# Kubernetes

## Installation

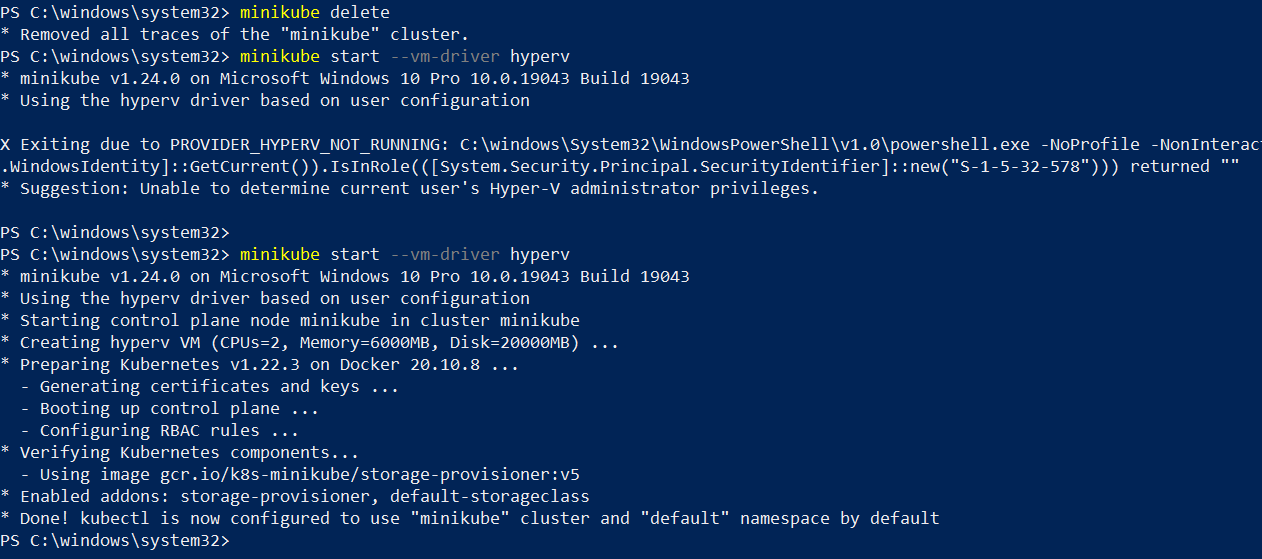
On windows- local- do install docker as well

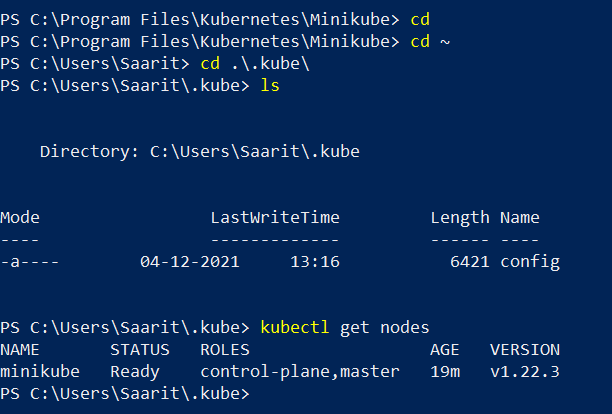
<https://minikube.sigs.k8s.io/docs/start/>

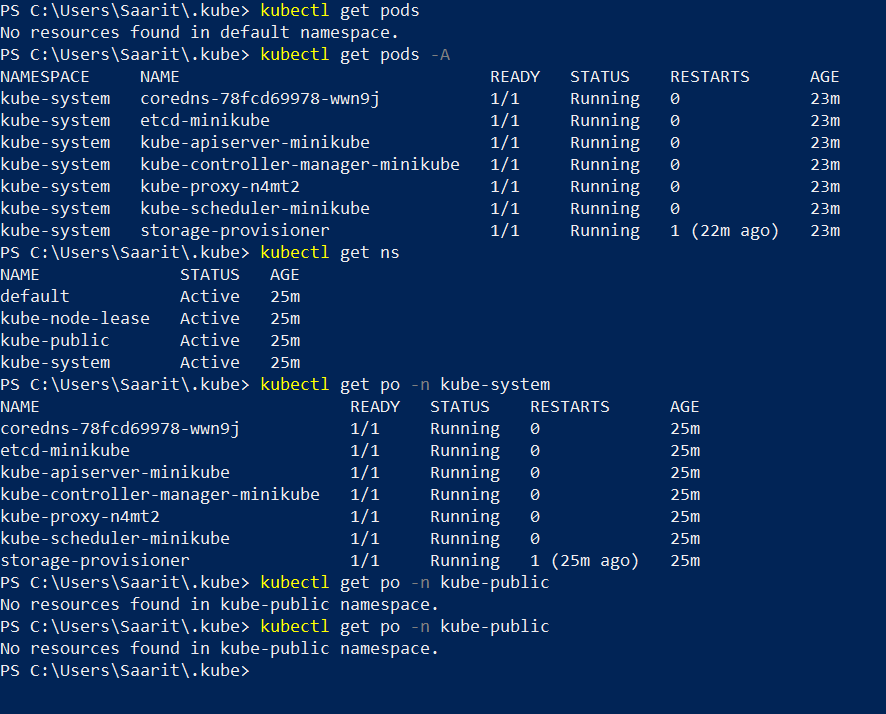
<https://github.com/kubernetes/minikube/issues/4783>

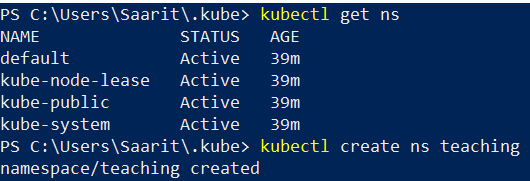
run as admin:

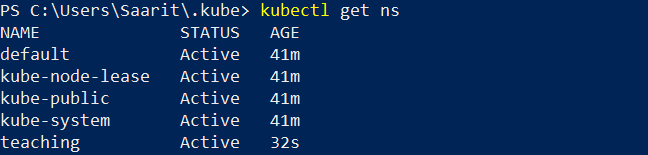
-minikube delete  
-minikube start --vm-driver hyperv



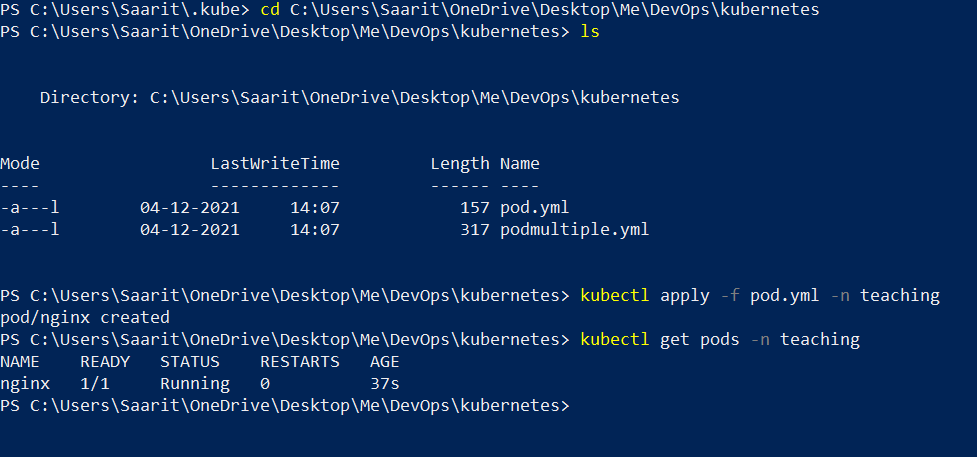


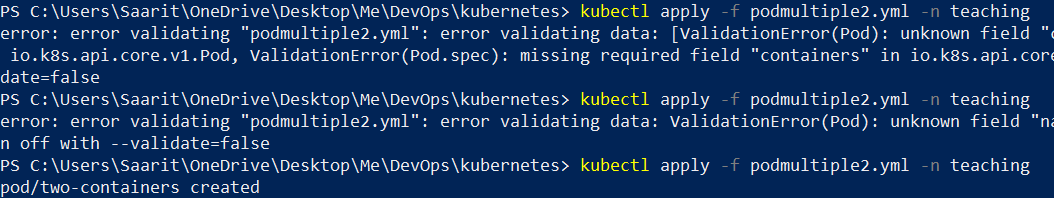


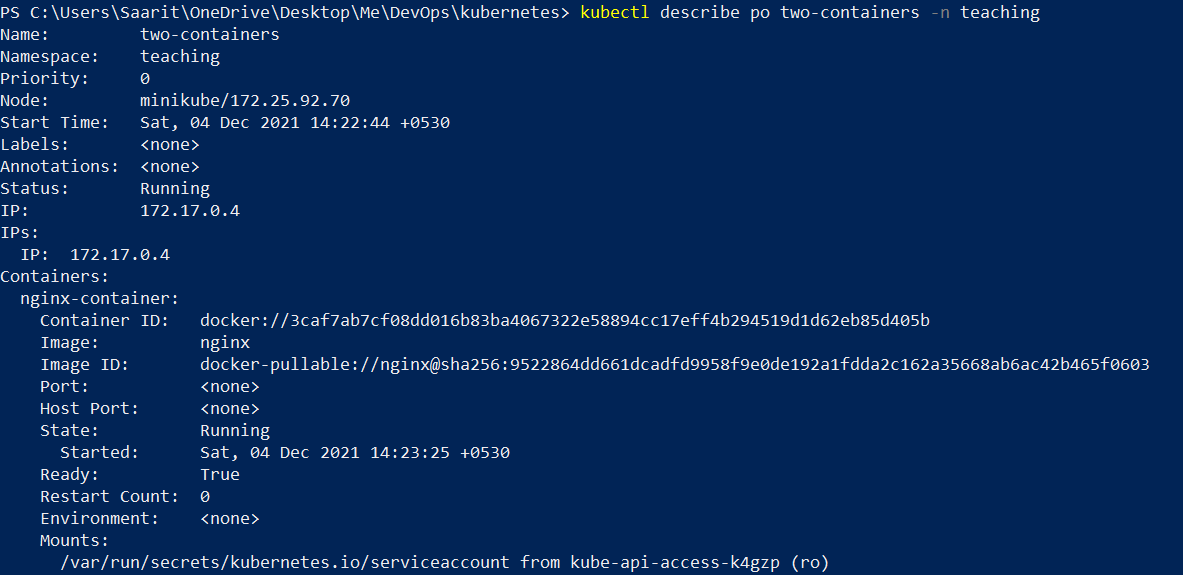




Metadata- data about data







Use of a faulty yml

apiVersion: v1

kind: Pod

metadata:

  name: two-containers-fault

spec:

  restartPolicy: Never

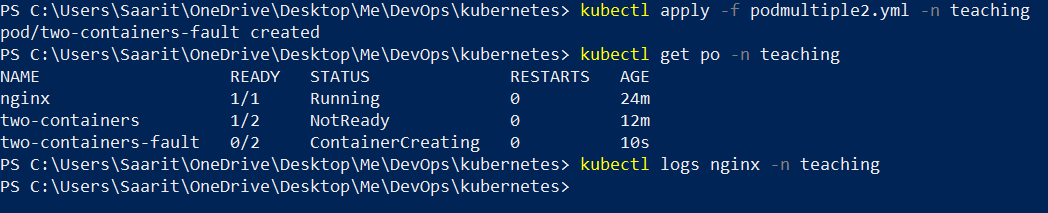
  containers:

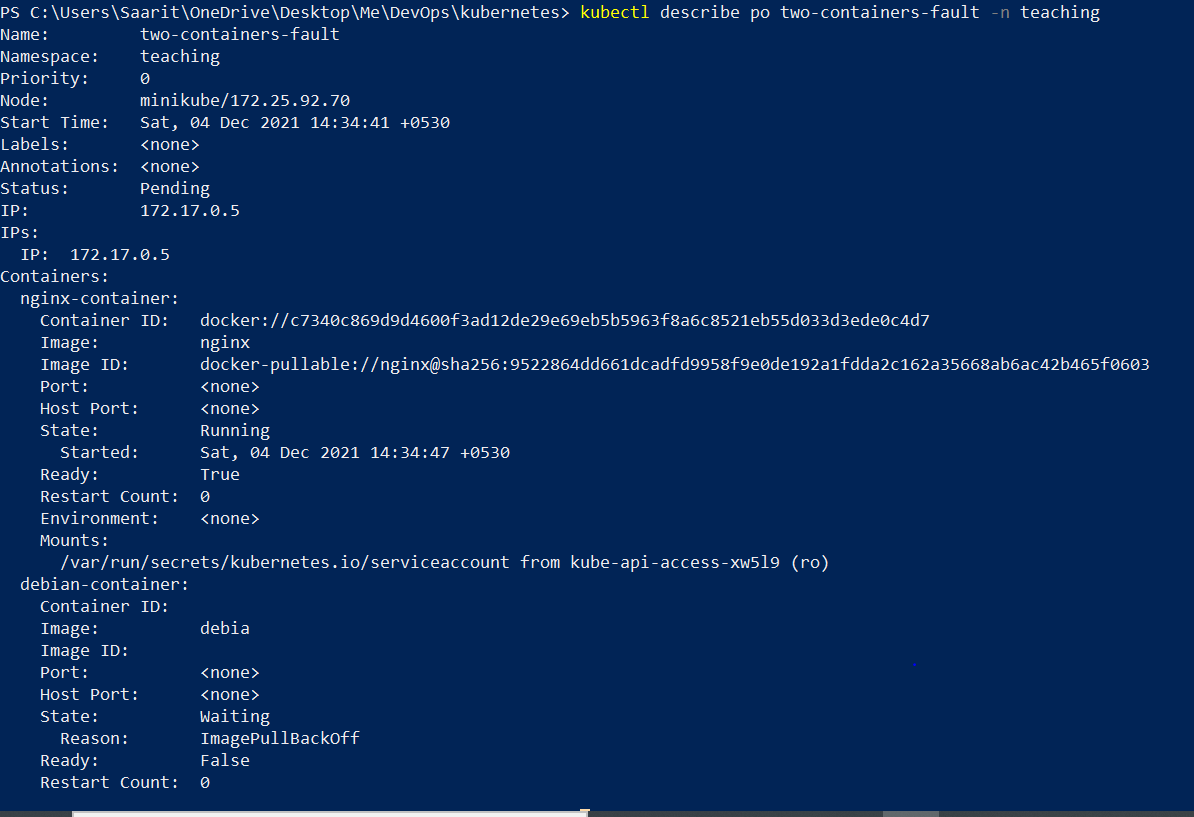
    - name: nginx-container

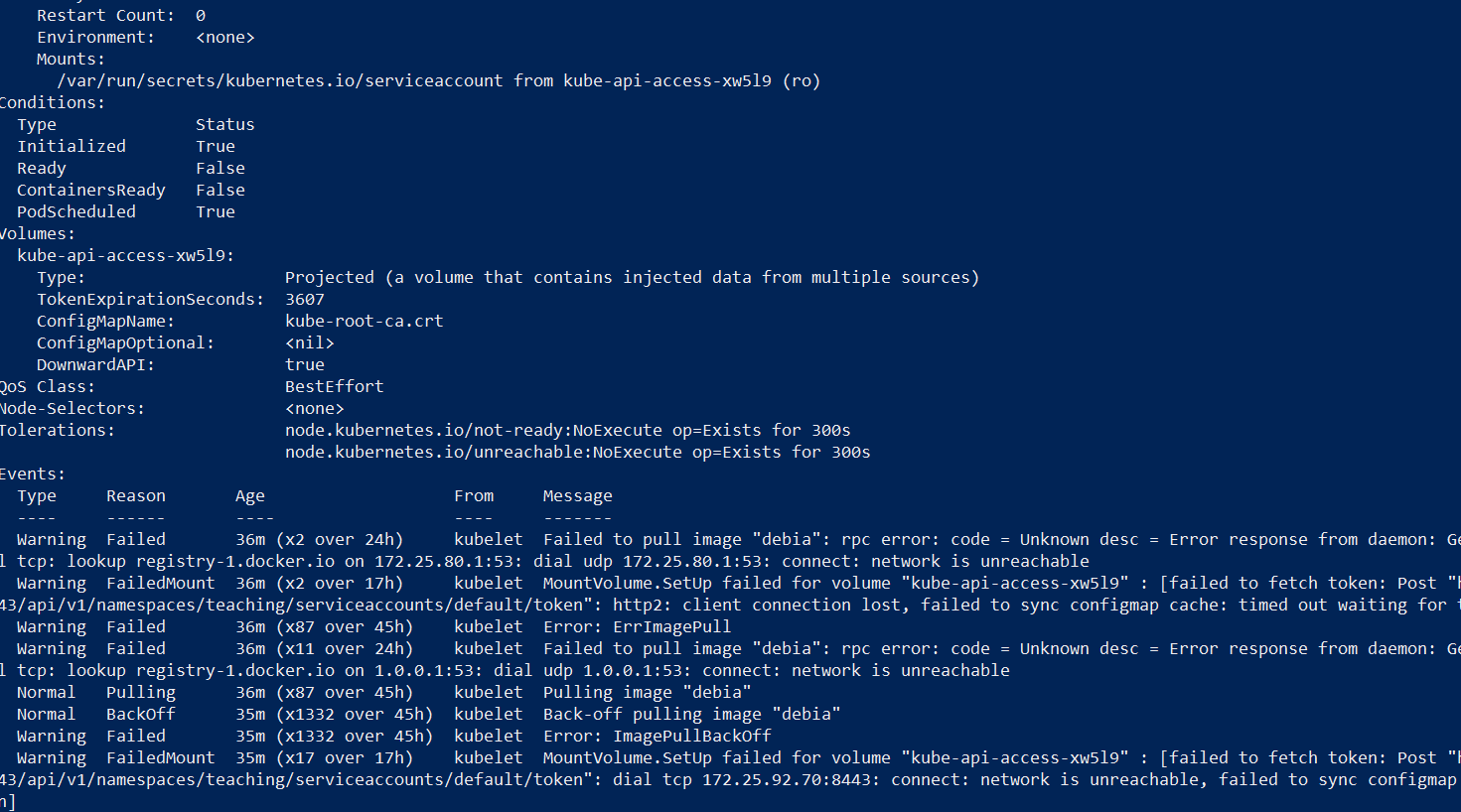
      image: nginx

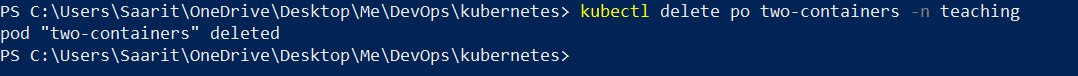
    - name: debian-container

      image: debia









apiVersion: v1

kind: Pod

metadata:

  name: two-containers-with-mysql

spec:

  restartPolicy: Never

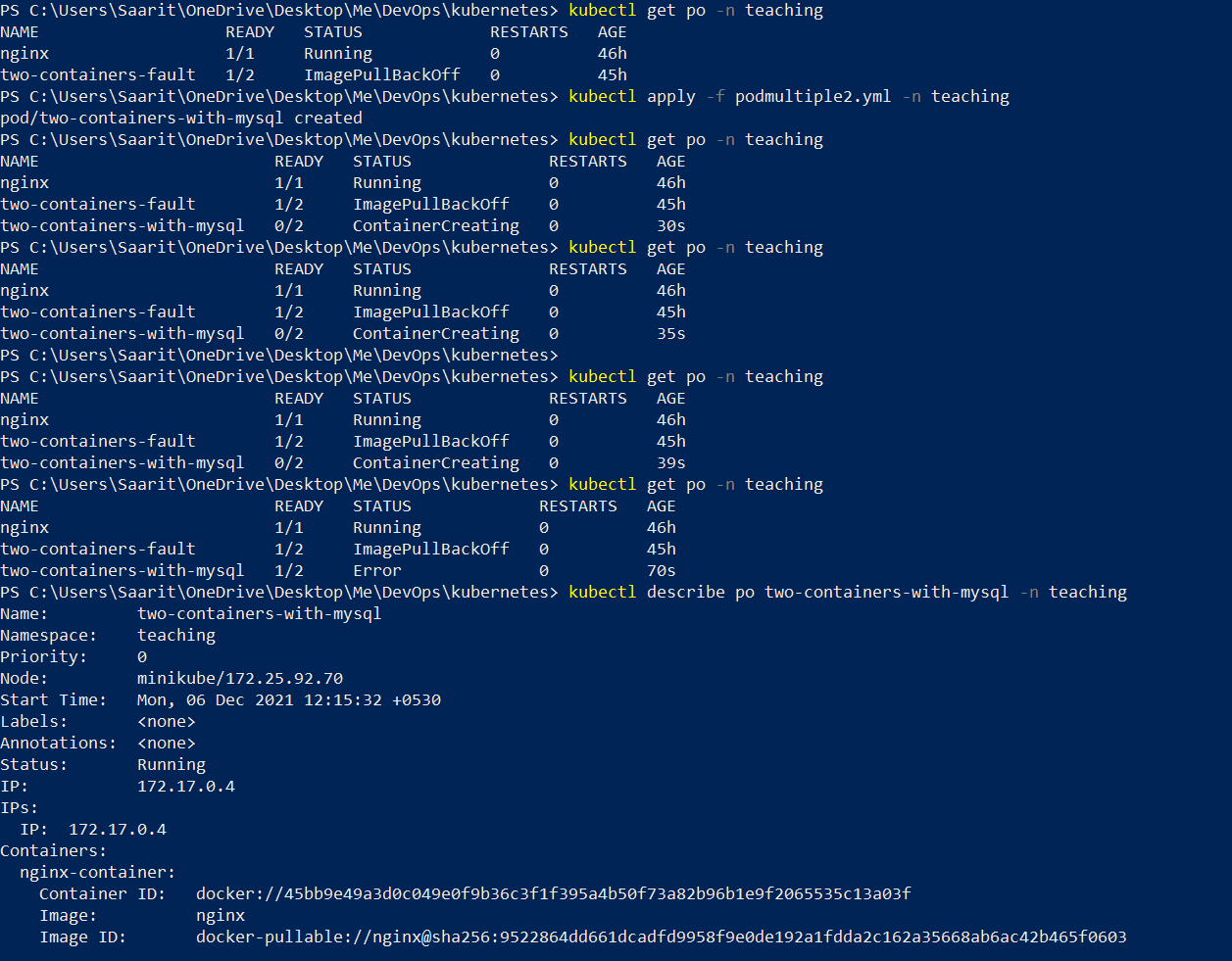
  containers:

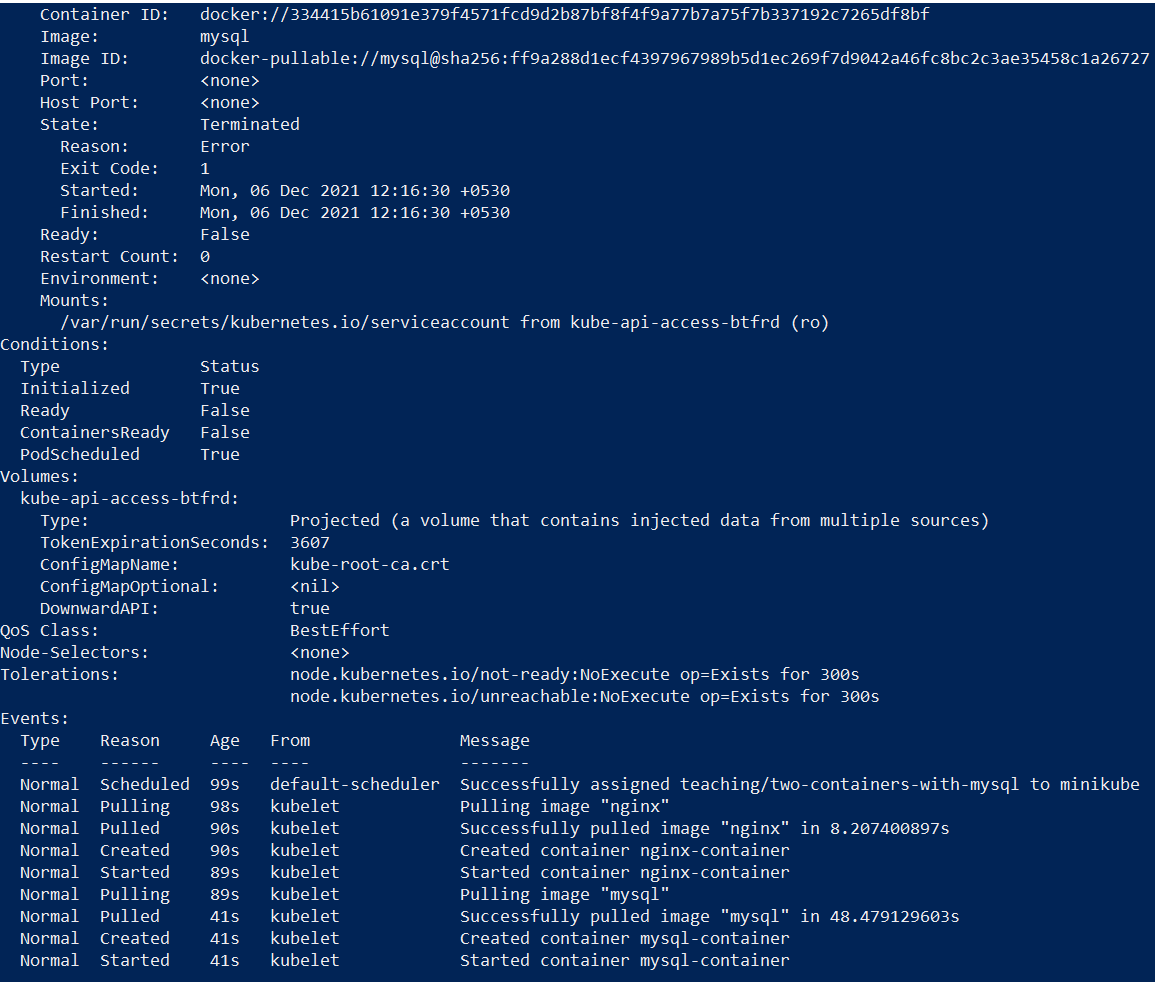
    - name: nginx-container

      image: nginx

    - name: mysql-container

      image: mysql





apiVersion: v1

kind: Pod

metadata:

  name: two-containers-with-debian

spec:

  restartPolicy: Never

  containers:

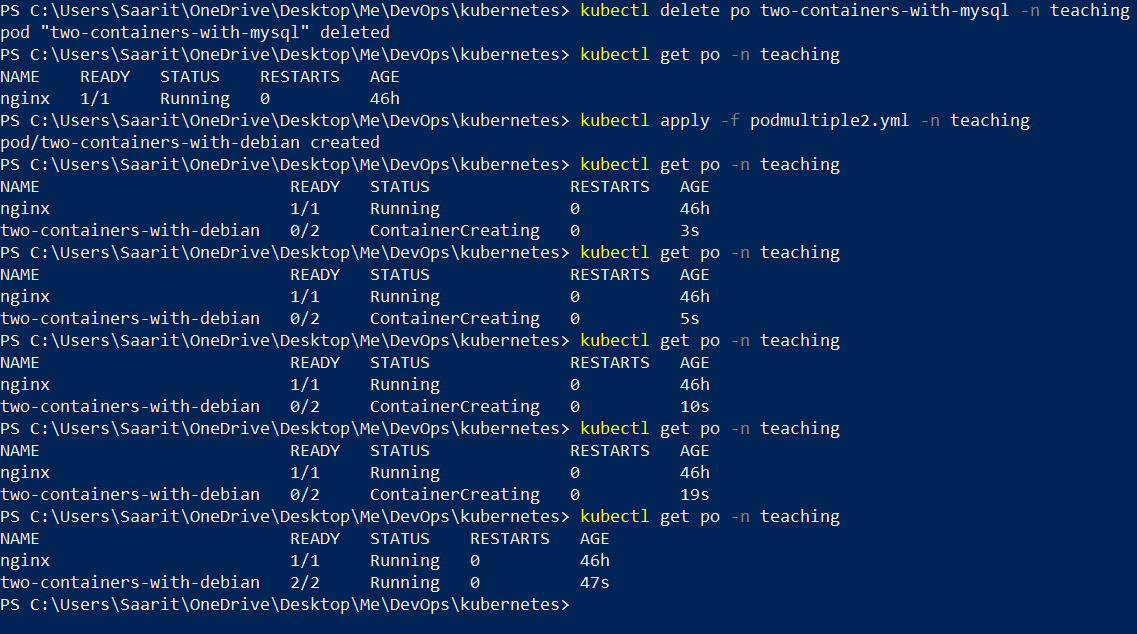
    - name: nginx-container

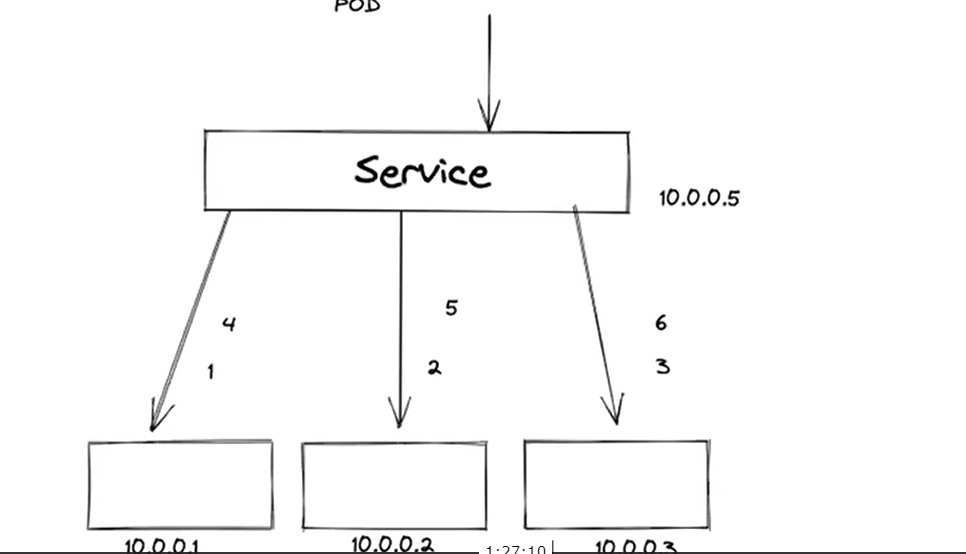
      image: nginx

    - name: debian-container

      image: debian

      command: ["sleep","1000"]





PS C:\Users\Saarit\OneDrive\Desktop\Me\DevOps\kubernetes> kubectl get po -n teaching -owide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

nginx 1/1 Running 0 46h 172.17.0.3 minikube <none> <none>

two-containers-with-debian 1/2 NotReady 0 24m 172.17.0.4 minikube <none> <none>

Service.yaml

apiVersion: v1

kind: Service

metadata:

  name: my-service

spec:

  selector:

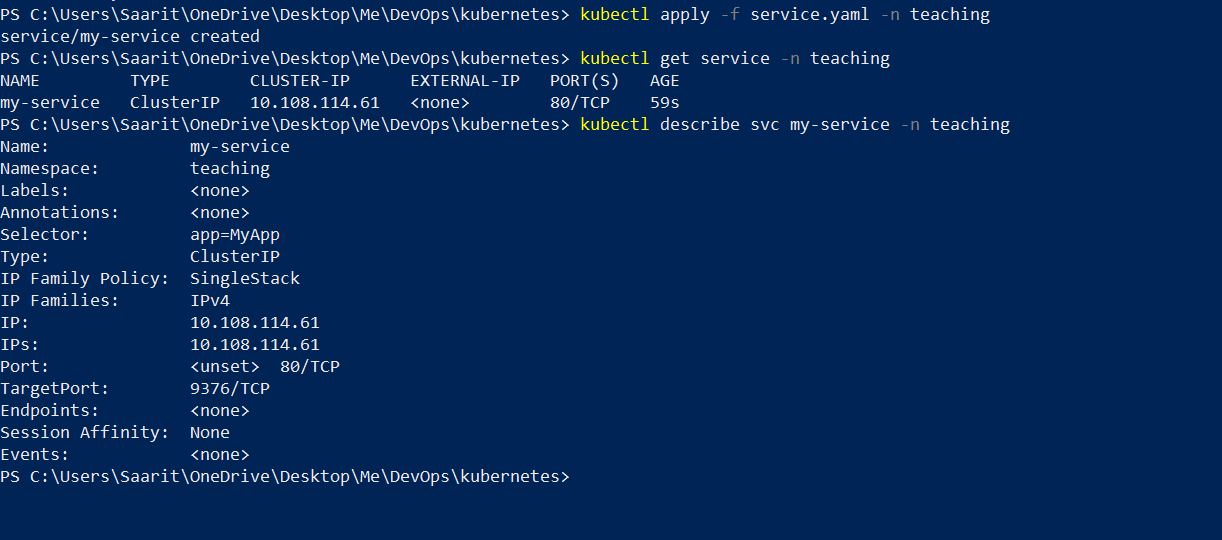
    app: MyApp

  ports:

    - protocol: TCP

      port: 80

      targetPort: 9376



Showing endpoints n label connect

Pod.yml

apiVersion: v1

kind: Pod

metadata:

  name: nginx

  labels:

    app: MyApp

spec:

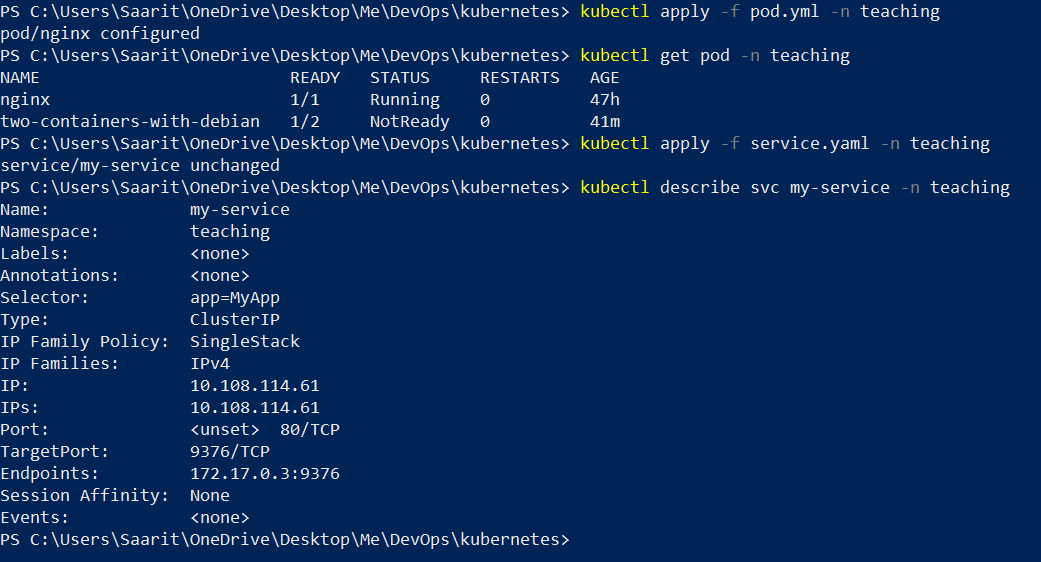
  containers:

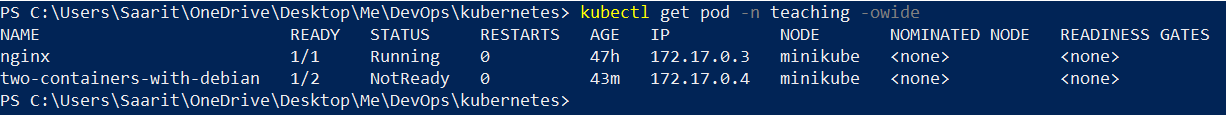
  - name: nginx

    image: nginx:1.14.2

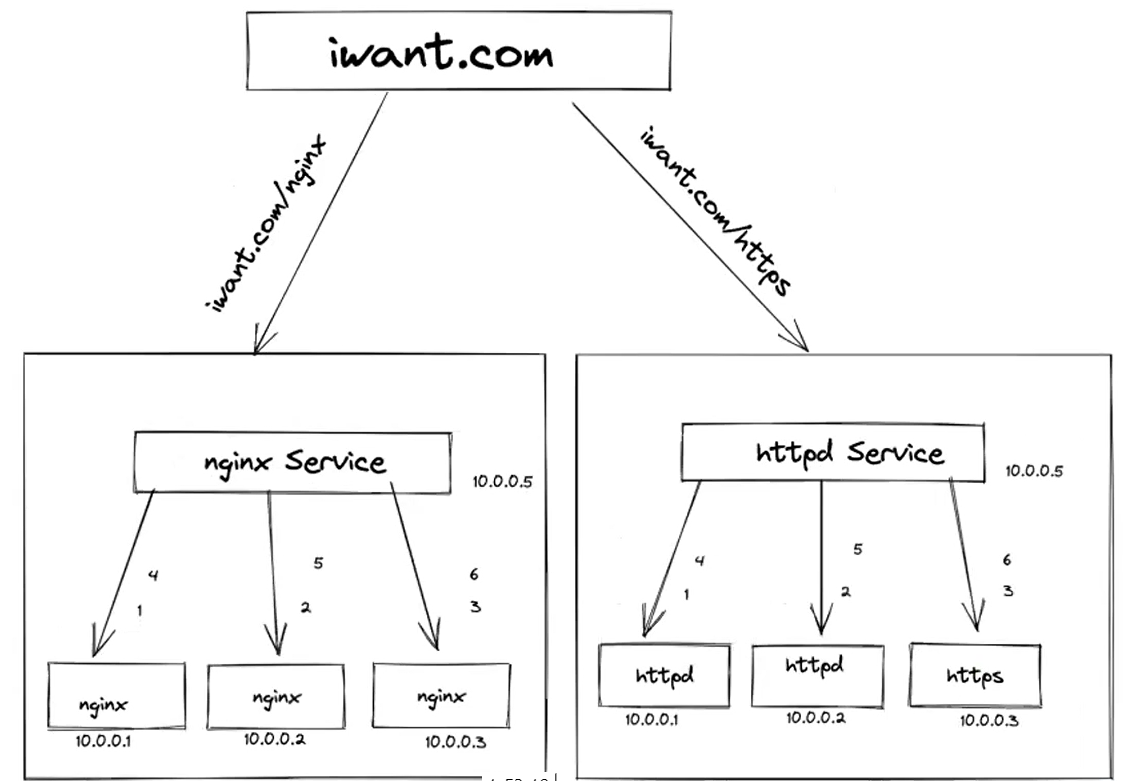
    ports:

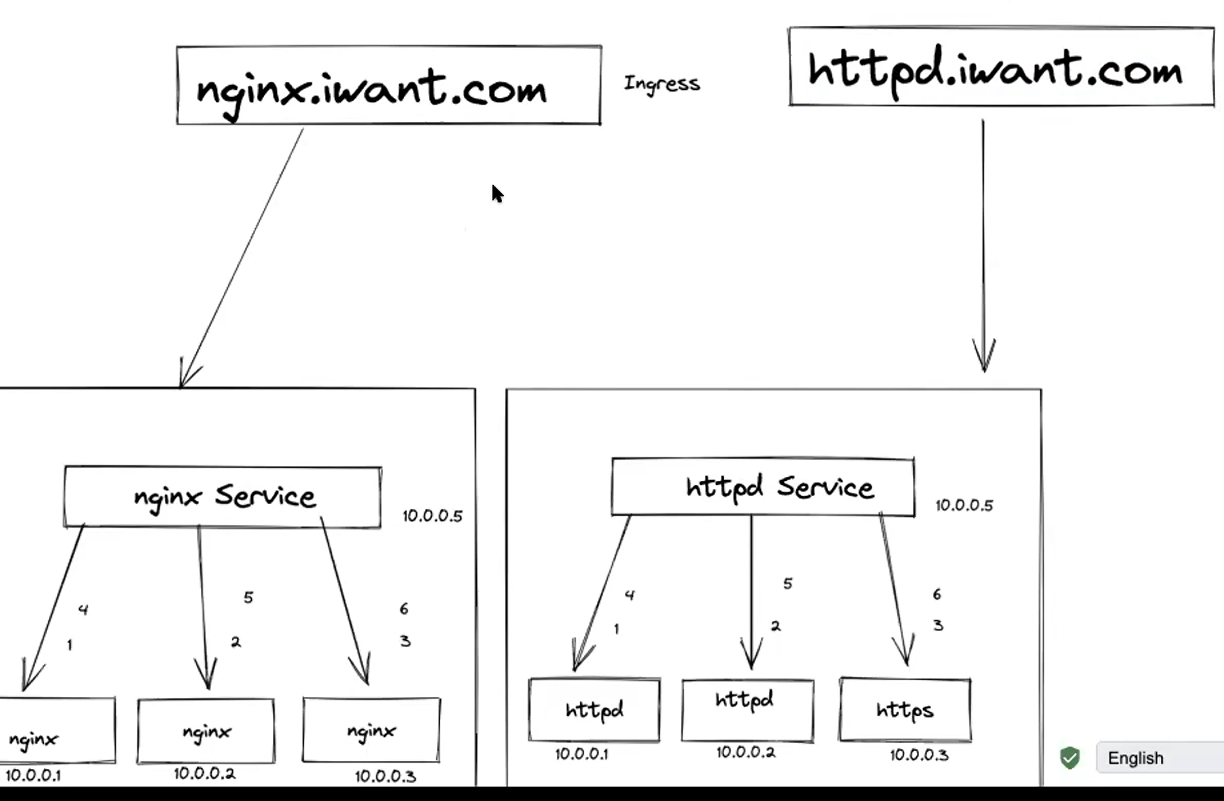
    - containerPort: 80





Ingress





Deployment

Deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

  labels:

    app: nginx

spec:

  replicas: 3

  selector:

    matchLabels:

     app: nginx

  template:

    metadata:

     labels:

       app: nginx

    spec:

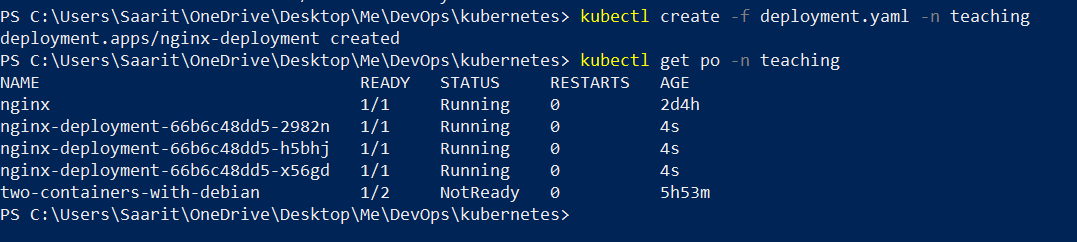
     containers:

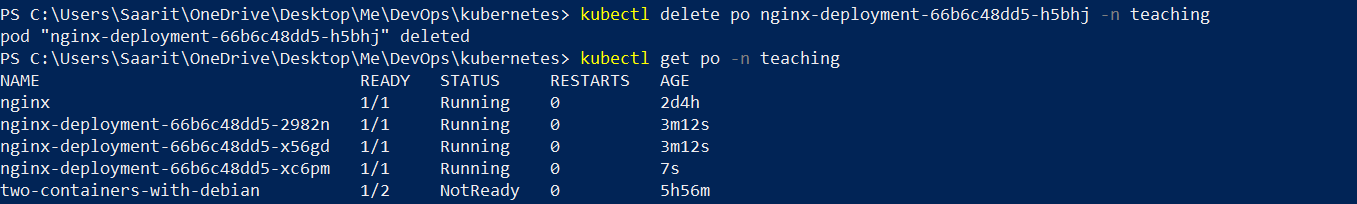
      - name: nginx

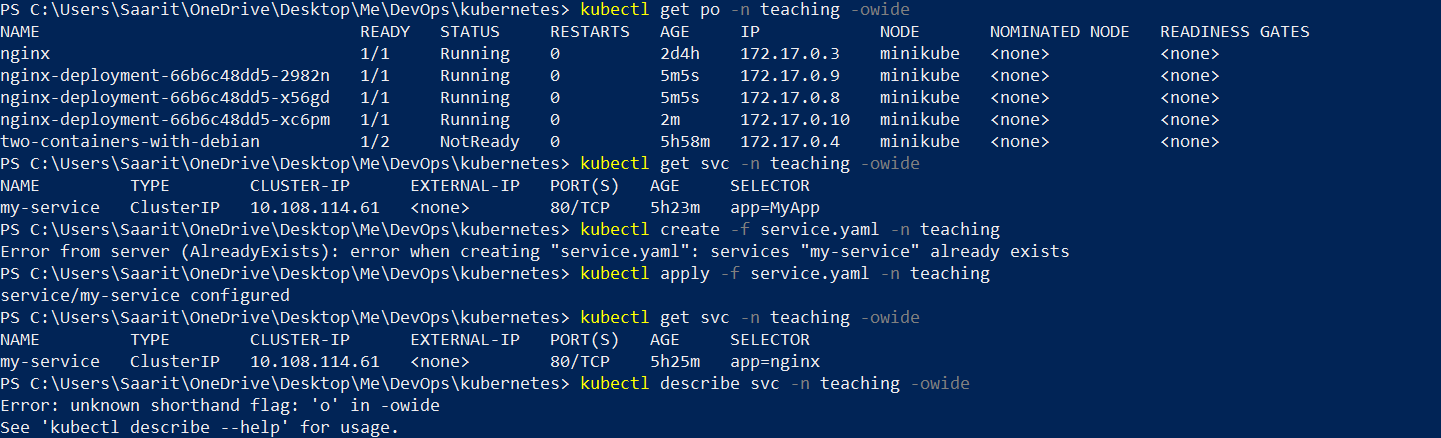
        image: nginx:1.14.2

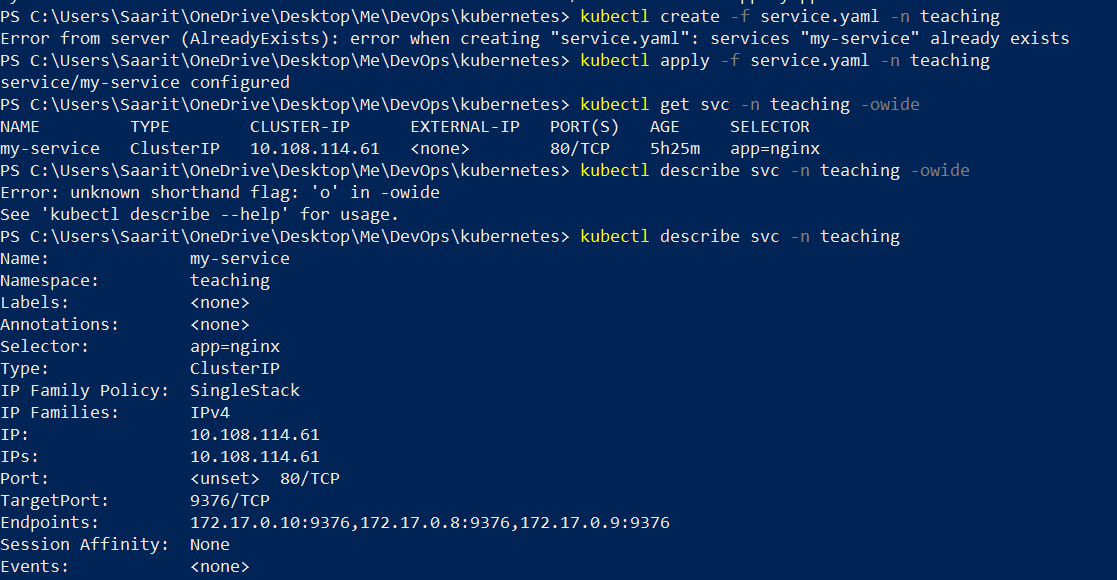
        ports:

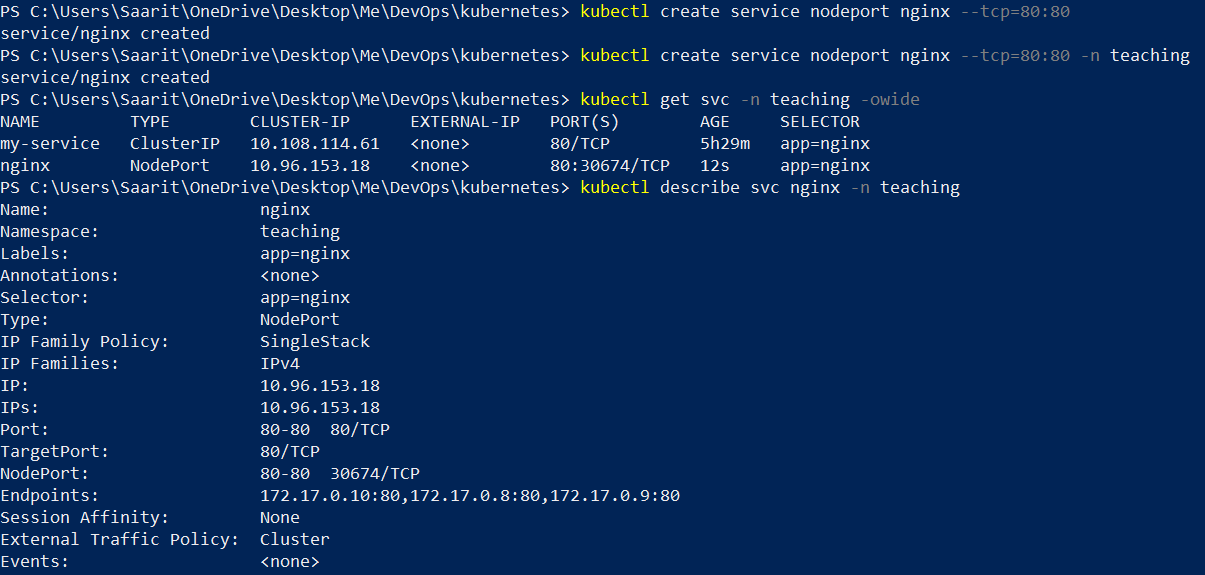
         - containerPort: 80





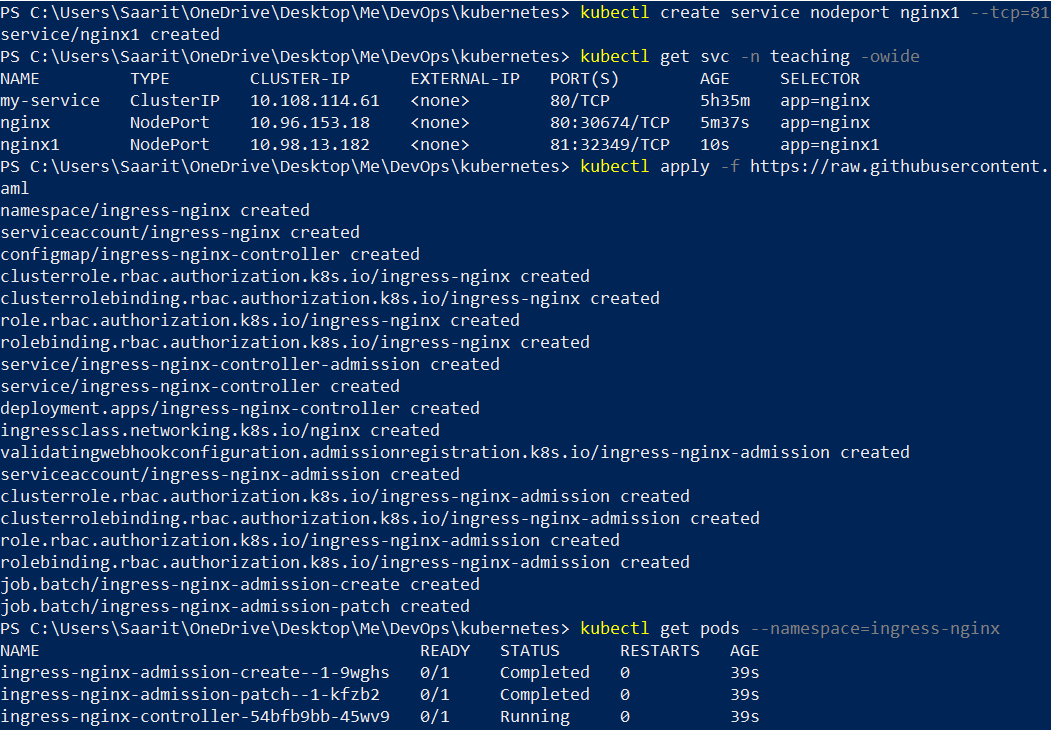






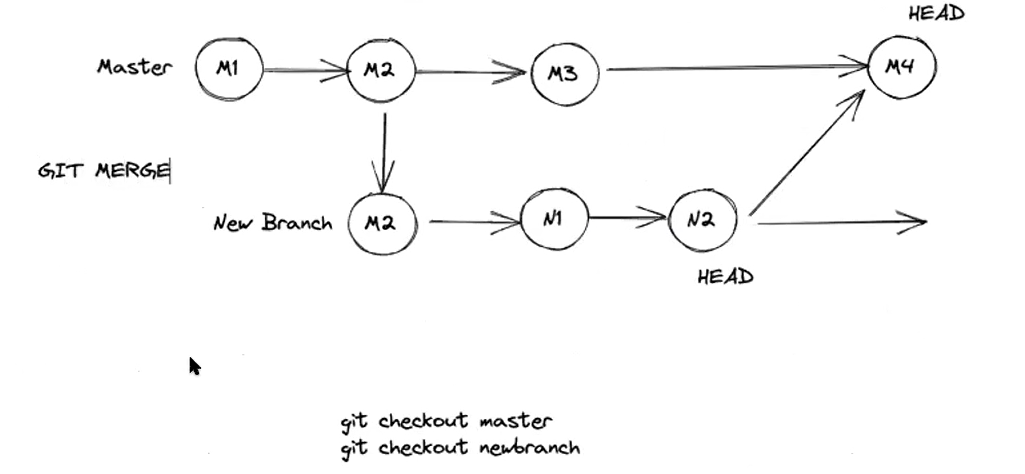


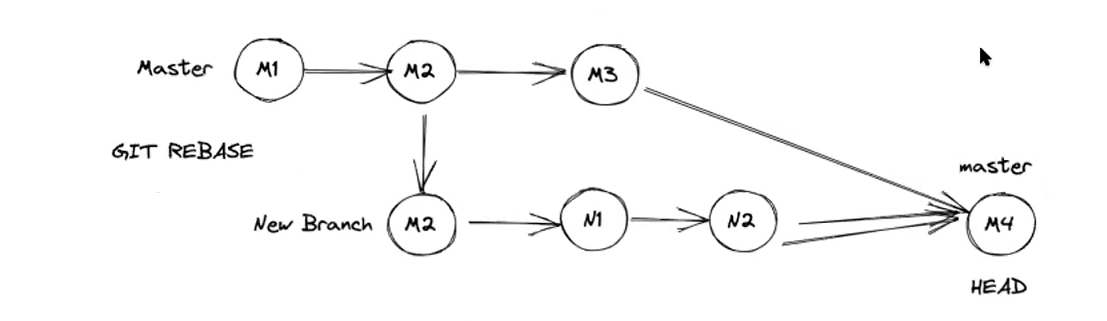
[ingress-nginx/index.md at main · kubernetes/ingress-nginx · GitHub](https://github.com/kubernetes/ingress-nginx/blob/main/docs/deploy/index.md)

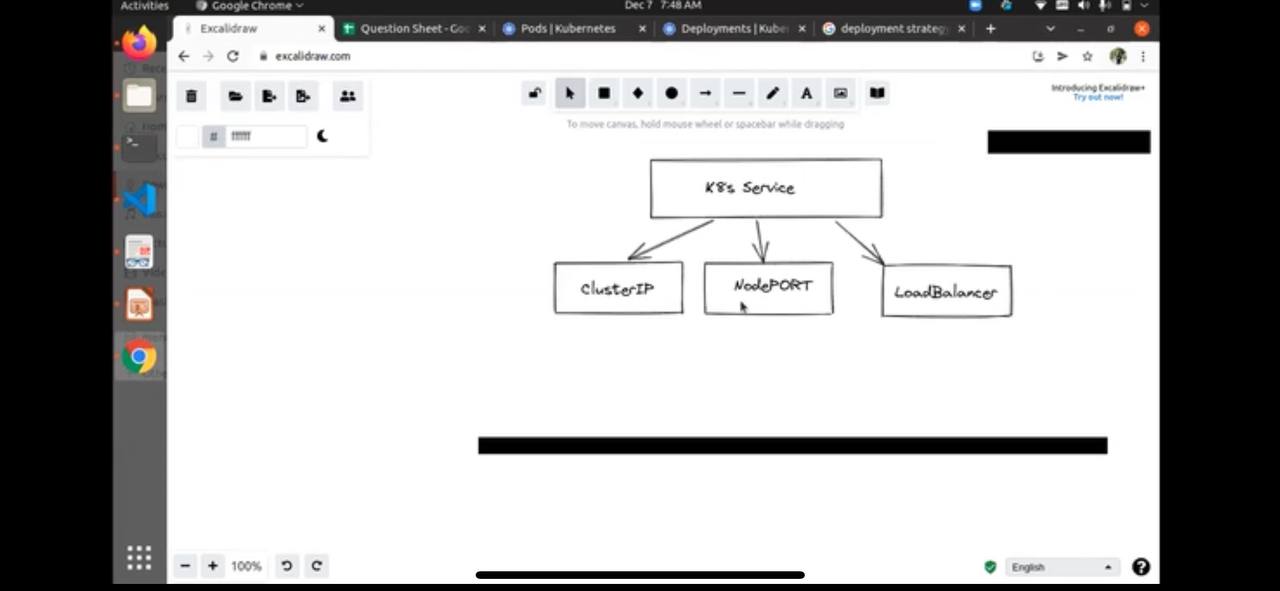


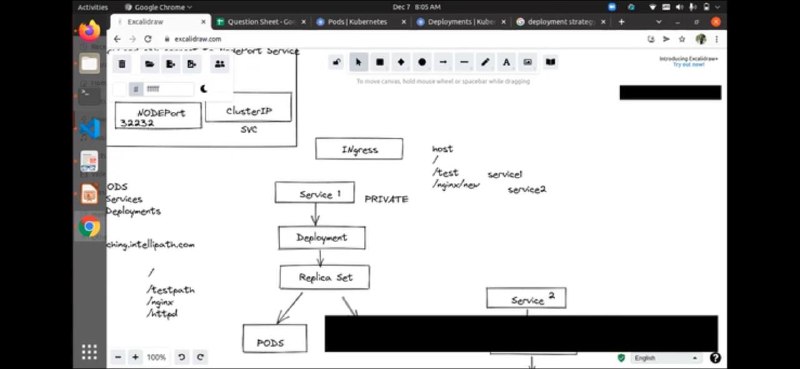
/// to automatically restart or revive a dead pod

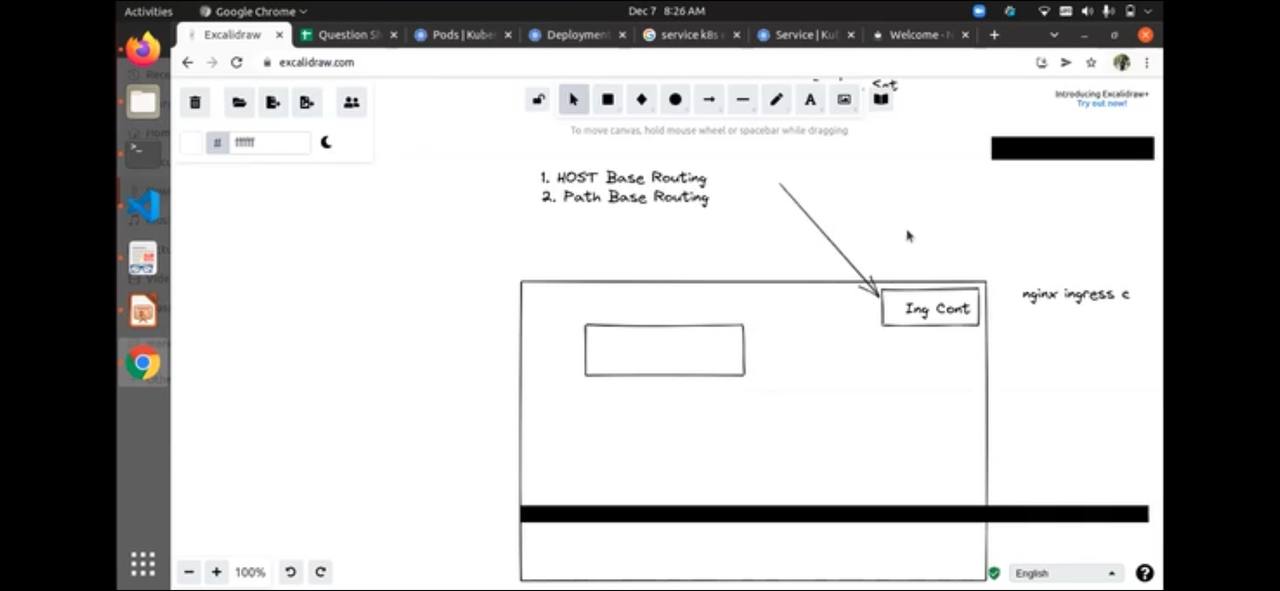
/////abt git

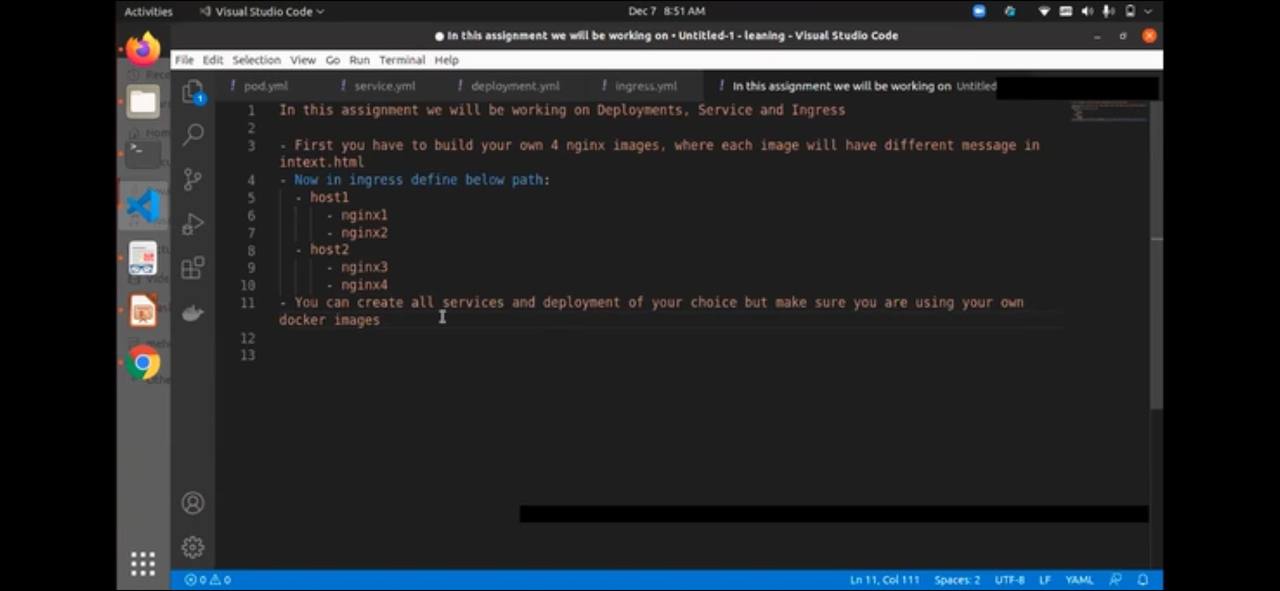










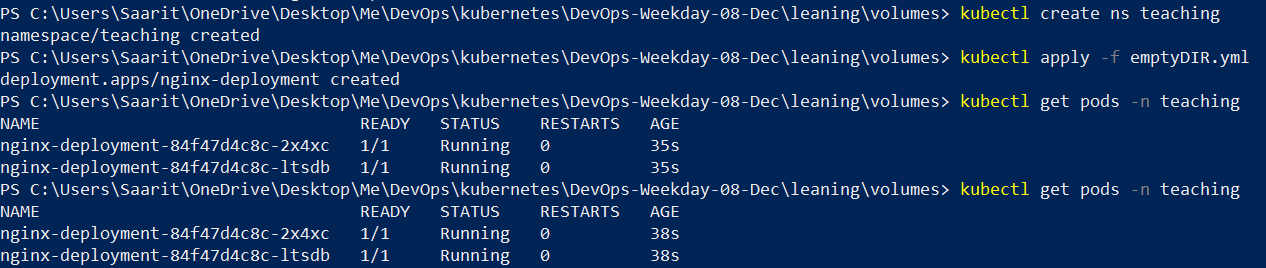


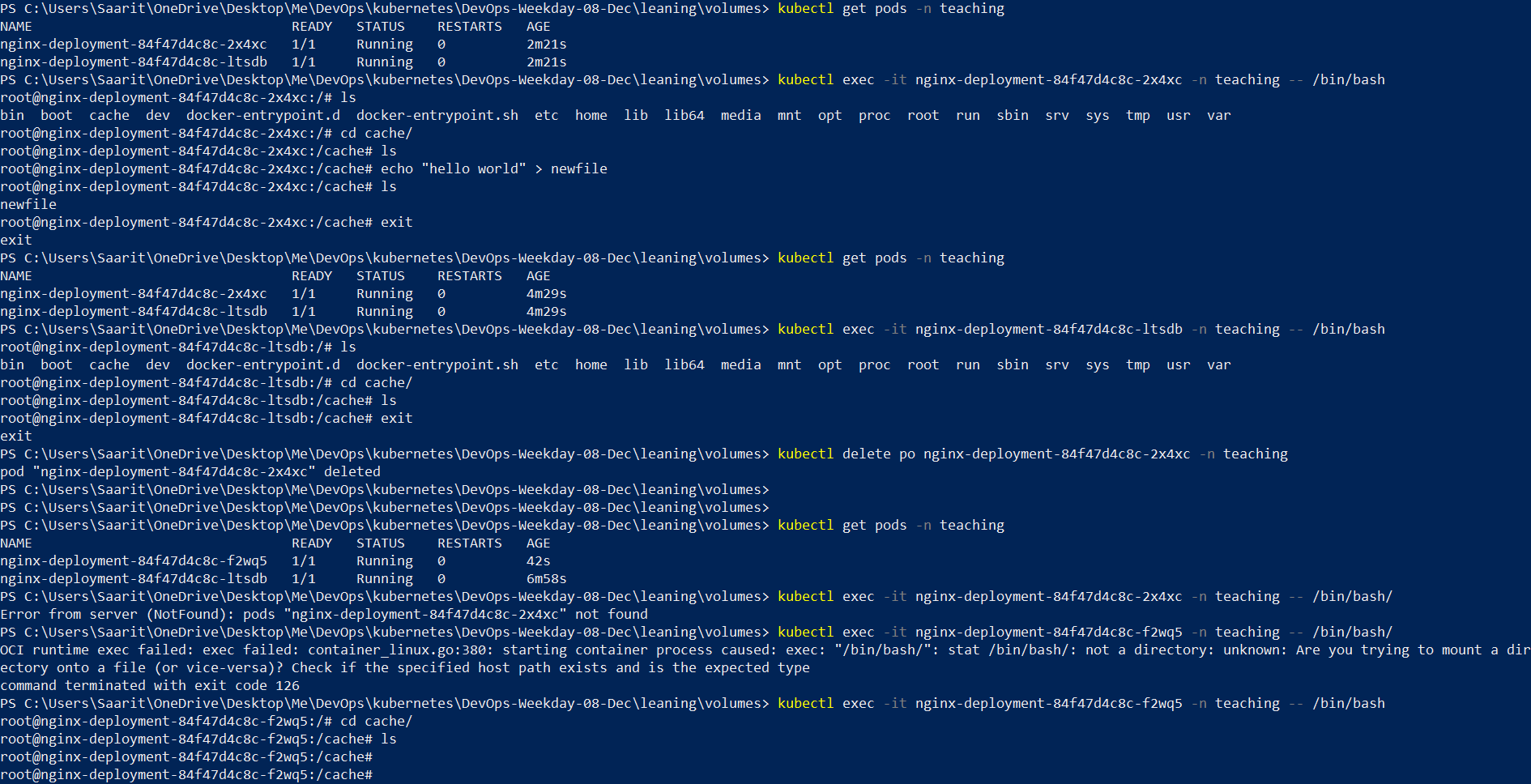
1. Login to Ubuntu with docker (done)
2. Create four nginx image with n upload it to dockerhub repository (done)
3. Deploy 4 nginx pods
4. Attach a service
5. Edit host file with hostname n minikube ip
6. Run ingress.yaml

## Volumes

Emptydirectory







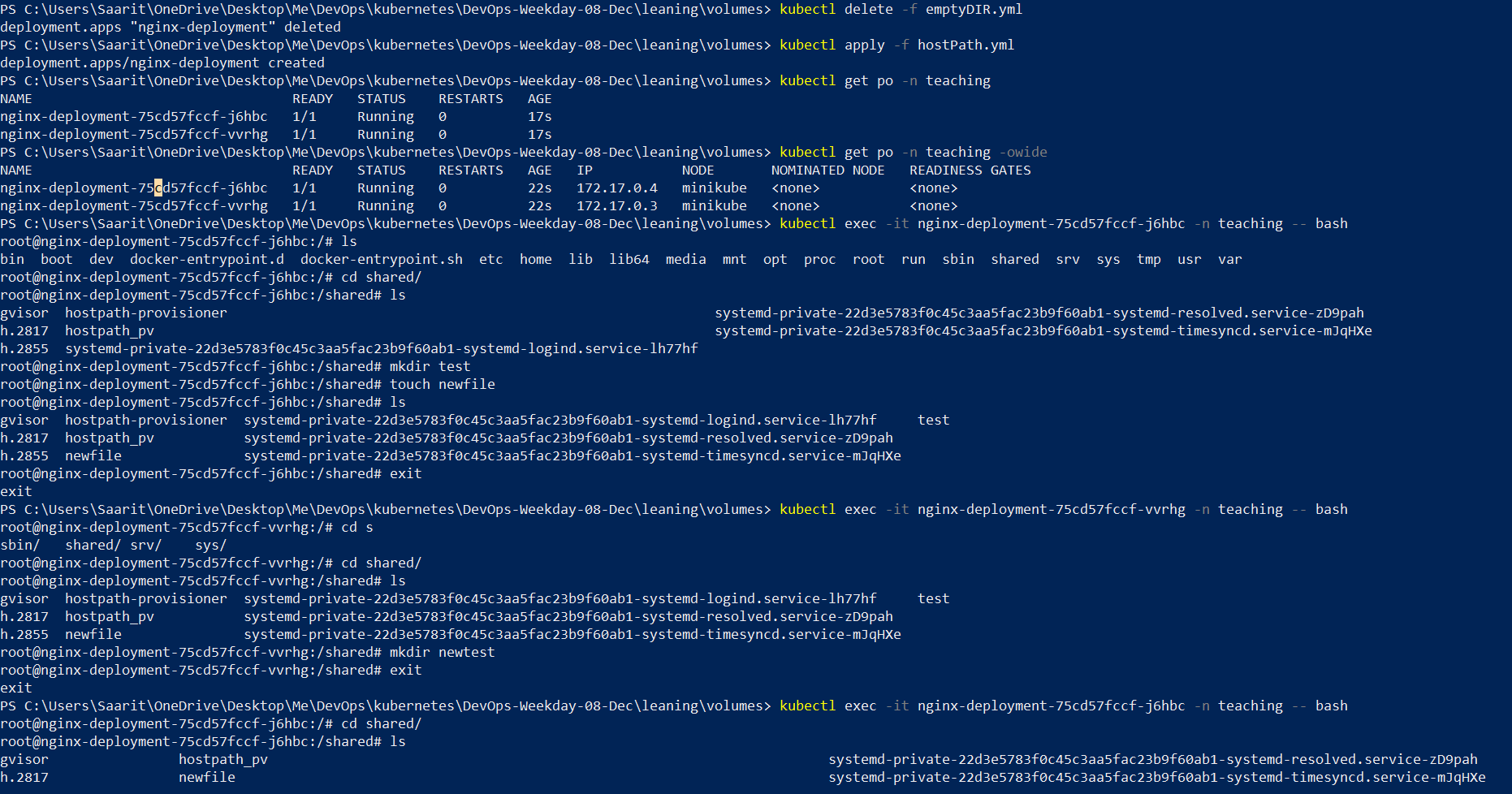


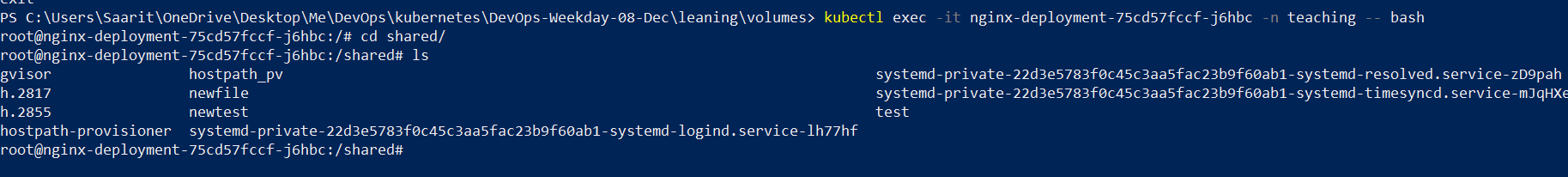
Hostpath

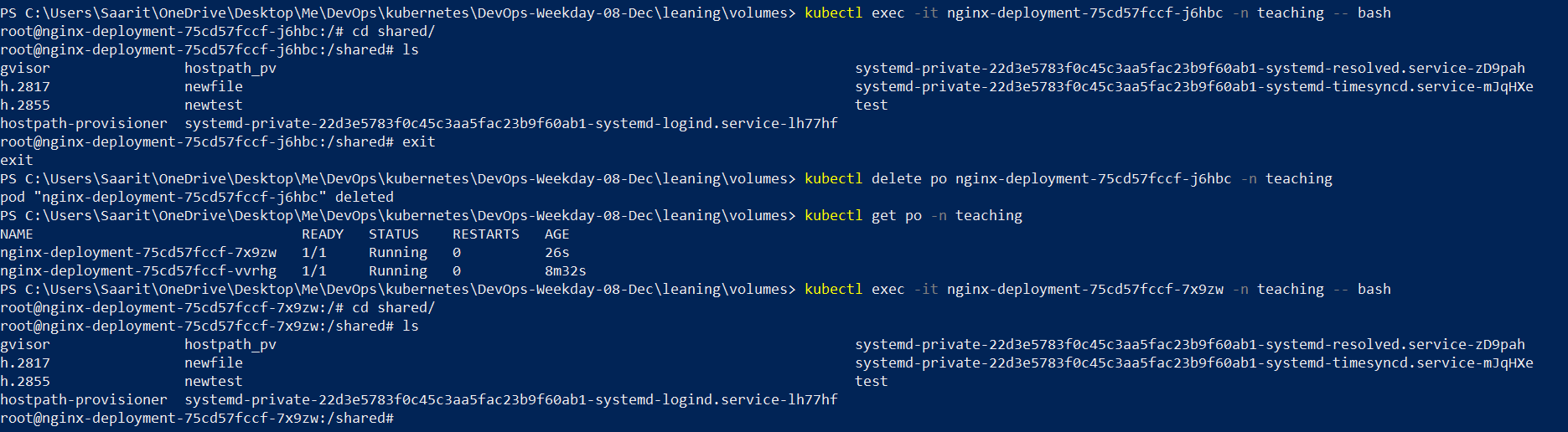
Data stored at node level not pod level

Less secured









PersistentVolume

Not free

Kubectl get pv –n elasticsearch

Kubectl describe pv *nameofpv* –n elasticsearch

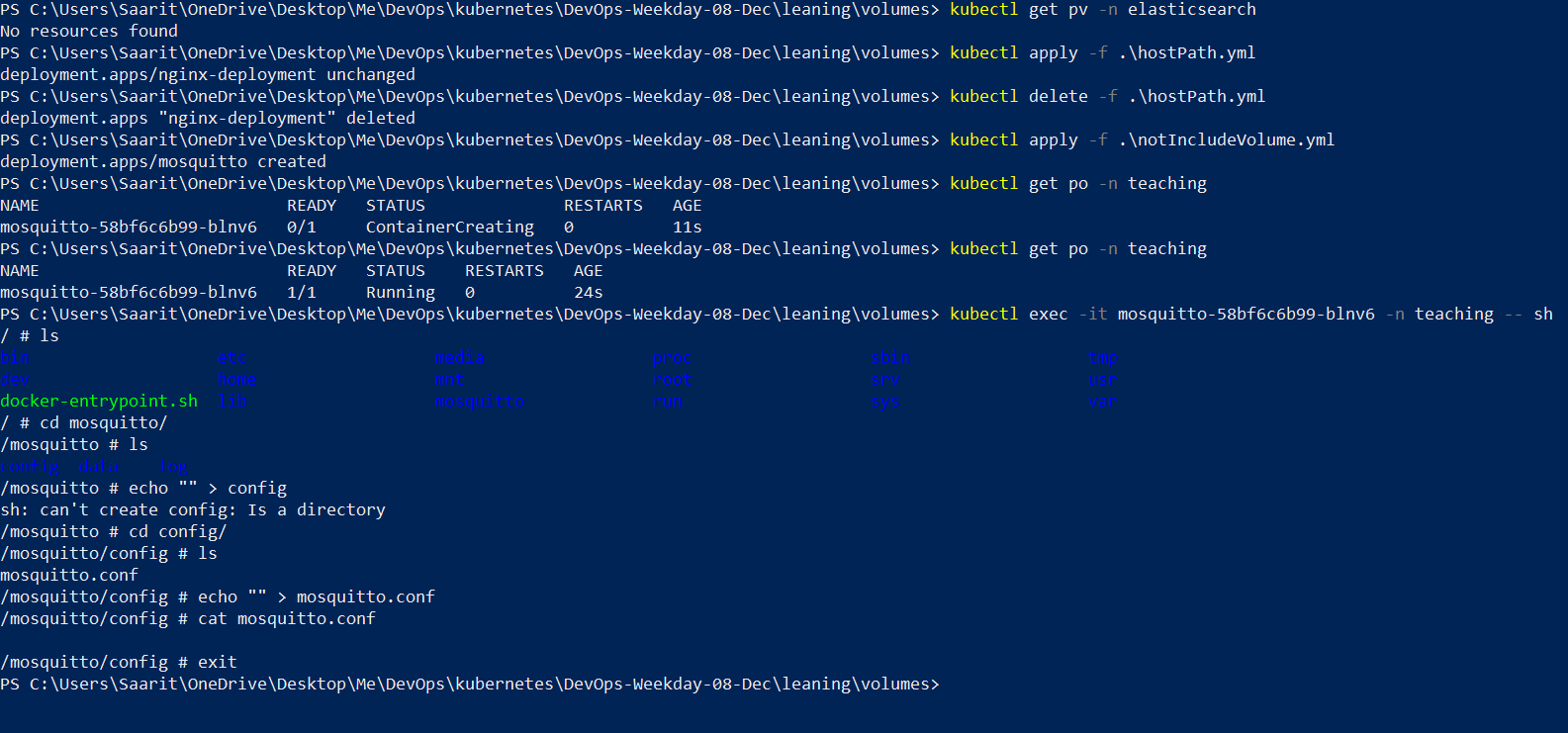
Kubectl get pvc –n elasticsearch

Kubectl describe pv *nameofpvc* –n elasticsearch

Storage can be increased but not decreased

Notincludevolume

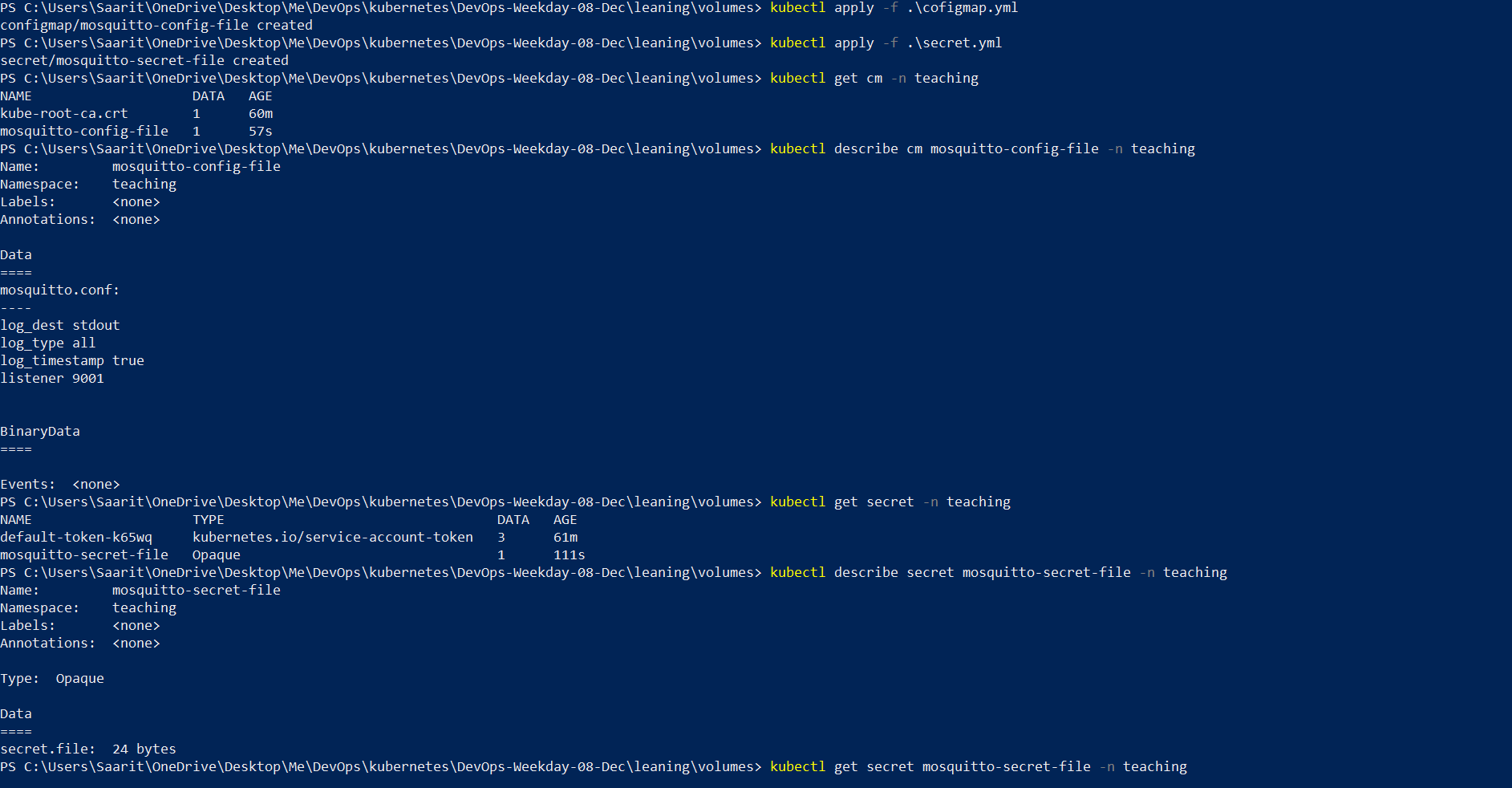


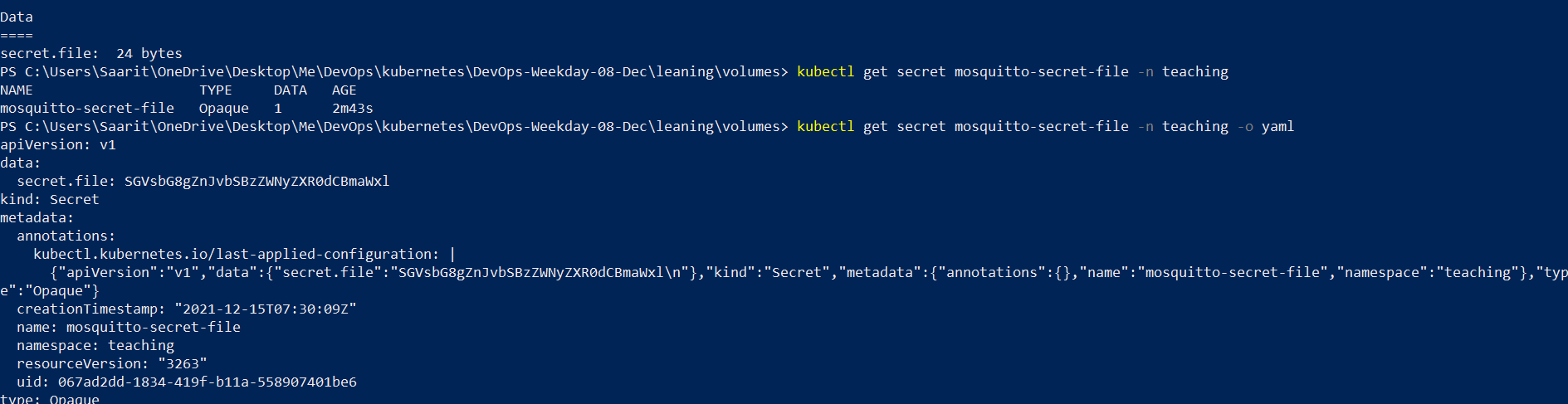


ConfigMAP

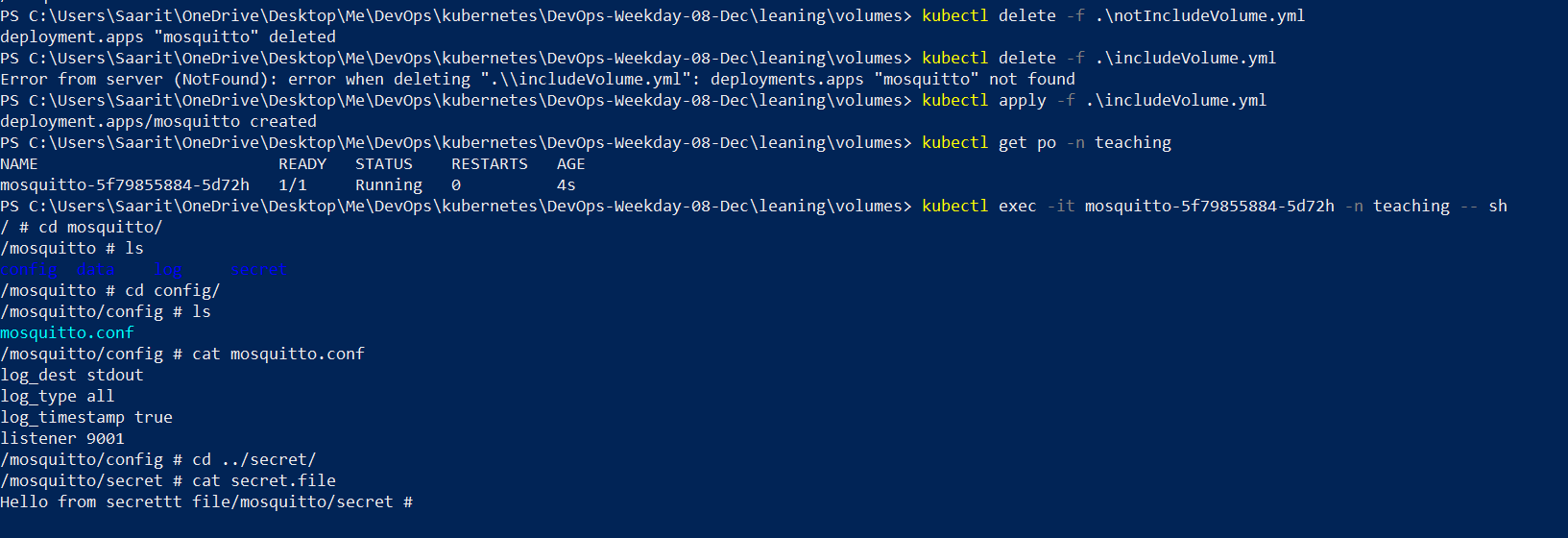
 

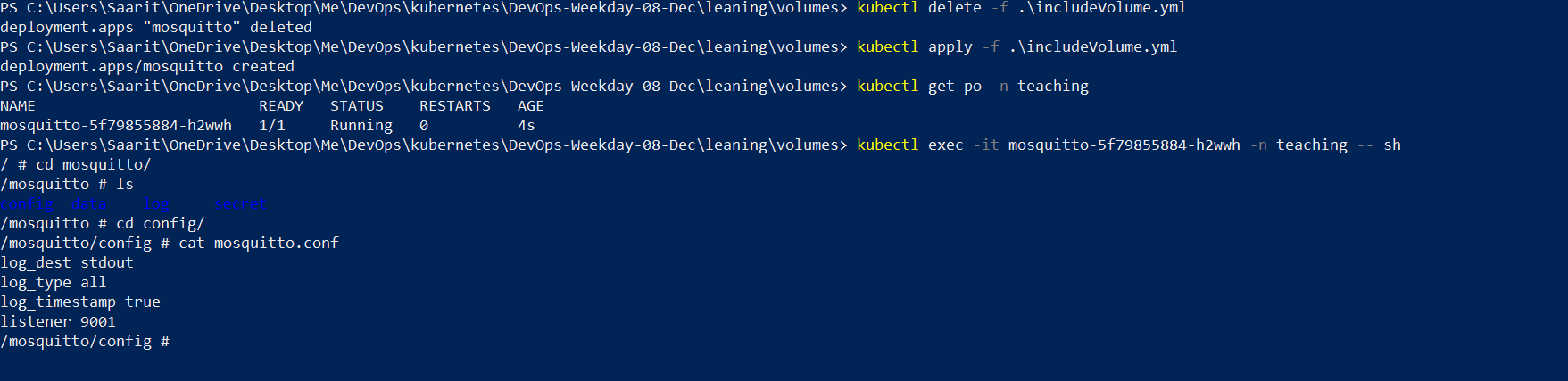
PS C:\Users\Saarit\OneDrive\Desktop\Me\DevOps\kubernetes\DevOps-Weekday-08-Dec\leaning\volumes> echo "SGVsbG8gZnJvbSBzZWNyZXR0dCBmaWxl" | base64 –d











## Delete ingress-controller

kubectl delete all --all -n ingress-nginx

### Ingress installation

kubectl delete all --all -n ingress-nginx

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.1/deploy/static/provider/cloud/deploy.yaml

kubectl get pods --namespace=ingress-nginx

kubectl wait --namespace ingress-nginx --for=condition=ready pod --selector=app.kubernetes.io/component=controller --timeout=120s

kubectl get pods --namespace=ingress-nginx

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.1/deploy/static/provider/baremetal/deploy.yaml>

## Dashboard kubernetes

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/dashboard/v2.4.0/aio/deploy/recommended.yaml>

kubectl get svc

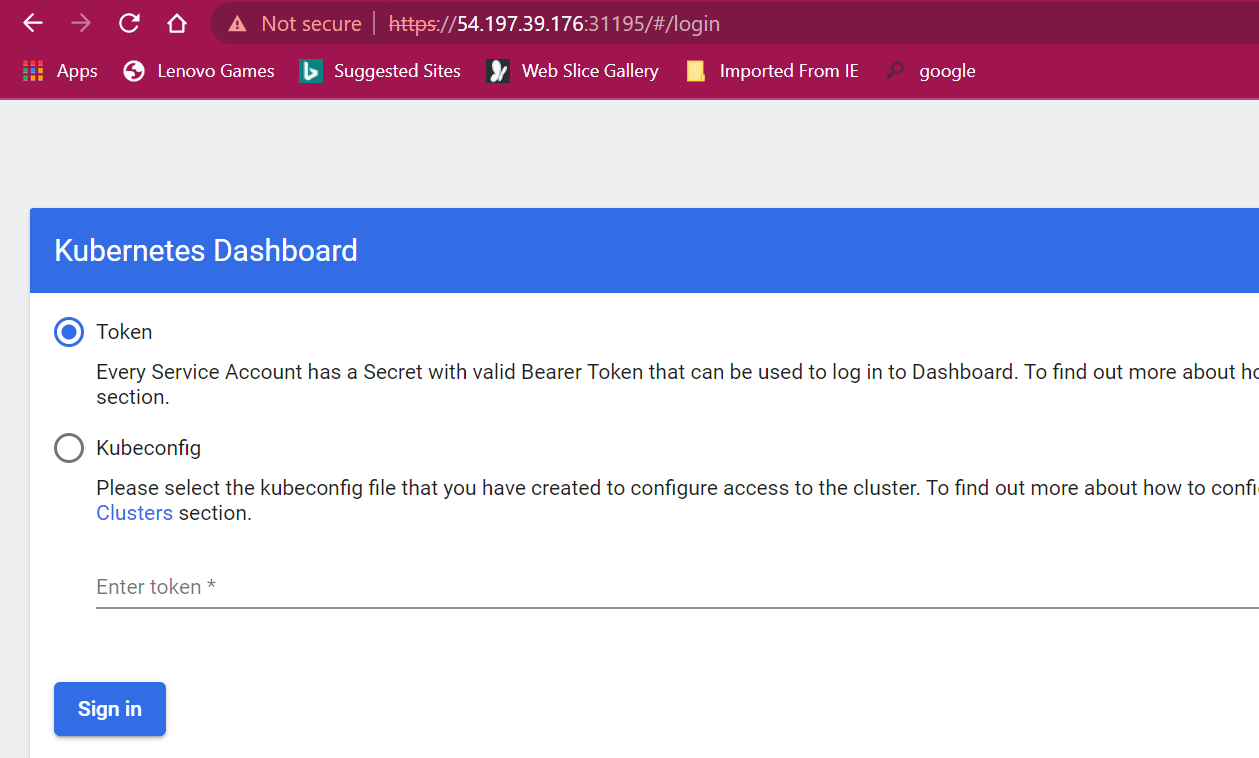
kubectl get svc --all-namespaces

kubectl edit services -n kubernetes-dashboard kubernetes-dashboard

------Edit: type: NodePort

-------Save

kubectl get svc --all-namespaces



kubectl create serviceaccount cluster-admin-dashboard-sa

kubectl create clusterrolebinding cluster-admin-dashboard-sa --clusterrole=cluster-admin --serviceaccount=default:cluster-admin-dashboard-sa

TOKEN=$(kubectl describe secret $(kubectl -n kube-system get secret | awk '/^cluster-admin-dashboard- sa-token-/{print $1}') | awk '$1=="token:"{print $2}')

echo $TOKEN

enter to token n login to dashboard

now, create a deployment in dashboard