Name: Kiran Yadav

Email: kiranyadav1102003@gmail.com

Course Name:Devops and Cloud Computing Course

Assignment Name: Kubernetes Assignment 1

Git Link:https://github.com/Hydra-Dev110/Kubernetes

Drive Link:
https://docs.google.com/document/d/1r2pbSBKMjvU3WisCdzfqRR9_9VbDthThrNHdEkSv_Zs/edit?usp=sharing

--

1. Kubernetes Setup: Set up a local Kubernetes cluster using Minikube or a managed Kubernetes service ?

1. KUBERNETES SETUP

Objective:
To set up a local Kubernetes cluster for running containerized applications.

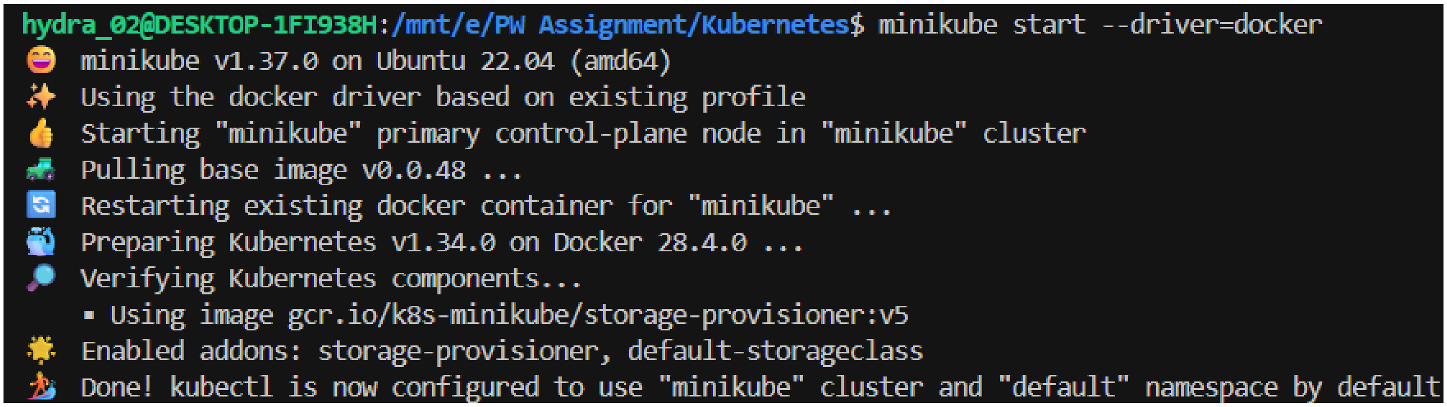
Tools Used:
- Docker
- kubectl
- Minikube

Minikube Architecture view in local

```
Laptop / VM
│
├── Minikube
│    │
│    ├── Control Plane
│    │    ├── API Server
│    │    ├── etcd
│    │    ├── Scheduler
│    │    └── Controller Manager
│    │
│    └── Worker Node
│         ├── kubelet
│         ├── kube-proxy
│         └── containerd / Docker
│
└── Local Kubernetes Cluster (Single Node)
```

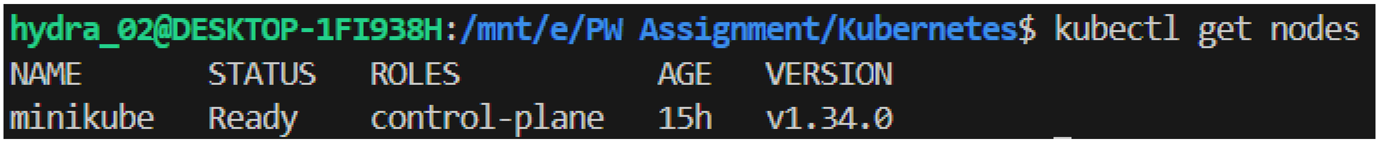
Procedure:

1. Install Docker, kubectl, and Minikube on the system.
2. Start the local Kubernetes cluster using Minikube:
   minikube start --driver=docker



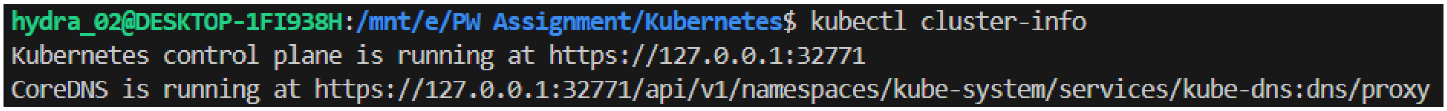
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ minikube start --driver=docker
😄  minikube v1.37.0 on Ubuntu 22.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

3. Verify the cluster:
   kubectl get nodes



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   15h   v1.34.0
```

kubectl cluster-info



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:32771
CoreDNS is running at https://127.0.0.1:32771/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

Result:

A single-node Kubernetes cluster was successfully created and the node status was Ready.

2. Application Deployment: Deploy a simple application to your Kubernetes cluster?

APPLICATION DEPLOYMENT

Objective:

To deploy a simple application on the Kubernetes cluster.

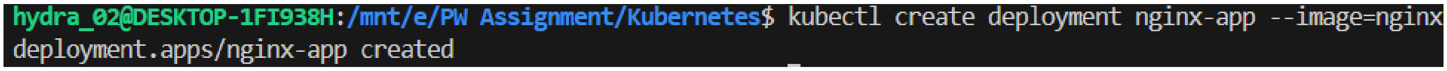
Application Used:

NGINX web server

Procedure:

1. Create a Kubernetes deployment:

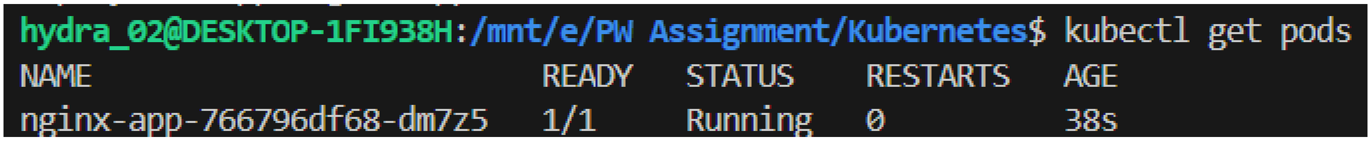
```
kubectl create deployment nginx-app --image=nginx
```



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl create deployment nginx-app --image=nginx
deployment.apps/nginx-app created
```

2. Verify that the pod is running:

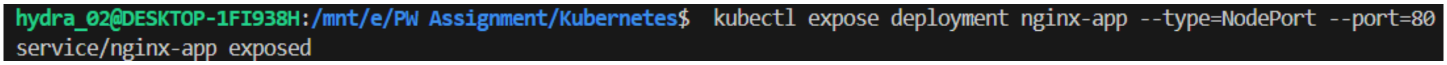
```
kubectl get pods
```



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
nginx-app-766796df68-dm7z5  1/1     Running   0          38s
```

3. Expose the deployment as a service:

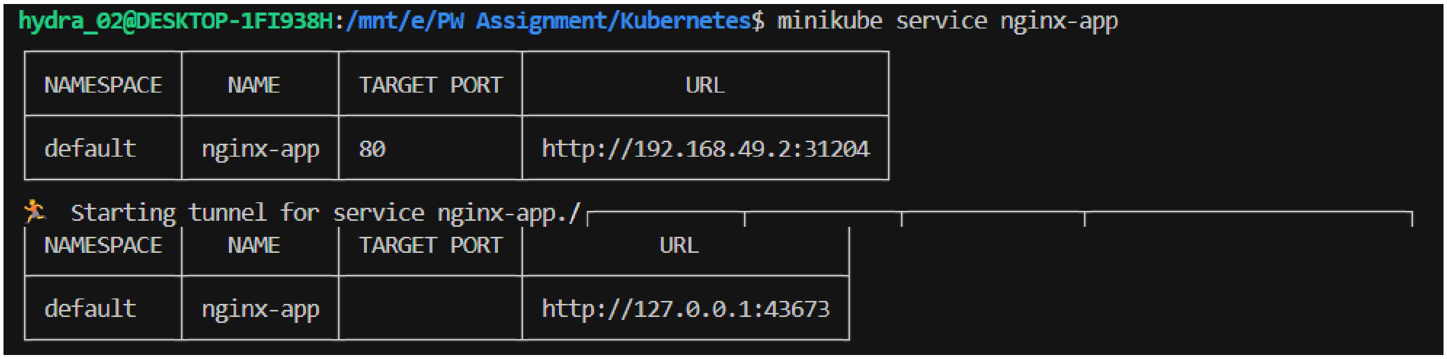
```
kubectl expose deployment nginx-app --type=NodePort --port=80
```



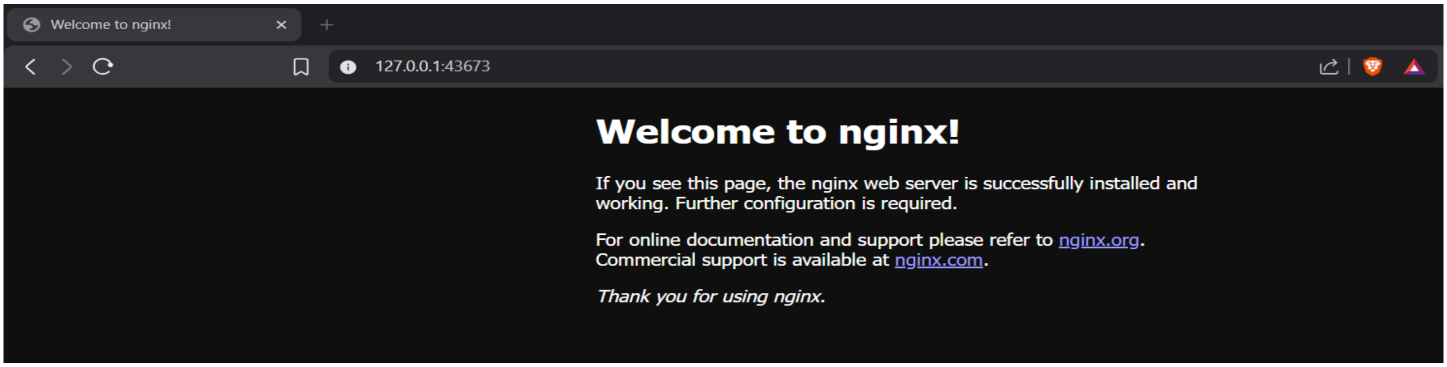
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$  kubectl expose deployment nginx-app --type=NodePort --port=80
service/nginx-app exposed
```

4. Access the application:

```
minikube service nginx-app
```



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ minikube service nginx-app
|-----------|-----------|-------------|---------------------------|
| NAMESPACE |   NAME    | TARGET PORT |            URL            |
|-----------|-----------|-------------|---------------------------|
| default   | nginx-app | 80          | http://192.168.49.2:31204 |
|-----------|-----------|-------------|---------------------------|
🏃  Starting tunnel for service nginx-app./
|-----------|-----------|-------------|------------------------|
| NAMESPACE |   NAME    | TARGET PORT |          URL           |
|-----------|-----------|-------------|------------------------|
| default   | nginx-app |             | http://127.0.0.1:43673 |
|-----------|-----------|-------------|------------------------|
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

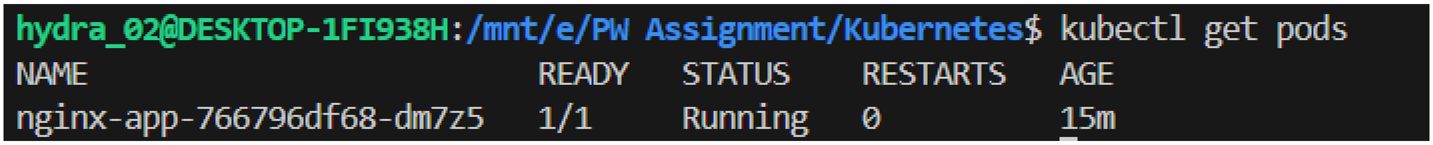
Result:

The NGINX application was successfully deployed and accessed through a browser

3. Resource Management: Practice managing Kubernetes resources like Pods, Services, and Deployments?

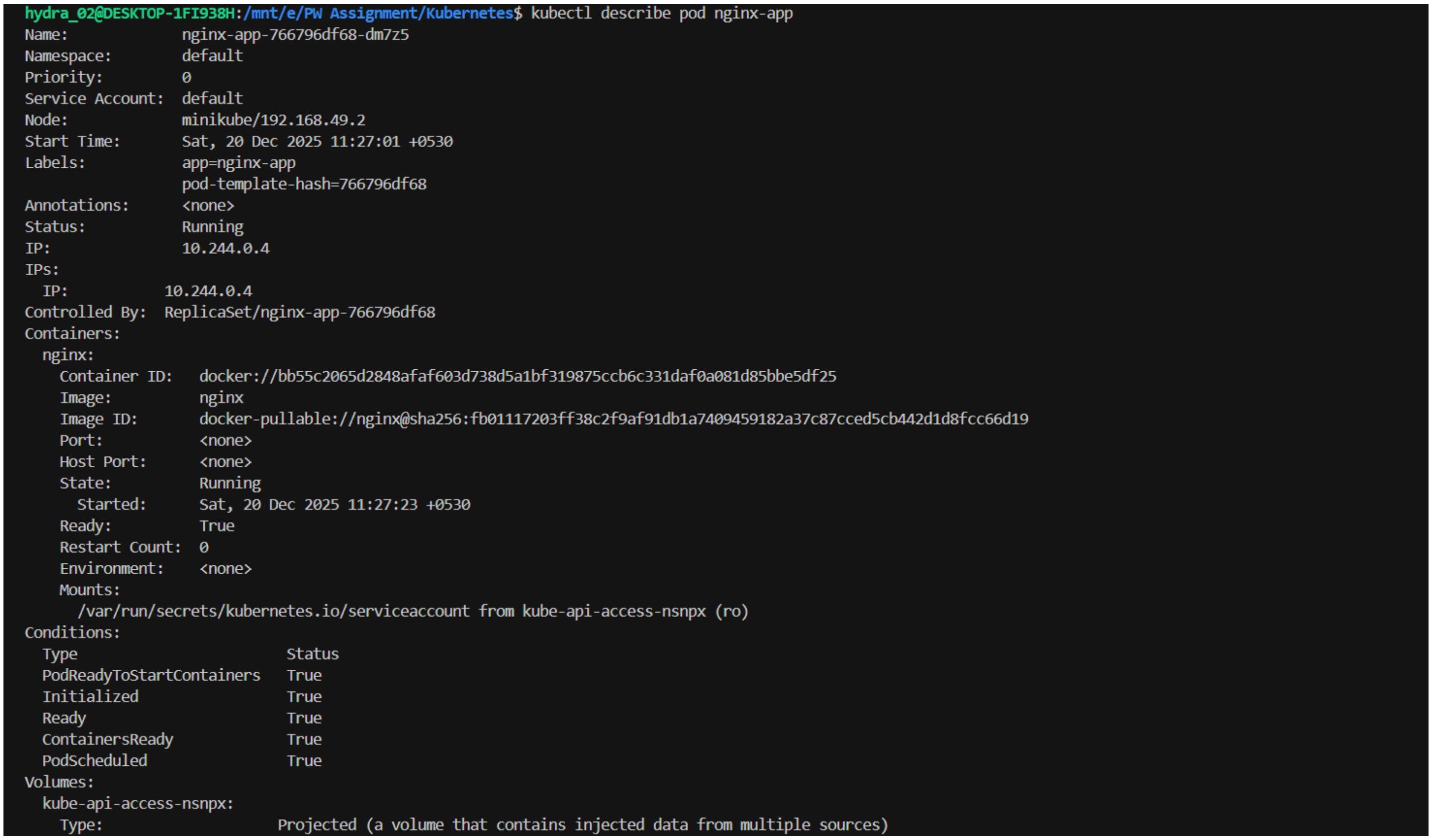
Pods:

- View running pods:
  kubectl get pods

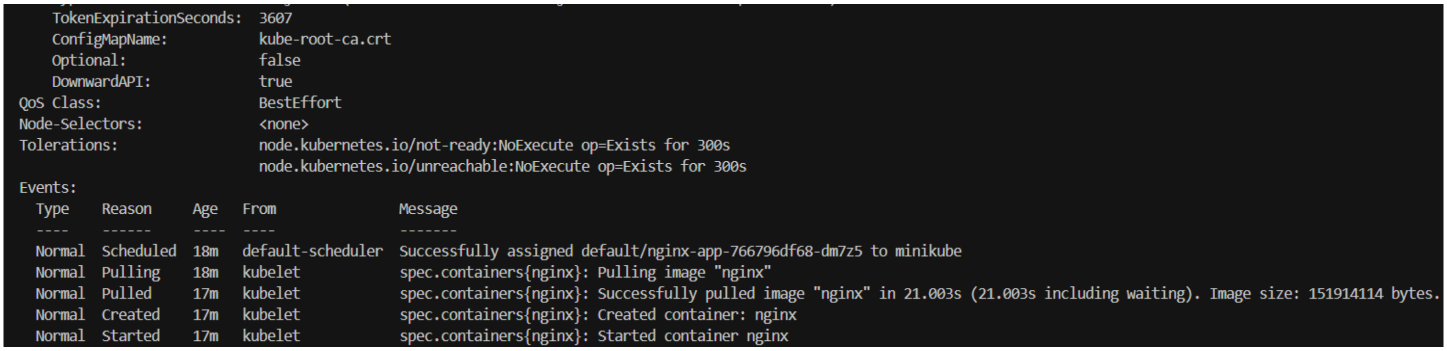


```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
nginx-app-766796df68-dm7z5   1/1     Running   0          15m
```

- Describe pod details:
  kubectl describe pod nginx-app

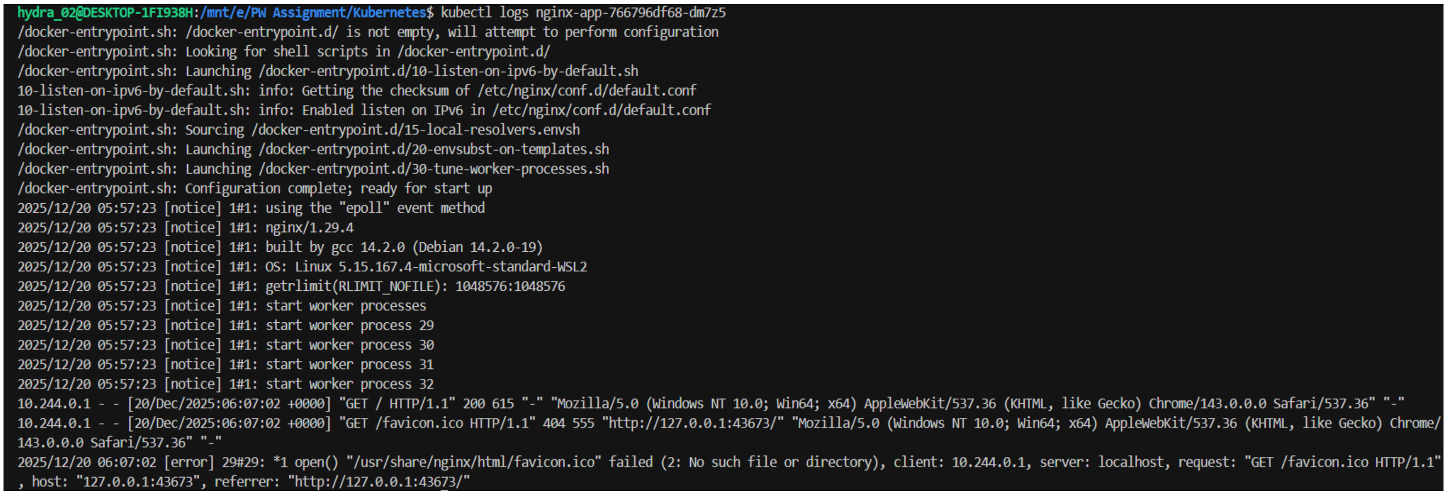


```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl describe pod nginx-app
Name:             nginx-app-766796df68-dm7z5
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Sat, 20 Dec 2025 11:27:01 +0530
Labels:           app=nginx-app
                  pod-template-hash=766796df68
Annotations:      <none>
Status:           Running
IP:               10.244.0.4
IPs:
  IP:           10.244.0.4
Controlled By:  ReplicaSet/nginx-app-766796df68
Containers:
  nginx:
    Container ID:   docker://bb55c2065d2848afaf603d738d5a1bf319875ccb6c331daf0a081d85bbe5df25
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d19
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sat, 20 Dec 2025 11:27:23 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nsnpx (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-nsnpx:
    Type:                    Projected (a volume that contains injected data from multiple sources)
```



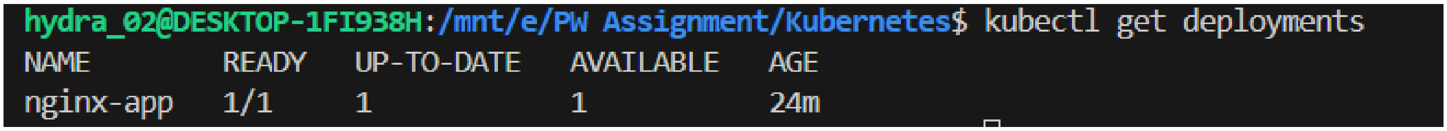
```
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  18m   default-scheduler  Successfully assigned default/nginx-app-766796df68-dm7z5 to minikube
  Normal  Pulling    18m   kubelet            spec.containers{nginx}: Pulling image "nginx"
  Normal  Pulled     17m   kubelet            spec.containers{nginx}: Successfully pulled image "nginx" in 21.003s (21.003s including waiting). Image size: 151914114 bytes.
  Normal  Created    17m   kubelet            spec.containers{nginx}: Created container: nginx
  Normal  Started    17m   kubelet            spec.containers{nginx}: Started container nginx
```

- View pod logs:
  kubectl logs nginx-app



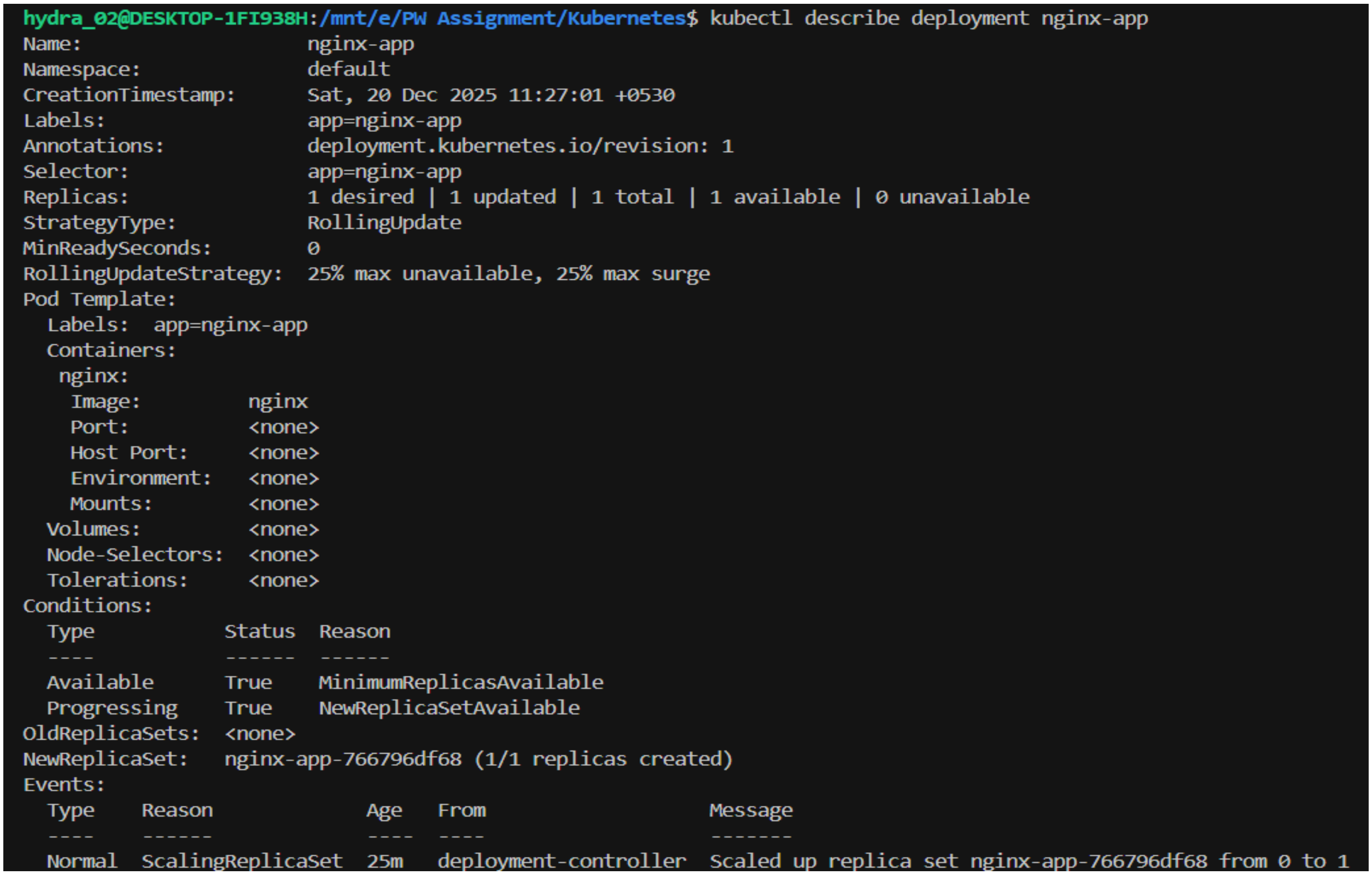
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl logs nginx-app-766796df68-dm7z5
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /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
2025/12/20 05:57:23 [notice] 1#1: using the "epoll" event method
2025/12/20 05:57:23 [notice] 1#1: nginx/1.29.4
2025/12/20 05:57:23 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/12/20 05:57:23 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/12/20 05:57:23 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/12/20 05:57:23 [notice] 1#1: start worker processes
2025/12/20 05:57:23 [notice] 1#1: start worker process 29
2025/12/20 05:57:23 [notice] 1#1: start worker process 30
2025/12/20 05:57:23 [notice] 1#1: start worker process 31
2025/12/20 05:57:23 [notice] 1#1: start worker process 32
10.244.0.1 - - [20/Dec/2025:06:07:02 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
10.244.0.1 - - [20/Dec/2025:06:07:02 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://127.0.0.1:43673/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
143.0.0.0 Safari/537.36" "-"
2025/12/20 06:07:02 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.244.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1"
, host: "127.0.0.1:43673", referrer: "http://127.0.0.1:43673/"
```

Deployments:
- View deployments:
  kubectl get deployments



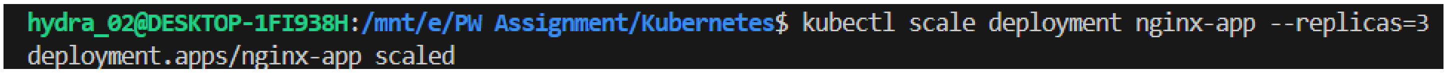
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app   1/1     1            1           24m
```

- Describe deployment:
  kubectl describe deployment nginx-app



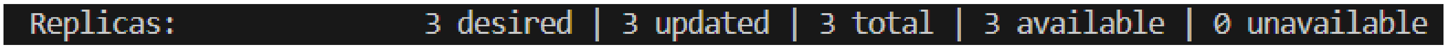
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl describe deployment nginx-app
Name:                   nginx-app
Namespace:              default
CreationTimestamp:      Sat, 20 Dec 2025 11:27:01 +0530
Labels:                 app=nginx-app
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx-app
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx-app
  Containers:
   nginx:
    Image:         nginx
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-app-766796df68 (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  25m   deployment-controller  Scaled up replica set nginx-app-766796df68 from 0 to 1
```

- Scale deployment:
  kubectl scale deployment nginx-app --replicas=3



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl scale deployment nginx-app --replicas=3
deployment.apps/nginx-app scaled
```

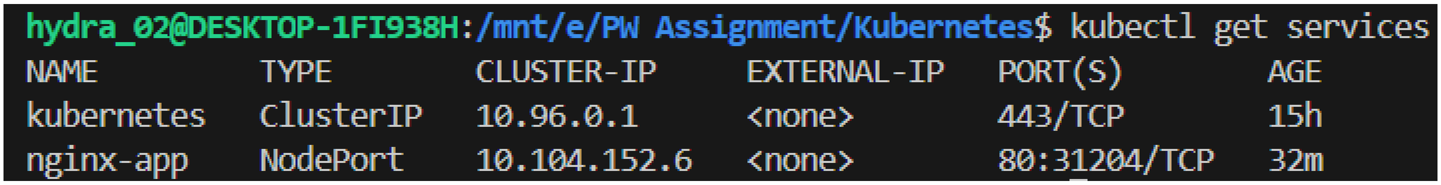
kubectl describe deployment nginx-app



```
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
```

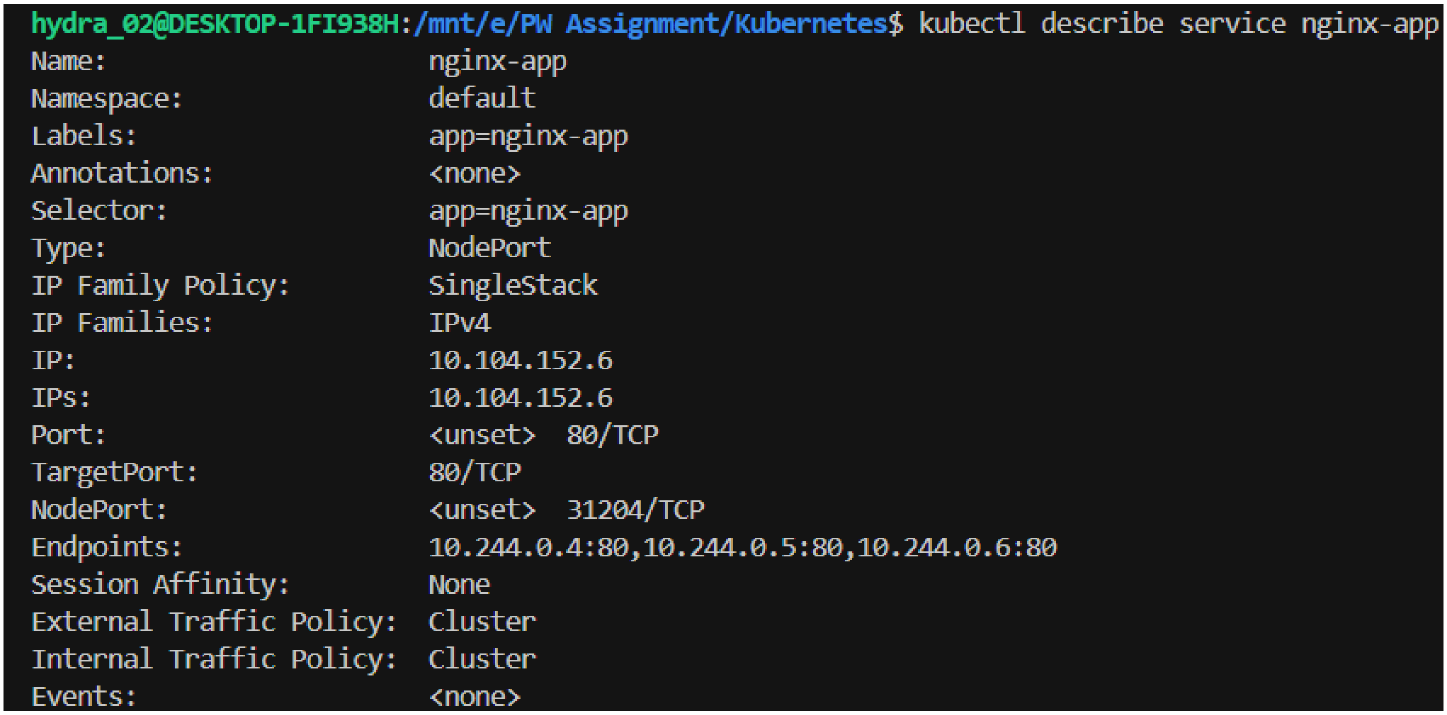
Services:

- View services:
  kubectl get services



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get services
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP        15h
nginx-app    NodePort    10.104.152.6   <none>        80:31204/TCP   32m
```

- Describe service:
  kubectl describe service nginx-app



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl describe service nginx-app
Name:                     nginx-app
Namespace:                default
Labels:                   app=nginx-app
Annotations:              <none>
Selector:                 app=nginx-app
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.104.152.6
IPs:                      10.104.152.6
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31204/TCP
Endpoints:                10.244.0.4:80,10.244.0.5:80,10.244.0.6:80
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
```

Result:
Kubernetes resources such as Pods, Services, and Deployments were successfully managed and scaled.

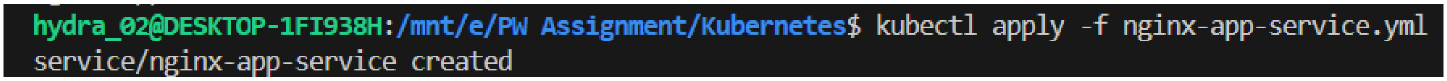
YAML-BASED RESOURCE CONFIGURATION

Deployment YAML Example:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-app-service
spec:
  type: NodePort
  selector:
    app: nginx-app
  ports:
  - port: 80
    targetPort: 80
```

Apply the configuration:

```
kubectl apply -f nginx-app-service.yml
```



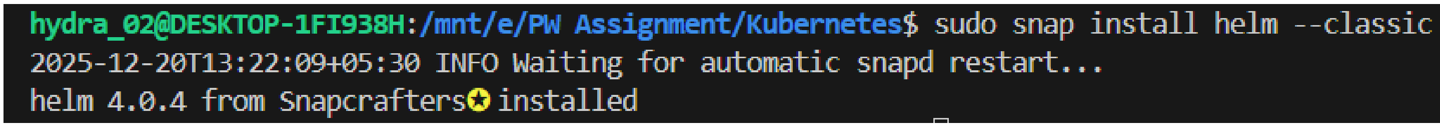
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl apply -f nginx-app-service.yml
service/nginx-app-service created
```

4. Helm Charts: Use Helm to package and deploy applications on Kubernetes?

Objective:
To package and deploy applications using Helm.

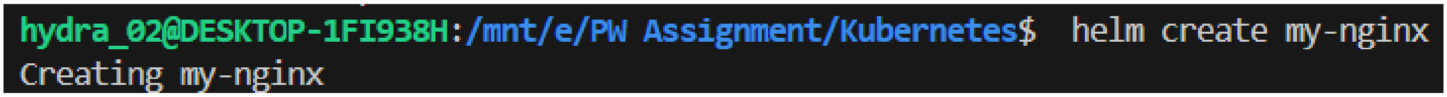
Procedure:

- Install Helm
  sudo snap install helm --classic



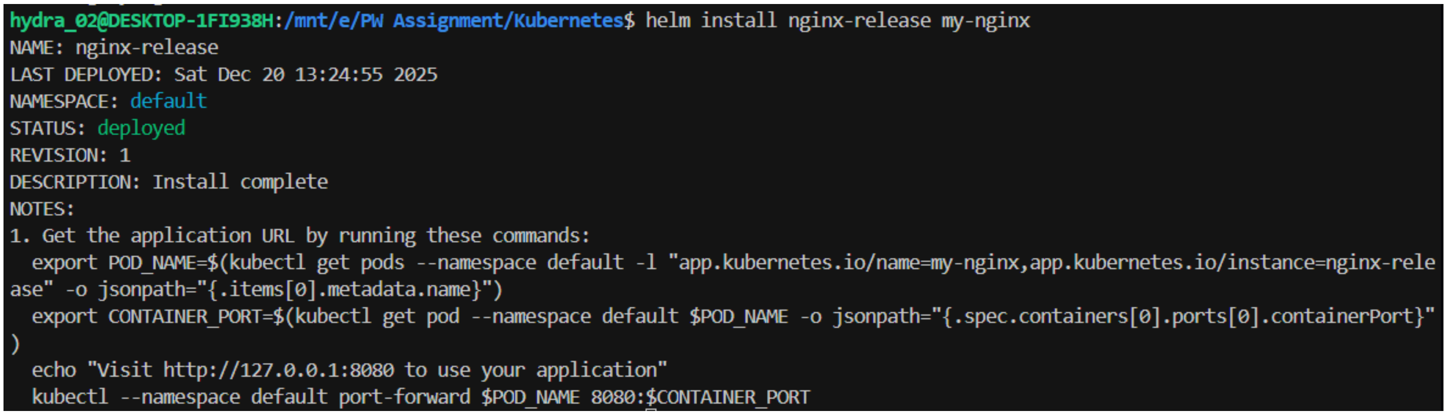
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ sudo snap install helm --classic
2025-12-20T13:22:09+05:30 INFO Waiting for automatic snapd restart...
helm 4.0.4 from Snapcrafters✪ installed
```

1. Create a Helm chart:
  helm create my-nginx



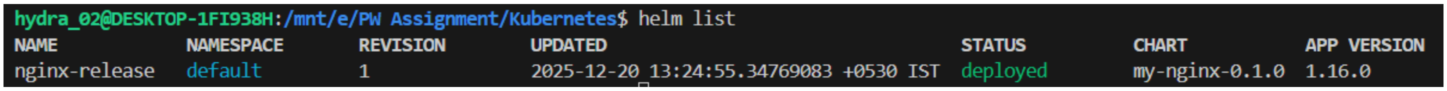
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$  helm create my-nginx
Creating my-nginx
```

2. Install the application using Helm:
  helm install nginx-release my-nginx

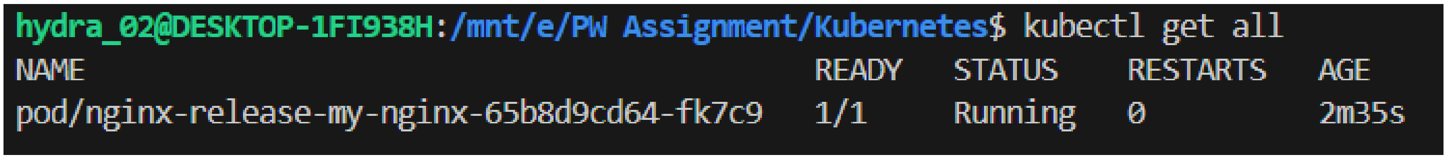


```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ helm install nginx-release my-nginx
NAME: nginx-release
LAST DEPLOYED: Sat Dec 20 13:24:55 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=my-nginx,app.kubernetes.io/instance=nginx-rele
ase" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}"
)
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
```

3. Verify Helm deployment:
  helm list
  kubectl get all



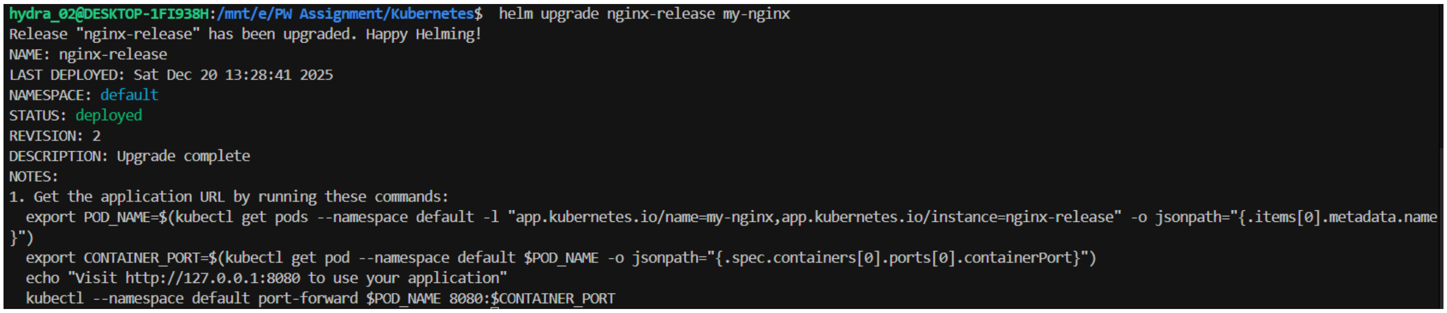
```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ helm list
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART           APP VERSION
nginx-release   default         1               2025-12-20 13:24:55.34769083 +0530 IST  deployed        my-nginx-0.1.0  1.16.0
```



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ kubectl get all
NAME                                       READY   STATUS    RESTARTS   AGE
pod/nginx-release-my-nginx-65b8d9cd64-fk7c9   1/1     Running   0          2m35s
```

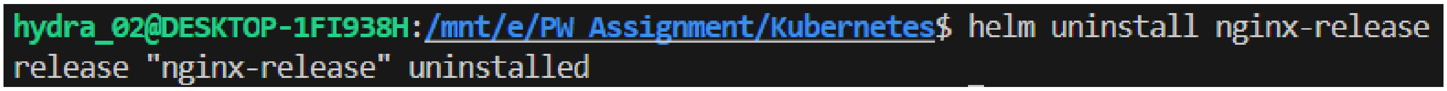
4. Upgrade or uninstall the release:

helm upgrade nginx-release my-nginx



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$  helm upgrade nginx-release my-nginx
Release "nginx-release" has been upgraded. Happy Helming!
NAME: nginx-release
LAST DEPLOYED: Sat Dec 20 13:28:41 2025
NAMESPACE: default
STATUS: deployed
REVISION: 2
DESCRIPTION: Upgrade complete
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=my-nginx,app.kubernetes.io/instance=nginx-release" -o jsonpath="{.items[0].metadata.name
}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
```

helm uninstall nginx-release



```
hydra_02@DESKTOP-1FI938H:/mnt/e/PW Assignment/Kubernetes$ helm uninstall nginx-release
release "nginx-release" uninstalled
```

Result:
The application was successfully packaged and deployed using Helm charts.

CONCLUSION

In this assignment, a local Kubernetes cluster was set up using Minikube.
A simple NGINX application was deployed and managed using Kubernetes.
Pods, Services, and Deployments were created, monitored, and scaled.
Helm charts were used to package and deploy the application efficiently,
demonstrating real-world Kubernetes deployment practices.