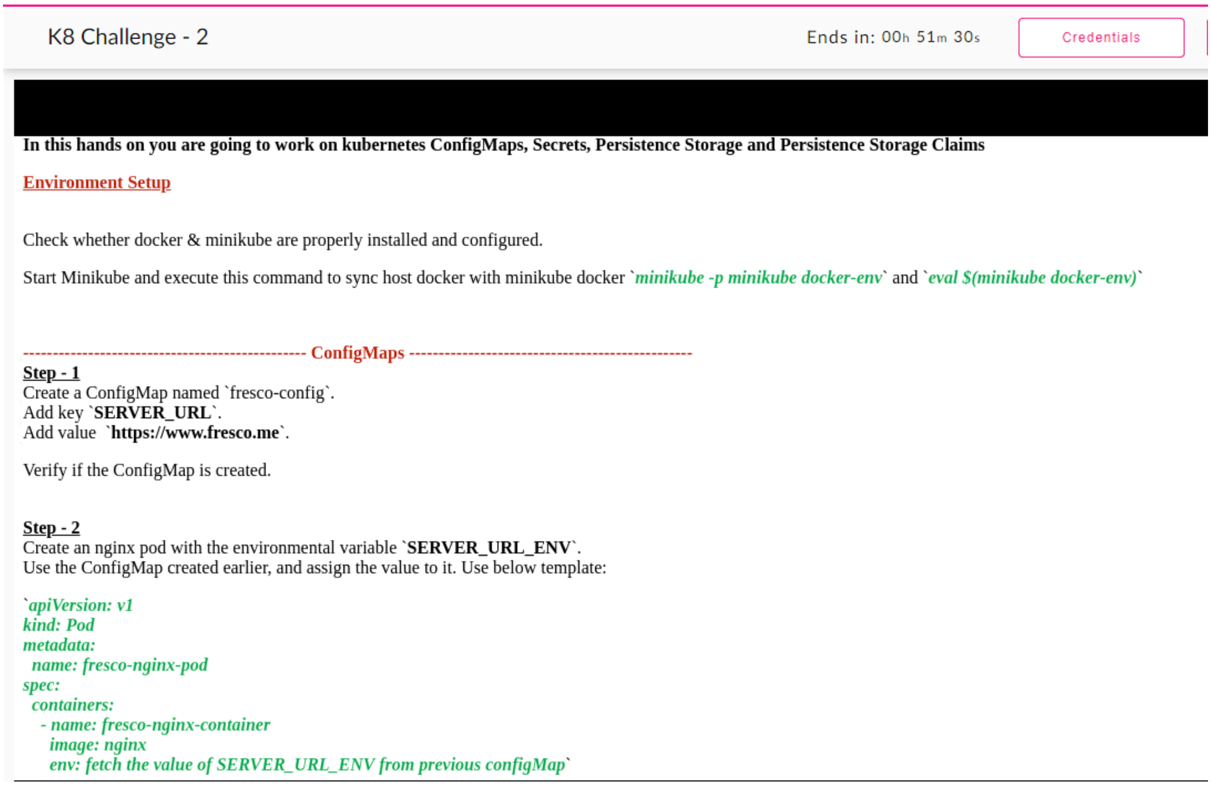
Challenge 2
Question



[redacted]

In this hands on you are going to work on kubernetes ConfigMaps, Secrets, Persistence Storage and Persistence Storage Claims

Environment Setup

Check whether docker & minikube are properly installed and configured.

Start Minikube and execute this command to sync host docker with minikube docker `*minikube -p minikube docker-env*` and `*eval $(minikube docker-env)*`

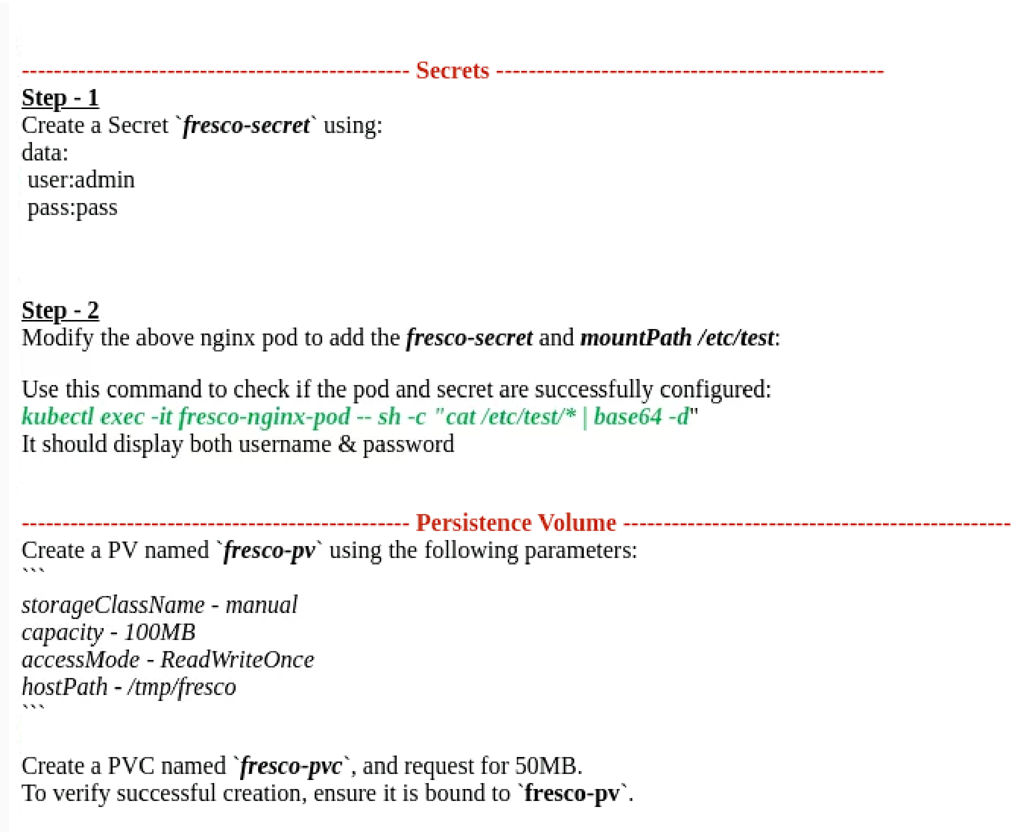
-- ConfigMaps --

Step - 1
Create a ConfigMap named `fresco-config`.
Add key `**SERVER_URL**`.
Add value `**https://www.fresco.me**`.

Verify if the ConfigMap is created.

Step - 2
Create an nginx pod with the environmental variable `**SERVER_URL_ENV**`.
Use the ConfigMap created earlier, and assign the value to it. Use below template:

```
`apiVersion: v1
kind: Pod
metadata:
  name: fresco-nginx-pod
spec:
  containers:
   - name: fresco-nginx-container
     image: nginx
     env: fetch the value of SERVER_URL_ENV from previous configMap`
```



-- Secrets --

Step - 1
Create a Secret `***fresco-secret***` using:
data:
 user:admin
 pass:pass

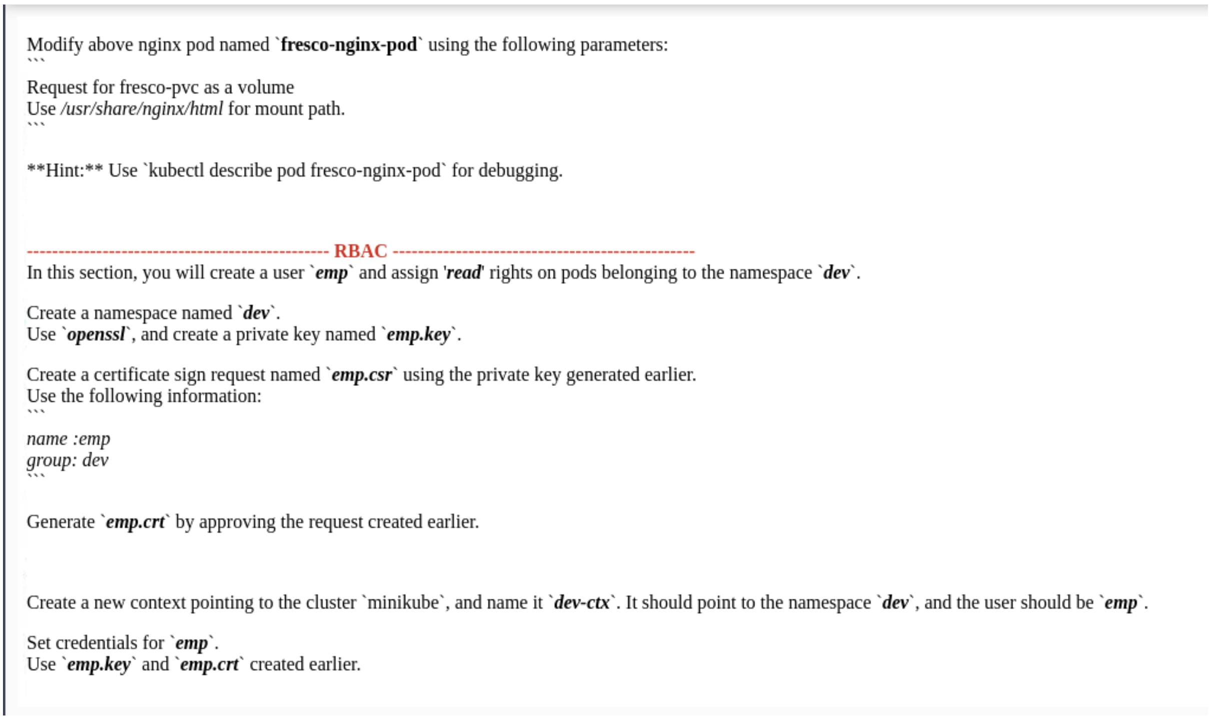
Step - 2
Modify the above nginx pod to add the ***fresco-secret*** and ***mountPath /etc/test***:

Use this command to check if the pod and secret are successfully configured:
kubectl exec -it fresco-nginx-pod -- sh -c "cat /etc/test/ | base64 -d*"
It should display both username & password

-- Persistence Volume --
Create a PV named `***fresco-pv***` using the following parameters:

```
storageClassName - manual
capacity - 100MB
accessMode - ReadWriteOnce
hostPath - /tmp/fresco
```

Create a PVC named `***fresco-pvc***`, and request for 50MB.
To verify successful creation, ensure it is bound to `**fresco-pv**`.



Modify above nginx pod named `**fresco-nginx-pod**` using the following parameters:
```
Request for fresco-pvc as a volume
Use */usr/share/nginx/html* for mount path.
```

Hint: Use `kubectl describe pod fresco-nginx-pod` for debugging.

--- **RBAC** ---

In this section, you will create a user `***emp***` and assign '***read***' rights on pods belonging to the namespace `***dev***`.

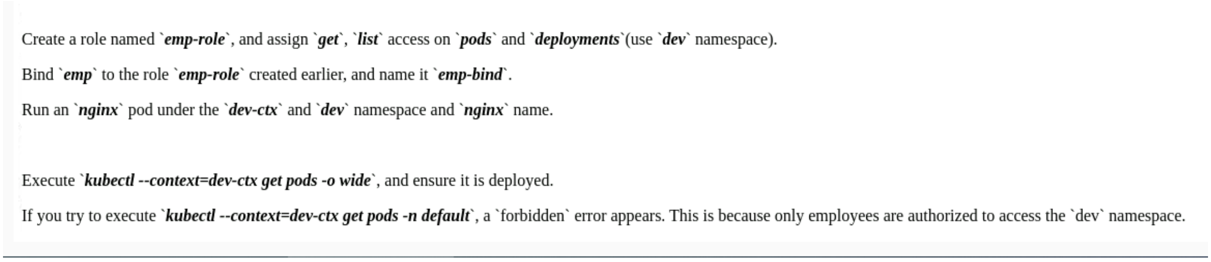
Create a namespace named `***dev***`.
Use `***openssl***`, and create a private key named `***emp.key***`.

Create a certificate sign request named `***emp.csr***` using the private key generated earlier.
Use the following information:
```
name :emp
group: dev
```

Generate `***emp.crt***` by approving the request created earlier.

Create a new context pointing to the cluster `minikube`, and name it `***dev-ctx***`. It should point to the namespace `***dev***`, and the user should be `***emp***`.

Set credentials for `***emp***`.
Use `***emp.key***` and `***emp.crt***` created earlier.



Create a role named `***emp-role***`, and assign `***get***`, `***list***` access on `***pods***` and `***deployments***`(use `***dev***` namespace).

Bind `***emp***` to the role `***emp-role***` created earlier, and name it `***emp-bind***`.

Run an `***nginx***` pod under the `***dev-ctx***` and `***dev***` namespace and `***nginx***` name.

Execute `***kubectl --context=dev-ctx get pods -o wide***`, and ensure it is deployed.

If you try to execute `***kubectl --context=dev-ctx get pods -n default***`, a `forbidden` error appears. This is because only employees are authorized to access the `dev` namespace.

Answer