Name: Taneeshq Cholekar Date:14/05/2025

Assignment-3

Docker and Kubernetes: The Container Masterclass

Problem Statement:

Setup Single Node Kubernetes Cluster with Minikube on Ubuntu, Minikube runs as a single node kubernetes cluster inside a VM on the laptop for the users who want to stry kubernetes and test codes locally on the kubernetes environment.

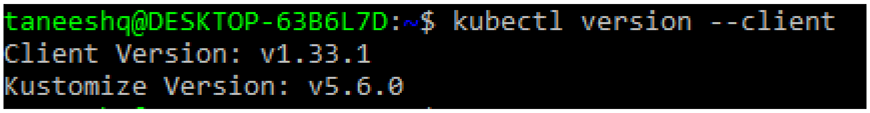
Methodology:

Step 1: Install kubectl and check version

```
curl –LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
chmod +x kubectl
```

```
sudo mv kubectl /usr/local/bin/
```

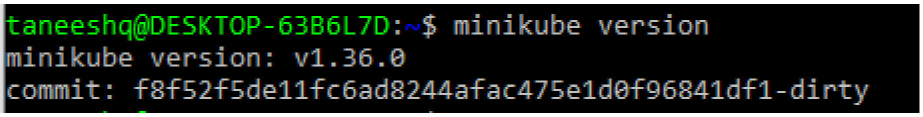


```
taneeshq@DESKTOP-63B6L7D:~$ kubectl version --client
Client Version: v1.33.1
Kustomize Version: v5.6.0
```

Step 2: Install minikube and check version

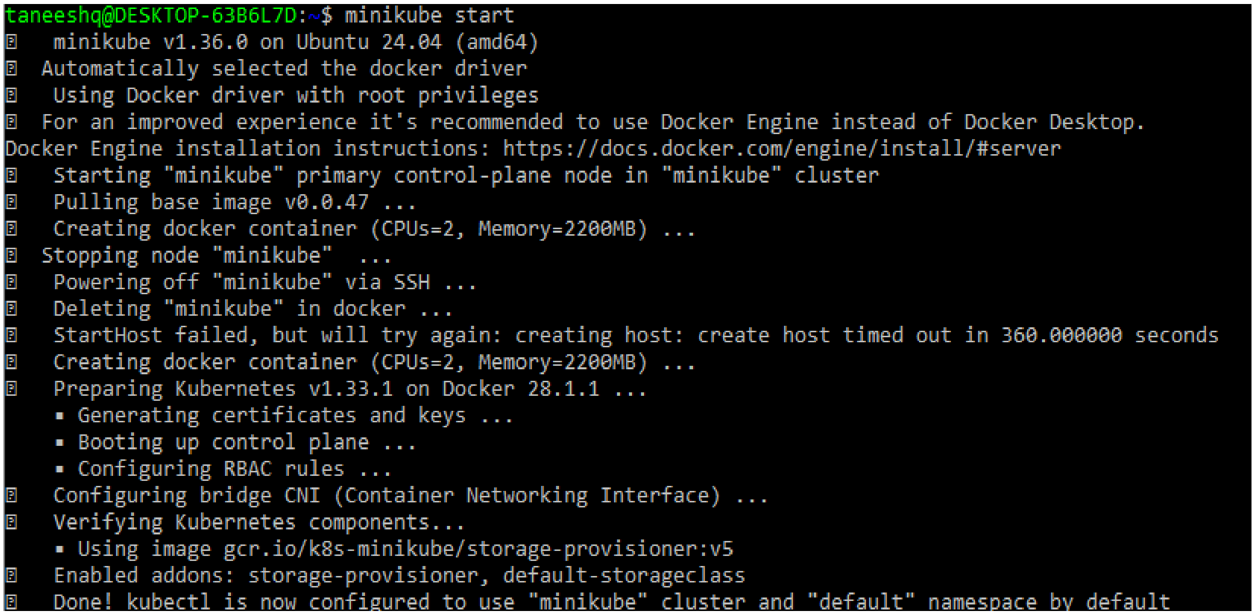
```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```



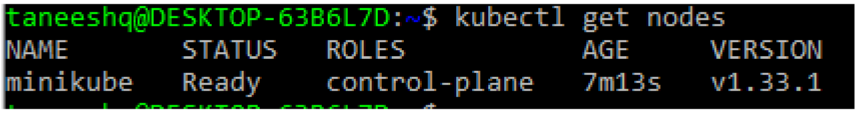
```
taneeshq@DESKTOP-63B6L7D:~$ minikube version
minikube version: v1.36.0
commit: f8f52f5de11fc6ad8244afac475e1d0f96841df1-dirty
```

Step 3: Start the minikube node



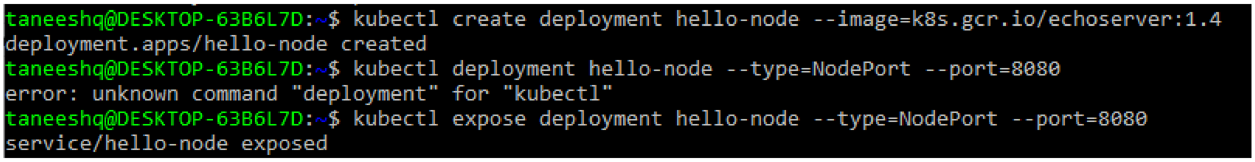
```
taneeshq@DESKTOP-63B6L7D:~$ minikube start
□  minikube v1.36.0 on Ubuntu 24.04 (amd64)
□ Automatically selected the docker driver
□  Using Docker driver with root privileges
□ For an improved experience it's recommended to use Docker Engine instead of Docker Desktop.
Docker Engine installation instructions: https://docs.docker.com/engine/install/#server
□  Starting "minikube" primary control-plane node in "minikube" cluster
□  Pulling base image v0.0.47 ...
□  Creating docker container (CPUs=2, Memory=2200MB) ...
□ Stopping node "minikube"  ...
□  Powering off "minikube" via SSH ...
□  Deleting "minikube" in docker ...
□  StartHost failed, but will try again: creating host: create host timed out in 360.000000 seconds
□  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
```

Step 4: Verify the cluster is started by listing the nodes



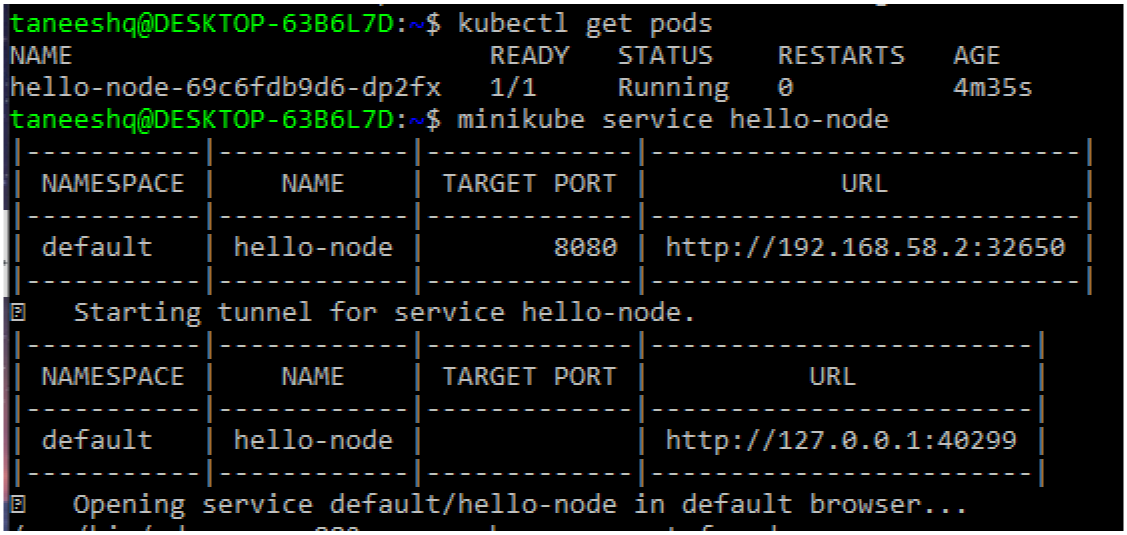
```
taneeshq@DESKTOP-63B6L7D:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   7m13s   v1.33.1
```

Step 5: Test the setup by deploying a sample image “hello-node”

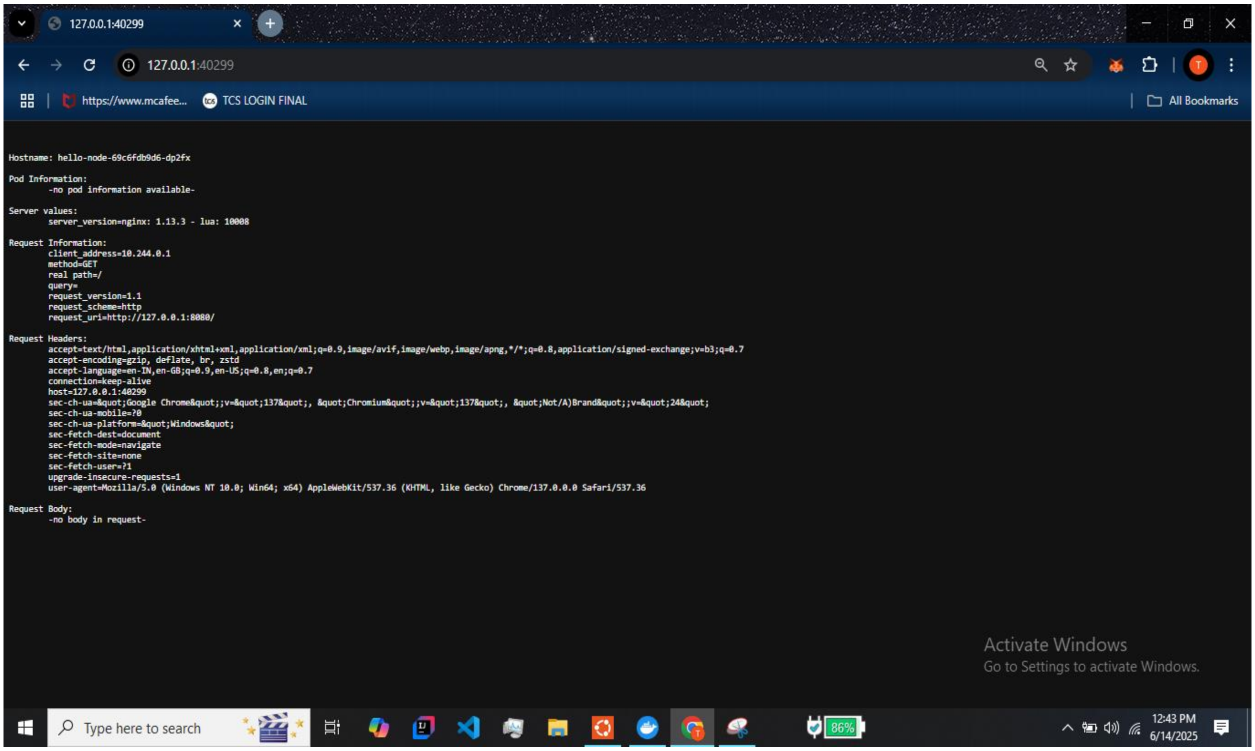


```
taneeshq@DESKTOP-63B6L7D:~$ kubectl create deployment hello-node --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-node created
taneeshq@DESKTOP-63B6L7D:~$ kubectl deployment hello-node --type=NodePort --port=8080
error: unknown command "deployment" for "kubectl"
taneeshq@DESKTOP-63B6L7D:~$ kubectl expose deployment hello-node --type=NodePort --port=8080
service/hello-node exposed
```

Step 6: Run the service to get the output. Also note the url given by minikube and verify it in browser.



```
taneeshq@DESKTOP-63B6L7D:~$ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
hello-node-69c6fdb9d6-dp2fx   1/1     Running   0          4m35s
taneeshq@DESKTOP-63B6L7D:~$ minikube service hello-node
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | hello-node |        8080 | http://192.168.58.2:32650 |
|-----------|------------|-------------|---------------------------|
□  Starting tunnel for service hello-node.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-node |             | http://127.0.0.1:40299 |
|-----------|------------|-------------|------------------------|
□  Opening service default/hello-node in default browser...
```



```
Hostname: hello-node-69c6fdb9d6-dp2fx

Pod Information:
        -no pod information available-

Server values:
        server_version=nginx: 1.13.3 - lua: 10008

Request Information:
        client_address=10.244.0.1
        method=GET
        real path=/
        query=
        request_version=1.1
        request_scheme=http
        request_uri=http://127.0.0.1:8080/

Request Headers:
        accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
        accept-encoding=gzip, deflate, br, zstd
        accept-language=en-IN,en-GB;q=0.9,en-US;q=0.8,en;q=0.7
        connection=keep-alive
        host=127.0.0.1:40299
        sec-ch-ua=&quot;Google Chrome&quot;;v=&quot;137&quot;, &quot;Chromium&quot;;v=&quot;137&quot;, &quot;Not/A)Brand&quot;;v=&quot;24&quot;
        sec-ch-ua-mobile=?0
        sec-ch-ua-platform=&quot;Windows&quot;
        sec-fetch-dest=document
        sec-fetch-mode=navigate
        sec-fetch-site=none
        sec-fetch-user=?1
        upgrade-insecure-requests=1
        user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36

Request Body:
        -no body in request-
```