Step - 1:

Create a ConfigMap named `fresco-config`:

```

**kubectl create configmap fresco-config --from-literal=SERVER\_URL=https://www.fresco.me**

```

Verify if the ConfigMap is created:

```

**kubectl get configmap fresco-config**

```

Step - 2:

Create an nginx pod with the environmental variable `SERVER\_URL\_ENV`:

**Write the below contents in pod.yaml and type: kubectl apply -f pod.yaml**

# pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: fresco-nginx-pod

spec:

containers:

- name: fresco-nginx-container

image: nginx

env:

- name: SERVER\_URL\_ENV

valueFrom:

configMapKeyRef:

name: fresco-config

key: SERVER\_URL

Test the configuration:

```

**kubectl exec -it fresco-nginx-pod -- sh -c "env | grep SERVER\_URL\_ENV"**

```

Secrets:

Step - 1:

Create a Secret `fresco-secret`:

```

**kubectl create secret generic fresco-secret --from-literal=user=admin --from-literal=pass=pass**

```

Step - 2:

Modify the nginx pod to add the `fresco-secret` and mountPath `/etc/test`:

**IMPORTANT STEP:**

**Before writing the following: you have to be careful**

**You have to first delete the pod using : kubectl delete -f pod.yaml**

**And then REPLACE the below content in pod.yaml by removing previous content.**

**and type: kubectl apply -f pod.yaml**

**# pod.yaml**

apiVersion: v1

kind: Pod

metadata:

name: fresco-nginx-pod

spec:

containers:

- name: fresco-nginx-container

image: nginx

env:

- name: SERVER\_URL\_ENV

valueFrom:

configMapKeyRef:

name: fresco-config

key: SERVER\_URL

volumeMounts:

- name: secret-volume

mountPath: "/etc/test"

volumes:

- name: secret-volume

secret:

secretName: fresco-secret

```

Check if the pod and secret are configured:

```

**kubectl exec -it fresco-nginx-pod -- sh -c "cat /etc/test/\* | base64 -d"**

**The above mostly probably won’t work saying invalid input 🡪 You can leave it no problem**

```

Persistence Volume:

Create a PV named `fresco-pv`:

**Write the below contents in pod.yaml and type: kubectl apply -f pv.yaml**

**# pv.yaml**

apiVersion: v1

kind: PersistentVolume

metadata:

name: fresco-pv

spec:

storageClassName: manual

capacity:

storage: 100Mi

accessModes:

- ReadWriteOnce

hostPath:

path: /tmp/fresco

```

Create a PVC named `fresco-pvc`:

**Write the below contents in pod.yaml and type: kubectl apply -f pvc.yaml**

**# pvc.yaml**

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: fresco-pvc

spec:

storageClassName: manual

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 50Mi

```

Check if the PVC is bound to the PV:

```

**kubectl get pvc fresco-pvc**

```

**IMPORTANT STEP:**

**Before writing the following: you have to be careful**

**You have to first delete the pod using : kubectl delete -f pod.yaml**

**And then REPLACE the below content in pod.yaml by removing previous content.**

**and type: kubectl apply -f pod.yaml**

**# pod.yaml**

Modify the nginx pod to use the PVC:

```

apiVersion: v1

kind: Pod

metadata:

name: fresco-nginx-pod

spec:

containers:

- name: fresco-nginx-container

image: nginx

env:

- name: SERVER\_URL\_ENV

valueFrom:

configMapKeyRef:

name: fresco-config

key: SERVER\_URL

volumeMounts:

- name: pvc-volume

mountPath: "/usr/share/nginx/html"

- name: secret-volume

mountPath: "/etc/test"

volumes:

- name: pvc-volume

persistentVolumeClaim:

claimName: fresco-pvc

- name: secret-volume

secret:

secretName: fresco-secret```

RBAC:

kubectl create namespace dev

openssl genrsa -out emp.key 2048

openssl req -new -key emp.key -out emp.csr -subj "/CN=emp/O=dev"

openssl x509 -req -in emp.csr -CA ~/.minikube/ca.crt -CAkey ~/.minikube/ca.key-CAcreateserial -out emp.crt -days 500

kubectl config set-context dev-ctx --cluster=minikube --namespace=dev --user=emp

kubectl config set-credentials emp --client-key=emp.key --client-certificate=emp.crt

kubectl create role emp-role --verb=get,list --resource=pods,deployments -n dev

# Bind role to user

kubectl create rolebinding emp-bind --role=emp-role --user=emp -n dev

# Create nginx pod under the dev-ctx and dev namespace

kubectl create pod --context=dev-ctx -n dev --image=nginx nginx