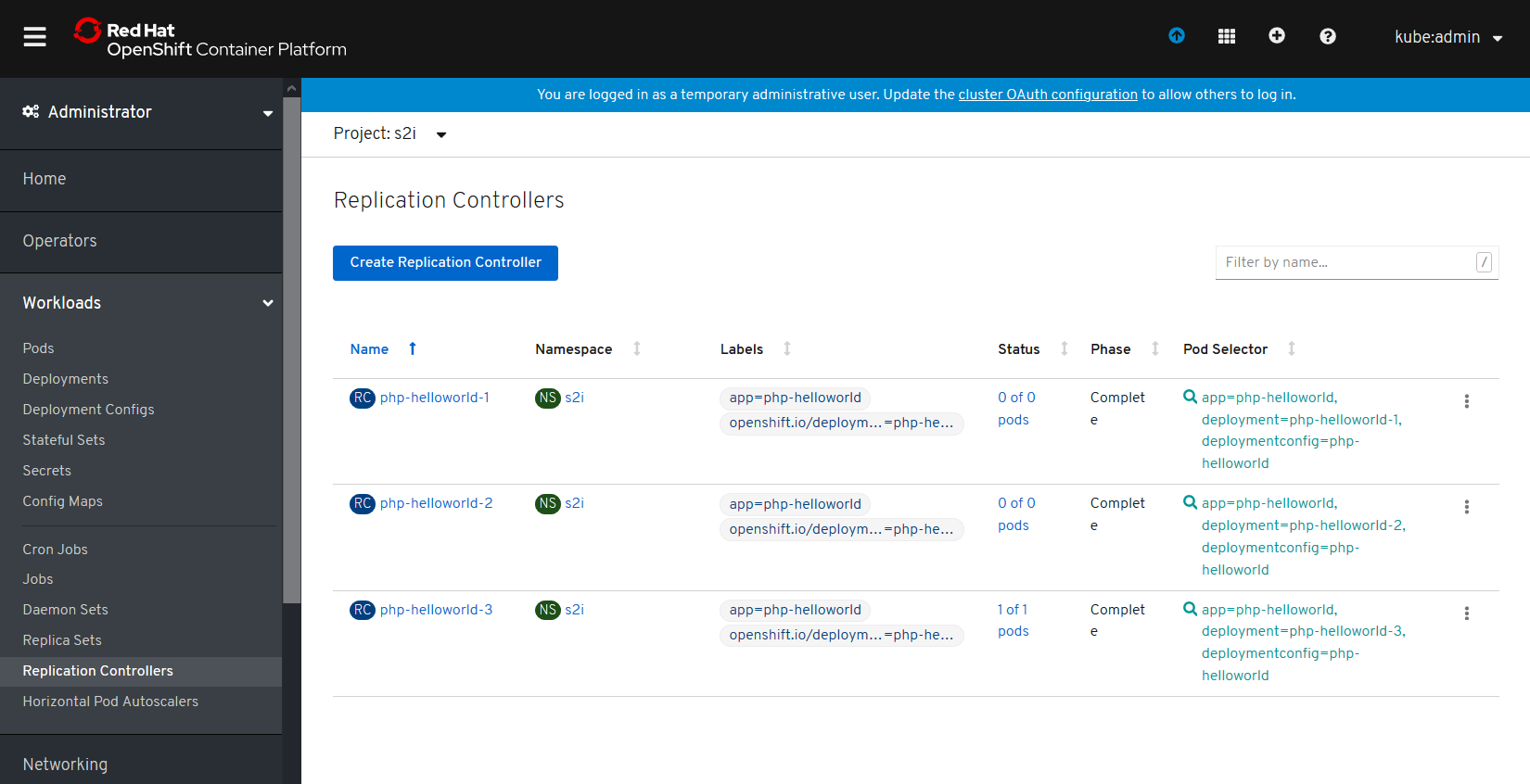
Config Map YAML Files for the third build of php-helloworld:





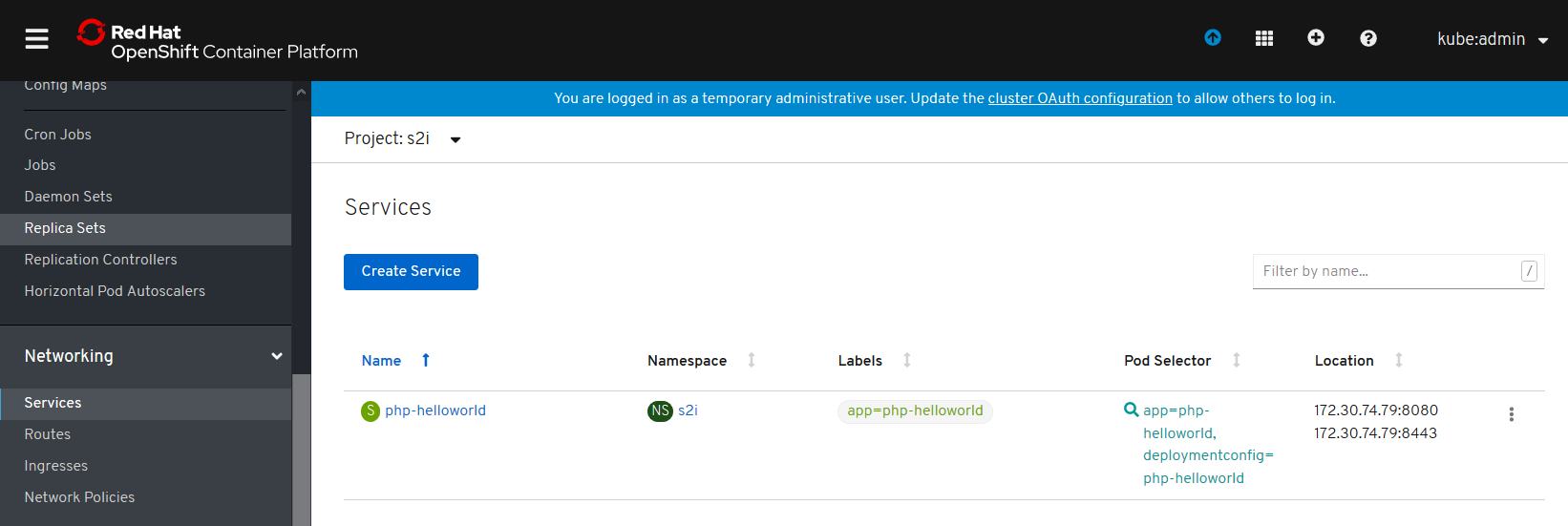


## Replication Controllers



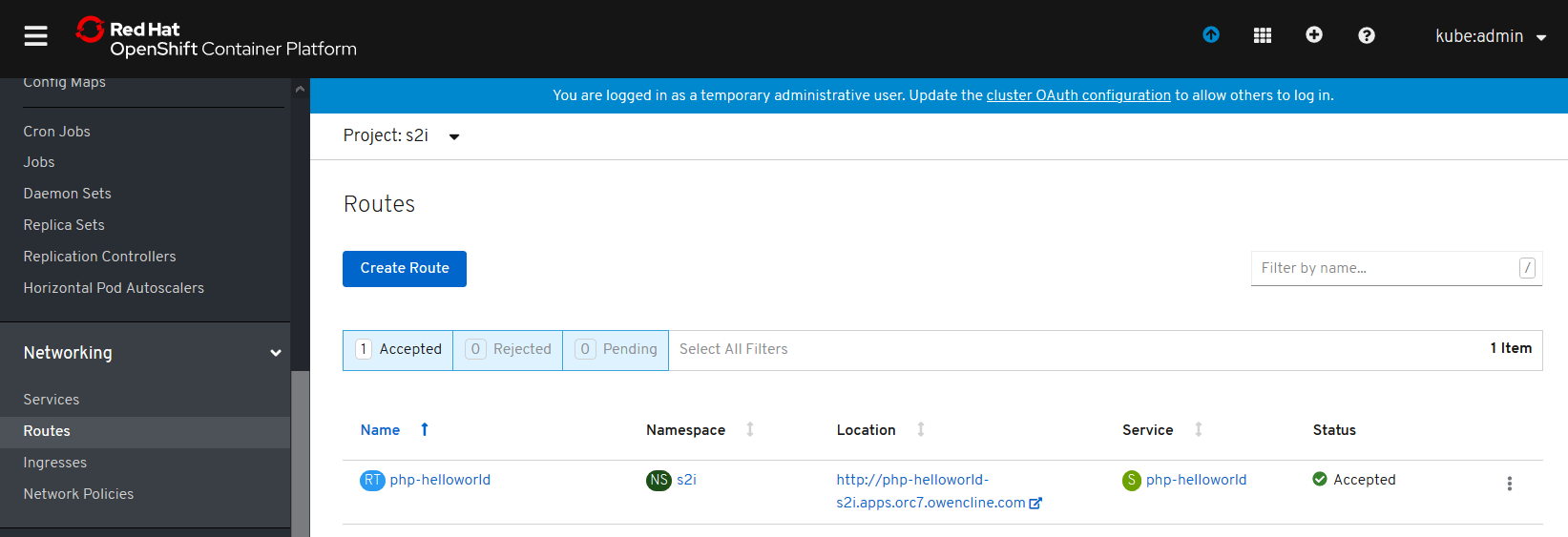


## Services



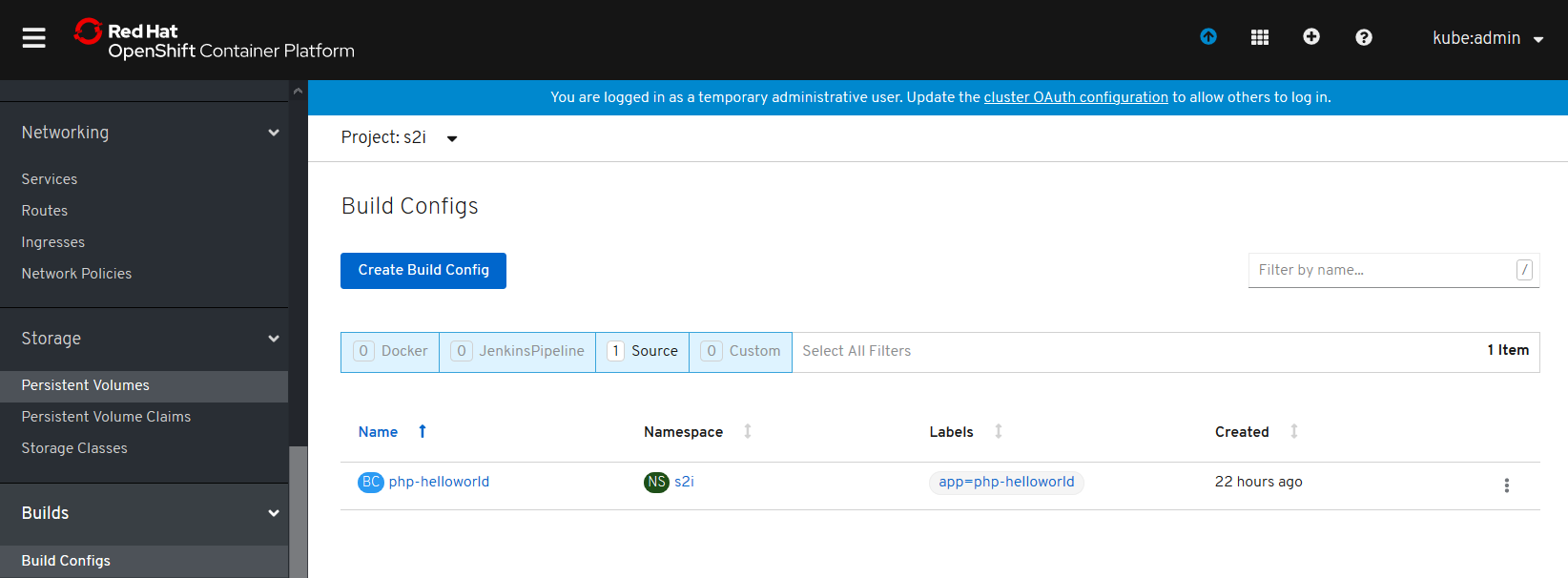


## Routes



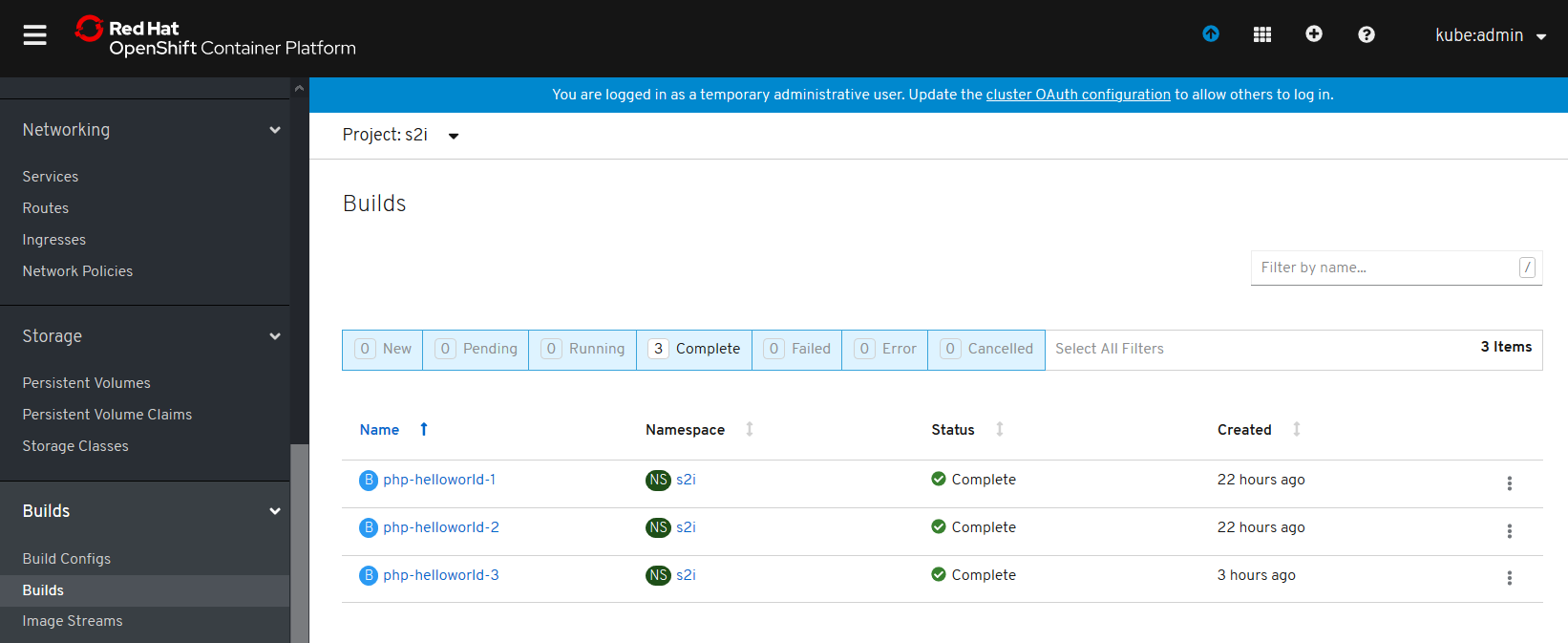


## Build Configs





## Builds





## Image Streams

