

## ► Extra Lab

# Deploying Multi-Container Applications

### Performance Checklist

In this lab, you will deploy a PHP Application with a MySQL database using an OpenShift template to define the resources needed by the application.

### Outcomes

You should be able to create an OpenShift application comprised of multiple containers and access it through a web browser.

### Before You Begin

Open a terminal on `workstation` as the `student` user and run the following commands:

```
[student@workstation ~]$ lab multicontainer-review start  
[student@workstation ~]$ cd ~/D0180/labs/multicontainer-review
```

1. Log in to OpenShift cluster and create a new project for this exercise.
2. Build the Database container image located in the `images/mysql` directory and publish it to your Quay.io repository.
3. Build the PHP container image located in the `images/quote-php` and publish it to your Quay.io repository.



#### Warning

Make sure both repositories are public in `quay.io` so OpenShift can get the images from it. Refer to the `Repositories Visibility` section of the Appendix C to read details about how change repository visibility.

4. Go to the `/home/student/D0180/labs/multicontainer-review/` directory and review the provided template `quote-php-template.json`.  
Note the definitions and configuration of the pods, services, and persistent volume claims defined in the template.
5. Upload the PHP application template so that any developer with access to your project can use it.
6. Process the uploaded template and create the application resources.
7. Expose the service.
8. Test the application and verify that it outputs an inspiring message.

### Evaluation

Grade your work by running the `lab multicontainer-review grade` command from your `workstation` machine. Correct any reported failures and rerun the script until successful.

```
[student@workstation ~]$ lab multicontainer-review grade
```

## Finish

To complete this lab, run the `lab multicontainer-review finish` command on `workstation`.

```
[student@workstation ~]$ lab multicontainer-review finish
```

This concludes the lab.

## ► Solution

# Deploying Multi-Container Applications

### Performance Checklist

In this lab, you will deploy a PHP Application with a MySQL database using an OpenShift template to define the resources needed by the application.

### Outcomes

You should be able to create an OpenShift application comprised of multiple containers and access it through a web browser.

### Before You Begin

Open a terminal on `workstation` as the `student` user and run the following commands:

```
[student@workstation ~]$ lab multicontainer-review start
[student@workstation ~]$ cd ~/D0180/labs/multicontainer-review
```

1. Log in to OpenShift cluster and create a new project for this exercise.

- 1.1. From `workstation`, log in as the user provided at the first exercise.

```
[student@workstation multicontainer-review]$ source /usr/local/etc/ocp4.config
[student@workstation multicontainer-review]$ oc login -u ${RHT_OCP4_DEV_USER} \
> -p ${RHT_OCP4_DEV_PASSWORD} ${RHT_OCP4_MASTER_API}
Login successful.
```

You don't have any projects. You can try to create a new project, by running

```
oc new-project <projectname>
```

If the `oc login` command prompts about using insecure connections, answer **y** (yes).

- 1.2. Create a new project in OpenShift named `deploy` and prefixed by your OpenShift username:

```
[student@workstation multicontainer-review]$ oc new-project \
> ${RHT_OCP4_DEV_USER}-deploy
Now using project ...output omitted...
```

2. Build the Database container image located in the `images/mysql` directory and publish it to your Quay.io repository.
  - 2.1. Build the MySQL Database image using the provided Dockerfile in the `images/mysql` directory.

```
[student@workstation multicontainer-review]$ cd images/mysql
[student@workstation mysql]$ sudo podman build -t do180-mysql-57-rhel7 .
STEP 1: FROM rhscl/mysql-57-rhel7
...output omitted...
STEP 4: COMMIT do180-mysql-57-rhel7
397a...5cfb
```

2.2. Push the MySQL image to the your Quay.io repository.

In order to make the image available for OpenShift to use in the template, give it the tag `quay.io/${RHT_OCP4_QUAY_USER}/do180-mysql-57-rhel7` and push it to the `quay.io` registry. In order to push images to `quay.io` you first need to log in with your own credentials.

```
[student@workstation mysql]$ sudo podman login quay.io -u ${RHT_OCP4_QUAY_USER}
Password: your_quay_password
Login Succeeded!
```

To tag and push the image run the following commands in the terminal window.

```
[student@workstation mysql]$ sudo podman tag do180-mysql-57-rhel7 \
> quay.io/${RHT_OCP4_QUAY_USER}/do180-mysql-57-rhel7
[student@workstation mysql]$ sudo podman push \
> quay.io/${RHT_OCP4_QUAY_USER}/do180-mysql-57-rhel7
Getting image source signatures
...output omitted...
Writing manifest to image destination
Storing signatures
```

Return to the previous directory.

```
[student@workstation mysql]$ cd ~/D0180/labs/multicontainer-review
```

3. Build the PHP container image located in the `images/quote-php` and publish it to your Quay.io repository.



**Warning**

Make sure both repositories are public in `quay.io` so OpenShift can get the images from it. Refer to the `Repositories Visibility` section of the Appendix C to read details about how change repository visibility.

3.1. Build the PHP image using the provided Dockerfile in the `images/quote-php` directory.

```
[student@workstation multicontainer-review]$ cd images/quote-php
[student@workstation quote-php]$ sudo podman build -t do180-quote-php .
STEP 1: FROM registry.access.redhat.com/ubi8/ubi
...output omitted...
STEP 8: COMMIT do180-quote-php
271f...525d
```

3.2. Tag and push the PHP image to your Quay.io registry.

In order to make the image available for OpenShift to use in the template, give it the tag of `quay.io/${RHT_OCP4_QUAY_USER}/do180-quote-php` and push it to Quay.io.

```
[student@workstation quote-php]$ sudo podman tag do180-quote-php \
> quay.io/${RHT_OCP4_QUAY_USER}/do180-quote-php
[student@workstation quote-php]$ sudo podman push \
> quay.io/${RHT_OCP4_QUAY_USER}/do180-quote-php
Getting image source signatures
...output omitted...
Writing manifest to image destination
Storing signatures
```

4. Go to the `/home/student/D0180/labs/multicontainer-review` directory and review the provided template `quote-php-template.json` file.

Note the definitions and configuration of the pods, services, and persistent volume claims defined in the template.

```
[student@workstation quote-php]$ cd ~/D0180/labs/multicontainer-review
```

5. Upload the PHP application template so that any developer with access to your project can use it.

Use the `oc create -f` command to upload the template file to the project.

```
[student@workstation multicontainer-review]$ oc create -f quote-php-template.json
template.template.openshift.io/quote-php-persistent created
```

6. Process the uploaded template and create the application resources.

- 6.1. Use the `oc process` command to process the template file. Make sure providing the `RHT_OCP4_QUAY_USER` parameter with the `quay.io` namespace where images are located. Use the pipe command to send the result to the `oc create` command to create an application from the template.

```
[student@workstation multicontainer-review]$ oc process quote-php-persistent \
> -p RHT_OCP4_QUAY_USER=${RHT_OCP4_QUAY_USER} \
> | oc create -f -
pod/mysql created
pod/quote-php created
service/quote-php created
service/mysql created
persistentvolumeclaim/dbinit created
persistentvolumeclaim/dbclaim created
```

- 6.2. Verify the status of the deployment using the `oc get pods` command with the `-w` option to monitor the deployment status. Wait until both pods are running. It may take some time for both pods to start up.

```
[student@workstation multicontainer-review]$ oc get pods -w
NAME      READY   STATUS            RESTARTS   AGE
mysql     0/1    ContainerCreating   0          21s
quote-php 0/1    ContainerCreating   0          20s
quote-php 1/1    Running           0          35s
mysql     1/1    Running           0          49s
^C
```

Press **Ctrl+C** to exit the command.

## 7. Expose the service.

To allow the PHP Quote application to be accessible through the OpenShift router and reachable from an external network, use the `oc expose` command to expose the `quote-php` service.

Run the following command in the terminal window.

```
[student@workstation multicontainer-review]$ oc expose svc quote-php
route.route.openshift.io/quote-php exposed
```

## 8. Test the application and verify that it outputs an inspiring message.

- 8.1. Use the `oc get route` command to find the FQDN where the application is available.  
Note the FQDN for the app.

Run the following command in the terminal window.

```
[student@workstation multicontainer-review]$ oc get route
NAME      HOST/PORT                               PATH  SERVICES  ...
quote-php  quote-php-your_dev_user-deploy.wildcard_domain  quote-php ...
```

- 8.2. Use the `curl` command to test the REST API for the PHP Quote application.

```
[student@workstation ~]$ curl -w "\n" \
> http://quote-php-${RHT_OCP4_DEV_USER}-deploy.${RHT_OCP4_WILDCARD_DOMAIN}
Always remember that you are absolutely unique. Just like everyone else.
```



### Note

The text displayed in the output above may differ, but the `curl` command should run successfully.

## Evaluation

Grade your work by running the `lab multicontainer-review grade` command from your workstation machine. Correct any reported failures and rerun the script until successful.

```
[student@workstation ~]$ lab multicontainer-review grade
```

## Finish

To complete this lab, run the `lab multicontainer-review finish` command on workstation.

```
[student@workstation ~]$ lab multicontainer-review finish
```

This concludes the lab.