# Solution to kubernetes handson 1

## minikube start

copy paste from instruction
copy paste from instruction

```
kubectl run firstapp --image gcr.io/google-samples/kubernetes-bootcamp:v1 --port=8080
```

```
kubectl expose pod firstapp --type NodePort
```

open the folder /home/labuser
create a deployment.yaml file

now type in the following content(with correct indentation):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

save and goto terminal

```
kubectl apply -f deployment.yaml
```

---------

again goto /home/labuser
create a file called service.yaml and type the following:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
```

```
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - nodePort: 30080
      port: 80
```

```
kubectl apply -f service.yaml
```

```
kubectl get pods
```

now copy the name of the nginx pod and use it in the next command:

kubectl exec -it <pod-name> /bin/bash

[if not present previously] create this path(i.e all the folders that are not present and the files):
/usr/share/nginx/html/index.html
cd /usr/share/nginx/html
echo 'Welcome to fresco nginx pod' > index.html

-----------------------------------------------------------------------------------

# Solution to kubernetes handson 2

Start minikube:

```
minikube start
```

Do the environmental setup as given in the problem statement

**Configmaps:**

kubectl create configmap fresco-config --from-literal=SERVER_URL=https://www.fresco.me

Step-2
open folder /home/labuser and create a file named: fresco-nginx-pod.yaml and type:

```
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
```

-----------------------------------------------------------------------------------------

## Secrets

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

now goto the nginx pod yaml file and add:

```
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: fresco-secret
      mountPath: "/etc/test"
  volumes:
  - name: fresco-secret
    secret:
      secretName: fresco-secret
```

kubectl delete pods --all  
kubectl apply -f fresco-nginx-pod.yaml

-----------------------------------------------------------------------------------------

**persistentVolumes**

open the fresco-nginx-pod.yaml file and add:

```
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: fresco-secret
      mountPath: "/etc/test"
    - name: fresco-pvc
      mountPath: "/usr/share/nginx/html"
  volumes:
  - name: fresco-secret
    secret:
      secretName: fresco-secret
  - name: fresco-pvc
    persistentVolumeClaim:
      claimName: fresco-pvc
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: fresco-pv
spec:
  storageClassName: manual
  capacity:
    storage: 100M
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/tmp/fresco"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: fresco-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 50M
```

```
kubectl delete -f fresco-nginx-pod.yaml
kubectl apply -f fresco-nginx-pod.yaml
```