Lab: Managing a Cluster with the Web Console

In this lab, you will manage the OpenShift cluster using the web console.

**Outcomes**

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

* Modify a secret to add htpasswd entries for new users.
* Configure a new project with role-based access controls and resource quotas.
* Use an OperatorHub operator to deploy a database.
* Create a deployment, service, and route for a web application.
* Troubleshoot an application using events and logs.

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 a directory for the exercise files.

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

1. Log in to the OpenShift web console as the admin user.
   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.

**NOTE**

If prompted with an untrusted certificate message, click **Add Exception** and then click **Confirm Security Exception**.

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

[HIDE SOLUTION](https://rol.redhat.com/rol/app/)

1. Add htpasswd entries to the localusers secret for users named dba and tester using redhat as the password.
   1. In the Red Hat OpenShift Container Platform web UI, click **Workloads** → **Secrets**, and then select openshift-config from the **Project** search list to display the secrets for the openshift-config project.
   2. Scroll to the bottom of the page and click the **localusers** link to display the localusers **Secret Details**.
   3. Click **Actions** → **Edit Secret** at the top of the page to navigate to the **Edit Key/Value Secret** tool.
   4. Use the workstation terminal to generate an encrypted htpasswd entry for both users.
   5. **[student@workstation ~]$ htpasswd -n -b dba redhat**
   6. dba:$apr1$***YF4aCK.9$qhoOTHlWTC.cLByNEHDaV***
   7. **[student@workstation ~]$ htpasswd -n -b tester redhat**
   8. tester:$apr1$***XdTSqET7$i0hkC5bIs7PhYUm2KhiI.0***
   9. Append the terminal output from the htpasswd commands to the htpasswd value in the OpenShift web console's secrets editor and then click **Save**.
   10. admin:$apr1$Au9.fFr$0k5wvUBd3eeBt0baa77.dae
   11. leader:$apr1$/abo4Hybn7a.tG5ZoOBn.QWefXckiy1
   12. developer:$apr1$RjqTY4cv$xql3.BQfg42moSxwnTNkh.
   13. **dba:$apr1$*YF4aCK.9$qhoOTHlWTC.cLByNEHDaV***
   14. **tester:$apr1$*XdTSqET7$i0hkC5bIs7PhYUm2KhiI.0***

1. [HIDE SOLUTION](https://rol.redhat.com/rol/app/)
2. Create a new app-team group that contains the developer and dba users.
   1. Click **User Management** → **Groups**, and then click **Create Group**. Use the YAML editor to define a Group resource as follows:
   2. apiVersion: user.openshift.io/v1
   3. kind: Group
   4. metadata:
   5. name: **app-team**
   6. users:
   7. - **developer**

- **dba**

Click **Create** to add the new app-team group.

1. [HIDE SOLUTION](https://rol.redhat.com/rol/app/)
2. Create a new console-review project with a view role binding for the tester user and an edit role binding for the app-team group. Set a resource quota that limits the project to four pods.
   1. Click **Home** → **Projects** to view the **Projects** page, and then click **Create Project**. Type console-review in the **Name** field, and then provide an optional **Display Name** and **Description**. Click **Create**.
   2. Click **Role Bindings** and then click **Create Binding**. Complete the form as follows to create a namespaced Role Binding for the app-team group.

**Table 8.5. App Team Role Binding Form**

| **Field** | **Value** |
| --- | --- |
| **Name** | app-team |
| **Namespace** | console-review |
| **Role Name** | edit |
| **Subject** | Group |
| **Subject Name** | app-team |

Click **Create** to create the namespaced RoleBinding.

* 1. Click the Role Bindings link to return to the **Role Bindings** page, and then click **Create Binding**. Complete the form as follows to create a namespaced Role Binding for the tester user.

**Table 8.6. Tester Role Binding Form**

| **Field** | **Value** |
| --- | --- |
| **Binding Type** | Namespace Role Binding (RoleBinding) |
| **Name** | tester |
| **Namespace** | console-review |
| **Role Name** | view |
| **Subject** | User |
| **Subject Name** | tester |

Click **Create** to create the namespaced RoleBinding.

* 1. Click **Administration** → **Resource Quotas**, and then click **Create Resource Quota**. Modify the YAML document to specify a limit of four pods as follows:
  2. apiVersion: v1
  3. kind: ResourceQuota
  4. metadata:
  5. name: **quota**
  6. namespace: console-review
  7. spec:
  8. hard:

pods: **'4'**

Remove the CPU and memory requests and limits, and then click **Create**.

1. [HIDE SOLUTION](https://rol.redhat.com/rol/app/)
2. Install the certified version of the Cockroach Operator for use in the console-review namespace.
   1. Click **Operators** → **OperatorHub**, and then click **Database** to display the list of database operators available from OperatorHub.
   2. Filter the **Provider Type** to display only **Certified** operators, and then click **Cockroach Operator**.
   3. Click **Install** on the **Cockroach Operator** page. On the **Install Operator** page, select console-review under **Installed Namespace**, and then click **Install**.

[HIDE SOLUTION](https://rol.redhat.com/rol/app/)

1. Create a RoleBinding that allows the dba user to view resources in the openshift-operators project.
   1. Click **User Management** → **Role Bindings**, and then click **Create Binding**. Fill out the form as follows.

**Table 8.7. DBA OpenShift-Operators Role Binding Form**

| **Field** | **Value** |
| --- | --- |
| **Binding Type** | Namespace Role Binding (RoleBinding) |
| **Name** | dba |
| **Namespace** | openshift-operators |
| **Role Name** | view |
| **Subject** | User |
| **Subject Name** | dba |

Click **Create** to add the namespaced RoleBinding.

1. [HIDE SOLUTION](https://rol.redhat.com/rol/app/)
2. As the dba user, deploy a CockroachDB Cluster database instance into the console-review project using the OpenShift web console. You must modify the CrdbCluster resource to disable TLS. From the **YAML View**, ensure the tlsEnabled setting has a value of false.
   1. Click **admin** → **Log out**, and then log in as the dba user with the password redhat.
   2. Click **Home** → **Projects**, and click the **console-review** project link to switch to the console-review project.
   3. Click **Operators** → **Installed Operators**, and then click the Cockroach Operator name.
   4. From the **Operator Details** page, click **Create Instance**. Use the **YAML View** to configure the CrdbCluster resource. Use the name crdb-example, and ensure the tlsEnabled setting has a value of false. Click **Create** to create the CrdbCluster resource.
   5. apiVersion: crdb.cockroachlabs.com/v1alpha1
   6. kind: CrdbCluster
   7. metadata:
   8. **name: crdb-example**
   9. namespace: console-review
   10. spec:
   11. **tlsEnabled: false**
   12. nodes: 3
   13. dataStore:
   14. pvc:
   15. spec:
   16. accessModes:
   17. - ReadWriteOnce
   18. resources:
   19. requests:
   20. storage: 10Gi

volumeMode: Filesystem

1. [HIDE SOLUTION](https://rol.redhat.com/rol/app/)
2. As the developer user, create a deployment, service, and route in the console-review project with issues that you will troubleshoot in the next step. Use the quay.io/redhattraining/exoplanets:v1.0 image, and name all of the new resources exoplanets. When correctly configured, the exoplanets application connects to the CockroachDB cluster and displays a list of planets located outside of our solar system.

**NOTE**

You can copy the deployment and service YAML resources from ~/DO280/labs/console-review/ on the workstation machine.

Specify the following environment variables in the deployment:

**Table 8.8. Deployment Environment Variables**

| **Name** | **Value** |
| --- | --- |
| DB\_HOST | localhost |
| DB\_PORT | '26257' |
| DB\_USER | root |
| DB\_NAME | postgres |

**IMPORTANT**

You will troubleshoot issues with the deployment in the next step.

* 1. Click **dba** → **Log out**, and then log in as the developer user with the password of developer.
  2. Click **Home** → **Projects**, and then click the **console-review** project to switch to the console-review project.
  3. Click **Workloads** → **Deployments**, and then click **Create Deployment** to display the web console YAML editor. Update the YAML as follows and then click **Create**:
  4. kind: Deployment
  5. apiVersion: apps/v1
  6. metadata:
  7. name: exoplanets
  8. namespace: console-review
  9. spec:
  10. replicas: 1
  11. selector:
  12. matchLabels:
  13. app: exoplanets
  14. template:
  15. metadata:
  16. labels:
  17. app: exoplanets
  18. spec:
  19. containers:
  20. - name: exoplanets
  21. image: 'quay.io/redhattraining/exoplanets:v1.0'
  22. ports:
  23. - containerPort: 8080
  24. protocol: TCP
  25. readinessProbe:
  26. httpGet:
  27. path: /healthz
  28. port: 8080
  29. env:
  30. - name: DB\_HOST
  31. value: localhost
  32. - name: DB\_PORT
  33. value: '26257'
  34. - name: DB\_USER
  35. value: root
  36. - name: DB\_NAME

value: postgres

* 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**:
  2. kind: Service
  3. apiVersion: v1
  4. metadata:
  5. name: exoplanets
  6. namespace: console-review
  7. spec:
  8. selector:
  9. app: exoplanets
  10. ports:
  11. - protocol: TCP
  12. port: 8080

targetPort: 8080

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

**Table 8.9. Create Route Form**

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

[HIDE SOLUTION](https://rol.redhat.com/rol/app/)

1. Troubleshoot and fix the deployment issues.
   1. Click **developer** → **Log out**, and then log in as the admin user with the password redhat.
   2. Click **Home** → **Events**, and then select console-review from the project list filter at the top. Notice the exoplanets quota error:

(combined from similar events): Error creating: pods "exoplanets-***5f88574546-lsnmx***" is forbidden: exceeded quota: quota, requested: pods=1, used: pods=4, limited: pods=4

* 1. Click **Administration** → **Resource Quotas**, and then select console-review from the **Project** filter list.
  2. Click the quota link in the list of resource quotas, and then click the YAML tab. Modify the spec to specify a limit of six pods as follows, and then click **Save**.
  3. kind: ResourceQuota
  4. apiVersion: v1
  5. metadata:
  6. name: quota
  7. namespace: console-review
  8. *...output omitted...*
  9. spec:
  10. hard:
  11. **pods: '6'**

*...output omitted...*

**NOTE**

The project requires a pod for the exoplanet's specified replica and an additional pod in order to roll out a change.

* 1. Click **Workloads** → **Pods**, and review the list of pods. The exoplanets pod may take a minute or two to appear on the list and display a CrashLoopBackOff status.
  2. Click the pod name, and then click the **Logs** tab. Notice the connection error.
  3. 2020/09/25 13:33:25 Connecting to database with: **host=localhost port=26257** user=root password= dbname=postgres sslmode=disable

2020/09/25 13:33:25 dial tcp [::1]:26257: connect: connection refused

* 1. Click **Networking** → **Services** to view the services in the console-review namespace. There are two services related to the Cockroach database. The service name with the -public suffix is the only Cockroach service with an IP address, and the service uses port 26257.

The exoplanets pod attempts to connect to port 26257 on the localhost instead of port 26257 on the database server. Copy the name of the service with the -public suffix to the clipboard.

* 1. Click **Workloads** → **Deployments**, and then click the exoplanets Deployment link. Click the **Environment** tab and change the DB\_HOST value from localhost to the name of the Cockroach service that you previously copied. Click **Save**.
  2. Click the **Pods** tab to verify that the new exoplanets pod updates to a Running status.

[HIDE SOLUTION](https://rol.redhat.com/rol/app/)

1. Navigate to the exoplanets website in a browser and observe the working application.
   1. Click **Networking** → **Routes**, click the exoplanets route name, and then click the link in the Location column. Firefox will open a new tab rendering a table of exoplanets.

[HIDE SOLUTION](https://rol.redhat.com/rol/app/)

**Evaluation**

As the student user on the workstation machine, use the lab command to grade your work. Correct any reported failures and rerun the command until successful.

**[student@workstation ~]$ lab console-review grade**

**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 console-review finish**

This concludes the section.