1. Where is the default kubeconfig file located in the current environment?
~/.kube/config

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJWURxa3hMOGtlOFF3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1Wl
hSbGN6QWVGdzB5TkRBM01UQXhPREUzTXpWYUZ3MHpOREEzTURneE9ESXlNelZhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUNYdjJYdDVm
QjE3VVc4NDhNWTZFV3NjZUtBa3FyOExWcmViT2Rjci9QVDFraG54amVaUk5FYm9zUlEKSWcwRnJFbktoTGVZbEhmRjQ0OGpEWk5raUxNVk85Zk1rRG5GRUtBR3BpSFYrUGJvblAwakJDSHFvNTZGUFh0YwpBOE5HeGNsNnRkWTBaUG
o3aGtYYnNMTk5ZcXRMcHpZTXlpUHZxcms2cFdxK2JsNk5jTFl3OE5jeXFOd3NyeEJqCmRNUG9aUGMxTDV3eUlmWGlrUUozSFhNWU1rdVRzYTNXNXdiUzRlVWJhVUFxT041c3FHMTgvTmpsYndrODFWZDYKbWtkalRXeU50aVJTN3lX
TzkyL2dXYWFLNHZiN1JsVEJ4eWRhd2oxdEUzVkRUN2NlT2JCbUo3WDkzRjJ3VlNMVQpFbStNcDNJc3hYUjlSVStVSlNQR3ArdmVOSlhqQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQU
RBUUgvTUIwR0ExVWREZ1FXQkJRWWlLalJidzBYTXZpaG43YlZXR3hRYXlOaGpUQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQTdiU0F5RUNuZwpIRWMrTEJHUHY3S0gzb2FX
MGZxVGFyLzcvMmwxU2hTSDhaWEZCd1llR2N3QmFjdFozMHBYTmdydWQvOXdKRmR6ClJMclV2RlNGNzcraC9BTEpvY09Nci9VZ1BvZmNCeWVXTkNoZlZYMDNVYVBKcHJVdENFbHJWUEZ6SlJuK2RKNGMKNDhTc2RFcTU4a1REQUF5Wj
F0NE5XZFhHQWdWek5LRUc1Vm05MzBLQzU0ZjFheWV6ZWVleHVBMHE5TVF0Wkx5UQozWE9yY2pFWmthaG9UeVJRalJCd3BVditESzFJV2FXakdpSGVUWUExYlVCbmVOS0pBUkVOaHlrSVpUZENUamlJClZrUWZML1RwbHk3aWtpSndt
ekROUFQraW5kc2lqZXFTNnY0amVkdFNVWm5qOEZqaDJmMGIyWFdXbVpuNWhQaEoKV0tEeXJjQlRSWXk2Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    server: https://192.168.1.10:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
```

2. How many clusters are defined in the default kubeconfig
one cluster : kuberentes .

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJWURxa3hMOGtlOFF3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1Wl
hSbGN6QWVGdzB5TkRBM01UQXhPREUzTXpWYUZ3MHpOREEzTURneE9ESXlNelZhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUNYdjJYdDVm
QjE3VVc4NDhNWTZFV3NjZUtBa3FyOExWcmViT2Rjci9QVDFraG54amVaUk5FYm9zUlEKSWcwRnJFbktoTGVZbEhmRjQ0OGpEWk5raUxNVk85Zk1rRG5GRUtBR3BpSFYrUGJvblAwakJDSHFvNTZGUFh0YwpBOE5HeGNsNnRkWTBaUG
o3aGtYYnNMTk5ZcXRMcHpZTXlpUHZxcms2cFdxK2JsNk5jTFl3OE5jeXFOd3NyeEJqCmRNUG9aUGMxTDV3eUlmWGlrUUozSFhNWU1rdVRzYTNXNXdiUzRlVWJhVUFxT041c3FHMTgvTmpsYndrODFWZDYKbWtkalRXeU50aVJTN3lX
TzkyL2dXYWFLNHZiN1JsVEJ4eWRhd2oxdEUzVkRUN2NlT2JCbUo3WDkzRjJ3VlNMVQpFbStNcDNJc3hYUjlSVStVSlNQR3ArdmVOSlhqQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQU
RBUUgvTUIwR0ExVWREZ1FXQkJRWWlLalJidzBYTXZpaG43YlZXR3hRYXlOaGpUQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQTdiU0F5RUNuZwpIRWMrTEJHUHY3S0gzb2FX
MGZxVGFyLzcvMmwxU2hTSDhaWEZCd1llR2N3QmFjdFozMHBYTmdydWQvOXdKRmR6ClJMclV2RlNGNzcraC9BTEpvY09Nci9VZ1BvZmNCeWVXTkNoZlZYMDNVYVBKcHJVdENFbHJWUEZ6SlJuK2RKNGMKNDhTc2RFcTU4a1REQUF5Wj
F0NE5XZFhHQWdWek5LRUc1Vm05MzBLQzU0ZjFheWV6ZWVleHVBMHE5TVF0Wkx5UQozWE9yY2pFWmthaG9UeVJRalJCd3BVditESzFJV2FXakdpSGVUWUExYlVCbmVOS0pBUkVOaHlrSVpUZENUamlJClZrUWZML1RwbHk3aWtpSndt
ekROUFQraW5kc2lqZXFTNnY0amVkdFNVWm5qOEZqaDJmMGIyWFdXbVpuNWhQaEoKV0tEeXJjQlRSWXk2Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    server: https://192.168.1.10:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
```

3. What is the user configured in the current context?
Kuberentes-admin .

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://192.168.1.12:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

4. Create a Persistent Volume with the given specification. ➔ Volume Name: pv-log ➔ Storage: 100Mi ➔ Access Modes: ReadWriteMany ➔ Host Path: /pv/log

```
io.k8s.api.core.v1.PersistentVolume (
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
spec:
  capacity:
    storage: 100Mi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /pv/log
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./pv.yml
persistentvolume/pv-log unchanged
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

5. Create a Persistent Volume Claim with the given specification. ➔ Volume Name: claim-log-1 ➔ Storage Request: 50Mi ➔ Access Modes: ReadWriteMany

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./pv-claim.yml
persistentvolumeclaim/claim-log-1 created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-log-1
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 50Mi
```

6. Create a webapp pod to use the persistent volume claim as its storage. ➔ Name: webapp ➔ Image Name: nginx ➔ Volume: PersistentVolumeClaim=claim-log-1 ➔ Volume Mount: /var/log/nginx

```yaml
io.k8s.api.core.v1.Pod (v1@pod.json)
apiVersion: v1
kind: Pod
metadata:
  name: webapp
spec:
  containers:
    - name: nginx
      image: nginx
      resources:
        limits:
          memory: "512Mi"
          cpu: "500m"
      volumeMounts:
        - mountPath: /var/log/nginx
          name: log-storage
  volumes:
    - name: log-storage
      persistentVolumeClaim:
        claimName: claim-log-1
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./webapp-pod.yml
pod/webapp configured
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get pods -o wide
NAME     READY   STATUS    RESTARTS   AGE    IP             NODE           NOMINATED NODE   READINESS GATES
webapp   1/1     Running   0          100s   10.244.1.104   managednode1   <none>           <none>
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

7. volume-share-datacenter pod.

volume-share-datacenter-pod.yml > {} spec > [ ] volumes > {} 0
io.k8s.api.core.v1.Pod (v1@pod.json)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: volume-share-datacenter
spec:
  containers:
    - name: volume-container-datacenter-1
      image: centos:latest
      command: ["/bin/bash", "-c", "sleep 10000"]
      resources:
        limits:
          memory: "128Mi"
          cpu: "500m"
      volumeMounts:
        - name: volume-share
          mountPath: /tmp/news
    # ------------------------------ SECOND CONTAINER :
    - name: volume-container-datacenter-2
      image: centos:latest
      command: ["/bin/bash", "-c", "sleep 10000"]
      resources:
        limits:
          memory: "128Mi"
          cpu: "500m"
      volumeMounts:
        - name: volume-share
          mountPath: /tmp/cluster
  volumes:
    - name: volume-share
      emptyDir: {}
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./volume-share-datacenter-pod.yml
pod/volume-share-datacenter created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get pods -o wide
NAME                    READY   STATUS         RESTARTS   AGE     IP             NODE           NOMINATED NODE   READINESS GATES
volume-share-datacenter   0/2     ErrImagePull   0          9s      10.244.1.106   managednode1   <none>           <none>
webapp                  1/1     Running        0          6m32s   10.244.1.104   managednode1   <none>           <none>
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

8. webserver pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webserver
spec:
  containers:
    - name: nginx-container
      image: nginx:latest
      resources:
        limits:
          memory: "256Mi"
          cpu: "500m"
      volumeMounts:
        - name: shared-logs
          mountPath: /var/log/nginx

    - name: sidecar-container
      image: ubuntu:latest
      resources:
        limits:
          memory: "128Mi"
          cpu: "250m"
      command:
        - "sh"
        - "-c"
        - "while true; do cat /var/log/nginx/access.log /var/log/nginx/error.log; sleep 30; done"
      volumeMounts:
        - name: shared-logs
          mountPath: /var/log/nginx

  volumes:
    - name: shared-logs
      emptyDir: {}
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./webserver-pod.yml
pod/webserver created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get pods -o wide
NAME        READY   STATUS              RESTARTS   AGE     IP             NODE           NOMINATED NODE   READINESS GATES
webapp      1/1     Running             0          9m49s   10.244.1.104   managednode1   <none>           <none>
webserver   0/2     ContainerCreating   0          6s      <none>         managednode1   <none>           <none>
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get pods -o wide
NAME        READY   STATUS              RESTARTS   AGE     IP             NODE           NOMINATED NODE   READINESS GATES
webapp      1/1     Running             0          9m52s   10.244.1.104   managednode1   <none>           <none>
webserver   0/2     ContainerCreating   0          9s      <none>         managednode1   <none>           <none>
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE     IP             NODE           NOMINATED NODE   READINESS GATES
webapp      1/1     Running   0          9m59s   10.244.1.104   managednode1   <none>           <none>
webserver   2/2     Running   0          16s     10.244.1.107   managednode1   <none>           <none>
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

9. Create a new service account with the name pvviewer. Grant this Service account access to list all PersistentVolumes in the cluster by creating an appropriate cluster role called pvviewer-role and ClusterRoleBinding called pvviewer-role-binding.

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: pvviewer
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: pvviewer-role
rules:
  - apiGroups: [""]
    resources: ["persistentvolumes"]
    verbs: ["list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: pvviewer-role-binding
subjects:
  - kind: ServiceAccount
    name: pvviewer
    namespace: default
roleRef:
  kind: ClusterRole
  name: pvviewer-role
  apiGroup: rbac.authorization.k8s.io
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./pv-viewer-service-account.yml
serviceaccount/pvviewer created
clusterrole.rbac.authorization.k8s.io/pvviewer-role created
clusterrolebinding.rbac.authorization.k8s.io/pvviewer-role-binding created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get persistentvolumes --as=system:serviceaccount:default:pvviewer
NAME     CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                 STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
pv-log   100Mi      RWX            Retain           Bound    default/claim-log-1                  <unset>                          22m
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

10. Create a ConfigMap named nginx-config

```yaml
# nginx-config.yml > { } spec > [ ] containers > { } 0 > image
# io.k8s.api.core.v1.Pod (v1@pod.json) | io.k8s.api.core.v1.ConfigMap (v1@configmap.json)
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
data:
  nginx.conf: |
    events {}
    http {
      server {
        listen 80;
        location / {
          return 200 'Hello from custom Nginx config!';
        }
      }
    }
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx
      image: nginx:latest
      resources:
        limits:
          memory: "128Mi"
          cpu: "500m"
        requests:
          memory: "64Mi"
          cpu: "250m"
      volumeMounts:
        - name: nginx-config-volume
          mountPath: /etc/nginx/nginx.conf
          subPath: nginx.conf
  volumes:
    - name: nginx-config-volume
      configMap:
        name: nginx-config
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./nginx-config.yml
configmap/nginx-config created
pod/nginx-pod created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl logs nginx-pod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl exect -it nginx-pod -- curl http://localhost
error: unknown command "exect" for "kubectl"

Did you mean this?
        exec
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl exec -it nginx-pod -- curl http://localhost
Hello from custom Nginx config!mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

Deploy HaProxy

1- namespace haproxy-controller-devops

```
io.k8s.api.core.v1.Namespace (v1@namespace.json)
apiVersion: v1
kind: Namespace
metadata:
  name: haproxy-controller-devops
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./haproxy-namepsace.yml
namespace/haproxy-controller-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

2- ServiceAccount haproxy-service-account-devops

```
io.k8s.api.core.v1.ServiceAccount (v1@serviceaccount.json)
apiVersion: v1
kind: ServiceAccount
metadata:
  name: haproxy-service-account-devops
  namespace: haproxy-controller-devops
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./haproxy-service-account.yml
serviceaccount/haproxy-service-account-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

3- ClusterRole haproxy-cluster-role-devops

```
io.k8s.api.rbac.v1.ClusterRole (v1@clusterrole.json)
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: haproxy-cluster-role-devops
rules:
  - apiGroups: [""]
    resources:
      - "configmaps"
      - "secrets"
      - "endpoints"
      - "nodes"
      - "pods"
      - "services"
      - "namespaces"
      - "events"
      - "serviceaccounts"
    verbs:
      - "get"
      - "list"
      - "watch"
      - "create"
      - "patch"
      - "update"
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl -f ./haproxy-cluster-role.yml
error: unknown shorthand flag: 'f' in -f
See 'kubectl --help' for usage.
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./haproxy-cluster-role.yml
clusterrole.rbac.authorization.k8s.io/haproxy-cluster-role-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./haproxy-cluster-role-binding.yml
clusterrolebinding.rbac.authorization.k8s.io/haproxy-cluster-role-binding-devops unchanged
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get clusterrolebinding haproxy-cluster-role-binding-devops -o wide
NAME                                  ROLE                                        AGE   USERS   GROUPS   SERVICEACCOUNTS
haproxy-cluster-role-binding-devops   ClusterRole/haproxy-cluster-role-devops   85s                    haproxy-controller-devops/haproxy-service-account-devops
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

5- backend deployment

```
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-deployment-devops
  namespace: haproxy-controller-devops
  labels:
    run: ingress-default-backend
spec:
  replicas: 1
  selector:
    matchLabels:
      run: ingress-default-backend
  template:
    metadata:
      labels:
        run: ingress-default-backend
    spec:
      containers:
        - name: backend-container-devops
          image: gcr.io/google_containers/defaultbackend:1.0
          ports:
            - containerPort: 8080
          resources:
            limits:
              memory: "128Mi"
              cpu: "500m"
            requests:
              memory: "64Mi"
              cpu: "250m"
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./backend-deployment.yml
deployment.apps/backend-deployment-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get deployment backend-deploymen-devops -n haproxy-controller-devops
Error from server (NotFound): deployments.apps "backend-deploymen-devops" not found
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get deployment backend-deployment-devops -n haproxy-controller-devops
NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment-devops   1/1     1            1           60s
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

6- service for backend

```yaml
apiVersion: v1
kind: Service
metadata:
  name: service-backend-devops
  namespace: haproxy-controller-devops
spec:
  ports:
    - name: port-backend
      port: 8080
      targetPort: 8080
  selector:
    run: ingress-default-backend
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./backend-service.yml
service/service-backend-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get svc -n haproxy-controller-devops
NAME                    TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
service-backend-devops   ClusterIP   10.97.228.23   <none>        8080/TCP   17s
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

deployment for frontend :

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: haproxy-ingress-devops
  namespace: haproxy-controller-devops
  labels:
    run: haproxy-ingress
spec:
  replicas: 1
  selector:
    matchLabels:
      run: haproxy-ingress
  template:
    metadata:
      labels:
        run: haproxy-ingress
    spec:
      serviceAccountName: haproxy-service-account-devops
      containers:
        - name: ingress-container-devops
          image: haproxytech/kubernetes-ingress
          args:
            - --default-backend-service=haproxy-controller-devops/service-backend-devops
          ports:
            - name: http
              containerPort: 80
            - name: https
              containerPort: 443
            - name: stat
              containerPort: 1024
          resources:
            requests:
              cpu: "500m"
              memory: "50Mi"
            limits:
              cpu: "1"
              memory: "100Mi"
          livenessProbe:
            httpGet:
              path: /health
              port: 1024
          env:
            - name: TZ
              value: "Etc/UTC"
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: POD_NAMESPACE
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./front-end-deployment.yml
deployment.apps/haproxy-ingress-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./front-end-deployment.yml
deployment.apps/haproxy-ingress-devops unchanged
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get deployment haproxy-ingress-devops -n haproxy-controller-devops
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
haproxy-ingress-devops   0/1     1            0           2m30s
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

front-end service :

```
io.k8s.api.core.v1.Service (v1@service.json)
apiVersion: v1
kind: Service
metadata:
  name: ingress-service-devops
  namespace: haproxy-controller-devops
spec:
  type: NodePort
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: 80
      nodePort: 32456
    - name: https
      port: 443
      protocol: TCP
      targetPort: 443
      nodePort: 32567
    - name: stat
      port: 1024
      protocol: TCP
      targetPort: 1024
      nodePort: 32678
  selector:
    run: haproxy
```

```
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl apply -f ./frontend-service.yml
service/ingress-service-devops created
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$ sudo kubectl get svc ingress-service-devops -n haproxy-controller-devops
NAME                     TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)                                  AGE
ingress-service-devops   NodePort   10.100.211.166   <none>        80:32456/TCP,443:32567/TCP,1024:32678/TCP   23s
mohamedharoon@mohamedharoon:~/Desktop/ITI/18-k8s/labs/lab5$
```

9- Access the proxy states