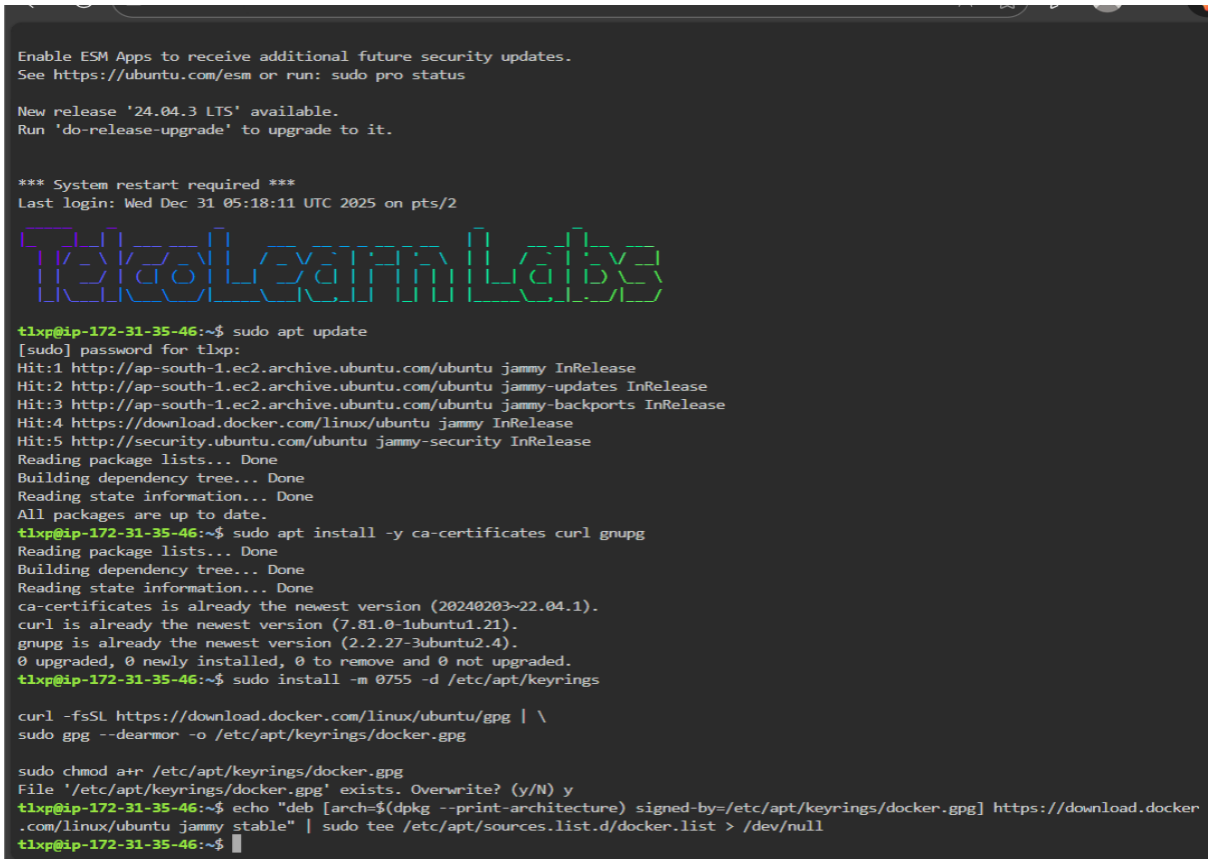
CKA Hands-on Labs

Step 1 : Install Docker on ubuntu(AWS VM)



```
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
tlxp@ip-172-31-35-46:~$ sudo apt install -y ca-certificates curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.21).
gnupg is already the newest version (2.2.27-3ubuntu2.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
tlxp@ip-172-31-35-46:~$ sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
File '/etc/apt/keyrings/docker.gpg' exists. Overwrite? (y/N) y
tlxp@ip-172-31-35-46:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker
.com/linux/ubuntu jammy stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
tlxp@ip-172-31-35-46:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
tlxp@ip-172-31-35-46:~$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
containerd.io is already the newest version (2.2.1-1~ubuntu.22.04~jammy).
docker-buildx-plugin is already the newest version (0.30.1-1~ubuntu.22.04~jammy).
docker-ce-cli is already the newest version (5:29.1.3-1~ubuntu.22.04~jammy).
docker-ce is already the newest version (5:29.1.3-1~ubuntu.22.04~jammy).
docker-compose-plugin is already the newest version (5.0.0-1~ubuntu.22.04~jammy).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
tlxp@ip-172-31-35-46:~$ docker --version
Docker version 29.1.3, build f52814d
tlxp@ip-172-31-35-46:~$
```

Step 2: Add User to the Docker Group

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
tlxp@ip-172-31-35-46:~$ docker --version
Docker version 29.1.3, build f52814d
tlxp@ip-172-31-35-46:~$ sudo usermod -aG docker $USER
tlxp@ip-172-31-35-46:~$ newgrp docker

tlxp@ip-172-31-35-46:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-12-31 05:09:44 UTC; 38min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 23223 (dockerd)
      Tasks: 9
     Memory: 23.9M
        CPU: 579ms
     CGroup: /system.slice/docker.service
             └─23223 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.605256758Z" level=info msg="Restoring containers: start."
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.652250827Z" level=info msg="Deleting nftables IPv4 rules">
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.659046396Z" level=info msg="Deleting nftables IPv6 rules">
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.977663429Z" level=info msg="Loading containers: done."
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.986270445Z" level=info msg="Docker daemon" commit=fbf3ed2>
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.986428366Z" level=info msg="Initializing buildkit"
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.009250490Z" level=info msg="Completed buildkit initializa>
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.015880330Z" level=info msg="Daemon has completed initiali>
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.015940672Z" level=info msg="API listen on /run/docker.soc>
Dec 31 05:09:44 ip-172-31-35-46 systemd[1]: Started Docker Application Container Engine.
lines 1-22/22 (END)
```

Step 3 : Test Docker with Some Basic Commands

```
tlxp@ip-172-31-35-46:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-12-31 05:09:44 UTC; 38min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 23223 (dockerd)
      Tasks: 9
     Memory: 23.9M
        CPU: 579ms
     CGroup: /system.slice/docker.service
             └─23223 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.605256758Z" level=info msg="Restoring containers: start."
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.652250827Z" level=info msg="Deleting nftables IPv4 rules">
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.659046396Z" level=info msg="Deleting nftables IPv6 rules">
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.977663429Z" level=info msg="Loading containers: done."
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.986270445Z" level=info msg="Docker daemon" commit=fbf3ed2>
Dec 31 05:09:43 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:43.986428366Z" level=info msg="Initializing buildkit"
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.009250490Z" level=info msg="Completed buildkit initializa>
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.015880330Z" level=info msg="Daemon has completed initiali>
Dec 31 05:09:44 ip-172-31-35-46 dockerd[23223]: time="2025-12-31T05:09:44.015940672Z" level=info msg="API listen on /run/docker.soc>
Dec 31 05:09:44 ip-172-31-35-46 systemd[1]: Started Docker Application Container Engine.

tlxp@ip-172-31-35-46:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
ea52d2000f90: Download complete
Digest: sha256:d4aaab6242e0cace87e2ec17a2ed3d779d18fbfd03042ea58f2995626396a274
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
```

```
tlxp@ip-172-31-35-46:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
tlxp@ip-172-31-35-46:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
6785bdb17070   hello-world   "/hello"   2 minutes ago   Exited (0) 2 minutes ago             sweet_archimedes
tlxp@ip-172-31-35-46:~$ docker images
                                                                                  i Info →  U  In Use
IMAGE               ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest  d4aaab6242e0       25.9kB         9.52kB    U
tlxp@ip-172-31-35-46:~$ docker rm d4aaab6242e0
Error response from daemon: No such container: d4aaab6242e0
tlxp@ip-172-31-35-46:~$ docker 6785bdb17070
docker: unknown command: docker 6785bdb17070

Run 'docker --help' for more information
tlxp@ip-172-31-35-46:~$ docker rm 6785bdb17070
6785bdb17070
tlxp@ip-172-31-35-46:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
tlxp@ip-172-31-35-46:~$ docker rmi d4aaab6242e0
Untagged: hello-world:latest
Deleted: sha256:d4aaab6242e0cace87e2ec17a2ed3d779d18fbfd03042ea58f2995626396a274
tlxp@ip-172-31-35-46:~$
```

Minicube Installation and Validation

```
tlxp@ip-172-31-35-46:~$ curl -LO https://dl.k8s.io/release/$(curl -L -s \https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubec
tl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    406      0 --:--:-- --:--:-- --:--:--   405
100 55.8M  100 55.8M    0     0  22.3M      0  0:00:02  0:00:02 --:--:-- 27.3M
tlxp@ip-172-31-35-46:~$ ^[[200~curl -LO https://dl.k8s.io/release/$(curl -L -s
> ^C
tlxp@ip-172-31-35-46:~$ curl -LO https://dl.k8s.io/release/$(curl -L -s \https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubec
tl.sha256
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    411      0 --:--:-- --:--:-- --:--:--   411
100    64  100    64    0     0    157      0 --:--:-- --:--:-- --:--:--   157
```

```
tlxp@ip-172-31-35-46:~$ curl -LO https://dl.k8s.io/release/$(curl -L -s \https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubec
tl.sha256
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    411      0 --:--:-- --:--:-- --:--:--   411
100    64  100    64    0     0    157      0 --:--:-- --:--:-- --:--:--   157
tlxp@ip-172-31-35-46:~$ echo "$(cat kubectl.sha256)  kubectl" | sha256sum -check
sha256sum: -check: No such file or directory
tlxp@ip-172-31-35-46:~$ echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
sha256sum: --check: No such file or directory
tlxp@ip-172-31-35-46:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubec
tl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    420      0 --:--:-- --:--:-- --:--:--   420
100 55.8M  100 55.8M    0     0  65.2M      0 --:--:-- --:--:-- --:--:--  138M
tlxp@ip-172-31-35-46:~$ ls -l kubectl
-rw-rw-r-- 1 tlxp docker 58597560 Dec 31 06:04 kubectl
tlxp@ip-172-31-35-46:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubec
tl.sha256"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    415      0 --:--:-- --:--:-- --:--:--   414
100    64  100    64    0     0    158      0 --:--:-- --:--:-- --:--:--   158
tlxp@ip-172-31-35-46:~$ echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
kubectl: OK
tlxp@ip-172-31-35-46:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[sudo] password for tlxp:
tlxp@ip-172-31-35-46:~$ kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1
tlxp@ip-172-31-35-46:~$ kubectl version --client --output=yaml
clientVersion:
  buildDate: "2025-12-17T12:41:05Z"
  compiler: gc
  gitCommit: 66452049f3d692768c39c797b21b793dce80314e
  gitTreeState: clean
  gitVersion: v1.35.0
  goVersion: go1.25.5
  major: "1"
  minor: "35"
  platform: linux/amd64
kustomizeVersion: v5.7.1
```

Install Minicube on Ubuntu

```
tlxp@ip-172-31-35-46:~$ docker --version
Docker version 29.1.3, build f52814d
tlxp@ip-172-31-35-46:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  133M  100  133M    0     0  7940k      0  0:00:17  0:00:17 --:--:-- 9064k
tlxp@ip-172-31-35-46:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
tlxp@ip-172-31-35-46:~$ minikube version
minikube version: v1.37.0
commit: 65318f4cfff9c12cc87ec9eb8f4cdd57b25047f3
tlxp@ip-172-31-35-46:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  133M  100  133M    0     0  11.5M      0  0:00:11  0:00:11 --:--:-- 14.9M
tlxp@ip-172-31-35-46:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
tlxp@ip-172-31-35-46:~$ minikube version
minikube version: v1.37.0
commit: 65318f4cfff9c12cc87ec9eb8f4cdd57b25047f3
tlxp@ip-172-31-35-46:~$ 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 s
tability issues.
💡 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...:  488.51 MiB / 488.52 MiB  100.00% 23.61 M
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 14.99 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-172-31-35-46:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   17s   v1.34.0
tlxp@ip-172-31-35-46:~$
```

Checking basic configuration of Kubernetes

```
🏄 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
tlxp@ip-172-31-35-46:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   17s   v1.34.0
tlxp@ip-172-31-35-46:~$ kubectl get pods
No resources found in default namespace.
tlxp@ip-172-31-35-46:~$ kubectl get pods --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
kube-system   coredns-66bc5c9577-9plh8           1/1     Running   0             112s
kube-system   etcd-minikube                      1/1     Running   0             117s
kube-system   kube-apiserver-minikube            1/1     Running   0             117s
kube-system   kube-controller-manager-minikube   1/1     Running   0             117s
kube-system   kube-proxy-n9df9                   1/1     Running   0             112s
kube-system   kube-scheduler-minikube            1/1     Running   0             117s
kube-system   storage-provisioner                1/1     Running   1 (81s ago)   115s
tlxp@ip-172-31-35-46:~$ kubectl get pods -o wide --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS        AGE     IP             NODE       NOMINATED NODE
  READINESS GATES
kube-system   coredns-66bc5c9577-9plh8           1/1     Running   0               4m54s   10.244.0.2     minikube   <none>
  <none>
kube-system   etcd-minikube                      1/1     Running   0               4m59s   192.168.49.2   minikube   <none>
  <none>
kube-system   kube-apiserver-minikube            1/1     Running   0               4m59s   192.168.49.2   minikube   <none>
  <none>
kube-system   kube-controller-manager-minikube   1/1     Running   0               4m59s   192.168.49.2   minikube   <none>
  <none>
kube-system   kube-proxy-n9df9                   1/1     Running   0               4m54s   192.168.49.2   minikube   <none>
  <none>
kube-system   kube-scheduler-minikube            1/1     Running   0               4m59s   192.168.49.2   minikube   <none>
  <none>
kube-system   storage-provisioner                1/1     Running   1 (4m23s ago)   4m57s   192.168.49.2   minikube   <none>
  <none>
tlxp@ip-172-31-35-46:~$ alias k=kubectl
tlxp@ip-172-31-35-46:~$ k get pods
k get pods -A
k get nodes
No resources found in default namespace.
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
kube-system   coredns-66bc5c9577-9plh8           1/1     Running   0             5m32s
kube-system   etcd-minikube                      1/1     Running   0             5m37s
kube-system   kube-apiserver-minikube            1/1     Running   0             5m37s
kube-system   kube-controller-manager-minikube   1/1     Running   0             5m37s
kube-system   kube-proxy-n9df9                   1/1     Running   0             5m32s
kube-system   kube-scheduler-minikube            1/1     Running   0             5m37s
kube-system   storage-provisioner                1/1     Running   1 (5m1s ago)  5m35s
```

Create pod from YAML, inspecting the pod and verify

```
tlxp@ip-172-31-35-46:~$ 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: 80tlxp@ip-172-31-35-46:~$ kubectl apply -f ~/yaml_files/my-5gc-pod.yaml
pod/my-5gc-pod created
tlxp@ip-172-31-35-46:~$ kubectl get pods
NAME         READY   STATUS              RESTARTS   AGE
my-5gc-pod   0/1     ContainerCreating   0          10s
```

Check pod Details

```
tlxp@ip-172-31-35-46:~$ kubectl describe pod my-5gc-pod
Name:             my-5gc-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Wed, 31 Dec 2025 06:21:18 +0000
Labels:           app=my-5gc
Annotations:      <none>
Status:           Running
IP:               10.244.0.3
IPs:
  IP:  10.244.0.3
Containers:
  my-5gc-container:
    Container ID:   docker://0b6bf63542d5f5802ace17a01c0722c09ccc913c2b3ec06bd685b11eaacac0df
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:ca871a86d45a3ec6864dc45f014b11fe626145569ef0e74deaffc95a3b15b430
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Wed, 31 Dec 2025 06:21:28 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sxhmj (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-sxhmj:
    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
```

Accessing the Pods

```
tlxp@ip-172-31-35-46:~$ kubectl exec -it my-5gc-pod -- /bin/bash
root@my-5gc-pod:/# hostname
my-5gc-pod
root@my-5gc-pod:/# exit
exit
tlxp@ip-172-31-35-46:~$
```

Exploring K8s Services

Check the Pod is Running

```
tlxp@ip-172-31-35-46:~$ kubectl get pod my-5gc-pod
NAME         READY   STATUS    RESTARTS   AGE
my-5gc-pod   1/1     Running   0          7m2s
```

Exposing via ClusterIP

```
tlxp@ip-172-31-35-46:~$ kubectl expose pod my-5gc-pod \
  --type=ClusterIP \
  --port=80 \
  --target-port=80 \
  --name=my-5gc-svc-clusterip
service/my-5gc-svc-clusterip exposed
tlxp@ip-172-31-35-46:~$ kubectl get svc
NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
kubernetes            ClusterIP   10.96.0.1        <none>        443/TCP   17m
my-5gc-svc-clusterip  ClusterIP   10.105.135.162   <none>        80/TCP    16s
```

Exposing via NodePort

```
tlxp@ip-172-31-35-46:~$ kubectl expose pod my-5gc-pod \
  --type=NodePort \
  --port=80 \
  --target-port=80 \
  --name=my-5gc-svc-nodeport
service/my-5gc-svc-nodeport exposed
tlxp@ip-172-31-35-46:~$ kubectl get svc my-5gc-svc-nodeport
NAME                  TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
my-5gc-svc-nodeport   NodePort   10.109.173.108   <none>        80:31419/TCP   11s
tlxp@ip-172-31-35-46:~$ minikube service my-5gc-svc-nodeport
|-----------|----------------------|-------------|---------------------------|
| NAMESPACE |         NAME         | TARGET PORT |            URL            |
|-----------|----------------------|-------------|---------------------------|
| default   | my-5gc-svc-nodeport  | 80          | http://192.168.49.2:31419 |
|-----------|----------------------|-------------|---------------------------|
🎉  Opening service default/my-5gc-svc-nodeport in default browser...
👉  http://192.168.49.2:31419
```

Exposing via LoadBalancer

```
tlxp@ip-172-31-35-46:~$ kubectl expose pod my-5gc-pod \
  --type=LoadBalancer \
  --port=80 \
  --target-port=80 \
  --name=my-5gc-svc-lb
service/my-5gc-svc-lb exposed
tlxp@ip-172-31-35-46:~$ kubectl get svc my-5gc-svc-lb
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
my-5gc-svc-lb   LoadBalancer   10.102.218.162   <pending>     80:31260/TCP   9s
tlxp@ip-172-31-35-46:~$ minikube service my-5gc-svc-lb
|-----------|---------------|-------------|---------------------------|
| NAMESPACE |     NAME      | TARGET PORT |            URL            |
|-----------|---------------|-------------|---------------------------|
| default   | my-5gc-svc-lb | 80          | http://192.168.49.2:31260 |
|-----------|---------------|-------------|---------------------------|
🎉 Opening service default/my-5gc-svc-lb in default browser...
👉 http://192.168.49.2:31260
```

Create a YAML File

```
tlxp@ip-172-31-35-46:~$ nano my-5gc-externalname.yaml
tlxp@ip-172-31-35-46:~$ kubectl apply -f my-5gc-externalname.yaml
service/my-5gc-externalname created
tlxp@ip-172-31-35-46:~$ kubectl get svc my-5gc-externalname
NAME                 TYPE           CLUSTER-IP   EXTERNAL-IP      PORT(S)   AGE
my-5gc-externalname  ExternalName   <none>       telcolearn.com   <none>    9s
```

```
  GNU nano 6.2                                  my-5gc-externalname.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-5gc-externalname
spec:
  type: ExternalName
  externalName: telcolearn.com
```

Create a pod using YAML

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f my-5gc-pod.yaml
pod/my-5gc-pod unchanged
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-5gc-pod    1/1     Running   0          24m
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f my-5gc-pod.yaml
pod/my-5gc-pod unchanged
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-5gc-pod    1/1     Running   0          25m
```

Describe the Pods

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl describe pod my-5gc-pod
Name:             my-5gc-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Wed, 31 Dec 2025 06:21:18 +0000
Labels:           app=my-5gc
Annotations:      <none>
Status:           Running
IP:               10.244.0.3
IPs:
  IP:  10.244.0.3
Containers:
  my-5gc-container:
    Container ID:   docker://0b6bf63542d5f5802ace17a01c0722c09ccc913c2b3ec06bd685b11eaacac0df
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:ca871a86d45a3ec6864dc45f014b11fe626145569ef0e74deaffc95a3b15b430
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Wed, 31 Dec 2025 06:21:28 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sxhmj (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-sxhmj:
    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
```

Create Multiple Pods using ReplicaSet and Verify it

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f my-5gc-replicaset.yaml
replicaset.apps/my-5gc-rs created
tlxp@ip-172-31-35-46:~/yaml_files$
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get replicasets
NAME        DESIRED   CURRENT   READY   AGE
my-5gc-rs   3         3         3       11s
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods -l app=my-5gc
NAME              READY   STATUS    RESTARTS   AGE
my-5gc-pod        1/1     Running   0          29m
my-5gc-rs-h9r5l   1/1     Running   0          21s
my-5gc-rs-r8c4n   1/1     Running   0          21s
```

Test self-healing

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl delete pod my-5gc-pod
pod "my-5gc-pod" deleted from default namespace
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods -l app=my-5gc
NAME              READY   STATUS    RESTARTS   AGE
my-5gc-rs-h9r5l   1/1     Running   0          2m22s
my-5gc-rs-r8c4n   1/1     Running   0          2m22s
my-5gc-rs-wpsqb   1/1     Running   0          21s
```

Delete old ReplicaSet (CleanUp)

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl delete -f my-5gc-replicaset.yaml
replicaset.apps "my-5gc-rs" deleted from default namespace
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods
kubectl get replicasets
No resources found in default namespace.
No resources found in default namespace.
```

Applying Deployment and Verifying it

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f my-5gc-deployment.yaml
deployment.apps/my-5gc-deployment created
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
my-5gc-deployment   3/3     3            3           22s
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get ReplicaSet
NAME                          DESIRED   CURRENT   READY   AGE
my-5gc-deployment-6d6cf88fd   3         3         3       34s
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods -l app=my-5gc
NAME                                READY   STATUS    RESTARTS   AGE
my-5gc-deployment-6d6cf88fd-cktt9   1/1     Running   0          52s
my-5gc-deployment-6d6cf88fd-v5hlb   1/1     Running   0          52s
my-5gc-deployment-6d6cf88fd-xzhgn   1/1     Running   0          52s
```

Scale Deployment to 6 Replicas

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl scale deployment my-5gc-deployment --replicas=6
deployment.apps/my-5gc-deployment scaled
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
my-5gc-deployment-6d6cf88fd-b4j4b   1/1     Running   0          7s
my-5gc-deployment-6d6cf88fd-cktt9   1/1     Running   0          3m2s
my-5gc-deployment-6d6cf88fd-l457m   1/1     Running   0          7s
my-5gc-deployment-6d6cf88fd-v5hlb   1/1     Running   0          3m2s
my-5gc-deployment-6d6cf88fd-xkpcm   1/1     Running   0          7s
my-5gc-deployment-6d6cf88fd-xzhgn   1/1     Running   0          3m2s
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get deployment my-5gc-deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
my-5gc-deployment   6/6     6            6           3m11s
```

Delete Deployment

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl delete deployment my-5gc-deployment
deployment.apps "my-5gc-deployment" deleted from default namespace
```

Create an Nginx Deployment

Create the deployment YAML, apply deployment and Verify

```
tlxp@ip-172-31-35-46:~/yaml_files$ nano nginx-deployment.yaml
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deploy created
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get pods -l app=nginx
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-77bf8679f9-tt77g   1/1     Running   0          19s
nginx-deploy-77bf8679f9-xnrwb   1/1     Running   0          19s
```

Expose Deployment via ClusterIP (Internal Access)

Create a ClusterIP Service

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl expose deployment nginx-deploy \
  --port=80 \
  --target-port=80 \
  --type=ClusterIP \
  --name=nginx-service
service/nginx-service exposed
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get svc nginx-service
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.100.53.63   <none>        80/TCP    12s
```

Test Network Access from INSIDE the Cluster

Run a temporary test pod

```
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl run tester \
  --image=busybox \
  --restart=Never \
  -it --rm \
  --command -- sh

All commands and output from this session will be recorded in container logs, including credentials and sensi
 through the command prompt.
If you don't see a command prompt, try pressing enter.
/ #
```

Test Service Connectivity

```
/ #
/ # 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>
```

ConfigMap

Inspect ConfigMap YAML

```
tlxp@ip-172-31-35-46:~/yaml_files$ cat my-5gc-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-5gc-config
data:
  APP_ENV: production
  LOG_LEVEL: debugtlxp@ip-172-31-35kubectl apply -f my-5gc-configmap.yaml5gc-configmap.yaml
configmap/my-5gc-config created
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get configmap my-5gc-config -o yaml
apiVersion: v1
data:
  APP_ENV: production
  LOG_LEVEL: debug
kind: ConfigMap
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"APP_ENV":"production","LOG_LEVEL":"debug"},"kind":"ConfigMap","meta
"my-5gc-config","namespace":"default"}}
  creationTimestamp: "2025-12-31T07:14:46Z"
  name: my-5gc-config
  namespace: default
  resourceVersion: "3810"
  uid: a59f738b-277f-4ed2-a3c7-2f9ce17a715f
```

Secrets

Check Secret Yaml file, apply secret, verify Secret.

```
tlxp@ip-172-31-35-46:~/yaml_files$ cat my-5gc-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: my-5gc-secret
type: Opaque
data:
  DB_PASSWORD: cGFzc3dvcmQxMjM=  # base64 encoded 'password123'tlxp@ip-172-31-35-46:~/yaml_files$ kubectl apply -f my-5gc-secret.yam
l                                kubectl apply -f my-5gc-secret.yaml
secret/my-5gc-secret created
tlxp@ip-172-31-35-46:~/yaml_files$ kubectl get secret my-5gc-secret -o yaml
apiVersion: v1
data:
  DB_PASSWORD: cGFzc3dvcmQxMjM=
kind: Secret
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"DB_PASSWORD":"cGFzc3dvcmQxMjM="},"kind":"Secret","metadata":{"annotations":{},"name":"my-5gc-secre
t","namespace":"default"},"type":"Opaque"}
  creationTimestamp: "2025-12-31T07:22:41Z"
  name: my-5gc-secret
  namespace: default
  resourceVersion: "4208"
  uid: fdb6411a-07d8-4723-8e99-f80b898db551
type: Opaque
tlxp@ip-172-31-35-46:~/yaml_files$ env
SHELL=/bin/bash
PWD=/home/tlxp/yaml_files
LOGNAME=tlxp
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/tlxp
LANG=C.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;
41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;
31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*
.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.
jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.c
ab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35
:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;
35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=
01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=
01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01
;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00
```

In this lab, an Nginx application was deployed using a Kubernetes Deployment with multiple replicas. The Deployment was exposed internally using a ClusterIP Service. Network connectivity was validated by launching a temporary BusyBox pod and accessing the service using DNS. ConfigMaps were applied to store application configuration data, and Secrets were used to securely store sensitive information. Optional steps demonstrated how Pods can consume ConfigMaps and Secrets as environment variables.