Minikube Setup on AWS EC2 (Ubuntu) and Namespace Exploration

1. Launch EC2 Instance

Instance Type: t2.medium or higher (2vCPU, 4GB RAM)

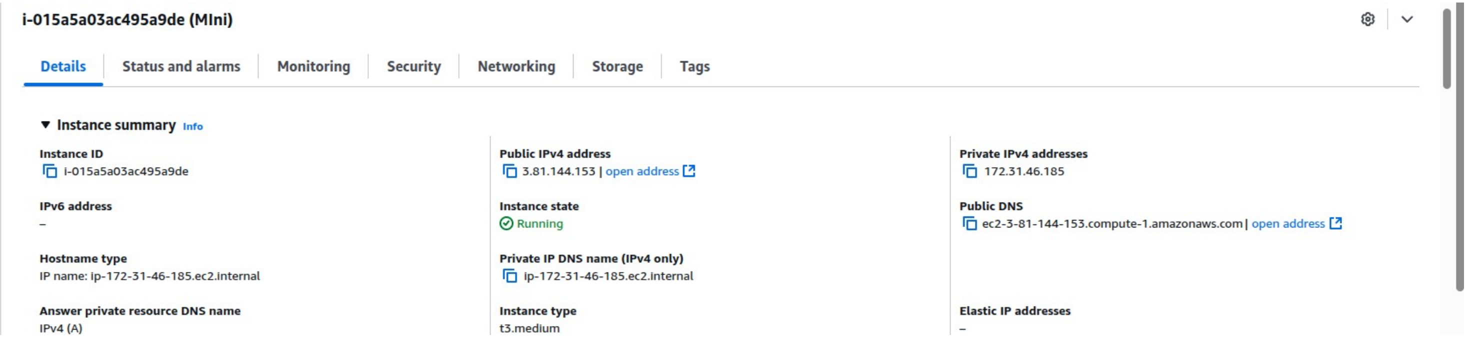
AMI: Ubuntu 22.04 LTS

Storage: Minimum 20GB

Security Group: Allow ports 22 (SSH), 30000-32767 (K8s NodePort range)

Connect via SSH:

```
ssh -i <your-key.pem> ubuntu@<ec2-public-ip>
```



i-015a5a03ac495a9de (MIni)

▼ Instance summary Info

Instance ID: i-015a5a03ac495a9de

IPv6 address: –

Hostname type: IP name: ip-172-31-46-185.ec2.internal

Answer private resource DNS name: IPv4 (A)

Public IPv4 address: 3.81.144.153 | open address

Instance state: Running

Private IP DNS name (IPv4 only): ip-172-31-46-185.ec2.internal

Instance type: t3.medium

Private IPv4 addresses: 172.31.46.185

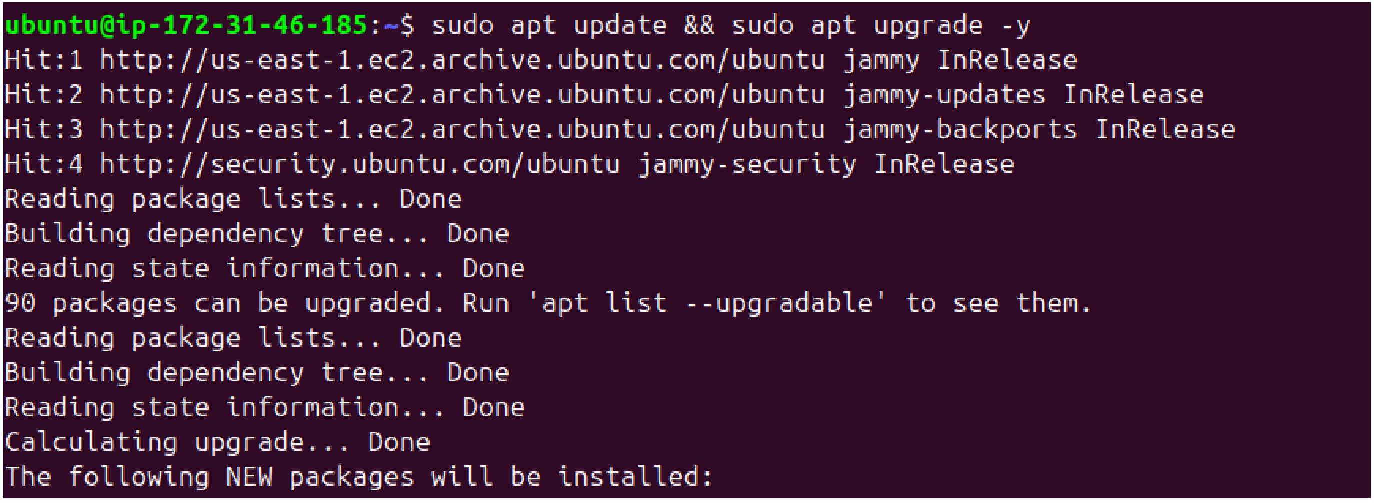
Public DNS: ec2-3-81-144-153.compute-1.amazonaws.com | open address

Elastic IP addresses: –

2. Install Dependencies

2.1 Update OS packages

```
sudo apt update && sudo apt upgrade -y
```

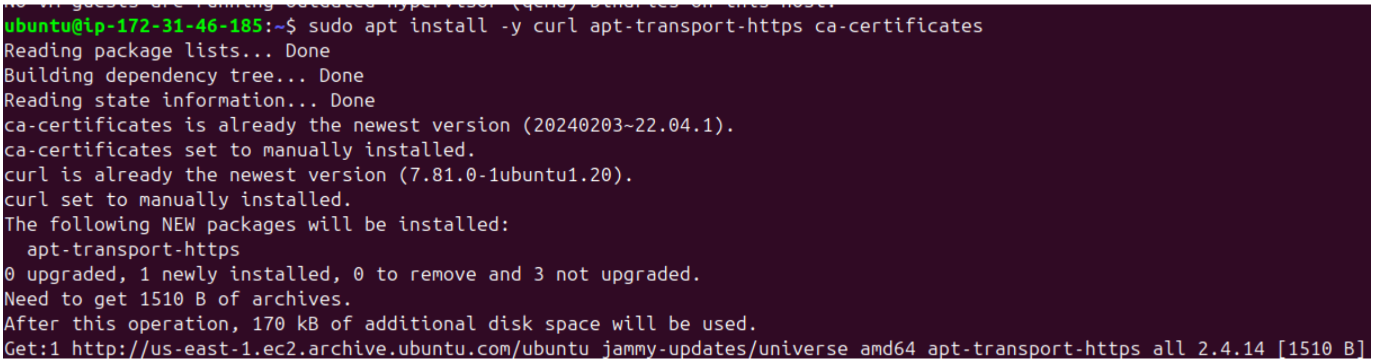


```
ubuntu@ip-172-31-46-185:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
90 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
```

2.2 Install essential packages

```
sudo apt install -y curl apt-transport-https ca-certificates
```

Minikube Setup on AWS EC2 (Ubuntu) and Namespace Exploration



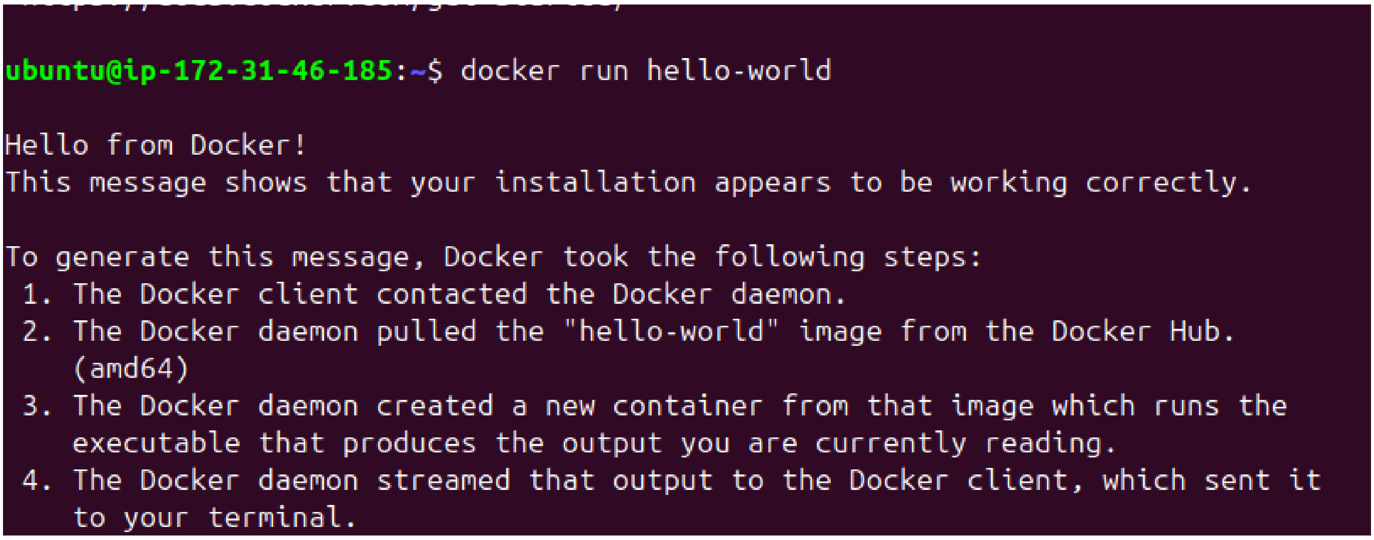
```
ubuntu@ip-172-31-46-185:~$ sudo apt install -y curl apt-transport-https ca-certificates
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.20).
curl set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 1510 B of archives.
After this operation, 170 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.14 [1510 B]
```

2.3 Install Docker

```
sudo apt install -y docker.io

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker $USER

newgrp docker
```

Verify Docker:

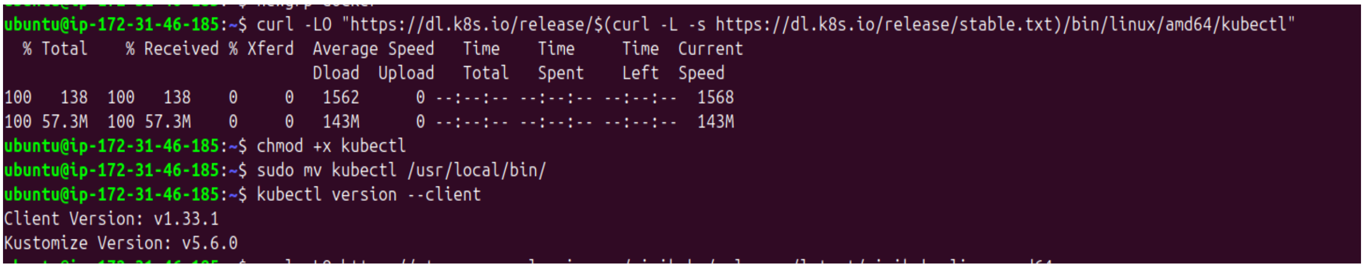
```
docker run hello-world
```



```
ubuntu@ip-172-31-46-185:~$ docker run hello-world

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.
```

2.4 Install kubectl

```
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/

kubectl version –client
```

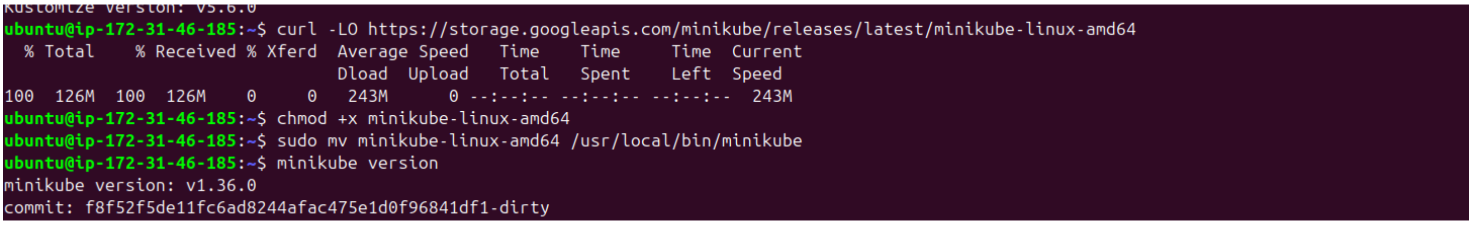


```
ubuntu@ip-172-31-46-185:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0   1562      0 --:--:-- --:--:-- --:--:--  1568
100 57.3M  100 57.3M    0     0   143M      0 --:--:-- --:--:-- --:--:--  143M
ubuntu@ip-172-31-46-185:~$ chmod +x kubectl
ubuntu@ip-172-31-46-185:~$ sudo mv kubectl /usr/local/bin/
ubuntu@ip-172-31-46-185:~$ kubectl version --client
Client Version: v1.33.1
Kustomize Version: v5.6.0
```

Minikube Setup on AWS EC2 (Ubuntu) and Namespace Exploration

2.5 Install Minikube

```
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
chmod +x minikube-linux-amd64
sudo mv minikube-linux-amd64 /usr/local/bin/minikube
minikube version
```



```
kustomize version: v5.6.0
ubuntu@ip-172-31-46-185:~$ 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  126M  100  126M    0     0   243M      0 --:--:-- --:--:-- --:--:--  243M
ubuntu@ip-172-31-46-185:~$ chmod +x minikube-linux-amd64
ubuntu@ip-172-31-46-185:~$ sudo mv minikube-linux-amd64 /usr/local/bin/minikube
ubuntu@ip-172-31-46-185:~$ minikube version
minikube version: v1.36.0
commit: f8f52f5de11fc6ad8244afac475e1d0f96841df1-dirty
```

3. Start Minikube

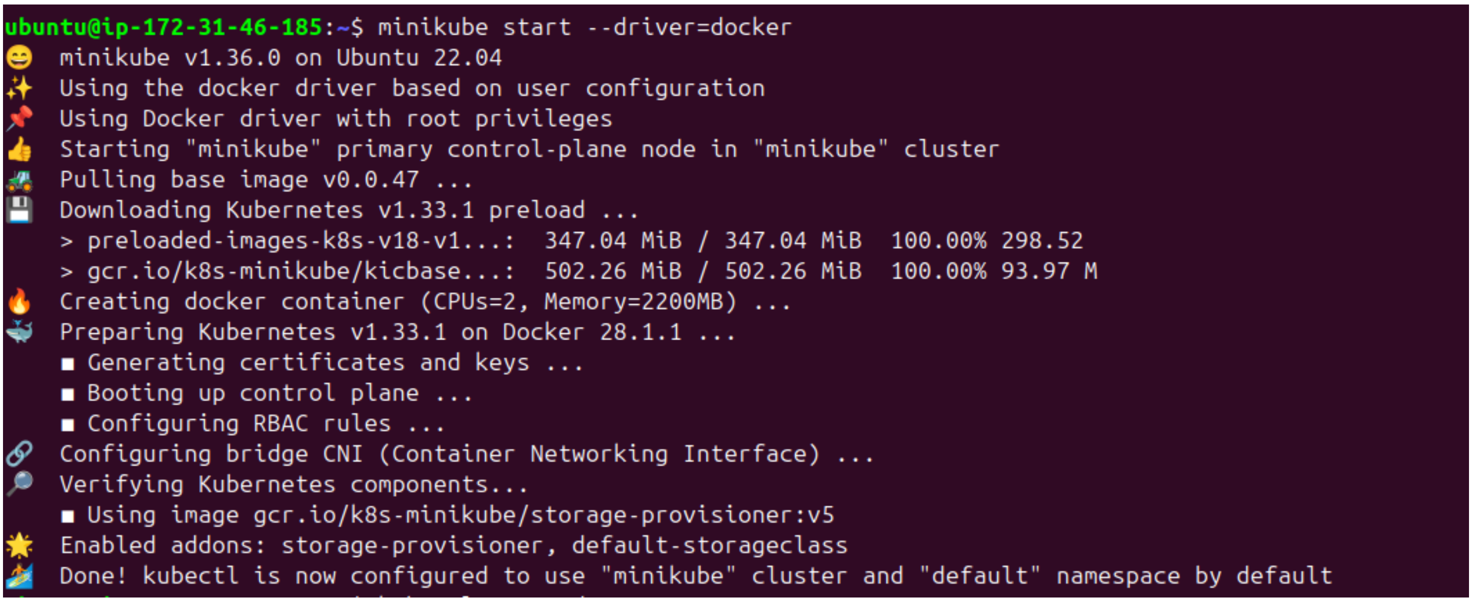
Option 1: Use Docker driver (Recommended)

```
minikube start --driver=docker
```

Option 2: Use None driver (runs Minikube directly on host)

If you want to try this, you may need to fix permissions (see Troubleshooting section).

```
sudo minikube start --driver=none
```



```
ubuntu@ip-172-31-46-185:~$ minikube start --driver=docker
😄  minikube v1.36.0 on Ubuntu 22.04
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.47 ...
💾  Downloading Kubernetes v1.33.1 preload ...
    > preloaded-images-k8s-v18-v1...:  347.04 MiB / 347.04 MiB  100.00% 298.52
    > gcr.io/k8s-minikube/kicbase...:  502.26 MiB / 502.26 MiB  100.00% 93.97 M
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  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
```

4. Explore Kubernetes Namespaces

4.1 Create a Namespace

```
kubectl create namespace dev-team
```

4.2 List Namespaces

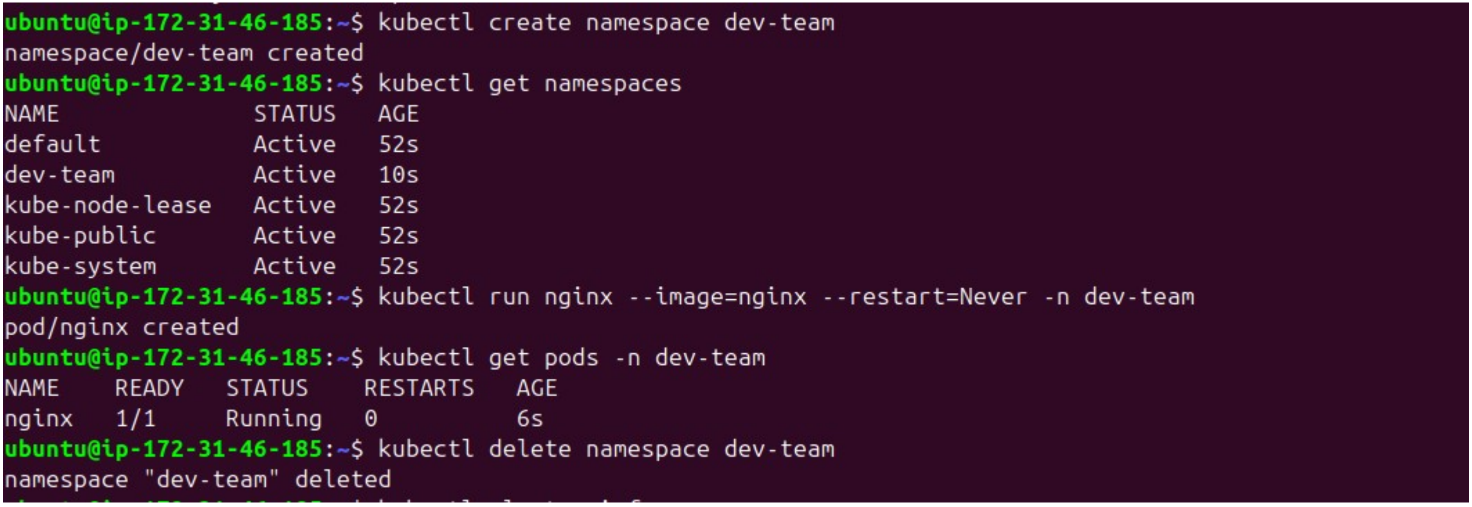
```
kubectl get namespaces
```

4.3 Run a Pod in the Namespace

```
kubectl run nginx --image=nginx --restart=Never -n dev-team

kubectl get pods -n dev-team
```

4.4 Delete Namespace

```
kubectl delete namespace dev-team
```



```
ubuntu@ip-172-31-46-185:~$ kubectl create namespace dev-team
namespace/dev-team created
ubuntu@ip-172-31-46-185:~$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   52s
dev-team          Active   10s
kube-node-lease   Active   52s
kube-public       Active   52s
kube-system       Active   52s
ubuntu@ip-172-31-46-185:~$ kubectl run nginx --image=nginx --restart=Never -n dev-team
pod/nginx created
ubuntu@ip-172-31-46-185:~$ kubectl get pods -n dev-team
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          6s
ubuntu@ip-172-31-46-185:~$ kubectl delete namespace dev-team
namespace "dev-team" deleted
```

5. Troubleshooting Common Issues

5.1 Permission denied on /tmp/juju-* when starting Minikube with none driver

```
sudo sysctl fs.protected_regular=0

sudo minikube delete

sudo rm -rf /tmp/juju-*

sudo minikube start --driver=none
```

5.2 Docker permissions issues

Check if your user is in the docker group:

```
groups
```

If docker is missing:

```
sudo usermod -aG docker $USER

newgrp docker
```

6. Useful Commands

Get cluster info:

```
kubectl cluster-info
```

List nodes:

```
kubectl get nodes
```

Check Minikube status:

```
minikube status
```

Delete Minikube cluster:

```
minikube delete
```

7. References

- Minikube official docs: https://minikube.sigs.k8s.io/docs/
- Kubernetes namespaces: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
- Kubectl installation: https://kubernetes.io/docs/tasks/tools/install-kubectl/
- Docker installation on Ubuntu: https://docs.docker.com/engine/install/ubuntu/

Minikube Setup on AWS EC2 (Ubuntu) and Namespace Exploration