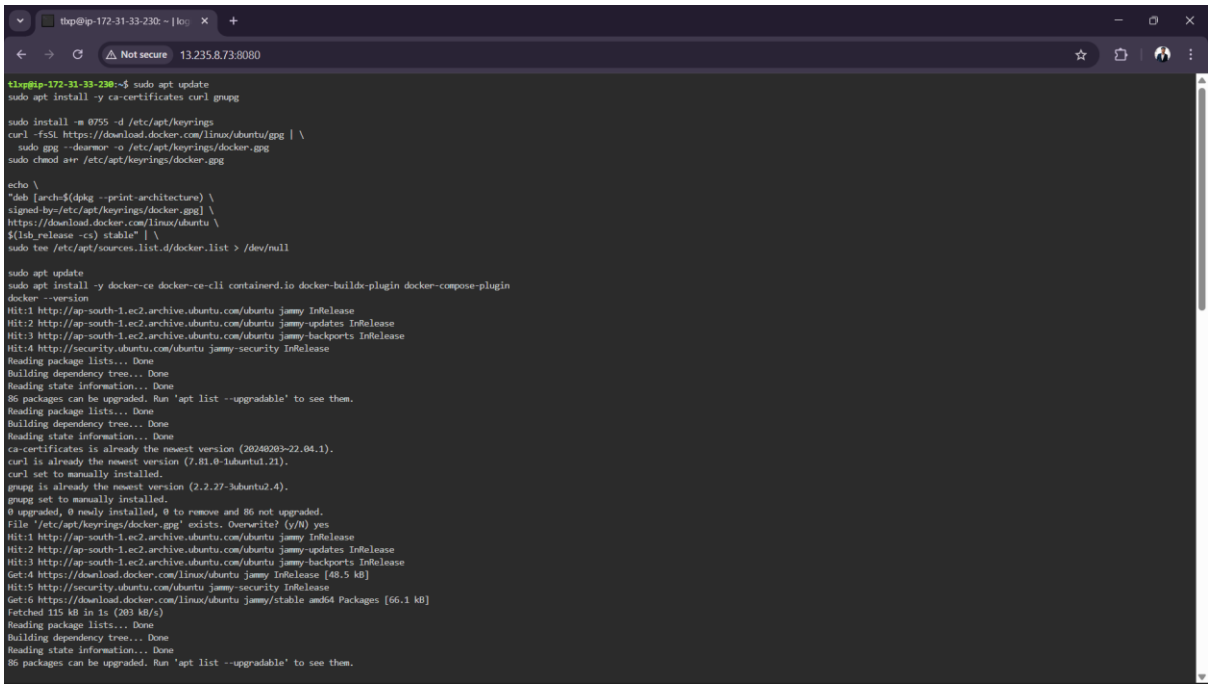
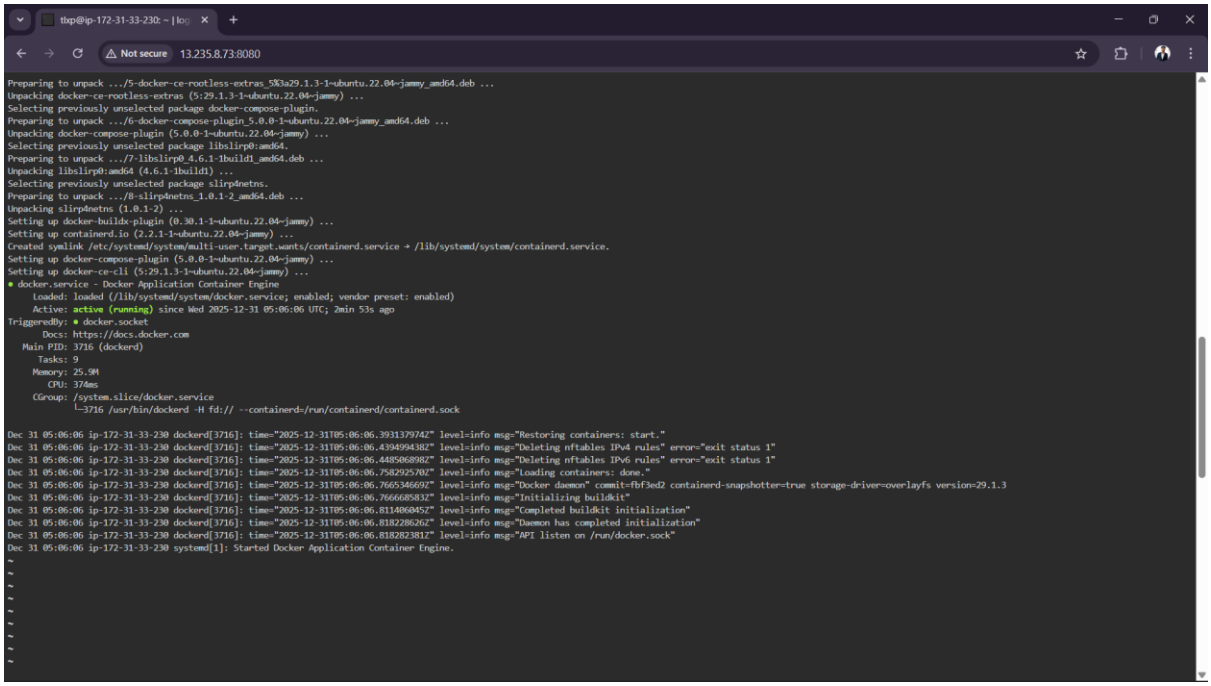
CKA Hands-on Labs

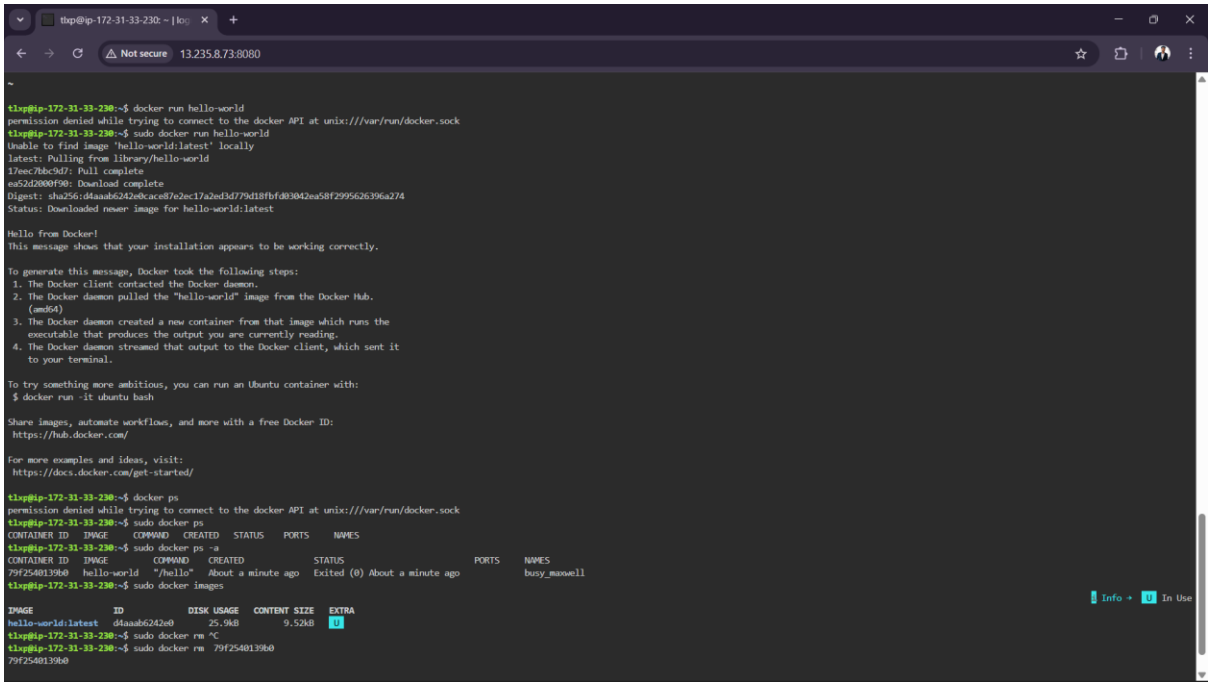
Docker Installation:



2:

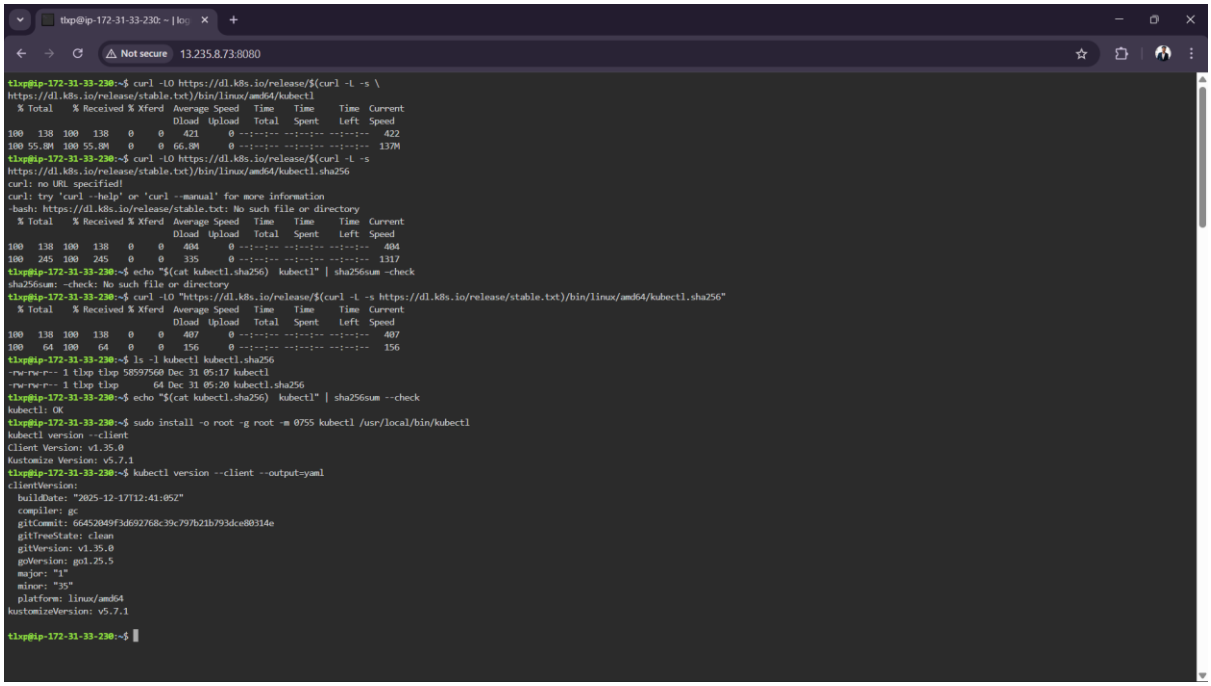


3:

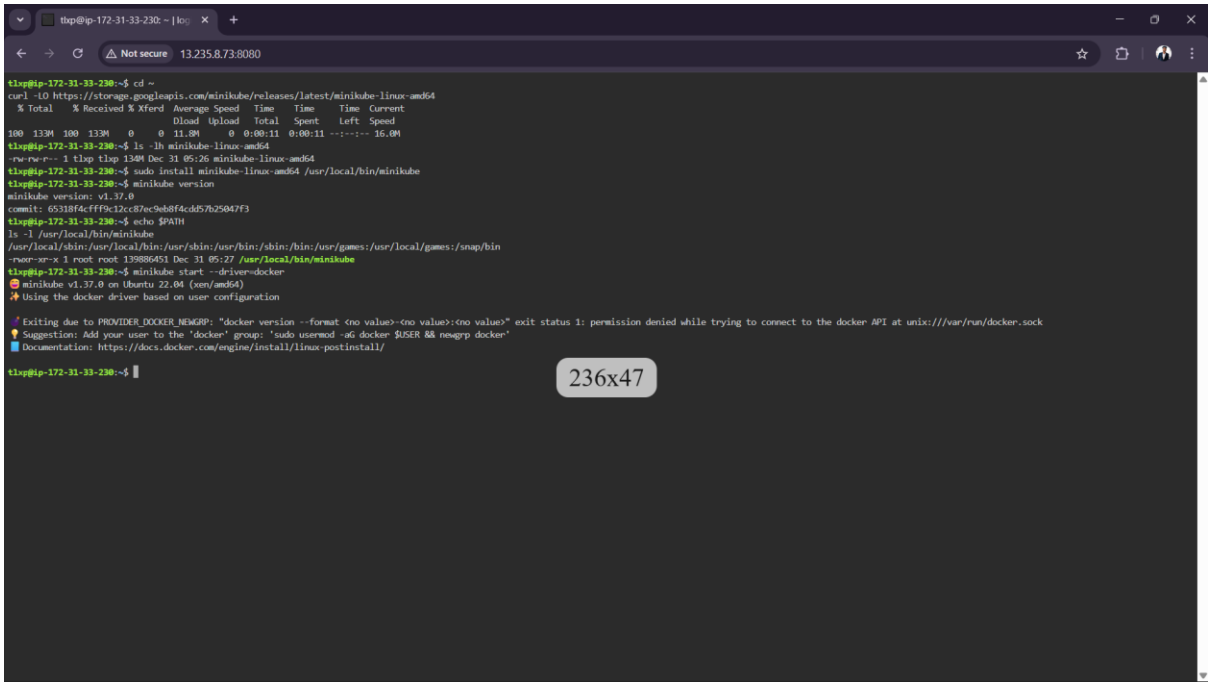


4:

MiniKube Installation:



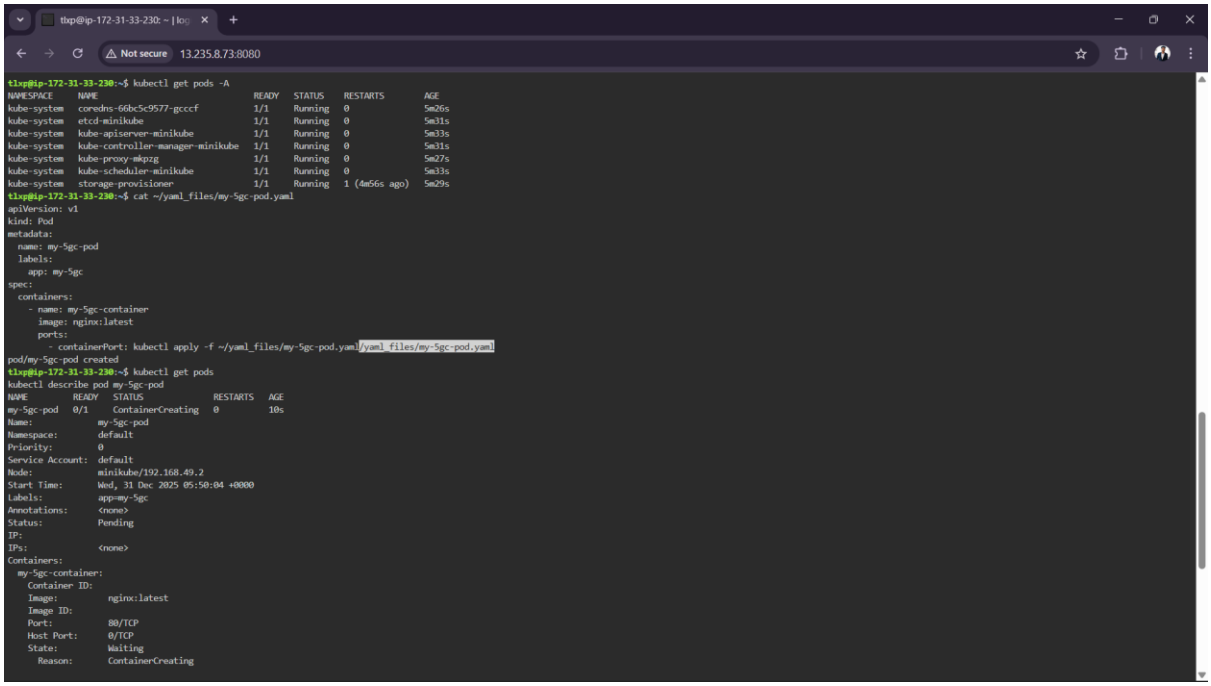
2:



3:

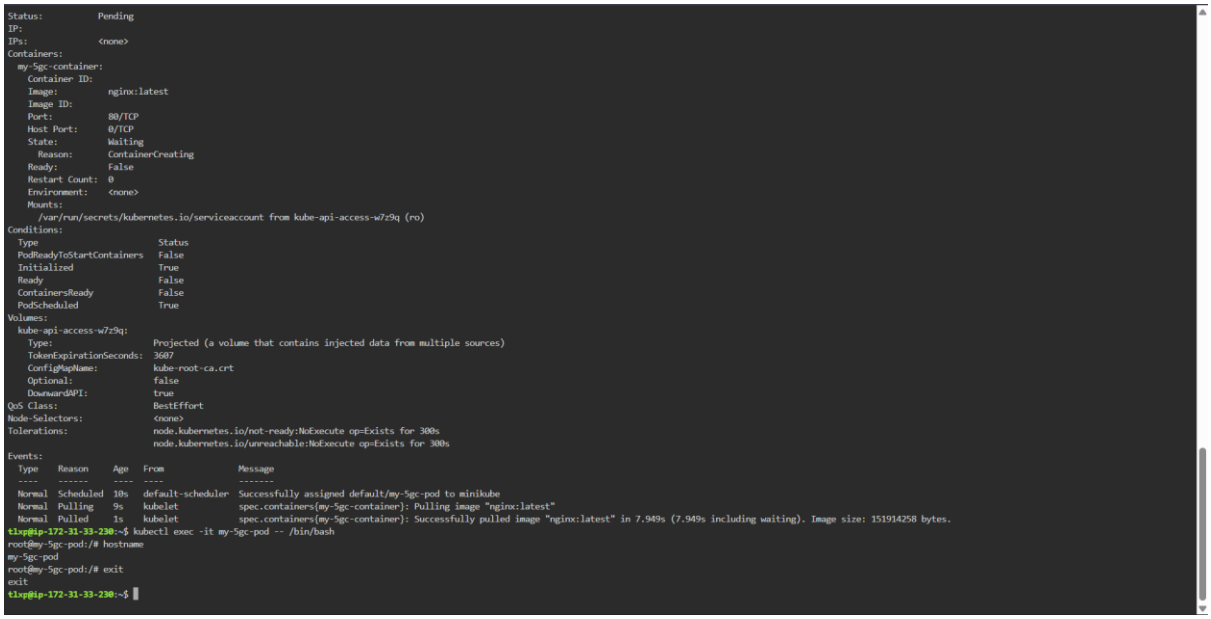
```
 Go version:       go1.25.5
 Git commit:       fbf3ed2
 Built:            Fri Dec 12 14:49:37 2025
 OS/Arch:          linux/amd64
 Experimental:     false
containerd:
 Version:          v2.2.1
 GitCommit:        dea7da592f5d1d2b7755e3a161be07f43fad8f75
runc:
 Version:          1.3.4
 GitCommit:        v1.3.4-0-gd6d73eb8
docker-init:
 Version:          0.19.0
 GitCommit:        de40ad0
tlxp@ip-172-31-33-230:~$ minikube start --driver=docker
😄 minikube v1.37.0 on Ubuntu 22.04 (xen/amd64)
✨ Using the docker driver based on user configuration

🤹 The requested memory allocation of 3072MiB does not leave room for system overhead (total system memory: 3912MiB). You may face stability iss
ues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=3072mb'

📌 Using Docker driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚜 Pulling base image v0.0.48 ...
💾 Downloading Kubernetes v1.34.0 preload ...
    > gcr.io/k8s-minikube/kicbase...:  488.52 MiB / 488.52 MiB  100.00% 23.18 M
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 15.04 M
🔥 Creating docker container (CPUs=2, Memory=3072MB) ...
🐳 Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔎 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
tlxp@ip-172-31-33-230:~$ minikube status
kubectl get nodes
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   60s   v1.34.0
tlxp@ip-172-31-33-230:~$
```

Create a Pod from YAML:



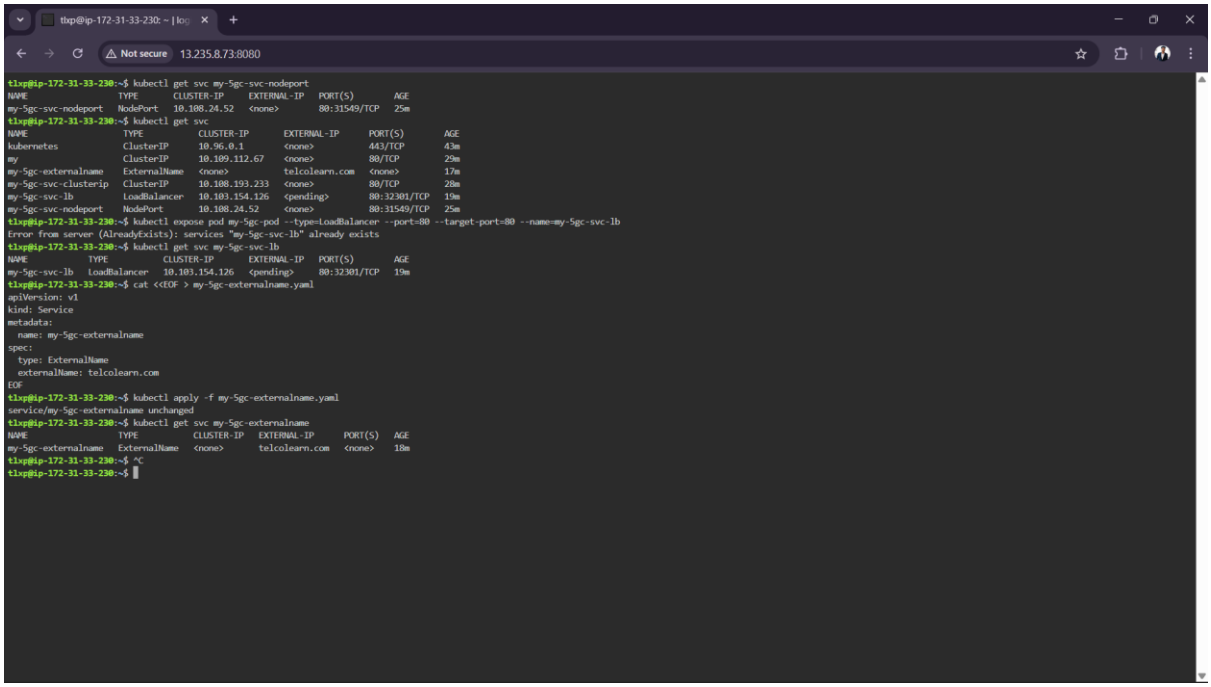
```
tlxp@ip-172-31-33-230:~$ kubectl get pods -A
NAMESPACE     NAME                                READY   STATUS    RESTARTS      AGE
kube-system   coredns-66bc5c9577-gcccf            1/1     Running   0             5m26s
kube-system   etcd-minikube                       1/1     Running   0             5m31s
kube-system   kube-apiserver-minikube             1/1     Running   0             5m33s
kube-system   kube-controller-manager-minikube    1/1     Running   0             5m31s
kube-system   kube-proxy-mkpzg                    1/1     Running   0             5m27s
kube-system   kube-scheduler-minikube             1/1     Running   0             5m33s
kube-system   storage-provisioner                 1/1     Running   1 (4m56s ago)   5m29s
tlxp@ip-172-31-33-230:~$ cat ~/yaml_files/my-5gc-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-5gc-pod
  labels:
    app: my-5gc
spec:
  containers:
    - name: my-5gc-container
      image: nginx:latest
      ports:
        - containerPort: kubectl apply -f ~/yaml_files/my-5gc-pod.yaml/yaml_files/my-5gc-pod.yaml
pod/my-5gc-pod created
tlxp@ip-172-31-33-230:~$ kubectl get pods
kubectl describe pod my-5gc-pod
NAME         READY   STATUS              RESTARTS   AGE
my-5gc-pod   0/1     ContainerCreating   0          10s
Name:             my-5gc-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Wed, 31 Dec 2025 05:50:04 +0000
Labels:           app=my-5gc
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  my-5gc-container:
    Container ID:
    Image:          nginx:latest
    Image ID:
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       ContainerCreating
```

2:

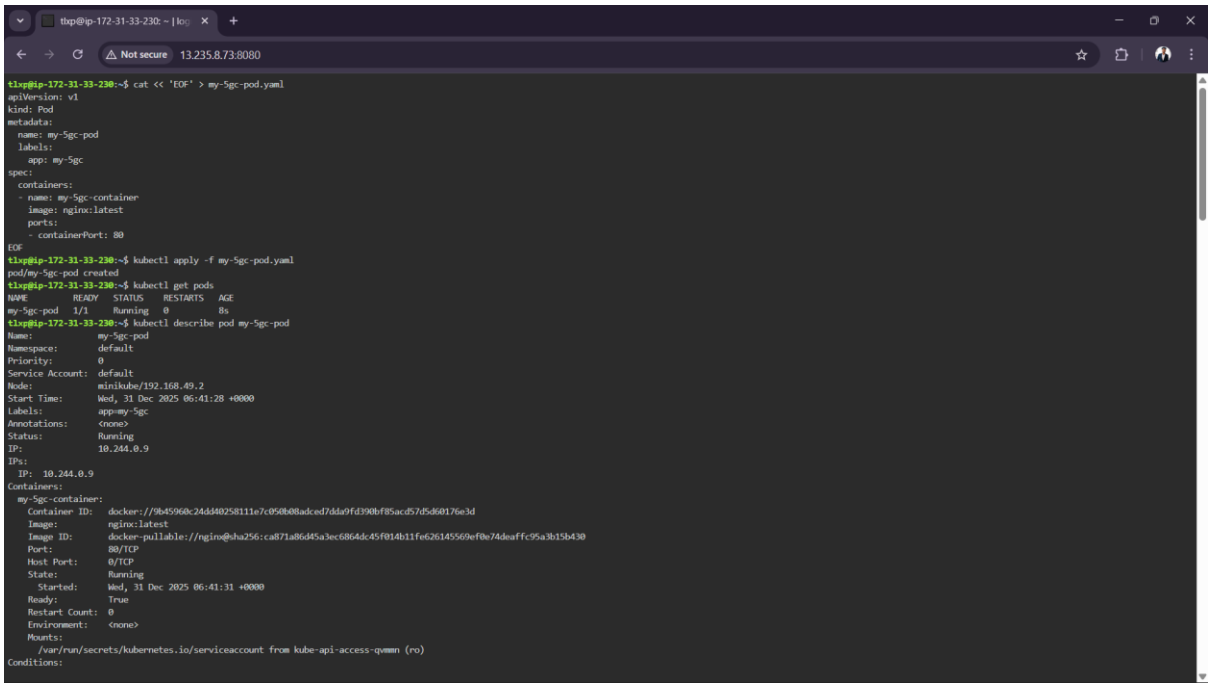


```
Status:           Pending
IP:
IPs:              <none>
Containers:
  my-5gc-container:
    Container ID:
    Image:          nginx:latest
    Image ID:
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-w7z9q (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   False
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  kube-api-access-w7z9q:
    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  10s   default-scheduler  Successfully assigned default/my-5gc-pod to minikube
  Normal  Pulling    9s    kubelet            spec.containers{my-5gc-container}: Pulling image "nginx:latest"
  Normal  Pulled     1s    kubelet            spec.containers{my-5gc-container}: Successfully pulled image "nginx:latest" in 7.949s (7.949s including waiting). Image size: 151914258 bytes.
tlxp@ip-172-31-33-230:~$ kubectl exec -it my-5gc-pod -- /bin/bash
root@my-5gc-pod:/# hostname
my-5gc-pod
root@my-5gc-pod:/# exit
exit
tlxp@ip-172-31-33-230:~$
```

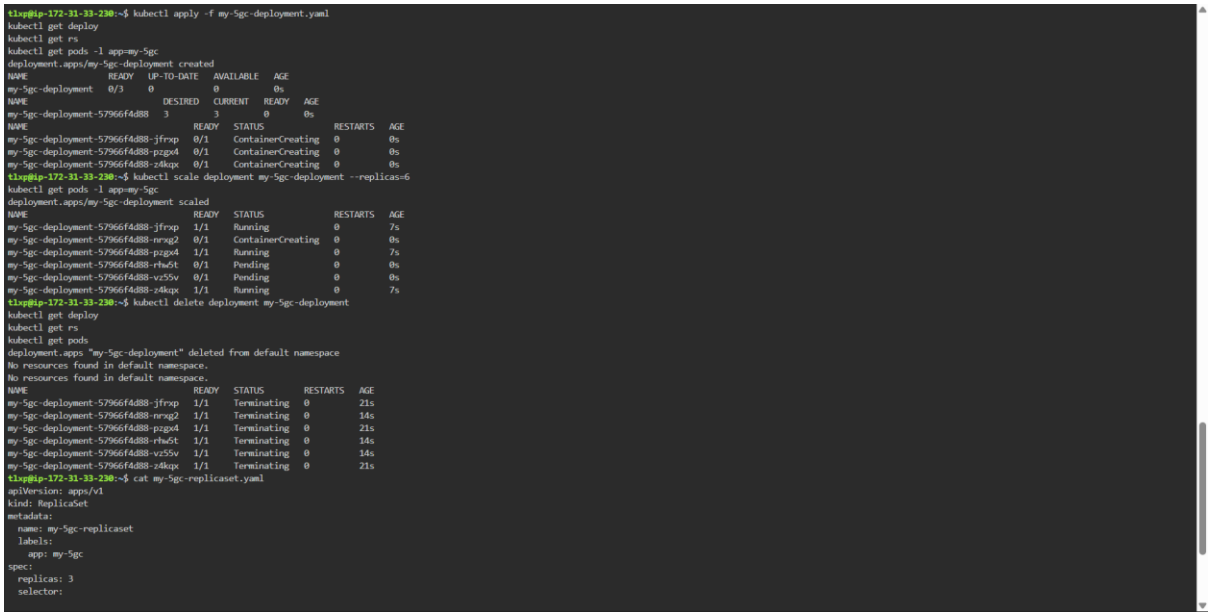
Exploring K8s Services:



K8s Workloads:

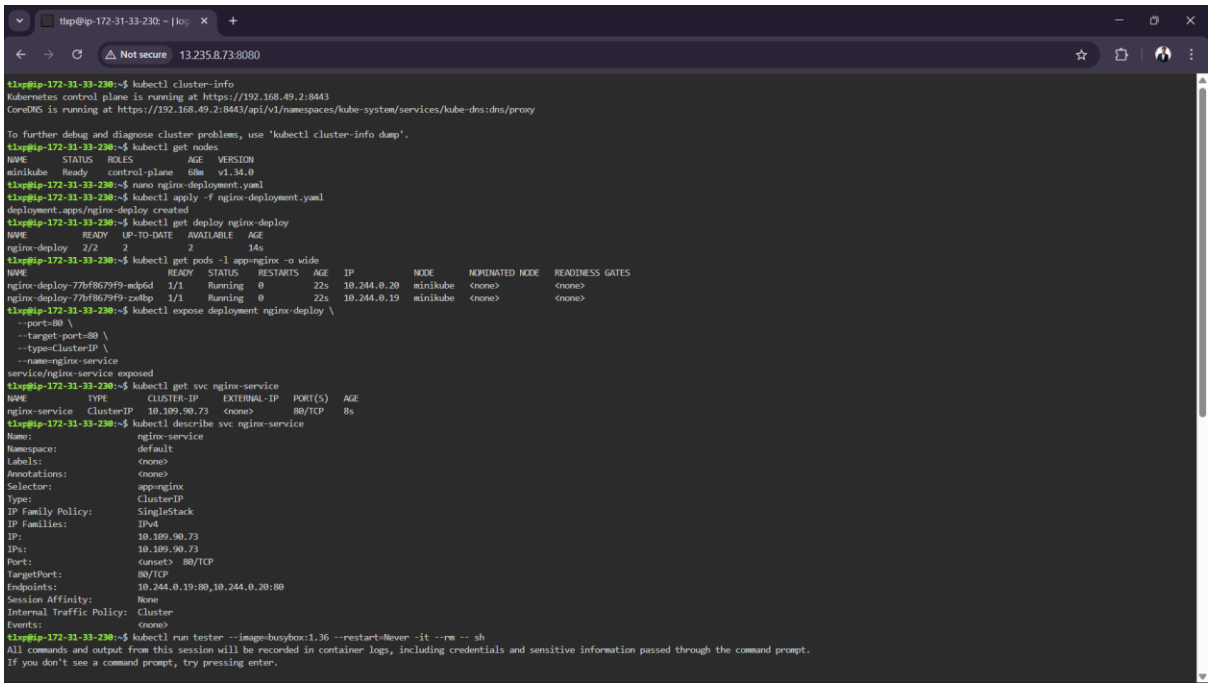


2:



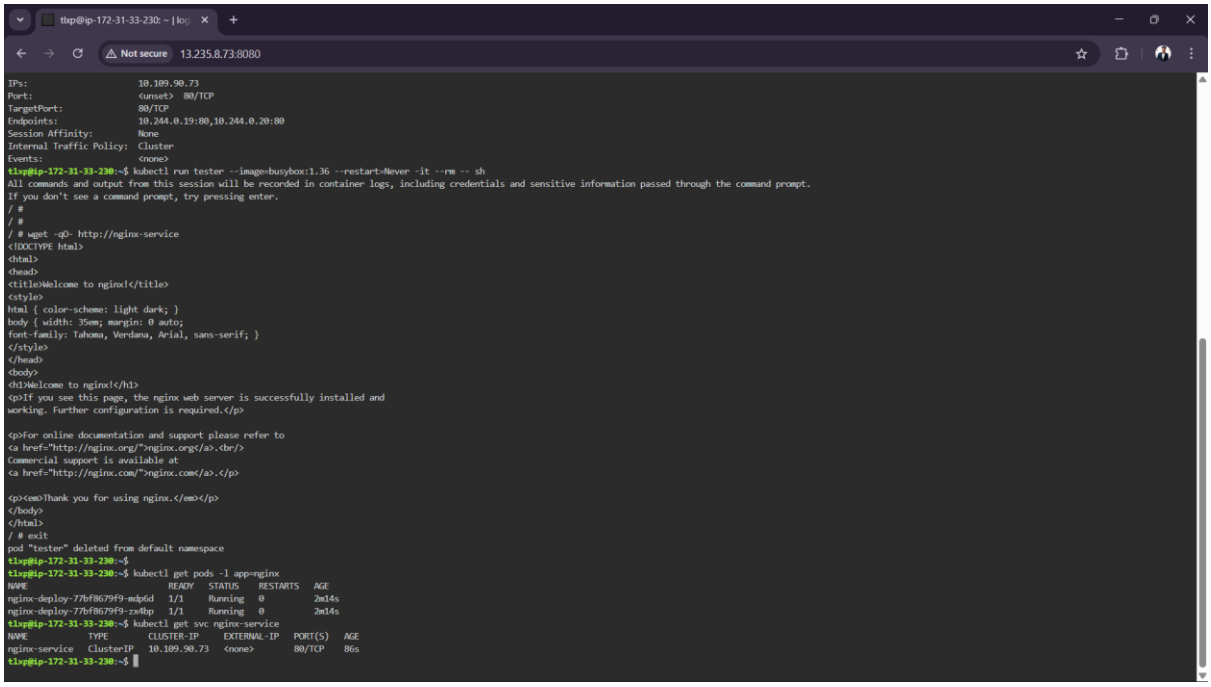
```
tlxp@ip-172-31-33-230:~$ kubectl apply -f my-5gc-deployment.yaml
kubectl get deploy
kubectl get rs
kubectl get pods -l app=my-5gc
deployment.apps/my-5gc-deployment created
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
my-5gc-deployment   0/3     0            0           0s
NAME                           DESIRED   CURRENT   READY   AGE
my-5gc-deployment-57966f4d88   3         3         0       0s
NAME                                 READY   STATUS              RESTARTS   AGE
my-5gc-deployment-57966f4d88-jfrxp   0/1     ContainerCreating   0          0s
my-5gc-deployment-57966f4d88-pzgx4   0/1     ContainerCreating   0          0s
my-5gc-deployment-57966f4d88-z4kqx   0/1     ContainerCreating   0          0s
tlxp@ip-172-31-33-230:~$ kubectl scale deployment my-5gc-deployment --replicas=6
kubectl get pods -l app=my-5gc
deployment.apps/my-5gc-deployment scaled
NAME                                 READY   STATUS              RESTARTS   AGE
my-5gc-deployment-57966f4d88-jfrxp   1/1     Running             0          7s
my-5gc-deployment-57966f4d88-nrxg2   0/1     ContainerCreating   0          0s
my-5gc-deployment-57966f4d88-pzgx4   1/1     Running             0          7s
my-5gc-deployment-57966f4d88-rhw5t   0/1     Pending             0          0s
my-5gc-deployment-57966f4d88-vz55v   0/1     Pending             0          0s
my-5gc-deployment-57966f4d88-z4kqx   1/1     Running             0          7s
tlxp@ip-172-31-33-230:~$ kubectl delete deployment my-5gc-deployment
kubectl get deploy
kubectl get rs
kubectl get pods
deployment.apps "my-5gc-deployment" deleted from default namespace
No resources found in default namespace.
No resources found in default namespace.
NAME                                 READY   STATUS        RESTARTS   AGE
my-5gc-deployment-57966f4d88-jfrxp   1/1     Terminating   0          21s
my-5gc-deployment-57966f4d88-nrxg2   1/1     Terminating   0          14s
my-5gc-deployment-57966f4d88-pzgx4   1/1     Terminating   0          21s
my-5gc-deployment-57966f4d88-rhw5t   1/1     Terminating   0          14s
my-5gc-deployment-57966f4d88-vz55v   1/1     Terminating   0          14s
my-5gc-deployment-57966f4d88-z4kqx   1/1     Terminating   0          21s
tlxp@ip-172-31-33-230:~$ cat my-5gc-replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: my-5gc-replicaset
  labels:
    app: my-5gc
spec:
  replicas: 3
  selector:
```

Kubernetes Networking Exercise:



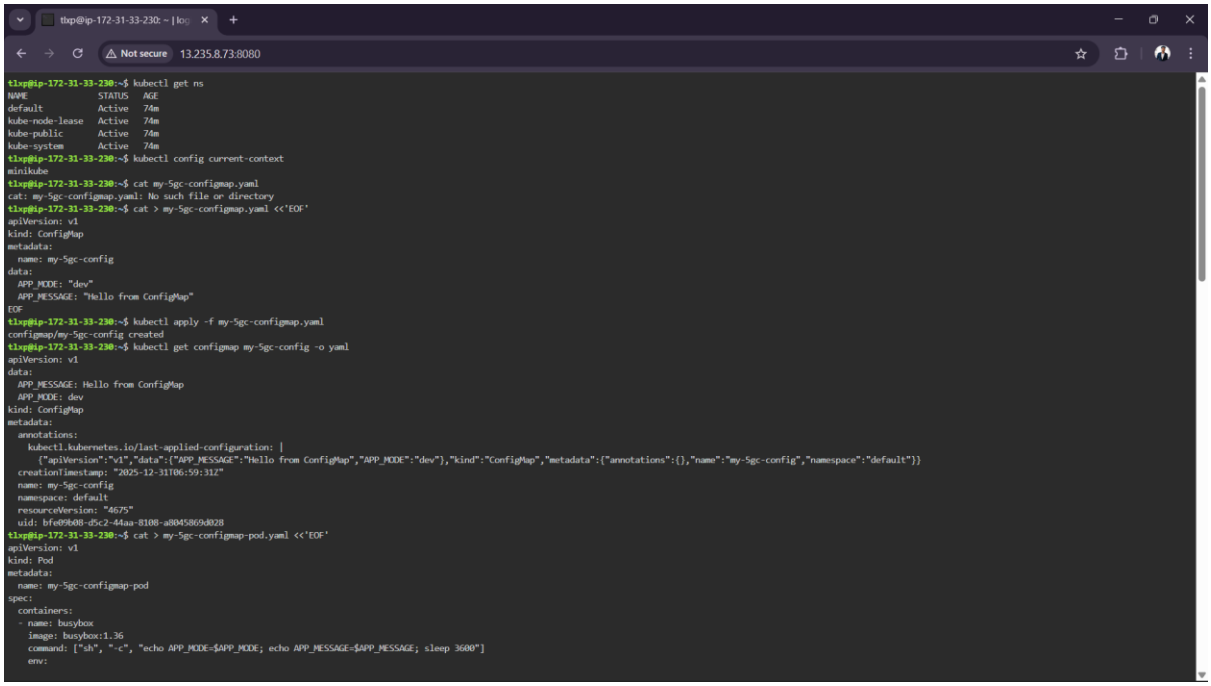
```
tlxp@ip-172-31-33-230: ~ | log
Not secure 13.235.8.73:8080

tlxp@ip-172-31-33-230:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
tlxp@ip-172-31-33-230:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   68m   v1.34.0
tlxp@ip-172-31-33-230:~$ nano nginx-deployment.yaml
tlxp@ip-172-31-33-230:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deploy created
tlxp@ip-172-31-33-230:~$ kubectl get deploy nginx-deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy   2/2     2            2           14s
tlxp@ip-172-31-33-230:~$ kubectl get pods -l app=nginx -o wide
NAME                            READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE   READINESS GATES
nginx-deploy-77bf8679f9-mdp6d   1/1     Running   0          22s   10.244.0.20   minikube   <none>           <none>
nginx-deploy-77bf8679f9-zx4bp   1/1     Running   0          22s   10.244.0.19   minikube   <none>           <none>
tlxp@ip-172-31-33-230:~$ kubectl expose deployment nginx-deploy \
  --port=80 \
  --target-port=80 \
  --type=ClusterIP \
  --name=nginx-service
service/nginx-service exposed
tlxp@ip-172-31-33-230:~$ kubectl get svc nginx-service
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.109.90.73    <none>        80/TCP    8s
tlxp@ip-172-31-33-230:~$ kubectl describe svc nginx-service
Name:                     nginx-service
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=nginx
Type:                     ClusterIP
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.109.90.73
IPs:                      10.109.90.73
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
Endpoints:                10.244.0.19:80,10.244.0.20:80
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
tlxp@ip-172-31-33-230:~$ kubectl run tester --image=busybox:1.36 --restart=Never -it --rm -- sh
All commands and output from this session will be recorded in container logs, including credentials and sensitive information passed through the command prompt.
If you don't see a command prompt, try pressing enter.
```

2:



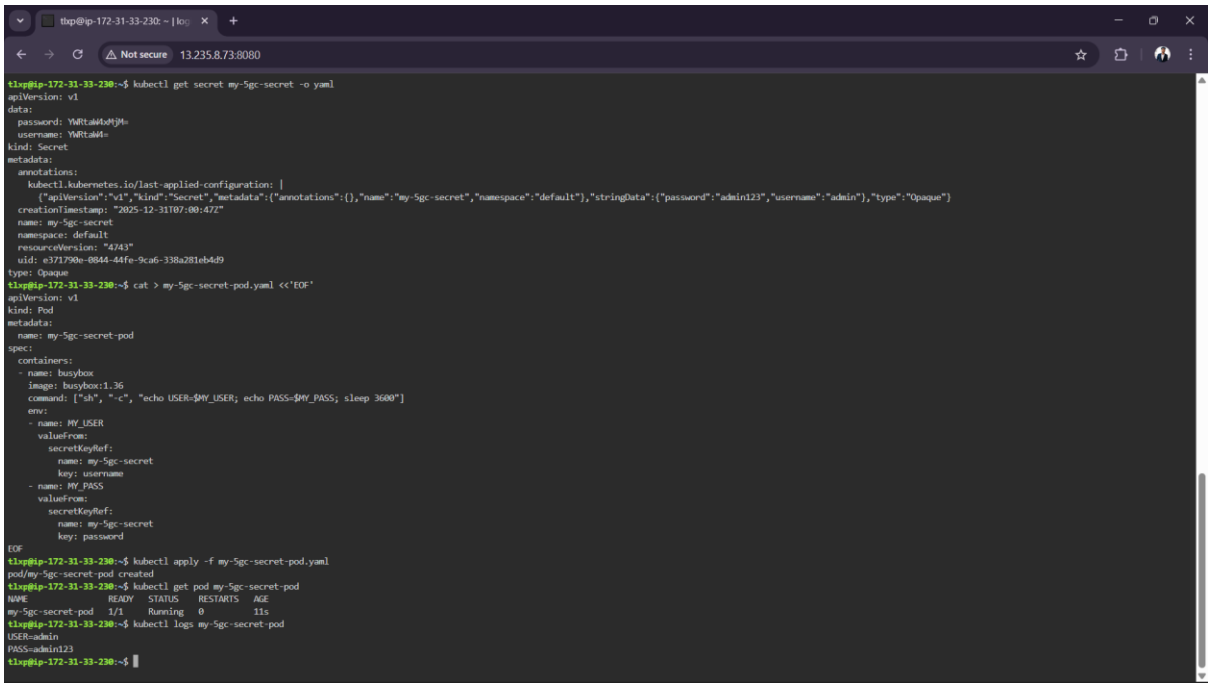
```
tlxp@ip-172-31-33-230: ~ | log
Not secure 13.235.8.73:8080

IPs:                      10.109.90.73
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
Endpoints:                10.244.0.19:80,10.244.0.20:80
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
tlxp@ip-172-31-33-230:~$ kubectl run tester --image=busybox:1.36 --restart=Never -it --rm -- sh
All commands and output from this session will be recorded in container logs, including credentials and sensitive information passed through the command prompt.
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ # wget -qO- http://nginx-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ # exit
pod "tester" deleted from default namespace
tlxp@ip-172-31-33-230:~$
tlxp@ip-172-31-33-230:~$ kubectl get pods -l app=nginx
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-77bf8679f9-mdp6d   1/1     Running   0          2m14s
nginx-deploy-77bf8679f9-zx4bp   1/1     Running   0          2m14s
tlxp@ip-172-31-33-230:~$ kubectl get svc nginx-service
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.109.90.73    <none>        80/TCP    86s
tlxp@ip-172-31-33-230:~$
```

ConfigMap and Secrets:



```
tlxp@ip-172-31-33-230:~$ kubectl get ns
NAME              STATUS   AGE
default           Active   74m
kube-node-lease   Active   74m
kube-public       Active   74m
kube-system       Active   74m
tlxp@ip-172-31-33-230:~$ kubectl config current-context
minikube
tlxp@ip-172-31-33-230:~$ cat my-5gc-configmap.yaml
cat: my-5gc-configmap.yaml: No such file or directory
tlxp@ip-172-31-33-230:~$ cat > my-5gc-configmap.yaml <<'EOF'
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-5gc-config
data:
  APP_MODE: "dev"
  APP_MESSAGE: "Hello from ConfigMap"
EOF
tlxp@ip-172-31-33-230:~$ kubectl apply -f my-5gc-configmap.yaml
configmap/my-5gc-config created
tlxp@ip-172-31-33-230:~$ kubectl get configmap my-5gc-config -o yaml
apiVersion: v1
data:
  APP_MESSAGE: Hello from ConfigMap
  APP_MODE: dev
kind: ConfigMap
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"APP_MESSAGE":"Hello from ConfigMap","APP_MODE":"dev"},"kind":"ConfigMap","metadata":{"annotations":{},"name":"my-5gc-config","namespace":"default"}}
  creationTimestamp: "2025-12-31T06:59:31Z"
  name: my-5gc-config
  namespace: default
  resourceVersion: "4675"
  uid: bfe09b08-d5c2-44aa-8108-a8045869d028
tlxp@ip-172-31-33-230:~$ cat > my-5gc-configmap-pod.yaml <<'EOF'
apiVersion: v1
kind: Pod
metadata:
  name: my-5gc-configmap-pod
spec:
  containers:
  - name: busybox
    image: busybox:1.36
    command: ["sh", "-c", "echo APP_MODE=$APP_MODE; echo APP_MESSAGE=$APP_MESSAGE; sleep 3600"]
    env:
```

2:



```
tlxp@ip-172-31-33-230:~$ kubectl get secret my-5gc-secret -o yaml
apiVersion: v1
data:
  password: YWRtaW4xMjM=
  username: YWRtaW4=
kind: Secret
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Secret","metadata":{"annotations":{},"name":"my-5gc-secret","namespace":"default"},"stringData":{"password":"admin123","username":"admin"},"type":"Opaque"}
  creationTimestamp: "2025-12-31T07:00:47Z"
  name: my-5gc-secret
  namespace: default
  resourceVersion: "4743"
  uid: e371790e-0844-44fe-9ca6-338a281eb4d9
type: Opaque
tlxp@ip-172-31-33-230:~$ cat > my-5gc-secret-pod.yaml <<'EOF'
apiVersion: v1
kind: Pod
metadata:
  name: my-5gc-secret-pod
spec:
  containers:
  - name: busybox
    image: busybox:1.36
    command: ["sh", "-c", "echo USER=$MY_USER; echo PASS=$MY_PASS; sleep 3600"]
    env:
    - name: MY_USER
      valueFrom:
        secretKeyRef:
          name: my-5gc-secret
          key: username
    - name: MY_PASS
      valueFrom:
        secretKeyRef:
          name: my-5gc-secret
          key: password
EOF
tlxp@ip-172-31-33-230:~$ kubectl apply -f my-5gc-secret-pod.yaml
pod/my-5gc-secret-pod created
tlxp@ip-172-31-33-230:~$ kubectl get pod my-5gc-secret-pod
NAME               READY   STATUS    RESTARTS   AGE
my-5gc-secret-pod  1/1     Running   0          11s
tlxp@ip-172-31-33-230:~$ kubectl logs my-5gc-secret-pod
USER=admin
PASS=admin123
tlxp@ip-172-31-33-230:~$
```