Lab: Container Orchestration with Kubernetes and OpenShift

Debug and deploy a multi-container application to the Red Hat OpenShift Container Platform (RHOCP).

Outcomes

You should be able to:

- Verify and correct the configuration of the `Service` and `Deployment` RHOCP objects.
- Deploy RHOCP objects.

In this exercise, your task is to deploy the `quotes` application to RHOCP.

The `quotes` application uses the `quotes-api` and `quotes-ui` containerized microservices.

Your colleague managed to deploy the first tier of the application, the `quotes-ui` container, to RHOCP. However, the pod crashes and does not respond to requests. Additionally, the colleague faces difficulties when trying to deploy the `quotes-api` container to RHOCP. You, the RHOCP expert in the company, are tasked with helping your colleague.

As the `student` user on the `workstation` machine, use the `lab` command to:

- Create the `ocp-lab` project.
- Deploy the `quotes-ui` microservice.

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

The lab script continuously evaluates the objectives of this lab. Keep the script running in a terminal window and complete the objectives of this lab from a new terminal window.

Instructions

1. Log in to the cluster as the `developer` user, and ensure that you use the `ocp-lab` project.

   Log in to the cluster as the `developer` user.

   ```
   [student@workstation ~]$ oc login -u developer -p developer \
     https://api.ocp4.example.com:6443
   Login successful.

   ...output omitted...
   ```

   Ensure that you use the `ocp-lab` project.

   ```
   [student@workstation ~]$ oc project ocp-lab
   Already on project "ocp-lab" on server "https://api.ocp4.example.com:6443".
   ```

2. Change to the `/home/student/DO188/labs/openshift-lab/` directory.

   This directory contains the `quotes-api` YAML files that your colleague created. Be aware that the YAML files might contain mistakes.

   ```
   [student@workstation ~]$ cd ~/DO188/labs/openshift-lab/
   ```

3. Use the `deployment.yaml` file to deploy the `quotes-api` container in the `ocp-lab` RHOCP project.

   Try to create the deployment by using the `deployment.yaml` file.

   ```
   [student@workstation openshift-lab]$ oc create -f deployment.yaml
   The Deployment "quotes-api" is invalid: spec.template.metadata.labels: Invalid value: map[string]string{"app":"quotes-
   api"}: `selector` does not match template `labels`
   ```

   The deployment defines an application pod with the `app=quotes-api` label. However, the `spec.selector.matchLabels` field uses a different label.

   Open the `deployment.yaml` file in a text editor, such as `gedit`, and modify the `spec.selector.matchLabels` field to use the same label as the `spec.template.metadata.labels` field.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: quotes-api
  name: quotes-api
spec:
  replicas: 1
  selector:
    matchLabels:
      app: quotes-api
  template:
    metadata:
      labels:
        app: quotes-api
    spec:
      containers:
      - image: registry.ocp4.example.com:8443/redhattraining/podman-quotes-api:openshift
        name: podman-quotes-api
```

Create the deployment by using the `deployment.yaml` file.

```
[student@workstation openshift-lab]$ oc create -f deployment.yaml
deployment.apps/quotes-api created
```

Verify that the `quotes-api` application pod is in the `RUNNING` state.

```
[student@workstation openshift-lab]$ oc get po
NAME                          READY   STATUS             RESTARTS        AGE
quotes-api-6c9f758574-nk8kd   1/1     Running            0               5s
quotes-ui-d7d457674-mljrb     0/1     CrashLoopBackOff   15 (3m9s ago)   55m
```

If the application pod is in the `ContainerCreating` state, then execute the previous command again after a few seconds.

4. Use the `service.yaml` file to configure the `quotes-ui` container networking in the `ocp-lab` project.

   Configure the `service.yaml` file to conform to the following requirements:

   - The `quotes-ui` container must reach the `quotes-api` container at the `http://quotes-api:8080` URL.
   - The `quotes-api` container listens on port `8080` by default.
   - Deploy the `quotes-ui` container after the `quotes-api` container becomes available on the `quotes-api` host. The application architect advised you to restart the `quotes-ui` application if it is deployed in the incorrect order.

   > **NOTE**
   >
   > If you make a mistake, delete and recreate the `Service` object.
   >
   > For example, you can use the `oc delete -f service.yaml` command to delete the `Service` object.

   Open the `service.yaml` file in a text editor, such as `gedit`. Then, configure the service to serve on port `8080`.

```
...file omitted...
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 3000
  selector:
    app: quotes
```

Configure the service to send requests to port `8080`.

```
...file omitted...
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    app: quotes
```

Configure the service to send requests to pods with the `quotes-api` label.

```
...file omitted...
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    app: quotes-api
```

Configure the service to be available on the `quotes-api` hostname.

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: quotes
  name: quotes-api
...file omitted...
```

Create the service by using the `service.yaml` file.

```
[student@workstation openshift-lab]$ oc create -f service.yaml
service/quotes-api created
```

Verify the service configuration.

The endpoint IP address might differ in your output.

```
[student@workstation openshift-lab]$ oc describe service quotes-api
Name:              quotes-api
Namespace:         ocp-lab
Labels:            app=quotes
Annotations:       <none>
Selector:          app=quotes-api
...output omitted...
Port:              <unset>  8080/TCP
TargetPort:        8080/TCP
Endpoints:         10.8.0.102:8080
...output omitted...
```

If your output differs from the highlighted output of the previous command, return to the previous steps and ensure you configured your service correctly.

Verify that the `quotes-ui` container is still failing.

```
[student@workstation openshift-lab]$ oc get po
NAME                          READY   STATUS             RESTARTS        AGE
quotes-api-6c9f758574-nk8kd   1/1     Running            0               20m
quotes-ui-d7d457674-mljrb     0/1     CrashLoopBackOff   15 (3m9s ago)   55m
```

Restart the `quotes-ui` container.

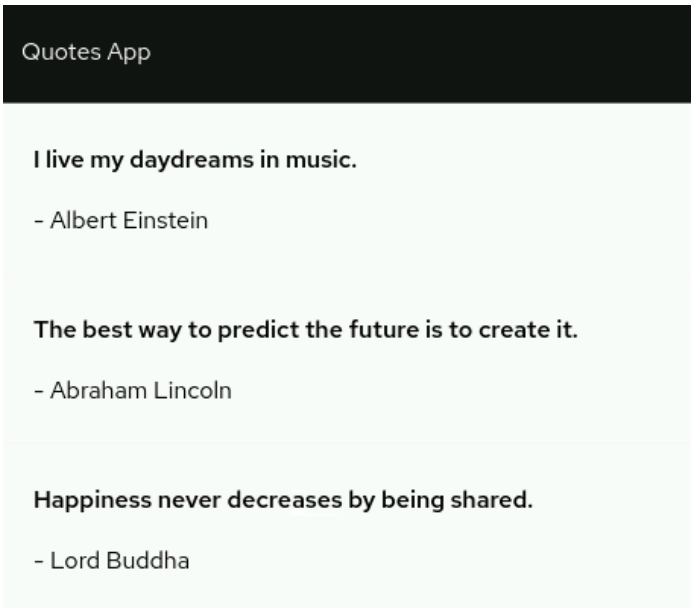
You can delete containers that contain the `app=quotes-ui` label, and let the `quotes-ui` deployment recreate the container.

```
[student@workstation openshift-lab]$ oc delete pod -l app=quotes-ui
pod "quotes-ui-d7d457674-9cw7l" deleted
```

Then, verify that the `quotes-ui` deployment created a new container.

```
[student@workstation openshift-lab]$ oc get po
NAME                          READY   STATUS    RESTARTS   AGE
quotes-api-6c9f758574-nk8kd   1/1     Running   0          39m
quotes-ui-d7d457674-rbkl7     1/1     Running   0          67s
```

5. In a web browser, go to `http://quotes-ui-ocp-lab.apps.ocp4.example.com` and verify that the application works.



Finish

As the `student` user on the `workstation` machine, use the `lab` command to complete this exercise. This is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish openshift-lab
```