

CKA Hands-on Labs

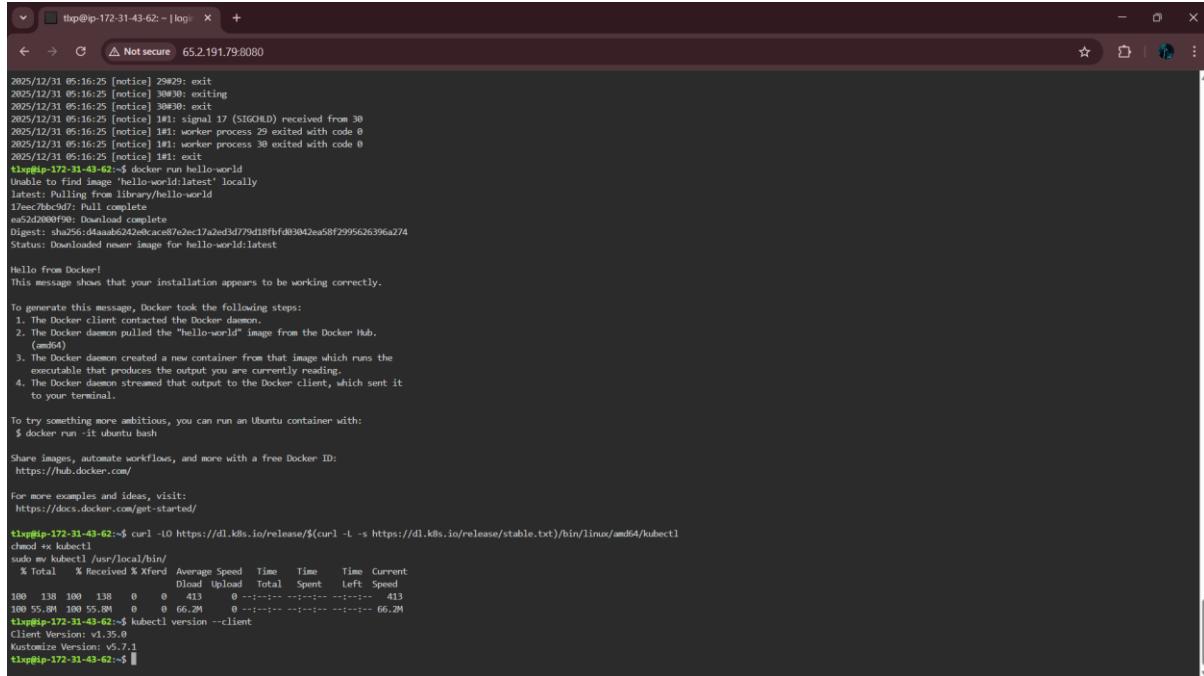
Step 1: Install Docker on Ubuntu (AWS VM) :

```
tbp@ip-172-31-43-62: ~ | log | +  
← → ⌂ △ Not secure 65.2.191.79:8080  
  
2025/12/31 05:12:51 [notice] 1#1: using the "poll" event method  
2025/12/31 05:12:51 [notice] 1#1: nginx/1.29.4  
2025/12/31 05:12:51 [notice] 1#1: built by gcc: gcc 4.2.2 (Debian 4.2.2-19)  
2025/12/31 05:12:51 [notice] 1#1: OS: Linux 6.8.0-1031-aws  
2025/12/31 05:12:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288  
2025/12/31 05:12:51 [notice] 1#1: start worker processes  
2025/12/31 05:12:51 [notice] 1#1: start worker process 29  
2025/12/31 05:12:51 [notice] 1#1: start worker process 30  
2025/12/31 05:12:51 [notice] 1#1: start worker process 31  
2025/12/31 05:12:51 [notice] 1#1: signal 17 (SIGCHLD) received  
*2025/12/31 05:16:12 [notice] 1#1: signal 29 (SIGIO/NONBLOCK) received  
*2025/12/31 05:16:25 [notice] 1#1: signal 29 (SIGIO/NONBLOCK) received, exiting  
2025/12/31 05:16:25 [notice] 29#2: exiting  
2025/12/31 05:16:25 [notice] 29#2: exit  
2025/12/31 05:16:25 [notice] 30#3: exiting  
2025/12/31 05:16:25 [notice] 30#3: exit  
2025/12/31 05:16:25 [notice] 1#1: signal 17 (SIGCHLD) received from 30  
2025/12/31 05:16:25 [notice] 1#1: worker process 29 exited with code 0  
2025/12/31 05:16:25 [notice] 1#1: worker process 29 exited with code 0  
2025/12/31 05:16:25 [notice] 1#1: worker process 30 exited with code 0  
2025/12/31 05:16:25 [notice] 1#1: exit  
t1y@ip-172-31-43-62: ~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
1/1:ec70bc9d7: Pull complete  
e552d2000f98: Download complete  
Digest: sha256:d4aaad6242e0cace87e2ec17a2ed3d779d18fbfd03042ea58f2995626396a274  
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.  
(andid)  
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:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas:  
https://docs.docker.com/get-started/
```

Step 2: Add Your User to the Docker Group :

```
tbp@ip-172-31-43-62: ~ | log | +  
↳ → C ⚠ Not secure 65.2.191.79:8080  
  
Selecting previously unselected package docker-compose-plugin.  
Preparing to unpack .../6-docker-compose-plugin_5.0.0-1ubuntu.22.04~jammy_amd64.deb ...  
Unpacking docker-compose-plugin (5.0.0-1ubuntu.22.04~jammy) ...  
Selecting previously unselected package libsslrp0:amd64.  
Preparing to unpack .../7-libsslrp0_4.6.1-1build0_amd64.deb ...  
Unpacking libsslrp0:amd64 (4.6.1-1build0) ...  
Selecting previously unselected package slirp4nntns.  
Preparing to unpack .../8-slirp4nntns_1.0.1-2_amd64.deb ...  
Unpacking slirp4nntns (1.0.1-2) ...  
Setting up docker-buildx-plugin (0.30.1-1ubuntu.22.04~jammy) ...  
Setting up docker-containervd (2.2.1-1ubuntu.22.04~jammy) ...  
Setting up containervd (2.2.1-1ubuntu.22.04~jammy) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.  
Setting up docker-compose-plugin (5.0.0-1ubuntu.22.04~jammy) ...  
Setting up docker-ce-cl1 (5:29.1.3-1ubuntu.22.04~jammy) ...  
Setting up libsslrp0:amd64 (4.6.1-1build1) ...  
Setting up pigna (2.6.1-1) ...  
Setting up docker-ce-containerv2-extras (5:29.1.3-3ubuntu.22.04~jammy) ...  
Setting up docker-ce-cl1 (5:29.1.3-4ubuntu.22.04~jammy) ...  
Setting up docker-ce (5:29.1.3-4ubuntu.22.04~jammy) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.  
Created symlink /etc/systemd/system/socket.target.wants/docker.socket → /lib/systemd/system/docker.socket.  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.10) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
tbp@ip-172-31-43-62:~$ docker --version  
Docker version 29.1.3, build f52814d  
tbp@ip-172-31-43-62:~$ sudo usermod -aG docker $USER  
newgrp docker  
  
tbp@ip-172-31-43-62:~$ docker run nginx  
Unable to find image 'nginx:latest' locally
```

Step 3: Test Docker with Some Basic Commands :



```
2025/12/31 05:16:25 [notice] 29#29: exit
2025/12/31 05:16:25 [notice] 30#30: exiting
2025/12/31 05:16:25 [notice] 30#30: exiting
2025/12/31 05:16:25 [notice] 181: signal 17 (SIGKILL) received from 30
2025/12/31 05:16:25 [notice] 181: worker process 29 exited with code 0
2025/12/31 05:16:25 [notice] 181: worker process 30 exited with code 0
2025/12/31 05:16:25 [notice] 181: exit
tixp@ip-172-31-43-62:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17ec57b8c9d7: Pull complete
Digest: sha256:d4aaab6242edebace8e2ec17a2ed3d779d18fbfd3042ea58f2995626396a274
Status: Downloaded newer image for hello-world:latest

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

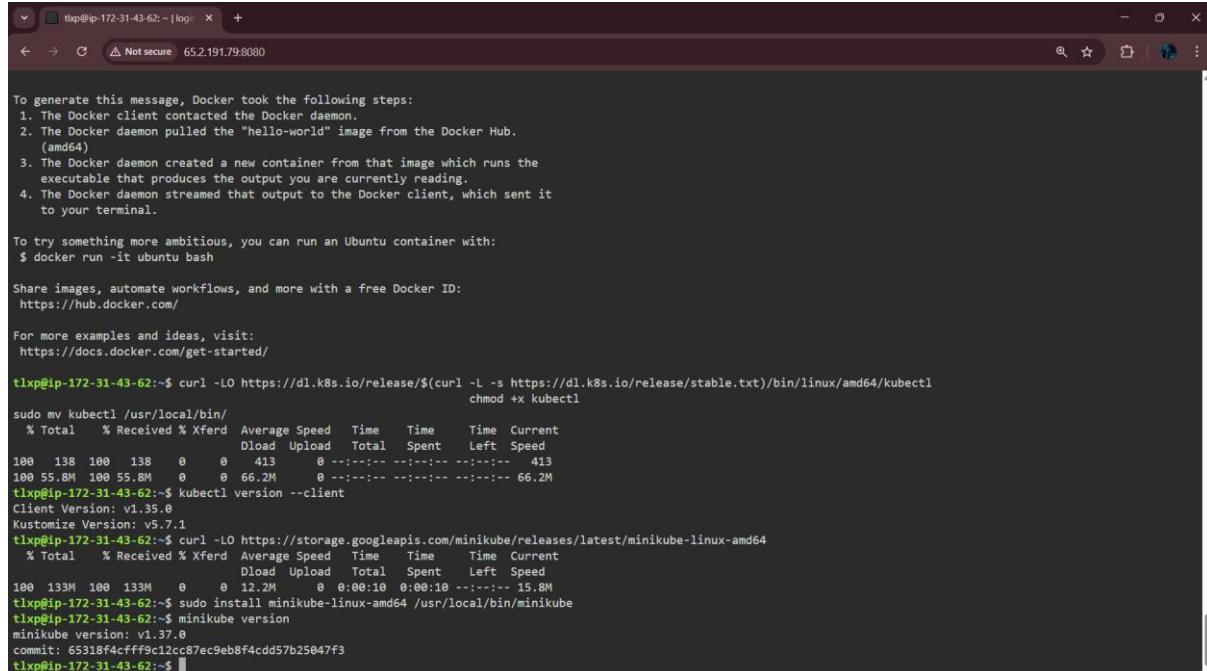
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the 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:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
tixp@ip-172-31-43-62:~$ curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
chmod +x kubectl
sudo mv kubectl /usr/local/bin/
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent   Left  Speed
100  13B  100  13B    0     0  0:00:00 --:--:-- 0:00:00 413
100 55.8M  100 55.8M   0     0  66.2M  0:00:00 --:--:-- 0:00:00 66.2M
tixp@ip-172-31-43-62:~$ kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1
tixp@ip-172-31-43-62:~$
```

MiniKube Installation :



```
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:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
tixp@ip-172-31-43-62:~$ curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
chmod +x kubectl
sudo mv kubectl /usr/local/bin/
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent   Left  Speed
100  13B  100  13B    0     0  0:00:00 --:--:-- 0:00:00 413
100 55.8M  100 55.8M   0     0  66.2M  0:00:00 --:--:-- 0:00:00 66.2M
tixp@ip-172-31-43-62:~$ kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1
tixp@ip-172-31-43-62:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent   Left  Speed
100 133M  100 133M    0     0 12.2M  0:00:10 0:00:10 --:--:-- 15.8M
tixp@ip-172-31-43-62:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
tixp@ip-172-31-43-62:~$ minikube version
minikube version: v1.37.0
commit: 65318f4cff9c12cc87ec9eb8f4cdd57b25047f3
tixp@ip-172-31-43-62:~$
```

```

Kustomize Version: v5.7.1
tlxpc@ip-172-31-43-62:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          0     0      0   12.2M    0:00:10  0:00:10 ---:--:-- 15.8M
tlxpc@ip-172-31-43-62:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
tlxpc@ip-172-31-43-62:~$ minikube version
minikube version: v1.37.0
commit: 65318f4ccff9c12cc87ec9eb8f4cd57b25047f3
tlxpc@ip-172-31-43-62:~$ 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 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% 23.68 M
  > preloaded-images-k8s-v18-v1...: 337.07 MiB / 337.07 MiB 100.00% 15.13 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: default-storageclass, storage-provisioner
📅 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
tlxpc@ip-172-31-43-62:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
minikube  Ready    control-plane   14m   v1.34.0
tlxpc@ip-172-31-43-62:~$ kubectl get pods
No resources found in default namespace.
tlxpc@ip-172-31-43-62:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
minikube  Ready    control-plane   15m   v1.34.0
tlxpc@ip-172-31-43-62:~$ 

```

Create Pods :

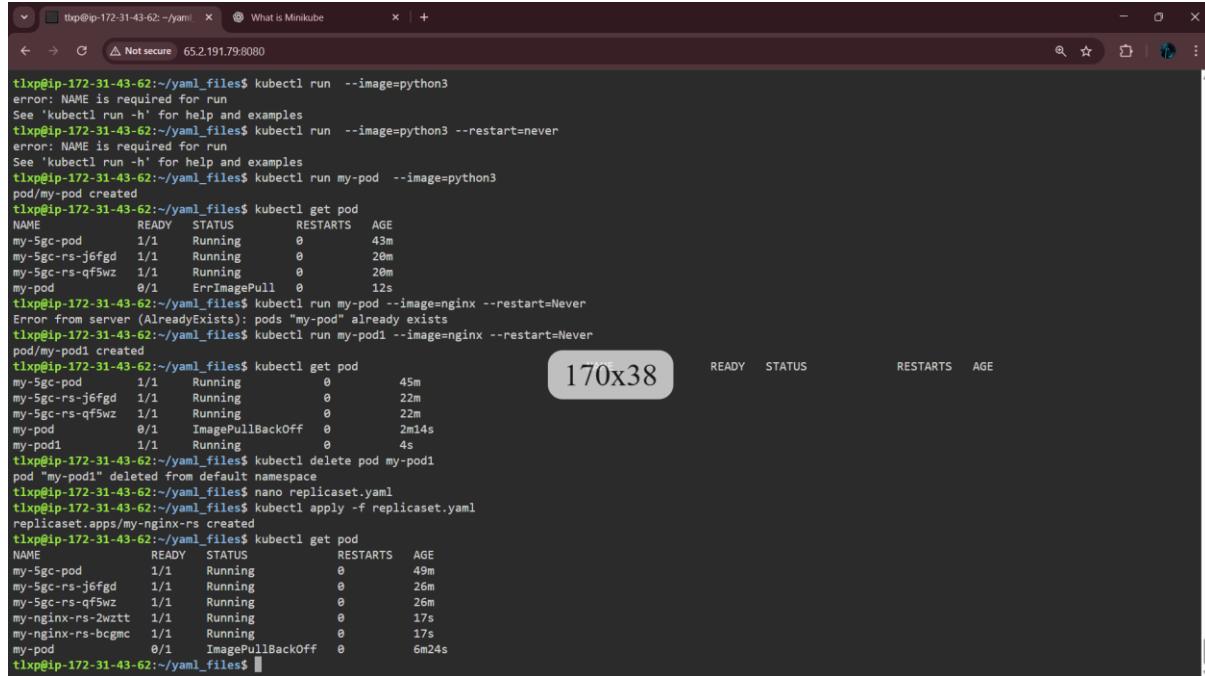
```

tlxpc@ip-172-31-43-62:~$ kubectl get pods -A
NAMESPACE        NAME        READY   STATUS    RESTARTS   AGE
kube-system      coredns-66bc5c9577-2qwgf   1/1    Running   0          16m
kube-system      etcd-minikube   1/1    Running   0          16m
kube-system      kube-apiserver-minikube   1/1    Running   0          16m
kube-system      kube-controller-manager-minikube   1/1    Running   0          16m
kube-system      kube-proxy-bzhpk   1/1    Running   0          16m
kube-system      kube-scheduler-minikube   1/1    Running   0          16m
kube-system      storage-provisioner   1/1    Running   1 (16m ago) 16m
tlxpc@ip-172-31-43-62:~$ kubectl get pods -o wide -A
NAMESPACE        NAME        READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED NODE   READINESS GATES
kube-system      coredns-66bc5c9577-2qwgf   1/1    Running   0          16m   10.244.0.2   minikube   <none>   <none>
kube-system      etcd-minikube   1/1    Running   0          16m   192.168.49.2   minikube   <none>   <none>
kube-system      kube-apiserver-minikube   1/1    Running   0          16m   192.168.49.2   minikube   <none>   <none>
kube-system      kube-controller-manager-minikube   1/1    Running   0          16m   192.168.49.2   minikube   <none>   <none>
kube-system      kube-proxy-bzhpk   1/1    Running   0          16m   192.168.49.2   minikube   <none>   <none>
kube-system      kube-scheduler-minikube   1/1    Running   0          16m   192.168.49.2   minikube   <none>   <none>
kube-system      storage-provisioner   1/1    Running   1 (16m ago) 16m   192.168.49.2   minikube   <none>   <none>
tlxpc@ip-172-31-43-62:~$ alias k=kubectl
tlxpc@ip-172-31-43-62:~$ kubectl apply -f ~/yaml_files/my-5gc-pod.yaml
pod/my-5gc-pod created
tlxpc@ip-172-31-43-62:~$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
my-5gc-pod  1/1    Running   0          11s
tlxpc@ip-172-31-43-62:~$ 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 05:43:03 +0000
Labels:       app=my-5gc
Annotations: <none>
Status:      Running
IP:          10.244.0.3
IPs:          IP: 10.244.0.3

```

K8s Workloads :

ReplicaSet output :

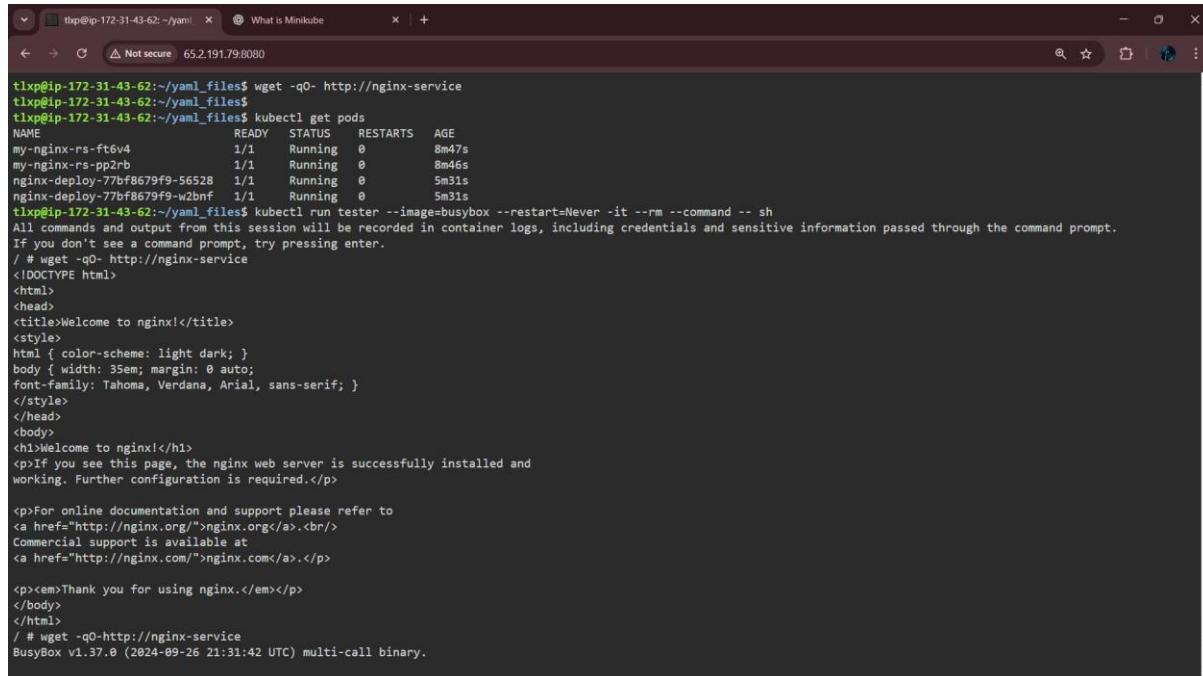


```
tixp@ip-172-31-43-62:~/yaml_files$ kubectl run --image=python3
error: NAME is required for run
See 'kubectl run -h' for help and examples
tixp@ip-172-31-43-62:~/yaml_files$ kubectl run --image=python3 --restart=never
error: NAME is required for run
See 'kubectl run -h' for help and examples
tixp@ip-172-31-43-62:~/yaml_files$ kubectl run my-pod --image=python3
pod/my-pod created
tixp@ip-172-31-43-62:~/yaml_files$ kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
my-5gc-pod     1/1     Running   0          43m
my-5gc-rs-j6fgd 1/1     Running   0          28m
my-5gc-rs-qf5wz 1/1     Running   0          20m
my-pod          0/1     ErrImagePull 0          12s
tixp@ip-172-31-43-62:~/yaml_files$ kubectl run my-pod --image=nginx --restart=Never
Error from server (AlreadyExists): pods "my-pod" already exists
tixp@ip-172-31-43-62:~/yaml_files$ kubectl run my-pod1 --image=nginx --restart=Never
pod/my-pod1 created
tixp@ip-172-31-43-62:~/yaml_files$ kubectl get pod
my-5gc-pod     1/1     Running   0          45m
my-5gc-rs-j6fgd 1/1     Running   0          22m
my-5gc-rs-qf5wz 1/1     Running   0          22m
my-pod          0/1     ImagePullBackOff 0          2m14s
my-pod1         1/1     Running   0          4s
tixp@ip-172-31-43-62:~/yaml_files$ kubectl delete pod my-pod1
pod "my-pod1" deleted from default namespace
tixp@ip-172-31-43-62:~/yaml_files$ nano replicaset.yaml
tixp@ip-172-31-43-62:~/yaml_files$ kubectl apply -f replicaset.yaml
replicaset.apps/my-nginx-rs created
tixp@ip-172-31-43-62:~/yaml_files$ kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
my-5gc-pod     1/1     Running   0          49m
my-5gc-rs-j6fgd 1/1     Running   0          26m
my-5gc-rs-qf5wz 1/1     Running   0          26m
my-nginx-rs-2wztt 1/1     Running   0          17s
my-nginx-rs-bcgm 1/1     Running   0          17s
my-pod          0/1     ImagePullBackOff 0          6m24s
tixp@ip-172-31-43-62:~/yaml_files$
```

170x38

Kubernetes Networking Exercise :

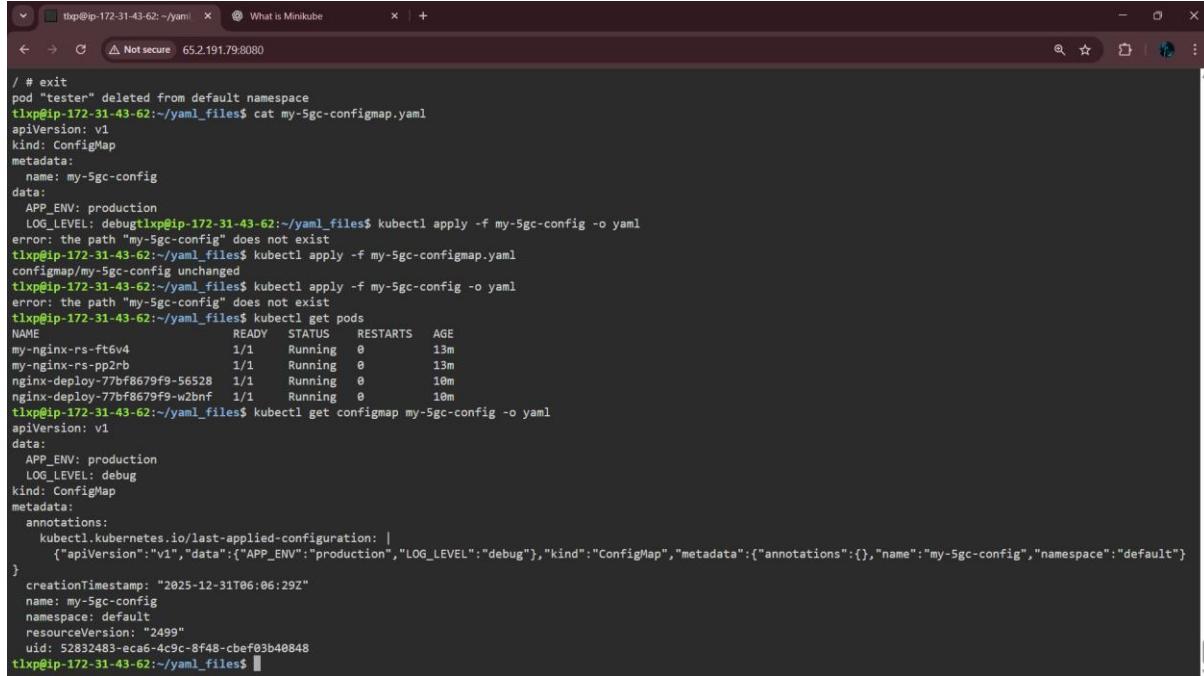
Nginx Deployment output:



```
tixp@ip-172-31-43-62:~/yaml_files$ wget -qO- http://nginx-service
tixp@ip-172-31-43-62:~/yaml_files$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
my-nginx-rs-ftcv4 1/1     Running   0          8m47s
my-nginx-rs-pp2rb 1/1     Running   0          8m46s
nginx-deploy-77bf8679f9-w5628 1/1     Running   0          5m31s
nginx-deploy-77bf8679f9-w2bnf 1/1     Running   0          5m31s
tixp@ip-172-31-43-62:~/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 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>
/ # wget -qO- http://nginx-service
BusyBox v1.37.0 (2024-09-26 21:31:42 UTC) multi-call binary.
```

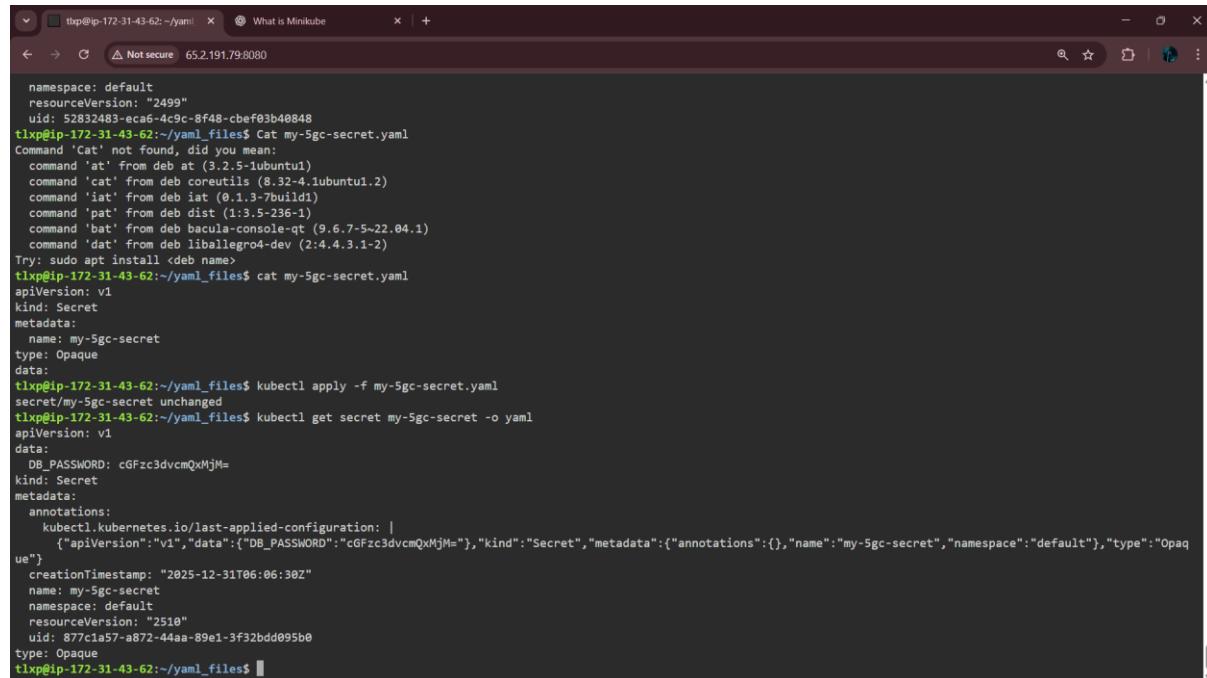
ConfigMap and Secrets :

Configmap output :



```
/ # exit
pod "tester" deleted from default namespace
t1xp@ip-172-31-43-62:~/yaml_files$ cat my-5gc-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-5gc-config
data:
  APP_ENV: production
  LOG_LEVEL: debug
  kind: ConfigMap
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion": "v1", "data": {"APP_ENV": "production", "LOG_LEVEL": "debug"}, "kind": "ConfigMap", "metadata": {"annotations": {}, "name": "my-5gc-config", "namespace": "default"}}
  creationTimestamp: "2025-12-31T06:06:29Z"
  name: my-5gc-config
  namespace: default
  resourceVersion: "2499"
  uid: 52832483-eca6-4c9c-8f48-cbef03b40848
apiVersion: v1
data:
  APP_ENV: production
  LOG_LEVEL: debug
  kind: ConfigMap
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion": "v1", "data": {"APP_ENV": "production", "LOG_LEVEL": "debug"}, "kind": "ConfigMap", "metadata": {"annotations": {}, "name": "my-5gc-config", "namespace": "default"}}
  creationTimestamp: "2025-12-31T06:06:29Z"
  name: my-5gc-config
  namespace: default
  resourceVersion: "2499"
  uid: 52832483-eca6-4c9c-8f48-cbef03b40848
t1xp@ip-172-31-43-62:~/yaml_files$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-nginx-rs-ft6v4   1/1     Running   0          13m
my-nginx-rs-pp2rb   1/1     Running   0          13m
nginx-deploy-77bf8679f9-56528  1/1     Running   0          10m
nginx-deploy-77bf8679f9-wzbnf   1/1     Running   0          10m
t1xp@ip-172-31-43-62:~/yaml_files$ kubectl get configmap my-5gc-config -o yaml
apiVersion: v1
data:
  APP_ENV: production
  LOG_LEVEL: debug
  kind: ConfigMap
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion": "v1", "data": {"APP_ENV": "production", "LOG_LEVEL": "debug"}, "kind": "ConfigMap", "metadata": {"annotations": {}, "name": "my-5gc-config", "namespace": "default"}}
  creationTimestamp: "2025-12-31T06:06:29Z"
  name: my-5gc-config
  namespace: default
  resourceVersion: "2499"
  uid: 52832483-eca6-4c9c-8f48-cbef03b40848
t1xp@ip-172-31-43-62:~/yaml_files$
```

Secret.yaml output :



```
namespace: default
resourceVersion: "2499"
uid: 52832483-eca6-4c9c-8f48-cbef03b40848
t1xp@ip-172-31-43-62:~/yaml_files$ cat my-5gc-secret.yaml
Command 'Cat' not found, did you mean:
  command 'at' from deb at (3.2.5-lubuntu1)
  command 'cat' from deb coreutils (8.32-4.lubuntu1.2)
  command 'iat' from deb iart (0.1.3-7build1)
  command 'pat' from deb dist (1:3.5-236-1)
  command 'bat' from deb bacula-console-qt (9.6.7-5~22.04.1)
  command 'dat' from deb liballegro4-dev (2:4.4.3.1-2)
Try: sudo apt install <deb name>
t1xp@ip-172-31-43-62:~/yaml_files$ cat my-5gc-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: my-5gc-secret
type: Opaque
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-secret", "namespace": "default"}, "type": "Opaque"}
  creationTimestamp: "2025-12-31T06:06:30Z"
  name: my-5gc-secret
  namespace: default
  resourceVersion: "2510"
  uid: 877c1a57-a872-44a4-89e1-3f32bdd095b0
type: Opaque
t1xp@ip-172-31-43-62:~/yaml_files$
```

