Guided Exercise: Managing Workloads and Operators

In this exercise, you will manage cluster workloads with the web console.

**Outcomes**

You should be able to use the OpenShift web console to:

* Install an operator from OperatorHub.
* Use a custom resource to create a database.
* Deploy and troubleshoot an application that uses the operator-managed resources.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command ensures that the cluster API is reachable and creates the resources required for this activity.

**[student@workstation ~]$ lab console-workloads start**

1. As the admin user, locate and navigate to the OpenShift web console.
   1. Log in to your OpenShift cluster as the admin user.
   2. **[student@workstation ~]$ oc login -u admin -p redhat \**
   3. **>**  **https://api.ocp4.example.com:6443**
   4. Login successful.

*...output omitted...*

* 1. Identify the URL for the web console.
  2. **[student@workstation ~]$ oc whoami --show-console**

https://console-openshift-console.apps.ocp4.example.com

* 1. Open a web browser and navigate to https://console-openshift-console.apps.ocp4.example.com.
  2. Click **Advanced** to reveal the untrusted certificate message, and then click **Add Exception**. In the **Add Security Exception** dialog box, click **Confirm Security Exception**.

You will need to do this twice to skip the warnings about the self-signed SSL certificate for both the console-openshift-console and oauth-openshift subdomains.

* 1. Click **localusers** and log in as the admin user with the password of redhat.

1. Inspect the openshift-console-operator and openshift-console deployments, replica sets, and pods.
   1. Click **Workloads** → **Deployments**, and select all projects from the project list at the top. Type console in the **Search by name** field.

Notice that OpenShift has a deployment named console-operator with a single pod in the openshift-console-operator namespace, which operates a deployment named console in the openshift-console namespace.

|  |
| --- |
|  |

* 1. Click **Workloads** → **Replica Sets**, and type console in the **Search by name** field.

Deployments declare a ReplicaSet to ensure that a specified number of pods are always running.

* 1. In the status column, click **2 of 2 pods** to display the console ReplicaSet pod list.

1. Install the community PostgreSQL operator provided by Dev4Devs.com from the **OperatorHub** page.
   1. Click **Operators** → **OperatorHub**, and then click **Database** to display the list of database operators available from OperatorHub.
   2. Type postgres in the **Filter by keyword** field, and then click **PostgreSQL Operator by Dev4Ddevs.com**. Click **Continue** to view the community operator page, and then click **Install**.

|  |
| --- |
|  |

* 1. Select the console-apps namespace, and then click **Install** to install the operator for use in the console-apps project. Leave the other form fields unchanged.

1. Log out as the admin user and log in as the developer user.
   1. Click **admin** → **Log out**.
   2. Log in as the user developer with a password of developer.
2. Provision a PostgreSQL database using the installed operator and Database Custom Resource Definition (CRD).
   1. On the **Projects** page, click the console-apps link to see the resources associated with the console-apps project.
   2. Click **Operators** → **Installed Operators**, and then click the PostgreSQL Operator by Dev4Ddevs.com link to display the **Operator Details** page.

**NOTE**

If the **Installed Operators** list does not load, make sure that the console-apps project is selected at the top of the page.

|  |
| --- |
|  |

* 1. Click the **Database Database** tab and then click **Create Database**.
  2. Switch from **Form View** to **YAML View**, and then update the Database YAML to specify the PostgreSQL image provided by Red Hat. Do not change the other default values.
  3. apiVersion: postgresql.dev4devs.com/v1alpha1
  4. kind: Database
  5. metadata:
  6. name: database
  7. namespace: console-apps
  8. spec:
  9. *...output omitted...*
  10. databaseUserKeyEnvVar: POSTGRESQL\_USER
  11. image: **registry.redhat.io/rhscl/postgresql-96-rhel7:1-51**

size: 1

* 1. Click **Create** to add the Database resource. The PostgreSQL operator will read the specification and automatically create the workload, network, and storage for the new database.

1. Review the resources created by the operator.
   1. Click **Workloads** → **Deployments**, and inspect the list of deployments. You will notice a database deployment and a postgresql-operator deployment.
   2. Click the database deployment, and then click the **Pods** tab to see the pod deployed by the database deployment. Click the pod name to display the **Pod Details** page.
   3. Click **Networking** → **Services**, and then click the database service name to see the details of the service created by the PostgreSQL operator.
   4. Click **Storage** → **Persistent Volume Claims**, and then click the database PVC to see the details of the Persistent Volume Claim created by the PostgreSQL operator.
2. Create a deployment, service, and route for a simple web application. The application will display a list of books stored in the database.
   1. Click **Workloads** → **Deployments**, and then click **Create Deployment** to display the web console YAML editor. Update the YAML as follows and then click **Create**.

**NOTE**

You can copy the YAML from the ~/DO280/labs/console-workloads/deployment.yaml file on workstation.

kind: Deployment

apiVersion: apps/v1

metadata:

name: books

namespace: console-apps

spec:

replicas: 1

selector:

matchLabels:

app: books

template:

metadata:

labels:

app: books

spec:

containers:

- name: books

image: 'quay.io/redhattraining/books:v0.9'

ports:

- containerPort: 8080

protocol: TCP

readinessProbe:

httpGet:

path: /healthz

port: 8080

env:

- name: DB\_HOST

value: database.console-apps.svc.cluster.local

- name: DB\_PORT

value: '5432'

- name: DB\_USER

value: postgres

- name: DB\_PASSWORD

value: postgres

- name: DB\_NAME

value: postgres

**IMPORTANT**

Do not expect the pods to run successfully after completing this step. You will troubleshoot the deployment issue later in this exercise.

* 1. Click **Networking** → **Services**, and then click **Create Service** to display the web console YAML editor. Update the YAML as follows and then click **Create**.

**NOTE**

You can copy the YAML from the ~/DO280/labs/console-workloads/service.yaml file on workstation.

kind: Service

apiVersion: v1

metadata:

name: books

namespace: console-apps

spec:

selector:

app: books

ports:

- protocol: TCP

port: 8080

targetPort: 8080

* 1. Click **Networking** → **Routes**, and then click **Create Route**. Complete the page as follows, leaving the other fields unchanged, and then click **Create**.

**Table 8.4. Create Route Form**

| **Field** | **Value** |
| --- | --- |
| **Name** | books |
| **Service** | books |
| **Target Port** | 8080 → 8080 (TCP) |

1. Troubleshoot and fix the deployment issue.
   1. Click **Home** → **Events**, and notice the error events. Messages such as Failed to pull image "quay.io/redhattraining/books:v0.9" and Error: ImagePullBackOff indicate an issue with the image name or image tag.

|  |
| --- |
|  |

* 1. Click **Workloads** → **Deployments**, and then click the books deployment. Scroll to the bottom of the page to inspect the **Conditions** table. Notice that the Available condition type displays a False status.

|  |
| --- |
|  |

* 1. Click the **Pods** tab at the top of the **Deployment Details** screen and locate the pod status. It displays ImagePullBackOff.
  2. Click the **YAML** tab at the top of the **Deployment Details** page to navigate to the YAML editor and fix the issue. Update the spec image value to 'quay.io/redhattraining/books:v1.4' and then click **Save**.

**NOTE**

When OpenShift updates a deployment resource while you are attempting to update it, the YAML editor will not allow you to save your changes without fetching the latest version first. If this happens, click **Reload**, perform the edit again, and then click **Save**.

* 1. Click the **Details** tab at the top of the **Deployment Details** page, and monitor the pod deployment. Unfortunately, the pod still fails to start.
  2. Click **Home** → **Events**, and search for evidence of additional problems. A new event message indicates a quota problem.

Error creating: pods "books-5c65dc95-z9bss" is forbidden: exceeded quota: quota, requested: limits.memory=5Gi, used: limit.memory=10752Mi, limited: limits.memory=12Gi

Updating the books deployment created a new replica set, but scheduling a pod from the new replica set would exceed the project quota for memory limits.

* 1. To solve this problem, identify the books replica set with an existing pod and delete it. Deleting the replica set with the failing pod reduces quota usage and allows scheduling the pod from the new replica set. Click **Workloads** → **Replica Sets**.

It is expected that there are two replica sets for the books deployment. The books replica set with a status of 1 of 1 pods specifies the wrong container image version. Delete that replica set using the vertical ellipsis menu for the row and selecting **Delete Replica Set**. Confirm the deletion by clicking **Delete**.

|  |
| --- |
|  |

* 1. Click **Workloads** → **Deployments**, and then click the link for the books deployment. Wait until the donut indicates that one pod is running.
  2. Click **Networking** → **Routes**, and then click the link in the Location column. Firefox will open a new tab rendering a list of books that were fetched from the database.
  3. You will continue to use the new console-apps project and books deployment in the next section, so you do not need to delete them.

**Finish**

On the workstation machine, use the lab command to complete this exercise.

**[student@workstation ~]$ lab console-workloads finish**