Guided Exercise: Update Application Image and Settings

Update the manifests of a database and a web application while minimizing interruption of service to their users.

Outcomes

You should be able to pause, update, and resume a deployment, and roll back a failing application.

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

This command ensures that all resources are available for this exercise. It also creates the `updates-rollout-db` project and deploys a MySQL database in that project. It creates the `updates-rollout-web` project and then deploys a web application with 10 replicas.

The command creates the `/home/student/DO180/labs/updates-rollout/resources.txt` file. The `resources.txt` file contains the name of the images and some commands that you use during the exercise. You can use the file to copy and paste these image names and commands.

```
[student@workstation ~]$ lab start updates-rollout
```

Instructions

1. Log in to the OpenShift cluster as the `developer` user with the `developer` password. Use the `updates-rollout-db` project.

   Log in to the OpenShift cluster.

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

   Set the `updates-rollout-db` project as the active project.

   ```
   [student@workstation ~]$ oc project updates-rollout-db
   ...output omitted...
   ```

2. Review the resources that the `lab` command created. Confirm that you can connect to the database. The MySQL database uses ephemeral storage.

   List the `Deployment` object and confirm that the pod is available. Retrieve the name of the container. You use that information when you update the container image in another step.

   ```
   [student@workstation ~]$ oc get deployment -o wide
   NAME   READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS ...
   mydb   1/1     0            1           17m   mysql-80   ...
   ```

   List the pods and confirm that the pod is running. The name of the pod on your system probably differs.

   ```
   [student@workstation ~]$ oc get pod
   NAME                   READY   STATUS    RESTARTS   AGE
   mydb-5c79866d48-5xzkk   1/1     Running   0          18m
   ```

   Retrieve the name of the image that the pod is using. The pod is using the `rhel9/mysql-80` image version `1-224`. Replace the pod name with your own from the previous step.

   ```
   [student@workstation ~]$ oc get pod mydb-5c79866d48-5xzkk \
     -o jsonpath='{.spec.containers[0].image}{"\n"}'
   registry.ocp4.example.com:8443/rhel9/mysql-80:1-224
   ```

   The classroom setup copied that image from the Red Hat Ecosystem Catalog. The original image is `registry.redhat.io/rhel9/mysql-80`.

   Confirm that you can connect to the database system by listing the available databases. Run the `mysql` command from inside the pod and connect as the `operator1` user by using `test` as the password.

```
[student@workstation ~]$ oc rsh mydb-5c79866d48-5xzkk \
  mysql --user=operator1 --password=test -e "SHOW DATABASES"
mysql: [Warning] Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| quotes             |
+--------------------+
```

3. You must implement several updates to the `Deployment` object. Pause the deployment to prevent OpenShift from rolling out the application for each modification that you make. After you pause the deployment, change the password for the `operator1` database user, update the container image, and then resume the deployment.

   Pause the `mydb` deployment.

   ```
   [student@workstation ~]$ oc rollout pause deployment/mydb
   deployment.apps/mydb paused
   ```

   Change the password of the `operator1` database user to `redhat123`. To change the password, update the `MYSQL_PASSWORD` environment variable in the pod template of the `Deployment` object.

   ```
   [student@workstation ~]$ oc set env deployment/mydb MYSQL_PASSWORD=redhat123
   deployment.apps/mydb updated
   ```

   Because the `Deployment` object is paused, confirm that the new password is not yet active. To do so, rerun the `mysql` command by using the current password. The database connection succeeds.

   ```
   [student@workstation ~]$ oc rsh mydb-5c79866d48-5xzkk \
     mysql --user=operator1 --password=test -e "SHOW DATABASES"
   mysql: [Warning] Using a password on the command line interface can be insecure.
   +--------------------+
   | Database           |
   +--------------------+
   | information_schema |
   | performance_schema |
   | quotes             |
   +--------------------+
   ```

   Update the MySQL container image to the `1-228` version.

   ```
   [student@workstation ~]$ oc set image deployment/mydb \
     mysql-80=registry.ocp4.example.com:8443/rhel9/mysql-80:1-228
   deployment.apps/mydb image updated
   ```

   Because the `Deployment` object is paused, confirm that the pod still uses the `1-224` image version.

   ```
   [student@workstation ~]$ oc get pod mydb-5c79866d48-5xzkk \
     -o jsonpath='{.spec.containers[0].image}{"\n"}'
   registry.ocp4.example.com:8443/rhel9/mysql-80:1-224
   ```

   Resume the `mydb` deployment.

   ```
   [student@workstation ~]$ oc rollout resume deployment/mydb
   deployment.apps/mydb resumed
   ```

   Confirm that the new rollout completes by waiting for the new pod to be running. The name of the pod on your system probably differs.

   ```
   [student@workstation ~]$ oc get pods
   NAME                   READY   STATUS    RESTARTS   AGE
   mydb-dd5dcbddb-rmf85   1/1     Running   0          2m2s
   ```

4. Verify that OpenShift applied all your modifications to the `Deployment` object.

   Retrieve the name of the image that the new pod is using. In the following command, use the name of the new pod as a parameter to the `oc get pod` command. The pod is now using the `1-228` image version.

```
[student@workstation ~]$ oc get pod mydb-dd5dcbddb-rmf85 \
  -o jsonpath='{.spec.containers[0].image}{"\n"}'
registry.ocp4.example.com:8443/rhel9/mysql-80:1-228
```

Confirm that you can connect to the database system by using the new password, `redhat123`, for the `operator1` database user.

```
[student@workstation ~]$ oc rsh mydb-dd5dcbddb-rmf85 \
  mysql --user=operator1 --password=redhat123 -e "SHOW DATABASES"
mysql: [Warning] Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| quotes             |
+--------------------+
```

5. In the second part of the exercise, you perform a rolling update of a replicated web application. Use the `updates-rollout-web` project and review the resources that the `lab` command created.

   Set the `updates-rollout-web` project as the active project.

   ```
   [student@workstation ~]$ oc project updates-rollout-web
   ...output omitted...
   ```

   List the `Deployment` object and confirm that the pods are available. Retrieve the name of the containers. You use that information when you update the container image in another step.

   ```
   [student@workstation ~]$ oc get deployment -o wide
   NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS      ...
   version   10/10   10           10          32m   versioned-hello ...
   ```

   List the `ReplicaSet` objects. Because OpenShift did not yet perform rolling updates, only one `ReplicaSet` object exists. The name of the `ReplicaSet` object on your system probably differs.

   ```
   [student@workstation ~]$ oc get replicaset
   NAME                 DESIRED   CURRENT   READY   AGE
   version-7bfff6b5b4   10        10        10      11m
   ```

   Retrieve the name and version of the image that the `ReplicaSet` object uses to deploy the pods. The pods are using the `redhattraining/versioned-hello` image version `v1.0`.

   ```
   [student@workstation ~]$ oc get replicaset version-7bfff6b5b4 \
     -o jsonpath='{.spec.template.spec.containers[0].image}{"\n"}'
   registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0
   ```

   Confirm that the `version` deployment includes a readiness probe. The probe performs an HTTP GET request on port 8080.

   ```
   [student@workstation ~]$ oc get deployment version \
     -o jsonpath='{.spec.template.spec.containers[0].readinessProbe}' | jq .
   {
     "failureThreshold": 3,
     "httpGet": {
       "path": "/",
       "port": 8080,
       "scheme": "HTTP"
     },
     "initialDelaySeconds": 3,
     "periodSeconds": 10,
     "successThreshold": 1,
     "timeoutSeconds": 1
   }
   ```

6. To watch the rolling update that you cause in a following step, open a new terminal window and then run the `~/DO180/labs/updates-rollout/curl_loop.sh` script that the `lab` command prepared. The script sends web requests to the application in a loop.

   Open a new terminal.

   Run the `/home/student/DO180/labs/updates-rollout/curl_loop.sh` script. Leave the script running and do not interrupt it.

```
[student@workstation ~]$ /home/student/DO180/labs/updates-rollout/curl_loop.sh
Hi!
Hi!
Hi!
Hi!
...output omitted...
```

7. Change the container image of the `version` deployment. The new application version creates a web page with a different message.

   Switch back to the first terminal window, and then use the `oc set image` command to update the deployment.

```
[student@workstation ~]$ oc set image deployment/version \
versioned-hello=registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.1
deployment.apps/version image updated
```

   Changing the image caused a rolling update. Watch the output of the `curl_loop.sh` script in the second terminal. If the output of the `curl_loop.sh` is not updating, then press **Ctrl**+**c** to stop the script in the second terminal. Then, restart the script.

   Before the update, only pods that run the `v1.0` version of the application reply. During the rolling updates, both old and new pods are responding. After the update, only pods that run the `v1.1` version of the application reply. The following output probably differs on your system.

```
...output omitted...
Hi!
Hi!
Hi!
Hi!
Hi! v1.1
Hi! v1.1
Hi!
Hi! v1.1
Hi!
Hi! v1.1
Hi! v1.1
Hi! v1.1
Hi! v1.1
...output omitted...
```

   Do not stop the script.

8. Confirm that the rollout process is successful. List the `ReplicaSet` objects and verify that the new object uses the new image version.

   Use the `oc rollout status` command to confirm that the rollout process is successful.

```
[student@workstation ~]$ oc rollout status deployment/version
deployment "version" successfully rolled out
```

   List the `ReplicaSet` objects. The initial object scaled down to zero pods. The new `ReplicaSet` object scaled up to 10 pods. The names of the `ReplicaSet` objects on your system probably differ.

```
[student@workstation ~]$ oc get replicaset
NAME                 DESIRED   CURRENT   READY   AGE
version-7bfff6b5b4   0         0         0       28m
version-b7fddfc8c    10        10        10      3m40s
```

   Retrieve the name and version of the image that the new `ReplicaSet` object uses. This image provides the new version of the application.

```
[student@workstation ~]$ oc get replicaset version-b7fddfc8c \
  -o jsonpath='{.spec.template.spec.containers[0].image}{"\n"}'
registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.1
```

9. Roll back the `version` deployment.

   Use the `oc rollout undo` command to roll back to the initial application version. Ignore the warning message.

```
[student@workstation ~]$ oc rollout undo deployment/version
deployment.apps/version rolled back
```

Watch the output of the `curl_loop.sh` script in the second terminal. The pods that run the `v1.0` version of the application are responding again. The following output probably differs on your system.

```
...output omitted...
Hi! v1.1
Hi! v1.1
Hi! v1.1
Hi! v1.1
Hi!
Hi! v1.1
Hi!
Hi! v1.1
Hi! v1.1
Hi!
Hi!
Hi!
...output omitted...
```

Press **Ctrl**+**C** to quit the script. Close that second terminal when done.

List the `ReplicaSet` objects. The initial object scaled up to 10 pods. The object for the new application version scaled down to zero pods.

```
[student@workstation ~]$ oc get replicaset
NAME                 DESIRED   CURRENT   READY   AGE
version-7bfff6b5b4   10        10        10      52m
version-b7fddfc8c    0         0         0       27m
```

Finish

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

```
[student@workstation ~]$ lab finish updates-rollout
```