- Display the Bitnami MySQL Helm Chart on this page: https://bitnami.com/stack/mysql/helm
- Make sure you checkout the `main` branch and pull the latest changes:

```
git checkout main
git pull
git checkout -b public-oci
```

- In the `my-cluster` directory, create a new file called `bitnami-oci.yaml` and add the following:

```
apiVersion: source.toolkit.fluxcd.io/v1beta2
kind: HelmRepository
metadata:
  name: mysql
  namespace: default
spec:
  type: oci
  interval: 5m0s
  url: oci://registry-1.docker.io/bitnamicharts
```

- Create a new file called `mysql-release.yaml`:

```
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: mysql
  namespace: default
spec:
  interval: 5m
  chart:
    spec:
      chart: mysql
      version: '9.10.9'
      sourceRef:
        kind: HelmRepository
```

```
        name: mysql
        namespace: default
      interval: 1m
  values:
    auth:
      username: "myuser"
      password: "mypassword"
      database: "mydatabase"
```

- Create the MR:

```
git add -A
git commit -m 'adds the Bitnami OCI repository and the MySQL Helm release'
git push --set-upstream origin public-oci
```

- Approve and merge the MR.
- Reconcile Flux CD:

```
flux reconcile kustomization flux-system --with-source
```

- Make sure we have the correct resources:

```
kuebctl get helmrelease
kubectl get pods
```

- Try logging in to the database using the supplied credentials:

```
kubectl exec -it mysql-0 -- mysql -u myuser -pmypassword mydatabase
CTRL-D
```

- Create a new Helm chart:

```
cd ../../charts
helm create apache
cd apache
```

- Open the `Chart.yaml` file and change the `appVersion` to be `2.4.57`.
- Open the `values.yaml` file and change the image repository to be `httpd` instead of `nginx`.
- Create a Helm package by running:

```
helm package .
```

- Get the GitLab token by running `export` then pressing the UP arrow to get the last command that used it.
- Login to the GitLab registry by running:

```
helm registry login -u abohmeed@gmail.com registry.gitlab.com
```

- When asked for the password, we need to give it the GitLab token: `glpat-KA-8zde5ux1Yyny9yHVU` but you need to copy it from the previous command.
- Push the package to the GitLab registry using the following command:

```
helm push apache-0.1.0.tgz
oci://registry.gitlab.com/abohmeed/myfluxrepo
```

- Go to the Flux repository on GitLab to verify that the chart was saved: https://gitlab.com/abohmeed/myfluxrepo then Deploy then Container Registry.
- Checkout the main branch and pull the latest changes:

```
git checkout main
git pull
git checkout -b private-oci
```

- Display the contents of the Docker config file:

```
cat ~/.config/helm/registry/config.json
```

- Decode the `auth` string using the `base64 -d` command to show its original content.
- Type the following command to the screen but **do not** execute it:

```
kubectl create secret docker-registry gitlab-credentials \
    --docker-username=abohmeed@gmail.com \
    --docker-password=glpat-KA-8zde5ux1Yyny9yHVU \
    --docker-server=registry.gitlab.com \
    --namespace=default
```

- Convert the contents of the Docker config.json file to base64:

```
cat ~/.docker/config.json| base64 | tr -d "\n"
```

- Create a new file called `gitlab-oci-secret.yaml` and add the following:

```
apiVersion: v1
kind: Secret
metadata:
  name: gitlab-credentials
  namespace: default
data:
  .dockerconfigjson: # paste the output of the previous command
type: kubernetes.io/dockerconfigjson
```

- Create the Helm OCI repository file called `gitlab-oci-repository.yaml` and add the following:

```yaml
apiVersion: source.toolkit.fluxcd.io/v1beta2
kind: HelmRepository
metadata:
  name: gitlab
  namespace: default
spec:
  type: oci
  interval: 5m0s
  url: oci://registry.gitlab.com/abohmeed/myfluxrepo
  secretRef:
    name: gitlab-credentials
```

- Create the Helm release file, call it `apache-helm-release.yaml` and add the following:

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: apache
  namespace: default
spec:
  interval: 5m
  chart:
    spec:
      chart: apache
      version: '0.1.0'
      sourceRef:
        kind: HelmRepository
        name: gitlab
        namespace: default
      interval: 1m
```

- Create an MR:

```
git add -A
git commit -m "Adds a private OCI repository and an Apache Helm
release that uses it"
git push --set-upstream origin private-oci
```

- Approve and merge it in GitLab.
- Verify that we have the resources that we have created:

```
kubectl get helmrepository
kubectl get helmrelease
kubectl get pods
```