Guided Exercise: Multi-pod Applications

Deploy a multi-pod application into Red Hat OpenShift Container Platform (RHOCP).

Outcomes

You should be able to use the `oc` command-line utility to:

- Create RHOCP deployments.
- Configure networking.
- Expose applications for external access.

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

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

Instructions

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

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

2. Create a `Deployment` resource for the Gitea application.

   Use the `oc create deployment` command to create the `Gitea` deployment.

   Configure the deployment port `3030` and use the `registry.ocp4.example.com:8443/redhattraining/podman-gitea:latest` container image.

   ```
   [student@workstation ~]$ oc create deployment gitea --port 3030 \
     --image=registry.ocp4.example.com:8443/redhattraining/podman-gitea:latest
   deployment.apps/gitea created
   ```

   Verify the pod status.

   ```
   [student@workstation ~]$ oc get po
   NAME                       READY   STATUS    RESTARTS   AGE
   gitea-6d446c7b48-b89hz     1/1     Running   0          75s
   ```

   If the pod is in the `ContainerCreating` status, then repeat the preceding command after a few seconds.

   The preceding command uses the `po` short name to refer to the `pod` resource.

3. Create a `Deployment` resource for a PostgreSQL database called `gitea-postgres`. The Gitea application uses the database to store data.

   You can use the completed file in the `/home/student/DO188/solutions/openshift-multipod` directory.

   Use the `oc create deployment` command to create the `gitea-postgres` database.

   Configure the deployment port `5432` and use the `registry.ocp4.example.com:8443/rhel9/postgresql-13:1` container image.

   Use the `--dry-run=client` and `-o yaml` options to generate a YAML file, and redirect the output to the `postgres.yaml` file.

   ```
   [student@workstation ~]$ oc create deployment gitea-postgres --port 5432 -o yaml \
     --image=registry.ocp4.example.com:8443/rhel9/postgresql-13:1 \
     --dry-run=client > postgres.yaml
   no output expected
   ```

The preceding command uses output redirection (>) to create the `postgres.yaml` file.

Open the `postgres.yaml` file in an editor, such as `gedit`, and add the following environment variables:

```
...file omitted...
      containers:
      - image: registry.ocp4.example.com:8443/rhel9/postgresql-13:1
        name: postgresql-13
        ports:
        - containerPort: 5432
        env:
        - name: POSTGRESQL_USER
          value: gitea
        - name: POSTGRESQL_PASSWORD
          value: gitea
        - name: POSTGRESQL_DATABASE
          value: gitea
...file omitted...
```

To get more information about the `env` property, execute the `oc explain deployment.spec.template.spec.containers.env` command.

WARNING

The YAML format is white-space sensitive. You must use correct indentation.

In the preceding example, the `env:` object indentation is 8 spaces.

Use the `oc create` command to create the deployment.

```
[student@workstation ~]$ oc create -f postgres.yaml
deployment.apps/gitea-postgres created
```

If you receive an error, compare your `postgres.yaml` file to the complete `/home/student/DO188/solutions/openshift-multipod/postgres.yaml` file.

Verify the pod status.

```
[student@workstation ~]$ oc get po
NAME                              READY   STATUS    RESTARTS   AGE
gitea-6d446c7b48-rf578            1/1     Running   0          6m18s
gitea-postgres-86d47f494d-7rl5g   1/1     Running   0          18s
```

4. Configure networking for the database.

   Expose the PostgreSQL database on the `gitea-postgres` hostname.

```
[student@workstation ~]$ oc expose deployment gitea-postgres
service/gitea-postgres exposed
```

   Verify that the PostgreSQL service uses the `gitea-postgres` name and the `5432` port.

```
[student@workstation ~]$ oc get svc
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
gitea-postgres   ClusterIP   172.30.124.174   <none>        5432/TCP   50s
```

5. Configure networking for the Gitea application.

   Expose the application within the RHOCP cluster.

```
[student@workstation ~]$ oc expose deployment gitea
service/gitea exposed
```

   Expose the `gitea` service for external access.

```
[student@workstation ~]$ oc expose service gitea
route.route.openshift.io/gitea exposed
```

6. Test your application functionality.

Verify the external URL for your application.

```
[student@workstation ~]$ oc get route
NAME     HOST/PORT                                  SERVICES    PORT    WILDCARD
gitea    gitea-ocp-multipod.apps.ocp4.example.com   gitea       3030    None
```

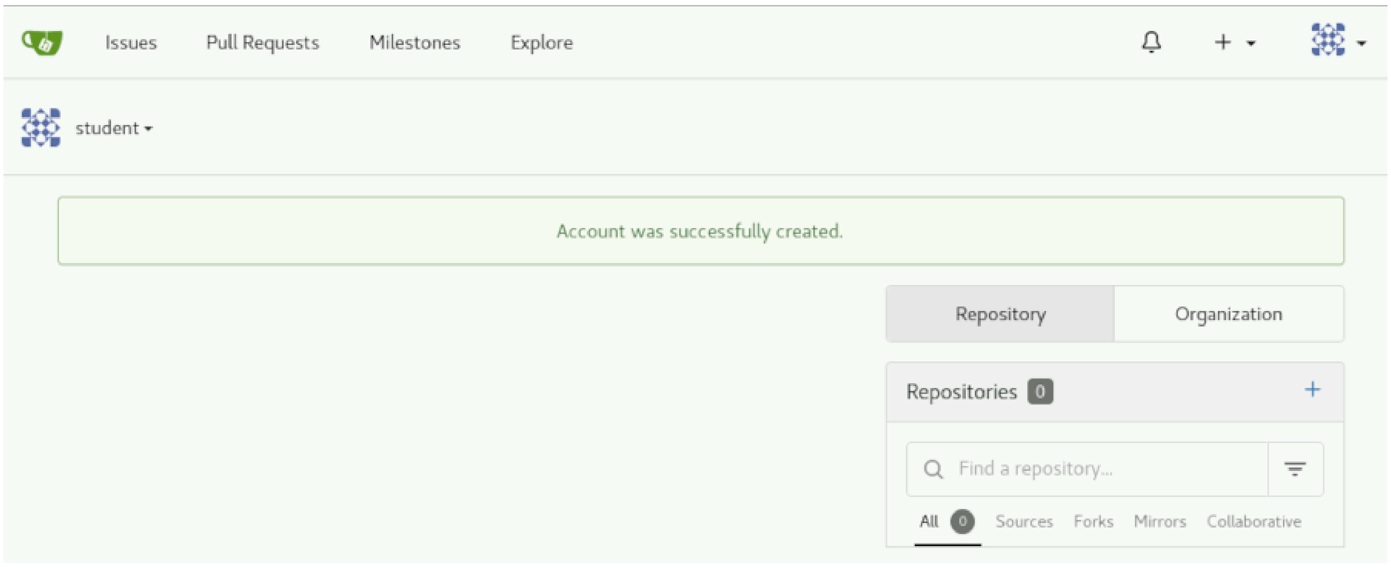
In a web browser, open the `gitea-ocp-multipod.apps.ocp4.example.com` URL and use the following configuration:

- **Database type**: `PostgreSQL`
- **Host**: `gitea-postgres:5432`
- **Username**: `gitea`
- **Password**: `gitea`
- **Database name**: `gitea`
- **Server Domain**: `gitea-ocp-multipod.apps.ocp4.example.com`
- **Gitea Base URL**: `http://gitea-ocp-multipod.apps.ocp4.example.com`

Then, click **Install Gitea**.

This step is successful if you see the login page.

Optionally, click **Register** to create a user and log in.



Account was successfully created.

Figure 8.6: A running Gitea instance when logged in

Finish

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-multipod
```