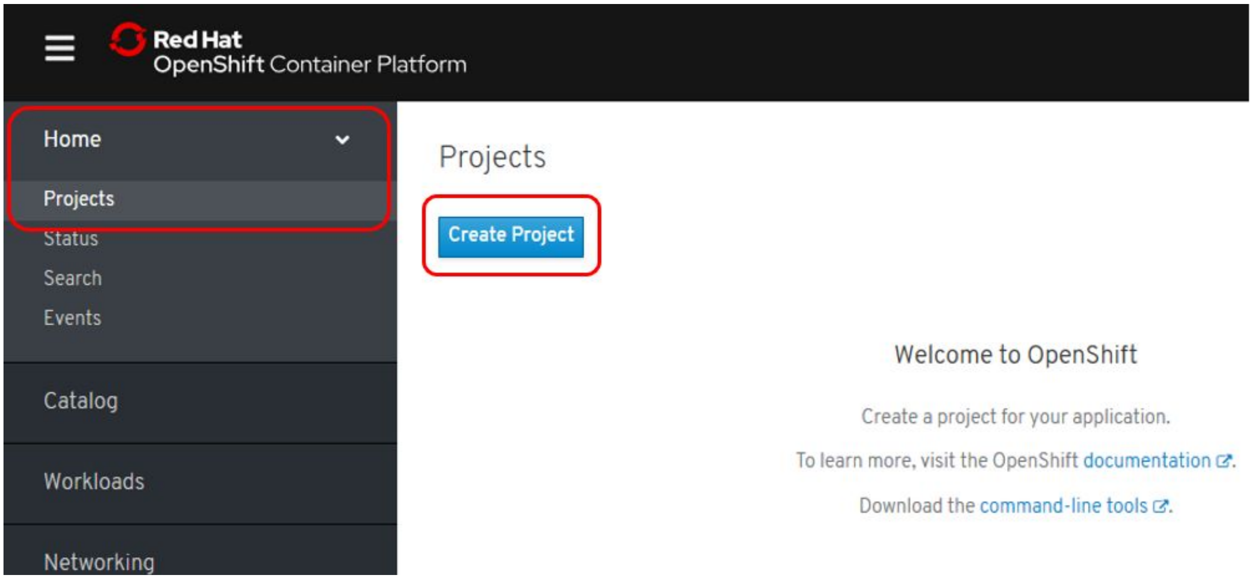
Openshift 4 Workshop Deploy from Source Code

Deploy from Source

In this workshop we will leverage the Openshift Console to create a new project and then add a new application to the project by pointing to a source code repository. The source code from the repository will be automatically retrieved, built and deployed within the container.

1. Log into Openshift Console - the Instructor will provide the URL and your user. Login with user<assigned #> and password is **r3dh4t1!**
2. Click on the **Home->Projects** menu, then select the **Create Project** button.



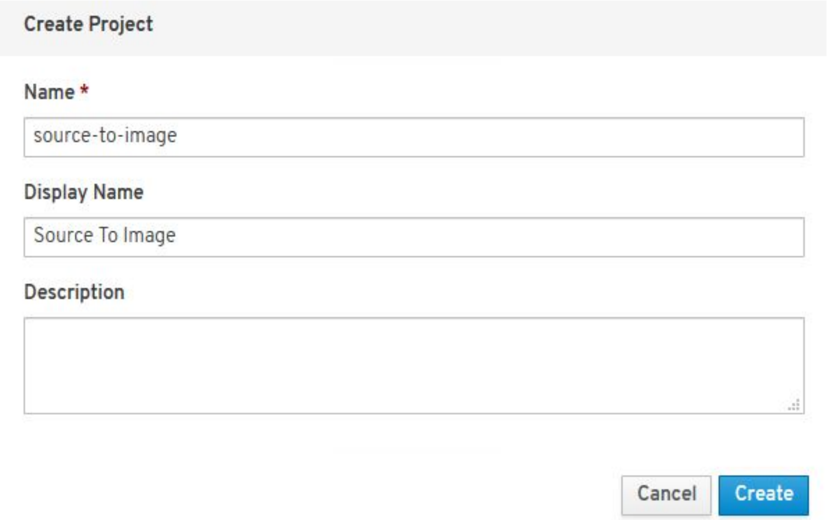
Welcome to OpenShift

Create a project for your application.

To learn more, visit the OpenShift documentation.

Download the command-line tools.

3. Populate **Create Project** dialog with the following information:



Create Project

Name *

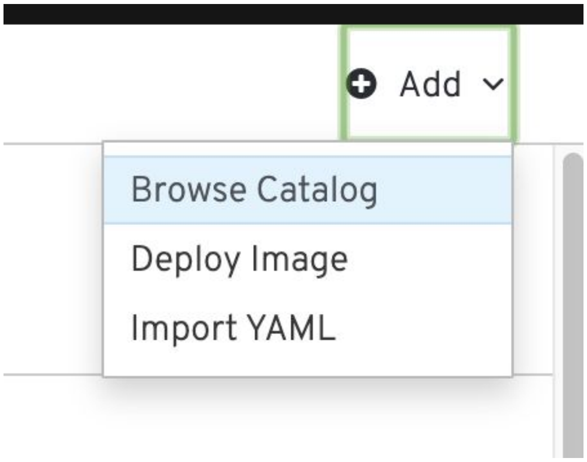
source-to-image

Display Name

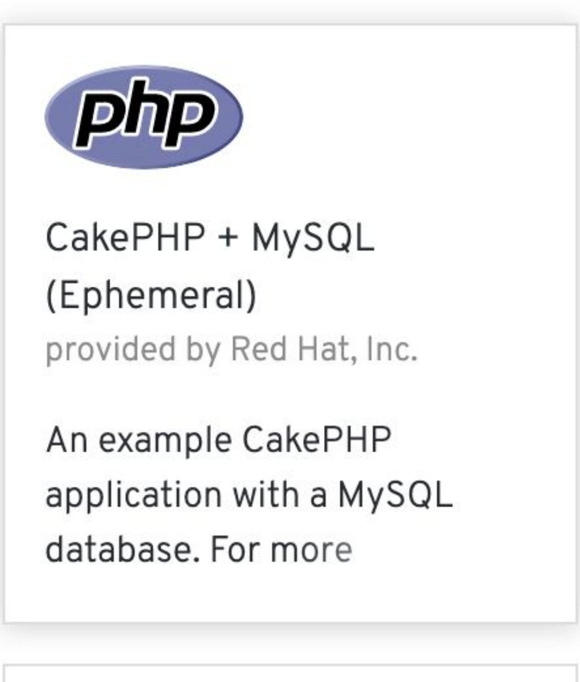
Source To Image

Description

4. In the source-to-image project, click on the **Add->Browse Catalog** button.



5. In the ***Filter by Keyword*** field search for **php**, select the **CakePHP with MySql Ephemeral** and click on it.

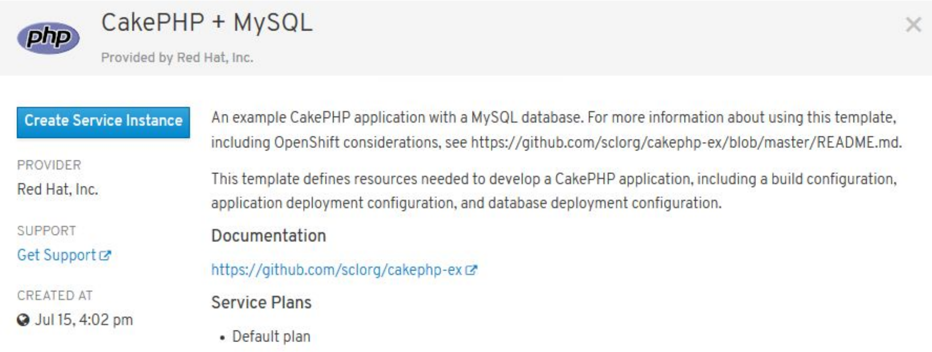


CakePHP + MySQL (Ephemeral)

provided by Red Hat, Inc.

An example CakePHP application with a MySQL database. For more

6. Click the **Create Service Instance** button. Take the default values on the screen that follows. Notice the field titled **"Git Repository URL"**, this field points to a GitHub repository where the source code for the application is stored. Openshift will retrieve the PHP code from the repository and use it when provisioning the container.



CakePHP + MySQL

Provided by Red Hat, Inc.

Create Service Instance

PROVIDER
Red Hat, Inc.

SUPPORT
Get Support

CREATED AT
Jul 15, 4:02 pm

An example CakePHP application with a MySQL database. For more information about using this template, including OpenShift considerations, see https://github.com/sclorg/cakephp-ex/blob/master/README.md.

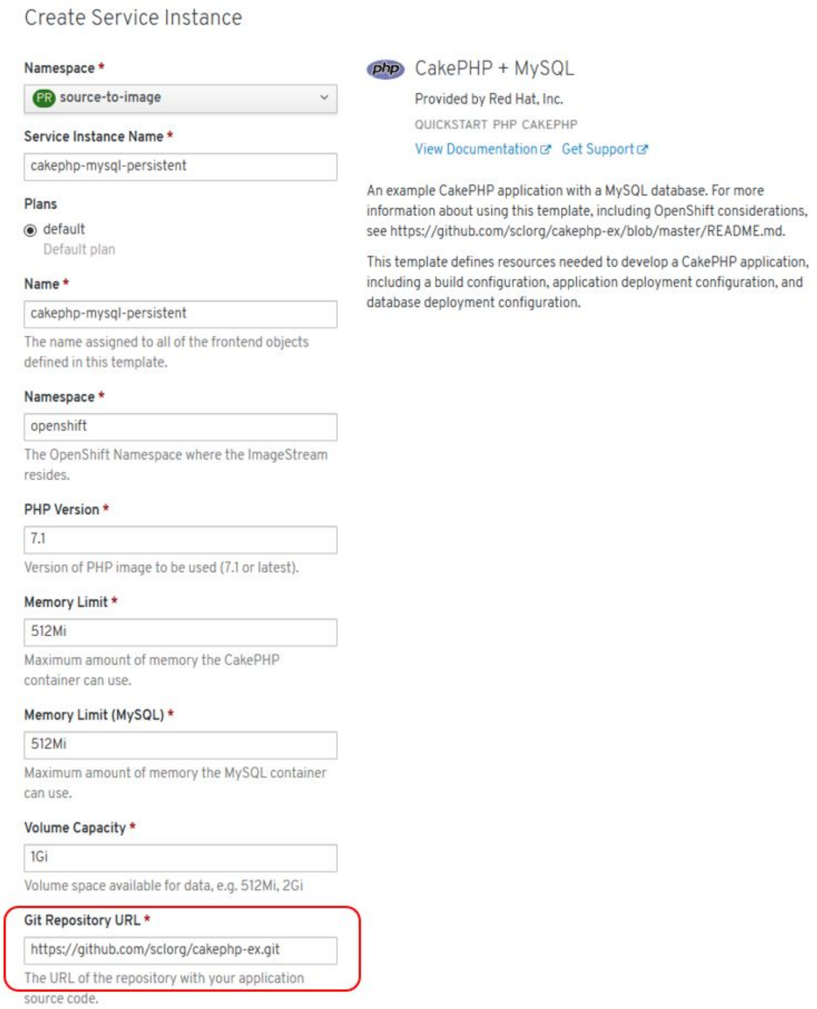
This template defines resources needed to develop a CakePHP application, including a build configuration, application deployment configuration, and database deployment configuration.

Documentation

https://github.com/sclorg/cakephp-ex

Service Plans

- Default plan



Create Service Instance

Namespace *
PR source-to-image

Service Instance Name *
cakephp-mysql-persistent

Plans
● default
Default plan

Name *
cakephp-mysql-persistent
The name assigned to all of the frontend objects defined in this template.

Namespace *
openshift
The OpenShift Namespace where the ImageStream resides.

PHP Version *
7.1
Version of PHP image to be used (7.1 or latest).

Memory Limit *
512Mi
Maximum amount of memory the CakePHP container can use.

Memory Limit (MySQL) *
512Mi
Maximum amount of memory the MySQL container can use.

Volume Capacity *
1Gi
Volume space available for data, e.g. 512Mi, 2Gi

Git Repository URL *
https://github.com/sclorg/cakephp-ex.git
The URL of the repository with your application source code.

CakePHP + MySQL

Provided by Red Hat, Inc.

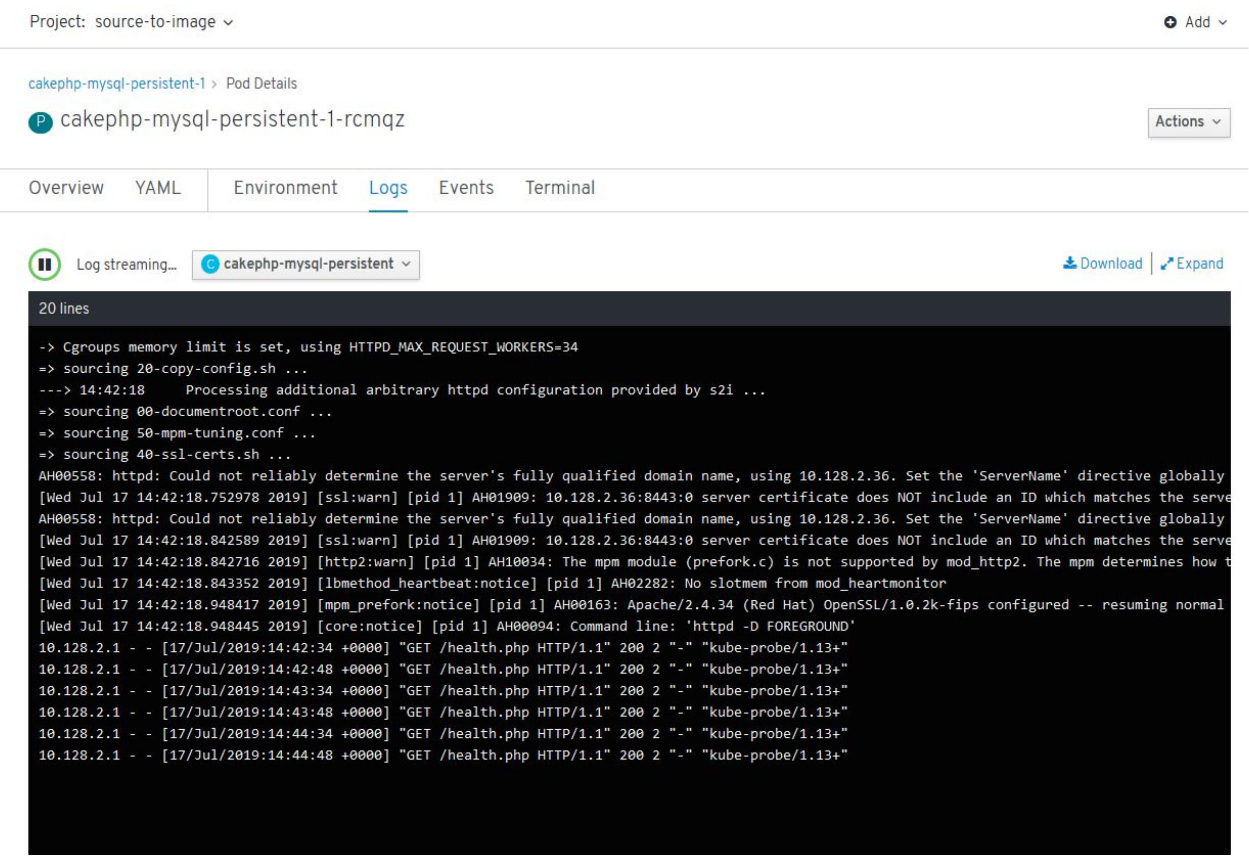
QUICKSTART PHP CAKEPHP

View Documentation Get Support

An example CakePHP application with a MySQL database. For more information about using this template, including OpenShift considerations, see https://github.com/sclorg/cakephp-ex/blob/master/README.md.

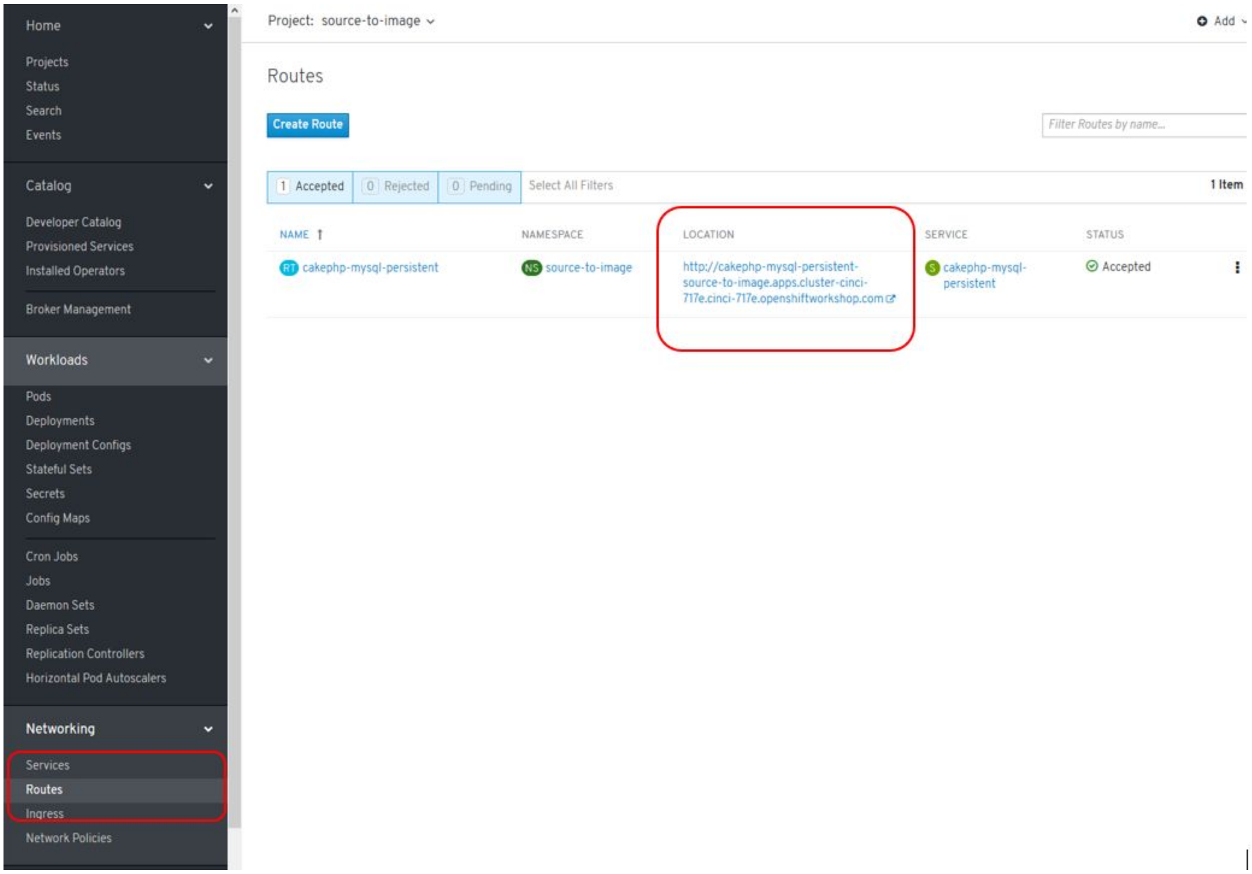
This template defines resources needed to develop a CakePHP application, including a build configuration, application deployment configuration, and database deployment configuration.

7. You can monitor the provisioning of the container by selecting the **Workloads->Pods** tab on the left. Select the **cakephp-mysql-persisten-1-xxxx** pod, and then select **Logs** tab. You will know the Pod was provisioned successfully when the log conveys "**Push Successful"** and then begins running the actual code.



```
-> Cgroups memory limit is set, using HTTPD_MAX_REQUEST_WORKERS=34
=> sourcing 20-copy-config.sh ...
---> 14:42:18     Processing additional arbitrary httpd configuration provided by s2i ...
=> sourcing 00-documentroot.conf ...
=> sourcing 50-mpm-tuning.conf ...
=> sourcing 40-ssl-certs.sh ...
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 10.128.2.36. Set the 'ServerName' directive globally
[Wed Jul 17 14:42:18.752978 2019] [ssl:warn] [pid 1] AH01909: 10.128.2.36:8443:0 server certificate does NOT include an ID which matches the serve
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 10.128.2.36. Set the 'ServerName' directive globally
[Wed Jul 17 14:42:18.842589 2019] [ssl:warn] [pid 1] AH01909: 10.128.2.36:8443:0 server certificate does NOT include an ID which matches the serve
[Wed Jul 17 14:42:18.842716 2019] [http2:warn] [pid 1] AH10034: The mpm module (prefork.c) is not supported by mod_http2. The mpm determines how t
[Wed Jul 17 14:42:18.843352 2019] [lbmethod_heartbeat:notice] [pid 1] AH02282: No slotmem from mod_heartmonitor
[Wed Jul 17 14:42:18.948417 2019] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.34 (Red Hat) OpenSSL/1.0.2k-fips configured -- resuming normal
[Wed Jul 17 14:42:18.948445 2019] [core:notice] [pid 1] AH00094: Command line: 'httpd -D FOREGROUND'
10.128.2.1 - - [17/Jul/2019:14:42:34 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
10.128.2.1 - - [17/Jul/2019:14:42:48 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
10.128.2.1 - - [17/Jul/2019:14:43:34 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
10.128.2.1 - - [17/Jul/2019:14:43:48 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
10.128.2.1 - - [17/Jul/2019:14:44:34 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
10.128.2.1 - - [17/Jul/2019:14:44:48 +0000] "GET /health.php HTTP/1.1" 200 2 "-" "kube-probe/1.13+"
```

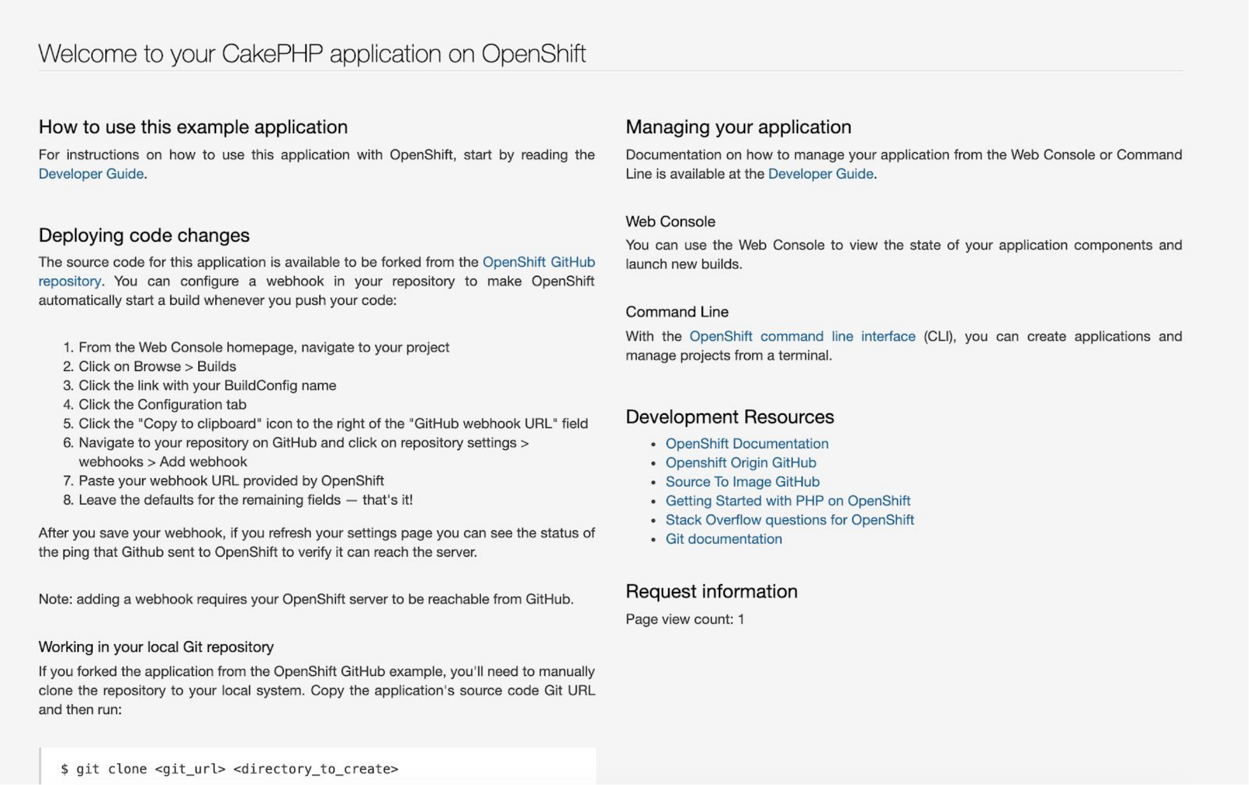
8. You can view the application in your web browser by locating its Route. To do this use the menu on the left and select **Networking->Routes**. This will display all routes. On the right panel you'll see the route for the **cakephp-mysql-persistent** application. Click on the **Location** field for this Route to display the application in another window.



Routes

NAME	NAMESPACE	LOCATION	SERVICE	STATUS
RT cakephp-mysql-persistent	NS source-to-image	http://cakephp-mysql-persistent-source-to-image.apps.cluster-cinci-717e.cinci-717e.openshiftworkshop.com	S cakephp-mysql-persistent	Accepted

9. The application should look like the following image:



Welcome to your CakePHP application on OpenShift

How to use this example application

For instructions on how to use this application with OpenShift, start by reading the Developer Guide.

Deploying code changes

The source code for this application is available to be forked from the OpenShift GitHub repository. You can configure a webhook in your repository to make OpenShift automatically start a build whenever you push your code:

1. From the Web Console homepage, navigate to your project
2. Click on Browse > Builds
3. Click the link with your BuildConfig name
4. Click the Configuration tab
5. Click the "Copy to clipboard" icon to the right of the "GitHub webhook URL" field
6. Navigate to your repository on GitHub and click on repository settings > webhooks > Add webhook
7. Paste your webhook URL provided by OpenShift
8. Leave the defaults for the remaining fields — that's it!

After you save your webhook, if you refresh your settings page you can see the status of the ping that Github sent to OpenShift to verify it can reach the server.

Note: adding a webhook requires your OpenShift server to be reachable from GitHub.

Working in your local Git repository

If you forked the application from the OpenShift GitHub example, you'll need to manually clone the repository to your local system. Copy the application's source code Git URL and then run:

```
$ git clone <git_url> <directory_to_create>
```

Managing your application

Documentation on how to manage your application from the Web Console or Command Line is available at the Developer Guide.

Web Console

You can use the Web Console to view the state of your application components and launch new builds.

Command Line

With the OpenShift command line interface (CLI), you can create applications and manage projects from a terminal.

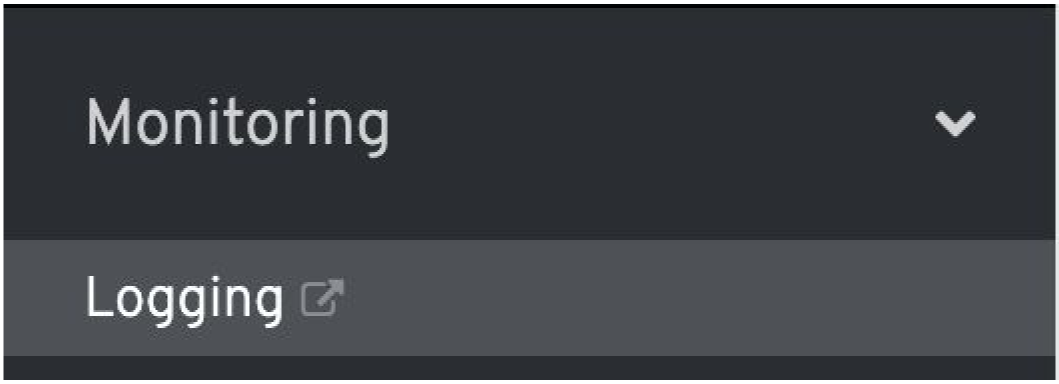
Development Resources

- OpenShift Documentation
- Openshift Origin GitHub
- Source To Image GitHub
- Getting Started with PHP on OpenShift
- Stack Overflow questions for OpenShift
- Git documentation

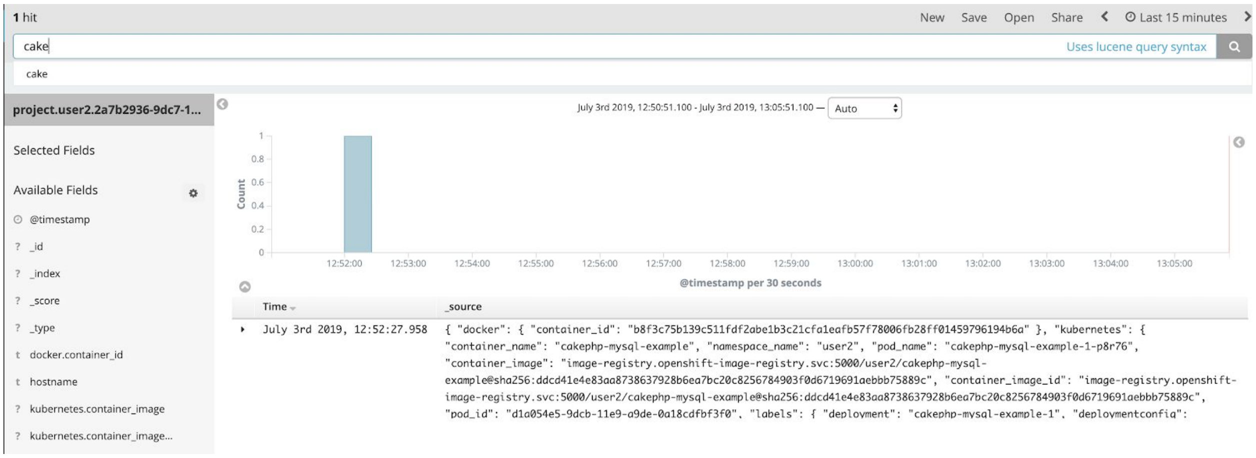
Request information

Page view count: 1

10. Check Monitoring available with Kibana



11. Click Logging, search for “cake”



```
July 3rd 2019, 12:52:27.958   { "docker": { "container_id": "b8f3c75b139c511fdf2abe1b3c21cfa1eafb57f78006fb28ff01459796194b6a" }, "kubernetes": {
                              "container_name": "cakephp-mysql-example", "namespace_name": "user2", "pod_name": "cakephp-mysql-example-1-p8r76",
                              "container_image": "image-registry.openshift-image-registry.svc:5000/user2/cakephp-mysql-
                              example@sha256:ddcd41e4e83aa8738637928b6ea7bc20c8256784903f0d6719691aebbb75889c", "container_image_id": "image-registry.openshift-
                              image-registry.svc:5000/user2/cakephp-mysql-example@sha256:ddcd41e4e83aa8738637928b6ea7bc20c8256784903f0d6719691aebbb75889c",
                              "pod_id": "d1a054e5-9dcb-11e9-a9de-0a18cdfbf3f0", "labels": { "deployment": "cakephp-mysql-example-1", "deploymentconfig":
```