

# TASK-01 KUBERNETES

**Date: 20/05/24**

**Q.1 Write a note on Kubernetes Architecture. Explain about each component of Kubernetes cluster.**

## **Kubernetes Architecture:**

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Its architecture consists of several components that work together to create and manage a distributed system of containers.

## **Key Components of Kubernetes Cluster:**

### **I. Master Node:**

- The control plane of the Kubernetes cluster.
- Manages and orchestrates the cluster's various components.
- Consists of several components:
  1. **API Server:** Exposes Kubernetes API, which clients (like kubectl) use to interact with the cluster.
  2. **Scheduler:** Assigns nodes to newly created pods based on resource requirements and other constraints.
  3. **Controller Manager:** Monitors the cluster's state and performs tasks such as node and pod management.
  4. **etcd :** A distributed key-value store that stores cluster state and configuration data.

### **II. Worker Node :**

- Runs the containers that form the application workload.
- Consists of several components:

1. **Kubelet:** Agent that runs on each node and communicates with the Kubernetes API server. It manages the node and its containers.
2. **Kube Proxy:** Maintains network rules on nodes. It handles routing of traffic to appropriate containers.
3. **Container Runtime:** Software responsible for running containers (e.g., Docker, containerd).

### III. Pods:

- A pod is the smallest deployable unit in Kubernetes.
- It represents a single instance of a running process in your cluster.
- Pods can contain one or more containers that are tightly coupled and share resources, such as networking and storage.
- They are ephemeral by nature, meaning they can be created, destroyed, and replaced dynamically.

#### Q.2 Prepare a documentation on Kubernetes setup on Ubuntu. With screenshot of each command.

- Launch 2 Instances.
- For the instance of **Master Node** choose instance type as **t2.small**
- For the instance of **Worker Node** choose instance type as **t2.micro**

(In Security Group of Master Node open port: 22, 443, 80, 8080, 179, 2379, 10250, 6443)

#### On Master node & Worker node:

- `sudo apt-get update -y`

```
get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/restricted Translation-en [18.7 kB]
get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [31.7 kB]
get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [10.1 kB]
get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [18.8 kB]
get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [6504 B]
get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [112 B]
get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [112 B]
get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [5812 B]
get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [2152 B]
get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [112 B]
get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [112 B]
get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [112 B]
Fetched 28.3 MB in 6s (5062 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-86-218: $
```

➤ `sudo apt-get install docker.io -y`

```
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu4) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
ubuntu@ip-172-31-86-218: $
```

# Download the latest release with the command

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

```
ubuntu@ip-172-31-86-218: $ curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
-bash: https://dl.k8s.io/release/stable.txt: No such file or directory
% Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 138 100 138    0    0 1793    0 --:--:-- --:--:-- --:--:-- 1793
100 212 100 212    0    0 1373    0 --:--:-- --:--:-- --:--:-- 1373
ubuntu@ip-172-31-86-218: $

No containers need to be restarted.

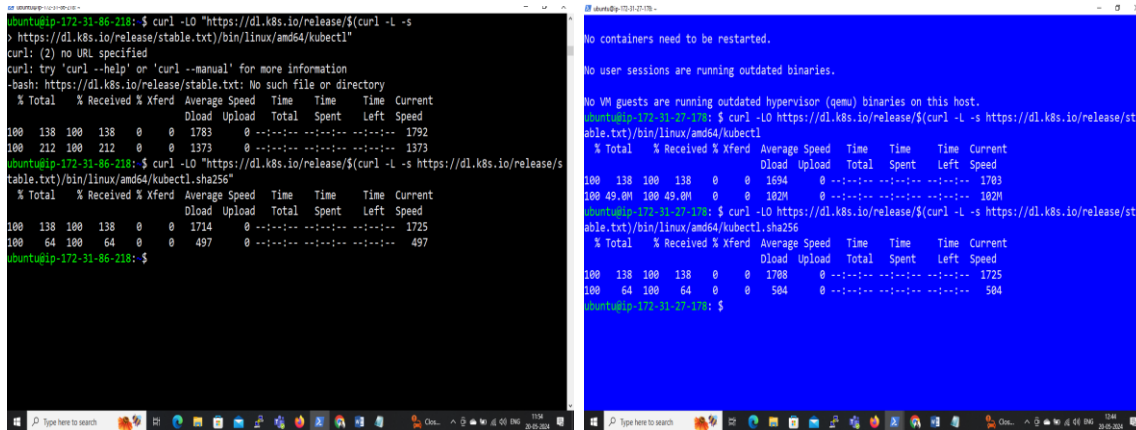
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-178: $ 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     Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 138 100 138    0    0 1694    0 --:--:-- --:--:-- --:--:-- 1703
100 49.0M 100 49.0M    0    0 102M    0 --:--:-- --:--:-- --:--:-- 102M
ubuntu@ip-172-31-27-178: $
```

# Validate the binary (optional)

# Download the kubectl checksum file:

`curl -LO https://dl.k8s.io/release/\$\(curl -L -s https://dl.k8s.io/release/stable.txt\)/bin/linux/amd64/kubectl.sha256`

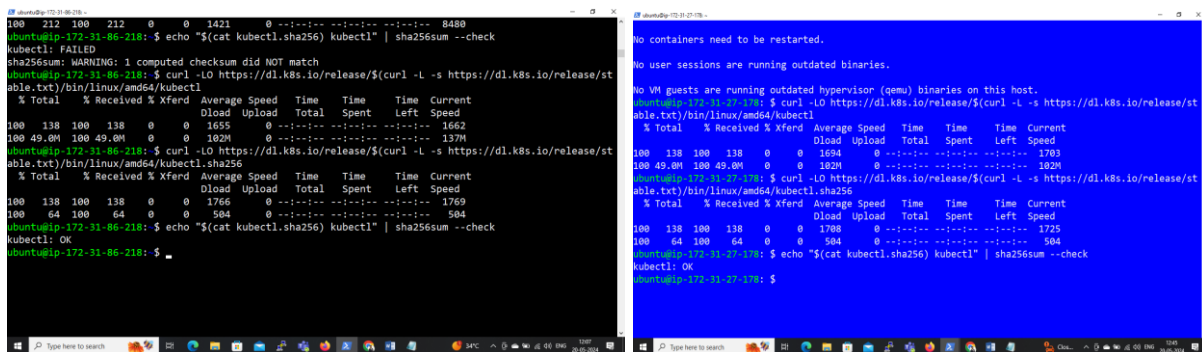


```
ubuntu@ip-172-31-86-218:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl"
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
-bash: https://dl.k8s.io/release/stable.txt: No such file or directory
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1783 0 --:--:-- --:--:-- --:--:-- 1783
100 212 100 212 0 0 1373 0 --:--:-- --:--:-- --:--:-- 1373
ubuntu@ip-172-31-86-218:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256"
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1714 0 --:--:-- --:--:-- --:--:-- 1714
100 64 100 64 0 0 497 0 --:--:-- --:--:-- --:--:-- 497
ubuntu@ip-172-31-86-218:~$

ubuntu@ip-172-31-27-178:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-178:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1708 0 --:--:-- --:--:-- --:--:-- 1725
100 64 100 64 0 0 504 0 --:--:-- --:--:-- --:--:-- 504
ubuntu@ip-172-31-27-178:~$
```

## # Validate the kubect. binary against the checksum file:

- echo "\$(cat kubectl.sha256) kubectl" | sha256sum --check



```
ubuntu@ip-172-31-86-218:~$ echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
kubectl: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
ubuntu@ip-172-31-86-218:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1655 0 --:--:-- --:--:-- --:--:-- 1662
100 49.0M 100 49.0M 0 0 102M 0 --:--:-- --:--:-- --:--:-- 137M
ubuntu@ip-172-31-86-218:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1766 0 --:--:-- --:--:-- --:--:-- 1769
100 64 100 64 0 0 504 0 --:--:-- --:--:-- --:--:-- 504
ubuntu@ip-172-31-86-218:~$ echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
kubectl: OK
ubuntu@ip-172-31-86-218:~$

ubuntu@ip-172-31-27-178:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-178:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1694 0 --:--:-- --:--:-- --:--:-- 1703
100 49.0M 100 49.0M 0 0 102M 0 --:--:-- --:--:-- --:--:-- 102M
ubuntu@ip-172-31-27-178:~$ curl -LO https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Time Current
100 138 100 138 0 0 1788 0 --:--:~ --:~:~ --:~:~ 1725
100 64 100 64 0 0 504 0 --:~:~ --:~:~ --:~:~ 504
ubuntu@ip-172-31-27-178:~$ echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
kubectl: OK
ubuntu@ip-172-31-27-178:~$
```

## # Install kubect.

- sudo install -o root -g root -m 0755 kubect. /usr/local/bin/kubect.

## # Test to ensure the version you installed is up-to-date:

- kubect. version --client

```
ubuntu@ip-172-31-86-218: ~  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           Dload  Upload   Total      Spent      Left   Speed  
100  138    100  138    0    0   1766      0 --:--:-- --:--:-- --:--:--  1769  
100   64    100   64    0    0    504      0 --:--:-- --:--:-- --:--:--   504  
ubuntu@ip-172-31-86-218:~$ echo "$(cat kubect1.sha256) kubect1" | sha256sum --check  
kubect1: OK  
ubuntu@ip-172-31-86-218:~$ sudo install -o root -g root -m 0755 kubect1 /usr/local/bin/kubect1  
ubuntu@ip-172-31-86-218:~$ kubect1 version --client  
Client Version: v1.30.1  
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3  
ubuntu@ip-172-31-86-218:~$
```

**# Update the apt package index and install packages needed to use the Kubernetes apt repository: `sudo apt-get update`**

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

```
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3  
ubuntu@ip-172-31-86-218:~$ sudo apt-get update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Reading package lists... Done  
ubuntu@ip-172-31-86-218:~$ sudo apt-get install -y apt-transport-https ca-certificates curl  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
ca-certificates is already the newest version (20240203).  
ca-certificates set to manually installed.  
The following additional packages will be installed:  
  libcurl3t64-gnutls libcurl4t64  
The following NEW packages will be installed:  
  apt-transport-https  
The following packages will be upgraded:  
  curl libcurl3t64-gnutls libcurl4t64  
3 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.  
Need to get 904 kB of archives.  
After this operation, 35.8 kB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.1  
4build2 [3974 B]  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu1
```

**# Download the public signing key for the Kubernetes package repositories.**  
**The same signing key is used for all repositories so you can disregard the version in the URL:**

- `curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg`
- `sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg`

**# Add the appropriate Kubernetes apt repository. If you want to use Kubernetes version different than v1.30, replace v1.30 with the desired minor version in the command below:**

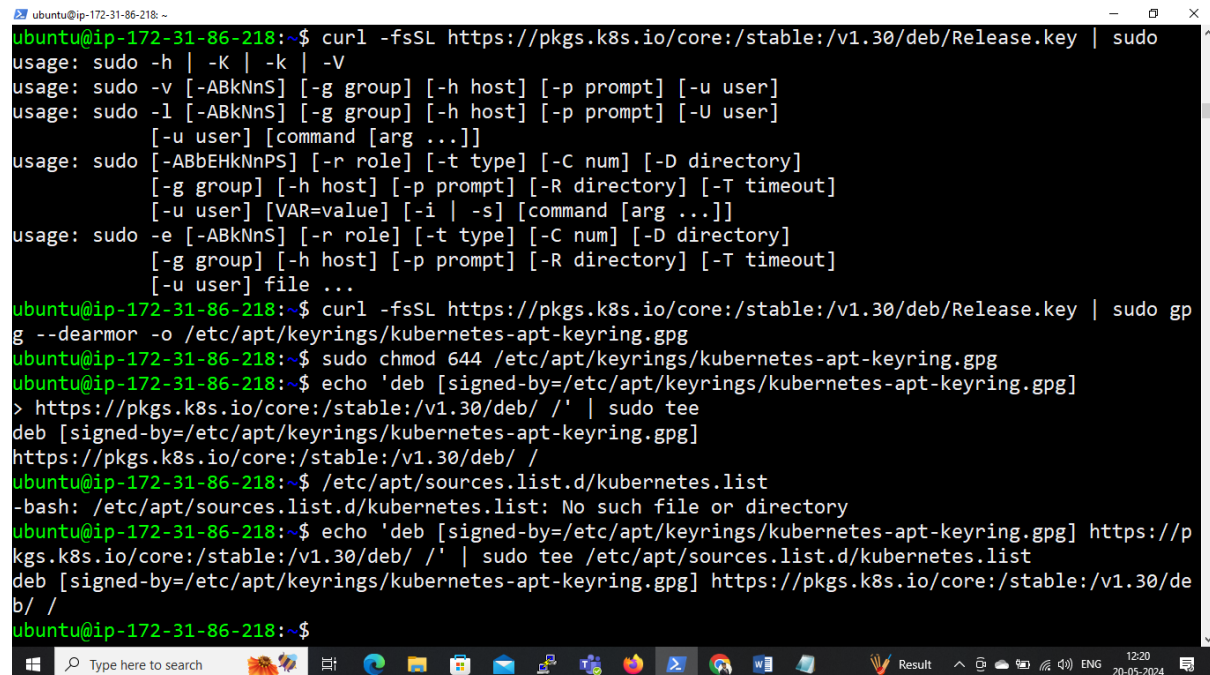
```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.30/deb/ ' | sudo tee
```

```
/etc/apt/sources.list.d/kubernetes.list
```

```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /
```



```
ubuntu@ip-172-31-86-218:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-86-218:~$ sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-86-218:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /
ubuntu@ip-172-31-86-218:~$ sudo apt-get update
```

- `sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list`

**# Update apt package index, then install kubectl, kubeadm and kubelet:**

- `sudo apt-get update`

➤ `sudo apt-get install -y kubectl kubeadm kubelet`

```
ubuntu@ip-172-31-86-218:~$ sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list
ubuntu@ip-172-31-86-218:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb InRelease
[1186 B]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb Packages
[3957 B]
Fetched 5143 B in 1s (6787 B/s)
Reading package lists... Done
ubuntu@ip-172-31-86-218:~$ sudo apt-get install -y kubectl kubeadm kubelet
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 18 not upgraded.
Need to get 93.9 MB of archives.
After this operation, 343 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1
```

# Start the kublet service and enable it:

➤ `sudo systemctl enable kubelet && sudo systemctl start kubelet`

```
ubuntu@ip-172-31-27-178:~$ sudo systemctl enable kubelet && sudo systemctl start kubelet
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.30.1-1.1) ...
Setting up ebtables (2.0.11-6build1) ...
Setting up socat (1.8.0.0-4build3) ...
Setting up cri-tools (1.30.0-1.1) ...
Setting up kubernetes-cni (1.4.0-1.1) ...
Setting up kubeadm (1.30.1-1.1) ...
Setting up kubelet (1.30.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
ubuntu@ip-172-31-27-178:~$ sudo systemctl enable kubelet && sudo systemctl start kubelet
ubuntu@ip-172-31-27-178:~$
```

(Above all commands run on both – Master and **Worker** nodes)



## On Master:

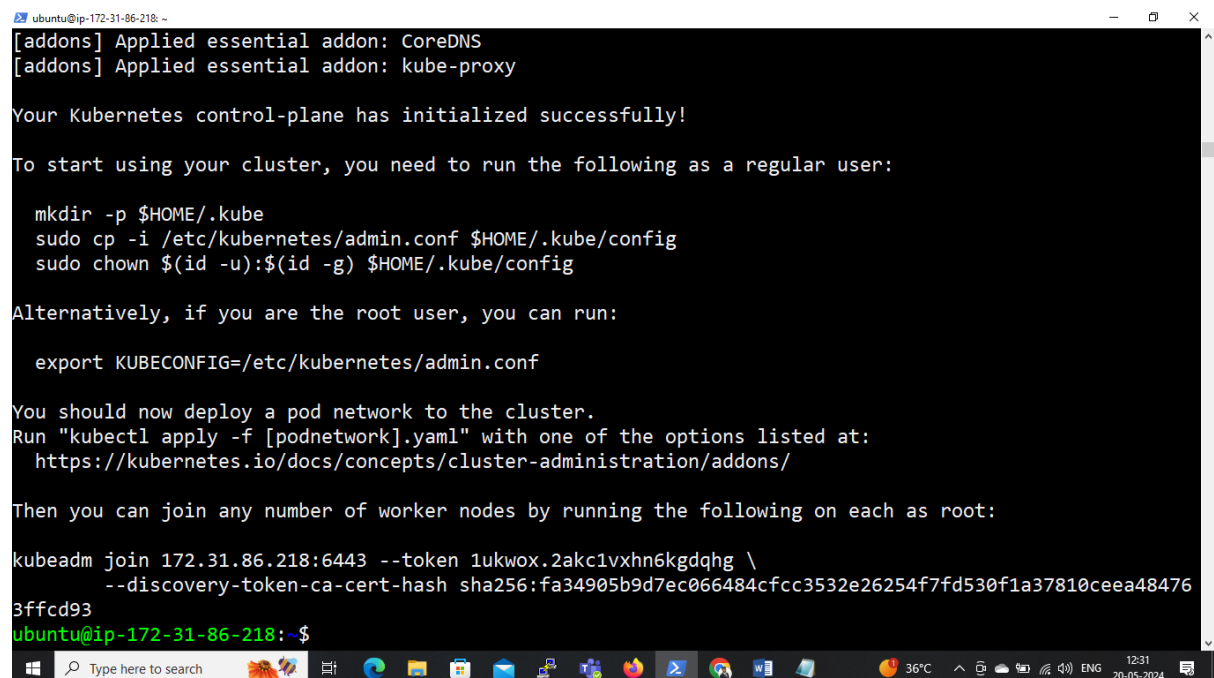
### # Initialize kubeadm

➤ `sudo kubeadm init --ignore-preflight-errors=all`

# After running this command we get the kube api token which shows below:

```
kubeadm join 172.31.86.218:6443 --token lukwox.2akc1vxhn6kgdqhg \  
--discovery-token-ca-cert-hash  
sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd  
93
```

### Save this token somewhere in our machine



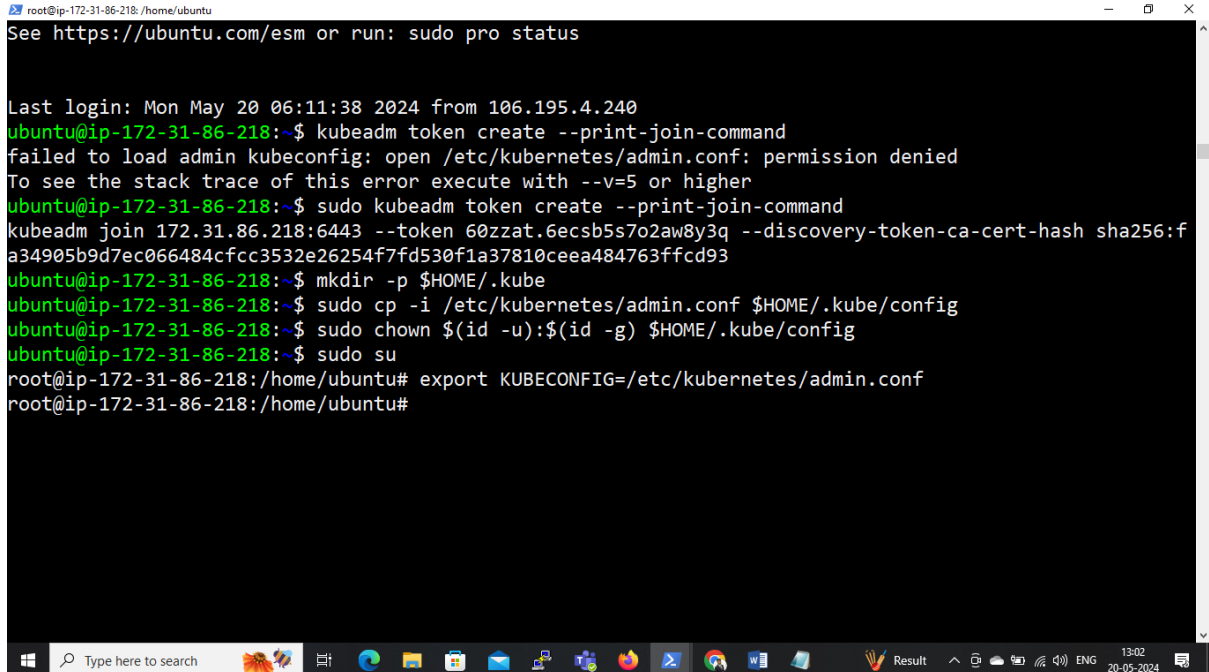
```
ubuntu@ip-172-31-86-218: ~  
[addons] Applied essential addon: CoreDNS  
[addons] Applied essential addon: kube-proxy  
  
Your Kubernetes control-plane has initialized successfully!  
  
To start using your cluster, you need to run the following as a regular user:  
  
  mkdir -p $HOME/.kube  
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
  sudo chown $(id -u):$(id -g) $HOME/.kube/config  
  
Alternatively, if you are the root user, you can run:  
  
  export KUBECONFIG=/etc/kubernetes/admin.conf  
  
You should now deploy a pod network to the cluster.  
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:  
  https://kubernetes.io/docs/concepts/cluster-administration/addons/  
  
Then you can join any number of worker nodes by running the following on each as root:  
  
kubeadm join 172.31.86.218:6443 --token lukwox.2akc1vxhn6kgdqhg \  
--discovery-token-ca-cert-hash sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd93  
ubuntu@ip-172-31-86-218:~$
```

Master node:-

- `mkdir -p $HOME/.kube`
- `sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`
- `sudo chown $(id -u):$(id -g) $HOME/.kube/config`



- sudo su
- export KUBECONFIG=/etc/kubernetes/admin.conf
- exit

A terminal window titled 'root@ip-172-31-86-218: /home/ubuntu' showing a sequence of commands and their outputs. The user attempts to create a token with 'kubectl token create', which fails due to permission issues. They then use 'sudo kubectl token create' to generate a token. Next, they create a directory '.kube' in their home directory, copy the admin.conf file to it, and change its ownership to the user. Finally, they use 'sudo su' to become root and export the KUBECONFIG variable to the path of the copied file.

```
root@ip-172-31-86-218: /home/ubuntu
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon May 20 06:11:38 2024 from 106.195.4.240
ubuntu@ip-172-31-86-218:~$ kubectl token create --print-join-command
failed to load admin kubeconfig: open /etc/kubernetes/admin.conf: permission denied
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-86-218:~$ sudo kubectl token create --print-join-command
kubectl join 172.31.86.218:6443 --token 60zzat.6ecsb5s7o2aw8y3q --discovery-token-ca-cert-hash sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd93
ubuntu@ip-172-31-86-218:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-86-218:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-86-218:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-86-218:~$ sudo su
root@ip-172-31-86-218:/home/ubuntu# export KUBECONFIG=/etc/kubernetes/admin.conf
root@ip-172-31-86-218:/home/ubuntu#
```

## # To install Calico on Master

- kubectl create -f <https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/tigera-operator.yaml>
- kubectl create -f <https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/custom-resources.yaml>

```
ubuntu@ip-172-31-86-218:~$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/tigera-operator.yaml
namespace/tigera-operator created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/apiservers.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/imagesets.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/installations.operator.tigera.io created
```

## # On master node check status & get nodes:

- kubectl get componentstatus
- kubectl get nodes

```
ubuntu@ip-172-31-86-218:~$ kubectl get componentstatus
error: unknown command "componentstatus" for "kubectl"
ubuntu@ip-172-31-86-218:~$ kubectl get nodes
NAME                 STATUS    ROLES    AGE   VERSION
ip-172-31-27-178     Ready    <none>    32m   v1.30.1
ip-172-31-86-218     Ready    control-plane 64m   v1.30.1
ubuntu@ip-172-31-86-218:~$ kubectl componentstatus
Warning: v1 ComponentStatus is deprecated in v1.19+
NAME                STATUS    MESSAGE    ERROR
controller-manager  Healthy   ok
scheduler           Healthy   ok
etcd-0              Healthy   ok
ubuntu@ip-172-31-86-218:~$ kubectl get nodes
NAME                 STATUS    ROLES    AGE   VERSION
ip-172-31-27-178     Ready    <none>    33m   v1.30.1
ip-172-31-86-218     Ready    control-plane 65m   v1.30.1
ubuntu@ip-172-31-86-218:~$
```

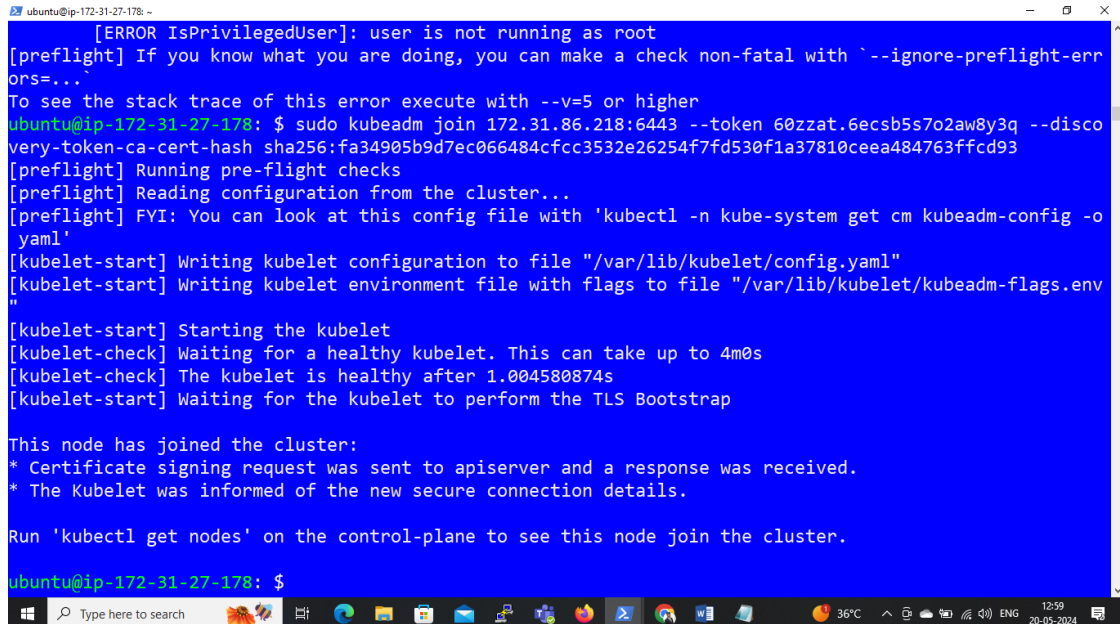
## On Worker:

```
sudo kubeadm join 172.31.86.218:6443 --token 60zzat.6ecsb5s7o2aw8y3q --disco
```

very-token-ca-cert-hash

sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd

93

A terminal window screenshot showing the execution of the 'sudo kubeadm join' command. The terminal output includes preflight checks, kubelet configuration, and a confirmation that the node has joined the cluster. The command used is 'sudo kubeadm join 172.31.86.218:6443 --token 60zzat.6ecsb5s7o2aw8y3q --discovery-token-ca-cert-hash sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd93'. The output shows the kubelet starting and becoming healthy, and the node joining the cluster.

```
ubuntu@ip-172-31-27-178: ~$ sudo kubeadm join 172.31.86.218:6443 --token 60zzat.6ecsb5s7o2aw8y3q --discovery-token-ca-cert-hash sha256:fa34905b9d7ec066484cfcc3532e26254f7fd530f1a37810ceea484763ffcd93
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.004580874s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

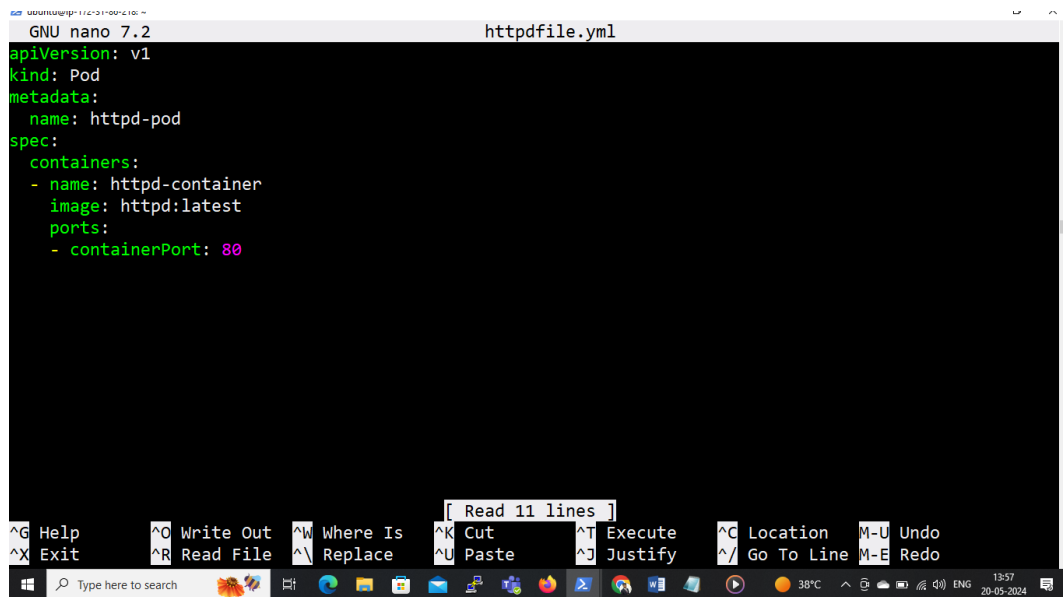
ubuntu@ip-172-31-27-178: ~$
```

**Q.3 Write a manifest file to create an httpd container in Pod, create pod using that manifest file. Also go inside that httpd container and create own pages and try to access those pages from inside the container.**

**On Master node :**

**# Create a file ,**

➤ **nano httpdfile.yml**



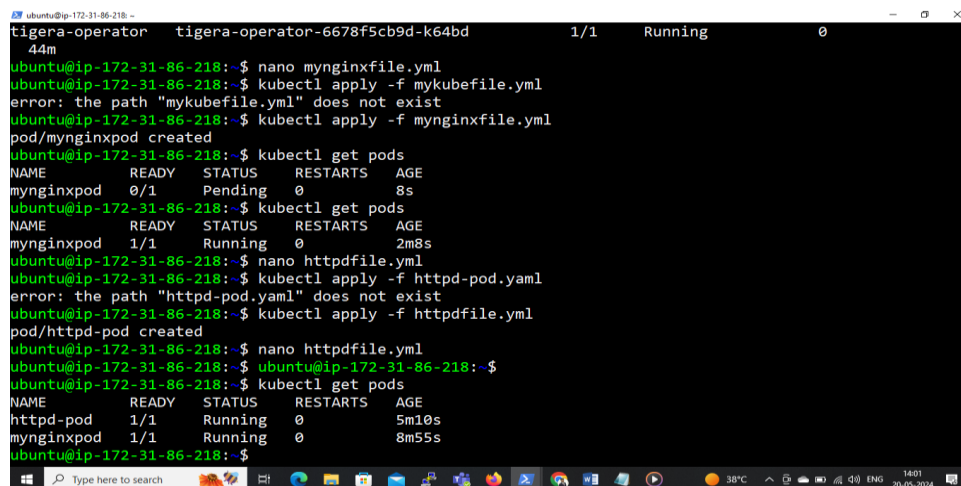
```
GNU nano 7.2 httpdfile.yml
apiVersion: v1
kind: Pod
metadata:
  name: httpd-pod
spec:
  containers:
  - name: httpd-container
    image: httpd:latest
    ports:
    - containerPort: 80
```

**# To create the Pod using this manifest file, we can run the following command:**

- `kubectl apply -f httpdfile.yml`

**# Command to check the pods**

- `kubectl get nodes`

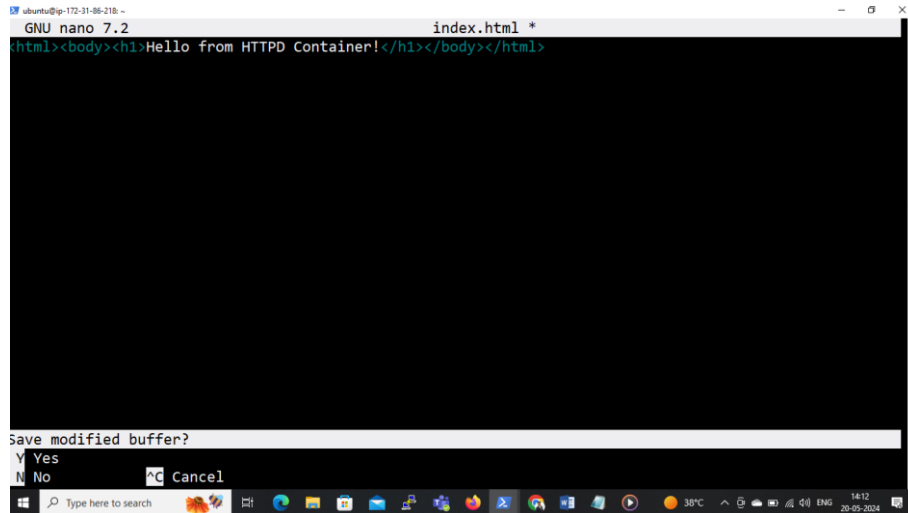


```
tigera-operator tigera-operator-6678f5cb9d-k64bd 1/1 Running 0
44m
ubuntu@ip-172-31-86-218:~$ nano mynginxfile.yml
ubuntu@ip-172-31-86-218:~$ kubectl apply -f mykubefile.yml
error: the path "mykubefile.yml" does not exist
ubuntu@ip-172-31-86-218:~$ kubectl apply -f mynginxfile.yml
pod/mynginxpod created
ubuntu@ip-172-31-86-218:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mynginxpod    0/1     Pending   0           8s
ubuntu@ip-172-31-86-218:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mynginxpod    1/1     Running   0          2m8s
ubuntu@ip-172-31-86-218:~$ nano httpdfile.yml
ubuntu@ip-172-31-86-218:~$ kubectl apply -f httpd-pod.yaml
error: the path "httpd-pod.yaml" does not exist
ubuntu@ip-172-31-86-218:~$ kubectl apply -f httpdfile.yml
pod/httpd-pod created
ubuntu@ip-172-31-86-218:~$ nano httpdfile.yml
ubuntu@ip-172-31-86-218:~$ ubuntu@ip-172-31-86-218:~$
ubuntu@ip-172-31-86-218:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
httpd-pod     1/1     Running   0          5m10s
mynginxpod    1/1     Running   0          8m55s
ubuntu@ip-172-31-86-218:~$
```

**# Once the Pod is running, We can enter in container & create HTML page.**

- `kubectl exec -it httpd-pod /bin/bash`
- `cd /usr/local/apache2/htdocs`

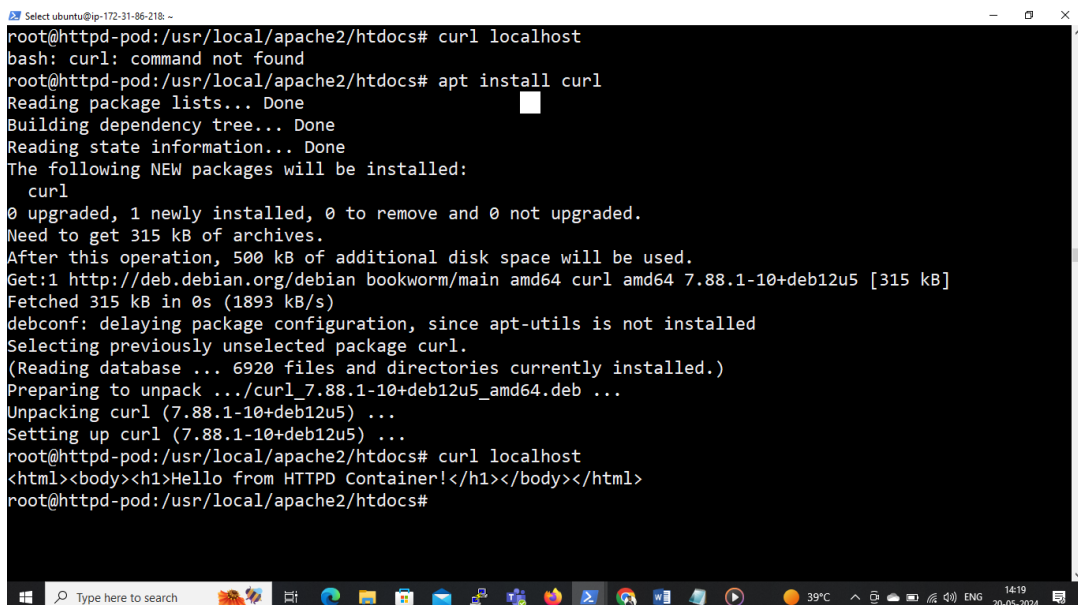
- apt update
- apt install nano
- nano index.html



```
ubuntu@ip-172-31-86-218:~  
GNU nano 7.2 index.html *  
<html><body><h1>Hello from HTTPD Container!</h1></body></html>  
Save modified buffer?  
Y Yes  
N No  
^C Cancel
```

**# To access these pages from inside the container, we can use following command:**

- apt install curl
- curl localhost



```
Select ubuntu@ip-172-31-86-218:~  
root@httpd-pod:/usr/local/apache2/htdocs# curl localhost  
bash: curl: command not found  
root@httpd-pod:/usr/local/apache2/htdocs# apt install curl  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  curl  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 315 kB of archives.  
After this operation, 500 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-10+deb12u5 [315 kB]  
Fetched 315 kB in 0s (1893 kB/s)  
debconf: delaying package configuration, since apt-utils is not installed  
Selecting previously unselected package curl.  
(Reading database ... 6920 files and directories currently installed.)  
Preparing to unpack .../curl_7.88.1-10+deb12u5_amd64.deb ...  
Unpacking curl (7.88.1-10+deb12u5) ...  
Setting up curl (7.88.1-10+deb12u5) ...  
root@httpd-pod:/usr/local/apache2/htdocs# curl localhost  
<html><body><h1>Hello from HTTPD Container!</h1></body></html>  
root@httpd-pod:/usr/local/apache2/htdocs#
```

